ETP-KA/2025-02

Karlsruher Institut für Technologie

Optimized Utilization of HPC Centers for High-Energy Physics Workflows and Jet Energy Corrections for the CMS Experiment

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat)

von der KIT-Fakultät für Physik des
Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M.Sc. Robin Hofsaess
aus Villingen-Schwenningen

Tag der mündlichen Prüfung: 25.04.2025

Referent: Prof. Dr. Günter Quast

Korreferent: Prof. Dr. Achim Streit

Betreuer: Dr. Manuel Giffels

Keywords: LHC, CMS, QCD, Jet Energy Calibration, Absolute Residual Corrections, WLCG, Distributed Computing, HPC, Sustainability, XRootD

I declare that everything, I have done is reproducible. I do not forgot or hid any information necessary that is important to understand the results.

Karlsruhe, 26.02.2025

. . . .

(M.Sc. Robin Hofsaess)

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 26.02.2025

. . . .

(M.Sc. Robin Hofsaess)

Abstract

This thesis presents a comprehensive study that bridges experimental High-Energy Physics with innovative strategies for a more sustainable future of the LHC computing in Germany and beyond. In the physics domain, the work describes state-of-the-art techniques for precise Jet Energy Calibration for the CMS experiment, with a key contribution being the first full derivation of the absolute residual Jet Energy Corrections for Run 2 Ultra Legacy data. The developed tools and methods were carefully adapted in close collaboration with the JERC group of the CMS Collaboration to meet the latest reprocessing requirements. The results were handed over as the final contribution to the Jet Energy Calibration effort from the KIT-CMS group. Comprehensive analyses demonstrated that the provided corrections minimize the discrepancies between data and simulation to sub-percent levels over large regions of the most relevant phase space, laying ground for high-precision QCD studies and searches for physics beyond the Standard Model – which both require highest accuracy in jet energy measurements. In addition, a synchronization process between the well-established MiniAOD-based calibration frameworks employed for Run 2 and the new NanoAOD tools for the Run 3 calibrations presented a full knowledge transfer into the collaboration and ensured consistent results of the next generation calibration workflows.

Complementing the experimental achievements, the thesis also tackles the increasing computational challenges inherent in modern High-Energy Physics (HEP) on the brink of the HL-LHC era. The sheer amount of data and the increasing complexity of the employed analyses and methods in combination with growing requirements on economical and ecological sustainability present major challenges for the future of world's most sophisticated scientific computing grid – the Worldwide LHC Computing Grid (WLCG). Starting with an in-depth review of this collaborative distributed computing infrastructure, this thesis provides a comprehensive introduction to the LHC computing, including the dynamic integration of (opportunistic) resources. Especially the integration of the HoreKa HPC cluster as part of the upcoming HEP computing strategy in Germany, focusing on sustainability and the utilization of HPC resources in the future, is addressed. By proposing new evaluation metrics, this work lays the groundwork for evaluating and optimizing the grid integration of such resources. A notable innovation is the XBuffer concept, an XRootd-based approach designed to mitigate bottlenecks in HPC resource utilization. Despite initial challenges, such as increased failure rates and timeouts, subsequent modifications and optimizations of the concept led to a prototype setup that not only improved reliability and efficiency but also allowed the HPC center to compete with traditional grid sites in reliability and efficiency. This optimized integration offers a promising outlook for the cooperation between the HEP community and national HPC centers, while also highlighting specific areas for further optimization.

By interweaving precise Jet Energy Calibrations with strategic advancements in scientific computing, this work not only supports the investigation of fundamental interactions at highest accuracy but also paves the way for a more sustainable and efficient computing infrastructure, ultimately bolstering the long-term research capabilities of High-Energy Physics at the LHC.

Reader's Guide

This thesis covers two important aspects of modern High-Energy Physics (HEP) that are inseparably linked: data-intensive high-precision measurements and effective data processing within the Worldwide LHC Computing Grid (WLCG).

From a physicist's point of view, the enormous amount of data recorded at the Large Hadron Collider (LHC) at CERN offers great opportunities. Thanks to high-precision calibrations, it is possible to use the data to test today's established theoretical models for the smallest deviations and thus confirm the theories or pave the way for new physics beyond the Standard Model (SM). However, from the computing perspective, it also presents major challenges when it comes to dealing with the data volumes in the exabyte era of High-Energy Physics. To realize this, enormous storage and computing resources are required as the other side of the medal. Providing sufficient resources for thousands of scientist all around the world is therefore a highly demanding and complex task – and incredibly important to make today's science progress possible! Efficient and sustainable use of the given resources is therefore an absolute necessity in order to be able to meet the ever-increasing demand while at the same time maintaining the social and ecological responsibility of research – which the HEP community is very aware of.

Throughout this thesis, both sides of this medal are addressed. It focuses on advanced techniques for the Jet Energy Calibration (JEC) for the Compact Muon Solenoid (CMS) experiment to being able to reach highest precision in physics analyses and explores the computational challenges and solutions needed to prepare the computing landscape in High-Energy Physics for the future.

How to Use This Guide:
This Reader's Guide is designed to help you navigate through the structure of the thesis and to decide for a reading path based on the individual background, experience, and topics of interest. Below, the individual chapters are described in detail and my contributions to the different topics are highlighted. After the overview, a recommendation for each chapter is provided.

Overview

Chapter 1: Introduction
The introduction sets the stage by outlining the two research problems, objectives, and the significance of the studies in both physics and computing. At first, the relevance of precise jet measurements in modern High-Energy Physics is discussed. With sophisticated Jet Energy Correction methods utilized at CMS, highest precision is ensured and a thorough testing of the Standard Model is enabled.

In terms of computing, the necessity of an efficient and sustainable utilization of the given resources is emphasized. Furthermore, the requirements on meaningful evaluation techniques as well as optimization strategies for the future strategy are described.

Chapter 2: Jet Energy Calibration for the CMS Experiment at $\sqrt{s}$=13 TeV
The first part of this chapter provides a detailed introduction to the physics part of this work. It highlights the necessity of accurate measurements of jets – collimated particle streams emerging from the high-energy collisions of hadrons at the LHC – particularly for high-precision QCD studies, which are usually in need of accurate jet energy measurements to investigate the properties of the strong force in detail. After describing the basic physics concept behind the formation and measurement of such jets and the CMS detector used to measure them, the sophisticated methods for the Jet Energy Calibration developed and utilized by the CMS Collaboration are explained. These methods are the back-bone of well calibrated datasets that are ultimately used for leading research carried out by scientists all around the world.

This chapter is therefore recommended for readers seeking an understanding of the methods employed for the Jet Energy Calibration at the CMS experiment and the underlying physics concepts.

Chapter 3: First Derivation of the Absolute Residual Jet Energy Corrections for Ultra Legacy Data with Z+Jet Events
In this chapter, the first derivation of the absolute residual Jet Energy Corrections for the full data measured with the CMS detector during Run 2 (2016-2018) with the latest – so-called Ultra Legacy – reprocessing of the datasets is presented. These corrections are minimizing the remaining discrepancies after all other correction steps have been applied and are therefore of outmost importance for guaranteeing the highest precision of jet measurements. For deriving the absolute residual corrections, the software frameworks were updated and adapted to the requirements and recommendations for the new datasets in collaboration with the responsible sub-group of the CMS Collaboration. These adjustments are described in the first part, addressing the data and event selection as well as additional pre-processing steps, which are relevant for the procedure. Afterwards, the results are presented and discussed. Furthermore, as part of this thesis, contributions to the preparations of the Jet Energy Calibration for Run 3 were made. These mainly comprised a framework synchronization, as well as a complete knowledge transfer to the successor group taking over the calibration for Run 3. The last sub-section provides a short summary and conclusion on the derivation and the Run 3 preparations.

This chapter is recommended for readers that are interested in the actual derivation and the current status of the Level-3 absolute residual corrections as part of the entire Jet Energy Calibration procedure for the CMS experiment.

Chapter 4: Computing in High-Energy Physics
This chapter provides an introduction and comprehensive review of the ideas and key aspects of HEP computing in general, and the WLCG in particular. The concept and structure of world's largest scientific computing grid are described in the beginning. This part is first addressing the provisioning of different types of compute resources, including the tools and solutions to provide a standardized grid environment on arbitrary resources – hence, the main technologies that are necessary to run a complex infrastructure like the WLCG. Second, data management and handling is explained, focusing on XRootD as a key technology for data storage and transfers in a worldwide storage federation. Finally, monitoring, benchmarking, and accounting, which are extremely relevant topics for the well-functioning of a global collaborative

computing infrastructure, are discussed.
The second part of the chapter is an introduction to the dynamic integration of opportunistic resources. It provides details on the concept and technologies that enable the utilization of divers resources. Additionally, HoreKa, the scientific HPC cluster at KIT, is introduced and its opportunistic integration is described in detail to provide practical insights on the aforementioned principles and tools.

This chapter is recommended for readers that are new to the field of HEP computing or interested in the tools and backgrounds in general. The relations and all necessary basic concepts and information are described that are relevant for understanding the rest of this work.

Chapter 5: The Pledged Integration of HPC Centers
Chapter 5 is dedicated to the future utilization of HPC centers in the WLCG and the evaluation of their use. In the first part, the current situation in the WLCG and in Germany are described. Afterwards, the future German HEP computing strategy is presented, motivated, and analyzed in detail. Here, in-depth changes are planned in the upcoming years, including a gradual consolidation of compute and storage resources and an increased utilization of HPC centers for providing the official German contributions to the WLCG. While this strategy is expected to be highly beneficial for the future, it also bears significant challenges and hurdles, since it is completely new terrain. These are discussed and the goals for the future integration of HPC resources are formulated.
The second part of the chapter focuses on the evaluation of integrated HPC sites, including an answer to the question on how the integration of grid resources can be better assessed in general. This is an important topic, since the traditional quantities for reviewing computing resources are often not sufficiently meaningful to realize a conclusive evaluation. For being able to better rate the new strategy and concepts, additional, more conclusive and expressive measures for their evaluation were developed in the course of this work. Additionally, a newly developed data-driven method for the determination of the pure performance of a grid site is presented. Ultimately, these measures are employed exemplary in the last part of the chapter to investigate the opportunistic integration of HoreKa – and if the self-set goals are in reach.

This chapter is interesting for everyone interested in the future German HEP computing strategy, including its motivation, challenges, and implementation. Furthermore, new, more expressive quantities for the evaluation of grid resources are introduced, which highly focus on sustainability. These quantities are the basis for (performance) comparisons of sites throughout the following practical part of the thesis and therefore essential for the final assessment of the HPC integration at the end.

Chapter 6: Optimizations for the Utilization of HPC Resources
With the analysis based on the newly introduced methods in the previous chapter, weak points of the integration of HoreKa as an opportunistic resource were revealed. However, due to limited monitoring capabilities at HPC centers, the identification of the actual bottlenecks and limitations is complicated. The beginning of chapter 6 is therefore attributed to this challenge. Based on the findings, mitigation and optimization strategies are discussed, eventually leading to an XRootD-based concept, which was developed for the optimized utilization of HPC centers for HEP work-

flows, called XBuffer. It combines different ideas for optimizations and provides further benefits for the integration of HPC centers as grid resources. A prototype was deployed at the HoreKa HPC cluster as a Proof of Concept, which is evaluated based on the traditional and the newly introduced measures during the chapter. Based on the knowledge gained, further optimizations are proposed which are capable of resolving the remaining weak points of the concept to guarantee a stable and efficient utilization of HPC centers as grid resources in the future.

This chapter is recommended for readers that are interested in optimization strategies for a more reliable integration of HPC centers focusing on sustainability. Furthermore, it shows the value of the newly introduced evaluation measures enabling a conclusive rating of the deployed prototype.

Chapter 7: Summary and Outlook
The final chapter briefly summarizes the results of the work and their relevance and provides a brief outlook on future developments.

Contents

Miscellaneous

List of Figures

List of Tables

1. Introduction

> The universe consists of seven regions: North, South, West, East, Before, After, and Home.
>
> (The 13½ Lives of Captain Bluebear – Walter Moers)

The Standard Model of particle physics (SM) is the most successful theory in high-energy physics today. With the discovery of the Higgs boson, the last missing piece of the SM was verified in 2012 [1, 2], marking a monumental milestone in the understanding of the fundamental particles and forces. However, this achievement does not signal the end of particle physics. Instead, it rather opens up two crucial research directions: rigorously testing the SM and searching for deviations that might indicate new physics beyond the well-established Standard Model (BSM).

Precision is the foundation for both of these endeavors. Accurate measurements are essential for exploring physics, as even smallest discrepancies between theoretical predictions and experimental results can provide hints for new phenomena. Achieving the required precision demands not only high-quality data but also vast amounts of it. This necessity requires the use of high-energy, high-luminosity hadron colliders, such as the Large Hadron Collider (LHC) at CERN, which generates enormous datasets of recorded particle collisions. However, these large datasets introduce significant challenges.

One major challenge arising from the high energies and the hadronic nature of the interactions is the precise measurement of so-called jets produced in such collisions. Jets are collimated streams of particles that emerge from the fragmentation and hadronization of quarks and gluons generated in hard scattering processes between the accelerated protons. The high-energy, high-luminosity environment gives rise to complex phenomena, like event overlay (pileup) or intricate Quantum Chromodynamics (QCD) effects, which complicate data interpretation. An accurate measurement of jets, however, is crucial for many analyses in high-energy physics, from precision studies of SM processes and QCD to the search for new phenomena that could indicate physics beyond the currently established theoretical framework. But these conditions directly affect their reconstruction, making corrections and a robust and precise calibration essential to ensure the reliability of measurements and enable high-precision analyses.

Consequently, a comprehensive Jet Energy Calibration (JEC) process was developed for the Compact Muon Solenoid (CMS) experiment [3] to systematically correct raw jet energies for detector response variations, pileup, and other possible sources of biases, thereby ensuring that jet measurements are both accurate and consistent across different analyses. As indicated, this process comprises a series of Jet Energy Corrections. This thesis focuses on the final high-precision step involving residual corrections. The involved data-driven methods address remaining discrepancies between the detector response in data and simulation to ensure that the calibrated jet energies accurately reflect the true particle-level energies.

In the first part of this work, the absolute residual corrections based on Z+Jet events are elaborated and the first full derivation of the correction step for the full Run 2 datasets of the CMS experiment is presented.

In addition to the challenges from the physics perspective, it is essential to also consider the substantial computational challenges involved in modern High Energy Physics (HEP). Both the calibration of measured data and its subsequent analysis demand enormous compute performance. This field of research therefore has now reached an epoch of complexity in which there can no longer be any significant progress without sufficient computing resources. As experimental luminosity continues to increase with the High-Luminosity Large Hadron Collider (HL-LHC) in the future, the requirements for storage capacity and computational power will expectedly even rise drastically.

Asking the questions about the nature of existence, which has always driven people and science, is therefore today directly coupled with the sufficient provisioning of compute power. In conclusion, this means that developments in the field of scientific computing to provide sufficient resources has become as important as the research itself, as at some point no significant progress would be possible without further innovations – also with respect to sustainability and the environmental impact of physics research. Because at the same time, the HEP community is fully aware of its responsibility towards society and the environment.

In order to reconcile the constantly growing demand for performance with the need for sustainability, it may not be enough to use the given resources responsibly but new approaches in computing must also be embraced. Although grid computing has proven itself, from the beginning of the Worldwide LHC Computing Grid (WLCG) until today, as reliable foundation of particle physics research, it may now be beneficial to look beyond traditional paradigms at the vicinity of the HL-LHC era to prepare for the expected surge in data rates.

One possible way out of this dilemma could be the consolidation of resources from many smaller grid sites into larger, more sustainable High-Performance Computing (HPC) centers – as planned with the future German HEP computing strategy decided in 2022. With this step, HEP research could benefit from the enormous performance potential of such clusters, while keeping the environmental impact of science as small as possible. This is therefore not meant as a replacement for the WLCG, but has to be understood as a supplement to increase the heavily required capabilities of HEP computing. The role of the WLCG will stay crucial And computers have thus developed from simple tools into an integral, irreplaceable part of High Energy Physics.

This approach is discussed in detail in the second part of this thesis, examining not only whether it is feasible, but also if it is sensible. Based on the multi-year experience of utilizing the scientific High-Performance Computing cluster HoreKa at Karlsruhe Institute of Technology (KIT) as an opportunistic resource, this study will examine two key aspects. Firstly, it will assess how convincingly one can evaluate the actual utilization of an (HPC) grid resource. And secondly, it explores strategies to optimize reliability, efficiency, and sustainability to pave the way for the successful utilization of HPC centers as prt of the German contribution to the WLCG in the future and to enable discoveries that could redefine the understanding of the universe.

2. Jet Energy Calibration for the CMS Experiment

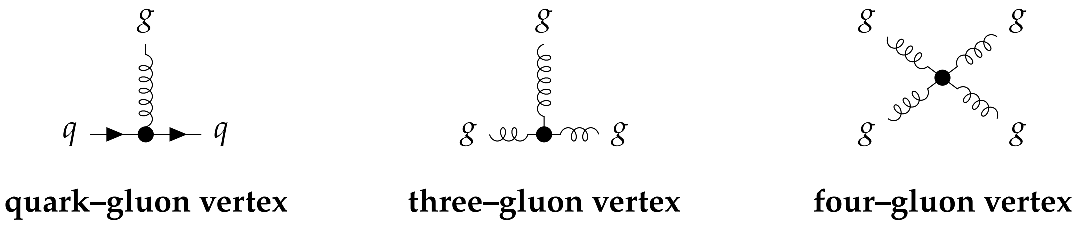
> The study of physics is also an adventure. You will find it challenging, sometimes frustrating, occasionally painful, and often richly rewarding.
>
> (Hugh D. Young)

Since the discovery of the Higgs boson, the search to uncover dark matter and explore physics beyond the SM has driven much of modern particle physics. But at the same time, high-precision measurements of fundamental physical quantities, like e.g. the properties of the Higgs boson or the coupling strength of the strong force, are still at the forefront of the HEP research agenda. Although the SM has stood the test of time, its continued success demands that the limits of precision need to be pushed further to reveal even the smallest deviations that could point towards new physics. This is essential to further complement the understanding of the Universe because other research fields, such as cosmology, provide hints on physics that is not covered by the SM, such as dark matter, neutrino masses, or the matter-antimatter asymmetry.

At hadron colliders, like the LHC, this in particular means that jets need to be detected and measured with a high accuracy. This is because these collimated particle streams, which mainly emerge from high-energy quarks or gluons originating from the hard scattering processes, are dominating the picture at such colliders. They play a central role in reconstructing the energy balance of collision events, which is key for identifying missing energy that may hint at neutrinos or potential dark matter candidates. Jets are fundamental to precision tests of QCD [4, 5] by comparing measured jet properties with theoretical models, which helps to refine the understanding of the strong interaction. In addition, jets serve as important signatures in searches for new physics, whether by directly signaling the production of new particles or by forming critical backgrounds that must be accurately modeled to reveal subtle signals.

In total, precise jet measurements are therefore essential for HEP not only for verifying theoretical predictions but also for driving discoveries, as they reduce systematic uncertainties, enhance signal–background discrimination, and ultimately enable the exploration of both the known and the unknown areas of particle physics. However, accurately measuring jets is inherently challenging. A reliable and accurate JEC is therefore required in modern particle physics experiments, ensuring that the measured energies of jets accurately reflect the underlying physics. Before delving into the concepts behind JEC, it is important to gain a general understanding of jet formation and an overview of the CMS experiment to better assess the challenges of accurate jet measurements.

In the next section, the formation process of jets and their measurement is briefly



quark–gluon vertex **three–gluon vertex** **four–gluon vertex**

Figure 2.1.: Basic QCD interaction vertices. While photons as exchange particles of the electromagnetic interaction do not carry a charge, the gluons – indicated with curly lines – carry a so-called color charge as well. As a consequence, self-interactions in form of 3g (mid) and 4g (right) vertices occur.

described. This discussion will shortly cover how QCD governs the evolution of jets, from the initial strong interaction of partons, through the hadronization of high-energy secondary quarks and gluons, to the final state hadronic jets observed in particle detector experiments. Additionally, the challenges involved in capturing these complex physics objects accurately are highlighted. After that, the CMS detector is described, including its mechanisms for jet measurement and reconstruction. The third and last part of the chapter covers the Jet Energy Corrections chain employed for the CMS experiment.

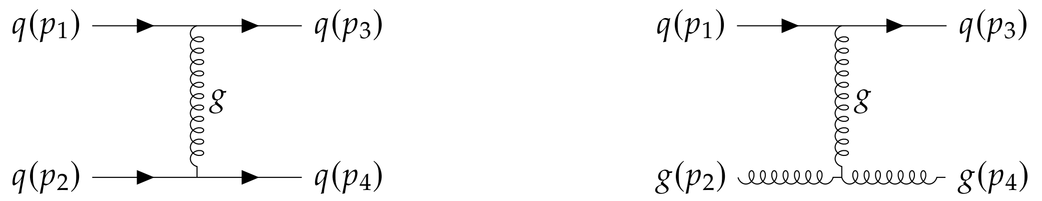
2.1. Jet Formation, Measurement, and Calibration

In HEP, jets commonly refer to collimated, cone-shaped streams of particles produced, e.g., in high-energy collisions at hadron colliders. These jets are mostly the observable signatures of the underlying hadronic interactions and subsequent QCD processes. At the LHC, the jet formation starts with the collision of two high-energetic hadrons. The basic, underlying strong interactions are depicted as Feynman diagrams [6] in Fig. 2.1.

The partons (quarks and gluons) behave like quasi-free particles in the hard scattering process at the energy scale of modern accelerators. This phenomenon is referred to as asymptotic freedom [7, 8] and is a consequence of the energy scale dependent, so-called ’running’ coupling constant α_s of the strong force [4, 9]. As an example, the value of the coupling constant at the the energy scale of the Z boson ($m_Z \approx 91.2$ GeV [9]) reads $\alpha_s(m_Z) = 0.1180 \pm 0.0009$ [9]. Already at an energy scale above a few GeV [10], quarks and gluons inside the incoming hadrons interact with a sufficient large momentum transfer, Q^2, so that the coupling strength becomes sufficiently small to assume asymptotic freedom. Exemplary leading-order parton–parton interactions are visualized as Feynman diagrams[1] provided in Fig. 2.2.

Additionally, the interactions depend on the momentum fraction of the involved quarks and gluons. The probability of finding a parton with a given momentum inside a proton is described by Parton distribution functions (PDFs) (see e.g. Ref. [4, p. 562] or [11]), which play a crucial role in predicting the outcomes of these collisions. However, these distributions encode the long-distance (low energy) structure of

[1] Accordingly, the cross-sections can be calculated perturbatively in leading order with the Feynman rules – see e.g. [4] – applied on the diagrams.



quark-quark scattering (t-channel)

quark-gluon scattering (t-channel)

Figure 2.2.: Two exemplary leading-order QCD Feynman diagrams illustrating hard scattering of partons. A quark–quark (left) and quark–gluon (right) scattering via a t-channel exchange of a gluon are depicted. These examples represent key interactions of the strong force in hard scattering processes.

hadrons where the strong coupling is significant. Due to non-perturbative effects – primarily caused by the self-interaction of the force carriers (gluons) via their color charge [4] – these functions cannot be calculated but must instead be determined experimentally, e.g., through global fits to deep inelastic scattering [12] and other high-energy measurements sensitive to α_s.

After the initial high-energy interaction, secondary quarks and gluons are produced in the fragmentation stage (see e.g. Ref. [13]). As these partons move away from the interaction point and apart from each other, additional particles emerge due to a feature of the strong force, called confinement [14]. This phenomenon prevents quarks and gluons from existing in isolation, which is why only color-neutral particles are observed in nature. Essentially, as the partons separate, the strong force mediated by gluons generates a gluon field between them. In contrary to the electromagnetic force described by Quantum Electrodynamics (QCD) [15, 16], the energy in the gluon field increases with distance. When it becomes sufficiently high, it is energetically favorable to form additional quark-antiquark pairs – analog to the electron-positron pair production in electromagnetic showers. This process initiates a cascade, also called shower, of hadronically interacting particles.[2].

As the energy of the participating particles further decreases, the coupling again becomes stronger during the later stages of the jet formation process and non-perturbative effects become dominant. This is again driven by the nature of the strong force, which requires that all final-state particles be color-neutral and eventually leads to hadronization [17], the transition from the partons into observable hadrons. This makes the jet formation a highly stochastic process, as the energy and momentum initially carried by the partons are distributed among a large number of hadrons. As a consequence, measuring jets is inherently complicated due to their composite nature, preventing a particle detector from observing them directly.

The measurement of a jet instead involves several steps. At first, all individual particles emerging from the collision need to be recorded. Then, in order to draw conclusions on the initial partons, all the individual particles that come from a particular collision must be combined into one physics object – the jet. Given the complex nature of particle showers and the possibility of overlapping signals, as well as invisible components like neutrinos, consistent and robust clustering algorithms

[2] Additionally, scattering, decays, and nuclear interactions can also cause an electromagnetic component of the shower, adding further complexity to its structure.

are essential for providing comparable and reliable results. Such algorithms usually combine the signals from various sub-detectors to reconstruct the jet as a single physics object, described by the standard properties, such as pseudorapidity η, azimuthal angle ϕ, and transverse momentum p_T (see Fig. 2.4).

Additionally, external factors, such as overlapping events, detector non-uniformities, and inherent limitations in the energy resolution can contaminate the measured signals and affect the reconstruction process, leading to deviations from the true jet energy. Therefore, sophisticated calibration methods are required to mitigate the impact of these effects and enable precise measurements. This precision is crucial, because analyzing the properties of these complex physics objects allows to infer details about the initial high-energetic interactions that occurred during the hadron collisions. Such insights enable the study of the fundamental principles and mechanisms of the strong force that bind quarks and gluons together, forming the building blocks of nature. Ultimately, the accuracy of jet measurements has a direct impact on interpreting the initial interactions and on linking experimental signatures to QCD predictions.

The primary goal of JEC is to correct for the above described detector-related effects and biases, obtaining a corrected jet momentum p^corr. This is a vital process in high-energy physics experiments, typically summarizing several correction steps to ensure that the measured jet energies are as close as possible to the true energies of the originating partons and therefore accurately reflect the underlying physics. The overall concept can be simply described as:

$$p^\mathrm{corr} = C \times p^\mathrm{raw} \, , \tag{2.1}$$

where C describes the correction factor that is applied to each component of the raw jet four-momentum vector p^raw [18, p. 8]. It usually includes offset corrections to address contributions from pileup and noise, simulation-based, relative and absolute corrections to adjust for non-uniform detector response and non-linearity, and data-driven, residual corrections to fine-tune the calibration using well-understood reference processes. One example for this last step are Z+Jet events where the boson is produced in a back-to-back topology with a jet, on which the focus of this work lies. With such events, the measured jet energy can be directly related to the more precisely measurable momenta of the decay products of the Z boson (e^+e^-, $\mu^+\mu^-$), allowing to correct the Jet Energy Scale (JES) for remaining differences in the sub-percent range to ensure highest precision.

Additionally, for a reliable calibration of jet energies, the so-called Jet Energy Resolution (JER)[3] needs to be considered as well, since it also affects the jet energy measurement of the detector. The JER corrections are typically employed in combination with the Jet Energy Corrections, as it adds to the overall precision by fully accounting for the detector's energy resolution. At the CMS experiment, for example, JER-based corrections are derived by comparing control samples, such as dijet and γ+ jet events [19, 18] with data. For this purpose, the jet response distributions – typically modeled by a Gaussian – in both data and simulation are measured, and the resolution is extracted as the standard deviation of the distributions. By computing the ratio between simulation and data, a scale factor can then be derived. To apply JER-based corrections to simulated jets, often also referred to as smearing,

[3] For the CMS experiment, see e.g. [19, 18, 3].

their energy distribution is convolved with a resolution function. Its width is scaled by the derived factors, to adjust the resolution so that it matches that seen in the actual detector data. With this follows for the corrected p_{T}:

$$p_{\mathrm{T,smeared}} = p_{\mathrm{T}} \times \mathrm{Gaussian}(\mu = 1, \sigma = \sqrt{k^2 - 1}\sigma_{\mathrm{MC}}) , \tag{2.2}$$

where k describes the Monte Carlo (MC) simulation-derived scale factor for the JER corrections between data and simulation (as described above) and σ_{MC} is the resolution in MC simulation (see Ref. [3, p. 29]). This process broadens the simulated energy distribution so that it more accurately reflects the resolution observed in recorded data. A better understanding of the resolution ensures that the corrected measurements yield values closer to the true energy, which is also essential for accurately assessing the measurement uncertainty.

To summarize, JEC and JER together provide a comprehensive framework for a more precise jet energy measurement. While the JEC corrects for systematic biases and ensures that the energy scale is correct, JER corrections provide additional insight into the precision limits of the energy measurement and further enhance the overall accuracy of the results.

2.2. The CMS Experiment at the LHC

The CMS experiment [20, 21] is one of the general-purpose experiments at CERN. Its detector is designed to investigate proton-proton and heavy ion collisions at the LHC. Together with the ATLAS experiment, it was able to find evidence for the Higgs boson – the last missing piece of the SM puzzle. Until today, it plays an important role in exploring fundamental physics, from investigating and measuring the properties of the Higgs boson to searching for new particles and interactions beyond the SM.

The LHC, depicted with the four main experiments in Fig. 2.3, is not only world's most powerful particle accelerator in terms of energy, but it is also the basis for a broad international collaboration with several thousands of physicists, engineers, IT specialists, and others. With a tunnel of about 27 km in the area of Geneva, the accelerator is the largest and one of the most complex (scientific) machines ever built. The protons that are brought to collision in the particle detectors distributed around the LHC have been pre-accelerated through a series of smaller accelerators before reaching their design collision energies of $\sqrt{s} = 14$ TeV (Run 3).

Overall, the LHC[4] is a technical masterpiece that combines numerous technologies to achieve the collision rates that enable high-precision HEP experiments. This is because the greatest possible luminosity is required to collect sufficient statistics for the direct evidence of new particles and to find small deviations from the theoretical models established today.

At the same time, the extreme number of collisions – around 40 million per second – poses an enormous challenge for the detectors. On the one hand, extreme data rates are achieved that need to be processed and stored, which makes fast and reliable trigger mechanisms and read-out mechanics indispensable. On the other, unavoidable side effects occur, because to increase the probability of interactions, not only

[4] More detailed information on the LHC in general and Run 2 in particular are provided in Refs.[23, 24]. The future design of the HL-LHC is, e.g., described in Ref. [25].

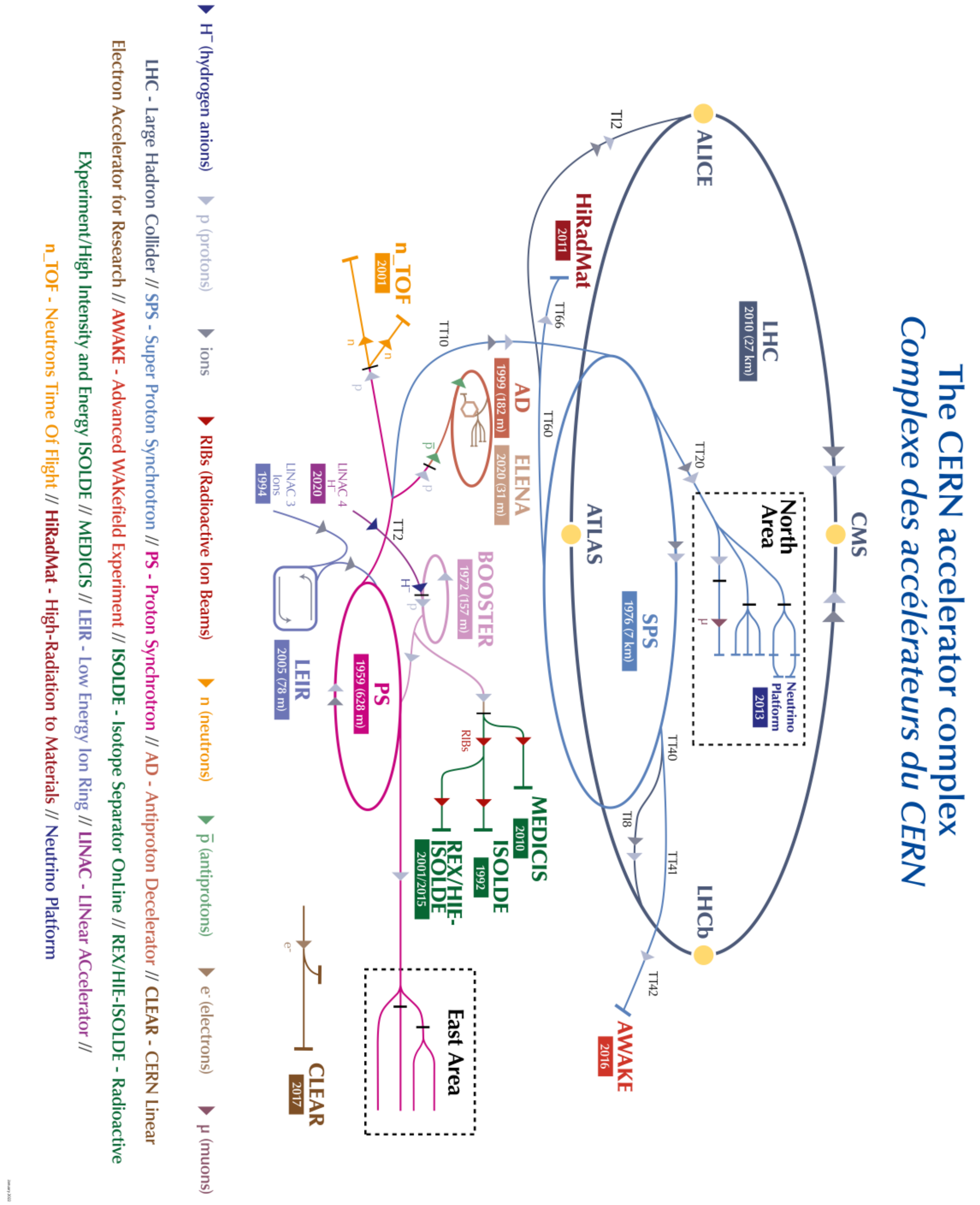

The CERN accelerator complex
Complexe des accélérateurs du CERN
ALICE
CMS
LHCb
ATLAS
LHC
2010 (27 km)
North Area
Neutrino Platform
2013
SPS
1976 (7 km)
TI2
TI8
TT20
TT40
TT41
TT42
TT60
TT66
TT10
TT2
AWAKE
2016
HiRadMat
2011
AD
1999 (182 m)
ELENA
2020 (31 m)
BOOSTER
1972 (157 m)
ISOLDE
1992
MEDICIS
2010
REX/HIE-ISOLDE
2001/2015
RIBs
East Area
CLEAR
2017
PS
1959 (628 m)
LEIR
2005 (78 m)
LINAC 4
H⁻
2020
LINAC 3
Ions
1994
n_TOF
2001
n
p
p̄
μ
e⁻
H⁻ (hydrogen anions)
p (protons)
ions
RIBs (Radioactive Ion Beams)
n (neutrons)
p̄ (antiprotons)
e⁻ (electrons)
μ (muons)
LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear
Electron Accelerator for Research // AWAKE - Advanced WAKefield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE-ISOLDE - Radioactive
EXperiment/High Intensity and Energy ISOLDE // MEDICIS // LEIR - Low Energy Ion Ring // LINAC - LINear ACcelerator //
n_TOF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials // Neutrino Platform

Figure 2.3.: This figure provides an overview of the LHC accelerator complex and main experiments as of 2022. As indicated, accelerators of previous generations are still used as pre-accelerators for the LHC. The ATLAS experiment is located around Meyrin, Switzerland, while the other experiments are located in France. *Source*: CERN/[22]

individual particles are accelerated, but so-called bunches of many billions of protons instead. In each bunch crossing, potentially multiple particles are interacting, resulting in a high instantaneous luminosity[5]. This not only increases the rate of rare events but also introduces significant challenges due to the presence of numerous overlapping proton-proton interactions, commonly referred to as pileup[6]. As a result, energy depositions in the detector from multiple collisions are mixing up, making it more difficult to isolate the true energy deposits associated with the primary event.

To tackle these challenges, the CMS detector employs advanced technologies and sophisticated data processing and reconstruction algorithms. It is designed as a multi-purpose, hermetic (almost 4π) detector with different, specialized layers focusing on various particle types and properties each. The central part of the detector, often referred to as barrel, is ranging in $|\eta| < 1.5$ in pseudorapidity. It is supplemented by the so-called endcaps, which are closing the detector nearly hermetically to ensure that as much information as possible is contained. In Fig. 2.4, the cross-section of the detector is shown. The individual parts are briefly described in the following.

2.2.1. Superconducting Solenoid Magnet

Essential for the CMS detector concept is its powerful solenoid magnet, responsible for the naming of the detector and a homogeneous magnetic field of about 4 T [21] inside it. The strong magnetic field allows a precise momentum measurement of charged particles by enabling a global, consistent tracking system. In contrast to the ATLAS detector, the magnet is located outside the calorimeters. This has the advantage that charged particle trajectories are bent consistently throughout all inner detector layers, further improving the momentum resolution. Additionally, it allows a more compact design which reduces gaps between the barrel and end caps leading to a better hermeticity.

When charged particle now traverse the magnetic field, they experience a Lorentz force that changes their trajectories. Since the curvature directly correlates with their momentum, a stronger field leads to a better resolution and additionally allows higher energies to be measured. Moreover, the magnetic field is also employed for the particle identification, as the curvature and corresponding energy measurements in the tracking and calorimetric systems provide critical information to distinguish between different types of charged particles.

2.2.2. Tracker System

As the innermost part, the silicon tracker system [31] is the heart of the CMS detector. It is directly surrounding the beam pipe, visible as the dark gray ring in the center of Fig. 2.4. The inner part of it is the pixel detector, which ensures a precise determination of the interaction points with a spatial resolution below 15 µm. It is surrounded by the silicon strip detector which consists of larger strips instead of pixels, but still provides a decent resolution to track the path of particles over a larger area. Thanks to the strong magnetic field, they allow a precise reconstruction of the trajectories of

[5] The delivered integrated luminosity in Run 2 is visualized in Fig. B.1.

[6] For further details, see e.g. Ref. [26].

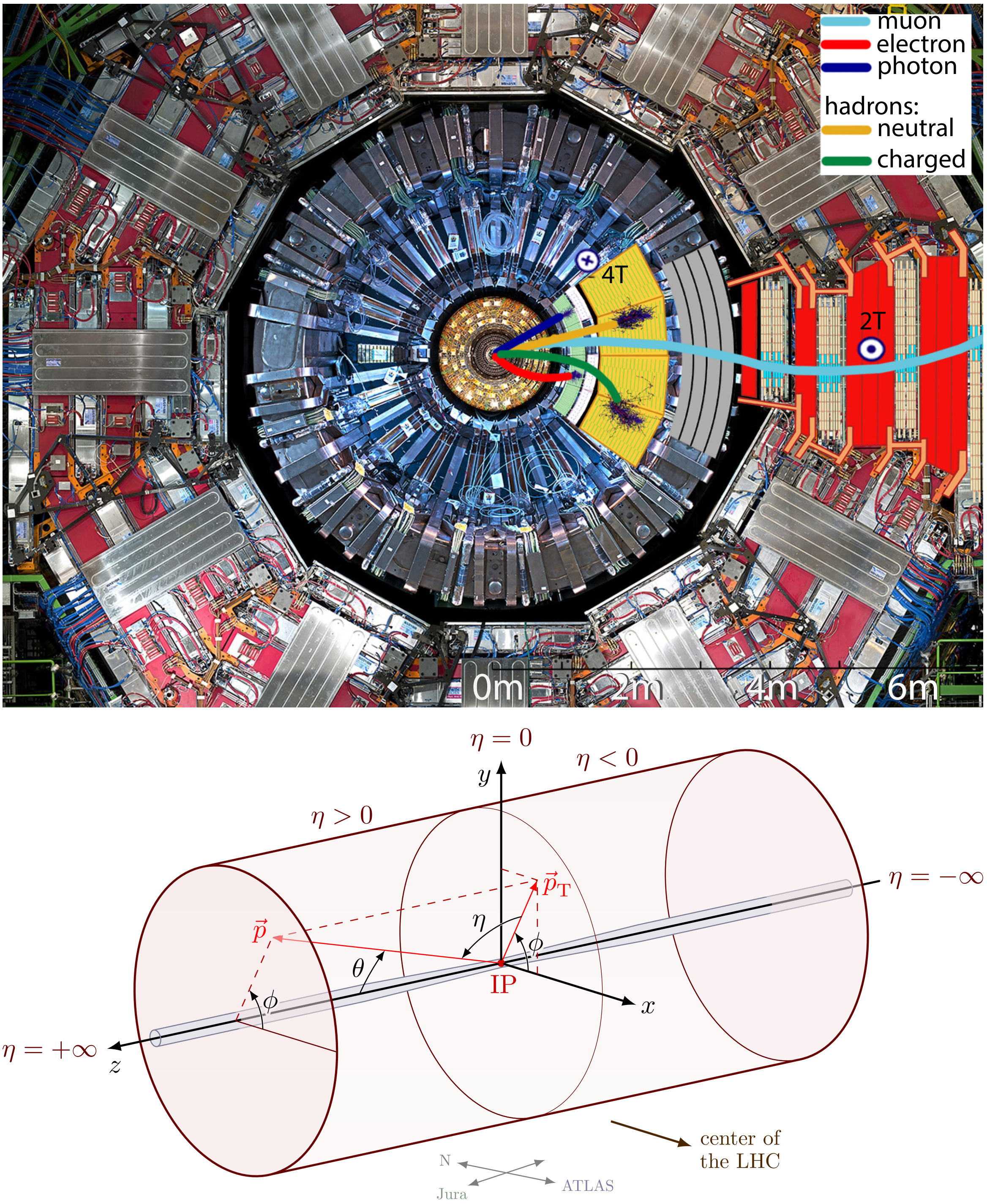


Figure 2.4.: Cross-section of the CMS detector (top) as a mesh-up of a real picture [27] and a schematic visualization of the different layers [28]. Particles originating from a collision are first traversing the silicon tracker system (inner gray) until they reach the electromagnetic calorimeter (picture: gold/scheme: light green). Photons and electrons are typically fully contained in this part of the detector. Charged and uncharged hadrons are ideally stopped in the hadron calorimeter (picture: blue, scheme: yellow), which is surrounded by the superconducting magnet (gray). Muons are often traversing the entire detector and can be identified with the outer muon system including the iron return yoke (red). *Note: The schematics are not perfectly matching the actual detector layers on the endcap for a better overview.*
Below, the coordinate system of the detector is depicted [29]. For more information, see e.g. Section 1.2 in Ref. [30].

charged particles back to their origin in order to keep them apart and identify them. This high-precision tracking system is therefore the basis for distinguishing between multiple collisions and is optimized for a full event reconstruction at highest collision rates.

It was upgraded in the beginning of 2017 with the goal to increase the robustness and tracking efficiency to cope with the higher instantaneous luminosity provided by the LHC during Run 2. The upgrade included an extension of another barrel layer (four in total) and endcap disk (three) while reducing the material budget through optimizations in the cooling system [32]. It is covering $|\eta| = 2.5$ in pseudorapidity. Further technical details are provided in e.g. [33, 32].

2.2.3. Calorimeters

The Electromagnetic Calorimeter (ECAL) [34] and Hadron Calorimeter (HCAL) [35], depicted in Fig. 2.4 as light green and yellow, respectively, are sub-detectors with the purpose of measuring the energy of particles. Both follow the same principle: traversing particles are fully absorbed and their energy is measured by converting it into detectable light. While the ECAL is specifically designed primarily for the measurement of electrons and photons, the HCAL has a greater stopping power enabling it to reliably measure the energy of hadrons, such as protons, neutrons, or pions. The difference between the two variants is mainly the material and the resulting structural design.

For the ECAL, lead tungstate crystals ($PbWO_4$) were chosen [34, 21] due to their high density and scintillation properties. When a charged particle enters the ECAL, it initiates an electromagnetic cascade of secondary particles, commonly called shower. This, in turn, deposits energy in the form of scintillation light within the crystals, which can be detected by photo multipliers at the beginning and end of the crystals. This process is repeated until all energy is deposited, allowing the calorimeter to accurately quantify the deposited energy of electrons and photons. The measured light intensity is directly proportional to the energy of the traversing particles. For hadrons, however, the material is not sufficient to stop the particles, which necessitates the additional HCAL.

The HCAL is a sampling calorimeter. Unlike the homogeneous design of the ECAL, it consists of alternating layers of dense absorber material (brass) and scintillators. When hadrons reach the first layer of the absorber material, they undergo nuclear (strong) interactions initiating hadronic showers. These showers are way more complex than purely electromagnetic showers, as they involve both primary nuclear interactions and secondary electromagnetic sub-showers, which together distribute the energy of the hadrons over a broader volume. The active layers of the HCAL measure the deposited energy. However, due to the complexity of hadronic interactions and the sampling nature of the calorimeter, the overall energy resolution is significantly worse than for the ECAL.

On top, the HCAL of the CMS detector has a comparably smaller nuclear interaction length, which is the price that has to be payed for the compact design. As a consequence, it is not always guaranteed that the entire hadronic shower is contained within the calorimeter. But at the same time, this also provides significant advantages. The segmentation of the HCAL provides information on the spatial distribution of the energy deposition over the detector area. And since both calorimeters are located within the magnetic field, they also provide complementary spatial information for reconstructing the tracks of diverse physics objects.

2.2.4. Muon System

The muon system is a crucial part of the entire detector concept, thus contributing to the name of the detector. Since muons are Minimum Ionizing Particles (MIPs, see e.g. Ref. [36]) which deposit little energy in matter, they are hard to measure accurately in a compact detector system like the CMS detector. To ensure a reliable detection and precise measurement of these particles, which are essential for a wide range of studies, the muon system consequently employs a different approach than the calorimeters. Instead of stopping the muons, the system relies on specialized tracking technologies (namely drift tubes, cathode strip chambers, and resistive plate chambers [30, 21]) that are optimized for fast timing and high spatial resolution. To achieve the best accuracy, the muon system is placed outside the solenoid magnet. In Fig. 2.4, it is visualized as the alternating red and silver/beige layers centered around the magnet. This positioning maximizes the coverage area and leverages the iron return yoke of the magnet, which not only provides a magnetic field but also serves as a natural filter for other particles than muons. Together with the information of the inner tracker and the calorimeters, these technologies accurately record the passage of muons starting from the interaction point, enabling precise momentum measurements and providing trigger signals employed for event selection.

Although the muon system is primarily dedicated to the triggering on global muons and to their precise detection and measurement, it also plays a supportive role in the overall event reconstruction. In particular, the identification of muons is important to distinguish between overlapping physics objects and can contribute to the calibration of the detector, as described later.

2.2.5. Trigger System and Data Acquisition

The CMS detector employs a comprehensive, multi-level triggering system [37]. Its main purpose is the fast filtering and selection of collision events of interest. The enormous collision rates are too high to be entirely processed by the data acquisition system – let alone the gigantic amount of data produced that would have to be stored. Furthermore, A full readout of the detector takes time, which makes a fast and reliable pre-selection of events indispensable. This first selection (Level-1 trigger) is realized with a hardware-based system that decides within 4 µs if an event should be rejected based on information of the calorimeters and muon detectors [37].

After the first trigger step, the event rate is reduced to roughly 100 kHz by selecting events based on criteria such as energy deposits and simple topological signatures. Since this rate is still too high for a full readout, the selected events are further evaluated by the software-based High-Level Trigger (HLT), which essentially is a server farm located directly next to the detector to achieve low latencies. The HLT applies simplified reconstruction and event analysis algorithms[7] to filter based on the informative value of the events for further physics analyses. Ultimately, the

[7] Accordingly, a high performance of the computer system is required. Nowadays, for Run 3, the HLT even employs GPUs for the fast, online reconstruction. For more details, see e.g. Ref. [38].

output rate is reduced to around 1 kHz[8] [30], which is acceptable for the data acquisition systems. For further technical details, one may consult [37, 30].
From a computing and data analysis perspective, the triggers have another crucial role. They apply a separation and selection of data that contain the same characteristics, like e.g. single or double muon events. With this, a lot of compute power is saved by simplifying the selection of adequate datasets for particular purposes[9]. However, this also means that a high trigger efficiency is absolutely essential. The efficiency and selection criteria of these triggers directly affect the data samples used, e.g. for JEC. Here, only with well-understood selection criteria it is ensured that the subset of events used is both representative and free from significant systematic biases, which is essential for the accurate calibration of jet energies recorded with the CMS detector.
Ultimately, after being recorded and archived, the data is processed and provided in different data tiers to enable a more demand-oriented use. The *RAW* data includes the full detector readout and all data collections, causing a size of 1-2 MB per event [30, p. 69] – which is huge when considering multi-billion event datasets. This data type is mainly archived as a full backup of the highly-valuable measurement data. After the first filtering and processing steps, only selected, reconstructed objects are kept, reducing the size to 500 kB per event in the *RECO* data type [30, p. 69]. This amount of data is still too large to be easily transferred and processed by end users all around the world. Therefore, a further condensation to the Analysis Object Data (*AOD*) format with the target of 50 kB per event, containing only high-level physics objects and flags, was conceived from the beginning, as stated in the technical design report [30, p. 70].
However, the average event size for *RECO* and *AOD* came out much larger – in the order of 1 MB. And since most of the analyses do not require all the data of the still very extensive *AOD* datasets, a further reduction to *MiniAOD* (see Ref. [39]) was decided for Run 2, reaching around 200-300 kB per event. During Run 2, the default analysis data format was further streamlined leading to *NanoAOD*, which has a reduced event size in the order of 1 kB [40] – this means a factor of 50 to 100 times less data to transfer, store, and process. This also leads to a faster production and reprocessing of the datasets, however, at the cost of a reduced content due to the neglect of further detailed information.

2.2.6. Particle Flow Approach and Reconstruction

The high-resolution tracking system, together with the spatially segmented calorimeters within the strong magnetic field provide the perfect conditions for the effective tracking and reconstruction of individual particles. At CMS, the offline reconstruction of a collision event is therefore conducted by combining the information of all sub-detectors to identify and reconstruct each individual particle candidate of an event. This concept is described as the Particle Flow (PF) algorithm [41, 26], as it allows to track the path of the individual particles from the interaction point

[8] For Run 2 and Run 3, this was further increased up to around 2 kHz and 5 kHz, respectively.

[9] As an example: For the 2018 Ultra Legacy data, no double electron (DoubleEG) dataset exists. Therefore, the entire EGamma dataset for the full period has to be processed and matching events that fulfill the required criteria have to be selected manually, which requires considerably more compute time.

throughout the entire CMS detector.

Its bottom-up approach begins with precisely reconstructing the vertices and charged particle tracks in the inner tracker, which are then associated with the corresponding energy deposits, commonly referred to as clusters, in both the electromagnetic and hadronic calorimeters. Finally, the information about the particle candidate is supplemented by the muon system and merged. Already at this stage, charged PF candidates can be identified as pileup based on the position relative to the primary vertex of the event. In the next step, recorded calorimeter clusters without a charged track are then identified as neutral particles, for example photons in the ECAL or neutrons in the HCAL. Afterwards, detector calibrations and alignment information (see e.g. Ref. [30, p. 70]) are applied to correct the individual particle candidates for detector response non-linearities, energy loss, and other systematic effects. This ensures that the measured energy and momentum of the candidates is in agreement with the true particle properties.

Ultimately, the exact process, of course, depends on the type of particle. Reconstructing muons, for example, mainly relies on the inner tracker and the muon system[10]. With dedicated muon reconstruction algorithms, so-called global (outside-in) or tracker (inside-out) muons are identified and reconstructed [42]. For this, an attempt is first made to match the tracks in the tracking system with the corresponding hits in the muon system. If a track from the inner tracker can be matched successfully with muon system hits, a global muon fit is performed, resulting in the PF candidate for the global object. Thanks to the additional information from the outer system, this improves the momentum resolution and ensures that the candidate is indeed a muon. Of course, the selection of muons therefore depends on the algorithms and conditions. Based on different muon identification (ID) variables, it can therefore be chosen between different levels of identification efficiency and purity [42]. The most important selections to mention are the Muon ID and Muon Isolation, which will be selected as *Tight* for the JEC described later. This is the typical selection for most analysis purposes. While the Muon ID determines the requirements on the identification as a muon, e.g., to be a global muon with a certain goodness of fit for the track reconstruction and others, the Muon Isolation describes the required isolation from other reconstructed objects. This prevents an identification as a stand-alone muon if it is part of, e.g., a B-jet. For further details on the different levels it is refered to [42, p. 6] and [43], which also contains more detailed information on the muon reconstruction process as a whole.

Once confirmed, the above described muon is integrated into the set of PF candidates within the reconstruction framework. Its momentum and trajectory information are then used to complement the full event reconstruction, ensuring that the overall interpretation remains consistent and that momentum conservation laws are satisfied.

In summary, with the PF approach, the complex collision environment can be disentangled efficiently. By combining the available information from all detector sub-systems, it provides a comprehensive and detailed event reconstruction, significantly enhancing the overall physics performance of CMS by effectively resolving overlapping signals.

As a next step, the accurate reconstruction of composite objects formed from the

[10] On top of that, information from the calorimeters can be used to verify the identification as muons by requiring a comparably low energy deposition along the track, which indicates minimum ionizing particles like muons.

individual PF candidates is described.

2.2.6.1. Jet Reconstruction

Following the reconstruction of all individual PF candidates, higher-level objects are built from them. In particular, jet reconstruction at CMS involves a clustering of these candidates using the anti-k_T algorithm [44], which groups nearby particles into jets. This sequential clustering is based directly on the individual reconstruction of all PF candidates. The first step is the calculation of the distance between each pair of particles (i, j):

$$d_{ij} = \min\left(p_{\mathrm{T}i}^{-2}, p_{\mathrm{T}j}^{-2}\right) \frac{\Delta R_{ij}^2}{R^2}\,, \tag{2.3}$$

where $p_{\mathrm{T}i,j}$ are the transverse momenta of the particles and ΔR_{ij} is the distance defined in the pseudorapidity-azimuth plane:

$$\Delta R_{ij} = \sqrt{(\Delta\eta_{ij})^2 + (\Delta\phi_{ij})^2}\,. \tag{2.4}$$

The jet radius parameter, R, defines the maximum distance within which particles can be clustered together into a single jet. It therefore reflects the size of the jet in the $\eta - \phi$ plane. The used value of R depends on the analysis and experimental conditions. Typical values are $R = 0.4$ and $R = 0.8$, representing common jet collections with a naming scheme linked to the algorithm and the jet radius: AK4 and AK8.

Additionally, the so-called distance to the beam [44, p. 2] is derived for every particle, reading:

$$d_{iB} = p_{\mathrm{T}i}^{-2}\,. \tag{2.5}$$

Here, the minus sign in the exponent defines the behavior of the sequential clustering and indicates a starting from the leading-energy PF candidate[11]. Note that 'beam' in this case does not refer to the proton beam, but is only an abstract reference representing the initial state defined by the incoming proton beams. Figuratively speaking, d_{iB} is used to decide, when a particle is sufficiently isolated, representing a simple criterion to decide when a particle should be declared as a final jet rather than being merged with another particle.

The clustering now works as follows: At first, the minimum of all d_{ij} and d_{iB} is identified. If the minimum is a pair-wise distance d_{ij}, the two PF candidates are merged into a new object by vectorially combining their properties, resulting in the momentum-weighted average of the two. If the smallest distance is a beam distance d_{iB}, the particle i is declared to be a final jet and removed from the list of candidates. This process is iteratively repeated until all particles are clustered. Additionally, while in principle all PF candidates can be included in the clustering, for analyses like the JEC, which focuses on hadronic jets, isolated leptons (such as muons that are identified as isolated) are often removed from the list of candidates prior to a manual (re)clustering process to avoid double-counting or an alteration of jet energies.

The anti-k_t approach is chosen for CMS because its proper p_T-weighting leads to more regular jets by weighting lower transverse momentum less, causing hard particles

[11] If the momenta are neglected ($d_{iB} = 0$), only the spatial proximity in the $\eta - \phi$ plane is considered. This clustering approach is referred to as Cambridge/Aachen, see Ref. [44, p. 2] or Ref. [45]. A comparison of different algorithms is provided in Fig. B.2.

to attract softer ones and ultimately form nearly conical jet shapes. In this configuration, the high p_{T} (hard) particles act as seeds for the jets, absorbing nearby low-p_{T} (soft) particles, which simplifies the jet structure and results in well-defined, regular jets boundaries. Furthermore, the anti-k_{t} algorithm is inherently infrared and collinear safe [44], because its distance measures ensure that adding arbitrarily soft particles, e.g. originating from infrared emissions, or splitting a particle into collinear fragments does not significantly change the final result of the jet clustering. It therefore ensures stable and robust jet definitions that are easier to calibrate and compare with theoretical predictions, ultimately improving the precision and reliability of the measurements.

In summary, the PF concept at CMS enables a reliable and accurate jet reconstruction and extends the possibilities for further precision improvements. By accurately considering each individual particle candidate within a jet, the PF algorithm, in combination with the anti-k_{t} clustering, ensures that the energy of each jet is measured with high accuracy. The robust jet reconstruction process is furthermore essential for deriving precise Jet Energy Corrections, which in turn lead to a better signal–background discrimination and reduced systematic uncertainties in physics analyses conducted with CMS data.

2.2.6.2. Missing Transverse Momentum and Pileup

Two additional factors that can significantly impact the overall event interpretation are the so-called Missing Transverse Momentum (MET)[12] (see e.g. Refs. [46, 47, 48, 21]) of an event and pileup [21, 48]. MET is conceptually based on the well-justified assumption that the transverse momenta of the incoming protons can be neglected. Consequently, in a perfect particle detector, this quantity would ideally be zero due to momentum conservation. However, undetectable particles, such as neutrinos or potential DM candidates, can lead to an imbalance in momentum in the plane perpendicular to the beam, thereby signaling their presence. MET is therefore defined as the negative vector sum of the transverse momenta of all PF candidates:

$$\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}} = -\sum_{i} \vec{p}_{\mathrm{T},i} \,. \tag{2.6}$$

Thanks to the accurate reconstruction of all individual particles, the PF approach employed at CMS makes a highly precise measurement of MET possible. For more details, it is referred to [46, 47, 21].

Furthermore, the overall momentum balance of an event is also influenced by remnants of additional proton-proton interactions that occur in the same or nearby bunch crossings as the primary event. These extra interactions produce additional particles that can be mistakenly reconstructed as part of a jet, or even form entirely separate *pileup* jet. The resulting contamination can distort the momentum balance, leading to poorer jet energy measurements, degraded resolution, and increased systematic uncertainties. To mitigate the impact, CMS employs a number of different strategies, two of which are: Charged Hadron Subtraction (CHS) [3] and PileUp Per Particle Identification (PUPPI) [49].

[12] Remark: In the past, $\vec{E}_{\mathrm{T}}^{\mathrm{miss}}$ was used to describe Eq. (2.6). The acronym therefore originates from the short form of 'missing E_{T}' – MET.

CHS utilizes the precise track reconstruction of charged PF candidates to identify their production vertices. Consequently, by subtracting charged hadrons that originate from vertices other than the primary event – so-called pileup vertices – the method effectively reduces the contribution of unrelated particles to the jet energy. This selective removal of pileup-associated tracks helps to improve the purity of the reconstructed jet and enhances the overall jet energy resolution. Further details on the employed pileup mitigations are available in [3, p. 12].

PUPPI employs a more comprehensive, per-particle approach that is able to effectively mitigate pileup contributions from both charged and neutral PF candidates. To achieve this, the algorithm evaluates the spatial and momentum distribution of particles in the vicinity of each candidate to determine whether its surrounding activity is more consistent with the primary interaction or with pileup [49]. PF candidates identified as likely originating from pileup are then weighted down in the calculation of the jet energy and momentum. This ultimately suppresses the influence of pileup on the final reconstructed jets, which are less contaminated by additional, unassociated energy deposits. In general, PUPPI has proven particularly effective in high-pileup environments, hence, it is predominantly used for most analyses today. However, it is not without danger to employ for more low-level actions, like JEC, as it applies per-particle weights to mitigate pileup, which inherently modifies the jet energy response in a non-linear fashion. For further details on the pileup jet identification process at CMS, it is referred to Ref. [48].

In summary, the MET definition in the general, *raw*, uncorrected form (Eq. (2.6)) is only an idealization that suggests it exclusively provides information on undetectable particles based on the imbalance (or missing energy) of an event. In reality, however, it also implicitly includes contributions from pileup and suffers from degradation of energy measurements due to reconstruction inefficiencies and detector noise. As a consequence, the raw MET does not provide a completely accurate representation of the investigated event. To reliably draw conclusions on the invisible parts of a collision event and to provide an overall more realistic and precise description, dedicated MET corrections and filtering are essential.

For the CMS experiment, different correction methods have been developed, addressing different quality aspects of the measurement. These are commonly referred to as *Type-I* and *Type-II* corrections (supplemented by an optional *Type-0* correction – depending on the analysis). A detailed description of these methods can be found in [46]. The corrections are applied after the pileup mitigation strategies discussed above and incorporate further techniques, such as Jet Energy Corrections, to refine the raw measurement. This ensures that the final more accurately reflects the true energy imbalance caused by undetected particles while reducing the impact of the other mentioned effects.

In practice, this process is further simplified for end-user analyses because the official datasets centrally provided by the collaboration already include so-called MET Filters[13] in the form of preprocessed filter flags, which address various quality assurance aspects and can be applied directly. Using these filters is widely recommended, see Ref. [50]. The filter flags relevant for this thesis are listed in Section A.1.2.

For more details on MET measurements and corrections for the CMS experiment, see Refs. [46, 47, 50].

[13] Note: Nowadays, these filters are typically referred to as 'Noise Filters', see Ref. [50].

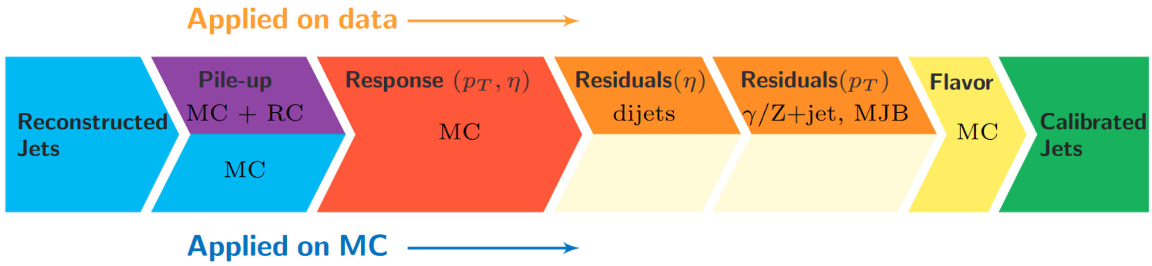


Figure 2.5.: Overview of the full CMS Jet Energy Correction chain. The individual steps are applied on MC simulations or data (as indicated by the horizontal separation). All steps are executed sequentially and use the output of the previous step (in form of corrected jets) as input to produce fully calibrated jets. The flavor-dependent calibration step (yellow) is optional and applied only when the analysis requires additional corrections for flavor-specific detector responses. More details can be found in [3]. *Source:* [51]

2.3. Overview of the Jet Energy Correction Steps

The precise calibration of jet energies is critical for a wide range of analyses in high-energy physics, particularly those that probe the strong interaction through differential measurements. One prominent example is the triple-differential Z+Jet cross-section measurement [52], which is highly sensitive to the details of QCD. In such measurements, even small biases in the JES can lead to significant distortions in the observed production cross-section as a function of jet transverse momentum. Such distortions can mask underlying physics effects and bias the measurement of QCD parameters. Within the above mentioned analysis, for example, the JEC uncertainty is the dominating systematic (experimental) uncertainty at low energy scales (for the Z Boson: 25 GeV $< p_{\mathrm{T}}^{\mathrm{Z}} <$ 50 GeV, see e.g. Ref. [52, p. 42] or Ref. [53]). As a consequence, a comprehensive set of Jet Energy Corrections as part of the whole JEC has been developed for the CMS experiment to ensure that jets are accurately measured and systematic distortions that could obscure underlying physics are avoided.

Jet Energy Corrections are applied as a series of steps addressing different aspects and challenges for the reliable energy measurement of jets. The individual steps are typically referred to as Levels, which are depicted in Fig. 2.5. They are designed to systematically remove various detector effects and biases and implement diverse MC-based and data-driven methods. Level-1 (L1, purple) describes offset corrections that remove extra energy from pileup, noise, and underlying event. The jet response correction relative in η (L2) and in absolute p_{T} (L3) are typically applied together and therefore referred to L2L3 (red). While the relative corrections aim to ensure a uniform response across different detector regions, the absolute corrections adjust the absolute energy scale (as a function of p_{T}). After this, the residual corrections follow (orange). These are data-driven (in-situ) methods based on dijet events (L2Res) and γ/Z+Jet or multi-jet events (L3Res/L2L3Res), which correct for remaining discrepancies between data and simulation. This step represents a fine-tuning of the calibration beyond what is achieved from simulation. Lastly, the flavor correction (yellow), is applied optionally, only if a flavor-dependent response is expected for a particular analysis.

From this, the total correction factor, C, from Eq. (2.1) denotes as [18, p. 8]:[14]

$$C = C_{\mathrm{L1(offset)}}(p_{\mathrm{T}}^{\mathrm{raw}}) \times C_{\mathrm{L2L3(MC)}}(p'_{\mathrm{T}}, \eta) \times C_{\mathrm{L2Res(rel)}}(\eta) \times C_{\mathrm{L2L3Res(abs)}}(p''_{\mathrm{T}}) \ , \tag{2.7}$$

where p'_{T} indicates the offset corrected value, and p''_{T} accordingly the value after the response corrections.

While the first correction steps have the highest impact on the overall result and can reach up to two-digit percent levels[15], the residual corrections are particularly crucial for high-precision physics, as they help to resolve remaining discrepancies between calibrated jet energies in data and simulation to the percent or even sub-percent level. This fine-tuning is essential for, e.g., high-precision differential cross-section measurements, where minimizing differences between data and MC models is required to accurately extract the underlying physics.

The full correction chain is described in detail in [3] and briefly in the following paragraphs. For further information on the uncertainties of the described methods it is referred to [3, 18], particularly [3, p. 31].

2.3.1. L1: Pileup Offset Corrections

The first level mitigates the effects of pileup and detector noise [3, p. 9]. In high-luminosity environments, not only additional proton-proton interactions introduce extra energy deposits unrelated to the primary scattering process, but also beam remnants and other side-effects can influence the energy measurement. L1 corrections therefore go beyond the steps described in Section 2.2.6.2 by subtracting these contributions from the jets. The default method for this is called (hybrid) jet area method (see Refs. [3, p. 13], [56], [18, pp. 8–9]). It assigns each jet an effective area A_j. This area quantifies the extent of the detector over which the jet collects energy. The basic idea is that larger jets, simply by their size, are more likely to pick up extra energy from pileup. Then, the pileup energy density, ρ, is estimated for the event from the median transverse momentum per unit area:

$$\rho = \mathrm{median}(p_{\mathrm{T}j}/A_j) \ . \tag{2.8}$$

This value is typically derived from regions of the detector which are less affected by the hard scattering process – i.e. far away from high-p_{T} activity in the detector. The expected pileup contribution is then calculated as:

$$p_{\mathrm{T}j,\mathrm{offset}} = A_j \times \rho \ , \tag{2.9}$$

and is subtracted from the raw jet transverse momentum:

$$p_{\mathrm{T}j}^{\mathrm{sub}} = p_{\mathrm{T}j} - A_j \times \rho \tag{2.10}$$

This correction ultimately provides a cleaner baseline jet energy measurements and reduces biases from additional, unassociated energy sources.

As an additional approach for minimizing residual differences between data and MC, the above method is adapted by randomly placing cones of a fixed size, A_{cone},

[14] For information on how to actually apply these corrections, one can consult e.g. [54, 55, 51].

[15] Of course, this is dependent on many factors and should therefore only be seen as a rough indication.

throughout the detector for estimating the pileup energy density of a given event. For every cone, the within contained transverse momentum $p_{\mathrm{T}}^{(i)}$ is measured and the density is derived analog to Eq. (2.8) but with the fixed area instead:

$$\rho = \mathrm{median}(p_{\mathrm{T}j}/A_{\mathrm{cone}})\ . \tag{2.11}$$

A key advantage of the method is that its random placement of the cones reduces the bias from hard jets by more effectively avoiding regions with high-p_{T} activity.
In practice, the L1 pileup offset corrections are derived using QCD dijet simulations processed both with and without pileup overlay [3, p. 71]. Additional adjustments accounting for residual differences between actual data and detector simulations are extracted via the random cone method applied to zero-bias events [3, p. 71].

2.3.2. L2L3: Simulated Response Corrections

After accounting for pileup effects, the relative and absolute response corrections [3, p. 21] are both derived from MC simulations. For this, simulated jets are reconstructed as usual and compared to their corresponding generator-level (MC truth) jets for every η region and p_{T} bin.
The relative corrections (L2) address variations in the detector response in dependence of the pseudorapidity, η, and are derived as:

$$C_{\mathrm{L2}}(\eta) = \frac{1}{\langle R(\eta)\rangle}\ , \tag{2.12}$$

where $\langle R(\eta)\rangle$ describes the average response computed for all jets in the η bin as $R(\eta) = p_{\mathrm{T}}^{\mathrm{reco}}/p_{\mathrm{T}}^{\mathrm{gen}}$. This correction is necessary, because the detector response can be influenced by, e.g., different material budgets in different regions, blind overlap regions and other effects that can result in significantly different energy measurements across η.
The absolute response corrections (L3) adjusts the overall jet energy scale so that the reconstructed jet energy of a simulated event accurately reflects the true particle-level energy. This step is critical for transforming raw jet measurements into a scale that closely represents the physical jet energies. It is again derived on simulation and achieved by matching the transverse momentum of the reconstructed jet to that of its corresponding generator-level jet and averaging over many events. The resulting correction factor calibrates the jet energies such that, on average, the corrected jet p_{T} matches the generator-level p_{T} and thus reflects the true energy scale:

$$C_{\mathrm{L3}}(p_{\mathrm{T}}) = \frac{1}{\langle R(p_{\mathrm{T}})\rangle} = \frac{\langle p_{\mathrm{T}}^{\mathrm{gen}}\rangle}{\langle p_{\mathrm{T}}^{\mathrm{reco}}\rangle}\ . \tag{2.13}$$

This factor is then applied to data and ensures that the reconstructed jets are properly scaled across the full momentum range, mitigating systematic effects and non-linearities introduced by the detector. The corrections therefore compensate for detector effects such as non-linear responses, energy losses in inactive material, and calibration offsets that alter measured jet p_{T}.
In practice, the correlation between η and the overall energy scale is assumed to be sufficiently small, which allows the factorization:

$$C_{\mathrm{L2L3}}(\eta, p_{\mathrm{T}}) = C_{\mathrm{L2}}(\eta) \times C_{\mathrm{L3}}(p_{\mathrm{T}})\ . \tag{2.14}$$

In general, some correlations may exist, but they are typically small enough so that the factorized approach remains a valid approximation for practical purposes. This combined L2L3 correction simplifies the overall correction procedure by encapsulating both the relative and absolute corrections in one step. However, it may influence the distinct systematic uncertainties associated with each component. Choosing between a separated or combined approach therefore depends on the specific requirements of an analysis.
In summary, the L2 (relative) corrections are derived in bins of η to address detector non-uniformities, while the L3 (absolute) corrections are derived in bins of p_{T} to correct the overall energy scale. Although the underlying methodology is similar, each targets a distinct aspect of the jet energy measurement.

2.3.3. L2Res: Relative Residual Corrections

After applying the MC-based L2L3 corrections, small differences between data and simulation can still persist due to limitations in the modeling of the detector or the physics processes. These remaining differences are mitigated with the data-driven L2Res correction level for different η regions. For this, measured jets from different η regions are compared to well-calibrated reference objects from the barrel region ($|\eta| < 1.3$). For deriving the relative residual response corrections, two different methods were developed for the CMS experiment: the p_{T} balance method and the missing transverse momentum projection fraction (MPF). Both methods are described in detail in Ref. [3] and Ref. [18] and are briefly summarized in the following.

Direct (p_{T}) balance method: For deriving the correction factors based on the p_{T} balance[16] method, the asymmetry between probe jets and reference (tag) objects is calculated per η bin:[17]

$$A^{\mathrm{DB}} = \frac{p_{\mathrm{T}}^{\mathrm{probe}} - p_{\mathrm{T}}^{\mathrm{tag}}}{p_{\mathrm{T}}^{\mathrm{probe}} + p_{\mathrm{T}}^{\mathrm{tag}}} \,. \tag{2.15}$$

Here, the denominator describes the categorization in bins of the average jet transverse momentum rather than solely by the tagged jet's p_{T}. This symmetric p_{T} binning is done to cancel out the first order relative biases from Initial-State Radiation (ISR) and Final State Radiation (FSR) and minimizes JER biases (see Ref. [3, p. 28] for more details). This asymmetry is then calculated and averaged over all events per η bin that fulfill certain quality criteria. The relative response in η follows as:

$$R^{\mathrm{DB}}(\eta) = \frac{1 + \langle A^{\mathrm{DB}} \rangle}{1 - \langle A^{\mathrm{DB}} \rangle} \,, \tag{2.16}$$

leading to the corresponding correction factor derived from the direct balance:

$$C_{\mathrm{L2Res,DB}}(\eta) = \frac{1}{R^{\mathrm{DB}}(\eta)} \,. \tag{2.17}$$

[16] Often, also referred to as direct balance (DB).

[17] Note: This describes the derivation of the corrections only conceptually. In practice, additional effects may be considered, as described in [18, pp. 15–18].

MPF method: Instead of relying directly on the measured jet p_{T}, the MPF method considers the overall hadronic recoil in an event by projecting the missing transverse momentum, $\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}}$ (see Eq. (2.6)), along the direction of the reference object.
Here, the response is calculated accordingly as:

$$R_{\mathrm{rel}}^{\mathrm{MPF}}(\eta) = \frac{1 + \langle A^{\mathrm{MPF}} \rangle}{1 - \langle A^{\mathrm{MPF}} \rangle} \,, \tag{2.18}$$

with:

$$A^{\mathrm{MPF}} = \frac{\vec{p}_{\mathrm{T}}^{\,\mathrm{miss}} \times \left(p_{\mathrm{T}}^{\mathrm{tag}} / p_{\mathrm{T}}^{\mathrm{probe}} \right)}{p_{\mathrm{T}}^{\mathrm{probe}} + p_{\mathrm{T}}^{\mathrm{tag}}} \,. \tag{2.19}$$

The correction factor is accordingly obtained as:

$$C_{\mathrm{L2Res,MPF}}(\eta) = \frac{1}{R^{\mathrm{MPF}}(\eta)} \,. \tag{2.20}$$

Mitigation of additional jet activity: It is important to note that both techniques assume an idealized scenario in which the event consists of a single jet produced in the collision. In practice, this assumption is rarely met due to the presence of pileup contributions, soft radiation, and other effects (see Ref. [18]). To mitigate the impact of these effects and to ensure a reliable calibration, strict event selection criteria are applied. For this, the additional (background) jet activity not coming from the leading jet is then calculated as:

$$\alpha = \frac{p_{\mathrm{T}}^{\mathrm{sub}}}{p_{\mathrm{T}}^{\mathrm{probe}} + p_{\mathrm{T}}^{\mathrm{tag}}} \,, \tag{2.21}$$

where $p_{\mathrm{T}}^{\mathrm{probe}}$ is the transverse momentum of the probe jet, while $p_{\mathrm{T}}^{\mathrm{tag}}$ can be a reference jet or another well-measured physics object. $p_{\mathrm{T}}^{\mathrm{sub}}$ describes the transverse momentum of a sub-leading jet contaminating the event. Based on this parametrization, an additional event selection criterion is introduced which ensures the event to be suitable for the correction methods. In practice, $\alpha < 0.2$ [18] is typically chosen as a threshold. As shown in Ref. [18], this choice is determined as the region where the calibration results remain stable and reliable and therefore represents a balance between reducing the contamination and keeping sufficient statistics for the derivation of the correction factors.
The correction factors are then evaluated in different α bins. By fitting a functional form to the obtained calibration factors in dependence of α, the corrected calibration factors can be estimated by extrapolating the fit to $\alpha \to 0$. The obtained value then approximate the calibration factor that would be obtained in the absence of any additional jet activity. This α extrapolation therefore effectively mitigates the impact of the above mentioned effects and ensures that the derived calibration factors accurately reflect the intrinsic detector response.

Comparison of the methods: In general, both methods are capable of providing a reliable calibration, but each has its own strengths and weaknesses. The direct balance method is more simplistic and directly compares the measured energy with a well-known reference object. However, the method is more sensitive to additional jet activity directly impacting the momentum balance, and requires a clean event topology to maintain a high accuracy – which can be critical in practice.

While the MPF method is less sensitive to jet resolution effects and additional activity in general[18], it specifically requires a well-modeled MET. This in turn means that uncertainties or mis-calibrations in jet reconstruction and energy measurements within the calorimeters have a more significant impact on the Jet Energy Corrections derived using the MPF method.

Overall, the advantages of the MPF outweigh the disadvantages for most analyses, which is why it is used in practice to derive the L2Res corrections, while the direct p_{T} balance method typically serves as an important cross-check and validation of the results. To actually derive these corrections, dijet events with a suitable topology are used.[19] After applying the corrections in data, a uniform response throughout all η regions to an even more precise level is achieved.

2.3.4. L3Res: Absolute Residual Corrections

The absolute residual corrections (L3Res) is another data-driven calibration step designed to supplement the L3 corrections. It fine-tunes the absolute jet energy scale and corrects for remaining residual differences between data and simulation after the L2L3 corrections have been applied. For high-precision measurements, this step is of outmost importance, since already small discrepancies in the jet energy scale can lead to significant systematic uncertainties in physics analyses.

Deriving the L3Res corrections on measured data employs the same methods, p_{T} balance and MPF, as introduced for the L2Res corrections in Section 2.3.3. Just as before, the α extrapolation is also used for the L3Res corrections, however, typically with a working point of $\alpha < 0.3$ [3, p. 36]. The difference to the relative residual corrections is, just as for L2 and L3, that while the relative corrections focus on normalizing the jet energy response across various detector regions as a function of η, the L3Res corrections adjust the absolute energy scale in p_{T}.

Hence, for deriving the L3Res corrections, the simulation-based corrections and the relative residual corrections are applied and events are selected with a central jet balanced in a back-to-back topology against a well-calibrated reference object, as depicted in Fig. 2.6 with a Z boson. Alternatively, γ+jet and multi-jet events are used as well. Here, however, the focus will be on the Z+Jet channel as basis for this thesis – but the principles are similar for the others. For further details, it is referred to Refs. [3, 18].

The channels Z→ $\mu^+\mu^-$ and Z→ e^+e^- are particularly chosen due to the clean final signature of such events. While the derivation based on muons is in general more accurate (with smaller systematics), electrons are additionally considered to increase the sample size and maintain sufficient statistics while employing high quality cuts.

According to the residual corrections of the previous level, the response therefore

[18] A detailed comparison can be found in Figure 19 of Ref. [3].

[19] In principle, a derivation based on events with well measured reference objects, such as Z bosons or photons, is also possible.

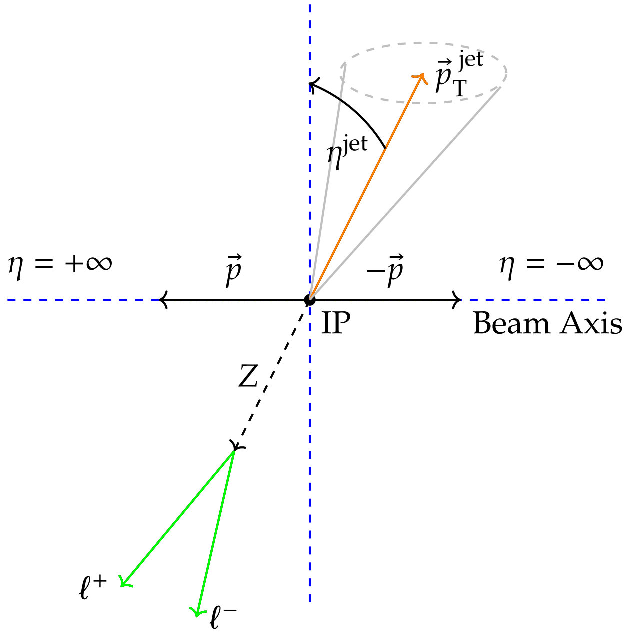


Figure 2.6.: Schematic representation of a Z+Jet event as used for absolute residual Jet Energy Corrections (L3Res). The Z boson (dashed arrow) decays into two leptons (green), $\ell^+\ell^-$, with $\ell = e, \mu$, in a back-to-back topology to a single jet (orange arrow). The reconstructed transverse momentum of the jet, $\vec{p}_{\mathrm{T}}^{\,\mathrm{jet}}$, is then balanced against the momentum of the Z boson as reconstructed from the precisely measurable decay products. This reflects the desired event topology for the calibration based on Z+Jet events and applies similar to the other possible channels.

can be derived as:[20]

$$R_{\text{abs}}^{\text{MPF/DB}}(p_{\text{T}}) = \frac{1 + \langle A^{\text{MPF/DB}} \rangle}{1 - \langle A^{\text{MPF/DB}} \rangle} \,, \tag{2.22}$$

with:

$$A^{\text{DB}} = \frac{p_{\text{T}}^{\text{jet}} - p_{\text{T}}^{\text{Z}}}{p_{\text{T}}^{\text{jet}} + p_{\text{T}}^{\text{Z}}} \quad \text{and} \quad A^{\text{MPF}} = \frac{\vec{p}_{\text{T}}^{\,\text{miss}} \times \left(p_{\text{T}}^{\text{Z}}/p_{\text{T}}^{\text{jet}}\right)}{p_{\text{T}}^{\text{jet}} + p_{\text{T}}^{\text{Z}}} \,. \tag{2.23}$$

The correction factor is accordingly obtained as:

$$C_{\text{L3Res,MPF/DB}}(p_{\text{T}}) = \frac{1}{R^{\text{MPF/DB}}(p_{\text{T}})} \,. \tag{2.24}$$

When applying this correction factor to data, differences between the jet transverse momentum and its true energy scale are reduced, ensuring the highest possible precision in p_{T} measurements so far.
In summary, the L3Res absolute residual corrections are a crucial step that refines the jet energy scale after simulation-based and relative corrections are applied to data and MC, thereby minimizing systematic uncertainties and enhancing the reliability of physics analyses at CMS.

2.3.5. L2L3Res: Combined Residual Corrections

The L2L3Res corrections are essentially the combined residual (data-driven) corrections, analog to the L2L3 corrections. With the same reasoning it is possible to simultaneously addresses both the relative, η-dependent, and absolute, p_{T}-dependent discrepancies between data and simulation after applying the MC-based L2L3 corrections. Although in practice the residual corrections are typically derived separately with dijet events for the relative part and γ/Z+Jet as well as multi-jet events for the absolute part, deriving combined L2L3Res corrections using a channel like Z+Jet can again serve as a useful cross-check to validate the consistency and robustness of the individual residual corrections.

2.3.6. Global Fit

Finally, the absolute jet energy scale is fitted simultaneously to the different channels mentioned in Section 2.3.4. Note that the p_{T} range differs per event type [3]. The fit is implemented as a χ^2-minimization, with uncertainties being incorporated as nuisance parameters and added quadratically to χ^2, as described in Ref. [3, p. 39]. From the fit, the final jet response corrections are ultimately extracted. It is important to note that JEC is derived iteratively, with the calibration procedure (including JER) repeated until convergence is achieved.
Further details on the fitting procedure are provided in Ref. [3] and latest results for the Legacy data of Run 2 are presented in Refs. [57, 58].

[20] A more detailed derivation including additional aspects can be found in Ref. [3, pp. 32–38].

3. Absolute Residual Jet Energy Corrections with Z+Jet Events

> [T]hroughout the infinite, the forces are in a perfect balance, and hence the energy of a single thought may determine the motion of a universe.
>
> (On Light and Other High Frequency Phenomena – Nikola Tesla)

This chapter presents the first derivation of the absolute residual correction (L3Res) as part of the JEC for the CMS experiment. The corrections are derived for the most advanced reprocessing of CMS Run 2 data ($\sqrt{s} = 13\,\text{TeV}$) at the time of writing[1], internally referred to as Ultra Legacy. It incorporates the latest reconstruction algorithms, calibrations, and alignment improvements to enhance data quality and precision.

In the following, the corrections are derived based on Z+Jet events in the two channels: $Z \longrightarrow \mu^+\mu^-$ and $Z \longrightarrow e^+e^-$. For this, the methods described in the previous chapter have been adapted to the specific requirements and recommendations for the Ultra Legacy reconstruction of the data recorded between 2016 and 2018.

The derivation process involves several key steps, starting with the data selection and preparation together with a skimming of the data. Afterwards, the cut-based event selection and the applied corrections are described and first results for the absolute residual correction are presented. The last part of the chapter discusses the preparations for Run 3 carried out as part of this work.

Remark: For simplicity, the datasets and periods are often abbreviated with e.g. 2016UL or even UL16. This is therefore used here as well according to the common standards in the JERC environment.

3.1. Data Selection and Preparation

At first, the datasets for the derivation of the Jet Energy Corrections are selected and prepared. For the data selection, the recommendations from Ref. [60] were followed. The derivation is utilizing the CMS `DubleMuon` (2016UL, 2017UL, 2018UL), `DoubleEG` (2016UL, 2017UL), and `EGamma` (2018UL) datasets in the MiniAODv2 format [39], as provided in Listing A.1. These datasets are additionally segmented in individual

[1] Note: The JEC is an iterative process until sufficient convergence. The here described corrections were derived based on JECv6 and JECv7 (see [59]) and provided in January 2023 to the CMS Collaboration.

so-called Intervals of Validity (IoVs), see Ref. [61]. The IoVs are selected to have a consistent set of detector conditions and are often also called *Run A,B,C,* and so on – not to be confused with the Runs of the LHC.
For 2016, it is additionally to mention that the dataset is further divided into two parts: APV/preVFP corresponding to the IoVs $BCDEF_{early}$ and non-APV/postVFP ($F_{late}GH$) [62]. This is due to different changes in the tracking systems, which reflect a significant change in the detector behavior and make individual derivations of the corrections for the periods necessary. This separation therefore ensures the correct accounting for the different detector conditions and calibrations associated with each period.
Since the residual corrections correct for remaining differences between data and simulations, matching MC datasets are required for the individual periods. The selected MC datasets, based on the recommendations in Ref. [62], are also listed in Listing A.1.
Once the appropriate data has been selected, pre-processing takes place. For this purpose, the well-tested skimming tool `Kappa` [63], developed and maintained by the KIT-CMS group, was adapted to the Ultra Legacy datasets and employed.

Skimming: As a first data preparation step, the datasets are reduced – commonly called skimming. The skimming process has two main tasks: condensing the datasets to the required minimum, and applying filters and corrections. This means, through the skimming, unnecessary events are filtered out and the remaining, useful events are reduced to only keep the parts relevant for the derivation of the corrections, thereby greatly reducing the overall size of the datasets. Such a preprocessing step is in principle not necessary, but strongly recommended, since it not only improves computational efficiency by discarding irrelevant events that do not need to be processed later on, but also simplifies storage, data handling, and reprocessing.
As an example, the total amount of Ultra Legacy data and MC simulations (both in the MiniAODv2 format) used for the here presented derivation of the L3Res corrections for Run 2 sums up to around 135 TB (of which approx. 24 TB are MC). After the skimming step, the pre-processed datasets are reduced to less than 70 TB, reducing the data that needs to be processed in later steps therefore by around 50 %. This significant reduction not only saves storage space but also greatly reduces the amount of core hours required for a reprocessing of the corrections, for example, with changed configurations or additional filters.
The skimming procedure for the Ultra Legacy datasets was configured to match the specific characteristics of each data-taking period, following the official recommendations [62, 64, 65, 66, 67, 68, 69], to ensure that the unique conditions of each subset are properly accounted for. At first, all relevant PF candidates and the *AK4CHS* jet collections (see Section 2.2.6.1) are selected. Additionally, the up-to-date JEC [59] and JER corrections – as basis for the iterative correction process – are carried out to the jet collections (see Listing A.3). Additional information on the involved uncertainties are e.g. providedin Ref. [70].
Furthermore, during the skimming of the datasets, additional flag-based filters are applied or pre-selected. The latter means that they are not yet applied at this stage, but only the ones that are relevant for the analysis are kept, such as the MET (or Noise) Filters, as described in Section 2.2.6.2. The full list of utilized MET filters is

provided in Section A.1.2. In case of the HLTs, this means that they are not explicitly enforced at this stage, since it can be useful to keep the data flexible in a sense that at a later stage of the analysis, different triggers still can be applied. Instead, they are only filtered based on the configured selection for the derivation of the L3Res corrections as listed in Section A.1.3 – which could in principle be extended by every available trigger flag in Ref. [71], if required. With this design, the skimming step is prevented from having to be repeated more often than necessary. This has the disadvantage that more data may be stored than is actually used later. However, this is still a good compromise in terms of computing power, as the entire skimming process can require tens of thousands of core hours for such an analysis (depending on the selected datasets).

Additionally, further quality criteria, such as *JetID*, *ElectronID*, or *MuonID* are preselected, which later on ensure that only events are used that match the basic quality criteria for the derivation of the L3Res corrections. Details on the selections and IDs are provided in Section A.1.4.

On top of that, different sub-groups of the CMS collaboration provide additional corrections and recipes for the processing of the datasets, for example *Energy Scale and Smearing* corrections by the EGamma group [68], which are also applied on-the-fly while skimming the datasets. For details, see Section A.1.5.

In conclusion, the skimming is a very useful procedure that prepares the data for the further processing while reducing the size at the same time. From the computing perspective, it is therefore highly recommended when using MiniAOD (or or even larger data tiers), as it reduces the resource demand and increases the overall efficiency of the further processing, since less data needs to be transferred, processed, and stored.

3.2. Cut-Based Event Selection

With this step, it is ensured that the event data is of the required quality to provide reliable and accurate corrections. Because especially for the direct p_{T} balance method, clean events with a very specific topology are required, as discussed in Section 2.3.3. Conceptually, this selection is rather simple: Only keep events with a high-energy leading jet that has been produced in a back-to-back topology with a Z boson, reconstructed from its decay products ($\mu^+\mu^-$ or e^+e^-). This was visualized in Fig. 2.6. However, as also discussed in the aforementioned section, such a clean environment is rather unlikely and some compromises have to be made, especially in the jet selection.

The cut-based event selection is carried out by the calibration framework [72], which is an extension to the MiniAOD processing framework [73]. Both of them were reviewed, updated, and adapted to the requirements and recommendations for Ultra Legacy data as part of this thesis.

First, a basic filter is applied to exclude events, which are marked as unreliable by the CMS Collaboration. For this, so-called Golden JSONs are provided centrally. These are simple text files that include the run and luminosity sections in which the detector was fully functional and the recorded data meets the central quality requirements. The application is carried out by simply comparing the sections and removing events that are outside the validated intervals.

After this, the HLTs and MET (Noise) filters are applied, see Section A.1.3 and Section A.1.2. Note that the HLTs were only pre-selected in the skimming step and are enforced here by the fact that the corresponding trigger flags must be set for an event that passes the selection.

As a next step, lepton IDs [66] and cuts are evaluated on event level to ensure that each used event contains sufficient high-quality leptons contained within the central region of the detector ($|\eta| < 2.3$ for muons and $|\eta| < 2.4$ for electrons). During this step, also so-called *Rochester corrections* [67] for muons are applied, see Section 2.3. The Z boson is then reconstructed and assessed whether it fulfills the desired quality criteria, such as a minimum p_{T} of at least 15 GeV and a sufficient agreement in the reconstructed mass, enforced as $|m^{Z}_{\mathrm{reconstructed}} - m^{Z}_{\mathrm{PDG}}| < 20\,\mathrm{GeV}$. The full list of cuts is provided in Section A.1.6 and the used IDs and Isolation requirements for the leptons are listed in Section A.1.4.

After the muons are sanitized, the jets are processed[2]. Of course, at first it is required that at least one jet within the central region of the detector ($|\eta| < 1.3$) can be found with a transverse momentum of $p_{\mathrm{T}}^{\mathrm{jet}} > 12\,\mathrm{GeV}$. In terms of sub-leading jets, the α working point is required to be $\alpha = p_{\mathrm{T}}^{\mathrm{secondjet}} / p_{\mathrm{T}}^{\mathrm{Z}} < 1$.

Furthermore, so-called hot-zone maps (also called jet veto maps) [74] are applied. They are centrally provided by the responsible sub-group [75] and map regions in the CMS detector with known issues, or where jets are observed to be inconsistent[3]. By applying these maps in the event selection, it is ensured that the measurement of the leading jet's energy can be trusted and is not altered by anomalous effects in these problematic regions, reducing systematic uncertainties in the JEC. The used maps for the L3Res corrections are listed in Section A.1.4.

With the leading jet and the Z boson in place, the back-to-back topology – as depicted in Fig. 2.6 – is enforced with $|\Delta\phi(\mathrm{leadingjet}, Z) - \pi| < 0.44$.

Finally, it is required that jets fulfill certain quality criteria represented by the JetID and isolation from leptons (*tightLepVeto*) and are not a pileup jet (PuJetID [76, 48]). This application at the end of the cutflow has practical reasons related to the framework synchronization process described in Section 3.4, since the jet collections can slightly differ between MiniAOD and NanoAOD. However, the impact of possible slight differences is greatly mitigated by the overall very high quality requirements on the events that already have been enforced. This is clearly visible in the full cutflow for 2017UL, as shown in Section A.1.6. The cut on the JetID does not have a significant impact on the event selection anymore, which is accordingly expected. The criteria used for the calibration of 2017UL and 2018UL are given in Table A.2. For 2016UL, the requirements slightly differ, as presented in Table A.1.

Furthermore, MC simulated events are processed with the same reconstruction algorithms and cutflow as the recorded data. However, certain details, such as the pileup conditions, are typically not perfectly modeled in simulation. To address this, a pileup reweighting procedure is implemented, see Refs. [77, 78]. In this process, the pileup distribution observed in data (derived from luminosity measurements and vertex counting) is compared to the pileup distribution in the MC simulation. Event-by-event weights are then calculated based on this comparison and applied to

[2] For details on the applied lepton cleaning of the jets, see Section A.1.4.

[3] In the vetoed regions, unusually high noise levels or spurious (too high) energy deposits are observed, leading to 'hot' areas, much like hotspots in thermal images – therefore the naming.

the simulated data. This reweighting adjusts the MC pileup profile to more closely match that of data, which improves the overall agreement between simulation and recorded events and reduces related systematic uncertainties. The used pileup files are listed in Section A.1.4.

As previously mentioned, this entire cutflow is presented for 2017UL in Section A.1.6. After finishing the data preparation and selection steps, the absolute residual corrections can now be derived based on a trustworthy foundation.

3.3. Derivation of the Absolute Residual (L3) Corrections For Ultra Legacy Data

Based on the previously described methods and selections, the absolute residual corrections have been derived in the $Z \rightarrow \mu^+\mu^-$ and $Z \rightarrow e^+e^-$ channels for Ultra Legacy data for the years 2016 to 2018. The results were made available to the CMS Collaboration in form of Combination Files [79].

In the following, results and control plots for **2018UL** are presented. Note that all of the provided distributions with a gray shade are normalized to unit area to allow for a direct comparison of their shapes, independent of their absolute yields. The results for the other years are contained in Section A.1.7.

At first, to ensure valid corrections for a reliable and accurate calibration of the jet energies, the input has to be validated. A good cross-check for this purpose is the Z boson mass distribution – relevant for the balancing in general – as well as the modeled MET, on which the MPF method relies. The Z boson mass is shown in Fig. 3.1, reconstructed from muons (left) and electrons (right). As clearly visible in the peak region, data and MC are in good agreement for both muons and electrons, indicating that all relevant effects are sufficiently well modeled in the simulations. The overall agreement, including the tails, is slightly better for muons. This is expected, as muons are measured with higher precision due to their minimal energy loss in the tracker and calorimeters, while the systematic uncertainties for electrons are generally higher. Additionally, the electron (EGamma) energy scale and resolution corrections in the simulation may not fully capture all detector effects ideally, contributing to the observed discrepancies in the tails. In general, since the deviations for the tails are significantly higher than in the immediate vicinity of the peak, it is considered for the future to reduce the Z mass window to 10 GeV, since the statistics are expected to be sufficient anyway.

The distribution of the MET is depicted in Fig. 3.2. In the range of $E_{\mathrm{T}}^{\mathrm{miss}} < 50$ GeV, the MET in data and MC is in good agreement. Above this value, the differences increase. Here, it seems that the missing transverse momentum is overestimated in the simulations. However, two of the IoVs are still in good agreement – consistent between the channels. This gives a hint that the reason is probably related to specific conditions in these two intervals. Nevertheless, since statistics are less dominant in the tail of the distribution, the impact is expected to be low.

This is underlined by the fact that the response derived with the MPF method is in good agreement, as shown on the right in Fig. 3.3. For electrons, the same is depicted in Fig. 3.4. Both figures show the comparison between the L2Res (left) and L3Res (right) correction levels. For both channels, the overall good agreement between data and simulation flattens out even further, reflecting the increased precision with

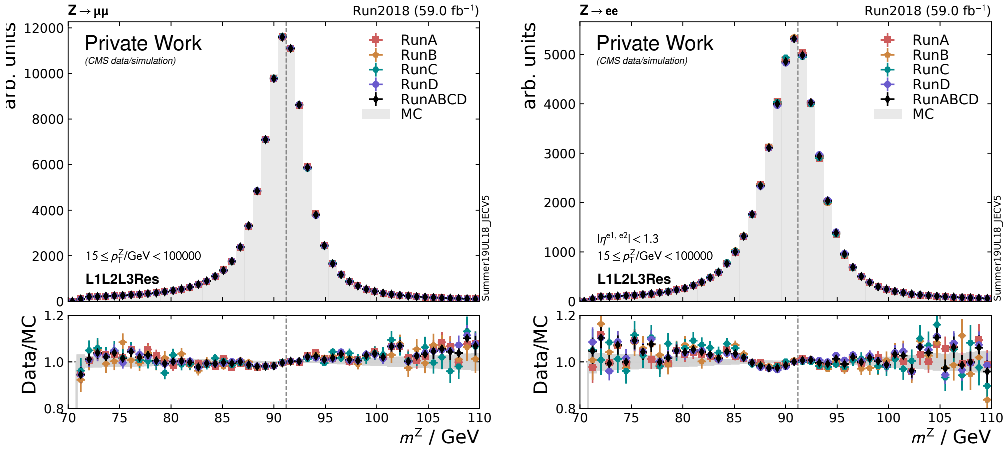


Figure 3.1.: Z boson mass distributions for 2018UL in the $\mu^+\mu^-$ (left) and e^+e^- (right) channels, respectively. In the region around the peak, both distributions are well modeled in the simulations. For the tails, however, the deviations are increasing, especially for the electron channel.

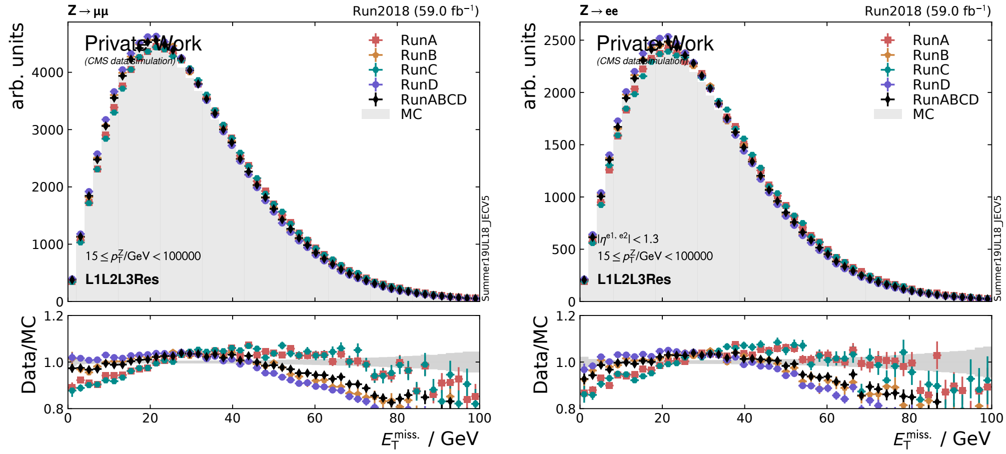


Figure 3.2.: MET distribution for the Ultra Legacy data (2018UL) in comparison to simulation, with different data-taking periods shown separately for muons (left) and electrons (right). The overall shapes are in good agreement. But as the data/MC ratio below shows, for both electrons and muons the modeling becomes worse with decreasing statistics above around 50 GeV indicating potential discrepancies. Remarkable is also the difference between the IoVs: While Run A and C seem to be in good agreement above 20 GeV, the channels B and D are worse and overall dominating the picture. However, since most events are contained in the region with good agreement, the impact is expected to be minor.

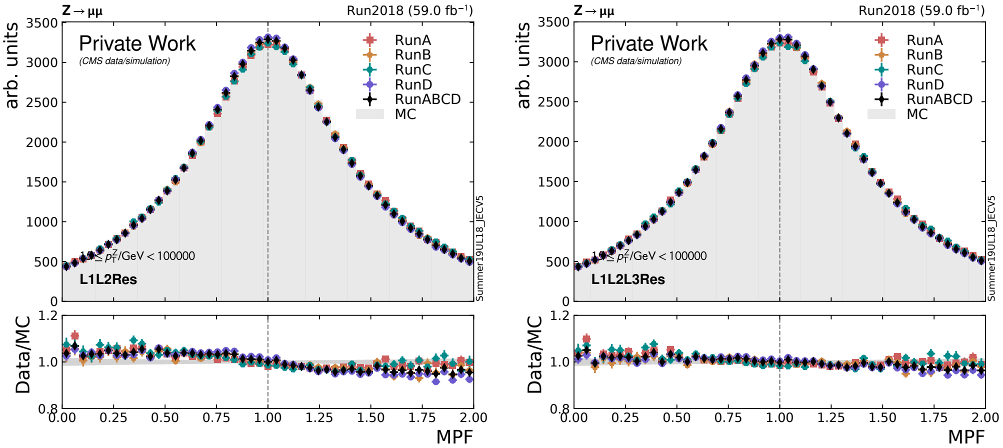


Figure 3.3.: Comparison of the response in the muon channel derived with the MPF method at L2Res level (left) and L3Res level (right). The absolute residual corrections bring the ratio into even better agreement, which underlines the effectiveness of the corrections for high-precision requirements.

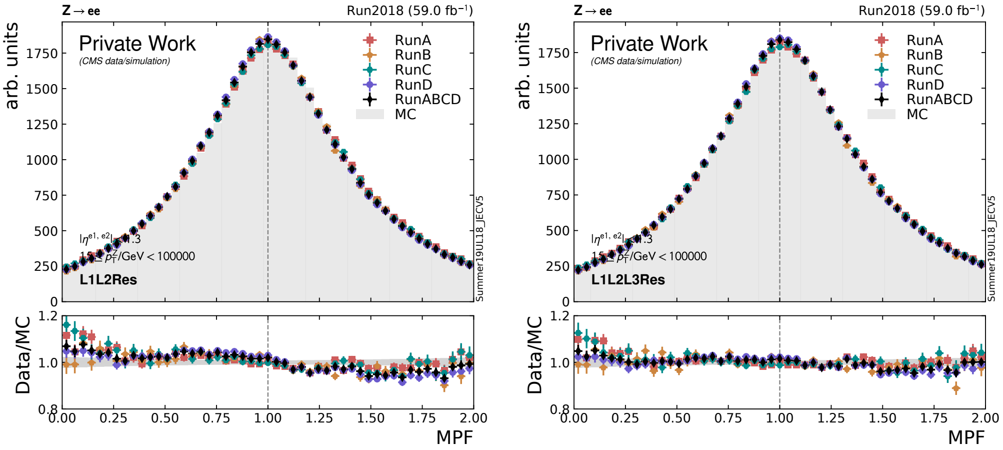


Figure 3.4.: Comparison of the response in the electron channel derived with the MPF method at L2Res level (left) and L3Res level (right). Here, the impact of the highest-order correction is even stronger in the tails, resulting in exceptionally good agreement.

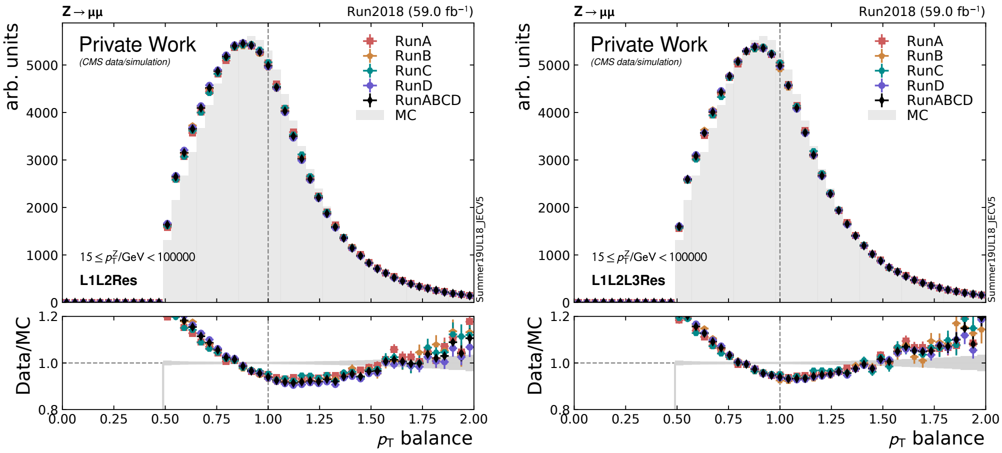


Figure 3.5.: Comparison of the response in the muon channel derived with the direct p_T balance method at L2Res level (left) and L3Res level (right). The overall agreement between data and simulation is significantly worse than for the MPF method, as expected.

the absolute residual corrections in place. With the L3Res corrections, the maximum deviation in the muon channel is less than 3 % and for the electron channel below 5 % for most of the total range. This result underlines the effectiveness of the corrections to achieve maximum accuracy.

In Fig. 3.5 and Fig. 3.6, the same comparison is depicted, but with the response evaluated based on the direct p_T balance method. Here, both channels yield again very comparable results. However, the overall deviations between data and MC are clearly larger than for the MPF method. This supports the statement that the MPF method should mainly be considered in the global fit (see Section 2.3.6), while the direct p_T balance could be used as a complementary method when appropriate or for validation and cross-checking.

Finally, to further validate the results and to quantify the impact of the corrections more precisely, it is worth to compare the methods in dependence of p_T and η directly for the different correction levels. MPF method.

When comparing the overall agreement of the MPF response derived in the muon channel in dependence of the pseudorapidity of the leading jet, η^{jet1}, for the L2Res (left) and L3Res (right) correction levels, an improvement in the order of 1 % is achieved with the absolute residual correction – which is significant in context of the high-precision that is desired.

Furthermore, when comparing the same, but in dependence of the transverse momentum of the Z boson, p_T^Z, as shown in Fig. 3.8, again a significant improvement is observed with the absolute residual corrections. Especially remarkable is the excellent agreement between data and simulation at the sub-percent level in the range of 50 GeV to 350 GeV, indicating a precise calibration of the absolute scale.

In Fig. 3.9 and Fig. 3.10, the according distributions are presented for the electron channel with an overall similar result. As the second figure shows, above 50 GeV again an excellent agreement is obtained at the L3Res correction level.

As an additional cross-check, the direct p_T balance is accordingly derived in dependence of the two quantities in both channels and depicted from Fig. 3.11 to Fig. 3.14.

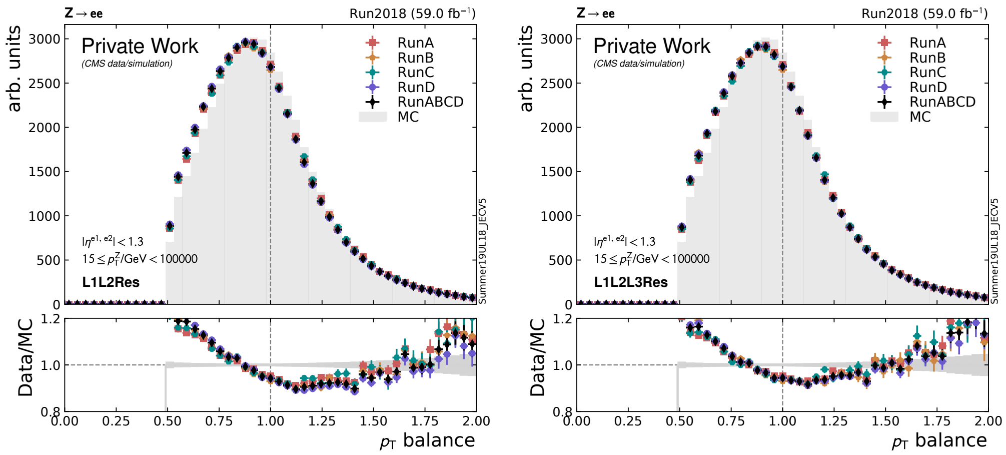


Figure 3.6.: Comparison of the response in the electron channel derived with the direct p_T balance method at L2Res level (left) and L3Res level (right). The result is comparable to the muon channel shown above, but significantly less ideal than the agreement of the response derived with the MPF method.

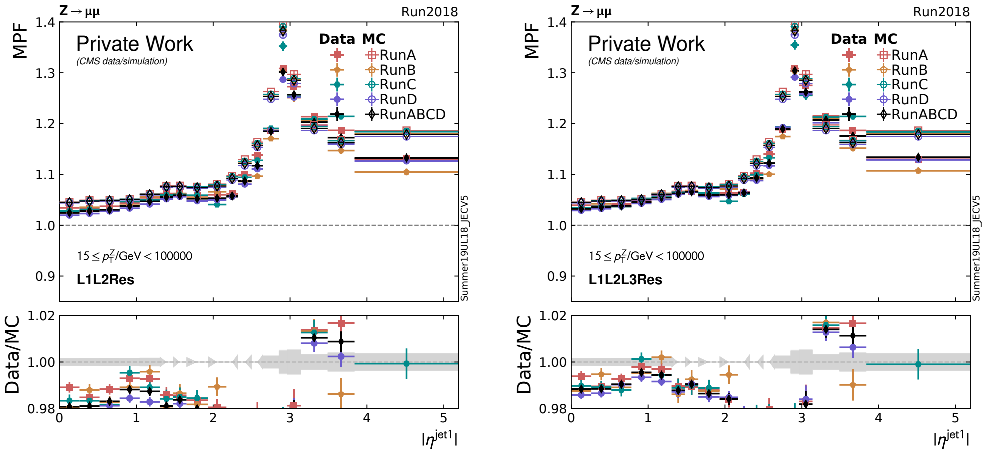


Figure 3.7.: Comparison of the response in the muon channel derived with the MPF method in dependence of the pseudorapidity of the leading jet, η^{jet1}, at L2Res level (left) and L3Res level (right). When comparing the two ratios, an improvement in the order of 1% (absolute) is achieved with the absolute residual corrections for the relevant regions ($|\eta| < 1.3$). Here, the final agreement between data and MC (bottom part of the figures) is good after the L3Res corrections.

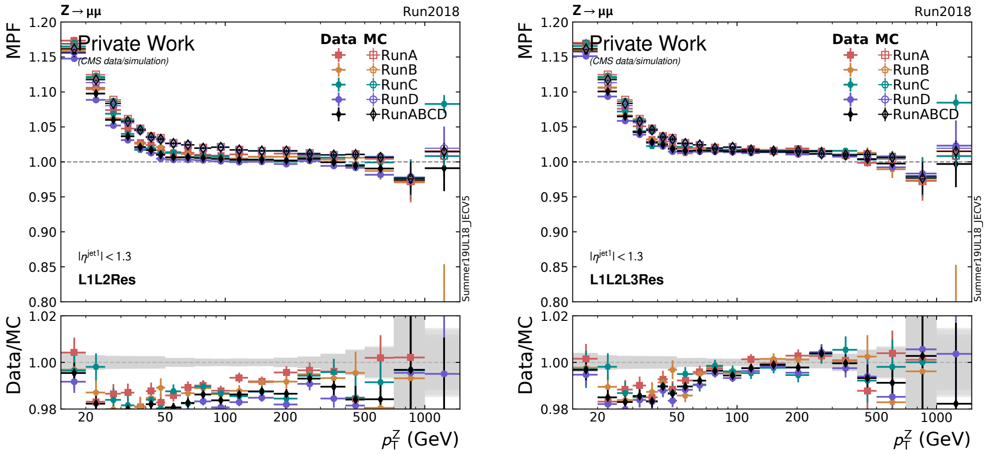


Figure 3.8.: Comparison of the response in the muon channel derived with the MPF method in dependence of the transverse momentum of the Z boson, p_T^Z, at L2Res level (left) and L3Res level (right). The comparison of the ratios between the correction levels clearly shows the impact of the absolute residual corrections. With the latest correction (L3Res) applied, from around 50 GeV to 350 GeV the response in data and MC is in agreement to a sub-percent level.

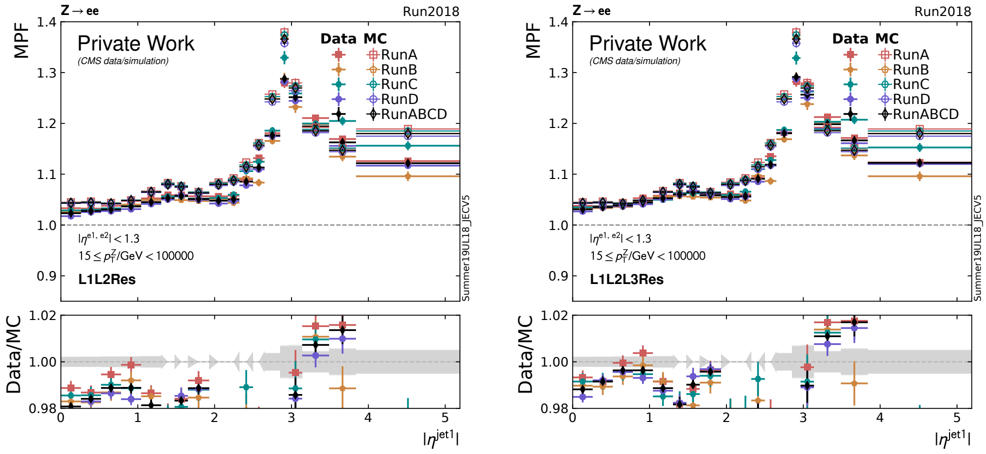


Figure 3.9.: Comparison of the response in the electron channel derived with the MPF method in dependence of the pseudorapidity of the leading jet (η^{jet1}) at L2Res level (left) and L3Res level (right). When comparing the two ratios, again a significant improvement is observed with the absolute residual corrections. Overall, the agreement is comparable to the muon channel.

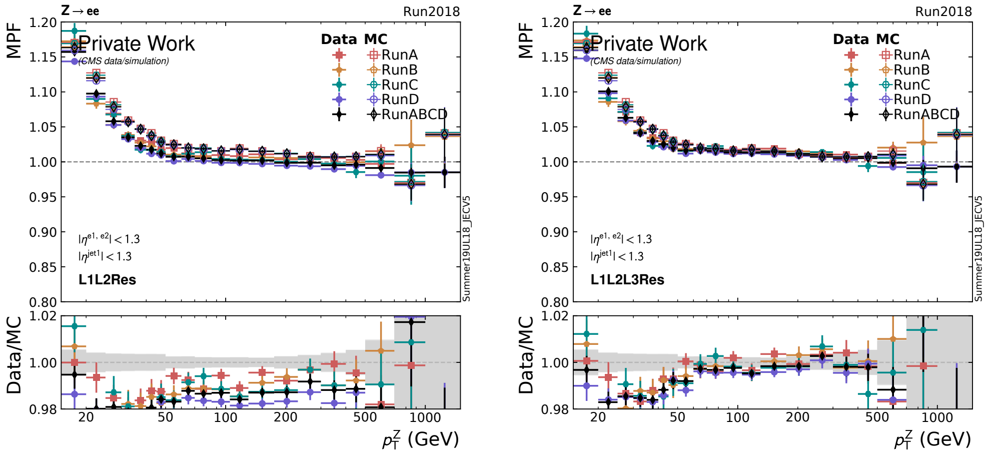


Figure 3.10.: Comparison of the response in the electron channel derived with the MPF method in dependence of the transverse momentum of the Z boson, p_T^Z, at L2Res level (left) and L3Res level (right). Here, the impact is even more pronounced as the electrons nearly reach a perfect data/MC ratio above 50 GeV when using the derived corrections. The ratios therefore again clearly show the impact of the absolute residual corrections. With the latest corrections applied, from around 50 GeV to 350 GeV the response in data and MC is in agreement to a sub-percent level.

Also with this method, a clear improvement is observed throughout all figures when applying the L3Res corrections. However, while the agreement in MPF response is able to partly reach the sub-percent level, here, the discrepancies between data and MC are considerably larger. Nevertheless, this additional method still serves as an important comparison because it shows, firstly, that the absolute residual corrections have a positive impact on the accuracy and, secondly, that the results are overall consistent.

Finally, when additionally comparing these means of the relative ratios directly, the overall improvement with the absolute residual corrections can be observed together in both variables. The comparison for the MPF method is presented for muons in Fig. 3.15 and for electrons in Fig. 3.16. It mainly shows an improvement between the correction levels (L2Res left, L3Res right) in the region above $p_T^Z > 50$ GeV and $|\eta^{\text{jet1}}| < 2$, as observed in the individual distributions. Here, the bin colors are more washed out (white reflects the optimum where MC and data are in perfect agreement). This again underlines the overall good result of the corrections.

For, the direct p_T balance method, the same 2-dimensional shapes are presented in Fig. 3.17 and Fig. 3.18. Overall, the result is slightly worse than for the MPF method (a decently more light-blue overall tone in comparison to Fig. 3.15 is visible), but between the correction levels, an improvement can be observed, validating the concept.

In conclusion, this analysis has shown that the derived absolute residual Jet Energy Corrections are able to reduce great parts of the remaining discrepancies between data and MC simulations for Ultra Legacy. Especially the MPF method yields a great result as the agreement between data and simulation can be brought to a sub-percent level in the most relevant regions, which is crucial for a high-precision

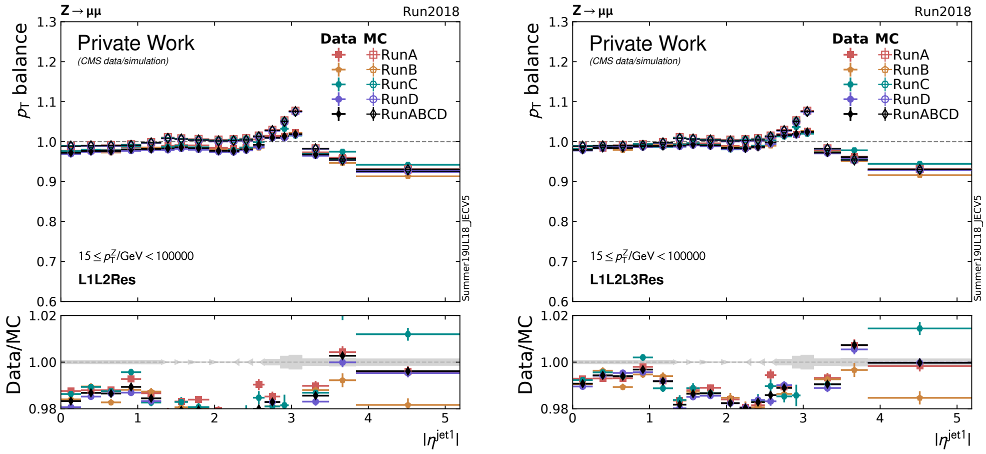


Figure 3.11.: Comparison of the response in the muon channel derived with the direct p_T balance method in dependence of the pseudorapidity of the leading jet, η^{jet1}, at L2Res level (left) and L3Res level (right). Also with this method an improvement up to 1% in the data/MC agreement is achieved with the absolute residual corrections applied. In the low η^{jet1} region, the result is overall comparable with the MPF method.

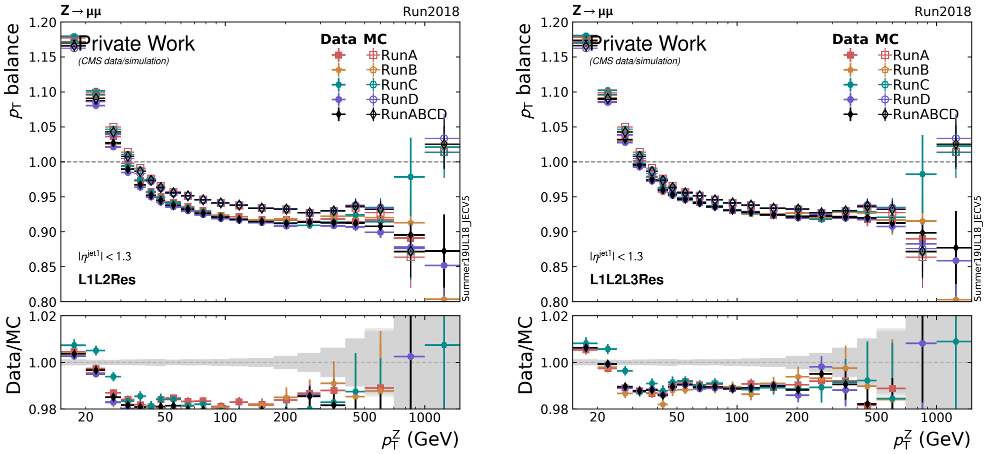


Figure 3.12.: Comparison of the response in the muon channel derived with the direct p_T balance method in dependence of the transverse momentum of the Z boson, p_T^Z, at L2Res level (left) and L3Res level (right). When comparing the two ratios, a clear improvement between the correction levels is visible. However, the overall agreement is about 1% less than for the MPF method with the L3Res corrections applied. This again clearly supports the usage of the MPF method to achieve highest precisions.

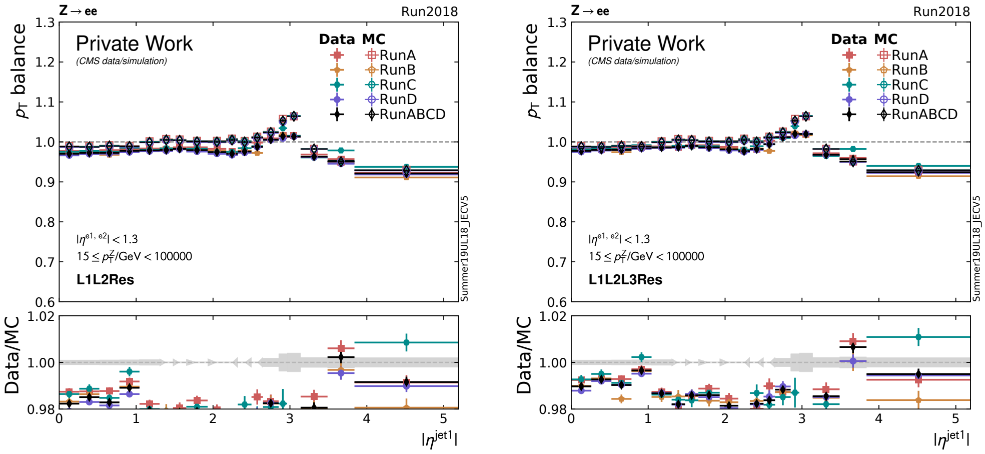


Figure 3.13.: Comparison of the response in the electron channel derived with the the direct p_T balance method in dependence of the pseudorapidity of the leading jet, η^{jet1}, at L2Res level (left) and L3Res level (right). For $|\eta^{jet1}| < 0.9$, the result is comparable to the MPF method. Above, electrons and muons are comparable, however, a slightly worse agreement can be observed in comparison to the MPF method.

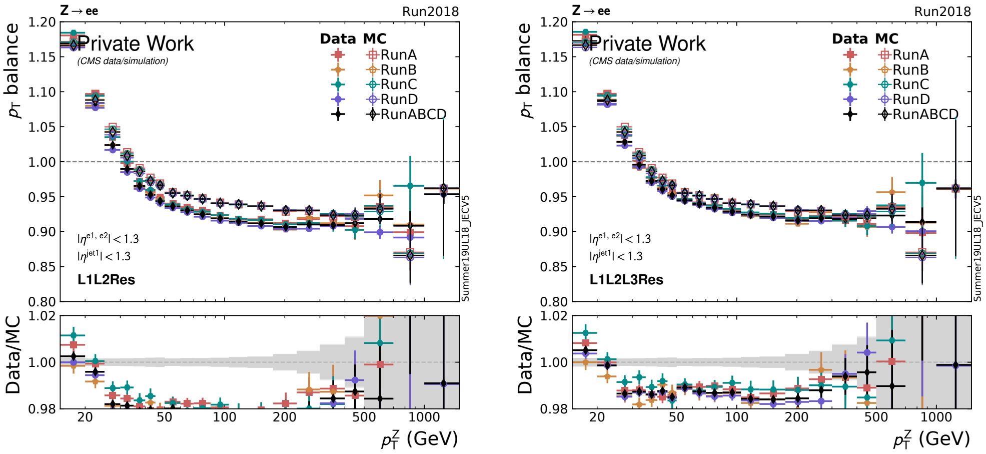


Figure 3.14.: Comparison of the response in the electron channel derived with the direct p_T balance method in dependence of the transverse momentum of the Z boson, p_T^Z, at L2Res level (left) and L3Res level (right). Here, the result is again comparable to the muon channel with L3Res applied, but clearly lacking behind the agreement in data and simulation with the MPF method.

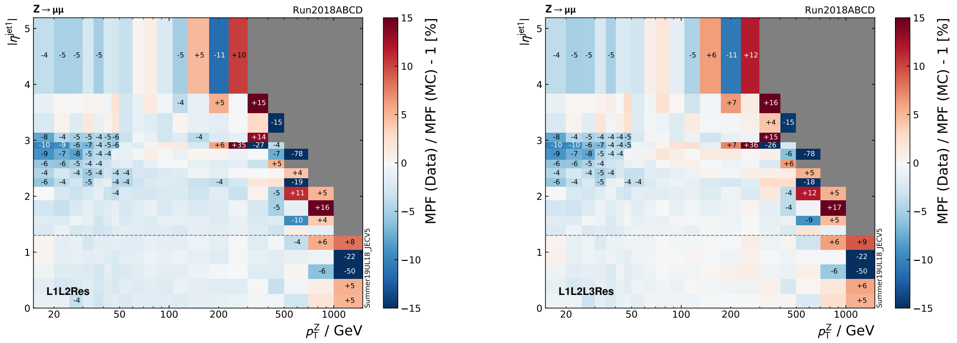


Figure 3.15.: Each figure presents the response derived with the MPF method in dependence of p_T^Z and η^{jet1}. The dashed line reflects the relevant η limit. Left, the L2Res correction level is presented. Comparing it to the right figure, where the L3Res corrections are additionally applied, shows an improvement mainly in the region of $p_T^Z > 50$ GeV and $|\eta^{jet1}| < 2$, where the overall result is close to the optimal agreement In general, applying the L3Res correction therefore results in a very good agreement and minimizes the remaining discrepancies between data and MC simulations.

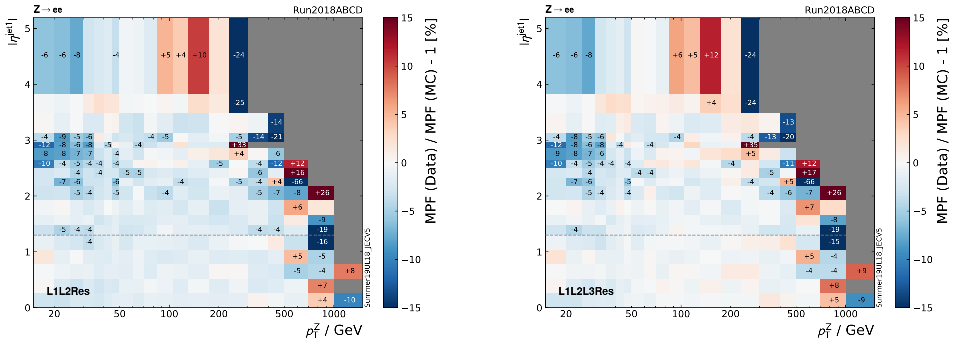


Figure 3.16.: Here, the same comparison as Fig. 3.15 is depicted but for the electron channel. The observations are similar to the muon channel.

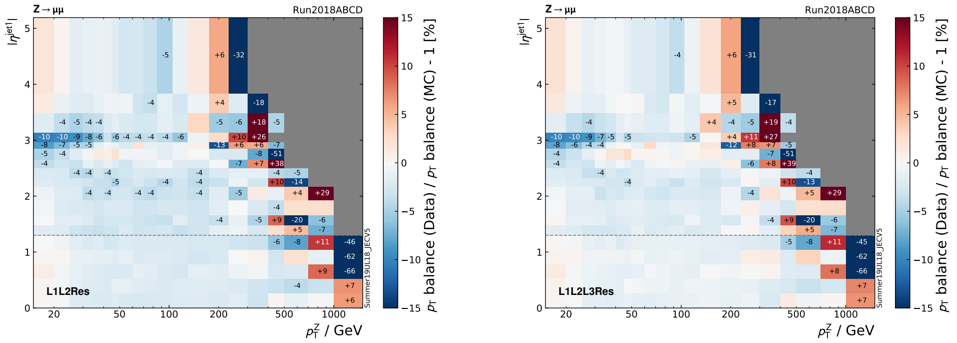


Figure 3.17.: The 2-dimensional comparison for the direct p_T balance method in the muon channel also shows an improvement between the correction levels. The overall result for the response agreement in comparison to the derivation based on the MPF method, however, is slightly worse.

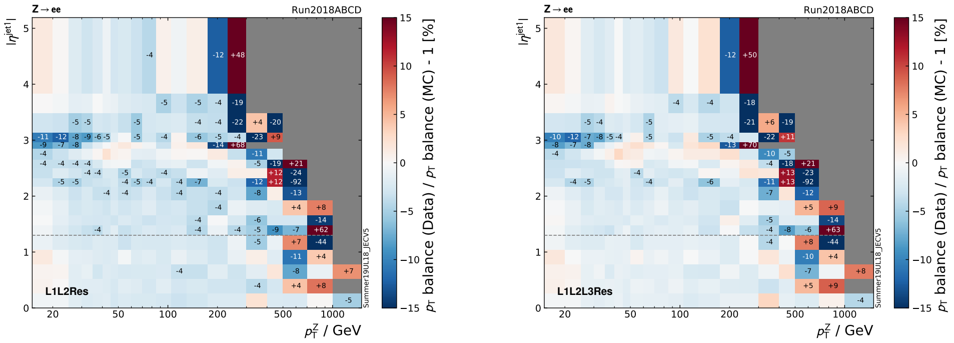


Figure 3.18.: 2-dimensional comparison for the direct p_T balance method in the electron channel. It still shows a slight improvement between the correction levels, but the overall agreement cannot compete with the muon channel, especially not when the MPF method is employed.

calibration. The direct p_T balance is less effective, but serves as a good cross-check for the improvements and the overall consistency of the methods.

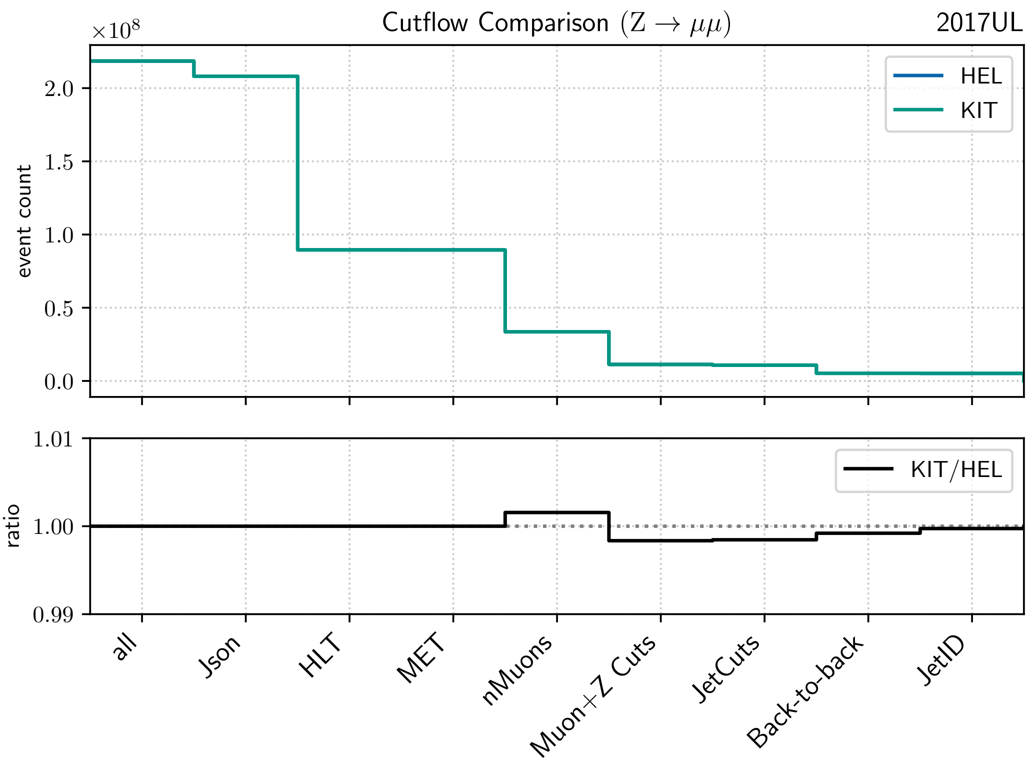


Figure 3.19.: Final cutflow comparison for the 2017UL muon dataset between the old MiniAOD framework (KIT) and the new, NanoAOD-based tool (HEL). The event selection of both frameworks is in great agreement. Details on the cuts are provided in Section A.1.

3.4. Preparations for Run 3: Framework Synchronization

As described in Section 3.1, the derivation of the absolute residual L3Res corrections for the full Run 2 Ultra Legacy data utilized the MiniAOD data tier. To avoid costly reprocessings of the entire datasets, which contain loads of information that are unnecessary for the calibration procedures, the skimming step is employed to reduce the overall size of the data to be processed. This improves the efficiency and sustainability from the computing perspective significantly, since usually reprocessings regularly happen as the calibration is an iterative process and individual aspects may be optimized over time again and again.

The introduction of NanoAOD followed the same philosophy. The often unnecessary verbosity of MiniAOD datasets led to the production of the reduced – and therefore more handy – NanoAOD format [40]. Based on the expected benefits, it was decided that the derivation of the absolute residual corrections should – in line with the other calibration steps in Run 3 – in the future rely on the NanoAOD data tier as well, if feasible. However, a migration from MiniAOD to NanoAOD is non-trivial since the two formats conceptually differ in important aspects. For more details, it is referred to Refs, [39, 40]. As a consequence, it was decided to retire the old, but well-established correction and calibration frameworks [63, 73, 72] for MiniAOD and move on to more modern analysis tools, such as the Columnar Object Framework For Effective Analysis (coffea) [80], to make use of the more efficient processing based on NanoAOD, which is already broadly used in the HEP community.

From the computing and sustainability perspective, this makes absolute sense. However, the implementation of the complex optimizations was based on a lot of in-depth knowledge and experience gathered by the KIT-CMS group since the beginning of Run 1 and is also based on past agreements and requests from the responsible JERC group conveners. Therefore, a framework synchronization process with the Helsinki group (HEL), which took over the responsibility for the absolute residual corrections for Run 3 and beyond based on a new NanoAOD framework, was initiated and conducted in 2022 in close contact with the JERC group. Within this process, the frameworks were compared iteratively[4] based on the full 2017UL datasets (see Section A.1.1) and brought to convergence.

In Fig. 3.19, the cutflow for the entire data selection is depicted, which is in great agreement after the synchronization process. Additional information on the cuts can be found in Section A.1.4 and Section A.1.6. Remaining discrepancies in the intermediate steps are resolved with all cuts applied in the end. The observed deviations were investigated and are mainly attributable to hard cuts applied during the production of the NanoAOD data and the application of the cut-based event selection. Because one significant difference between the MiniAODv2 datasets and the NanoAODv9 datasets, which are directly processed from the former, is a different p_T range. NanoAOD does not cover the full energy range, since typically the area of $p_\mathrm{T} < 15\,\mathrm{GeV}$ is often irrelevant for most of the analyses. For the calibration, however, energies that would even go to the single-digit GeV range would be desirable[5].

While saving valuable CPU performance and storage space, the hard cuts applied throughout the production – together with the selection procedure later on –introduce edge effects, which were revealed when comparing the jets and muons selected by the different frameworks on event-level during the synchronization process. Two practical explanations for the edge effects were identified: First, due to energy cuts applied to the raw energies, it can happen that jets (and therefore whole events) are corrected over the hard limit for MiniAOD, while these jets simply are not contained in the NanoAOD datasets and therefore cannot be corrected over the hard cut. To mitigate these effects, the limit for the analysis was accordingly adapted to the $p_\mathrm{T} = 15\,\mathrm{GeV}$ for the MiniAOD data as well to ensure a consistent result. The second effect can be tracked down to events where the leading and second leading objects (jets or leptons) are very close to each other. In such cases, minimal differences in the decimal places of e.g. the energies when selecting the leading muon or the leading jet, can cause a switching of the order, which in turn leads to differences in the comparison between the frameworks.

Both effects together are causing a relative deviation of around 0.3 % in total, as visible in the intermediate steps in Fig. 3.19. Fortunately, due to the overall very high quality requirements on the selected events, these effects are not introducing a significant difference in the number of events after the full selection applied, as visible in the last bin in Fig. 3.19.

Finally, the comparison of the transverse momentum (Fig. 3.20) and pseudorapidity (Fig. 3.21) distributions for the Z boson and the leading jet show that the synchron-

[4] This required a reprocessing of the data over ten times, proofing the value of the skimming by a significantly reduced processing time per iteration and an according reduction in the required compute performance!

[5] Remark: This still justifies the utilization of MiniAOD for Run 2, where NanoAOD was not yet adapted to the special requirements, which only followed with JMENano later on.

ization process converged well.
In summary, the framework synchronization was eventually successful and the results converged well – indicating a full knowledge transfer. Thanks to the in-depth analysis of the frameworks as well as the data, the overall L3Res correction method was hardened by eliminating minor issues on both sides and the process helped to verify the basic principle of the method. On top, it served as a cross-check between MiniAOD and NanoAOD, revealing edge cases where slight differences can be observed, which was additionally reported to the CMS Cross-Pog (XPOG). The results of the synchronization were presented to the CMS JERC group in [81].

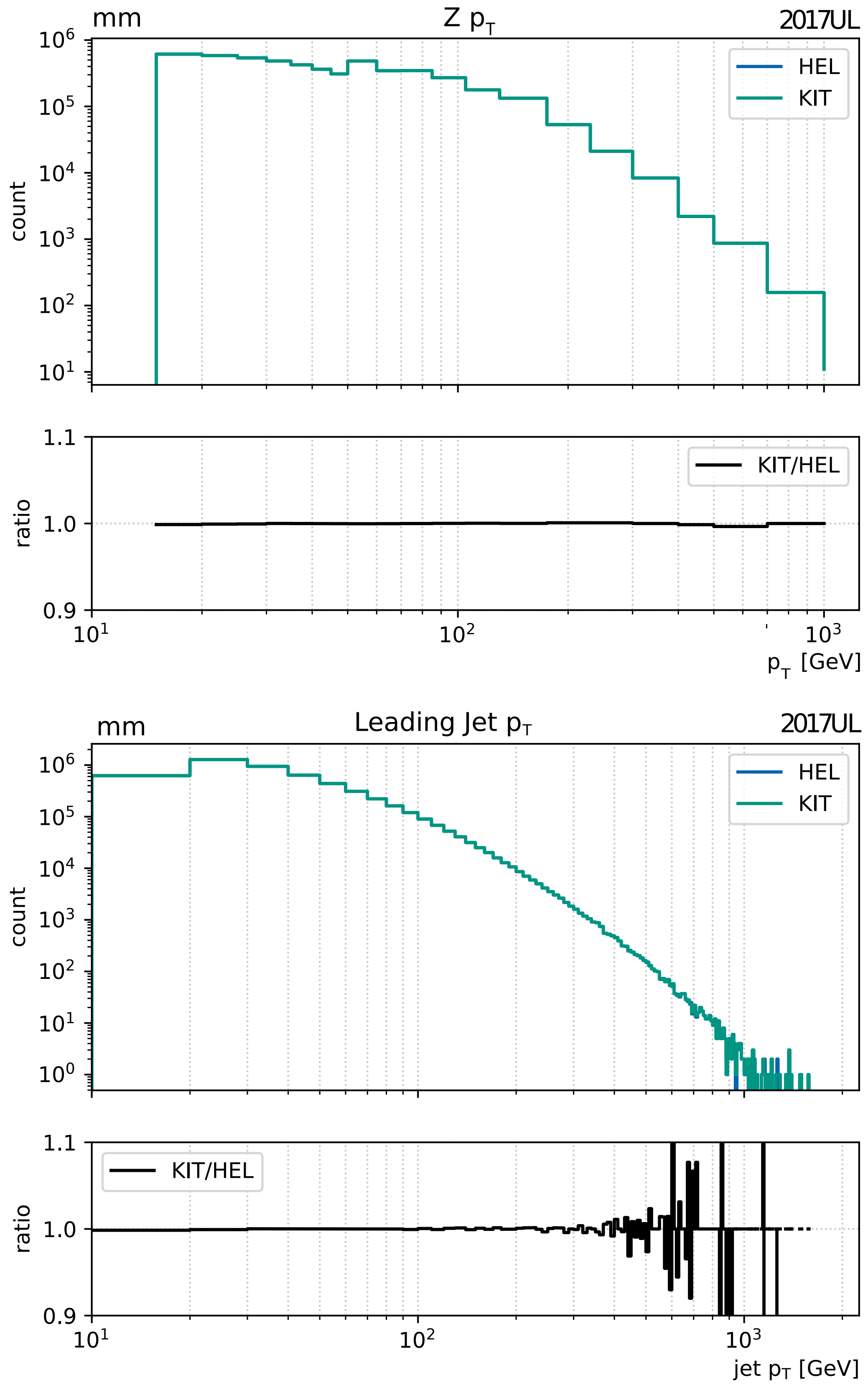


Figure 3.20.: Comparison of the transverse momentum distribution of the selected Z bosons, p_T^Z (top), and the leading jet p_T (bottom). As clearly visible, the data selected by the KIT framework and the HEL framework are in very good agreement. Only for the high p_T range of the leading jet with low event counts, a larger deviation is observed.

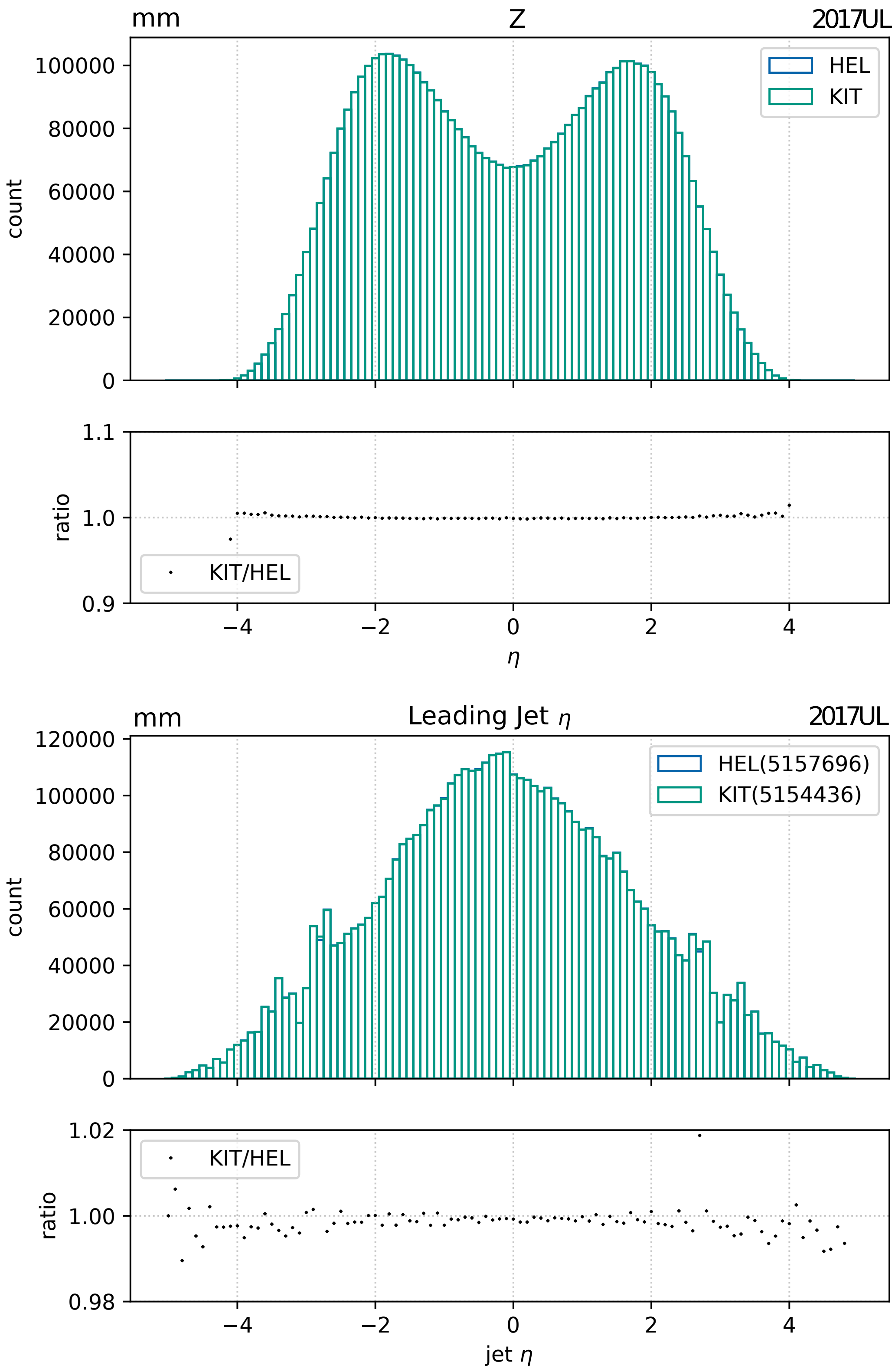


Figure 3.21.: Comparison of the Z (top) and leading jet (bottom) distributions in η between the KIT and HEL data selection. As visible, the synchronization converged. Remaining minimal differences in the leading jet distribution can occur from minor systematics, such as rounding effects in the data production, which e.g. can lead to a switching of the leading and sub-leading jets in an event when they are close. Additionally, slight differences can be caused by the hard hard cut edges which may lead to the rejection of an event that is kept in the other dataset.

3.5. Conclusion

In this chapter, the first derivation of the absolute residual Jet Energy Corrections for Run 2 Ultra Legacy data was presented. The tools and methods were adapted to meet the recommendations and requirements for the latest reprocessing of the CMS data and simulations in close collaboration with the JERC group and the results were handed over to the CMS Collaboration as the final contribution to the JEC from the KIT-CMS group.
The comprehensive analyses of the L3Res corrections in Section 3.3 demonstrated that these corrections significantly improve the alignment of the jet energy scale at highest precision by minimizing the remaining discrepancies between data and simulation. In great parts of the phase space, the agreement in response between data and MC is at sub-percent level, and as part of the global fit of the correction factors, the derived corrections will positively impact a broad variety of high-precision QCD analyses throughout the entire field.
Furthermore, synchronizing the legacy Run 2 frameworks based on MiniAOD with the new Run 3 tools utilizing NanoAOD enabled a complete knowledge transfer. The detailed comparisons revealed several minor issues on both sides, leading to further improvements of the utilized methods, of which in turn also the Run 2 corrections profited from.
In addition, this process not only harmonized the two frameworks in preparation of the future but also served as a cross-check between the two data tiers. Observed differences were reported to the relevant CMS sub-groups. The overall framework synchronization process has therefore additionally provided valuable insights for the future design of dedicated NanoAOD productions aimed at calibration tasks and JetMET studies, with extended energy ranges and enhanced precision.

In summary, the work presented here represents a significant contribution to the long-term effort to achieve the highest precision in QCD studies with the CMS experiment. This work ultimately enables the detection of even the slightest deviations from the current understanding of the SM. Furthermore, the advancements detailed above not only enhance data precision but also support the transition to optimized resource utilization within the JEC process for the CMS experiment. As modern HEP research becomes increasingly complex and heavily dependent on sufficient compute resources, not only is the efficient use of the given resources absolutely essential, but also the exploration of novel computing strategies to cope with the anticipated increase in demand during the HL-LHC era. The second part of this work addresses these challenges, beginning with an overview of today's worldwide computing infrastructure for HEP research in the next section.

4. Computing in High-Energy Physics

> The machine does not isolate man from the great problems of nature but plunges him more deeply into them.
>
> (Antoine de Saint-Exupéry)

4.1. Introduction

Modern High Energy Physics research relies fundamentally on advanced computing technologies and infrastructure. Without sufficient compute and storage resources, experiments like CMS and ATLAS would not be possible at all. Starting from simulation-based detector studies to build modern, high-performant particle detectors and large-scale MC simulation campaigns for particle collisions, over data acquisition, where immense compute power and extremely fast data transfer technologies are required to deal with the – already today – extreme collision rates, everything is heavily relying on computing resources. At this point, the data is available, but in a raw, unprocessed form, which still requires a reconstruction of the physics content as well as processing, e.g. in form of the earlier described Jet Energy Calibration. Finally, the collected data will eventually be used in large-scale end-user analyses all around the world that result in new scientific findings are costly in terms of compute resources. To put it in a nutshell, without (sufficient) computing resources, no scientific progress in the field of modern particle physics would be possible today!

Fortunately, the HEP community is well positioned with the Worldwide LHC Computing Grid. Since the early 2000s, Grid Computing has emerged as the cornerstone of computational efforts in HEP. The grid enabled physicists all over the world to collaboratively access and analyze WLCG irrespective of their geographical location. By this, it has become crucial for the scientific success of HEP. Only through the contribution of globally distributed resources from hundreds of institutions worldwide, the groundbreaking scientific results, like the discovery of the Higgs Boson by the LHC experiments in 2012 [1, 2], were made possible.

Furthermore, Grid Computing has proven to be an extremely reliable and useful concept for the global science community, even beyond the borders of HEP, leading to its adaption in many scientific fields. Even more: It introduced a fruitful collaboration of computing experts across the globe resulting in constant improvements in the related computing technologies. And the flexibility of the WLCG to incorporate and accommodate evolving concepts, e.g. from related fields like Cloud Computing and High-Performance Computing, or other modern technologies like containerization, has always been the guarantor of its success and ensures its continued relevance and adaptability. Sticking to this is extremely important and more in demand today than ever, especially with regard to the requirements on compute and storage for the HL-

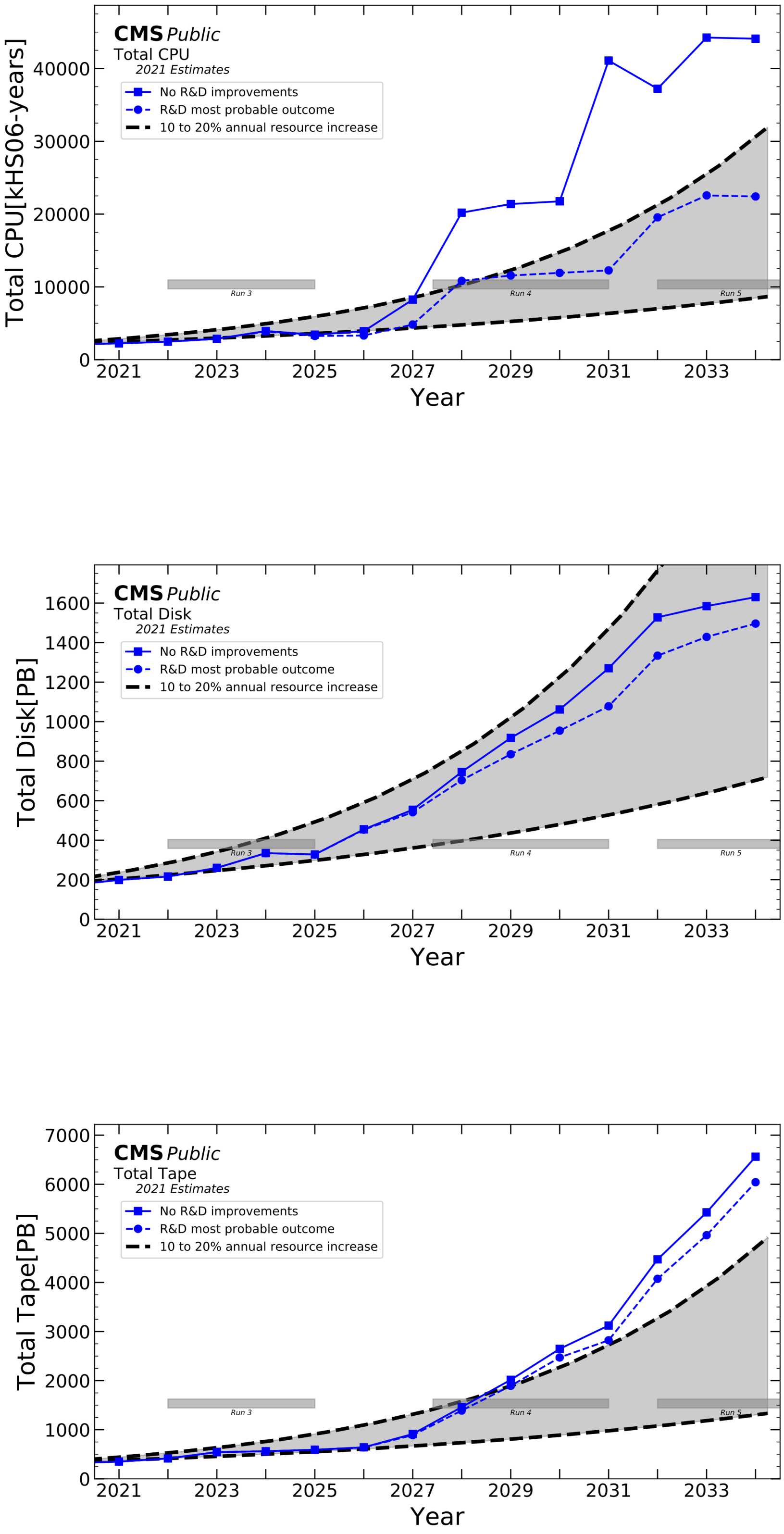
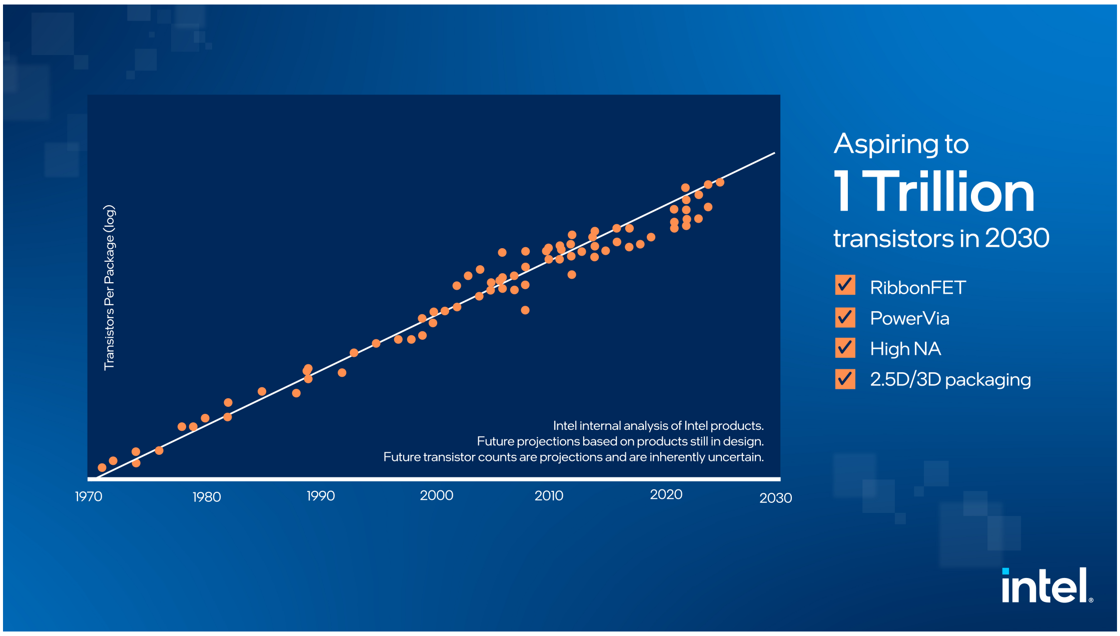


Figure 4.1.: The resource projection for the CMS experiment for CPU, disk, and tape from 2021 [82]. Since the schedule for the HL-LHC already changed, the times are not accurate anymore, but the message still stays valid that only with sufficient R&D effort, delivering sufficient resources in the future will be possible.



Aspiring to

1 Trillion

transistors in 2030

- ☑ RibbonFET
- ☑ PowerVia
- ☑ High NA
- ☑ 2.5D/3D packaging

Figure 4.2.: Moore's Law including a projection in the future by Intel. A nearly linear increase in log-scale, reflecting a doubling of the transistors every two years, as the law proposes, can be observed over decades. Intel is still confident that the pace can be kept up until the 2030. Let's hope that they are right! *Source*: [84]

LHC and beyond. The current planning considers more than 350 PB of raw data per year [83]. And this does not include Monte Carlo simulations and user analysis data. In total, after reconstruction and processing, several exabytes of data are expected over the full runtime of the HL-LHC. This increase will lead to a new era of unprecedented precision measurements, but it also presents unparalleled challenges for HEP computing, as shown with the resource projections with different underlying scenarios for the CMS experiment depicted in Fig. 4.1.

In terms of CPU, the projection is based on a yearly increasing demand by around 20 %. Even if Moore's Law is still on our site[1], as e.g. Intel suggests (see Fig. 4.2), we cannot only rely on him. Firstly, it is expected that during this decade the pace of development is decreasing due to physical limitations and manufacturing challenges, meaning that probably no factor of 2 can be hold for the next decade and optimization will become more important than ever. And secondly, at some point maybe hardware is not even the limiting factor anymore, but for example power consumption and energy availability. Such scenarios are not really reflected in the estimates. Therefore, in addition to the integration of further resources, it must be ensured that the existing ones are used as efficient as possible, also in terms of sustainability and the environmental impact of research in general.

In order to meet the expectations, adapting to the future conditions will be essential for an on-going success of the WLCG as a whole – and with it, the research in HEP. New options, like for example the usage of GPUs and the integration of more efficient HPC resources (on largest scales), are not yet fully considered in the estimates and may become a game-changer on the long run, if investigated further.

[1] Pun intended.

Here, it is emphasized that this does not mean that the HEP community will move away from the previous, very successful Grid Computing model. It is rather being increasingly supplemented and expanded by new concepts. In turn, this also means that with the HL-LHC era in sight, not only the demand for, but also the variety of compute resources to satisfy it are expected to rise strongly, which again increases the complexity of the overall computing environment.
Because of this, ongoing innovation, research, and development in these new fields are essential and became an integral part of High Energy Physics as a driver of success. This statement is clearly supported by the CMS Collaboration, as indicated by the *R&D most probable outcome* (blue, dotted lines) in the projection plots for each resource category in Fig. 4.1, and also reflected in the current WLCG strategy, strongly encouraging innovations [85]. Otherwise, HEP research will fall short of expectations on the long run.

4.2. The Worldwide LHC Computing Grid (WLCG)

After the LHC was approved in 1994, the requirements on the future IT infrastructure were defined [86]. This included:

- Storage and efficient processing: Enormous amounts of data need to be stored and integrity and worldwide accessibility for all collaborators must be ensured.

- Strong focus on redundancy (of data and services) and reliability: The extremely valuable experiment data must be secure at all costs, services need to be available, and resources should be reliably accessible at all times.

- Scalability and cost effectiveness: A computing model is required that allows to adapt to growing resource demands in the future and offers the possibility of integrating all available, often heterogeneous resources to ensure cost effectiveness, which, in view of public funding, is of high importance.

- Sustainability: In addition to financial sustainability, the computing model must also ensure the long-term availability of resources. Alongside, sustainable personnel responsibility is important, and opportunities for knowledge sharing, training, and skill development for scientists and engineers involved should be provided. Furthermore, in recent years, the awareness of the own ecological and socio-economic responsibility has been a driving force behind sustainable development and the responsible use of existing resources to minimize the environmental impacts of research.

- Collaboration and shared contributions: All interested members should be able to contribute and the model should foster international collaboration and the global scientific (computing) community.

These specifications quickly led to the conclusion that the on-premise provisioning of all required resources is not feasible and the concept needs to be decentralized. This assessment was supported by the MONARC project which was initiated to find a suitable infrastructure model for the LHC computing [87]. Based on the findings, the LHC Computing Grid (LCG) [86] was initiated in 2002 to develop a hierarchical

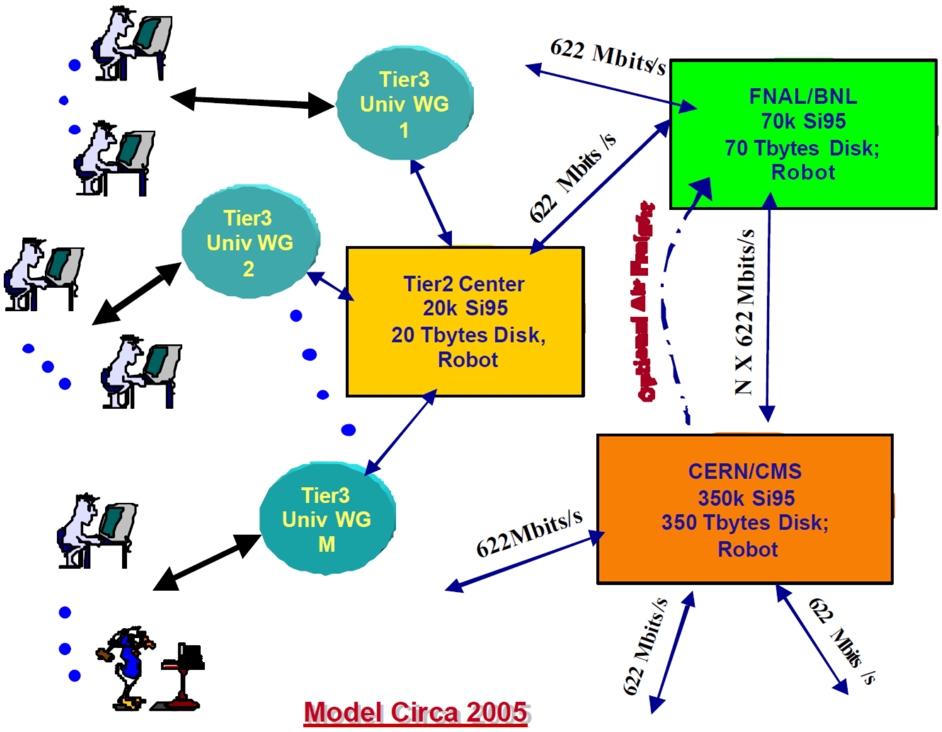


Figure 4.3.: The hierarchical computing infrastructure as proposed by the MONARC report in 2000 for the LHC experiments. *Source*: [87]

infrastructure, often referred to as the MONARC model, shown in Fig. 4.3 from the original report [87].

After the release of the official design report in 2005 with many contributions of diverse institutions and grid initiatives, like e.g Open Science Grid (OSG) and European Grid Infrastructure (EGI) – or better said their predecessors – the Worldwide LHC Computing Grid was officially founded in 2006 as a worldwide computing collaboration for the LHC experiments. The naming transition from LCG to WLCG underlined the worldwide character of the project. It reflects the expanding scope of the grid infrastructure to include computing resources from institutions all around the world, enabling collaboration on a global scale for data processing, storage, and analysis. This is also emphasized by the inclusion of non-LHC experiments, which are no direct members of the collaboration but can benefit from the knowledge and resources as well.

In 2025, the WLCG is world's most sophisticated open source, scientific computing grid and the backbone of HEP research providing more than 1.4 million cores, an average data throughput of around 60 GB/s [88], and around 1.800 PB of tape and 1000 PB of disk storage [89]. Around 20 % of the resources are provided on-site by the CERN data centers, while the rest is contributed by around 170 grid sites in 42 countries [88], forming the structure of the WLCG, as described in the next section.

4.2.1. Structure of the WLCG

As suggested by the MONARC studies, the WLCG was designed in a tiered, hierarchical structure with four different main levels, as shown in Fig. 4.4.

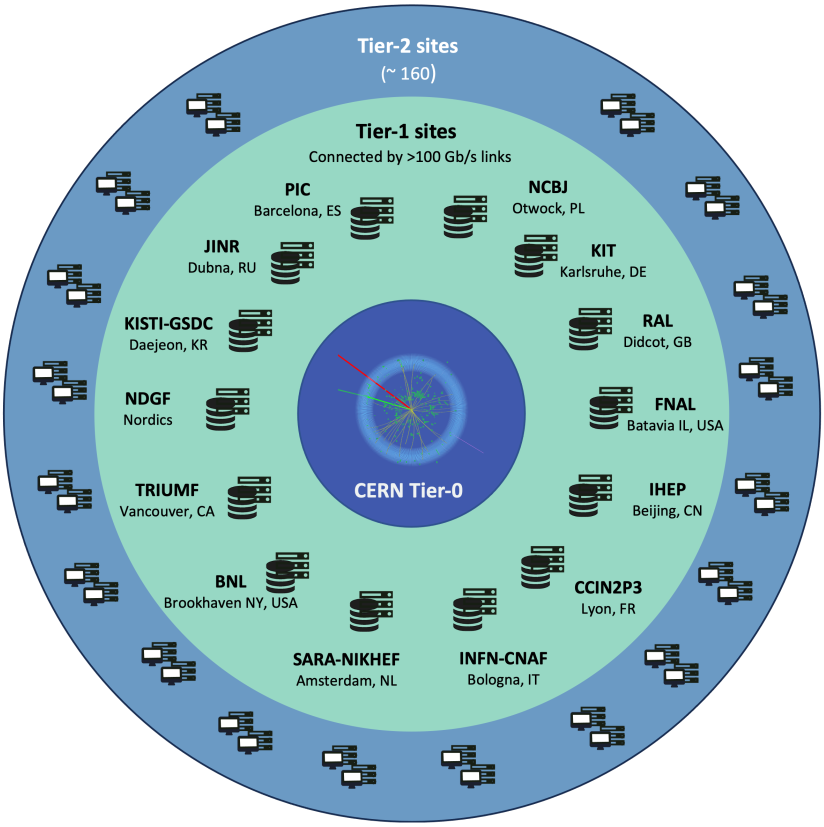


Figure 4.4.: The current WLCG architecture as of December 2024 [90]. Its hierarchical structure is built around the Tier-0 located at CERN and consists of 14 Tier-1 centers and more than 160 Tier-2 sites. Those are supplemented by several additional, divers resources summarized as Tier-3 centers which do not strictly follow a common concept and are therefore not visualized as a category here. *Source*: CERN/[90]
Remark: Every Tier-1 and Tier-2 center has typically an experiment affiliation to one or more experiments, which is not shown here but indicated in Fig. 4.5 for the Tier-1 centers.

Each different level, typically referred to as Tier, has its distinct tasks and responsibilities:

Tier-0 The Tier-0, also called CERN Data Center, is located at CERN and is the central instance the WLCG is build around, as depicted in Fig. 4.4. It provides large-scale long-term storage facilities and roughly 20 % of all compute resources [88]. All data recorded by the LHC experiments reaches the Tier-0, which keeps the first copy of all raw detector measurements and mainly does the initial processing and reconstruction. From there, the raw and reconstructed data is redistributed to the Tier-1 computing centers [90].

Tier-1 The Tier-1 centers, intended as main regional centers in different countries, keep a share of all measured data (second copy) and provide large-scale resources for production and analysis tasks. They are characterized by high network bandwidths and manage the redistribution of data to their local Tier-2 centers. A share of the produced simulation data at those attached centers is in turn again long-term stored at the Tier-1 centers. Additionally, they often provide support to the smaller sites in their vicinity.

Tier-2 Tier-2s are usually located at universities and national research institutions. They mainly provide compute resources for MC productions and user tasks as well as medium-sized (disk) storage capacities for a broad data distribution throughout the grid.

Tier-3 These resources do not follow a strict categorization and can range in principle from single servers over smaller-sized analysis clusters at universities that usually provide resources to local groups, up to large-scale HPC centers and (commercial) cloud providers. Despite there is no formal agreement between the WLCG and most of the Tier-3 resources [90], they contribute significantly to the available resources at an opportunistic level. Sometimes, they are additionally used as grid R&D resources, like e.g. the Throughput Optimized Analysis System (TOpAS) [91] at KIT, for developing and testing new technologies to improve HEP computing.

This structure ensures the goals of scalability, redundancy, and efficiency in storing, accessing, and processing data by distributing the load worldwide. The scalability is given by the open architecture of the concept. A resource provider that wants to contribute to the WLCG can easily be added to the according Tier. And in form of a Tier-3, all kinds of resources can be included without fixed obligations, from a single server to a huge HPC cluster.
The flexibility of the overall concept is reflected in the ability to in principle replace each resource of a certain Tier by another one of the same kind, as the tasks and responsibilities are very similar. One example for that is the decommissioning of the Russian Tier-1 in 2024, which was compensated by the other Tier-1 centers. And also the impact of downtimes, due to updates or problems on the overall operation can be strongly mitigated by this. Here, the WLCG profits from the fact that a computing grid is only loosely coupled without strict dependency, apart from network connectivity. To summarize, the infrastructure is therefore very reliable

and redundant, as there is no single point of failure, not even the Tier-0[2], since all (important) experiment data has backups and replicas or – in the worst case – can be reproduced from the raw data that is always kept save.

And lastly, an efficient operation is supported by the hierarchical, de-central concept. By distributing the data and decentralizing the workload across the Tier-1 centers and their associated Tier-2 centers, the system can handle massive datasets more efficiently and reduce data transfer bottlenecks by exploiting data locality and avoiding too much load on individual sites.

Overall, the structural concept of the WLCG has proven very successful over the years and is, until today, a guarantee for the success of HEP computing, as it remains open to improvements and its flexibility allows to regularly adopt new concepts. More details on the structure and the overall concept can be found in Ref. [92].

4.2.2. Networks

Another important point, without which there would be no grid at all, is networking. In order to form the WLCG as a joint computing infrastructure with the different Tiers, the participating sites are connected via dedicated, private networks. This is important to provide a secure environment for the scientific compute centers and to bypass perimeter firewalls for a more efficient data distribution and access. The hierarchical architecture is here also taken into account by means of two different network levels.

First, LHCOPN [93, 94] connects the Tier-0 and the Tier-1 centers and is used for the redistribution of raw and reconstructed experiment data to ensure data redundancy. Its characteristics are shown in Fig. 4.5. It is a high-speed optical network with a star topology that exceeds in total 2.88 Tbit/s between the Tier-0 and the Tier-1 centers [95] and most of the sites provide already a bandwidth of at least 100 Gbit/s or more. LHCOPN has moved around 658 PB[3] between November 2023 and October 2024 [95], which demonstrates the enormous capacities that the WLCG makes available for HEP computing.

And second, LHCONE [96] providers a dedicated network, parallel to the Internet, for the exclusive[4] communication between the WLCG Tier-1 centers and Tier-2 centers, as well as the Tier-0. It is build as an association of several scientific network service provides, such as GEANT, ESNet, or SURF (31 in total as of 2023 [94]), much like the public Internet, but with the difference that only authorized WLCG sites can connect. This allows sites to automatically trust the traffic coming from LHCONE making expensive firewalls obsolete and allowing very high bandwidths. Therefore, it is ideal for the redistribution of data between the Tier-1 centers and Tier-2 centers and for general data access across all Tiers. The very complex structure of LHCONE is shown in Fig. B.3.

Overall, the dedicated WLCG networks are a reliable foundation for the global HEP community. But for the future, a massive increase of the bandwidth will be necessary to guarantee an efficient operation of the WLCG with the beginning of the HL-LHC.

[2] Apart from data collection periods, which make it more complicated, as the Tier-0 is the first archiving instance from where all data is redistributed.

[3] And nearly 800 PB including LHCONE and the Internet [95].

[4] Remark: The network is not exclusive to the LHC experiments but also for partners of the WLCG, like Belle II, DUNE, or the Pierre Auger Observatory.

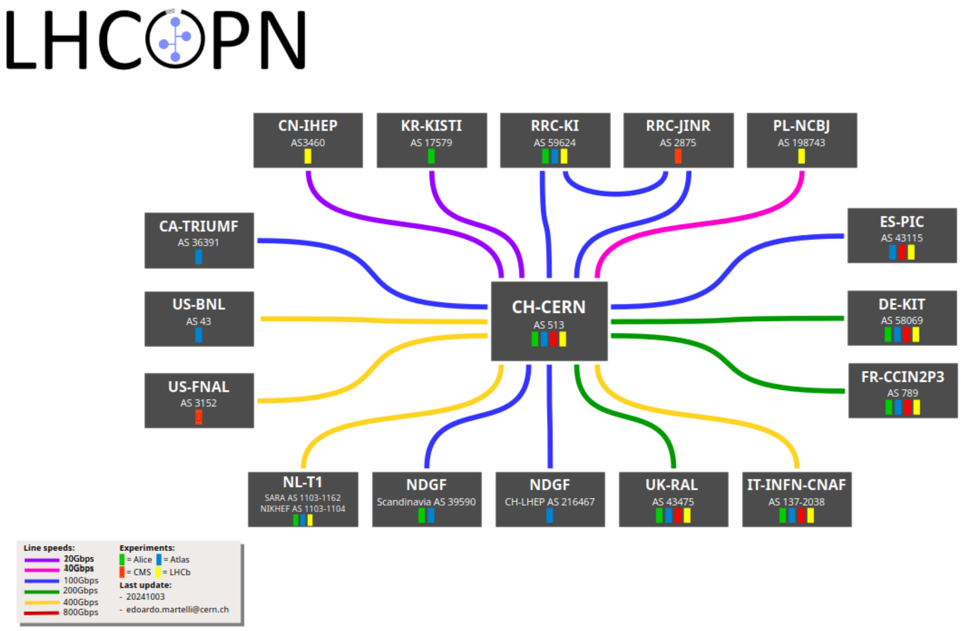


Figure 4.5.: LHCOPN in 2024. It is one of WLCG's private networks that connects the Tier-0 with the worldwide distributed Tier-1 centers and is dedicated to the transport of data between them. An important thing to notice is, indicated by the colored connections, that most of the sites are nowadays already providing a bandwidth of 100 Gbit/s or more. *Source*: [97]

At the moment, at least 4.8 Tbit/s (minimal model) are derived from the expected data rates, calculated as [83]:

$$\left(\sum \text{ATLAS,CMS,ALICE,LHCb}\right) \times 2^{(\text{for bursts})} \times 2^{(\text{safety-margin})} , \tag{4.1}$$

with an estimate of 1 Tbit/s for ATLAS and CMS during data-taking. Furthermore, a flexible model is considered that takes possible reprocessing and re-reconstruction of the collected data during data-taking into account by adding another factor of two (9.6 Tbit/s). The resulting requirements on site-level are 1 Tbit/s to the Tier-0 (LHCOPN) and 1 Tbit/s to the Tier-2 centers (LHCONE, aggregated) for the major Tier-1 centers. Each of the Tier-2 centers should be connected to LHCONE with at least 400 Gbit/s [83].

Even though the challenge to meet the requirements is enormous, the Data Challenges (DC) 2024 have shown that the WLCG is on the right track and the networks were not a bottleneck [98]. During the DC, a peak data transfer rate of around 260 GB/s (2.08 Tbit/s) was observed. The minimal target of 1.2 Tbit/s was nearly always reached and the flexible target of 2.4 Tbit/s was shown to be in range, observable in Fig. B.4.

With new technologies currently being explored, it is therefore very likely that the WLCG will be able to provide the required bandwidths, as demonstrated by the first 800G (2x400G) link between CERN and NIKHEF, the Dutch Tier-1, realized together with SURF and Nokia [83, 99].

4.2.3. Resource Provisioning and Batch Systems in HEP

One of the main purposes of the WLCG is the provisioning of compute resources to the experiments and affiliated researchers. With the HL-LHC, the anticipated data rates are expected to result in extraordinarily large and rising computational demands – in scale and complexity. Therefore, as shown in Fig. 4.1, alongside the research and development for the improved usage of available resources, the WLCG needs to be able to incorporate all potential resources of different types distributed across numerous geographic and administrative domains in the best possible way to meet the future requirements.

To make this possible, the HEP community employs a variety of resource management tools and batch systems to ensure efficient resource integration, provisioning, job scheduling, and workload management for all kinds of resources. These solutions, while inherently complex, offer a high degree of flexibility and adaptability to evolving academic and scientific needs.

In the following, the most relevant frameworks for the integration of resources in the context of this work focusing on CMS, namely HTCondor and Slurm as well as additional cloud-based approaches are described and evaluated with regard to their use in HEP for resource provisioning and batch scheduling.

4.2.3.1. Classic Resource Provisioning with HTCondor

HTCondor [100, 101] is an open-source, High-Throughput Computing (HTC) software framework for distributed resource integration and widely used within the WLCG, especially as basis for the CMS workload management. It is mainly developed at the University of Wisconsin-Madison, but has many users and supporters all around the world. Its primary intention was to harness unused compute resources of their local computers by pooling them together and providing a system to better utilize them [102]. From this, the concept eventually evolved to become a cornerstone in forming large-scale distributed computing infrastructures, like the WLCG. The system is designed to schedule and manage large numbers of compute jobs across flexibly pooled, heterogeneous, distributed compute resources, forming a so-called batch system.

For the scheduling of jobs in such a system, a match-making mechanism is provided that matches jobs based on user-defined constraints, like required memory, number of CPU cores, or specific hardware needs, e.g. GPUs, with available resources in the pool in predefined cycles. The goal of the match-making is to dynamically and efficiently allocate resources to jobs, so that each compute task is are assigned to the most suitable machines based on its resource requirements and the currently available capabilities of the resource pool. This is ensured by a so-called negotiator that guarantees that jobs are distributed fairly taking the current load of machines, job priorities, and any other predefined factors that may influence the scheduling process into account.

Within the WLCG, this match-making is rather secondary, at least for production workflows, as they always run in standardized pilot[5] jobs with predefined resource demands. Primarily, the system is therefore used for the integration of the resources

[5] Pilot jobs are placeholder jobs submitted by the experiments to all grid sites that fetch and execute actual payload jobs.

Table 4.1.: Overview of the scheduling efficiency for the CMS global job pool at selected Tier-1 centers from 29.11.2024 to 06.12.2024. As can be seen, an allocation efficiency of nearly 95 % on average can be achieved with HTCondor for CMS.
Source: MONIT/CMS monitoring (Grafana)

Site	Efficiency	Average Idle CPUs	Average Total CPUs
T1_US_FNAL	0.966	2136	62613
T1_UK_RAL	0.863	2243	16397
T1_RU_JINR	0.959	611	15024
T1_IT_CNAF	0.946	427	7952
T1_FR_CCIN2P3	0.965	289	8295
T1_ES_PIC	0.959	131	3197
T1_DE_KIT	0.981	222	11922
Mean	**0.949**		

and taking over the scheduling and pilot management. More detailed information on the underlying concepts can be found in the official documentation [103].

A major advantage of HTCondor in the context of loosely-coupled grid environments is the flexibility of this way of resource provisioning and matching. It can dynamically adapt to changing resources as machines join or leave the pool, ensuring that jobs are always matched with the most suitable resources. As an example, when a resource becomes available, it simply promotes its job slots to the pool. It then can be matched and starts executing jobs. If the resource becomes unavailable again at some point, it will simply not be considered anymore in the next cycle of match-making. This allows dynamic clusters and supports opportunistic scheduling by making use of resources as they become available.

A positive side-effect of the flexibility is the increased scheduling efficiency, as the system focuses on an effective allocation of resources to minimize idle times and maximizes the job throughput. Therefore, HTCondor obviously delivers best results for smaller jobs in terms of resource needs. Matching three 100-core jobs on two machines with 128 cores, for example, is way more complex and less efficient than matching 75 4-core jobs, even if the runtimes differ, since the scheduling and resource allocation is way more flexible.

A real-world demonstration of the high scheduling efficiency is shown exemplary for the CMS experiment in Table 4.1.

The downside of this dynamic, unplanned scheduling is that jobs with higher resource demands are put at a conceptual disadvantage, as the probability to get matching slots without a preemption mechanism is rather small, when there is enough job pressure – meaning enough work in the system to achieve a full allocation. This, however, is not too relevant in HEP production environments like the WLCG with the default, standardized pilot job scheduling.

To summarize, HTCondor mainly addresses the needs of distributed HEP workflows that do not require tight coupling between tasks and focuses on a maximization of throughput and minimization of idle times, rather then high performance and parallelization.

In opposition, HPC-oriented batch systems, like Slurm (see below), are designed for maximizing computational performance and efficiency when providing resources

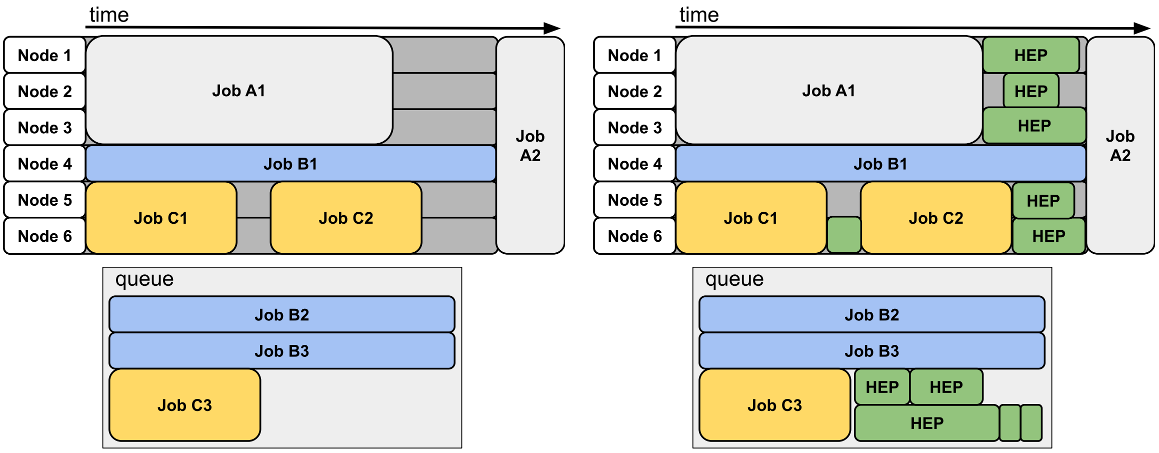


Figure 4.6.: Visualization of Slurm scheduling. In the fixed scheduling procedure, gaps can occur because of multiple reasons leading to idle times, indicated in dark gray. On the right, it is demonstrated how HEP jobs could back-fill the gaps leading to a better cluster allocation and therefore less loss of CPU time.

for tasks that require intense processing power and low-latency communication, such as complex simulations and large-scale numerical computations, which may also become more and more relevant in the future of HEP computing. The special features, particularly in relation to HEP, are discussed in the next section.

4.2.3.2. Provisioning of HPC Resources with Slurm

While the development of HTCondor is strongly driven by the particle physics community in the WLCG context, the Simple Linux Utility for Resource Management (Slurm) open-source tool is more often used for resource management and scheduling at HPC centers. It focuses more on the demands of HPC workloads than on HTC. Slurm is particularly designed for massively parallel high-performance workloads, including GPU-based calculations, which incorporate multiple compute nodes for the same task and may require low-latency communication between them. This is also reflected in the fact that – not necessarily but typically – whole nodes are provided instead of a direct matching of slots to the jobs requirements, like in HTCondor. From the perspective of typical HEP workflows, e.g. production jobs from the global CMS job pool, this is not ideal, as additional mechanisms for the scheduling of the default job pilots are necessary when they do not match the overall allocation. Nevertheless, HPC and HEP can form a great symbiosis, particularly due to the inherent differences in their typical workloads, which will be explained in more detail later.

The other big differences to HTCondor are rather static resource pools and a fixed, planned scheduling. This makes it highly scalable and ideal for large, parallel compute jobs but overall more rigid. Slurm does not dynamically and automatically adapt on short time scales to rapidly changing workloads with different resource demands or changes in the infrastructure. A consequence of this is the comparably lower scheduling efficiency.

On the left side of Fig. 4.6, some reasons are visualized. The most trivial cause is the preemption for large jobs. For example, if e.g. all 6 nodes are requested by a high-priority job A2, they need to be free at the designated starting time. The nodes that have finished their work then have to wait until the last one is ready to execute the parallel job, causing idle times at the end of the scheduling block, as indicated in dark gray. The second source for inefficiencies depicted are jobs that terminate early or unexpectedly, also causing gaps in the schedule. In Fig. 4.6, they are represented by the jobs C1 and C2. These gaps are often too small – mainly in terms of walltime constraints rather than compute resources – to accommodate other large-scale computations in the queue. For dealing with these inefficiencies, two options are provided by Slurm: rescheduling or back-filling.

With its static concept, however, Slurm is not designed for frequent rescheduling the entire plan as this would not scale. In general, it happens but is undesirable, particularly for smaller, low-priority users. Because rescheduling may cause large, higher-priority jobs to be preponed, which would lead to the opposite behavior than with HTCondor, strongly preferring large resource demands[6]. Therefore, in multi-user HPC environments, ensuring fair scheduling while maintaining high utilization is complex, especially in the scientific context, where the computational demands by different communities may differ strongly.

The other option to deal with a reduced scheduling efficiency resulting in under-utilization of resources is a back-filling of the gaps in schedule while keeping the original timing. By default, Slurm provides a mechanism to optimize resource utilization by running smaller jobs in the gaps left between larger jobs. But this requires workloads that fit into the gaps and an appropriate priority handling[7], which must allow a bypassing of the official job queue to potentially fill the gaps – which can be rather complicated to realize. In addition, typical HPC jobs are often long-running and may require proper preparation, like software provisioning or data prefetching, and therefore not suited for a proper short-term back-filling. This is depicted by the queue on the left side of Fig. 4.6, which indicates that there are no feasible HPC jobs available for filling the gaps. And that is exactly where the flexibility of HEP workloads may be beneficial and provide a good starting point for a joint venture of HPC providers with the HEP community.

In contrast to huge, massively parallel calculations, like e.g. climate simulations, which require immense computing capabilities, the simulation of a single particle collision in high-energy physics is fairly simple – at least in terms of resource demand. And since they are independent of each other, the processing or simulation trivially parallelizable. This means, that typical HEP (production) jobs could in principle be configured to simulate or process a single collision event and therefore be deployed on even half of a core, or a billion of events filling whole clusters. From that results a flexible, horizontal scalability in time and resources.

In reality, this flexibility is, mainly for operational reasons, not exploited to such an extreme extend as described, since a fixed pilot size and rather constant walltimes are desirable for a more constant job throughput and standardization. But conceptually, this is a major advantage and makes HEP jobs ideal to fill the gaps and could help regaining idle resources in usual HPC centers, as exemplified on the right of Fig. 4.6.

So to summarize, HPC with Slurm and HEP have distinct focuses and are not an

[6] Ultimately, this is a matter of policies and can be done properly also with regards to smaller users. But it is complicated and may even require a regular intervention of administrators.

[7] Here, also contingencies have to be considered. Eventually, it is again a matter of policies whether or not a job of a user above its contingent, e.g. with lowest priority, may start anyway to realize a better cluster allocation.

obvious match at first glance, but they can complement each other to the benefit of both sides. Especially for the future of HEP computing for the HL-LHC and beyond, this symbiosis has a significant potential. It is definitely worth exploring and intensifying further, as it complements the default HTCondor use-case within the WLCG and may extend the possibilities of HEP computing in the future.

4.2.3.3. Cloud Environments

Alongside the utilization of HPC resources, there are also projects that explore models for incorporating Cloud Computing infrastructures and concepts with Grid Computing resources. TARDIS, described in detail in Section 4.5.2, or HEPCloud, developed at Fermilab (see, e.g. Ref. [104]), demonstrate how clouds can supplement traditional resources. Additionally, HEP computing has also seen an increasing adoption of Cloud Computing technologies over the last years. Namely, a transition away from bare metal resources and classic virtualization to modern concepts like full containerization or OpenStack. As described in more detail in the next part (Section 4.2.4), this simplifies the deployment of experiment software stacks already today and provides a consistent environment across diverse infrastructures ensuring operability and reproducibility. And also modern orchestration technologies like Kubernetes are complementing traditional grid- and cluster-based infrastructures within the WLCG, e.g. with OpenShift at the Tier-0 at CERN.

But inherently, the overall Cloud Computing concept again does not match the typical HEP use-case. Here, the focus lies on an elastic scalability instead of fixed, on-premises resources. While scalability is important for HEP as well, the elasticity is rather unnecessary, as the compute concept of the WLCG does not foresee a timely dependence of the resource demands as it is, e.g., the case for popular web services, like Netflix, or else. Fluctuating workload patterns, though common in HEP data processing, are typically covered by the sharing of resources between different Virtual Organization (VO)s instead of an elastic provisioning.

The on-demand character after a pay-as-you-go principle of commercial cloud resources is in principle also interesting but not really feasible for HEP computing today. In theory, the ability to automatically demand and provision additional resources, like virtual machines, containers, or even entire cloud clusters during the processing of urgent campaigns or general demand peaks is highly desirable. But despite the concept of elasticity is intended for ensuring cost-effectiveness by paying only for what is used, the prices of commercial providers are usually too high and the concept is not financially sustainable, particularly with the enormous amount of data that has to be transferred and stored.

Nevertheless, in the future, cloud resources could become interesting for the WLCG again as an on-demand extension for a limited time, e.g. during peak data-taking periods, when the available resources are not sufficient on short scales. But this is still a long way off. What is certain, however, is that the WLCG must avoid a vendor lock-in at all costs, as it would terminate its flexibility.

Looking ahead, many of the cloud concepts may also be of interest, specifically in terms of sustainability. The fast, dynamic provisioning of cloud-based resources that are scaled up in relation to the amount of renewable energy in the current energy mix would for example reduce the negative impact of scientific computing on climate change, just to name one possible application.

As a summary, Cloud Computing in HEP represents again a complementary resource model that, like HPC clusters, enables flexible and scalable computing solutions alongside the classic grid concept. Already today, many technologies and synergies of the concepts are exploited and may provide new opportunities to increase the sustainability of the computing infrastructure in the future. And since the LHC experiments expectedly continue to grow in complexity and data volume with the HL-LHC and beyond, an on-demand integration of additional cloud resources could help to cover demand peaks and ensure the functionality of the WLCG.

4.2.4. Software Provisioning in the WLCG

A critical aspect of operating a worldwide computing grid is the reliable and efficient provisioning of standardized software environments to hundreds of geographically dispersed compute sites with diverse hardware, operating systems, and software configurations. Also in terms of reproducibility, mechanisms that ensure uniformity in the deployed software across the entire infrastructure are required to avoid inconsistencies and compatibility issues. This is why the standardized grid environment provided by the WLCG has always been foundational to the successful utilization of distributed grid resources and has greatly contributed to its continued success.
The researchers at the LHC were aware of this even at the beginning of the LCG. But making the software available at this time was complicated and heavily relied on manual deployment and site-specific installations - with the associated effort. The following short excerpt of the Technical Design Report for the CMS Computing Project [20] gives an impression on the complexity and hurdles:

> *CMS software and externals used by it are distributed in the form of RPM packages [...] CMS requires to be able to directly verify that the advertised software is installed correctly, by checking for end libraries and programs mentioned in a package manifest and by verifying file checksums. CMS also requires to check the system software configuration in a similar fashion.*
>
> *The software is installed at each site under a single root location in a hierarchy defined by CMS. [...] The location must be writable by the CMS software installation managers. The area must be reserved for CMS software and should not include software from the underlying system nor other experiments or projects. [...]*
>
> *User code is supplied as a prebuilt custom code based on preinstalled public CMS software, and is delivered directly in the job sandbox. [...]*
>
> *The RPMs are provided through a single central authoritative software repository.[...]*
>
> *A site must provide for automatic installation of software by some combination of submitting Grid jobs and deploying an automatic software installation service appropriate to the Grid involved. [..]*
>
> *Information about installed software should be advertised in the Grid information system, for use by the workload management for job matching. A site can remove software, provided it also removes the corresponding entries from the information*

> *system. The CMS software managers have the ability to remove software by submitting a Grid job to the site.*
>
> – The Computing Project. Technical Design Report, The CMS Collaboration. LHCC 20005-023. 2005. [20, pp. 60–61]

Today, a provisioning like this would not be possible anymore due to the vast variety of different software tools and versions that were developed and validated for different stages of the LHC experiments and are still used in parallel. Also in terms of storage, providing replicas of everything everywhere just became too inefficient. Gladly, with continued technological and software advancement, more sophisticated mechanisms have been developed over time with the objective of significantly streamlining the deployment process for dynamically changing environments. These mechanisms allow users and production applications nowadays to access the required, standardized and validated software environments with minimal overhead while maintaining a high degree of flexibility for deployment and updates, as they are nearly independent of the host systems.

The following subsections introduce the two fundamental concepts for software provisioning in the WLCG: the adoption of containerization technologies for encapsulating and running standardized software environments independently of the host systems and the use of the CernVM File System (CVMFS) for centralized software management and distribution.

4.2.4.1. Containerization

Containerization has become a crucial part of WLCG for ensuring consistent software environments across the heterogeneous resources in the grid. It has revolutionized interoperability and portability of software through isolation and self-containment of dependencies and reproducibility by providing encapsulated runtime environments independent of the diversity of underlying host systems.

The concept itself is nothing new and exists essentially since the 1970s, where the `chroot(2)` system call was introduced for changing the apparent `root` directory for a running process. With this, it provided the first form of filesystem isolation as it essentially restricted a process and its children to a certain directory tree [105]. The limited security mechanisms then led to the development of more sophisticated isolation methods, such as Linux `namespaces` for process isolation [106], `cgroups` [107] for resource allocation and limitation, and `union` filesystems that allow a combination of different filesystem layers essential for writing in isolated directories [108], which form the basis of modern containerization technologies.

Latest with Docker [110], they became the de facto standard for application virtualization, when no absolute isolation is required, supplementing the classic VMs with their hypervisors. The advantage of the containers over these technologies, especially in terms of scalability, is that they are comparably lightweight and easy to deploy, as they share the kernel with the host system and run as simple applications, instead of a full virtualization – without major safety compromises. The two different concepts are visualized in Fig. 4.7. With modern orchestration tools, like e.g. Kubernetes [111] or OpenShift [112], it also became more convenient to deploy and maintain complex services and infrastructures, and are now widely used from big web- and cloud services over HPC to the WLCG, as mentioned in Section 4.2.3.3.

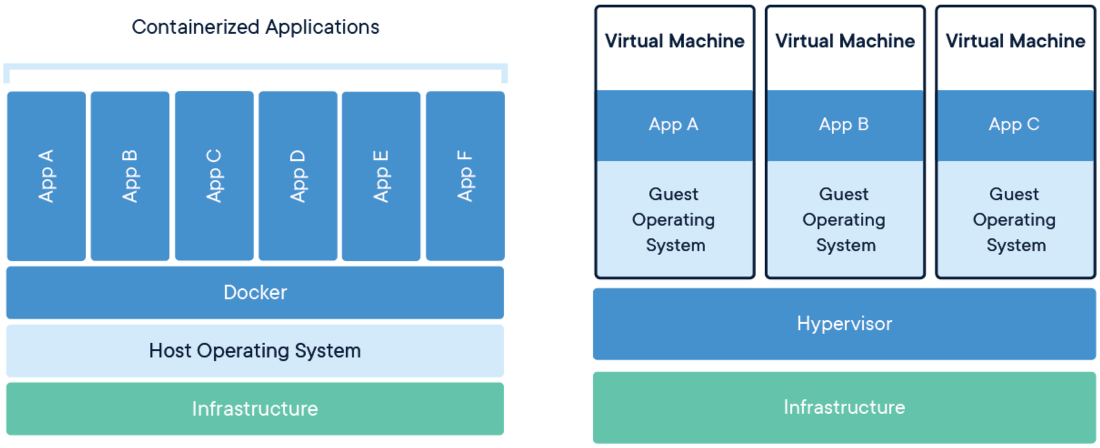


Figure 4.7.: Containers (left) compared to classic virtualization (right). The main difference is that containers share the kernel with the host system and provide a strict isolation based on Linux namespaces and cgroups, while classic virtualization utilizes a so-called hypervisor that entirely mimics the underlying hardware and allows a full simulation of a guest OS, including the kernel. On the one hand, this provides a complete separation between Virtual Machine (VM)s, but on the other, it creates an additional overhead that – if not explicitly necessary – can be avoided with using containers for a more efficient resource utilization. In general, their use-cases differ in that sense that containers are meant to execute a single task while VMs run fully operational, virtualized systems that are used like normal, physical computers. *Source:* [109]

For HEP computing, containers are mainly interesting because of two features: First, containerization allows the shipping of pre-build container images to everywhere and provide a standardized, validated environment, with all necessary software dependencies so that workloads run consistently, independent of the host system. This is crucial for the integration of various, heterogeneous resources and simplifies the distribution of software a lot. Second, they make isolated environments available which allow the execution of the HEP software without further permissions while minimizing interference between parallel workloads (and users), which is essential for optimizing the utilization of shared, distributed computing resources. This point is also very interesting from the site perspective. Because the classic case, especially in HPC, was that users request software and the responsible site personal prepares the environments and installs the software et cetera. With containers, users/communities are able to choose and change their environment themselves which essentially transitions the maintenance effort mainly to the users, as the site solely provides the resources. But the users, in turn, gain a lot in flexibility, as they can use them as desired, even without `root` access to the resources. Therefore, both sides benefit strongly from containerization.

Nowadays, with regard to multi-user, distributed systems and HPC, containerization is usually realized with Apptainer (formerly Singularity) [113], as it has some operational advantages over Docker. The latter offers a comprehensive solution in terms of security and isolation, but this can only be implemented with `root` rights in some cases, which is why the container management with Docker is done with a daemon that requires `root`. For the grid and HPC use-case, this is not ideal as the communication between the Docker daemon and its clients happens over a privileged Unix socket, which effectively provides `root`-equivalent permissions on the host for anyone with access to the socket and therefore introduces a higher security risk in environments where multiple users share resources. Apptainer, in contrast, is reducing the risks by not running a daemon[8] with `root` permissions and allowing users to start containers as themselves, fully operating in `userspace` and preventing privilege escalation by design.

To summarize, it is no exaggeration to say that containerization has revolutionized the Internet as well as the WLCG. It has greatly simplified the integration of heterogeneous resources and set new standards for interoperability and the provisioning of standardized environments as the basis for scientific computing and research. Additionally, with their low overhead and high flexibility, containers help to use the given resources as efficient as possible.

4.2.4.2. The CernVM File System (CVMFS)

While containers are ideal to provide standardized environments and interoperability, shipping all software in containers is possible, but neither recommended nor intended. For the actual distribution of software to the grid sites and beyond, CernVM File System [114, 115] was developed. It is a virtual POSIX read-only filesystem that is fully operational in user space (as a FUSE modul). Initially, it was developed for the distribution of software and condition databases for the HEP community, but today, it is widely used also in other collaborations. Alongside the repositories for the LHC experiments, also others, like e.g. Belle II or IceCube, are using it for shipping their software stacks.

Conceptually, the system consists of a so-called Stratum 0 repository server and several Stratum 1 replica servers, as depicted in Fig. 4.8. The Stratum 0 is storing the data and is the only read/write part of the system. To provide updates and new data, CVMFS uses a writable scratch area on the Stratum 0 that is regularly synchronized with the read-only CVMFS filesystem and then published [116]. The Stratum 1s are standard web servers and public read-only mirrors of the Stratum 0 exporting the filesystem via HTTP. Additionally, local squid proxies [117] supplement the system with caching mechanism, increasing the efficiency and reducing the load on the Stratum 1s.

With this hierarchical structure, CVMFS is providing a scalable way to efficiently deliver software stacks across many disperse sites, reducing the need of an actual replication of the manifold software frameworks and versions while ensuring on-demand access to them at all times. The design as a redundant content delivery network with multiple Stratum 1s additionally ensures reliability and a distribution of the load [116].

Furthermore, with CVMFS `unpacked`, a centrally managed, innovative way of shipping containers is directly integrated with CVMFS. As the name indicates, the images are provided in an unpacked manner by integrating the extracted layers and root filesystems from a registry directly into a CVMFS repository [116]. This allows a direct execution of a container, for example with Apptainer, without the need of unpacking the image layers and preparing the filesystem, as usual. The consequence is a reduced startup time and no duplication of containers, which can be significant for

[8] Instead, it is simply running as an application in user space – therefore the name.

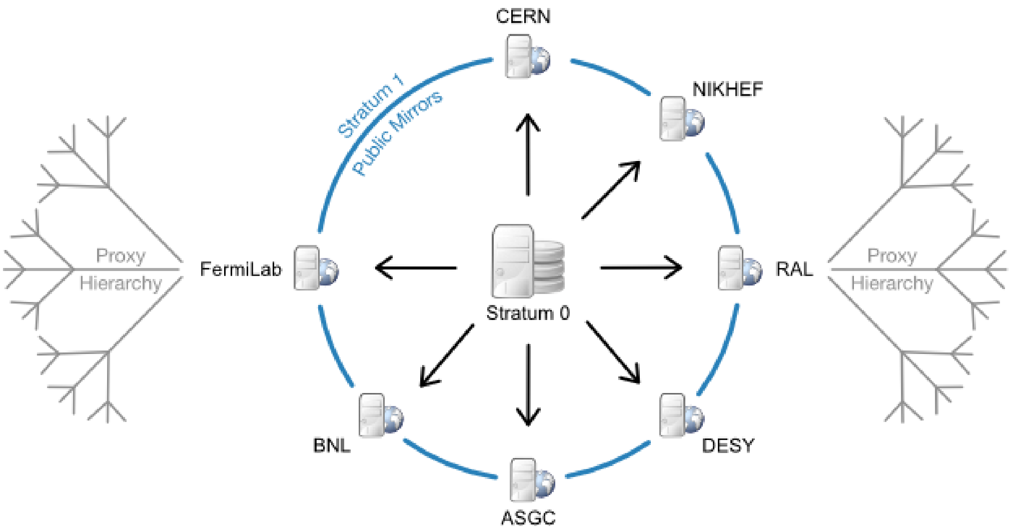


Figure 4.8.: The concept of CVMFS as a content delivery network. The main repository server (Stratum 0) is synchronized with several public mirrors, the Stratum 1 replica servers. They export the data repositories via HTTP as read-only filesystems, accessible for clients under /cvmfs. The network can be supplemented with local (squid) caching proxies, as indicated with the proxy hierarchy attached to FNAL and RAL. *Taken from:* [116]

a cluster with hundreds of nodes. It also reduces the fetched data, as the unpacked images can be cached as well as the other (software) repositories.

The advantages for sites and users using CVMFS compared to the classic way of distributing and installing software in form of actual replicas everywhere, can be summarized as follows: First, it has a minimal overhead, is easy to deploy[9], and reduces the maintenance effort for providing standardized software. Additionally, the mounting of the repositories (via `auto-fs`) can run completely in user space and does not require any administrative intervention. Second, the universal namespace with `/cvmfs` is simplifying the distributed usage and ensures interoperability of the software with consistent paths, avoiding complicated configurations and soft linking. With the way CVMFS is working, it also implicitly provides a version-control and ensures that everywhere the same version is used. Third, the provisioning of a certain software stack works fully on-demand, making it way more efficient by downloading only what is necessary, especially with exploiting local caches. And last, the clients fully relying on out-going HTTP connections avoids blocking by firewalls.

CVMFS together with containerization technologies therefore builds the foundation for interoperability and reproducibility within the WLCG.

Remark: Though it is one of the requirements of official grid sites, CVMFS is uncommon and often unavailable on HPC resources. For this use-case, `cvmfsexec` [118] was introduced which allows mounting CVMFS without an installation. It works either based on `fusermount`, in unprivileged fuse mount `namespace`, or as an Apptainer container.

[9] Nowadays, even fully containerized as deployed at the Institute for Experimental Particle Physics at KIT.

4.2.5. Storage and Data Management

For every scientific community, its data is the most valuable good. The measurements and simulations are crucial for every step in the scientific process. Consequently, for successful and efficient research, the data must be accessible and a proper data management is of most importance to keep the overview in the exabyte era of the LHC. And the loss for the researchers and science in total is immeasurable, if measured data gets lost. From this, two major principles for data handling within the WLCG can be formulated: Data must be *safe* and *accessible*.

To guarantee this, the WLCG provides a multi-facet data management and organization concept incorporating different storage technologies and a comprehensive software toolset for data storage, replication, and access for all LHC experiments. The foundation here is, as described in Section 4.2.1, a hierarchical structure with distinct tasks and responsibilities per level. In terms of storage, they can be divided in two main responsibilities, namely long-term storage for archiving data, and direct-access storage for the everyday usage. For archiving, mainly tape storage systems are used. Modern automatic tape systems[10] combine a decent performance with a high capacity and are unprecedented in terms of TB per € making the archiving of exabytes of data affordable.

The availability of such long-term storage is given at the CERN Data Center (Tier-0) and the Tier-1 centers and is one of the main differences between the Tier-1 centers and the Tier-2 centers[11] Further details on the CERN Tape Archives are provided here: [120]. Safety of the collected data is guaranteed by a multi-copy principle: The Tier-0 and all the Tier-1 centers together hold one complete copy of all raw data each. The most important, processed datasets that form the basis for the main physics datasets, are also often replicated to provide a backup at different sites, as they are costly to produce and crucial to be available for production and eventually analysis purposes. By this, if one site is not available or even suffers a data loss, another one can step in as data source, resulting in great availability and redundancy. But to realize this, the available storage needs to be huge. An overview of the tape pledges in 2024 is provided on the right side of Fig. 4.9.

For datasets that need to be directly available on a daily basis, typically usual disk storage clusters are employed. Here, Tier-2s contribute significantly as well. In total, the WLCG is providing more than an exabyte of disk space (as of 2024, see Fig. 4.9, left), making the required data directly accessible for scientists all around the world. And here, only officially pledged space is shown which is used for official data. This does not include additional space that is, e.g., provided by the universities for their research communities. At GridKa, for example, 3 PB of grid storage are available to be used by all CMS members in Germany for their analyses. The actual amount of data is therefore not known at all but significantly bigger than what the pledges reflect.

What makes things even more difficult is that in addition to the enormous amount of data, the data itself varies greatly. In general, all experiments somewhat distinguish between raw data, processed data, and user level data in decreasing size, and simulations that may have the same hierarchy. Additionally, there are unique data

[10] E.g., as deployed at CERN: LTO-9 from IBM with up to 45 TB (compressed) per cartridge and a transfer rate of up to 400 MB/s without compression [119].

[11] Note: This strict distinction is weakened today, as there are also T2s, like MIT, that provide tape, but it applies in general.

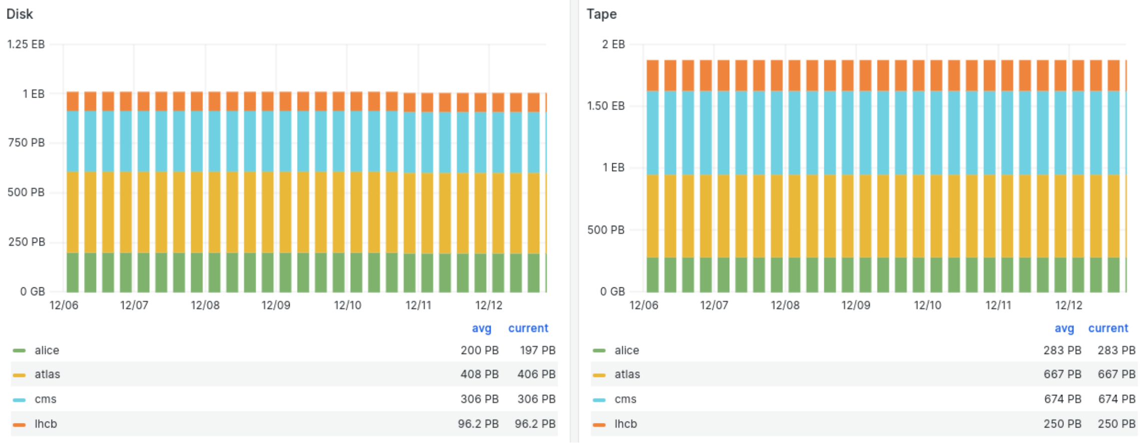


Figure 4.9.: Current storage pledges (2024). Tape pledges within the WLCG meanwhile sum up to enormous 1874 PB. Disk space reached the exabyte era as well with astonishing 1005 PB. As indicated by the two graphics, the whole storage infrastructure is a collaborative project of several experiments as part of the WLCG. Source: CRIC/MONIT

types for the experiments, like e.g. the CMS PREMIX datasets, which are used for the background mixing of MC simulations and can be of multiple petabyte size. They not only differ greatly in size, but also in access patterns. While raw data is comparably large (PBs), it is typically not accessed very often. User analysis level data (GBs for typical CMS NanoAOD) is significantly smaller but accessed more often. And additionally, the processed data, derived from the raw data, is provided in more different verbosity levels for making data access more efficient as users are able to selectively read only what is necessary. This is great for increasing processing efficiencies but can be a nightmare for fileservers, as it can lead to extremely different loads. This is, of course, also not ideal in the collaborative sense if a few institutions have to carry the maximum load while others only store big data tiers that are rarely accessed.

Gladly, there is a solution to this problem available within the WLCG in form of proper data management with Rucio.

4.2.5.1. Data Management, Distribution, and Access

Managing the countless datasets of different experiments in parallel on a shared, distributed storage infrastructure is no easy task, however, crucial for data integrity and flawless accessibility. On the one hand, save-keeping, distribution, replication, and cleansing of datasets must be ensured and on the other hand, the data has to be tracked properly so that researchers actually can find what they need. And ideally, all of this is kept a secret for the everyday scientists while simply providing them what they need.

For this purpose, Rucio [121] [122] was developed for ATLAS and later also introduced for CMS as the standard data management tool.[12] It is a scalable and flexible

[12] There, it replaced the old data management service, called PhEDEx, for Run 3. But in parallel, CMS is still maintaining its DBS [123] for physics metadata, including detailed information about datasets and processing history.

data management system tailored to the complex HEP use-case with highly distributed, heterogeneous storage backends and exabytes of data. The main features are a declarative, policy-driven data placement, automated dataset replication, and monitoring of data accesses while ensuring data integrity and availability across continents. On top, it provides a unified namespace helping users to locate the physical storage location of the datasets, which is highly significant in terms of accessibility and convenience. For the experiments, the data management with Rucio is employed as a rule-based system. The rules specify which datasets or files need to be replicated, where they should be stored for how long, and what data preservation policies apply. The replication policies ensure that critical and popular data has multiple copies stored in different locations. Rucio also tracks the access popularity of files potentially leading to a replication or a deletion depending on the access patterns. Availability and access efficiency for distributed users is enhanced by this, since a broad distribution of copies may lead to a more localized data access with higher bandwidth and lower latencies. The advantage is that it also ensures redundancy and reliability of the system and relieves the dedicated, global LHC networks, as the load is distributed between different regions.

The system then initiates automated data transfers and deletions, dynamically responding to resource availability and user requests, while optimizing storage and network resources. But Rucio is only used to determine the rules for optimal data placement, it is not actually executing the transfers, nor storing the data. For the data movement between the distributed storage systems, the WLCG relies on the File Transfer Servive (FTS) [124]. It is building the execution layer between the WLCG networks and the data management. Today, it provides real-time monitoring, a full REST-api, and multi-protocol support. Most important to mention here are WebDAV/HTTP, widely applicable as the Internet standard, and xroot, a HEP specific data transfer protocol for general data access and transfer.

For actually storing and accessing the distributed data, the WLCG heavily relies on eXtended ROOT Daemon (XRootD) [125, 126], the software framework implementing the xroot protocol[13]. It was introduced in the early 2002s at SLACK with the goal to address the challenges of data storage and access in distributed environments with an emphasis on scalability and performance. In the late 2000s, it was more and more used by the LHC experiments and from 2010 on, it became the core component of EOS, the distributed storage system developed by CERN, feasible of handling exabytes of data with billions of files.

Today, the whole software stack provides a high-performance, scalable, fault-tolerant, and secure solution for data organization, large-scale data access, and data processing in distributed computing environments and is build around a client-server architecture.

The client side of XRootD is designed for efficient and scalable data access (read and write) across distributed storage infrastructures and WANs enabling low-latency on-demand transfers of data to supply scientist all around the world with what they need for their analyses. In general, it supports various protocols and functionalities, including a POSIX-like interface for seamless file access and integration with existing software, like ROOT, allowing users to interact with remote files as if they were locally available. Alongside the XRootD protocol, specifically optimized for HEP applications, XRootD also supports WebDAV/HTTP, the multi-purpose, de facto

[13] The full protocol specification can be found in the official documentation, see e.g. Ref. [127].

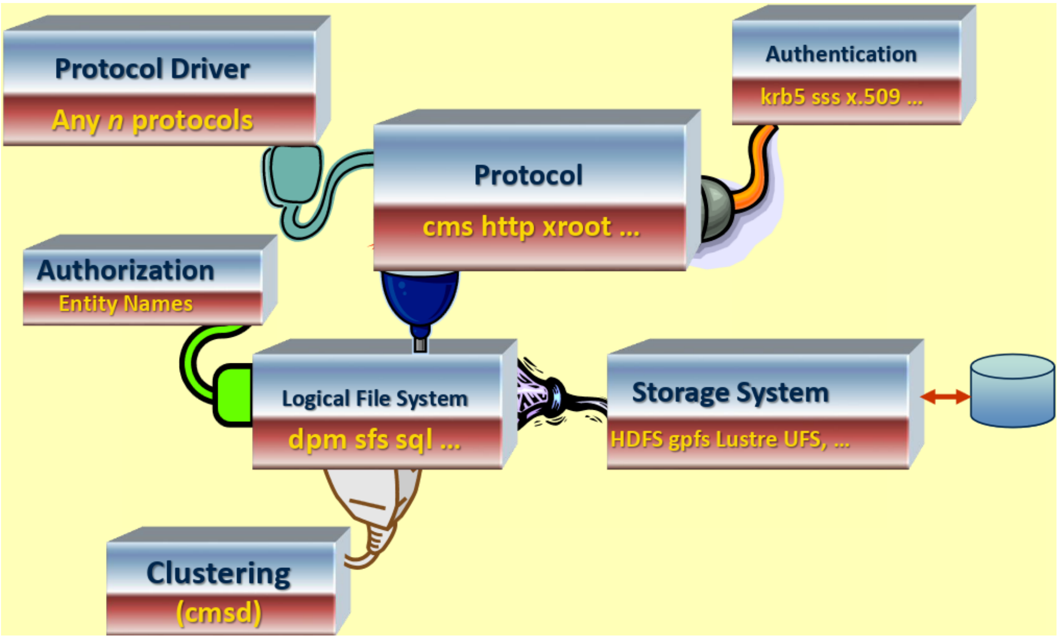


Figure 4.10.: The XRootD plugin structure. With different adapters, multiple protocols, various storage backends, and different authentication mechanisms can be implemented. The concept is therefore highly flexible and can be adapted to complex use-cases. Furthermore, additional functionalities, like e.g. caching, can be added with dedicated plugins. *Taken from*: [128]

industry standard, which makes it broadly applicable and portable. However, its biggest advantage (with xroot by design) in comparison to other data transfer tools are its low latency and overhead and sophisticated mechanisms for optimizing data access, like data streaming. The enabling of partial file accesses and on-demand retrieval of only the required parts of the data minimizes bandwidth usage and accelerates processing workflows. This is essential for the efficient data provisioning in distributed environments with the sheer amounts of data like within the WLCG, as only data is accessed and transferred that is actually needed, leading to overall better efficiencies and less (local) storage and memory requirements.

Furthermore, the XRootD client can be extended with plugins to modify its features and behavior, strictly following the modular, plugin-based software design approach of XRootD. These plugins make it highly flexible and adaptable to any kind of specific use-cases.

This also applies to the server side of XRootD, as depicted in Fig. 4.10. Each core component of the framework is provided as a separate layer and all additional features are then *plugged* into the core layers by configuring them accordingly.

By default, all special features and extensions of XRootD are disabled, reflecting the most simple setup possible: a data server that is exposing its underlying filesystem via the xroot protocol, as visualized on the top left in Fig. 4.11. Starting from this, each functioning layer is stacked with the others and each additional plugin with a designated functionality extends the overall setup (and can again be extended by a plugin!) to provide a more and more complex structure making it adaptable to every specific purposes while keeping it simple for simpler cases. The philosophy behind this plugin approach is that all use-cases, from a basic standalone data server to a complex cluster of several different kinds of specialized XRootD servers can be

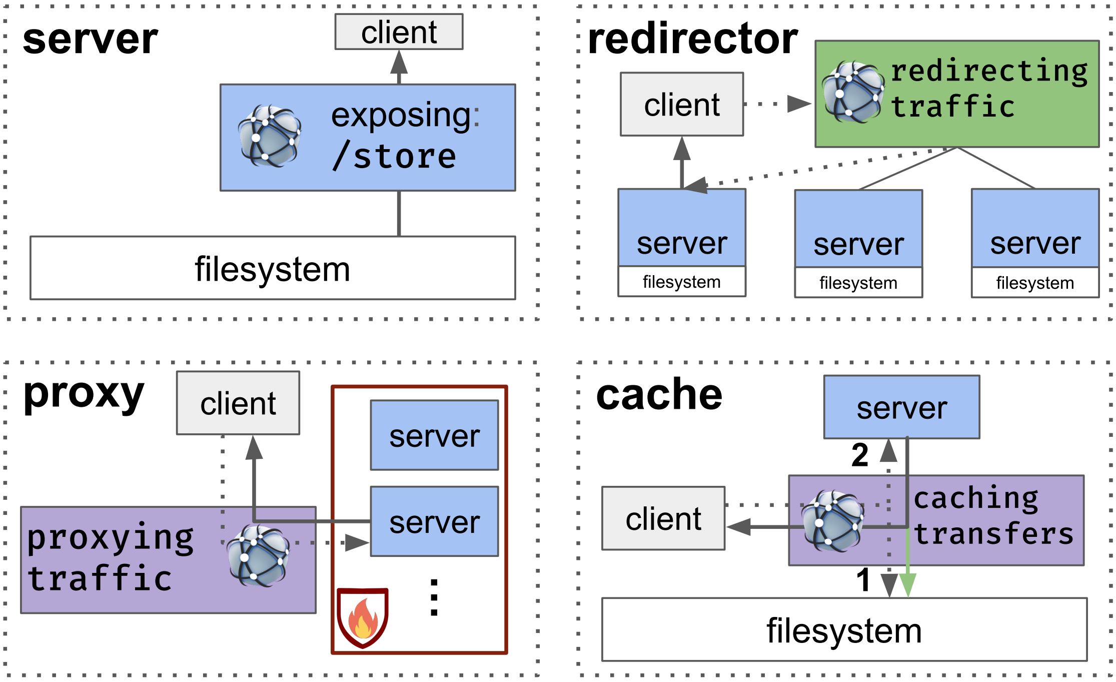


Figure 4.11.: Overview of the different default operational modes of XRootD. The most easy is a simple data server exposing a certain filesystem path (top left). An XRootD Redirector (top right) is a special server that manages multiple data servers (or other Redirectors as a meta manager). If a client asks for a file, indicated by the dotted arrow, the Redirector locates the requested file and provides the best origin for the transfer. The client then initiates a request at the reported origin and transfers the file. The Redirector itself, however, is not involved in the actual transfer. An XRootD Proxy (bottom left) is also redirecting a request of a client, but in distinction to a Redirector, the destination is fixed and the proxy actually acts as a client getting the data from the remote origin and redirecting/proxying the traffic, e.g. through a firewall, to the initial client. This functionality can be extended by a caching feature (bottom right). XCache, as this mode is typically called, acts accordingly to an usual XRootD Proxy, but the transferred data is additionally stored on local disk. Therefore, when a client requests a file at the XCache, XRootD first checks, if the file is locally available (1) and if yes, serves it from there. Else, it is – like the normal Proxy – requesting the file from remote (2) and additionally caching it on-the-fly on its local filesystem (green arrow), while serving it to the client.

covered, essentially just distinct in a more complex configuration file. All default plugins are shipped with the official installation and an overview is provided in Table B.1.

With the according plugins, the server side of XRootD can utilize different storage backends and allows to handle exabytes of data on-premise, as demonstrated with EOS. Furthermore, it is designed for creating a worldwide data federation by being able to loosely connect hundreds of independent, distributed storage resources. This is accomplished using dynamic data discovery in an hierarchical concept introduced by so-called XRootD Redirectors[14] that dynamically reroute client requests to the appropriate physical storage resources based on data availability and locality/latency. The functionality is visualized at the top right in Fig. 4.11, where a client asks an XRootD Redirector for a file. A combination of multiple layers of redirectors can form a flexible, distributed data federation. Exemplary, this is presented as deployed for CMS, in Fig. 4.12. Here, it is noted that the Redirector does not provide the files to the requesting client itself but only the designated address of the best matching origin where they can be accessed. By this, clients do not have to worry about where the data actually is stored, as they see the entire data federation only through a single point of entry, the XRootD Redirector, effectively hiding the full complexity of the global system. Implicitly, this behavior also introduces a natural load balancing mechanism, as the Redirectors take load and latency into account when selecting a remote origin. Additionally, security mechanisms like authentication and authorization, encryption, and data validation (checksum-checking) are provided on server level, which is important for a distributed infrastructure like the WLCG with many independent participants.

For the proper organization of the files, XRootD servers can manage and expose their stored data in a structured, hierarchical form, much like a traditional POSIX-filesystem, logically organized into nested folders and files. Accessibility for the experiments then is ensured by a standardized path structure. As a positive side-effect, this allows to separate the data of different VOs logically on the same physical hardware by different absolute path prefixes. Alternatively, it is also possible to export the whole storage and then specify the access rules via authentication and authorization.

Another important functionality that is to be mentioned in the context of this work is the XRootD Proxy server. It is essentially just a differently configured XRootD server, meaning it runs the same server backend with additional plugins enabled that allow traffic redirection, making it a proxy. This is useful for bypassing firewalls, as indicated at the bottom left in Fig. 4.11, or for redirecting traffic internally. In distinction to an XRootD Redirector, which only provides the location of a file, the proxy actually acts as a client fetching the data from a remote origin and providing it to the initial (local) client.

This functionality can be extended by a caching feature, called the XRootD proxy file cache (pfc). An XRootD server configured for this purpose is typically referred to as XCache. It acts as a transparent proxy for clients with the additional benefit of caching the transferred data. Consequently, if a client requests a file that was already cached, the XRootD Caching Proxy checks, if it is available and then serves it from the local storage instead of fetching it from a remote origin, which is visualized in the bottom right of Fig. 4.11. Here, it also supports the partial caching and accessing

[14] Technically, they are called XRootD Managers, but *Redirector* has become established in the jargon.

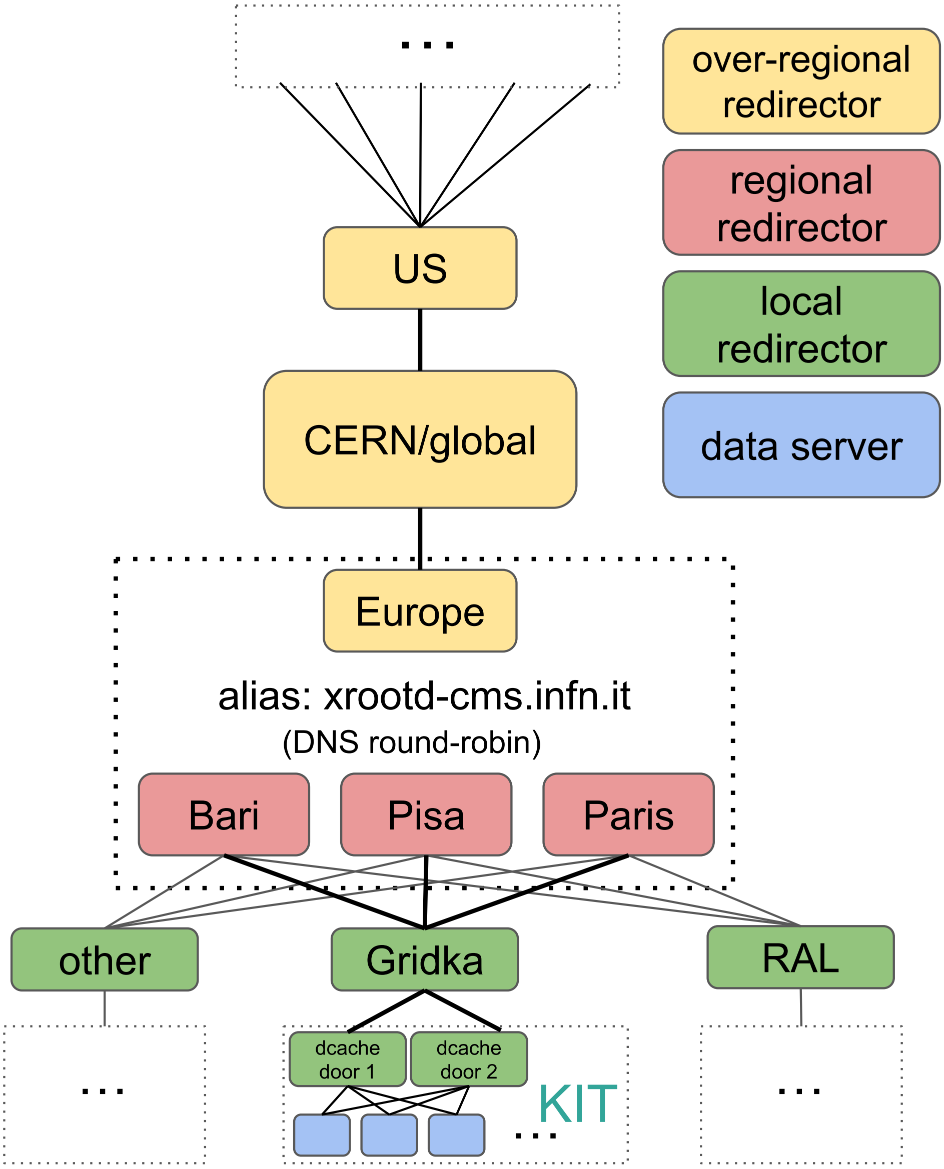


Figure 4.12.: XRootD Redirectors forming the CMS Any Data, Anytime, Anywhere (AAA) data federation. The hierarchical concept allows to form a globally distributed network of servers and to find files based on dynamic data detection and localization instead of a laborious bookkeeping, e.g. in a database. An example for accessing CMS data over the full redirector chain of the data federation is presented in Fig. 4.13.

of a file, as it is capable of storing only the requested and transferred parts (or only a small bit ahead with prefetching), instead of the full file, optimizing space usage and access speed. This can increase the bandwidth and reduce the latency and remote traffic, and additionally spares the remote servers, letting both sides benefit from the feature. As an example, it can be deployed at regional Tier-2 centers or Tier-3s, if they have spare disk space, to optimize transfers from the Tier-1 centers. However, whether caching is useful or not is a very complex question that heavily depends on the circumstances and use-case. The topic will be discussed later in detail (see Section 6.2.2). Here, it is only briefly mentioned that caching can of course, be helpful, as described above. But practically, there are also scenarios in which it is of little to no use, or – in the worst case – can even be disruptive. Therefore, the decision must be well-founded.

In summary, Rucio, FTS, and XRootD are all extremely valuable tools from the HEP community for the HEP community, together solving many everyday problems of scientific computing at unprecedented scales. Today, they are utilized for hundreds of thousands of data accesses and transfers in the WLCG on a daily basis and user as well as production workflows heavily rely on them. Ongoing improvements and continuous developments for adapting to the extreme needs of the HL-LHC are consolidating their role in the future of HEP computing. Examples of what is possible already today, can be found in Refs. [129, 130, 131].

The next section covers a real-world example for the application of the tools in the grid.

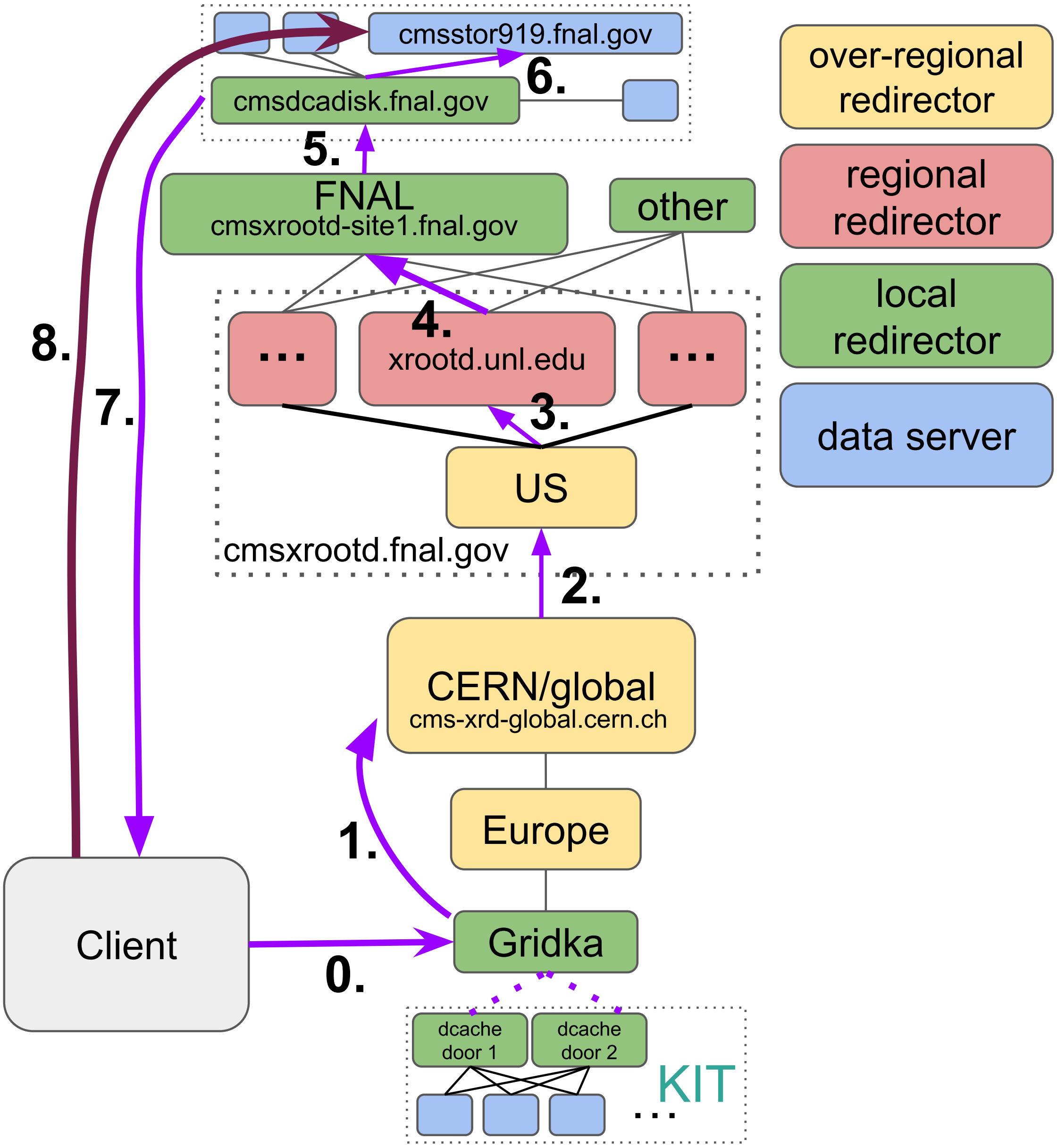
4.2.5.2. The CMS AAA Data Federation

The CMS AAA data federation is an impressive example for the capabilities of the above described tools. It is mainly showcasing how XRootD is able to efficiently form a global data federation across hundreds of distributed grid storage resources integrated into one system. In AAA, robust, scalable, and fault-tolerant access to petabytes of experiment data is provided to all CMS members, independent of their location, enabling the collaborative work on a global scale.

The data management is based on a global namespace `/store` subdivided into prefixes, like `/store/mc` for MC simulations, or `/store/data` for measured data. Files are then uniquely identified with their logical file name (lfn) and their storage location(s) are defined by the according physical file name (pfn)s. lfn and pfn are mapped together in the Rucio file catalog allowing to locate all files and their replicas within the data federation.

If this information is required, a user can query Rucio with the lfn or dataset name for finding the physical locations of all copies within the WLCG. But usually, this is not necessary for typical user or production workflows, as they just need access to the file – ideally from the best connected location with the lowest latency. This information, however, is not available in Rucio, as it depends on both transfer endpoints and is volatile and changes. For CMS's computing model with an on-demand streaming of the data, XRootD is therefore ideal with its dynamic data detection capabilities, which is the reason why it is fully integrated into CMSSW.[15]

[15] Remark: For ATLAS, this is not as relevant as for CMS, as its computing model foresees the pre-fetching of the data instead of the dynamic acquisition. ATLAS therefore has a full integration of Rucio into its experiment software for planing the data placement for job executions.



Redirect trace-back:

```
0. Redirected from: root://cmsxrootd-test.gridka.de:1094//store/mc/RunIISummer20ULPrePremix/[...]
1. Redirected from: root://cmsxrootd-test.gridka.de:1094/ to: root://cms-xrd-global.cern.ch:1094/
2. Redirected from: root://cms-xrd-global.cern.ch:1094/ to: root://xrootd.unl.edu:1094/
3. Redirected from: root://xrootd.unl.edu:1094/ to: root://cmsxrootd-site1.fnal.gov:1094/
4. Redirected from: root://cmsxrootd-site1.fnal.gov:1094/ to: root://cmsxrootd-site1.fnal.gov/
5. Redirected from: root://cmsxrootd-site1.fnal.gov:1093/ to: root://cmsdcadisk.fnal.gov:1095/
6. Redirected from: root://cmsdcadisk.fnal.gov:1095/ to: root://cmsstor919.fnal.gov:15461/
```

Figure 4.13.: Dynamic data localization with XRootD. In the top part of the graphic, the iterative process of locating a file within the hierarchical structure of the CMS AAA data federation is visualized. The process initiated by the client is running completely isolated and automatic. Eventually, the user is provided with the ideal file location and starts the transfer/data access. The bottom part of the picture is an excerpt of the CMSSW/XRootD log showing the summary of the multi-step redirection process.

This is also great in terms of convenience, as the automatic location does not require any individual effort. A job just simply sends its request for a file to an XRootD Redirector, as a single entry point to the AAA data federation, and is provided with the pfn. What happens in the background is visualized and described in form of a full redirect traceback in Fig. 4.13: Here, a client asks the redirector at GridKa for a file (0.). The redirector then checks its registered storage end points – indicated with the purple, dotted lines – and finds that the files are not locally available. It therefore refers upwards in the hierarchy of the data federation, in this case to the global redirector at CERN, for the file (1.). The request is redirected from the global, over-regional redirector to the consortium of regional redirectors located in the US (2.). The DNS round-robin is selecting `xrootd.unl.edu` (3.), which in turn redirects further to the local redirector at FNAL (4.). This redirector talks to one of the storage entry points (5.) to locate the file, which then identifies (6.) the physical storage element, here `cmsstor919.fnal.gov`. This information is then propagated back to the client (7.), who then initiates the data transfer or open request for the file directly at the remote origin.

This rather complex redirection chain, as taken from a CMSSW log (see bottom part of Fig. 4.13), works fully automatic and the client is essentially just provided with the pfn. As a positive side effect, this also provides an automatic fail-over mechanism. If a primary source is reporting problems or is unavailable, the client is automatically asking for an alternative origin, which is then provided by the initially asked redirector after the same principle.

By this, everyday, the CMS AAA data federation is utilized by hundreds of users for hundreds of thousands of file accesses and transfers, which form the basis for collaborative scientific research.

4.3. Monitoring

For a complex, worldwide distributed computing infrastructure like the WLCG, active monitoring and alerting mechanisms play an important role for ensuring an efficient, reliable, and secure grid operation with several experiments, thousands of users, and hundreds of resources, components, and services that all need to work flawlessly together to provide sufficient resources. A proper mid- and long-term planning is also only possible, if enough monitoring data is available to make estimations for the future resource demand. And short-term, keeping track of storage allocation and currently running workflows is crucial for the everyday operation. Imagine the WLCG is running out of tape space during data taking. That would be a disaster and must be prevented by proper planning and timely clean-up campaigns. And in terms of running workflows, a proper monitoring of efficiency and job failures is important to prevent wasting valuable resources. In this sense, a verbose monitoring is also the basis for improvements. Tracking the performance and identification of issues and bottlenecks is crucial to improve efficiency and operation in the future.

Therefore, CERN provides a comprehensive monitoring toolset for the WLCG with hundreds of use-cases and hundreds of gigabytes of monitoring data every day, all centralized in MONIT [132, 133]. A schematic overview of the complex monitoring system is given in Fig. 4.14.

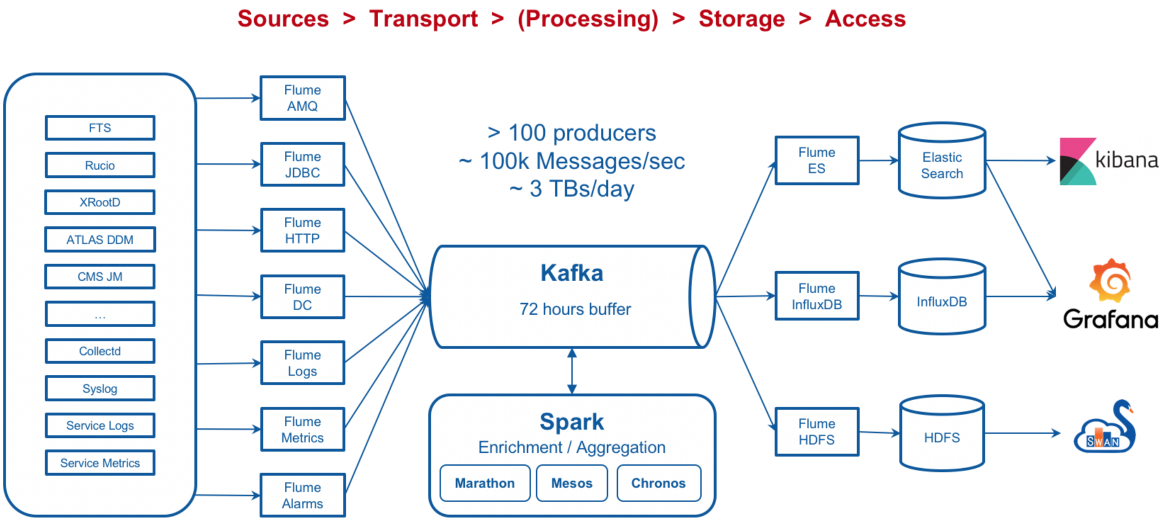


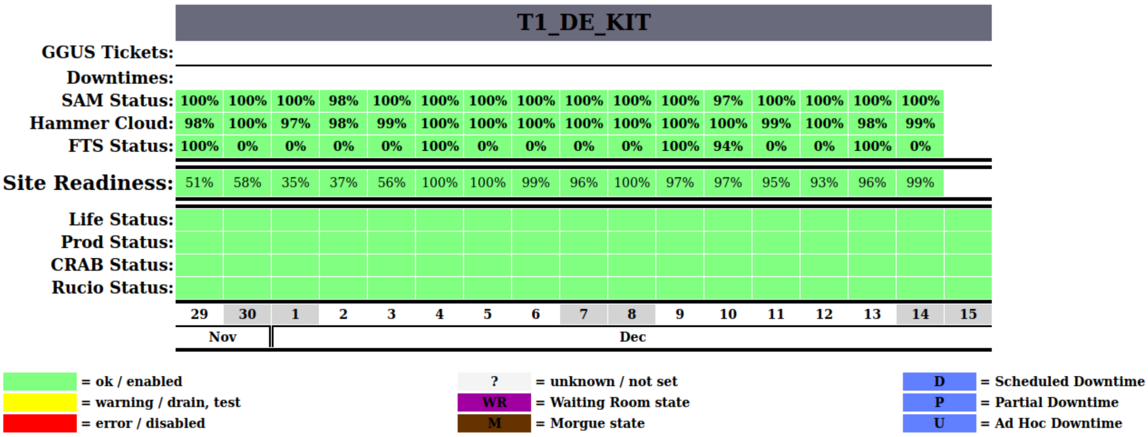
Figure 4.14.: With MONIT, CERN provides a centralized monitoring service for the WLCG sites, experiments, and internal IT services, like databases. The data is collected from different sources and aggregated in a message queue (MQ) system (here, kafka). Then, the monitoring data is ingested, processed, and distributed to the matching databases. *Source:* [134]

Most important for the WLCG site staff and the grid users is here the right part of the graphic, as all collected data is accessible via three different services: OpenSearch [135] with Dashboards, Grafana [136], and SWAN [137], a web-based analysis service providing Jupyter notebooks for data evaluation. Online, real-time monitoring databases, such as OpenSearch, however, only provide a limited scope (currently 30 days) for performance and storage reasons. The complete monitoring data is stored in an Hadoop Distributed File System (HDFS) [138] cluster at the CERN data center. To perform monitoring studies over a longer period of time or access older data therefore requires direct access to HDFS. This can be achieved via SWAN [137]. A comprehensive documentation can be found here: [139]. For accessing the long-term monitoring data on HDFS, SWAN has a spark cluster integration. Via the so-called Analytix cluster, it can be queried and the monitoring data retrieved. The systems are usable by every CERN user upon request.
The different types of available monitoring are briefly described in the following.

4.3.1. Infrastructure and Service Monitoring

With the complexity of the WLCG, it is important to keep an overview of all integrated infrastructure and services forming the grid. Therefore, monitoring[16] is implemented for the sites, from Tier-0 to Tier-3, that shows their status, availability, and reliability and helps to identify and solve problems fast. For this kind of monitoring, typically two ways of collecting the data are used.
The first is the central collection of logs and metrics from services, middleware, or even physical hardware, that are ingested by the full monitoring stack, as depicted in Fig. 4.14. After being processed and saved in the according database, they can be accessed via the web-interfaces of Grafana or OpenSearch Dashboards and for

[16] The WLCG monitoring can be accessed here: https://monit-grafana.cern.ch/?orgId=20 (2024).



	T1_DE_KIT																
GGUS Tickets:																	
Downtimes:																	
SAM Status:	100%	100%	100%	98%	100%	100%	100%	100%	100%	100%	100%	97%	100%	100%	100%	100%	
Hammer Cloud:	98%	100%	97%	98%	99%	100%	100%	100%	100%	100%	100%	100%	99%	100%	98%	99%	
FTS Status:	100%	0%	0%	0%	0%	100%	0%	0%	0%	0%	100%	94%	0%	0%	100%	0%	
Site Readiness:	51%	58%	35%	37%	56%	100%	100%	99%	96%	100%	97%	97%	95%	93%	96%	99%	
Life Status:	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Prod Status:	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
CRAB Status:	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Rucio Status:	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Nov		Dec														

🟩 = ok / enabled
🟨 = warning / drain, test
🟥 = error / disabled

? = unknown / not set
WR = Waiting Room state
M = Morgue state

D = Scheduled Downtime
P = Partial Downtime
U = Ad Hoc Downtime

Figure 4.15.: The CMS site readiness report for the German Tier-1 at KIT over a week in the end of 2024. As can be seen, all services are green and therefore fully functioning. Reports for all sites can be accessed here: [142].

example show the status of different XRootD Redirectors or the number of available cores at a certain site, et cetera.

The second way is active probing of the infrastructure and services to check the responsiveness and overall health of the systems. Different testing mechanisms were introduced as part of the WLCG Experiments Test Framework to verify on a regular basis that services are up, running, and working as intended, providing a reliable Grid Computing environment. The availability and functionality of sites is probed with so-called SAM tests, see e.g. Ref. [140] for CMS. These tests are centrally submitted to all sites and include checks of all crucial services for a reliable site operation, including the CEs, squids, the availability of CVMFS, data access via XRootD, correct site configurations, and many more. A full list for CMS is provided here [141]. From the results, one can get a comprehensive overview of the operations in the WLCG. In addition to being accessible within MONIT, they can be further aggregated and displayed in experiment-specific ways, like e.g. CMS is doing with their site readiness reports, depicted in Fig. 4.15.

The second kind of active tests used in the WLCG are so-called HammerCloud [143, 144] tests. Here, the focus lies on performance and functional testing of distributed computing resources with realistic HEP workloads instead of availability and reliability. The test framework submits (regularly or on-demand) synthetic, but experiment-specific tasks to mimic real-world workloads and can be used for stress-testing the infrastructure or conducting performance studies on grid or experiment software, like CMSSW. The results can be accessed over a web-interface and contain site and workload information as well as performance metrics.

In addition to the central monitoring, the sites – especially the Tier-1s – typically have own advanced monitoring systems available supplementing the standard WLCG monitoring to ensure a reliable site operation and the provisioning of resources. These depend on the infrastructure and the services that are provided. In the case of GridKa, for example, additional dCache monitoring is available for controlling the fill state of the different pools, or further information on integrated sub-sites.

And in general, it applies that additional, more basic information can be collected on site level than provided by the default WLCG monitoring, such as hardware metrics, power draw and others. These additional information are valuable, as they, for example, add information on the physical nodes as well as on pilot jobs that are wrapping usual grid jobs. Such extended monitoring capabilities are completing the picture and allowing more in-depth debugging and analyses, not only on payload level but on pilot or even physical node level. These information are useful, e.g. for efficiency studies, as they include the scheduling overhead, possible under-utilization of a physical machine, the out-running of pilot jobs, and other things that can degrade the actual efficiency of data processing but are hidden when only looking on job level. More details on those aspects are provided in Section 5.5.2. However, this kind of monitoring is mostly private or at least not directly integrated into the central monitoring.

In addition to the sites, also the central services at CERN are monitored, like MONIT itself, the CERN databases, and so on. As well as the networks, namely LHCOPN and LHCONE, with dedicated tools like perfSONAR. The publicly available network monitoring data, however, mainly focuses on data transfers, as described below.

4.3.2. Data Monitoring

In addition to the compute resources, the storage resources and data transfers must of course also be monitored. This includes the physical resources, the stored data, and the transfers within the WLCG that all need to work flawlessly to guarantee a smooth grid operation.

Over-watching the data placement is done with Rucio, as described in Section 4.2.5.1, which also provides internal monitoring and transfer information of individual storage resources, see e.g. Ref. [145]. Additionally, the sites report their disk and tape usage. The availability of the underlying storage systems, however, needs to be tracked separately. For testing the functionality of the Storage Element (SE)s, specific SAM tests are used that ensure accessibility of the servers. Eventually, all information is aggregated and provided to the experiments and users within the MONIT web-services (OpenSearch/Grafana).

Monitoring of the data transfers is mainly based on FTS and XRootD and also reflects the overall network usage within the WLCG. The two tools are capable of reporting their transfers, success rates, and throughput to the central monitoring instances. The comprehensive monitoring ensures that the data transfers and storage operations are completed correctly and possible file corruptions are detected and resolved automatically. In terms of XRootD, bigger changes are planned for the monitoring in the near future, as work is currently underway to integrate the XRootD Shoveler everywhere [146, 147, 148], an additional monitoring layer for aggregating more in-depth monitoring streams of XRootD on large-scale – mainly developed by OSG.

4.3.3. Payload Monitoring

The third important area is the payload monitoring of the experiments for which also CERN's central monitoring infrastructure is employed. For an efficient utilization of highly distributed, heterogeneous resources, the currently running workflows, their failure rates, their average efficiencies, and so on and so forth, need to be tracked. Only with this information, the experiments can optimize their software and workflows. And it is also important for detecting global problems, e.g. with faulty workflows or the AAA data federation, helping the responsible staff to react before they have a significant impact on the operation of the grid and too much compute power is lost.

The collected data and its origin, however, is typically different for each experiment as it mainly depends on the experiment's workflow management and analysis software and how it is provided. Here, only the CMS perspective is explained, as it is the most relevant for this work.

The available monitoring data for CMS jobs origins from HTCondor and WMAgent[17] [149] – the workflow management tool employed by the CMS Collaboration – and are written in two different indices in OpenSearch: `condor_raw_metric*` and `wmarchive*`. From the HTCondor monitoring, real-time[18] information on the match-making (ClassAds), the occupied resource slot, and classic monitoring metrics, such as wall clock time, CPU time, memory, disk usage, and others, can be extracted. The WMAgent monitoring data is written on job completion and includes workflow specific information, like e.g. used dataset lfns, and also error logs of failing jobs.

From the site operation perspective, this information is extraordinarily useful for the identification of problems with the own or another site's compute or storage resources and helps with fast troubleshooting, as it provides insides on the reasons for job failures. From the experiment's point of view, it can be used for the debugging of production workflows and in-depth efficiency studies to find bottlenecks, which is the first necessary step for improvements, as demonstrated later on.

One problem with the two separate indices is that the two data sources contain complementary data that are both useful, but very hard to match on job level. Out of scope of this thesis, I found a possibility based on a self-constructed global job identifier, the log URL pointing to the out-staged job log file on EOS, and provided a P.o.C. implementation to the CMS computing operations and monitoring team. For further details, see Refs. [150, 151].

In addition to the two described monitoring sources for CMS workflows, the software logs of failed jobs are saved on EOS, including the job report and the actual CMSSW logs. An automatic collection of this data is not in place – and would also go beyond the scope of the default monitoring – but the logs can be accessed manually. They contain valuable insights on what a grid job was actually doing – including the timings – which can be useful to find out where and why problems occurred. Furthermore, they are necessary for debugging complicated job failures or identify bottlenecks in the processing chain, when jobs do not reach the performance and (CPU) efficiency targets.

In this context, such job logs can be important for efficiency studies, as they supplement the overall picture by what was happening inside a job. While the general monitoring just provides a value for the CPU efficiency of a certain step in the processing chain, it is not providing any insights into why or how well certain sub-task as performing. Is a job e.g. waiting for data because of having trouble with a remote site, or is it maybe even a network issue? Or just a configuration issue of the workflow that may affect the efficiency of a certain part of the job? This brief discussion shows that such information can contribute to improve the overall grid operation,

[17] To be precise: WMStats, which is also part of WMCore and serves the monitoring data from WMAgent.

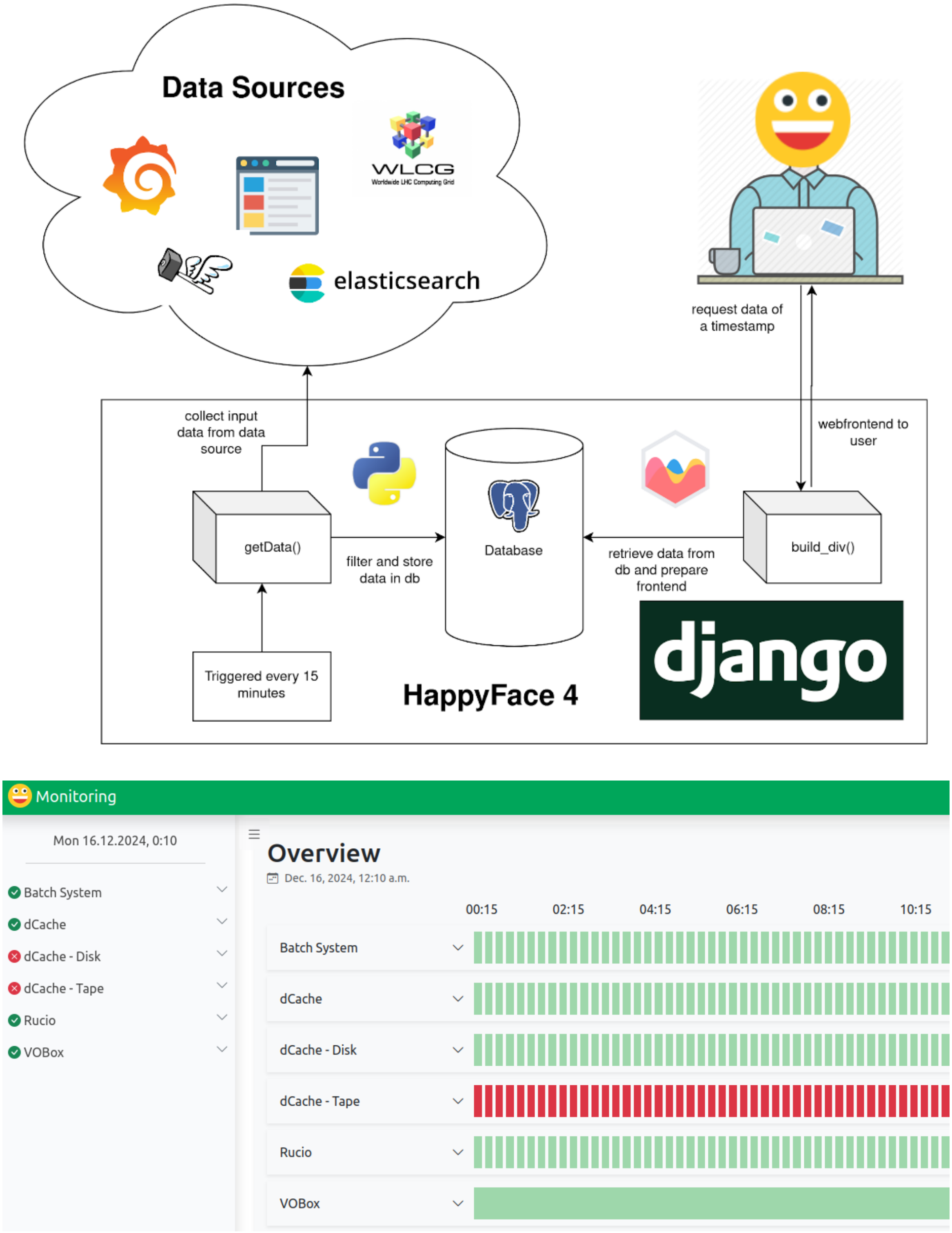
[18] Aggregated in 12-minute bins for practical and performance reasons.

even if it is not propagated to the central logs.
The downside of this is that the accessing is laborious and these logs are only available for failed jobs, firstly for storage reasons, as saving all logs would be too costly, and secondly because random selection of successful job logs would not be expedient as it would not guarantee that the required logs are actually available. To fix this issue, I found a way to collect successful job logs, provided in Ref. [152].
Taken all together, the available monitoring provides a comprehensive picture on what is going on in the grid and allows to investigate issues and improve the overall operation. However, it is also very complicated to keep an overview by checking the different monitoring sources manually. A simple solution for this challenge is described in the following section.

4.3.4. Meta-Monitoring

In a complex system like the WLCG it is difficult to keep track of everything as the different, relevant information can be found in various places and may require manual effort to be retrieved only to get an overview of what is going on in the grid. This makes the identification of problems more complicated, as keeping track of all the above described monitoring sub-categories with the many sources and relevant parameters and metrics is time-consuming. Meta-monitoring and visualization tools, like Grafana, which is aggregating information from different sources, are helping here. However, they often have their limitations when provided as central services, as they typically do not include site-specific, local monitoring information and the security policies often limit the permissions of default users.
A more flexible and easy to deploy solution for grid sites is HappyFace4, a meta-monitoring tool developed at KIT [153]. It is a django-based web-service with a PostgreSQL backend designed to collect and digest monitoring data from arbitrary sources and input types, as depicted in Fig. 4.16 (top). By condensing the variety of diverse monitoring information in one place, HappyFace4 provides a simple solution for this struggle, thereby greatly simplifying the process of getting an overview. The interface is shown as deployed at KIT[19] at the bottom of Fig. 4.16, providing a status overview of all services at GridKa on one page. Further, more detailed information are then provided in the according categories. In the depicted example, the tape system reports problems, which now can be investigated in detail by the shifter team. Further details can be found in the official documentation, see [154].
Today, HappyFace4 is widely used in the German WLCG community helping the local teams to provide reliable grid resources.

[19] Public HappyFace4 instance: https://hf.etp.kit.edu/


Data Sources
WLCG
Worldwide LHC Computing Grid
elasticsearch
request data of a timestamp
collect input data from data source
webfrontend to user
getData()
filter and store data in db
Database
retrieve data from db and prepare frontend
build_div()
Triggered every 15 minutes
HappyFace 4
django

Monitoring

Mon 16.12.2024, 0:10

- ✓ Batch System
- ✓ dCache
- ✗ dCache - Disk
- ✗ dCache - Tape
- ✓ Rucio
- ✓ VOBox

Overview

Dec. 16, 2024, 12:10 a.m.

	00:15	02:15	04:15	06:15	08:15	10:15
Batch System						
dCache						
dCache - Disk						
dCache - Tape						
Rucio						
VOBox						

Figure 4.16.: The top picture shows the schematic concept of the HappyFace4 meta-monitoring tool based on django with a PostgreSQL backend and arbitrary data sources (*Credits: Dr. S. Brommer*). The bottom picture shows the landing page of the HappyFace4 instance deployed at KIT for GridKa. The very helpful tool provides an overview of the entire site status within seconds and gives hints on services that may need to be investigated further. In the case presented here, already on the first glance, a shifter can identify issues with the tape systems, while all other services are well functioning.

4.4. Benchmarking and Accounting

In addition to the operational aspects, the monitoring is also the foundation for a fair and transparent resource utilization and enables appropriate accounting and recognition. Because despite all the flexibility, the WLCG also needs to guarantee planning security for both the experiments and the resource providers. It needs to be ensured that member institutions commit specific amounts of compute and storage resources to meet the demands of the experiments. This is done through official yearly pledges listed in the Computing Resource Information Catalog (CRIC) [89, 155]. There, the commitment of the resource providers is quantified and recorded. And in turn, the usage of the resources must be tracked properly, on the one hand, for fairness between the VOs, and on the other, for a proper accounting and attribution of provided resources, which is often important for the sites in terms of funding and recognition.

Despite the rather homogeneous, hierarchical concept of the WLCG, this is not trivial at all, since the tiered architecture defines the tasks and requirements for a site, but does not specify how they are to be met, e.g. in terms of hardware. From this results again a very heterogeneous environment, because different architectures, hardware performance profiles, and local configurations must be considered accordingly.

Since an evaluation and rating of every single machine out there in advance is not feasible and a benchmarking on-the-fly when a job is executed would create a significant overhead, a standardized, consistent, and reproducible benchmarking process of resources as a basis for comparability is required for a proper accounting. For many years, commissioned in 2009, this was provided by HepSpec06 (HS06), a benchmark that rates the performance of compute nodes by running a suite of standard tests based on the all_cpp benchmark subset of SPEC CPU 2006 [156, 157]. In 2023, with HepScore23 (HS23) [159], a modern successor was introduced to provide a more realistic rating for modern systems – especially for multi-core applications – and to include other architectures. It evaluates the performance under typical HEP conditions instead of synthetic scenarios. The benchmark is executing seven different workloads from five LHC experiments (3 single- and 4 multi-threaded) [158] for different sites and each individual node configuration of a cluster, e.g. yielding different measurements for a certain CPU, as depicted in Fig. 4.17. To ensure comparable results, it is running in containerized environments with fixed, identical software and settings regardless of the underlying architecture. By using whole nodes for benchmarking, potential biases caused by the interference with other running jobs are eliminated. Eventually, the results are calibrated and normalized to a reference system[20] [158] to get a comparable score representing the throughput of the system. The performance of all other systems is expressed as a ratio to this reference system, enabling a consistent and comparable measure. Results are publicly available here: [160].

These scores are then used for accounting within the WLCG by a standard procedure. Each grid job provides monitoring data and logs containing its consumed resources that are digested and reported via Accounting Processor for Event Logs (APEL) [161, 162] to EGI. For official EGI VOs, see Ref. [163].

However, since the accounting toolset is not covering all use-case in today's grid

[20] Normalization: 1 to 1 with HS06 for the reference CPU Intel Xeon Gold 6326 CPU @ 2.90 GHz with hyper-threading.

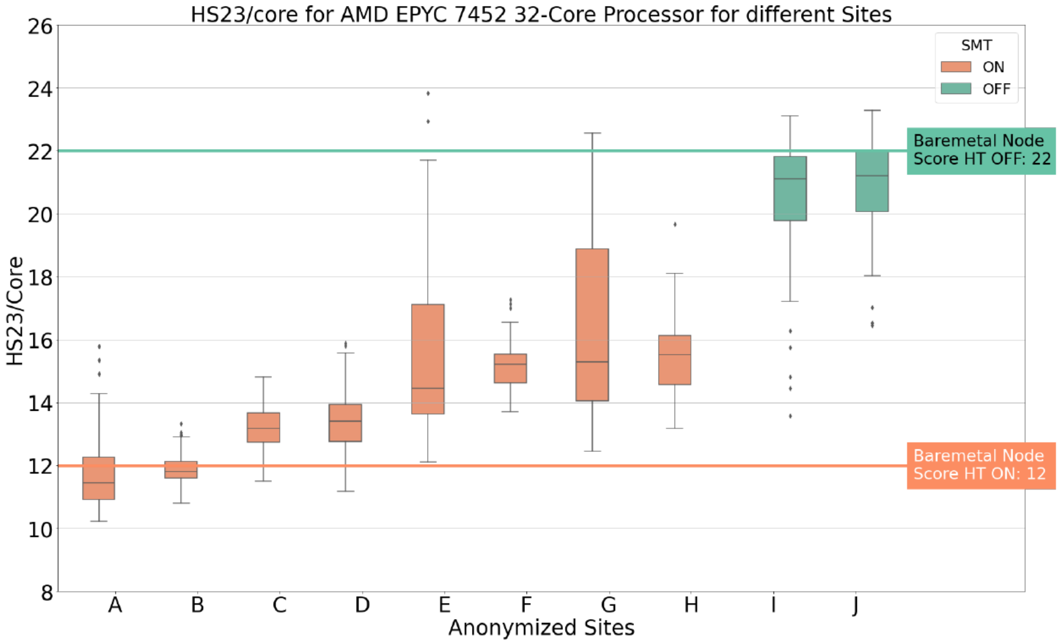


Figure 4.17.: HS23 benchmarks for the AMD EPYC 7452 CPU deployed at different sites. The result shows a 10-20 % spread between the sites – which is expected as the exact configuration of worker nodes may differ – and a significantly different throughput per core with and without AMD's Simultaneous Multi-Threading enabled. To account for all those differences, the sophisticated benchmarking mechanisms of the WLCG are necessary. *Taken from:* [158]

computing environment, new tools, like **A**cco**U**nting **D**ata handl**I**ng **T**oolbox for **O**pportunistic **R**esources (AUDITOR) [164], were introduced to include additional aspects, like sub-sites of official WLCG sites. Further details can be found here: [165, 166], and the concept is depicted in Fig. 4.18.

In summary, benchmarking and accounting are key aspects of grid computing and of much importance for the WLCG. Standardized performance metrics for the diverse infrastructures allow fair resource allocation across all sites and VOs and an accurate accounting which ensures that resource usage is transparently tracked and reported, enabling a proper attribution of commitments.

4.5. The Dynamic Integration of Opportunistic Resources

As described above, benchmarking and accounting are the basis for a fair and functioning cooperation between the sites forming the WLCG. The basis for that are formal agreements between the institutions providing the resources and the WLCG. Official grid sites typically have to guarantee a fixed commitment (pledge), a minimum availability, as well as a default grid environment with the required middleware, so that the functionality of the whole system can be guaranteed. These agreements ensure that enough resources are available for the scientific community, satisfying the demand for computational and storage capacity over extended periods, typically spanning at least one year.

However, not all resource providers can meet these requirements to participate as classic grid sites in the WLCG, but want to contribute anyway. One way is to

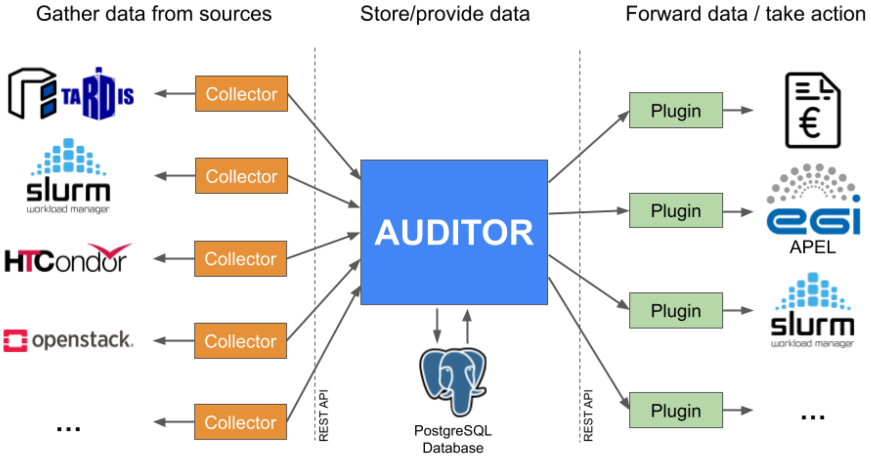


Figure 4.18.: The AUDITOR Accounting Tool builds an additional interface layer between the classic WLCG accounting and the resource providers. It facilitates the collection, processing, and reporting of accounting data from various resources to various destinations. With its collector and plugin architecture, it is very flexibly and can be used for more complicated use-cases, like the accounting of WLCG sub-sites to EGI with its APEL plugin. It furthermore provides plugins for an advanced resource utilization and priority management.
Taken from: [166]

provide available resources to individuals or groups of scientists, e.g. in form of a local analysis facility at an university. Other than that, institutions are also allowed to share whatever resources they can provide with all members of the WLCG by integrating them on a temporary and availability basis into the grid.
While the first described resources are the outermost part of the hierarchical WLCG concept in form of Tier-3 centers providing compute power mostly to local users, the second type of resources are regarded as so-called opportunistic resources, described below in more detail. Obviously, both of such resource types are very welcome – even if they are not guaranteed – as they help with the mission of providing sufficient computing power to the scientific community.

4.5.1. Opportunistic Resources

All kinds of resources that temporarily contribute to WLCG without a formal agreement or fixed commitment – from a single server to a large-scale HPC cluster, or even commercial cloud providers – are summarized as opportunistic resources. They are typically allocated dynamically and temporarily, meaning they contribute when they are available without any guarantee of commitment in advance. The reasons for institutions not fully participating in the grid are as manifold as the resource providers themselves, but should not hinder to contribute to the WLCG anyway, if desired.
For instance, some institutions may cannot commit fixed pledges or meet the strict availability requirements for becoming an official grid site. A local analysis group at a university with some compute resources in form of a local Tier-3[21] in their

[21] For example: TOpAS at KIT provides slots to the WLCG when not fully utilized by the local users.

basement, for example, may does not need them all the time and could in principle make them available to the WLCG when they are idle. However, as they cannot accurately predict their own usage, they cannot make any firm commitments regarding a pledged contribution. And additionally, they typically lack the necessary person power to provide neither the services required for official WLCG sites, such as CVMFS, nor to guarantee a reliable 24/7 operation.

And other resources may serve a different original purpose and are neither dedicated to the WLCG, nor interested in becoming an official part of it, but just have idle resources from time to time that they want to share on a temporary basis, like scientific HPC clusters, for example.

In general, even if resources cannot meet the requirements or provide fixed commitments, they are much appreciated from the WLCG perspective as they provide compute power in times of ever growing demands and may even extend its capabilities with technologies that are not (yet) an official part of the WLCG computing model, like for example GPU resources. With the increased diversity, also a better utilization of the resources can be achieved, as the probability rises that jobs can be matched to resources where they can ran the most efficient. And ultimately, the additional resources can cover shortages, bridge bottlenecks, and help to mitigate the impact when resource failures occur at official grid sites.

But, as already mentioned briefly above, the main difference of opportunistic resources is that they do not have a formal agreement with the WLCG, so this is not guaranteed. For the providers, this means on the one hand that the corresponding resources cannot be accounted for and contribution is fully voluntary{22}, but on the other hand, this gives them a lot of freedom as to how and when they are integrated. This, however, disqualifies storage resources obviously, as for them a proper planning and reliability is required. Therefore, opportunistic resources usually only contribute computing power, but no storage other than volatile caches or scratch space that is required for job execution, as they are normally provided without long-term planning or strategic intent.

In general, allowing the contribution of resources on an opportunistic basis is a good compromise from which both sides can benefit and is being practiced broadly across the WLCG. Already today, such resource account for a significant part of the delivered core hours of the WLCG. The opportunistic contribution in Germany in 2024, for example, adds up to more than 1 million core hours per month on average available for different experiments, as shown in Fig. 4.19, which corresponds to the contribution of an average ATLAS Tier-2.

However, the integration of opportunistic resources poses major challenges, as they comprise many different types with heterogeneous infrastructures and divers compute and operational models and policies. Since they typically do not have an Memorandum of Understanding (MoU), which defines how resources are allocated, shared, and managed, they usually cannot be integrated directly, as they do not have the required tools and services available, like e.g. HTCondor CEs for the resource management and scheduling. And additionally, without an MoU they are also excluded from LHCONE.

It is such an opportunistic grid resource.

{22}Remark: This disqualifies cloud providers with a payed on-demand provisioning of cloud resources, as described in Section 4.2.3.3, as opportunistic resources in the classic sense, even if they can in principle be integrated opportunistically.

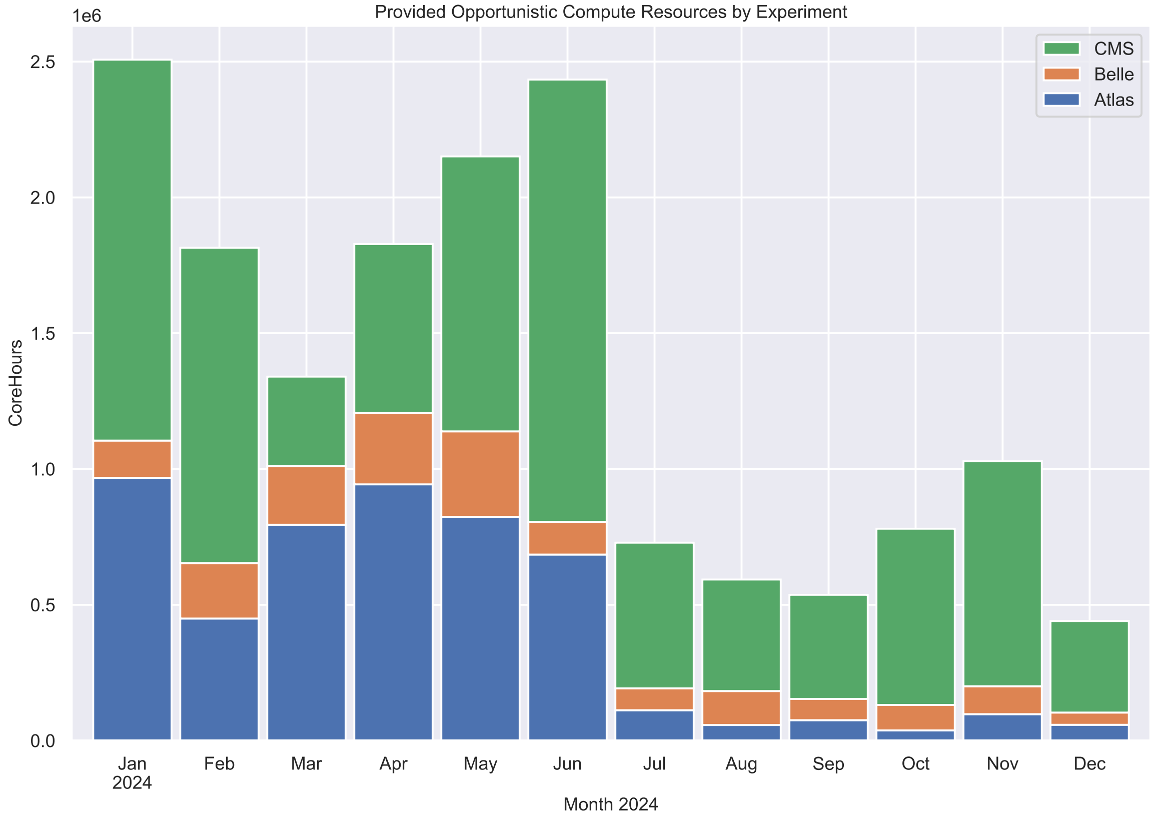


Figure 4.19.: Opportunistic resource contribution in Germany in 2024. With more than 15 million core hours provided by several different institutions across Germany, which are not exclusively dedicated to the WLCG, but provide resources intermittently when they are available and have capacities. The contribution is comparable with an average ATLAS Tier-2. The provision for 2022 and 2023 are shown in Fig. B.5.
Source: Dr. Manuel Giffels

Therefore, modern and sophisticated tools, such as TARDIS, are required to enable the dynamic and transparent integration of such resources.

4.5.2. COBalD/TARDIS

The efficient integration of opportunistic resources is one of the biggest challenges since the integration process strongly differs for all kind of resources. It is a big difference, if a single server should be integrated or an HPC cluster. Also within the categories the process might be different, as e.g. usually no two HPC centers are to hundred percent identical with different setups and different batch systems. And even if they are very similar hardware and software wise, they may differ in policies or political factors that come into play.

To resolve this issue, COBalD/TARDIS [167, 168, 169, 170] was introduced. It is designed to provision and orchestrate (cloud) resources dynamically. COBalD is the core framework for the dynamic scaling as it acts as a resource broker. It monitors the batch queues and resource allocation and utilization to evaluate the resource demand and can adjust the resource pool accordingly. With this, the tool ensures that over-provisioning is avoided and resources are used as efficient as possible.

TARDIS then handles the actual provisioning. It is a plugin for COBalD, which builds up an interface for requesting and managing the resources. With different so-called site adapters [171] for all kinds of resources, it simplifies the integration of different resource types significantly by handling the interaction with the local resource management systems, like e.g. Slurm, OpenStack, or Kubernetes. For this, a so-called drone is started that allocates the resources, typically provided in form of containers, VMs, or even whole physical compute nodes. The drones are basically skeleton jobs that are used to provide an environment according to the needs of the WLCG pilots and make all relevant tools for a seamless integration of the resource available. The software framework therefore helps overcoming the problem of middleware availability and closing the gap between the local resource allocation systems and the workload management systems of the WLCG. The concept is described in detail in Ref. [169].

Resources that are allocated by TARDIS are then registered and seamlessly and transparently integrated into one so-called Overlay Batch System (OBS) creating a unified, dynamic resource pool that can span several resources and can be addressed by a job submission system. The whole concept is visualized in Fig. 4.20.

In the context of the WLCG, the job submission is handled via HTCondor cloud CEs (CMS) or ARC CEs (ATLAS) that work as gateways providing an entry point to the local OBS. The workloads of the experiments are matched with the resources transparently as the available slots appear as normal resources in the global pool, while hiding the complexity from the users and reduce the maintenance effort for the resource providers, since all this happens fully automatic. In Fig. 4.21, the current situation in Germany is depicted. Several resources are integrated into one HTCondor OBS fronted by two cloud CEs at GridKa. Currently opportunistically used by CMS are TOpAS, the local KIT Tier-3 center and the university's scientific HPC cluster HoreKa, which is described in the next section.

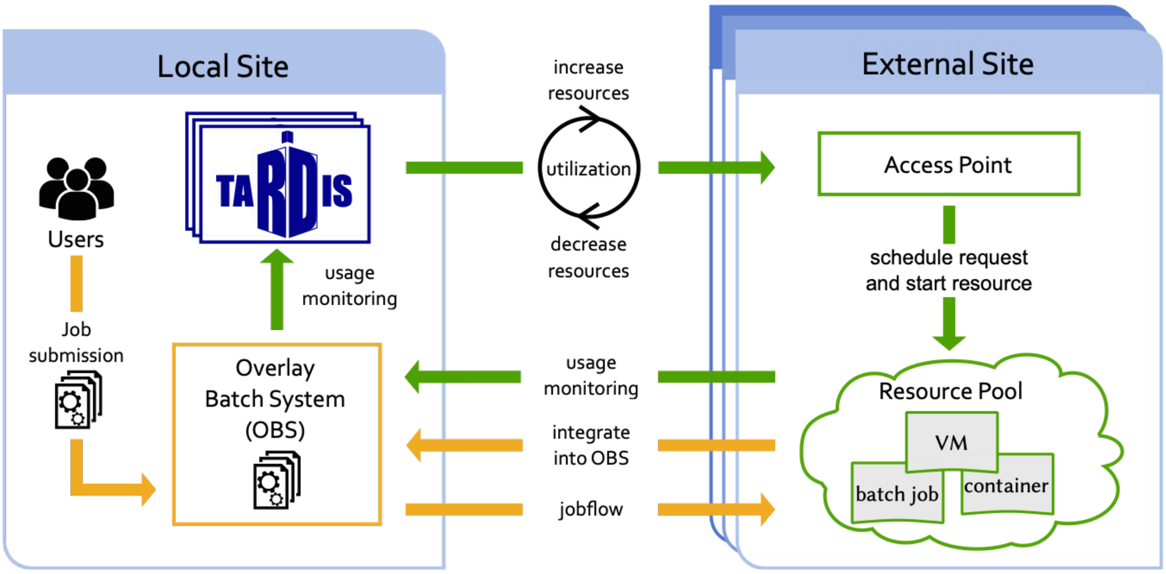


Figure 4.20.: COBalD/TARDIS builds up a layer between the providers of opportunistic resources and the users. Depending on the demand, the system allocates resources and integrates them transparently in an OBS. By this, the complexity of the requesting and provisioning of the resources is hidden from the users as they appear to them just like normal grid resources.
Source: Dr. Manuel Giffels

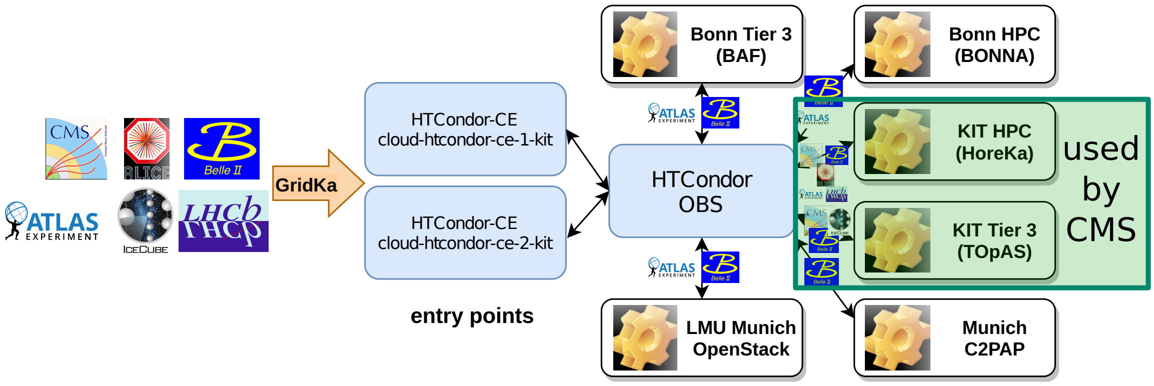


Figure 4.21.: Dynamic integration of opportunistic resources with COBalD/-TARDIS in Germany. The system allows an efficient allocation and utilization of various resources. From local Tier-3 centers (Karlsruhe, Bonn) to scientific HPC centers (HoreKa, BONNA) and private clouds (LMU) are integrated opportunistically and provide several million core hours to different experiments every year, as depicted in Fig. 4.19. *Source: Dr. Manuel Giffels*

4.5.3. The Opportunistic Integration of the HoreKa HPC Cluster

As described above, the provided software framework enables the dynamic and transparent integration of all kinds of distributed compute resources, including HPC centers.

The opportunistic usage of such resources obviously has advantages for the HEP community, as additional, high-performant resources are provided. But also the HPC centers can profit from the joint venture with the HEP community, as described in Section 4.2.3.2 and visualized in Fig. 4.6.

One example for such a cooperation is HoreKa, a local scientific HPC cluster at KIT. Gladly, the GridKa computing team was allowed to integrate the cluster as an opportunistic resource into the German Tier-1 center since its commissioning as part of a research and development project for the future utilization of HPC resources in the German HEP community.

4.5.3.1. The HoreKa HPC Cluster

The Hochleistungsrechner Karlsruhe (HoreKa) is an HPC system located at the Karlsruhe Institute of Technology in Germany. It is a mid-sized HPC system and one of the nine centers in the German National High-Performance Computing Alliance (NHR) initiative, alongside Aachen (CLAIX), Göttingen (EMMI), and others. The NHR centers are funded jointly by German federal and state governments and are designed to support research in various fields such as physics, chemistry, biology, climate science, and engineering not only with compute resources, but also with support and trainings [172].

In terms of computing resources, HoreKa can provide classic CPU and additional GPU resources, which are separately partitioned in a hybrid architecture, as shown in Fig. 4.22. Its design was selected with a particular focus on efficiency and sustainability. The warm-water cooling system greatly enhances energy efficiency and the waste heat absorbed by the warm water is used for heating the nearby Scientific Computing Center at KIT Campus North, aligning with the overall commitment to environmental sustainability and innovative energy management.

Even if not used for the rest of this work, HoreKa Teal should be mentioned in particular in this context, as it reached the 6th place of the Green500 list (06/2024, [174, 175]) with an outstanding energy efficiency of 63 Gflops per Watt, which gives a glimpse of what is possible in the future of computing. In view of the expected increase in resource demands for HEP computing in the future and the self-set goals in terms of social responsibility and sustainability, such developments are of great importance for the LHC computing as a whole.

The technical setup of the CPU partition of the cluster – including the connectivity of the different nodes – is shown in Fig. 4.23. Every worker node is a dual socket Intel Ice Lake server with around 500 GB of memory. The nodes provide local scratch space and additionally, each user has a 250 TB share on the RDMA-enabled IBM Spectrum Scale/GPFS which is mounted on all nodes. All worker nodes are internally interconnected via fast InfiniBand with IPoIB enabled, also with the login nodes, as indicated in light blue. The cluster edge nodes provide a 50 Gbit/s connection to the Internet, while the workers also have a WAN connection, but only with 1 Gbit/s. A central firewall is regulating the traffic to the cluster.

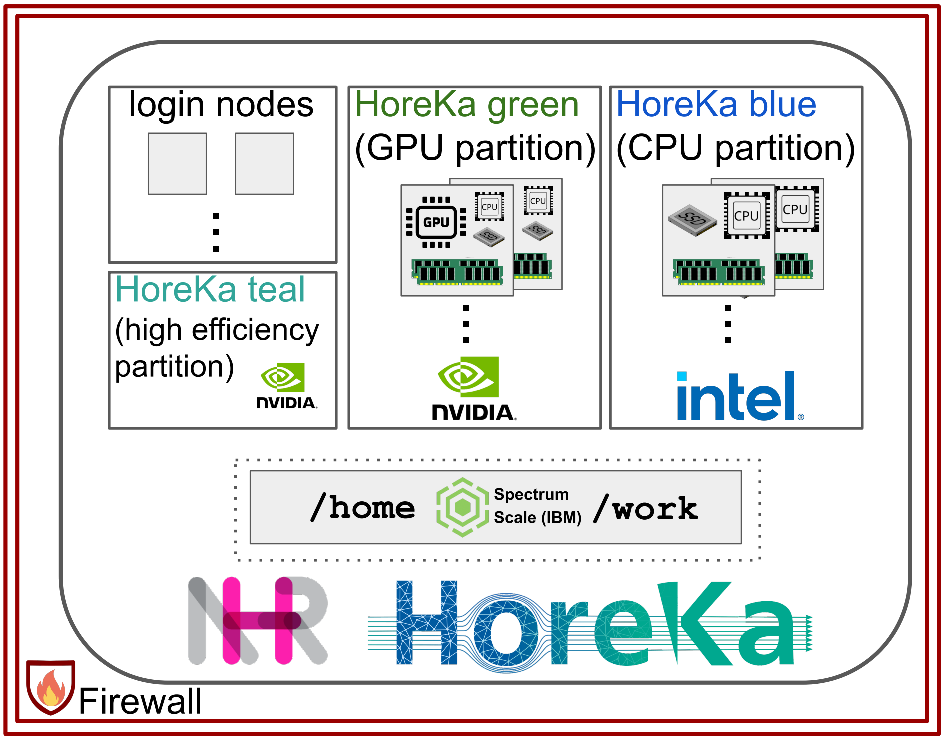


Figure 4.22.: The HoreKa hybrid architecture. The design comprises two main partitions: HoreKa blue, the CPU partition (60.000 cores), and HoreKa green, the GPU partition (668 nVidia A100) [173]. In total, circa 300 TB main memory are available. Additionally, a high-efficiency research partition, called HoreKa Teal, is providing 88 nVidia H100 GPUs. Furthermore, around 16 PB storage are available, mounted via fast IP over InfiniBand.

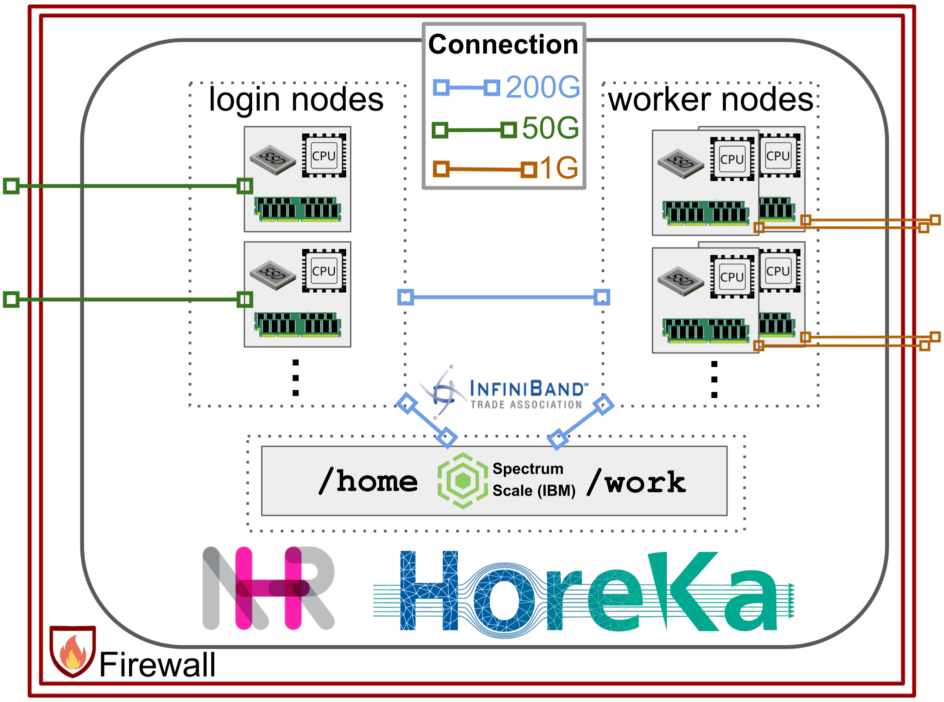


Figure 4.23.: The setup of the HoreKa CPU partition. The servers provide 76 physical cores in a dual socket setup and 500 GB of memory. The login nodes are internally connected to the worker nodes via fast InfiniBand and externally with 50 Gbit/s. The worker nodes only provide a 1 Gbit/s WAN connection to the public Internet.

For using HoreKa on a user basis, every researcher with a doctoral degree and an affiliation to a German accredited university can hand in a proposal for NHR resources to carry out their research projects. Within the scope of approved projects, also non-doctoral scientists can use the resources [176].

For collaborations or experiments, such as CMS, this is only possible in a special co-operation like the research project with the local GridKa R&D team, which integrates the cluster opportunistically into the WLCG since more than three years. Details on setup and integration are provided in the next section.

4.5.3.2. Integration Details

As shown in Fig. 4.21, HoreKa is one of the opportunistic resources that is currently integrated with COBalD/TARDIS into the WLCG. It can be used by CMS and ATLAS independently. While conceptually similar, the integration details will only be given for CMS in this work.

At the moment, only the CPU partition is used as opportunistic resource, but in principle GPUs could be provided in the future, too – if there is demand. For the local resource management and scheduling, Slurm is used, as very common for scientific HPC centers. With it, users can request whole nodes in each of the partitions.

The informal agreement for the utilization of HoreKa for CMS is based on a back-filling approach and (currently) only includes production jobs. For this, TARDIS – configured with its Slurm site adapter – is requesting compute nodes with the lowest

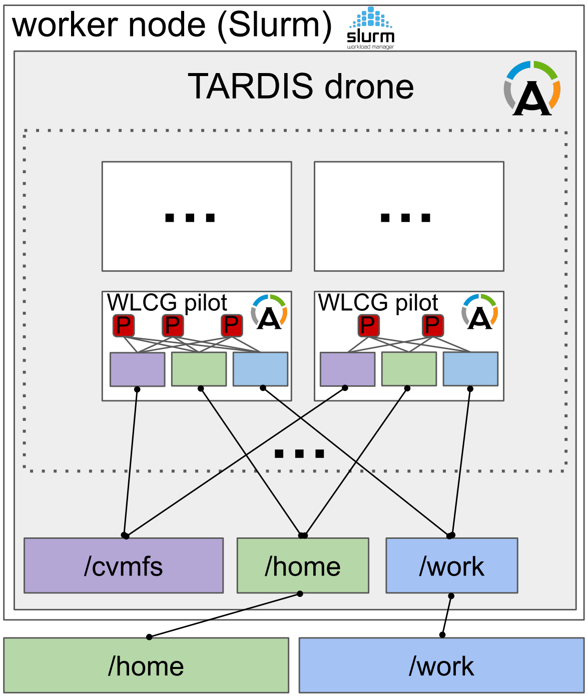


Figure 4.24.: Schematic of the HoreKa integration concept. With COBalD/TARDIS, a stacked container design enables the integration of the resource into the WLCG. The outermost layer is a drone spanning the whole node. It includes bind mounts of the home and work directories with the necessary configuration files and provides CVMFS via cvmfsexec. The bind mounts are then propagated to WLCG pilots that provide the actual environment for payload jobs – indicated in red – to run.

possible priority. Hence, when no other user is requesting the slots and they would be idling, they are assigned to it and integrated in the OBS. By this, it supports a better utilization of the cluster without blocking off other users.

When Slurm now schedules a whole node with a walltime of 20 hours to the TARDIS user, a drone is started in form of an Apptainer container, as visualized in Fig. 4.24. The `/home` and `/work` directories are available on all nodes and are included as bind mounts in the drone. They contain the site configurations and scratch space, respectively, as well as a local CVMFS cache for optimized performance in acquiring the software, containers, and configurations. Within the container, on startup, the environment is prepared via `Ansible`, a common automation and configuration tool. It installs the required software toolset and configurations for the integration, like HTCondor, for participating in the OBS, as well as CVMFS for the whole node via `cvmfsexec`, since HoreKa does not provide it natively. Inside the drone container is now everything available that is required to run official CMS job pilots and therefore to provide the resource to the WLCG. As indicated in the graphic, in the next step, within the drone container, up to twelve 8-core job slots can be published to the global HTCondor resource pool, accessible via GridKa's cloud CEs. Production job pilots are then submitted and start running within the drone container, labeled as WLCG pilot in Fig. 4.24. They inherit their bind mounts from the outer container which makes CVMFS available for all jobs inside.

Ultimately, within the pilots actual payload jobs then can start running, as indicated in red within the pilots. The definitive number depends on the requested cores by a job and can vary. This reflects one of the few requirements to run this setup for integrating resources opportunistically: `usernamespaces`.[23] Only with them enabled, a flexible stacking of containers to this extent is possible. The alternative would be to run the drones as VMs in which the container stack can be executed and jobs can run, but at the cost of a bigger overhead. At HoreKa, this is luckily not necessary, as the prerequisites are met.

The integration of HoreKa with COBalD/TARDIS shows that the framework is very flexible and can adapt to all kinds of different systems with different conditions. And overall, even if the chain of stacked containers seems quite complex, the abstraction and automation for an efficient integration of the resource is entirely done by the software framework. The resource provider only needs to configure it accordingly to match the conditions and policies, which is fairly simple. This is underlined by the fact that an initial integration of such an opportunistic resource can be done within a few hours.

And also in terms of operation, the system is lightweight and the maintenance effort is low for integrated resources, as the framework is well tested and established. Problems can typically be identified fast with the default WLCG monitoring, described in Section 4.3.3, even though the site monitoring capabilities are often limited. Locally at HoreKa, only Slurm logs for the whole nodes are available and further monitoring is restricted or at least not easy to access for automated systems due to the firewall and local restrictions, which is often the case in HPC environments. In addition, HoreKa is also integrated in the HappyFace4 instance, as described in Section 4.3.4, which helps to keep an overview over the opportunistic resources as well.

In summary, this section explained how HoreKa is currently integrated with COBalD/TARDIS to provide resources opportunistically for CMS in a flexible and automated manner. The integration leverages the Slurm-based HPC resource management, ensuring minimal interference with local users while maximizing the cluster utilization. With the opportunistic integration of HoreKa, significant compute power was provided on the one hand and on the other, valuable experience on the utilization of HPC resources was collected for the future of HEP computing, which will be addressed in the next section

[23] Additionally, cgroups v2 are recommended for additional Apptainer features, especially monitoring statistics and fine-granular configurations, but not mandatory.

5. The Pledged Integration of HPC Centers

> He had discovered a great law of human action, without knowing it – namely, that in order to make a man or a boy covet a thing, it is only necessary to make the thing difficult to attain.
>
> (The Adventures of Tom Sawyer – Mark Twain)

For industry as well as for HEP research, HPC resources are becoming more and more important. The resources provided by such centers are immense, impressively demonstrating their scalability. As an example, the two leading supercomputers in the Top500, El Capitan and Frontier together – place one and two as of November 2024 – provide more CPU cores [177] than the whole WLCG[1] [88]. And that is not even taking GPUs into account. Especially with regard to future scientific exploitations of GPUs, e.g. for reconstruction or simulation of HEP collision data, these centers offer incredible possibilities and development potential. However, their use in HEP is still in its infancy and the benefits cannot yet be realized on a larger scale.

The CPU parts of the HPC centers are already used, as shown with HoreKa and others, but their concept is typically not inherently suited for HEP workflows, as their compute models strongly differ from the classic grid sites forming the WLCG. While the computing concept of HPC centers favors mainly computations that rely on massive parallelization, all HEP jobs are trimmed to roughly stay within predefined, comparably low resource limits. Parallelism is then achieved by a large, horizontal scale-out of many parallel payloads running at the same time. Therefore, HEP workflows usually do not need the extreme parallelism but instead need to transfer enormous amounts of data for the physics research. For massively parallel HPC computations the internal data throughput can be enormous, but often little value is placed on the external connectivity of such centers – in the worst case even completely unavailable[2] – as large-scale remote data access is rather uncommon.

An integration of such resources is nevertheless possible with tools like COBalD/TARDIS, as described in Section 4.5.2 and Section 4.5.3, but this discrepancy can consequently lead to bottlenecks and limitations, especially with regard to the HL-LHC. At the same time, users of HPC clusters are expected to utilize these valuable resources responsible without wasting compute power, which can also lead to challenges for the HEP community with the integration of such resources.

[1] The monthly usage statistics are provided by CERN at: https://wlcg.web.cern.ch/using-wlcg/monitoring-visualisation/monthly-stats

[2] See e.g. Mare Nostrum at the Barcelona Supercomputing Center [178].

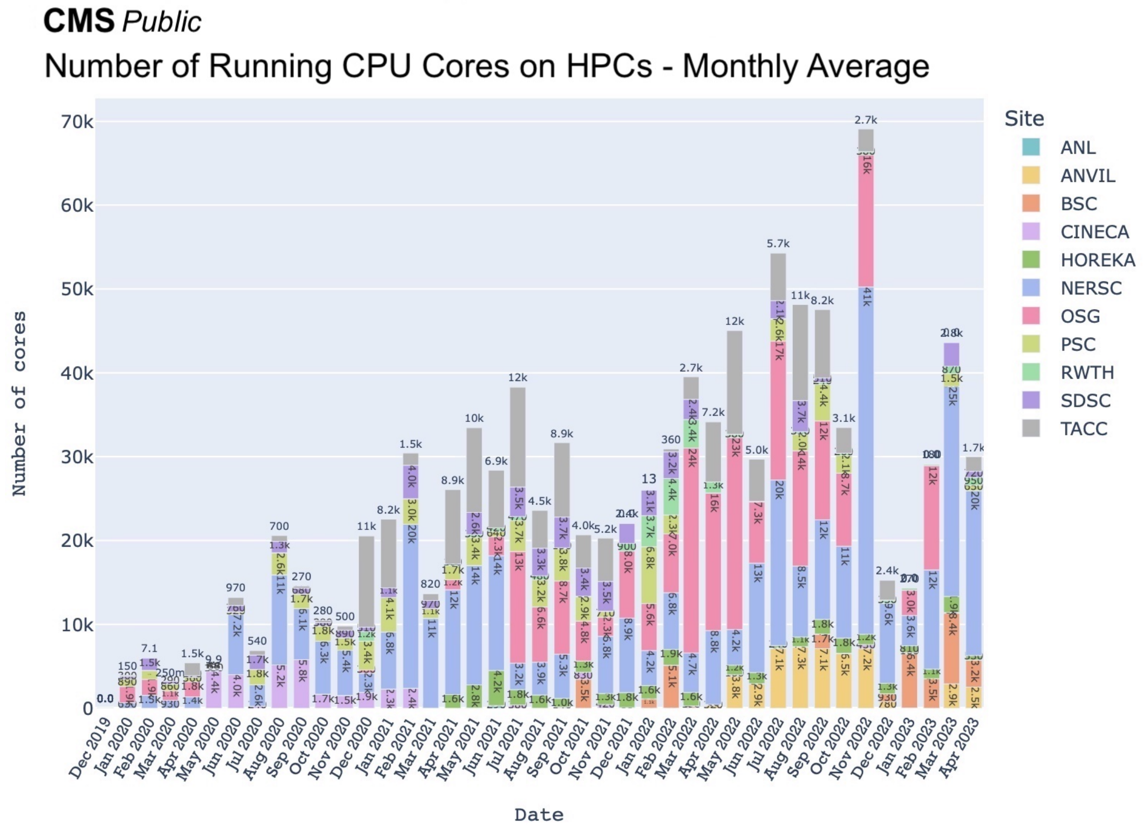


Figure 5.1.: The average number of cores integrated for CMS at different HPC centers. While the largest contributions come from scientific centers in the US, also two German HPC centers, HoreKa and CLAIX (RWTH), participated. *Source: CMS public results* [180]

But despite these obstacles, already today, the opportunistic integration of HPC CPU resources makes a significant contribution to the computing power of the WLCG. The average number of cores provided by HPC centers worldwide for CMS from 2021 to mid 2023 is shown in Fig. 5.1, where two German HPC centers (HoreKa and CLAIX (RWTH)[3] in the figure) contributed.

The opportunistic integration in terms of CPU time in 2024 is listed in Table 5.1. Perlmutter located at NERSC/Berkeley has by far the largest opportunistic contribution to the CMS resource pool and gives a glimpse on what is possible with such centers. HoreKa was providing the largest opportunistic HPC contribution from Europe.

In the future, the contributions by HPC centers will further grow, especially in Germany. Because in March 2022, a new HEP computing strategy in preparation of the HL-LHC era was introduced as a perspective paper by the Committee for Particle Physics[4] in Germany [182]. The new strategy has a strong focus on the increased integration of HPC resources to fulfill the German WLCG obligations. This represents an unprecedented change compared to the current WLCG computing environment and introduces questions and challenges on how pledges can reliably and efficiently be provided with this fundamental conceptual change.

The following sections address the pledged integration of HPC resources and dis-

[3] Details on the integration of CLAIX, the NHR center in Aachen can be found in Ref. [179].

[4] More information: https://www.ketweb.de/

Table 5.1.: Core hours opportunistically contributed to CMS by HPC centers. The Million Core Hours column describes the contribution until the end of November 2024. The Maximum Allocation is the amount that is provided as a maximum from the according resources during the Allocation Period. Although the big scientific US HPC centers have the highest contribution by far [181], an increased contribution in the next years is also expected by European HPC centers.

HPC Resource	Allocation Period	Million Core Hours	Maximum Allocation
NERSC Perlmutter	01.2024 - 01.2025	259.8	337.4
TACC Frontera	06.2024 - 05.2025	20.2	35.8
Purdue Anvil	10.2024 - 09.2025	9.7	23
HoreKa HPC	01.2024 - 12.2024	5.1	–
SDSC Expanse	10.2024 - 09.2025	4.4	23
PSC Bridges-2	10.2024 - 09.2025	1.1	23

cuss these questions. At first, the current situation including the German WLCG and HPC environment is described. In the second part, the new HEP computing strategy is explained in detail and the motivation and challenges associated with the changes are outlined. The third section will evaluate the experiences with the opportunistic integration of HoreKa with focus on limitations and challenges. And finally, optimization strategies for the pledged integration of HPC resources will be derived and discussed.

5.1. Situation in Germany

In 2024, Germany was one of the biggest resources providers in the WLCG responsible for around 8 %[5] of the worldwide CPU contributions for the LHC and HEP research. All participating institutions that reliably contributed to the official pledge are depicted in Fig. 5.2. Alongside GridKa, the German Tier-1 center, several institutional Tier-2 centers, marked in blue, and Tier-2 centers at universities, marked in orange, formed the German part of the WLCG. Tier-3 centers and opportunistic resources are not listed, as they usually do not contribute to the official obligations. In general, a distinction is mainly made between KIT, DESY, and GSI, which are all members of the Helmholtz Association of German Research Centers, and the smaller Tier-2 sites. The Helmholtz centers are the largest resource providers in Germany and together, they form the backbone of the German WLCG infrastructure by providing services and giving support to the other institutions.

5.1.1. The German WLCG Contribution

As mentioned, Germany is a very reliable part of the WLCG and provides around 10 % of its resources (including the opportunistic contribution). In the past years, the (CPU) pledges always have been over-fulfilled. The history of the German pledges

[5] With a total contribution of around 11 % including opportunistic resources [163].

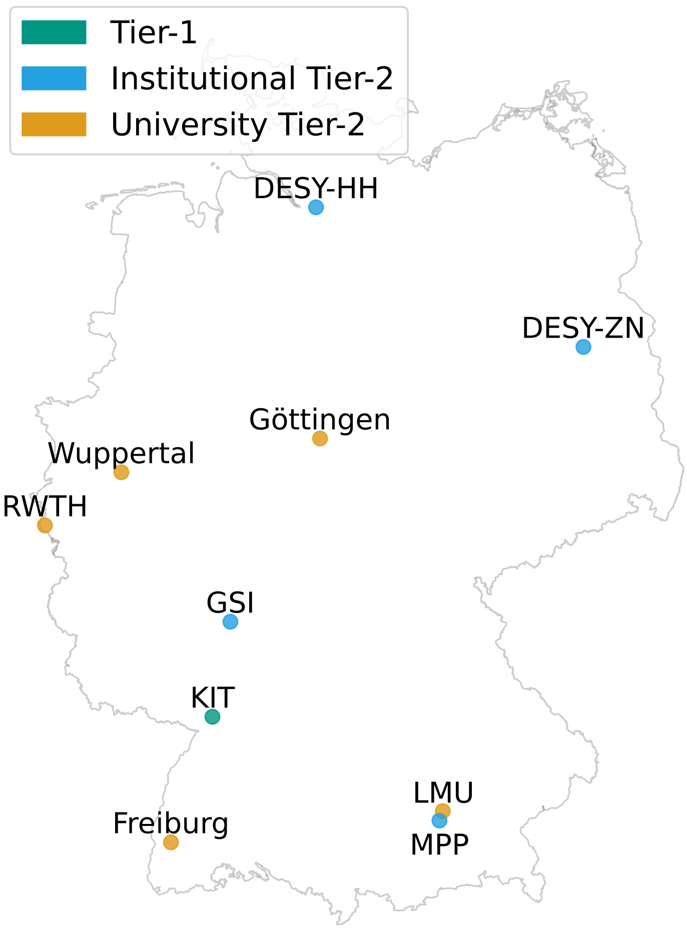


Figure 5.2.: Map of the German WLCG sites. Alongside the Tier-1 at KIT, two kinds of Tier-2 centers exist: institutional Tier-2 centers (blue) and Tier-2 centers located at university (orange). Additionally, several Tier-3 centers and other opportunistic resources exist which are not shown.

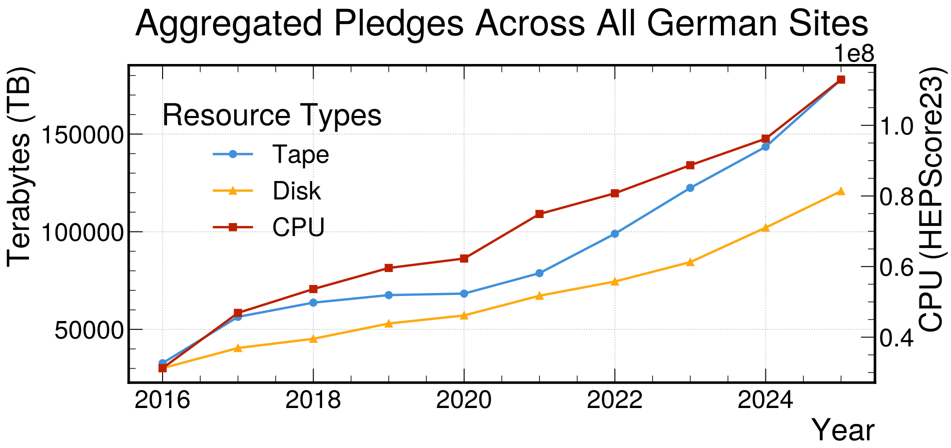


Figure 5.3.: The German WLCG pledges since 2016 aggregated over all VOs. Germany is a very reliable partner and has always (over-)fulfilled its pledges. It is particularly remarkable that since 2024, the German disk pledge has exceeded the magic mark of 100 PB. *Data source*: [89]

and their evolution is depicted in Fig. 5.3 and Fig. 5.4, respectively, including the pledges for the year 2025.

This shows that the aforementioned assumption of a 20 % increase of the resource demand every year (see Section 4.1) is well backed by the past development.

The pledge for the tape storage is fully covered by GridKa, as common in the WLCG, and also for the other resource types the Tier-1 provides the biggest part. Additionally, it is one of the leading sites in terms of reliability and availability [133]/WLCG SiteMon. As a multi-VO site, it provides resources to all LHC experiments. The distribution is shown in Fig. 5.5. Additionally, it also supports non-LHC VOs, such as Belle II, DARWIN/XLZD, Auger, and IceCube. On top of that, it provides petabytes of grid storage for analysis purposes to all German CMS members (dCMS) and supports other sites with different services, like Compute Element (CE)s. Computing R&D is also part of its responsibilities for the reliable provisioning of sufficient resources in the future.

The contributions of the other German sites is given in Fig. 5.6. Tier-2 centers typically have a fixed experiment affiliation and provide only resources for one experiment each. GSI is providing CPU and storage resources for ALICE and LRZ-LMU/MPI (Munich) for ATLAS. Only DESY, the other Helmholtz Center in particle physics alongside KIT, is an exception. It is the biggest German Tier-2 site and provides with a total pledge of 189’000 HS23 and 18.6 PB for ATLAS, CMS, and LHCb together roughly as much as the other Tier-2 centers together in 2025 [89]. This is not directly reflected in the diagrams, as the pledges are combined with other sites. The split can be retrieved from CRIC [89]. DESY also takes over additional responsibilities as it for example provides a CVMFS Stratum 1 (see Fig. 4.8) and is maintaining and developing dCache – a widely used storage solution within the WLCG.

The institutional Tier-2 centers are supplemented by the university Tier-2s. Wuppertal, Freiburg, and Göttingen are ATLAS Tier-2 sites and at RWTH in Aachen is an additional CMS Tier-2. The ATLAS sites will provide around 110’000 HS23 of compute and 8 PB of disk storage in 2025. At the same time, the contribution of the

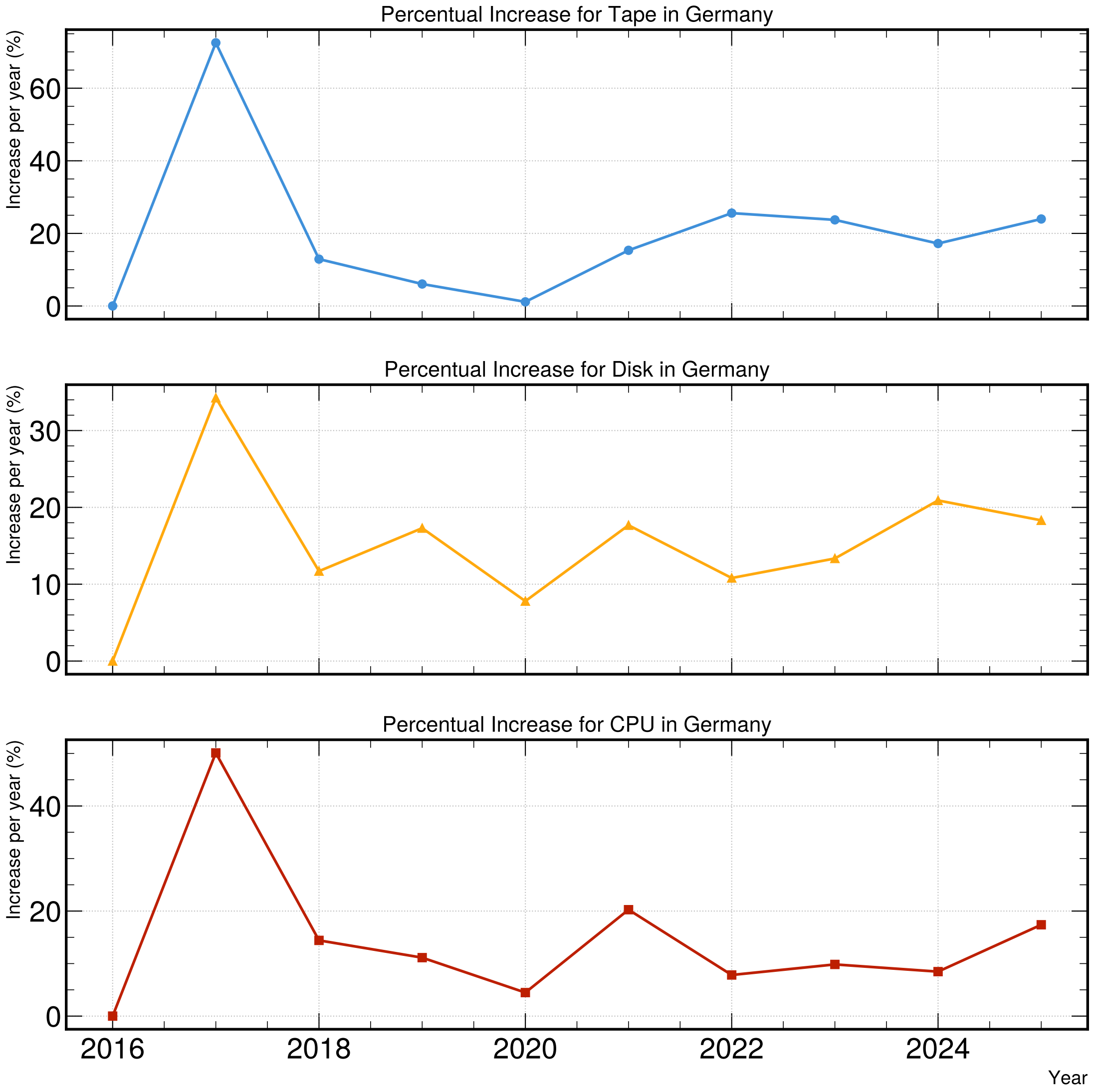


Figure 5.4.: The percentual evolution of the German WLCG pledges shown as relative increase compared to the year before. The data is only considering the pledges since 2016 which was chosen arbitrary for visualization and does not reflect the actual starting date. Therefore, the starting year of the graph (2016) is taken as baseline and shown with 0 % – not reflecting the increase in comparison to 2015. Overall, all three kinds of pledges were always increased between 10 % and 20 % per year. It is expected that with the HL-LHC, the demand will increase even faster to cope with the expected data rates and required compute performance. *Data source*: [89]

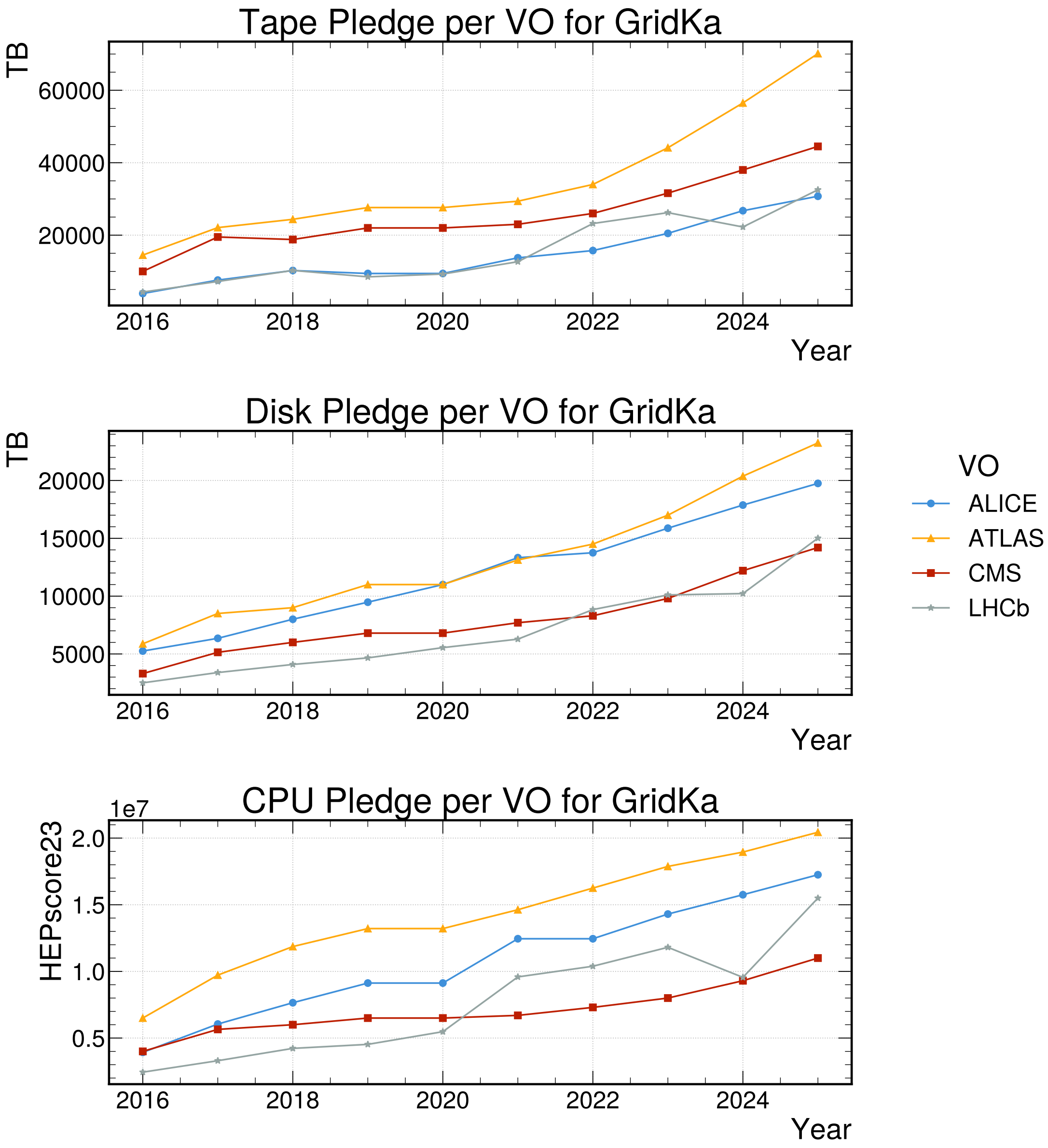


Figure 5.5.: GridKa pledges over time for tape, disk, and CPU. It is the largest German WLCG site and provides resources, services, and support to different VOs and other sites. The contributions are more or less linearly increasing every year with only a few exceptions and the pledges were always over-fulfilled. With the upcoming pledge for 2025, all pledges are nearly quadrupled compared to 2016. Additionally, several opportunistic resources are integrated and fed over the Cloud-CEs hosted at GridKa which are not included in these charts.
Data source: [89]

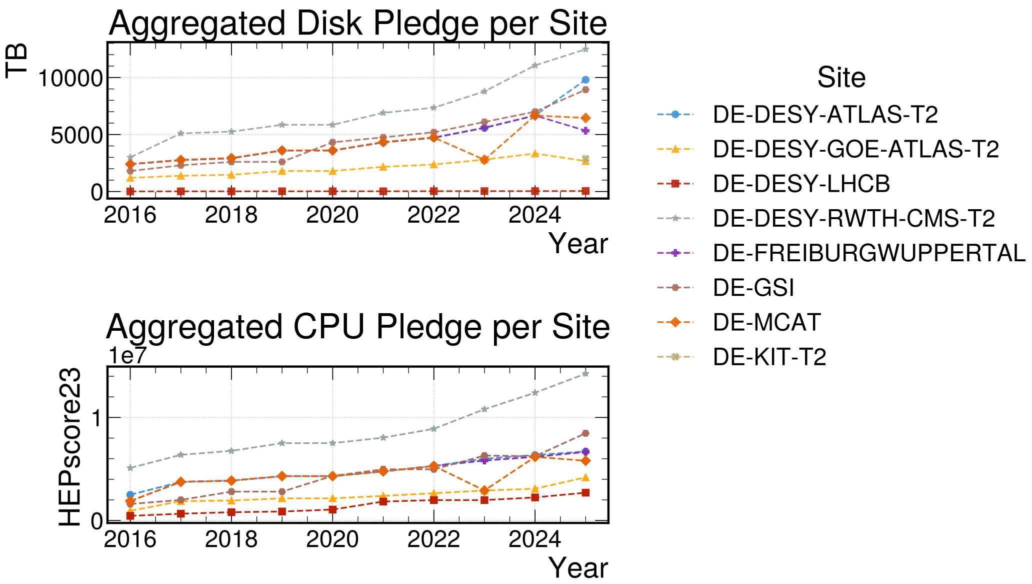


Figure 5.6.: The German Tier-2 pledges since 2016 for the different federations (sites). All of them provide CPU and disk storage for one VO, as common. Only DESY is a multi-VO Tier-2 center. Additionally, in 2025, a new KIT Tier-2 storage contribution is visible the first time. *Data source*: [89]

CMS Tier-2 in Aachen amounts to 47500 HS23 hours and around 3 PB of disk storage. As also can be seen in Fig. 5.6, in 2025, the new KIT Tier-2 storage contribution – as part of the future German HEP computing strategy – will provide 650 TB of disk storage for the first time, which will be discussed in detail later.
In parallel to the German part of the WLCG, another network of scientific resource providers exists that consists of several scientific HPC centers and is described in the next section.

5.1.2. The National High Performance Computing Alliance

The NHR Alliance is an association of nine German national scientific HPC centers located at different universities [183]. It introduced a joint coordination of procurement, operation, and further development of the computing centers to reduce duplicate structures and to make optimal use of the available resources. The annual budget of 62.5 million € [172] is provided by the German federal government and the participating states. The initial funding period is set initially to ten years, ending in 2030. A continuation of the already successful project is very likely. All members of the alliance are shown in Fig. 5.7.
Their purpose is to provide computing resources and support for scientists of divers research fields. In addition to the daily support, they also offer trainings and a Graduate School as well as workshops and symposiums for the exchange of experience between researchers of different scientific fields and computing experts. And through the shared use of the resources, the alliance aims to foster synergies between communities and provide a basis for collaboration and knowledge sharing.
The provisioning of resources by the NHR centers is free of charge for researchers associated with a German university and based on a standardized application pro-

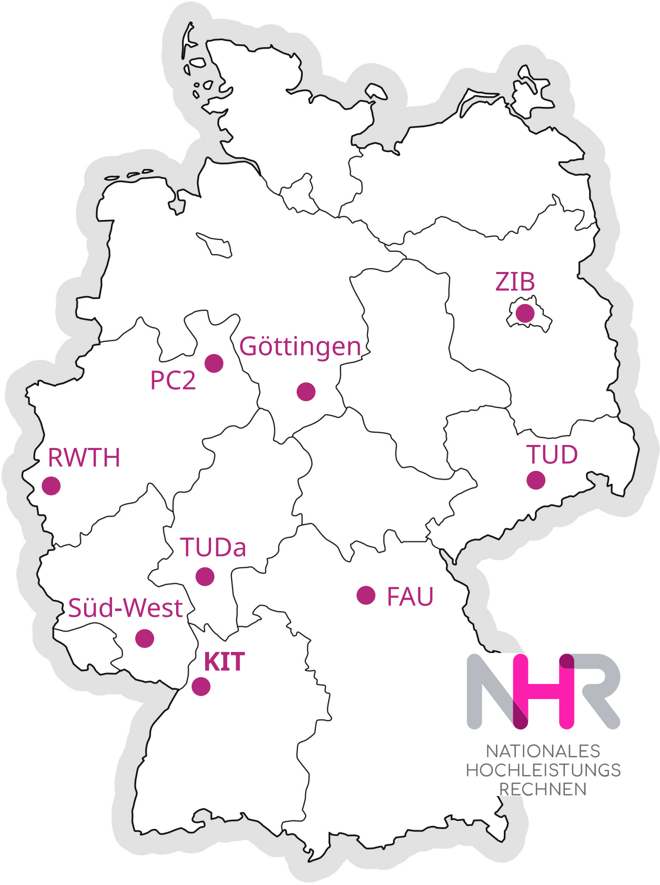


Figure 5.7.: The NHR Alliance involves nine scientific HPC centers in Germany. Some of them are collocated with WLCG grid sites, mainly Karlsruhe, Aachen, and Göttingen.

cess [176]. Alongside direct project proposals for testing and development (500k core hours), also rolling calls and quarterly calls are available that comprise up to 10 million or 30 million core hours, respectively. The usual project duration is one year with the option of a simplified renewal.
With the NHR Alliance, Germany is strengthening its position in international research projects on top of the contribution to the WLCG for the LHC experiments, as HPC infrastructures have become indispensable in many areas of research.
And also in the upcoming HEP computing strategy in Germany, they will gain importance as described in the following section.

5.2. The Future German HEP Computing Strategy

Overlapping with the end phase of Run 3, a new era for the German HEP computing will start in 2025. As current trends, like e.g. artificial intelligence and GPU computing, have also found their way into particle physics, while the question of the environmental impact of research is becoming increasingly important for science in general, HPC resources are becoming more and more interesting for the HEP community.
This is reflected in the future German HEP computing strategy, developed by the Committee for Particle Physics in Germany [182]. While the classic Grid Computing model has proven very successful over the past 15 years, innovation and collaborative developments are more important than ever to meet the increasing resource demand of the HL-LHC era starting in the beginning of the 2030s. In order to exploit synergy effects with other disciplines and to profit from modern technologies, HEP in general and the WLCG computing in particular must therefore open up further to new ways. A special focus here lies on the most efficient and environmentally friendly provisioning and utilization of computing resources. This is required due to the social responsibility of scientific research which is mainly financed by public funds. Furthermore, it is also very important as the WLCG is a pioneer in many respects and has a major influence on computing in other scientific fields.
Because of these arguments, a consolidation of resources in Germany was decided.

5.2.1. Consolidation of Storage Resources

While the initial, hierarchical concept of the WLCG is based on a wide spread of resource providers forming a fine mesh around the world, the new HEP computing strategy foresees a more central approach on national level. This is also supported by the current WLCG consensus [182]. It suggests to consolidate resources – especially storage that should be provisioned in a data lake model – which is expected to have advantages in the future.
A first reason is that on the one hand, it is certainly an advantage to achieve a large distribution of data with local centers that accordingly provide sufficient compute resources that can benefit from the resulting data locality. But on the other hand, the share of the total disk storage at the smaller university Tier-2 centers is fairly too small, as shown in Fig. 5.3 and Fig. 5.6, to actually make a bigger difference on the long run. The consequence is that the overhead for buying and maintaining such storage resources with a small team at university Tier-2 centers may outweigh

the benefit of the data distribution. Especially, when they are, for example, used to produce MC samples, which usually requires to access PREMIX datasets remotely at CERN or other Tier-1 centers that are also capable of storing parts of these huge and heavily accessed datasets. And this effect will be amplified by the expected increase in the to be stored data with the HL-LHC. Eventually, this will reduce the number of distributed copies of datasets, but an increased quality of service is planned [182]. The storage space gained from the reduction of replications could actually be necessary in the future to store all the data for the next Run.

A second reason is that the effort to raise the share for a smaller center is significantly higher than for one of the bigger ones, like KIT or DESY, which could push smaller centers to the edge of feasibility over the next years with constantly high increases expected.

For example, the disk pledges for CMS in Germany will increase from 23 270 TB in 2024 to 26 025 TB in 2025. If the CMS Tier-2 in Aachen with a disk pledge of 3690 TB in 2024 has to raise its share in 2025 by 650 TB, this would corresponds to an increase of around 20 %. If GridKa takes over this pledge, the additional increase will be of only around 5 %[6]. Or more visually spoken, if a site has one rack with servers and needs to maintain another one, the effort is relatively higher as for a site with 20 racks that gets one more. And also this effect would be amplified by the expected increase of the pledges in preparation for the HL-LHC over the next few years.

A third reason for the consolidation of resources is the increased efficiency in an ecological and economic sense. Maintaining multiple data centers, all of which require space, cooling, and technical staff is less efficient than consolidating them into fewer centers with higher capabilities. And the investment and operation costs are comparably lower for larger sites, as smaller ones may not receive the same procurement discounts and need funding for additional dedicated personal. Especially with respect to the increased demand in the future, also the scalability and sustainability of larger resources is much better than for several smaller individual resources. Because, for example, better cooling technologies at fewer, larger centers are more capable and energy efficient than the classical air cooling usual for smaller sites.

Consequently, the new German HEP computing strategy foresees a gradual transition of dedicated storage resources at university Tier-2 centers to the Helmholtz Centers, KIT, DESY, and GSI. The centralization of the storage resources in a data lake model will guarantee an ongoing, reliable, and efficient provisioning of the German WLCG obligations through the HL-LHC era. The start of the transition can be viewed in the top figure of Fig. 5.6, where the first disk pledge for the newly formed KIT-T2 (DE-KIT-T2, gold) is present for 2025, marking the beginning of the transition. Additionally, also DESY will increase its share accordingly. The transition is planned to be completed in 2030, as show in Fig. 5.8, for the start of the HL-LHC. The full schedule is depicted in Fig. B.6.

The second part of the new strategy concerns the compute resources.

5.2.2. Transition to Pledged NHR CPU Resources

The second fundamental change concerns the CPU resources located at university Tier-2 centers. Just like the disk storage, following the new strategy, no new CPU resources should be acquired at those centers. What is different, however, is that

[6] The impact, however, is even lower if the total disk pledge for all VOs is considered.

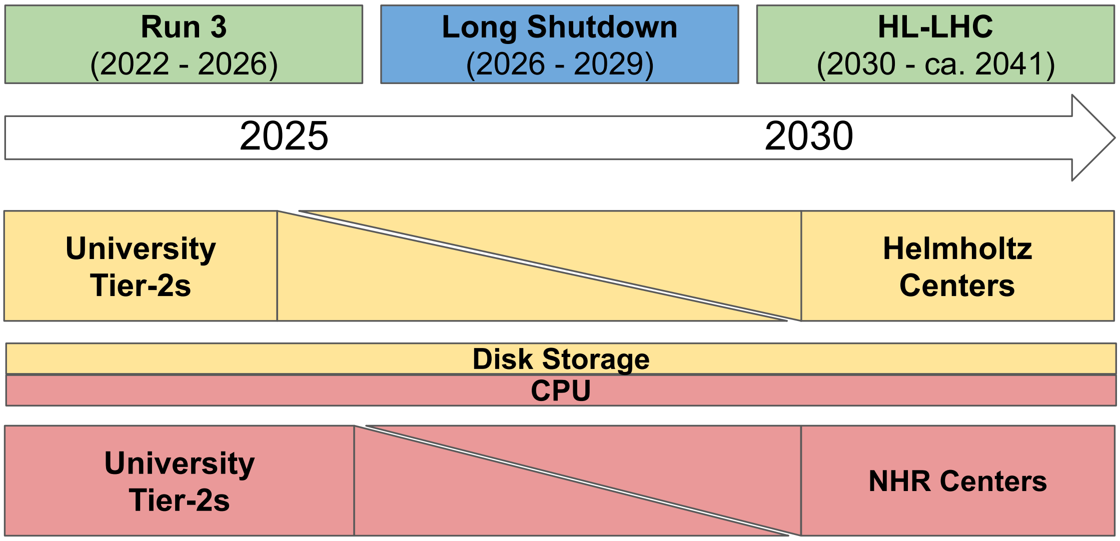


Figure 5.8.: Visualization of the German HEP computing strategy for the next decade, introduced by the Committee for Particle Physics in Germany (KET). Gradually, WLCG Tier-2 storage resources located at universities will be transitioned to the Helmholtz Centers, starting in 2025. The associated compute resources (CPU) will be gradually replaced by shares on national HPC centers within the NHR Alliance that are co-located to the WLCG sites. This process is in preparation and will start end of 2025 or beginning of 2026.
Adapted from: Markus Schuhmacher/[182]

an official part of the compute pledges will also be covered by shares on national HPC centers of the NHR Alliance, which is an absolute novelty in the WLCG. This is realized as a gradual transition starting in 2025. While HPC resources were used opportunistically by the experiments for several years, as described in Section 4.5, a mandatory contribution of CPU time by such centers is completely new terrain and requires an adequate adaptation to the new circumstances. Therefore, a lot of time and funding has been invested in Germany for research and development over the last few years to investigate the endeavor and make tools like COBalD/TARDIS ready for such a profound step.

Important to mention here is that the idea behind this change is **not** to replace the current WLCG Tier-2 centers at universities entirely. The transition only means that the provided resources are not hosted by the Tier-2 teams anymore. The responsibility for the obligations stays with the current groups. Therefore, three NHR centers were chosen that are co-located to the university Tier-2 centers. This has the advantage that both, geographical proximity as well as personal closeness between the Tier-2 sites and the HPC staff are present. This is important to realize the project as direct communication and support from the actual resource providers are obligatory for a successful transition and reliable operation in the future.

This future mode of operation then foresees that the Tier-2 in Göttingen will request resources at EMMY, the local NHR center there, Aachen will move to CLAIX, the HPC center on site, and Freiburg will request resources at HoreKa, the NHR center at KIT. The new tasks of the local university teams then consist of applying for and integrating the resource shares, as well as monitoring the operation and solving problems to ensure a reliable resource provisioning for the WLCG. They are in direct contact with the NHR centers and contribute their specialized knowledge in the field of HEP computing for the smooth and efficient utilization of the resources. And in addition, the discontinuation of the old hardware and reduced obligations can free up new capacities for research and development in order to be optimally prepared for the future requirements of WLCG computing in the HL-LHC era.

Furthermore, an advantage of this concept is that the application is rather flexible in comparison to dedicated hardware. If strategies, plans, and obligations change on national level in a shorter time period than the typical hardware life cycle of at least five years, the new concept is more adaptable. And also, as already mentioned, the pledging of high-efficient HPC resources is ecologically preferable compared to running older hardware with own cooling systems locally to reduce the overall climate impact. When additional resources are required on shorter scales, chances are high to get extended shares faster than new hardware with the rather uncomplicated NHR assessment process instead of lengthy and complicated project funding applications.

Additionally, another benefit of the new strategy could be that in the future, such centers will also be able to provide other resources, like GPUs and accelerators. The advantage of this over having own dedicated accelerators is that – in addition to greater sustainability and reduced maintenance effort – it provides on-demand provisioning, which prevents resources from being idle and thus achieves a better overall utilization. Obviously, this may also mean that resources are not immediately available, but with flexible resource pools managed by COBalD/TARDIS and based on HTCondor, this is not necessarily a disadvantage, because they are just allocated when becoming available, as mentioned earlier.

The above discussed ideas show that the new German strategy is a valuable addition to the classic Grid Computing model and that the HEP community is gaining in flexibility and innovation, which is extremely important for the future.

The transition process is planned to start later the same year as for the storage resources, as indicated in the bottom part of Fig. 5.8. Finally, when the transition is finished around 2030, the new, extended computing model will be well established, tested, and ready for the enormous demands in the HL-LHC era. But in general, also for the initial pilot phase no major complications with the new strategy are to be expected with the provisioning of resources.

For the first pledged NHR contribution, a share of 4.8 million core hours will be requested for CMS. This is of the order of the opportunistic contribution of HoreKa in 2024, see Table 5.1, and should therefore not pose a major challenge. For ATLAS, an application at EMMY and HoreKa for approximately 37 million core hours is foreseen. This is significantly more, but as a scale test with more than 38000 cores [184] simultaneously allocated at EMMY has shown, this can be handled with the comprehensive toolset available for the integration.

5.2.3. Summary

With the transition of compute and storage resources from small university Tier-2 centers to the larger Helmholtz Centers and national scientific HPC centers within the NHR Alliance, Germany's new HEP computing strategy is breaking new, unknown ground. Especially the pledged integration of HPC resources reflects a novelty in the WLCG computing community. With the opportunistic integration of HoreKa, described in Section 4.5.3, and other prototypes has been shown that HPC centers can be used for running HEP workflows in principle, despite the inherently different computing models. However, this does not yet guarantee an efficient utilization or the suitability as an official grid resource with the corresponding challenges, which are addressed in the next section.

The Committee for Particle Physics in Germany as well as the HEP computing community are convinced that in the future, the official use of HPC resources and the consolidation of storage resources will be beneficial. It will help to achieve a better energy efficiency and to enhance the utilization of available resources, which both will reduce the ecological impact of particle physics research and increase the sustainability of the computing model in general [182]. Furthermore, the collaborative use of the NHR resources is expected to foster synergies between different research fields [182] and may also help others to benefit from the advanced toolset and experience of the WLCG community.

In addition, the diversification of WLCG resources that comes with the new strategy has the potential to improve the capabilities and flexibility of the computing infrastructure. Because expanding it with new, beneficial technologies for the HL-LHC era, while reducing cost and effort and freeing up more resources for research and development at the university Tier-2 centers will help to adapt to growing demands in the future. Here, it is again emphasized that the planned transition is not intended to entirely replace the Tier-2 centers at local universities. It only shifts the focus and tasks from the sole resource provisioning to a more research-oriented assignment of tasks, as described above.

Finally, with regard to the initial phase, no major complications for the new strategy are expected, as described in the previous section. It is therefore highly likely that the strategy will pay off, despite the challenges associated with the pledged integration of HPC resources, which will be discussed in the next chapter.

5.3. Discussion of Challenges and Limitations

The utilization of HPC for HEP computing as basis for the future strategy is possible as described and shown with the opportunistic integration of several such centers into the WLCG. For a pledged integration, however, significantly higher requirements apply, both from the WLCG and from the resource provider side.

From the LHC perspective, especially the reliable provisioning and operation of the integrated resources is important. This includes that they need to be available when needed and fulfill all necessary prerequisites, like containerization, software, or outgoing Internet connections, to allow an efficient execution of HEP workflows. In other words, everything that is normally regulated by an MoU. Furthermore, there is a firm commitment by the WLCG to using resources in the most environmentally friendly and efficient way possible. This is exactly what is also demanded by the other side. The NHR centers are based on proposals and free of charge. Consequently, each project that takes care of the pledged integration of one of the centers will be evaluated after the first pilot phase. Because only if an efficient utilization is guaranteed, both sides benefit from the cooperation, with the different compute models forming beneficial synergies[7]. A renewal of the compute budget

[7] In addition to the potential benefits of an opportunistic back-filling of HPC resources, as depicted

therefore also depends, at least to some extent, on how well the provided resources were utilized by the applying groups and with it the LHC community. Therefore, the experiments are responsible for designing workflows efficiently so that neither energy nor computing time is wasted.

In general, the reasons for suboptimal use of such resources can be manifold. They can be influenced by a variety of factors, some of which are within one's control, while others, caused by external circumstances, may be unchangeable. Some challenges and limitations that are relevant in this context are now discussed.

Differing Operation Mode A first challenge of the pledged integration of HPC is the changed operating mode of the resources. When it comes to the reliable provision of computing resources, there is a big difference whether one provides dedicated resources and has access to the hardware or if those resources are requested from an external, independent institution and are therefore out of direct control.

On the one hand, this offers an advantage, since one is not responsible for the resources and can in general assume that an extremely reliable operation of such large HPC centers within the NHR Alliance is given. Furthermore, experts take care of maintenance and service recovery in case of problems. But on the other, the price that has to be paid for this is a significant reduction in access rights and general flexibility, while still being fully responsible for the provisioning. Because the transition effectively means a change from administrative `root` access to normal user access without special rights or permissions, which eventually depend heavily on the security policies and environments of the centers and can differ significantly. Meaning, the WLCG community becomes one equal user among many using the the NHR resources and – especially in the pilot phase – a comparably small one. This must be taken into account for the pledged integration. The negative effects are largely mitigated by the complete containerization and the comprehensive software toolset in form of COBalD/TARDIS for the integration (see Section 4.5.2), but there are still disadvantages and limitations for the operation and provision of resources.

Hardware and Software Prerequisites First of all, to actually realize an integration, the prerequisites must be met, like e.g. the availability of `usernamespace`, or alternatively VMs, as discussed in Section 4.5.3.2. After that, the WLCG hardware requirements for HEP jobs should match the hardware of the pledged HPC centers as close as possible from the outset, since there is neither direct influence on it nor the ability to change anything, which needs to be considered in advance. The same applies to the network connectivity. Although it is not absolutely necessary for the worker nodes to be connected to the Internet, as the integration of Mare Nostrum at the Barcelona Supercomputing Center impressively demonstrates [178], it is highly recommended, as otherwise operation is significantly more complicated, less flexible, and possibly also less efficient. Another consequence could be the loss of automation if no external access to the resource is allowed, which significantly increases the effort for the integration, because, for example, the compute nodes need to be requested manually and the drones may need to be started by hand.

in Fig. 4.6.

Connectivity and Bandwidth Furthermore, ideally the connection should not only be available but also fast enough to handle the huge amount of data in HEP. Otherwise, insufficient throughput could lead to bottlenecks and inefficiencies in data processing and simulation. This is particularly dangerous at HPC resources, since their focus is rarely on high throughput, but rather on parallel performance. Here, the required data is typically transferred in advance, because low-latency access is often required, which is hardly realizable with remote data. Therefore, the ATLAS computing model, which relies more heavily on pre-fetching of the required data before running a workflow, is less affected than CMS. But for both, this has to be considered in advance to avoid hardware limitations, since there is no direct influence on that.
To make it even more difficult, the Tier-2 centers have an MoU, but this is not automatically transferred to the NHR centers. This means that if there is no direct connection with a firewall bypass between them, they will not have direct access to LHCONE and the firewall will need to be considered as an additional limitation, as remote data access needs to be handled via the public Internet.

Reduced Permissions Furthermore, the change to a normal user also means a reduction of permissions in the operational sense. This is reflected, for example, in limited monitoring options due to security restrictions and reduced possibilities for software-based optimization of the integration. Essentially, everything needs to be executable in containers (or VMs need to be provided), since no additional software can be installed bare-metal – at least not without an agreement with the resource providers. As a consequence, low-level monitoring data, such as node I/O or hardware metrics, is often not accessible, which can make an efficient operation more difficult. A reliable operation, however, is normally still possible with the available payload monitoring via CERN MONIT, which is usually sufficient for the identification of potential error sources and their correction. But the identification of bottlenecks and limitations on lower level is much more complicated without the additional information that would provide a more complete picture of the situation. The only possibilities in such a case are dedicated benchmarking studies, or a comparison-based evaluation, as shown in the next section.

Different Provisioning Modes Additional to the operation mode, also the provisioning itself is different between HPC resources, often based on Slurm, and classical grid sites, typically based on HTCondor (for CMS, see Section 4.2.3.1). According to the pre-planned scheduling principle common for massively parallel environments, in contrast to the dynamic allocation within the WLCG, resources are typically made available in bunches, see Section 4.2.3.2. Consequently, depending on the schedule, at a certain time maybe 500 cores are allocated, later 3000, and the next day none.
This is inherently different to the provisioning mode of classic WLCG sites and not ideal for Grid Computing, for which a rather constant provisioning is desirable to achieve a balanced allocation across many sites. Therefore, the grid sites typically provide a rather constant and comparably small resource pool over a long time. However, this in turn is rather unusual for HPC and therefore unlikely to achieve in an agreement with the NHR centers. What this shows once again is that the computing models do not fit optimally and compromises are necessary to be made. Because insisting on it from the WLCG side would undermine the synergy effects and could

lead to a worse allocation, since the flexibility of HEP computing with its dynamic resource pools is a good complement to the rather rigid HPC concept. And at the same time, it can be assumed that the centers certainly would not fundamentally adapt their concept for the HEP community.

Another side-effect of this difference is that due to the performance potential and capacity of the NHR centers, which is many times greater than that of the classic university Tier-2 centers, the entire pledge of such a Tier-2 could theoretically be provided immediately in the most extreme case. That is of course not intended, but can fortunately be regulated with tools like COBalD/TARDIS. The framework allows, e.g., to just request a maximum number of parallel nodes at the same time, however, naturally without a guarantee that these will be provided constantly.

But ultimately, this should not lead to greater problems as long as a good allocation is achieved and the obligations are met on the long run, because this is what counts in the end. Additionally, the effects are also mitigated by the OBS setup with TARDIS. It integrates the resources transparently behind one single point of entry (see Section 4.5.2) and the allocation happens in a fair-share principle. Consequently, when multiple of such resources and additional opportunistic resources are combined in a resource pool, the chances are high that there will always be something available somewhere for the experiments.

Strategically, if one large pool of such resources for a joint ATLAS and CMS computing compound may be decided in the future, this should definitely no longer be a problem. On the contrary, then, the flexibility and elasticity resulting from the integration of high-potential HPC resources and their massively parallel provisioning model could even become a great advantage for the LHC computing. Because in phases of increased demand, the concept allows the resource request to be dynamically scaled up to compensate for high-pressure phases, somewhat like on-demand Cloud Computing, as described in Section 4.2.3.3. Something like this is not possible with a classic WLCG Tier-2 center, which is mainly limited in hardware.

And with TARDIS, this can even be done fully automatically on large scales in dependence of the demand in the queues and the current allocation of the available resources. Of course with the advantage that they can then be scaled down again elastically, which is not as easily possible with dedicated resources. Therefore, this challenge may become a feature that could be very interesting for the HL-LHC era.

Official Agreements Furthermore, it has to be mentioned that at the time of writing this thesis, yet no official long-term agreement for the allocation of compute time for the WLCG groups at the NHR centers exists. All ongoing prototype integrations of HoreKa, CLAIX, and EMMY are realized with agreements between the local groups and the co-located HPC centers or project-based applications. Therefore, the gradual transition is chosen, as visualized in Fig. 5.8, even if a direct migration would be possible, to ensure a smooth process until the start of the HL-LHC. Because on the one hand, the direct discontinuation of the available Tier-2 resources would be uneconomical and on the other hand, the continuous process will allow the evaluation of the concept and the finding of practicable long-term agreements that are able to secure the German WLCG obligations in the future. But for the initial phase, there should be no problem, as already discussed above in Section 5.2.3.

Storage and Data Localization Another currently unresolved challenge for the pledged integration of compute resources from NHR centers arises from the transition of the storage resources. The usual WLCG computing model foresees a distribution of workloads in dependence of the locally available datasets. This is obviously not possible anymore, when the university Tier-2 storage resources are fully migrated to the Helmholtz Centers. Scheduling based on data placement is therefore only of limited practical use with a national data lake concept. And for the NHR resources, typically no large-scale mass storage is available anyway and the longer-term archiving of datasets at the centers is not intended. The consequence for both would be that the sub-sites do not get any work anymore without any future adaptions of the workload distribution concept. And on top, such sites could get problems with the stage-out of processed data, which happens normally (but not necessarily) to the attached grid storage of a site from where it is redistributed further.
Both challenges are no game-changers, but require more low-level changes beyond the scope of a single community. Such conceptual adjustments need to be decided globally, since standardized approaches are of outmost importance in a worldwide computing grid, as already explained previously. In general, there are already concepts available for solving both problems. For example, volatile storage resources, like for example caches at the existing Tier-2 locations, or concepts such as virtual data placement based on Rucio, which is already used by ATLAS today. In any case, it is clear that further research, development, and agreements are needed here.

This discussion shows that the pledged integration of HPC resources in cooperation with the German NHR centers is a complicated undertaking with multiple challenges and limitations. However, over the last few years, reliable solutions have been devised and developed through constant research and development in the WLCG computing community. As a result, there are already ways of mitigating many of the challenges and limitations, but some still have no concrete solutions and require further investigation to ensure an efficient and reliable utilization of the pledged resources in the future.
But at the same time, this discussion also shows that the strategy has been chosen carefully and is of great potential for the future of German HEP computing.

5.4. Goal for the Pledged Integration

The previous integration of HPC resources, like HoreKa, in Germany was fully opportunistic. HEP jobs were mainly used in a back-filling manner with the lowest priority to utilize idle resources, as described in Section 4.2.3.2 and Section 4.5. With this setup, it was ensured that resources are only used when no one else is requesting them. The minimal target in this case is that the integration works reliably and HPC resource can be employed for HEP computing tasks, as previously described in Section 4.5.3.[8]
In such a scenario, efficient usage is of course desirable – simply to meet the self-set aspirations. But a less efficient utilization is not actively harmful either, as these jobs do not steal CPU time from other users, but utilize unallocated resources. To go even

[8] Note: The transition of storage will not be further targeted within this work but since KIT and DESY are already large and reliable storage providers for the WLCG, there are no major challenges expected.

further – contrary to intuition – a not fully efficient utilization still can be beneficial even from the sustainability point of view. Because the idle power draw of modern servers in HPC centers is typically optimized in comparison to older hardware, but still, it has to be accounted as a full loss, since it is invested energy without any usable outcome. On top, in the full calculation also the overhead for power supply, storage, network, and in particular the cooling systems need to be taken into account. This leads to the following considerations when judging the inefficient integration: Sub-optimal usage additionally increases the overall energy consumption, but at least provides a usable outcome. The processed data per invested energy is then possibly not ideal, but still better than none. In total cost accounting for the resources, to not use them would therefore be even worse – of course this ultimately depends very much on the actual circumstances.{9} The goal of the opportunistic integration can therefore be formulated based on these considerations as: *Make the most out of the energy that otherwise would be wasted.*

This argumentation, however, must of course not to be seen as a justification for the inefficient or misuse of valuable compute resources! It is merely intended to show that when formulating the goals of the integration, it could be necessary to think outside the box in order to find the global optimum, which of course must integrate all participants. Because for example, the inefficient usage of a more (energy) efficient resource can still be more sustainable than the other way round, as will be further discussed in Section 5.5.1.2. The only thing that is actually wasted in this case is the potential, because it could have been better.

With the new HEP computing strategy, explained in Fig. 5.8, this changes significantly. Because the circumstances for the pledge integration of the NHR resources are entirely different. The transition to official shares goes naturally together with higher expectations, because in this situation, the HEP community becomes a normal user like all others with the same priorities and same responsibilities. As a consequence, resources may be occupied that in principle can be contested by others. With this, the situation changes. Because the inefficient utilization of a given resource manifests a real loss in this case, when someone else could use them more efficiently at the same time. Then, it is not only lost potential anymore, but an actual waste of resources in the sense of sustainability, which definitely has to be avoided.

A well functioning integration with a high reliability and efficient utilization of the resources therefore becomes more critical with the pledged integration and the expectations rise. In a perfect world, the goal for the utilization would obviously be to use the resources ideally to avoid the above described scenario and the associated resulting loss. However, the optimal result is de facto not achievable in reality and this goal is doomed to failure. Too many limiting factors that cannot be influenced can degrade the result, as previously discussed in Section 5.3. When formulating a concrete goal for the pledged integration, this needs to be taken into account.

{9} As an alternative to the opportunistic usage, one could argue that a shut-down of unused servers would be more efficient than a non-ideal utilization. What is correct is that the energy consumption would indeed go down in this case. But in turn, this would increase the relative overhead. On top, the complexity and effort in large-scale HPC resources are way too high for a regular shut-down of resources. A more feasible alternative may could be an under-clocking to save energy. But the saving potential is significantly smaller and it is in question if the saved energy would be high enough to justify keeping valuable resources unused. Nevertheless, it could definitely be worth of further investigation, e.g. in combination with the availability of renewable energies – or the lack of power budget, in contrary.

As an example, the average CPU efficiency of all resources in the WLCG utilized by CMS is shown in Fig. B.7 (for a limited time-period). From this, the target in terms of CPU efficiency can already be reduced to 80 %, as it is very unlikely that the expected result will turn out even better than the rest of the grid. Here, the experiments are in demand to work on optimizations as this is out of scope of the grid sites and limitations must therefore be accepted. On top, possible additional losses depending on the job mix also need to be tolerated, as some job types are significantly less efficient on average. This is also clearly observable in Fig. B.7. While definitely not ideal from the resource provider's perspective, this always has to be accepted – but also kept in mind.

As a consequence, the formulation of the goal for the HPC integration needs to be reduced to: *Make the most out of the available resources without wasting more energy than what could not be avoided.* This dilutes the target, but still reflects that the resources should be used responsibly and in the best possible way – in consideration of the circumstances. What is best, however, is still hard to tell, since one never has a complete picture of all the complex systems working together, preventing an absolute definition.

The best way to circumvent these hurdles is therefore to set the goals of the pledged integration in relation to the existing, dedicated WLCG resources, which the HPC resources are intended to replace. From this perspective, the goal for the transition can be formulated more concrete as: *The NHR resources should perform comparably well as the WLCG sites – or better, of course.*

One last thing to mention is that the time until the full replacement of the university Tier-2 centers by HPC resources of the NHR Alliance is still considered a pilot phase. The official integration of HPC into the WLCG is absolute bleeding edge and therefore not expected to perfectly work out-of-the-box. Further research and development will be conducted to ensure a responsible and best-as-possible utilization of the given resources when the HL-LHC era arrives.

In summary, the transition to HPC resources is a mayor, but necessary challenge for the HEP community. A pledged integration will be a novelty and measured against the classic grid sites. The final goal, after steadily increasing the share on the NHR resources, is therefore to achieve a reliable operation comparable to the WLCG sites, while exploiting the enormous potential of the HPC resources as good as possible.

The next section is dedicated to introduce appropriate and meaningful evaluation measures that allow to reliably evaluate how well a grid site is performing, which is essential for assessing wether the goals for the integration have been achieved.

5.5. Performance and Reliability Evaluation of a Grid Resource

In preparation for the pledged integration of NHR HPC resources, a look at the opportunistic integration of HoreKa is valuable to evaluate and assess the impact of the challenges and limitations described in the previous chapter. As a reminder, HoreKa has been available as an opportunistic grid resource since its commissioning in 2021. The full cluster setup is shown in Fig. 4.23 and the integration concept is described in detail in Section 4.5.3. Since the center is only used opportunistically by the HEP community until today (early 2025), there does not yet exist an agreement

on a fixed share and it is only providing compute resources when available in a back-filling manner. Accordingly, just as described as a challenge above, there is no constant provisioning of a minimal amount of cores running at a time.
Nevertheless, valuable data was collected on the integration of HEP into the WLCG with this project. Even without full coverage and gaps in the data depending on the allocation of resources, the data can be used for a basic evaluation of the performance and further optimization studies. Because the requirements for the HPC utilization are high in terms of availability, reliability, efficiency, and sustainability, as the NHR resources are intended to completely replace the dedicated resources of the university Tier-2 centers on the long run. And this is only possible if it is ensured that the concept works well.
But how can one actually determine how well the integration is working? This supposedly simple question is not easy to answer, because first, universal measures for the performance need to be defined, and second, it must be decided what is actually considered sufficiently good while taking additional general aspects and limitation into account.

5.5.1. Appropriate Evaluation Measures

The evaluation of how well an integrated grid resources works is a complicated endeavor. It always involves several, individual and site-specific aspects and heavily depends on what characteristics are to be rated. Therefore, adequate measures are required to achieve a representative evaluation.
As already indicated above, the failure rate and CPU efficiency are well suited for general performance evaluations of a grid resources as they allow overall conclusions on the reliability and the efficiency of the utilization. But they also have their limitations that make different measures necessary, dependent on what sort of evaluation is desired.
In the following, different quantities are described that are mostly based on the always available payload monitoring and can be used to evaluate and compare the performance of a compute resource, mainly focusing on three categories: performance, sustainability, and reliability.

Remark: The variable names are chosen according to the naming scheme of the CMS HTCondor monitoring. A complete overview of the relevant ones for this work is provided in Table B.2. Variables and values that follow the official naming are marked with an asterisk (*), while own definitions are not marked.

5.5.1.1. Performance Rating

The pure compute performance that reflects the potential of the resource can be assessed with benchmarks and is important for the accounting, as described in Section 4.4. It is expected to be high for a modern HPC center, like e.g. HoreKa, which has a HS23 value of around 20 [185]. However, this measure is representing the optimal use of a resource but is not very indicative of how well the resource is integrated and utilized, as other factors can deteriorate the performance in production. For example, if a job needs to wait on remote data, the pure performance rating is not meaningful for a site as a whole as it does not reflect the actual throughput limited by I/O waiting times.

Therefore, in this sense, the CPU efficiency is a better quantity for evaluating the integration of a resource, as it rates the efficiency of the utilization independently of the pure performance.

5.5.1.2. CPU Efficiency

The CPU efficiency of a grid job, can be defined as:[10]

$$\mathrm{CpuEff}^* = \frac{\mathrm{CpuTimeHr}^*}{\mathrm{CommittedWallClockHr}^* \times \mathrm{RequestCpus}^*} = \frac{\mathrm{CpuTimeHr}^*}{\mathrm{CommittedCoreHr}^*}\,, \tag{5.1}$$

which reflects the ratio between the total time the allocated part of the CPU was utilized aggregated over all cores (`CpuTimeHr*`) and the total runtime (`CommittedWallClockHr*`) multiplied by the requested number of cores (`RequestCpus*`), representing the theoretical maximum of the `CpuTimeHr*`.
However, it has to be mentioned that the efficiency can go above 100 %, despite an efficiency is theoretically capped at 100 %. This can happen because of multiple reasons, but all of them can be attributed to the fact that the `CpuTimeHr*` may not turn out as expected. One explanation is that the utilized number of cores contributing to the `CpuTimeHr*` in the numerator is greater than the requested number in the denominator. This can happen when CMS jobs overbook resources. Meaning, while a job officially only requests e.g. eight CPU cores, it may still uses ten or more, if it is allowed to. Since this is not considered in the officially accounted number of cores in the denominator in Eq. (5.1), it can be a reason for a CPU efficiency above 100 %.
Additionally, if CPUs support dynamic boosting, as most of modern CPUs if not deactivated, the actual processing power can exceed the baseline computation rate, increasing the `CpuTimeHr*` per core in the nominator again. The result is a `CpuTimeHr*` that can again exceed the `CommittedWallClockHr*` multiplied by the number of cores. In theory, it can also happen that a logical overbooking of resources differs from the exact physical core utilization, if Hyperthreading (Intel) or Simultaneous Multithreading (SMT, AMD) are activated. However, this should be accounted for in Eq. (5.1) by the number of cores in the nominator if the resource is configured correctly.
Overall, this limits the informative value, if the CPU efficiency is viewed without further context. Especially, because it is also highly dependent on the tasks that are executed. If an inherently inefficient job is running, the result is not meaningful for the site if not reviewed in context to others. Nevertheless, it is still one of the best available measures for the efficient utilization of a grid resource, because it gives an indication on if the invested CPU time is well used or wasted.
If derived for a certain workflow or the entire resource, the total CPU efficiency is also a good basis for a sustainability evaluation, as it describes how much of the invested compute power was successfully used for the task, and with it the invested energy. It can be calculated as:

$$\mathrm{CpuEff}_{\mathrm{tot}} = \frac{\sum_{\mathrm{jobs}} \mathrm{CpuTimeHr}^*}{\sum_{\mathrm{jobs}} \mathrm{CommittedCoreHr}^*}\,, \tag{5.2}$$

[10] Remark: The HTCondor CPU efficiency (`CpuEff*`) uses `CoreHr*` instead which can lead to differences. But an evaluation of the available monitoring data for HoreKa has shown that the presented definition is less affected of erroneous reporting and therefore more robust against faulty monitoring data.

where the sums go over all selected jobs, e.g. of a certain workflow. This quantity is representative for the efficiency of an integrated site as it stands for the total fraction of compute that was used in a productive way. Here, it has to be mentioned that this value must not be confused with the averaged CPU efficiency:

$$\text{averaged CpuEff} = \frac{1}{\text{nJobs}} \sum_{\text{nJobs}} \text{CpuEff}^* \,, \tag{5.3}$$

where `nJobs` is the total number of selected jobs. Because simply averaging over the CPU efficiency of every single job weights each job identically, independent of its runtime. This can lead to a deviation from the value of the total CPU efficiency above and makes it way more vulnerable to distortions caused by outliers. For rating the efficiency of a grid site, the total CPU efficiency is therefore preferred over the averaged one as it gives a more accurate picture of the overall efficiency by implicitly considering that jobs that take more time also have a greater impact on the final result.

For the interpretation, however, it has to be kept in mind that the resulting efficiency is sort of inconclusive without a reliable baseline. As an example, an arbitrary efficiency rating of 6/10 is not considered ideal. However, it can be still sufficiently good according to the circumstances in the case that the other ratings are with only 5/10 even worse. Applied to the integration of resources, this could be interpreted as follows: If a compute resource is not performing well, it always has to be considered if this is a problem of the specific resource provider or maybe a global effect. An evaluation independent of the environment is therefore always dangerous and can lead to misinterpretations.

Also for the interpretation in terms of sustainability, caution is required. Because without an absolute rating of the resource itself, e.g. based on the power consumption (see Section 5.5.1.10 below), the CPU efficiency is inconclusive: If, for example, a ten year old CPU is used with 100 % efficiency, the output in relation to the invested energy could still be worse than for a modern CPU that is only running at 80 % due to the much higher performance per Watt. And also the overhead for cooling, inefficiencies of the power supply, and other general factors are not taken into account by this simple evaluation.

5.5.1.3. Processing Efficiency

Alternatively, one could also go another step further and set the invested `CpuTimeHr*` of a grid job in direct context to the output, i.e. the number of processed events (measured in thousands, `KEvents*`) it produced, to define the direct processing efficiency, P, based on the `CpuTimeHr*`, as a measure for the throughput:

$$\text{P} = \frac{\text{KEvents}^*}{\text{CpuTimeHr}^*} \,. \tag{5.4}$$

It has to be noted that this quantity is not a classic efficiency, but reflects the throughput per invested compute and therefore how performant and efficient the processing is. The difference is that it provides an absolute value for the rating of a site in contrast to the CPU efficiency, which says nothing about the actual performance, but only how well the allocated resources are used. If derived correctly, this measure is therefore more meaningful in the HEP context for the performance rating of an integrated resource.

However, without a corresponding baseline, it is not possible to draw conclusions about the actual efficiency of the processing based on this absolute quantity. Because a more performant CPU may still process more events despite a lower CPU efficiency in comparison to an older CPU in the same time period, which would lead to false conclusions if no independent performance rating of the used hardware is taken into account. This is the downside of the direct processing efficiency, because the throughput obviously also strongly depends on the hardware. A simple definition like above for the event throughput is regardless of what hardware is involved and therefore does not provide an independent rating.

At first glance one therefore might think: Then, the direct processing efficiency is ideal for evaluating a resource with unchanged hardware over a longer period of time against itself. But unfortunately it's not that simple, because the throughput is of course also heavily dependent on the workload. A simple averaging over all jobs therefore does not really provide an independent measure for a grid site. As a consequence, the direct processing efficiency in this simple form is only meaningful and representative in comparison to other sites with the same hardware that run the same workflows.

In order to reduce the hurdles and to obtain a more realistic result, the invested compute time needs therefore to be weighted in dependence of the hardware performance to represent its actual value. To implement this, the simple definition can be extended accordingly to introduce the direct weighted processing efficiency of a grid job:

$$\mathrm{P}'(p) = \frac{\mathrm{KEvents}^*}{\mathrm{CpuTimeHr}^* \times p}\,, \tag{5.5}$$

where p is an adequate performance rating of a given site, like e.g. the HS23 score. This now creates better absolute comparability, but with the disadvantage that the p factor has no exclusive definition and the results therefore vary with it. Since it is only a valid measure if the basic performance of the resources is considered correctly, it therefore should only be used on a comparative level with other sites whose p was derived the same way. One possible method to derive the performance factor from the available job monitoring data was developed and is provided together with an exemplary evaluation in Section A.3.

To also take inefficiencies into account, e.g. caused by remote transfers limited by the available bandwidth, the definition can be extended. Considering the whole committed job times multiplied by the number of cores (`CommittedCoreHr*`), which include startup, idle times, I/O waiting times, and so on, instead of the pure `CpuTimeHr*` leads to:[11]

$$\mathrm{P}(p) = \frac{\mathrm{KEvents}^*}{\mathrm{CommittedCoreHr}^* \times p}\,. \tag{5.6}$$

Since the evaluation of the quantity for a single job is not robust against outliers and fluctuations, it is more reasonable to evaluate the quantity for a complete workflow.

[11] Remark: When the direct processing efficiency is referred to in the following, it is always this quantity. If the initial definition from above is used, it is actively described.

For this, it can be simply aggregated over all selected jobs of the evaluated sample:

$$\mathrm{P}_{\mathrm{agg}}(p) = \frac{\sum_{\mathrm{sel}} \mathrm{KEvents}^*}{\sum_{\mathrm{sel}}(\mathrm{CommittedCoreHr}^* \times p)} \,. \tag{5.7}$$

Accordingly, the unweighted aggregated processing efficiency, $\mathrm{P}_{\mathrm{agg}}$, can be defined by neglecting the performance factor p.

The aggregated processing efficiency is the most expressive when compared to the result for the same workflow that ran at another site. Because just like described before, external dependencies and inherently inefficient workflows cannot be avoided. But with averaging over all jobs of a workflow and comparing with another site, this can be mitigated best.

For highest accuracy, here also has to be considered that p can be different if an integrated resources is not perfectly homogeneous in hardware. The denominator therefore can be evaluated per node type. However, since this is way more complicated, an averaged p for a whole site should provide a sufficiently reliable result. Additionally, it has to be kept in mind that the interference with other running jobs in production is not directly considered with this quantity. But with utilizing only the monitoring data at job level, this is too complicated or even impossible to consider and therefore has to be neglected. Nevertheless, when evaluating the measure for a full workflow, this is implicitly included in the overall tendencies.

In conclusion, with this measure, a comparable, absolute value can be derived for a grid job that now also takes inefficiencies into account. When averaging the result for a workflow, it provides a good foundation for comparison with other sites based on an absolute value reflecting the performance and throughput. Conclusions based on the relative processing efficiencies of different sites derived for representative workflows with sufficient information available can therefore be a good foundation for a grid site evaluation.

The next step to be even more precise is to also take job failures into account for an evaluation.

5.5.1.4. Failure Rate

The job failure rate, F, for a site is rather trivial to define as the ratio of failed jobs to all completed jobs at the site:

$$\mathrm{F} = \frac{\text{number of failed jobs}}{\text{total number of jobs}} \,. \tag{5.8}$$

In the first approximation, it can be seen as a measure for the reliability of a grid resource and is always a good indicator for problems occurring at a certain site or in the grid as a whole, if appearing on multiple sites. But drawing reasonable conclusions about site performance in general from this is not quite so easy as it is not a feasible measure for a site if viewed independently of all circumstances, since the WLCG is a highly complex, worldwide system with many participants. Therefore, there can be plenty of reasons for job failures that cannot be accounted to the resource provider but, for example, to a misconfiguration of workflows or problems with the CMS AAA data federation, which is making an independent evaluation of the site reliability even more complicated.

And additionally, the simple failure rate as described in Eq. (5.8), does not take different runtimes of failing jobs into account, leading to similar problems as for the averaged CPU efficiency defined in Eq. (5.3). Therefore, no conclusions can be drawn directly from it on the inefficiencies of a site and the resulting CPU time wasted. Because for example, if a million jobs fail within a second on startup, the failure rate increases drastically. However, the loss of compute power is rather small. And the other way round, a small number of long running jobs failing can have a much greater impact on the overall site performance but are not significantly reflected in the failure rate. Consequently, direct conclusions from the failure rate can easily be misleading unless the failures are put into the context of the job runtimes (`CommittedWallClockHrs*`). Therefore, the quantity is very helpful as a tendency but also no absolute measure for the reliability of a compute site as it can easily be spoiled.

Nevertheless, it is conceptually an important supplement to consider job failures for rating how good a resources is actually utilized. Because if, for example, jobs run highly efficient but fail in the end, the overall efficiency of the resource may still be decent but the actual output, represented by the processing efficiency, is significantly reduced – which is not reflected by the classic CPU efficiency as a measure!

Therefore, the already discussed measures can be refined by setting them in context to job failures.

5.5.1.5. Corrected Processing Efficiency

By taking job failures into account when calculating the aggregated processing efficiency ($\mathrm{P}_{\mathrm{agg}}(p)$), for a workflow, a more accurate quantity can be derived which, while still an hardware- and workload dependent measure, better reflects the absolute processing performance of a site. This correction is required for a more realistic site rating, since the direct and aggregated processing efficiency do not consider that failed jobs did not produce any usable output and therefore reduce the overall site performance accordingly. The corrected quantity for a given workflow reads:

$$\mathrm{P}_{\mathrm{agg}}^{\mathrm{corr}}(p) = \frac{\sum_{\mathrm{all}} \mathrm{KEvents}^* - \sum_{\mathrm{failed}} \mathrm{KEvents}^*}{\sum_{\mathrm{all}} \mathrm{CommittedCoreHr}^* \times p}\,, \tag{5.9}$$

where all indicates all jobs of the same workflow. The numerator describes the real throughput in form of the number of actually usable events processed in successful jobs by correcting the total number of produced events by the sum over the failed events. The denominator remains unchanged compared to Eq. (5.7) with the same considerations applied. A simple summing over all selected jobs resulting in the total allocation time weighted with the averaged performance factor of the site, however, should be fine. The unweighted version, $\mathrm{P}_{\mathrm{agg}}^{\mathrm{corr}}$, can be defined accordingly by simply neglecting the performance factor p.

With this definition, job failures are explicitly taken into account, together with implicitly considering inefficiencies, when evaluating the processing performance of a site in comparison to others. The disadvantage of it is still the direct hardware dependence. Without a proper rating, the measure is not expressive and two sites cannot be set into direct comparison. In such a case, the site can only compared against itself by comparing the different measures. But with the p factors estimated with the method presented in Section A.3 and a reliable baseline, the quantity can yield a realistic, meaningful estimation for a grid site.

For a more basic, efficiency-related evaluation independent of the hardware, it can also be useful to define the corrected CPU efficiency, which only has an intrinsic hardware dependence, as it does not provide an absolute value but a relative comparison of the time periods eradicating the direct dependency.

5.5.1.6. Corrected CPU Efficiency

While considering the total CPU efficiency of a site or workflow gives a good first impression of the performance and sustainability, it is not realistically taking job failures into account. Since the reasons for failures can be manifold and often do not originate from the resource provider, this still can be a valid foundation for a performance evaluation of a grid resource. Because the advantage of the classic CPU efficiency is that external errors are not assessed at the expense of the evaluated site. However, especially in terms of sustainability, a more representative performance evaluation of a grid site in the greater picture can only be achieved when all failures, also the externally caused ones, are taken into account. For this, the usual total CPU efficiency, $\mathrm{CpuEff}_{\mathrm{tot}}$ (Eq. (5.2)), of a site or workflow can be corrected by the wasted core hours of failed jobs that did not provide any usable output by subtracting them from the total `CpuTimeHr*`:[12]

$$\mathrm{CpuEff}_{\mathrm{tot}}^{\mathrm{corr}} = \frac{\sum_{\mathrm{all}} \mathrm{CpuTimeHr}^{*} - \sum_{\mathrm{failed}} \mathrm{CpuTimeHr}^{*}}{\sum_{\mathrm{all}} \mathrm{CommittedCoreHr}^{*}} \,. \tag{5.10}$$

While being more representative on the actual efficiency of a resource, this quantity is, like the processing efficiency, still a measure that strongly correlates with the overall grid performance. This again only can be mitigated by a comparison to other sites running the same workflows, which may corrects these effects to a certain extent under the assumption that the externally caused job failures are of a similar order.

In comparison to the corrected processing efficiency, an advantage of this quantity for the evaluation is that it is way simpler without further dependencies, as it is only specified relative to the maximum possible and therefore, the absolute performance rating of the resource is not required. But in turn, this makes it less comparable as it only provides a relative but no absolute value, as explained in Section 5.5.1.2, which needs to be taken into account when comparing sites in terms of sustainability.

In conclusion, both CPU efficiency definitions can be important for the performance evaluation of a site, depending on which aspects are more emphasized for a certain analysis. And by combining both, further conclusions can also be drawn: If they are similar, one can conclude that the failing jobs had a similar efficiency, making it less probable that inefficiencies were the reasons for job failures, like e.g. connection timeouts or else.

5.5.1.7. CPU and Compute Time Loss

While the failure rate gives a good first impression on the reliability of a grid site, it is quite limited in its informative value. Especially when considered on its own, it can be misleading. The reasons have already been explained in Section 5.5.1.4. In

[12] Here, it is important to mention that a simple weighting of the average CPU efficiency with the failure rate would not yield a correct result, as the rate must not be interpreted in form of a correction factor.

combination with the corrected CPU efficiency, a more complete and realistic picture of a site's efficiency emerges. Very likely the two variables are correlated and can provide indications of the reasons for their deterioration. For example, when there are problems with the data federation, the CPU efficiency may drop due to throttled data transfers and the failure rate increases when jobs eventually fail. Together, they are therefore an important measure in terms of operation and reliability. However, both are exclusively relative and do not provide an absolute value for the actual impact of failures and inefficiencies.

The problem with this can be explained using a simple example: If only two jobs are running at a site and one of them fails, the failure rate is 50 %. The main inadequacy of the measure is now that 500000 failing jobs out of a million lead to the same 50 % failure rate. With the same relative result, the actual extent of the problem cannot be assessed and the measure is therefore inconclusive in an absolute sense. One time, the value indicates massive problems, but the actual impact is rather small, and the other, the negative impact is enormous. To avoid this restriction, the absolute number of failures and the runtimes of the failed jobs, as described in Section 5.5.1.4, must be considered.

This can be done by deriving the CPU Time Loss, c, of a workflow or entire site. The first step is to evaluate, how much CPU time was lost due to the failures, thus simply aggregating the lost `CpuTime*` of failed jobs:

$$c_{\mathrm{agg}} = \sum_{\mathrm{failed}} \mathrm{CpuTimeHr^*} \; . \tag{5.11}$$

It is therefore a quantity that directly reflects the impact of the errors and failures, and not just the simple frequency of occurrence. Taken on its own, this measure is not yet representative. But in relation to the total invested `CpuTimeHr*` for the workflow or site under investigation, it is representative for the efficiency and reliability, as it reflects how much of the invested CPU time lead to a productive outcome. The relative aggregated CPU Time Loss, $c_{\mathrm{agg}}^{\mathrm{rel}}$, is derived accordingly as:

$$c_{\mathrm{agg}}^{\mathrm{rel}} = \frac{\sum_{\mathrm{failed}} \mathrm{CpuTimeHr^*}}{\sum_{\mathrm{all}} \mathrm{CpuTimeHr^*}} \; . \tag{5.12}$$

It can be interpreted as an inverted return-of-invest efficiency and with it, how well and reliable a site is performing. This measure also mitigates the previously described danger of a misleading interpretation of a small rate of failures. In combination with very long running jobs, a low rate still can cause a significant waste of CPU time, reducing the site's reliability and efficiency. To come back to the aforementioned example, this small exemplary calculation shows the advantage of the relative CPU Time Loss over the failure rate and the CPU efficiency:

Assuming, ten jobs with a typical runtime of 10 hours are executed at a grid site. If two fail after one hour and the rest is successful, the failure rate is 20 %, while the relative CPU Time Loss accumulates to only 2.44 % of the total invested compute power (82 core hours). Instead, if only one job failed but after 9 hours of calculations, the failure rate is significantly smaller with only 10 %. This gives the impression that the grid site operation in the second scenario is more reliable when only considering the failure rate. But when deriving the relative CPU Time Loss of 11 %, it shows that it is actually the other way round. This quantity therefore better reflects the actual (relative) impact of job failures and prevents the misleading interpretation of small failure rates.

Even more beneficial is that, together with the total CPU time, the relative CPU Time Loss is more expressive than the simple failure rate and the CPU efficiency, as it projects the failures on an absolute value with more meaning (the absolute CPU Time Loss). '300 out of 1000 core hours were lost due to failures' is a way more meaningful statement than '300 out of 1000 jobs failed', as it explicitly considers the invested compute.

What is not yet considered with this quantity is that the realistic loss of a failed job is more than the CPU time. The attentive reader will have noticed that failing jobs that had a bad CPU efficiency and therefore a relatively smaller CpuTimeHr* actually reduce the CPU Time Loss (relative and absolute) accordingly, again giving the impression that the impact is smaller than it actually is. Just like the failure rate, this measure still can lead to a misconsumptions about a grid site as it penalizes when failing jobs are running efficiently, which is not intended for a reliable measure.

To correct for this flaw, the measure therefore needs to be extended by incorporating the full compute time to show the impact of the failure rate on reliability and sustainability of a grid site, thus introducing the relative Compute Time Loss, C:

$$C_{\mathrm{agg}}^{\mathrm{rel}} = \frac{\sum_{\mathrm{failed}} \mathrm{CommittedCoreHr}^*}{\sum_{\mathrm{all}} \mathrm{CommittedCoreHr}^*} \,, \tag{5.13}$$

with the aggregated Compute Time Loss defined analog to Eq. (5.11):

$$C_{\mathrm{agg}} = \sum_{\mathrm{failed}} \mathrm{CommittedCoreHr}^* \,. \tag{5.14}$$

This extension ensures a more realistic rating of a grid site by correctly considering the total compute time lost due to job failures. Therefore, it directly reflects how well a site is performing.

In terms of efficiency, this quantity unfortunately allows only to draw conclusions on the CPU efficiency of failed jobs, which is considered implicitly and sets Eq. (5.11) and Eq. (5.14) in relation:

$$c_{\mathrm{agg}} = \mathrm{CpuEff}_{\mathrm{tot}}^{\mathrm{failed}} \times C_{\mathrm{agg}} \,, \tag{5.15}$$

where $\mathrm{CpuEff}_{\mathrm{tot}}^{\mathrm{failed}}$ is the total efficiency (Eq. (5.2)) of all failed jobs from the evaluated sample. From the ratio of the two Losses, conclusions can be drawn about the failures. If it is ~1, this shows that the failed jobs were executed with a similar efficiency as the successful ones. It can be deduced from this that a direct correlation between the failures and the efficiency is unlikely. Conversely, a larger deviation indicates inefficiencies that are directly related to the failures and thus point to an intrinsic problem, since successful jobs are less (or not) affected. This knowledge can be helpful for troubleshooting and debugging. Therefore, the relation of the two Losses can be another useful quantity in the available evaluation toolset. In the absolute sense, however, the ratio has no real meaning for the evaluation, since the efficiency of failed jobs has no influence on the absolute loss, as the full CommittedCoreHr* is accounted anyway in the end.

In conclusion, the (relative and absolute) Compute Time Loss defined in this way introduces a meaningful and hardware-independent metric for the evaluation of a grid site. It sets the failure rate in direct context to its actual impact and provides a more expressive measure by reducing the space for misleading interpretations, as it correctly covers failing jobs that are nevertheless efficient.

What it does not incorporate are losses of compute power that occur during successful jobs due to inefficient processing. These are addressed below.

5.5.1.8. Inefficiency Loss

Jobs not running fully efficient waste compute time, no matter who or what is responsible. In terms of sustainability, it is therefore important to consider also this loss, even if external reasons are causing the inefficiencies.

The Inefficiency Loss, I, of a singe job is directly derived from its CPU efficiency and is defined as the difference instead of the ratio, absolutely quantifying the loss:

$$\mathrm{I} = \mathrm{CommittedCoreHr}^* - \mathrm{CpuTimeHr}^* \,. \tag{5.16}$$

It describes the absolute number of core hours that are lost due to an inefficient processing of a single job. For a site or workflow, the quantity can be obtained by simply summing over all selected job. However, to avoid a double-accounting of failed jobs, this selection should only consider successful jobs, since failed ones should entirely be considered as loss – independent of their efficiency! With this follows for the aggregated absolute Inefficiency Loss:

$$\mathrm{I}_{\mathrm{agg}} = \sum_{\mathrm{succ}} (\mathrm{CommittedCoreHr}^* - \mathrm{CpuTimeHr}^*) \,. \tag{5.17}$$

From this, the relative Inefficiency Loss of a job can be derived:

$$\mathrm{I}^{\mathrm{rel}} = \frac{\mathrm{CommittedCoreHr}^* - \mathrm{CpuTimeHr}^*}{\mathrm{CommittedCoreHr}^*} \,. \tag{5.18}$$

This quantity is literally the *inefficiency* of a job, as it represents the complement of the CPU efficiency: 1 – CpuEff*. Under consideration of all successful jobs of the desired selection, the relative aggregated Inefficiency Loss for a site follows:

$$\mathrm{I}^{\mathrm{rel}}_{\mathrm{agg}} = \frac{\sum_{\mathrm{succ}} (\mathrm{CommittedCoreHr}^* - \mathrm{CpuTimeHr}^*)}{\sum_{\mathrm{succ}} \mathrm{CommittedCoreHr}^*} \,. \tag{5.19}$$

It has to be mentioned that this quantity, just like the CPU efficiency, is not conclusive for a grid site's performance without additional context. Although the total Inefficiency Loss is already an important factor that stands for itself ('X out of Y core hours were wasted'), the quantity is not expressive for the performance of a certain grid site independently of a comparable baseline. The reason is, same as for the CPU efficiency, that it depends heavily on external factors, like intrinsic inefficiencies of workflows or the reliability of the data federation. For a fair and realistic evaluation, a direct comparison with another site is therefore always recommended, assuming that it is affected by the external factors to the same extent.

5.5.1.9. Total Loss

Ultimately, to obtain a complete picture, the aggregated Compute Time Loss, C_{agg}, and the aggregated Inefficiency Loss, I_{agg}, can be combined to form the Total Loss, T, for a site or workflow:

$$T = C_{agg} + I_{agg} \ . \tag{5.20}$$

which stands for the total amount of core hours invested without any usable outcome. Set in relation to the total invested core hours (CommittedCoreHr*), the relative Total Loss can be defined analog:

$$T^{\,rel} = \frac{C_{agg} + I_{agg}}{\sum_{all} \text{CommittedCoreHr*}} \ , \tag{5.21}$$

where 'all' indicates all jobs of the selected (sub-)set. With incorporating both Losses, this measure explicitly includes the impact of the failure rate as well as job inefficiencies, leading to a realistic assessment of a site's performance and reliability. Additionally, it has the advantage that it does not have a direct hardware dependency, in contrast to the aggregated, weighted processing efficiency (Eq. (5.7)), but still yields a realistic, absolute rating of a site.

Another benefit is that the Total Loss is already a meaningful measure on its own for the integration, even if reviewed independently. In contrast to all other described metrics that are only fully conclusive in comparison to other grid sites or require additional information for an interpretation, the Total Loss can be interpreted directly. It describes the wasted compute power (total or relative to the invested), which is an extremely important indicator, also from the resource provider's perspective. Since the participating sites demand from the LHC experiments to make responsible and efficient use of their contributed resources, this quantity is already useful for evaluating the utilization. And it is significantly more accurate than the classic evaluation measures – failure rate and CPU efficiency – that are typically used, but do not reflect the actual impact.

Nevertheless, the reasons for inefficiencies and errors cannot be deduced solely by evaluating the Total Loss. It can originate from an inefficient integration as well as from an inefficient utilization, which are both included in the measure but not isolated. A comparison of the relative Total Loss with other sites and workflows therefore again provides the most conclusive results on the integration and helps to identify – or at least exclude – some possible origins of occurring problems.

Finally, as a last extension of the concept, the employed hardware can be considered as well to be even more representative. Because for now, each lost core hour is considered equally valuable. By reviewing the results for the aggregated and relative Total Loss under the consideration of an hardware performance factor, as introduced in Eq. (5.6), it adds another layer to the interpretation. Because two sites with an identical relative Total Loss of e.g. 25 % are not necessarily identical in the sense of wasted potential. Therefore, this can be useful for an evaluation of grid sites, since wasting compute at a more performant and more expensive resource is additionally penalized under this consideration. Eventually, this is not essential for how well a site is integrated, as described above, but provides additional context when comparing the results to other sites.

One additional thing to mention is that this extended Total Loss still needs to be interpreted carefully. It is only expressive for evaluating how efficiently an integrated resource is utilized but does not reflect the energy efficiency. E.g. even if the lost potential is higher at an HPC resource, the energy efficiency still can be better compared to an efficiently utilized resource with ten year old servers. It is the same principle as discussed at the end of Section 5.5.1.2. This shows that, most importantly, the measure must fit the evaluation target.

For an evaluation that addresses the sustainability of the system, therefore, additional aspects must be considered to avoid drawing wrong conclusions, which will be discussed in the next part.

5.5.1.10. Sustainability Measures

Ultimately, aligning with the goal of the WLCG, not only the performance but also the sustainability would be an adequate measure for rating the integration of a grid site. This is already today reflected in newer trends and developments within the HEP community. For example, first steps are going in the direction of benchmarking resources not only on the basis of a pure performance assessment, but in relation to the invested energy in order to take sustainability into account [186].

In general, there are many additional possibilities given, if information on the power consumption of the resource is available. This, however, is already the biggest limitation for such an evaluation, since the required information is, if available, typically not easily accessible. Especially not for other sites, as this is not part of the usual payload monitoring data, mostly preventing direct comparison studies on the performance of a site based on the sustainability.

If accessible, the previously introduced measures can be extended by this aspect to better reflect the sustainability (in terms of energy efficiency) of a given resource. One way to realize this would be to introduce a measure for the sustainability, s, analog to the performance factor p in Eq. (5.6). This value could for example describe the power draw per core hour, which is a reasonable measure for the energy efficiency. The problem here is that an independent, reliable rating for the resources requires dedicated benchmarks for ensuring comparability. Unfortunately, nothing like HS23 is available (yet) for the energy efficiency evaluation in the HEP context at the time of writing. Therefore, an averaged sustainability factor, $\overline{s}$, can only be derived data-driven at the moment by summing over all utilized worker nodes that are part of the integration:

$$\overline{s} = \frac{\sum_{\text{all}} \text{energy consumption}}{\sum_{\text{all}} \text{contributed core hours}} \, . \tag{5.22}$$

The calculation sets the aggregated core hours of all utilized worker nodes in context to their energy consumption and yields the energy that needs to be invested to provide one core hour. Here it is to be mentioned that such an evaluation is only really feasible on node or even site level. An evaluation on job or workflow level is way more complicated as it would require to correctly considered the interference with other jobs running on the same physical node. In principle, on HPC with whole node scheduling, this is indeed possible but requires more complicated analysis steps. At first, the whole dataset needs to be separated by each individual host. Then the running jobs need to be matched with the according drone jobs they were running in, based on the timestamps. If this all could be separated and identified, the according power consumption per job can be evaluated depending on how much it used of the total consumed `CommittedCoreHr*` of the whole node, and eventually aggregated per workflow. But unfortunately, with this direct evaluation, the averaging effect is lost and failure rates and external reasons for inefficiencies need to be considered more carefully again.

Therefore, this could be interesting from the experiment's perspective, as it may be used to roughly evaluate the power consumption of a workflow, but for evaluating a grid site, it is less useful. It would only provide a basis for comparison, if dedicated benchmark jobs are used to determine an absolute rating for a resource, like s introduced above. However, the averaged value, $\overline{s}$, derived from data is not capable for creating a reliable basis for comparison. This factor could in principle be truncated by idle times, inefficient jobs, and a less than full utilization of a resource. But for an internal evaluation, it should be sufficient. Only the comparison with other sites has to be handled with care.

Following the idea of the sustainability factor, the quantities can be adapted. At first, the unweighted corrected processing efficiency (see Section 5.5.1.5) is expressed as a function of $\overline{s}$:

$$\mathrm{P}_{\mathrm{agg}}^{\mathrm{corr}}\left(\overline{s}\right) = \frac{\sum_{\mathrm{all}} \mathrm{KEvents}^* - \sum_{\mathrm{failed}} \mathrm{KEvents}^*}{\sum_{\mathrm{all}} \mathrm{CommittedCoreHr}^* \times \overline{s}} \,, \tag{5.23}$$

describing the number of processed events per invested Watt hour (Wh). The problem with this measure is that it again depends on the workloads and is therefore only reliable when compared to other sites running the same workflows. However, since $\overline{s}$ itself is not official and not perfectly comparable when derived data driven, this measure is currently not a good candidate for the evaluation.

And also the Total Loss can be converted in the same way. With:

$$\mathrm{C}_{\mathrm{agg}}\left(\overline{s}\right) = \sum_{\mathrm{failed}} \mathrm{CommittedCoreHr}^* \times \overline{s} \,, \tag{5.24}$$

and

$$\mathrm{I}_{\mathrm{agg}}\left(\overline{s}\right) = \sum_{\mathrm{succ}} (\mathrm{CommittedCoreHr}^* - \mathrm{CpuTimeHr}^*) \times \overline{s} \,, \tag{5.25}$$

follows the total Energy Loss, E, of a site or workflow:

$$\mathrm{E}\left(\overline{s}\right) = \mathrm{C}_{\mathrm{agg}}(\overline{s}) + \mathrm{I}_{\mathrm{agg}}(\overline{s}) \,. \tag{5.26}$$

The quantity describes the total energy invested (in Watt hours, Wh) that did not lead to a usable outcome.

Accordingly to the other quantities, the Energy Loss set in relation to the total invested energy defines the relative Energy Loss:

$$\mathrm{E}^{\mathrm{rel}}\left(\overline{s}\right) = \frac{\mathrm{C}_{\mathrm{agg}}\left(\overline{s}\right) + \mathrm{I}_{\mathrm{agg}}\left(\overline{s}\right)}{\sum_{\mathrm{all}} \mathrm{CommittedCoreHr}^* \times \overline{s}} \,. \tag{5.27}$$

This quantity is an important measure that directly reflects the energy efficiency and with it the sustainability of an integrated resource. It is even useful to evaluate the environmental and economic impact of the research directly, as the result – the lost energy – stands for itself.

However, it should also be noted that the value determined in this way is only a minimal estimate of the power draw and does not include a lot of overhead that must be considered for a more realistic review. The energy that has to be invested for storage, network, cooling, and idle times of the worker nodes must not be neglected on the long run when evaluating the sustainability of a resource. Problematic is that this information is very unlikely to be available though, hindering a more accurate usage of this measure – at least at the time of writing this thesis.

Nevertheless, taking the step to directly relate the reliability and efficiency to the consumed energy adds additional meaning to the measures. And even if it is not yet ideal for use today, this could change in the future. As already mentioned in Section 4.1, at some point maybe the accounting will be even moved away from the traditional measures like core hours to, for example, a sustainability related power budget that strictly enforces an efficient utilization. Especially, if in the future, maybe not hardware, but energy availability becomes a limiting factor in the computing world. Then, the focus will lie on the sustainability of resources, which is why e.g. the exploration of ARM CPUs for HEP applications is ongoing, as they provide a better performance per Watt [186].

In conclusion, meaningful sustainability measures are important and gain even more importance on the long run. Because a sustainable operation is already being pursued, but will become absolutely mandatory in the future. The measures presented enable a reliable evaluation of a (grid) resource to eventually make decisions for the future – also in terms of technologies like ARM. However, while being highly representative and descriptive, they are beneficial, but not yet fully feasible for a conclusive evaluation today.

5.5.2. Additional Aspects

After introducing feasible measures as a basis for a reliable evaluation of an integrated resource, some additional aspects need to be considered. As described alongside the different measures above, each one has its benefits and disadvantages. The highest informative value can be achieved, when the best matching quantity is chosen according to the aspect that should be reviewed. Therefore, the procedure must primarily be oriented towards the type of resource, the relevant characteristics to be assessed, and what monitoring data is available.

Especially the last point is crucial, because there can be various limitations, as previously discussed in Section 5.3, in particular for HPC resources. At HoreKa, for example, monitoring data that goes beyond process level is limited due to its security policies. For I/O and bandwidth information, only very limited monitoring is available, and only for the whole nodes that are scheduled. This excludes an in-depth evaluation of hardware-related problems on pilot and even on job level, which can be critical for the identification of inefficiencies. E.g. the identification of data access bottlenecks is rather complicated as no immediate information on other remote data transfers at the same time is available. To this strict isolation, the HEP community needs to adapt with its tools and methods.

Consequently, often only the individual payload monitoring can be employed for an evaluation. All the measures presented in Section 5.5.1 are therefore developed to provide meaningful results solely based on what is definitely available from the official WLCG monitoring sources. Even if this unfortunately only provides a limited picture of the overall situation, it is still the best way of making a comparable assessment for a grid site. But in general, it has to be kept in mind that several additional aspects may not be covered that could in principle deteriorate the overall performance. Therefore, the results based on a job-level evaluation has always to be seen as a maximum estimate for the overall performance. Overarching effects that may influence the overall performance and stay undetected can be:

- **Virtualization Overhead:** As described in Section 4.2.4, the virtualization is crucial for enabling all kinds of resources to be used in HEP. Especially VMs, but also (stacked) containerization to a limited extent, can cause an overhead that is not considered in the payload monitoring.

- **Allocation inefficiencies:** It is not possible to draw reliable conclusions about the overall allocation and utilization of a resource from the jobs alone[13]. Scheduling inefficiencies and idle times can have a significant impact on the general performance of an integrated resource, but are not reflected in evaluation based only on independent, individual jobs.

- **Pilot Inefficiencies:** Job pilots running empty can have effects on the overarching efficiency but are not reflected in the efficiency of the individual jobs. This can become significant, when e.g. eight 1-core jobs start within a pilot and after some time, only one of them is still running. For the remaining runtime of the job, the allocation is then only 1/8th of the optimum, if the remaining wallclock time is to small to schedule additional jobs to this pilot.

- **Interference Effects:** Another effect that can stay undetected in the individual job monitoring are interference effects between jobs. If, for example, one very data intensive task is utilizing the full available bandwidth, this obviously influences the performance of all the other jobs running on the same worker node (or even site). From only evaluating individual jobs, this cannot be identified. For the whole site, this effect is considered by averaging over the performance metrics for all jobs. But if e.g. only a certain workflow that ran on multiple sites is utilized for the evaluation of a full resource, this should not be neglected as it can have a significant impact on the performance. However, without additional information, this is extremely hard to consider and requires complex methods.

- **Intrinsically Inefficient Workloads:** If inefficient workflows are submitted by the experiments, this obviously reduces the performance of a grid site. If reviewed independently to other sites running the same workflow and out of context, misinterpretations are the result.

Nevertheless, most of the above mentioned scenarios are beyond the scope of the resource provider anyway and need to be mitigated and optimized on collaboration level. Obviously, they are definitely relevant for e.g. the sustainability evaluation and may even can be addressed implicitly with the aforementioned sustainability measures, if sufficient monitoring data is available. But for evaluating the performance and efficient integration of a resource, they are only secondary, since a grid site usually cannot do anything about it.
The same applies for inefficient workflows submitted by the experiments. They alter the performance of a resource, which is particularly bad for the integration of

[13] Fortunately, this is implicitly covered by the dynamic integration with COBalD/TARDIS in this case.

valuable HPC resources, but also here, this has to be accepted. If intrinsic reasons cause inefficiencies to an unbearable extent, the only possibility would be to stop the integration in such a case, which is definitely not desirable. Therefore, all sides need to work closely together to enable a most efficient grid operation and it is the responsibility of the LHC community, to use the given resources as good as possible. In conclusion, these aspects are affecting all participating sites more or less similar and have to be accepted to a certain extent. They are of course negative, but do not necessarily reduce the informative value of a job-based site evaluation.

A last point to emphasize again is that all integrated grid sites participating in the WLCG are running production workflows. This adds complexity and uncertainty to the evaluation of a site, since no identical jobs are running at two different sites. The consequence is that a comparison based on individual production jobs cannot be considered a real benchmark, but only provides overall tendencies. However, despite this is not ideal, with sufficient statistics, the results should be reliable to fundamentally evaluate the performance of a site based on the payload monitoring data.

The easiest way to mitigate these problems is a comparison-based evaluation of the overall performance. For this, sites should be selected that are integrated under similar circumstances and conditions. Then, under the assumption that those sites are affected to a comparable extent by the above-mentioned external influences and untraceable effects, the comparison is the most meaningful.

For the integration of HoreKa as a prototype for the utilization of an NHR center within the WLCG, an evaluation in relation to the two other sites at KIT therefore makes the most sense. Especially TOpAS, which is also integrated with COBalD/TARDIS as an opportunistic resource in the same OBS, is a perfect candidate for a direct performance comparison. Secondly, the to-be-replaced university Tier-2 center for CMS in Aachen is also a good choice to provide a comparison baseline.

With this, sufficiently expressive conclusions can be drawn from the analysis on job level for the purpose of this work. But it has always to be kept in mind that additional aspects beyond the own influence have an impact in a complex system like the WLCG.

5.6. Evaluation of the Opportunistic Integration of HoreKa

The pledged integration of NHR resources in the future is an absolute novelty and it has to be ensured that these HPC resources are used as well as possible – aligning with the goals formulated in Section 5.4. The experience from the opportunistic integration of HoreKa is therefore extremely valuable, as it offers unprecedented insights into the potential, but also possible limitations of such resources. Since HPC centers have an inherently different computing model and focus, such a review is important before increasing the shares on the centers, as planned with the future German HEP computing strategy. Because only by evaluating how well the integration is working with HEP workflows it can be confirmed that the HEP community is ready for the next step into the future. Or alternatively measures can be taken to enhance the efficiency and reliability of such resources to ensure a proper utilization for the HL-LHC era.

This is therefore also the first step to optimize the integration. Because reviewing efficiency, performance, and sustainability based on the derived quantities under the explained considerations helps to separate internal and external reasons for inefficiencies and allows to develop strategies for the mitigation of potential problems. Hence, the following analysis of HoreKa is paving the way for a successful pledged integration in the future.
But before analyzing the payload monitoring data, it is important to do a proper data scouting and selection in advance to ensure the reliability of the data. This process is described in the following sub-section.

5.6.1. Data Preparation and Validation

A proper data preparation is the prerequisite for a consistent evaluation of a grid site over multiple years (2022-2024). Of course, things can change over time and achieving 100% compatibility is almost impossible. But with an intelligent data selection and adequate actions to mitigate the influence of errors in the monitoring data, the most reliable results can be obtained.
The first problem that needs to be addressed is the fact that most of the basic monitoring quantities are available in different versions within the CMS monitoring data. Sometimes, this can be helpful for validation and to recover data, as utilized in Section A.2.3. But in the case of the `CoreHr*` and `CommittedCoreHr*`, both describing the invested core hours of a grid job, it happens that they are often not in agreement. It is unclear, why this is the case and the official documentation is inconclusive. Unfortunately, those are the most relevant ones for this work, as they are essential for most of the measures introduced in Section 5.5.1 and need to be reliable. Therefore, as a first step, a selection must be made, which quantities will be used for the grid site evaluation. For this, it has to be decided which data is the most reasonable.
An in-depth analysis of the entire monitoring data for the different sites from 2022 to 2024 has shown that the `CommittedCoreHr*` is the more reliable candidate and will be used for the further evaluations. Details are provided in Section A.2.1.

The full data preparation process is now separated in a filtering of duplicates, a job type selection, and the identification, recovery – if possible –, and filtering of erroneous monitoring data. Optionally, additional filtering and selections can be necessary, depending on the planned analysis. E.g. for some evaluation measures, it is important to restrict the data to the same workflows to achieve the best comparability between different sites.

Filtering of Duplicates: Due to unknown reasons, the monitoring data can contain duplicates. Within the HoreKa data from 2022 to 2024, they make up a significant fraction of around 13 % of all job data entries (12 % for all reviewed sites together). To correct the dataset for the site evaluation, it is safe and sufficient to simply keep only one data point and exclude all duplicates, since they are entirely identical. More details are given in Section A.2.2.

Job Type Selection: HoreKa is used since the beginning as an opportunistic, but regular grid resource and is therefore running standard CMS workflows. The categorization of those is represented by the `Type*` of a job that can be one of *production**,

Table 5.2.: The number of jobs per different type that ran at HoreKa between 2022 and 2024. The overview shows that Production* and Processing* jobs are the most relevant for the evaluation of a grid site. All others are not really representative, as they make up only a small fraction of the invested compute performance and are not representative for HEP in general.

CMS_JobType*	Number of Jobs	Average CommittedCoreHr*
Production*	733467	13.028
Processing*	292149	8.672
LogCollect*	31324	0.0301
Cleanup*	49330	0.011
Merge*	226	0.198
Harvesting*	1537	0.201
Analysis*	1307	1.048
All	1109340	10.900

*analysis**, or *test**. To date, mainly *production** jobs were running at the HPC center. A more precise categorization is done by the CMS workflow management with the following sub-categories of `Cms_JobType*`: *Production**, *Processing**, *Merge**, *Cleanup**, *LogCollect**, *Analysis**, and *Harvesting**.

The main purpose of *Production** jobs is the generation of simulated datasets. They typically do not access real data, but only PREMIX datasets that are used to optimize the MC generation. *Processing** is the category that summarizes all jobs that process data (or simulation). This involves e.g. reconstruction steps, or the production of different AOD tiers. *Merge** jobs are used to combine multiple smaller datasets into a larger dataset. Therefore, they are mainly data intensive and do not require lots of CPU performance. *Cleanup** jobs remove temporary or intermediate files generated during the multiple steps of the CMS processing pipeline. *LogCollect** are collecting, transferring, and cleaning up local log files. *Harvesting** workflows describe the final extraction, aggregation, and preparation of processed data before finally storing it.

An overview of the jobs per category running at HoreKa in the past is provided in Table 5.2 in combination with their average `CommittedCoreHr*`. This shows that mainly the two types *Production** and *Processing** are relevant for the performance evaluation of a grid site. The few Analysis* jobs that were executed have been small, successful tests that do not have a significant impact on the evaluation of HoreKa and can be neglected. The other job types do not represent real HEP workflows and have no significant influence on the overall performance due to their short runtimes. Additionally, they are typically running at very low CPU efficiencies (in single-digit percentage range) but very reliable with only small failure rates. An exclusion increases the total CPU efficiency of HoreKa for the entire period by only 0.006 %. They can therefore be safely neglected without a loss of significance.

In conclusion, the restriction of the `CMS_JobType*` to *Production** and *Processing** jobs is reasonable and provides a representative data basis for the further evaluation.

Identification of Faulty Job Reports: In general, the central payload monitoring is very reliable. However, with all tools and complex systems working together in a computing grid, errors can always happen at different stages, as demonstrated above with the examples. Possible reasons are countless and mostly not reflected in the data itself. For this thesis, the effort to identify them stands in no relation to the outcome. Therefore, it is more feasible (and sufficient) to rather simply identify the wrong job reports and fix, if possible, or else exclude them from the further analyses. The entire process is described in detail in Section A.2. Additionally, a summary of the individual steps is given in Table A.3.

To provide a consistent, comparable basis, the data for all relevant WLCG sites – T1_DE_KIT, T2_DE_RWTH, KIT-HOREKA, KIT-T3, and RWTH-HPC – has been prepared for the following evaluations and comparisons accordingly.

5.6.2. Performance Factor Estimation

The comprehensive data preparation process provides a consistent data basis for a reliable grid site evaluation. In addition, for some measures, an additional performance factor is required to create absolute comparability of the measure between sites. Otherwise an independent assessment based on the quantities would not be conclusive, as described in Section 5.5.1.3.

A representative way that was introduced to rate the pure performance of a grid resource for HEP workloads is provided with the HS23 benchmarks. They provide an independent, absolute rating for a site, or a certain worker node configuration, to be more precise. In principle, this can be used as the performance factor p, see Eq. (5.6). However, it is only officially available for WLCG sites. For opportunistically integrated grid sites, like HoreKa, the contribution is not accounted for and no official performance score is provided. In some cases, it may be possible to run the benchmark suite to obtain the factor. But this only works if direct access to the resources is available, as they need whole nodes exclusively. This is rarely the case if one is not responsible for the resource.

As part of this work, a new, data-driven method was developed, solely based on the available payload monitoring data, to circumvent this problem. It takes advantage of the fact that, although direct derivation is impossible, the performance of a site can be evaluated relative to another one. Since many of the introduced measures are anyway most conclusive in comparison to other sites, this is absolutely no problem.

The method is based on the idea that the direct processing efficiency, P (Eq. (5.6)), for the same job from the same workflow on the same hardware should be identical – or at least within the range of statistical fluctuation – under the condition that the job was successful and running fully efficient. Because only under these circumstances, all invested CPU time is productive and differences in the outcome only depend on the pure performance of a resource. If the hardware now changes, the performance rating p changes, taking the performance impact on the quantity into account. This can be exploited to derive the relative performance rating of two similar jobs, job1 and job2, of the same workflow running on different resources, site1 and site2:

$$\frac{p_{\text{site1}}}{p_{\text{site2}}} = \frac{\mathrm{P}_{\text{job1}}}{\mathrm{P}_{\text{job2}}}\,, \tag{5.28}$$

under the condition that both were running at ideally 100 % CPU efficiency. Because in this case, the p ratio realistically reflects the difference in the direct processing efficiency of the two sites, as described above. To stabilize the result, all jobs of a

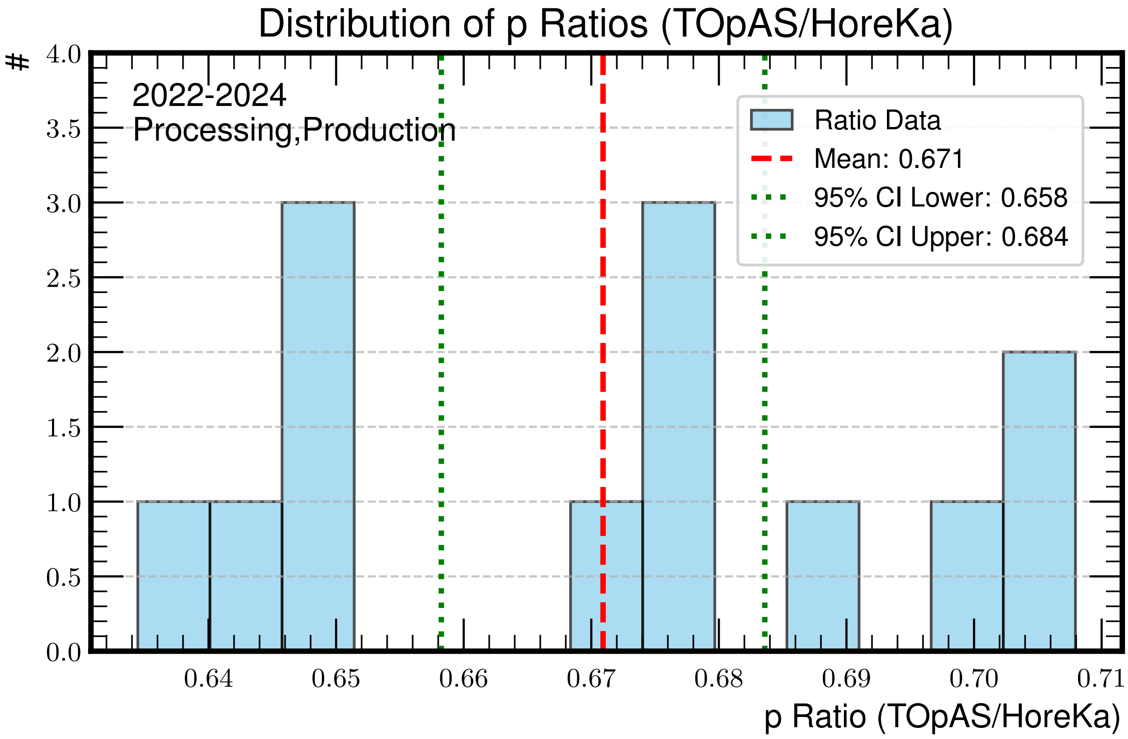


Figure 5.9.: Distribution of the performance ratios per workflow between TOpAS and HoreKa. Two unreasonable or faulty workflows were excluded. The full evaluation – including these outliers – is provided in Fig. A.37. The result is in good agreement with unofficial benchmarks for the sites [185], see Table 5.4. Accordingly, the p factors for all other evaluated sites are estimated. All p ratio distributions are provided in Section A.3.1.

workflow meeting these requirements can be used to calculate the ratio. Examples for this process are shown in Fig. A.34 and Fig. A.35.

To obtain even more reliable results, all workflows that meet the requirements can be included in the evaluation. The result is a distribution, as shown in Fig. 5.9, from which the final value for the ratio can be derived.

For all relevant grid sites, the method was used with TOpAS as a baseline, leading to the results given in Table 5.3. The full version of the table is provided in Table A.4, including an additional consistency check.

With TOpAS as the baseline (HS23 score: 13.5 [185]), an absolute value for p can eventually be derived using the results of the relative evaluation from the table. The resulting performance ratings, as equivalents to HS23, are given in Table 5.4. As the comparison shows, with TOpAS as baseline the newly derived method yields very good results that are in agreement with the HS23 scores. An additional consistency check, provided in Table A.5, shows that uncertainties of around 5-10 % are realistic. Here, it is emphasized once again that all examined sites run real production workflows for CMS. This means, in contrary to a dedicated benchmark, in no case are two exactly identical jobs running at two different sites. The result is therefore remarkable, especially in the context that the method is purely data-driven and does not require any explicit interventions.

The only larger deviation is observed for the Tier-2 in Aachen (RWTH). A detailed

Table 5.3.: The table shows the p ratios between TOpAS and each other site. Parentheses indicate the cases where only a small number of common workflows meeting the requirements to evaluate the performance ratio are available. Overall, the results are very consistent. The last column contains the values that are eventually used for further analyses – if not specified else.

Sites	2022	2023	2024	2022-2024
$p_{\text{TOpAS}}/p_{\text{HoreKa}}$	0.66 ± 0.02	(0.69 ± 0.01)	0.67 ± 0.02	0.68 ± 0.01
$p_{\text{TOpAS}}/p_{\text{GridKa}}$	1.05 ± 0.05	(1.01 ± 0.18)	0.98 ± 0.02	1.06 ± 0.05
$p_{\text{TOpAS}}/p_{\text{RWTH}}$	1.33 ± 0.07	(1.20 ± 0.13)	0.94 ± 0.05	1.09 ± 0.05
$p_{\text{TOpAS}}/p_{\text{RWTH-HPC}}$	(0.85 ± 0.03)	(0.88 ± 0.02)	(0.76 ± 0.02)	0.82 ± 0.02

Table 5.4.: The table shows the HS23 scores derived with the described method and TOpAS (bold) as the absolute basis for comparison. The up-to-date (2024) official values for GridKa and RWTH are taken from CRIC. In parentheses, HS23 values are provided which were privately derived with the official benchmark suite [185, 187]. The derived HS23 values are using the results for the entire period from 2022-2024, see Table 5.3.

Site	Official HS23	HS23 Equivalent
TOpAS	(**13.50**) [185]	**baseline**
HoreKa	(20.20) [185]	20.15
GridKa	12.75 [89]	12.74
RWTH	14.41 [89]	12.39
RWTH-HPC	(16.7) [187]	16.46
RWTH (2024)	14.41 [89]	14.36

explanation is provided in the full description of the method in Section A.3. In short, the hardware changed more drastically over the evaluated years, which is also reflected in a changing p ratio in Table 5.3 for RWTH over time. It is therefore simply an averaging effect – which should still yield valid results when evaluating the entire time period. If the current value (2024) is used for the calculation of the absolute performance factor, the agreement is again in extraordinarily good agreement, as the last line of Table 5.4. This underlines the consistency of the method.

In conclusion, the newly introduced, data-driven method for the pure performance evaluation of a grid site based on the production monitoring data is providing results in good agreement with official benchmarks. For (opportunistic) grid resources without official benchmark values, the method presents a plausible alternative, enabling a conclusive comparison. Especially for resources like RWTH, where the performance change over time is not realistically reflected in the officially available benchmarks scores from 2021 and 2024, a more accurate result can be obtained with the new method. Finally, this analysis shows the extraordinary potential that the HPC resources provided by the NHR Alliance have to offer. If they can be used efficiently, their integration represents an enormous enrichment for the HEP community.

5.6.3. Analysis of the Initial Phase of the Integration

With now everything at hand to provide a well-funded evaluation of the performance and reliability of a grid resource, the opportunistic integration of HoreKa can be reviewed. For this, all discussed measures from Section 5.5.1 are evaluated for the HPC center. From an independent evaluation, first conclusions can be drawn. But the most expressive results can be obtained when comparing the integration with other grid sites, as previously described. Therefore, the same analysis has been done for the WLCG sites, GridKa and the university Tier-2 center in Aachen (RWTH), as well as other opportunistic resources, TOpAS, the Tier-3 at KIT, and CLAIX (RWTH-HPC), the NHR center in Aachen.

The period under review covers the beginning of 2022 until the end of June 2023. This is referred to as initial phase (of the whole pilot phase for the pledged integration) in the following. During this time span, HoreKa was simply integrated as is – without any optimizations. It therefore serves as the baseline for the investigation of improvements, as described later.

The full evaluation of the entire period yields the results presented in Table 5.5. An analysis of all individual quantities is described below.

Classic Quantities: CPU Efficiency and Failure Rate Even if not absolutely representative, the two default, classical measures of the modern computing world still provide a good first impression on how well and reliable a grid resource is utilized and can serve as an indicator for problems.

What is immediately noticeable when comparing the absolute results for HoreKa with the others is that it is the only resource that sticks out with by far the lowest overall CPU efficiency (†). This may can be partly attributed to the fact that the proportion of *Processing** jobs at the site was the highest during this time period. Since the CPU efficiency on average for this job type is typically significantly lower (see exemplary Fig. B.7), this is an external factor that needs to be considered when

Table 5.5.: Total evaluation of all sites for the initial phase (January 2022 until end of June 2023). Here, all workflows of type *Production** and *Processing** are considered. The most interesting results are marked with symbols. For more details, it is referred to the text.

Measure	GridKa	RWTH	TOpAS	HoreKa	RWTH-HPC
Number of Jobs	5.04M	1.57M	262K	504K	418K
Fraction of *Processing**	19.7%	8.2%	27.0%	42.0%	6.4%
Fraction of *Production**	80.3%	91.8%	73.0%	58.0%	93.6%
$\text{CpuEff}_{\text{tot}}$ (Eq. (5.1))	83.7%	89.1%	82.8%	58.0%†	89.0%
$\text{CpuEff}_{\text{tot}}$ (*Processing**)	81.7%	82.6%	81.2%	48.6%	86.4%
$\text{CpuEff}_{\text{tot}}$ (*Production**)	83.9%	89.5%	83.1%	61.8%	89.1%
Failure Rate (Eq. (5.8))	8.3%	4.3%	5.6%	13.9%	40.1%§
$\text{CpuEff}_{\text{tot}}^{\text{corr}}$ (Eq. (5.10))	80.8%	86.5%	80.3%	51.8%	64.7%
$\text{CpuEff}_{\text{tot}}^{\text{corr}}$ (*Processing**)	76.0%	75.8%	73.2%	42.0%	79.1%
$\text{CpuEff}_{\text{tot}}^{\text{corr}}$ (*Production**)	81.3%	87.0%	81.4%	55.8%	64.0%
CommittedCoreHr*	104M	37.0M	3.68M	6.45M	7.19M
Inefficiency Loss I_{agg} (Eq. (5.17))	13.6M	3.68M	558K	1.97M	522K
Inefficiency Loss $I_{\text{agg}}^{\text{rel}}$ (Eq. (5.18))	13.1%	9.9%	15.2%	30.5%	7.3%
ComputeTimeLoss C_{agg} (Eq. (5.14))	6.35M	1.33M	166K	1.14M	2.0M
ComputeTimeLoss $C_{\text{agg}}^{\text{rel}}$(Eq. (5.13))	6.1%	3.0%	4.5%	17.8%	28.0%
Total Loss T^{rel} (Eq. (5.21))	19.2%	13.6%	19.7%	48.2%‡	35.3%‡

classifying the result for the HPC center. For a more in-depth investigation, the table also provides the total CPU efficiency for *Processing** and *Production** jobs separately for each site. Here is noticeable that the difference between the job types for all other sites but HoreKa is only in the lower single-percent range and therefore much smaller than expected. Also in comparison to the WLCG average, the German sites seem to perform extraordinarily well for processing jobs. For HoreKa, the difference between the job types is more significant with around 15 %. Overall, the efficiency is for both types significantly lower than for the sites in comparison. Therefore, the higher fraction of *Processing** jobs has an influence on the worse result but is definitely not the primary cause. For *Production** jobs only, the overall CPU efficiency is still between 20 and 30 % lower than for the other sites.

Important is that this result must not be misinterpreted as a general trend for the NHR/HPC centers. CLAIX/RWTH-HPC, in contrary, has a remarkable CPU efficiency that can compete with the WLCG sites over the full period of time, as shown in the top part of Fig. 5.10. In some periods, also HoreKa comes in range of the other sites, but overall, it is lacking significantly behind.

This shows that the evaluation of the CPU efficiency, even if not a fully representative quantity for the integration of a grid site when reviewed stand-alone, can definitely be beneficial. Here, it provides an indication that the reasons for the inefficiencies at HoreKa will likely be found in the differences of the two HPC centers. An in-depth investigation will follow later, as no reliable conclusions can be drawn from the simple evaluation of the standard quantities.

At the same time, this example underlines the dangers when using the CPU efficiency

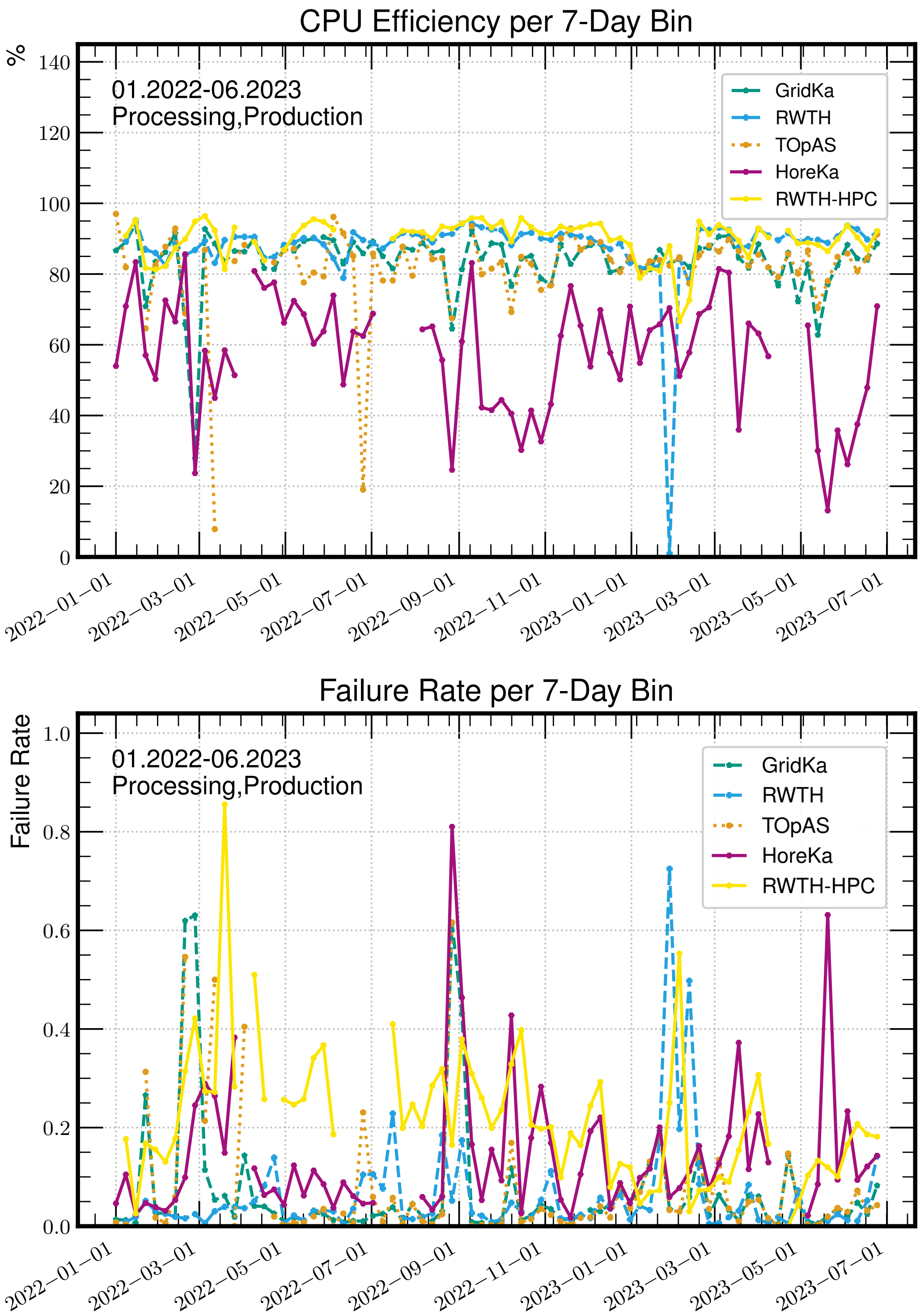


Figure 5.10.: The top figure shows the total CPU Efficiency distribution of all investigated sites during the initial phase of the integration. While CLAIX/RWTH-HPC can compete with the WLCG sites, HoreKa has clearly issues with the CPU efficiency. Possible reasons will be investigated later.
The failure rates for all investigated sites during the initial phase is displayed below. It shows that overall, the WLCG sites are much more reliable than the NHR centers. While HoreKa is still comparable in parts, RWTH-HPC has significant issues. However, what is remarkable is that the CPU efficiency for RWTH-HPC does not suffer from the high failure rate, meaning that the failing jobs were anyway highly efficient.

as a measure. Drawing the conclusion that a site is well integrated and efficiently utilized solely from the CPU efficiency is misleading, because it does not take the failure rate into account. For RWTH-HPC, the efficiency is great, but the fraction of failures is very high (§ in Table 5.5), which indicates reliability problems and, of course, has a significant influence on the overall performance of the resource. This is visualized in the bottom part of Fig. 5.10. When comparing the distribution with the CPU efficiency above, one can conclude that the failing jobs were anyway highly efficient. This shows that CPU efficiency and failure rate are not necessarily correlated, which is already an interesting and important finding from this basic analysis. The consequence is that an incautious interpretation of the quantities can lead to false conclusions on the integrated grid sites. Because the failing jobs actually did not provide any usable outcome. Therefore, the impact of the failures, is not reflected by the CPU efficiency alone, as described in Section 5.5.1.4.
For HoreKa, the failure rate is better, but still between 5 and 10 % higher than for the WLCG sites. A direct comparison combining both, the CPU efficiency and failure rate, as depicted in Fig. 5.11, leads therefore to the conclusion that there are issues with the integrated HPC site and may can help to identify the underlying problem. Solely from the CPU efficiency point of view, the opportunistically integrated HPC center therefore does not meet the expectations and goals. But as described above, this analysis is not telling the whole story and premature conclusions based on the classic quantities should be treated with caution.
Also the reasons for the job failures and inefficiencies, cannot be extracted from this basic evaluation. They can range from file read errors over software or even hardware problems. An identification and classification therefore requires a more complete picture and is the task of the site admins/ responsible personal[14].

Corrected CPU Efficiency: As the analysis of RWTH-HPC has shown, the classic CPU efficiency can be misleading. Because failed jobs technically still can be very performant, but the de facto CPU efficiency is 0 %, since the total invested CPU time does not provide any usable outcome. The corrected CPU efficiency (Eq. (5.10)) follows this philosophy and was accordingly derived for all sites. It is depicted in Fig. A.49 for the Tier-2 (RWTH), HoreKa, and CLAIX(RWTH-HPC). The difference between the normal and the corrected CPU efficiency is shown in Fig. 5.12. While for the WLCG sites the deviation is small, the impact of the correction on the NHR centers is significantly larger in big parts of the total time period. Especially for RWTH-HPC, the high failure rate is causing a drastic correction of the overall CPU efficiency of around 25 % on average. But also HoreKa is falling further back in comparison to the more reliable -2 in Aachen. For the WLCG grid sites and TOpAS, the difference is only of the order of 3 %, which is why they are neglected here.
This small analysis shows that the default CPU efficiency is not correctly reflecting reality and the introduced correction is enhancing the informative value of the quantity. With the corrected CPU efficiency, the failure rate is correctly taken into account and misleading interpretations for the overall grid site performance are avoided.
However, what is not yet considered is the runtime of the grid jobs. Because ultimately, this also has a huge influence on the resulting loss of compute time due to

[14] Here, the specialized knowledge of the computing teams of the university Tier-2 centers is required to fix problems and optimize the utilization of such resources.

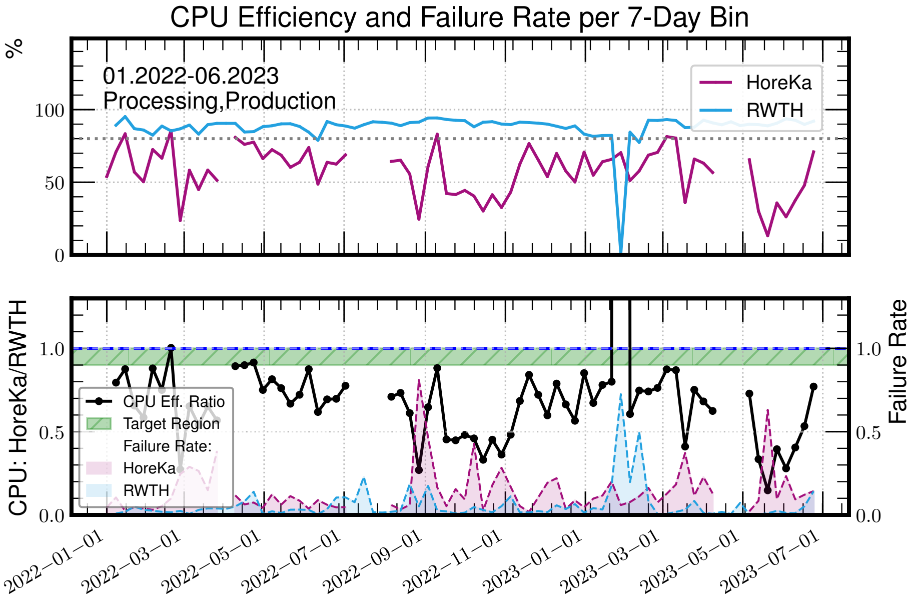


Figure 5.11.: Here, HoreKa and RWTH are compared directly in terms of CPU efficiency and failure rate. The gray, dotted line indicates the 80 % mark for the CPU efficiency. This shows that the Tier-2 center is performing very well, while HoreKa is mostly below the line. The CPU efficiency ratio is shown in the bottom part. The green area indicates the target region for a sufficiently well working integration in terms of CPU efficiency. The target for the failure rate is, of course, zero. Most of the time, HoreKa cannot reach the targets and does not meet the expectations.

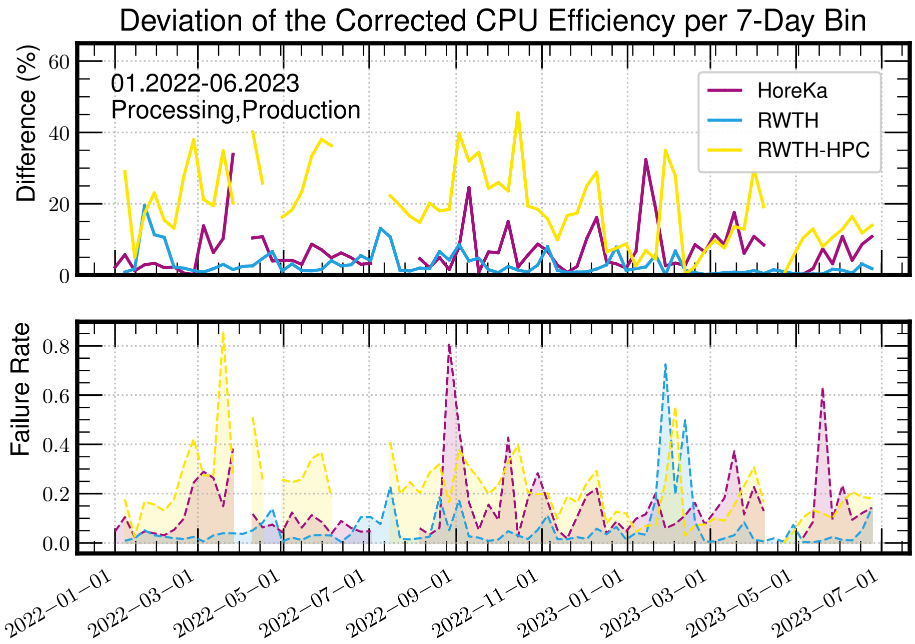


Figure 5.12.: The figure shows the difference between the CPU efficiency (Fig. 5.11, top) and the corrected CPU efficiency (Fig. A.49) for the two NHR centers and the WLCG Tier-2. Below, the failure rates are displayed. The other sites differ only slightly and have been omitted to provide a better overview. As clearly visible, considering the failure rates in the CPU efficiency evaluation has a significant impact on the result.

failures and inefficiencies. To integrate this into the evaluation, the different Losses were introduced in Section 5.5.1.7.

The Loss of Compute Time: The Total Loss, T (Eq. (5.20)) is composed of the loss due to inefficient processing (Eq. (5.16)) and the lost compute time due to failures (Eq. (5.14)). Both are a consequent progression from the classic measures that are used to quantify the impact of these quantities in absolute values, namely core hours that did not provide a productive outcome – and therefore are lost. The evaluation of the aggregated Compute Time Loss, C_{agg}, for HoreKa yields that the failure rate of 13.9 % is slightly underestimating the actual impact on the integration of the resource. Job failures are responsible for a loss of around 18 % of the invested compute power, as presented in Table 5.5. This is far from ideal and show that the integration of the NHR sites is clearly lagging behind. The dedicated WLCG sites, especially the Tier-2 in Aachen (RWTH), are outperforming the NHR centers in terms of reliability. Nevertheless, this must not be seen entirely negative, as it provides a good starting point for future optimizations.

The second contribution to the Total Loss is the amount of core hours that were lost due to an inefficient processing. This is described with the Inefficiency Loss, I_{agg} (Eq. (5.16)). As given in Table 5.5, with a loss of about 30 % of the invested compute performance, HoreKa is the inglorious winner. In comparison to RWTH, this value is around three times as high in relative terms. The significantly lower CPU efficiency for the HPC center takes its toll. This is far away from the target of being able to use the resource roughly as well as the dedicated WLCG sites, as set in Section 5.4. But

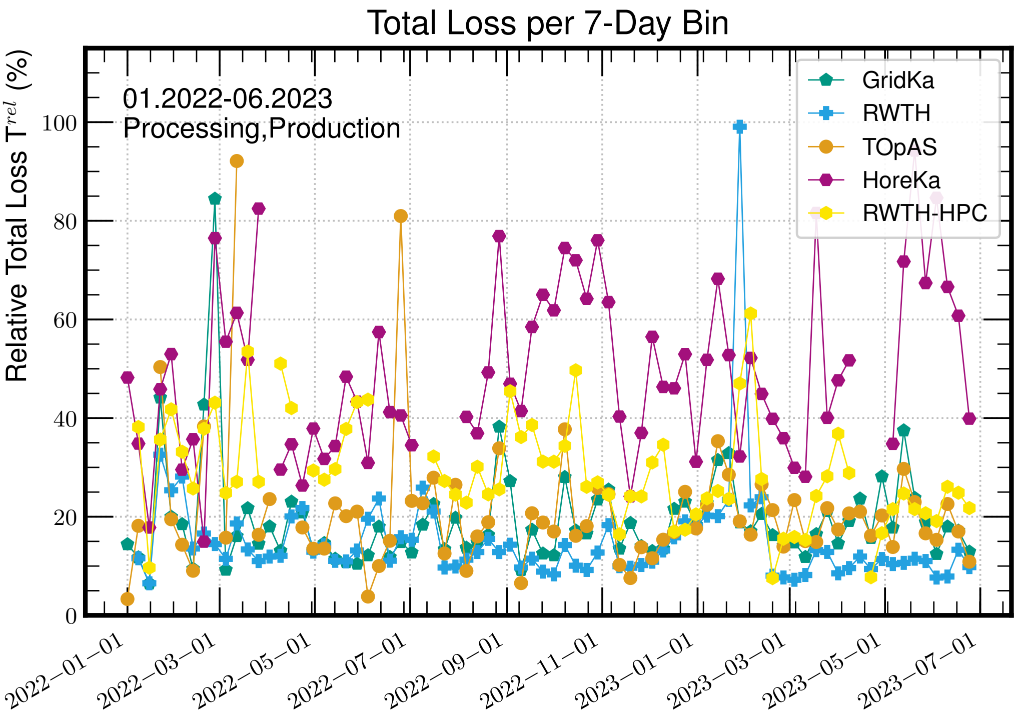


Figure 5.13.: The Total Loss combines the Inefficiency Loss and the Compute Time Loss. It is a measure that directly reflects the impact of inefficiencies and job failures onto the overall site performance. The WLCG sites and TOpAS show very similar results with a Total Loss of around 15 to 20 % of the invested compute time. For the NHR centers, the relative loss of the invested compute is overall higher. While HoreKa mainly suffers from the low CPU efficiency, the result for RWTH-HPC is degraded due to the high failure rate, both shown in Fig. A.50.

again, this information is very important as it provides insights on what has to be improved for the pledged integration in the future to achieve a more optimal result. Combining both losses forms the Total Loss, which is shown distributed over the entire initial phase in Fig. 5.13 for all investigated sites. How the Total Loss is composed of the two individual losses is evaluated in Fig. A.50, showing the three quantities per site.

The overall result for the NHR centers, marked with a ‡ in Table 5.5, is not satisfactory. This value shows that the combined impact of job failures and inefficiencies for HoreKa leads to a loss of nearly 50 % of the invested compute[15]. The integration therefore does not meet the expectations, as the NHR centers do not provide a comparable result to the classic grid sites. For the future, this clearly must be improved to ensure that the pledged integration of HPC will work out – especially in terms of sustainability and the responsible use of the given resources.

This demand is even stronger, if the performance potential of the HPC centers is additionally considered for the review. Because when used efficiently, the HPC centers could provide in principle the highest performance, as analyzed in the following part.

[15] In principle, this value can be converted to an Energy Loss, as described in Eq. (5.27). But since this information is not available for other sites to compare, it is refrained from a detailed analysis here.

Processing Efficiency: The processing efficiency[16], is directly reflecting the usable output in relation to the invested compute (see Section 5.5.1.3). It is a very interesting quantity, as it provides not only more tangible information on the return of invest, but also gives a hint on the maximum performance potential of a resource – and how much of it is used. That is very unique about this measure, since all others are mainly focused on how well a resource is used but do not provide information on how capable it is. Furthermore, it directly shows the impact of inefficiencies and failures on the produced outcome. This is a great benefit, since it adds informative value which is fully neglected when evaluating a grid site solely with CPU efficiency-based measures.

When evaluating the processing efficiency, comparability between different grid sites is only given if their pure compute performance is correctly taken into account. To derive a comparable performance rating of the investigated resources during the evaluated period, the method described in Section 5.6.2 (and more detailed in Section A.3) is used. As visible by comparing Table 5.6 to Table 5.4, this mainly has an impact on the evaluation of the Tier-2 in Aachen. By only considering the specific time period under investigation, the performance rating is more accurate compared to the averaged values. The other sites are in good agreement.

When it comes to comparing the processing efficiency between compute sites, another challenge arises. Different workloads can have a significantly different event throughput and the number of processed events, of course, is highly dependent on the actual task. This means, no absolute maximum value exists, making a direct comparison on job level not feasible. Because one job can be above average with only one produced event per core hour, while another one is maybe performing badly with thousands of less complex simulations per invested core hour. As a consequence, the absolute result is not comparable and cannot be easily combined for the evaluation of an entire grid site. To create comparability for the processing efficiency, the measure has therefore to be evaluated for the same workflow under the assumption that all associated jobs are sufficiently similar. In this case, the absolute values are conclusive and can be compared between different computing centers running this workflow. The review of the processing efficiency then correctly describes the throughput of the sites and becomes useful for their overall performance[17] evaluation.

A comparison of only one workflow, however, may still not realistically reflect how well a site is performing, as the influence of external factors is more dangerous for the result of a single one. To mitigate this, all workflows are selected for the evaluation that ran at all of the compared sites. Additionally, a threshold is applied to ensure reliable results: Only workflows are considered, of which at each site at least 10 jobs were executed. The impact of these two filtering steps is given in Table A.6.

The next step is to combine the results for several workflows to derive a more complete picture and draw an overall conclusion on how well different sites are performing. However, this cannot be done by simply averaging the processing efficiencies, since they are not normalized – and no universal normalization exists. As absolute quantities, the averaging would again be unrepresentative due to the

[16] As described in Section 5.5.1.3, this quantity is no real efficiency in the mathematical sense, since the throughput of events has no theoretical maximum. But it reflects how efficient a site is performing, which is why it is anyway called an efficiency.

[17] In the sense of how well a site is performing in general – not to be confused with the evaluation of the pure compute performance!

individual conditions of each workflow. The easiest way to create comparability is therefore to evaluate each workflow per site and normalize the result in relation to the best performing site.

This process is depicted in Fig. 5.14 for seven different workflows. The top part of the figure shows the aggregated processing efficiency per workflow, P_{agg} (see Section 5.5.1.3), without considering the performance factor p. Below, the result of the calculation for the same workflows is shown, but with the performance of each resource taken into account: $P_{agg}(p)$ (Eq. (5.7)). The labels on the x-axis are composed of the number of the workflow, which can be looked up in Listing A.6, and the normalization factor N, corresponding to the result of the best-performing site. When comparing these factors, the differences in throughput between the different workflows is clearly visible. Over the bars, the total CPU efficiency, CpuEff_{tot}, of the according workflow at the associated site is given. From this, already first conclusion on the integration and utilization of the different sites can be drawn.

The first thing that stands out from the top figure is that one or both of the NHR centers always achieve the highest direct processing efficiency per workflow. This shows the performance potential of the HPC resources and why the p factors are important for an expressive comparison of this quantity. But what is nevertheless remarkable with this direct comparison is workflow 3. Here, HoreKa has the lowest total CPU efficiency with a significant difference to the other sites, but nevertheless achieves the best result. This means that despite the low CPU efficiency, it produced the most events per core hour. In such a case, the less-efficiently used HPC resource provides still the same output like the official WLCG resources. This means, the outcome for the HEP community is the same. And under the assumption that the HPC center is more sustainable as a whole, on the total scale not even energy was wasted. However, what is definitely wasted is the potential that these resources provide. Because as clearly visible, if it would have been used more efficient, the result would outrun the others by far. And, of course, this does not meet the high standards for sustainability and the responsible use of the given resources, the HEP community sets itself. This, together with workflow seven indicates that there is room for improvement in terms of the efficient integration of HoreKa.

The second figure, where the same workflows are presented but with the pure compute performance considered through the performance factor p, better reflects the influence of the inefficiencies at HoreKa. As a consequence, the results are clearly shifted in favor of the dedicated WLCG sites. Only CLAIX/RWTH-HPC, which performs significantly better than HoreKa, can still keep up with the WLCG resources.

But this still does not ideally reflect the reality as job failures are not yet considered. To do so, the same analysis, but with the corrected quantities (Section 5.5.1.5), is visualized in Fig. A.51. It shows that with the failure rate considered, both NHR centers, but especially CLAIX, achieve a worse result and fall back behind the WLCG sites, which all have a comparably lower failure rate. Only for workflows with a good overall CPU efficiency, the results are still comparable.

Although the presented workflows provide interesting insights into the behavior and performance of the various grid sites, it has to be kept in mind that the small sample of 7 workflows is not meaningful for the overall evaluation. Tendencies may already be identified, but it is not possible to make definitive statements for the sites as the sample is not representative. To draw more reliable conclusions about how well

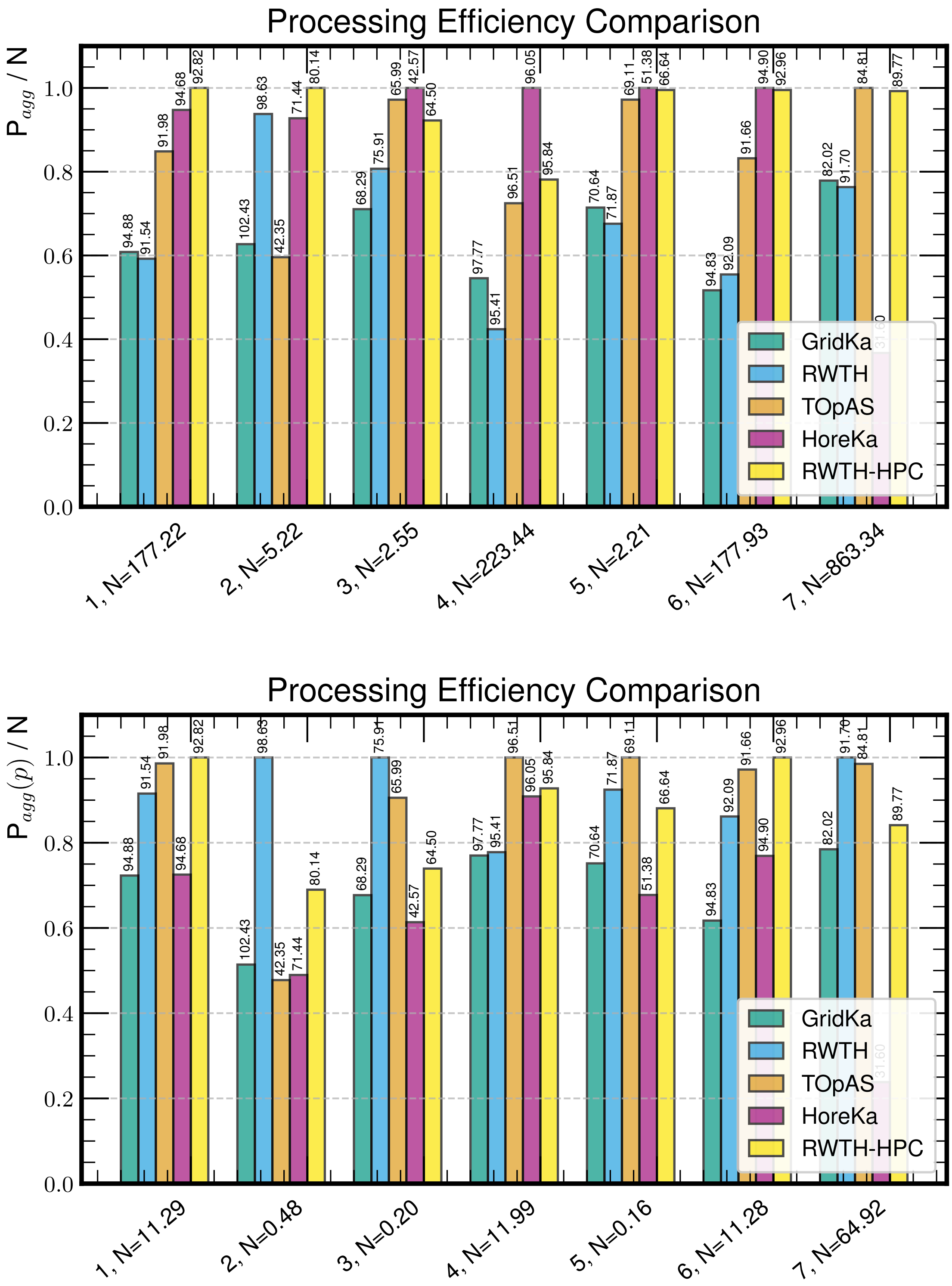


Figure 5.14.: Comparison of the aggregated, unweighted processing efficiency per workflow and site (top), and the aggregated, weighted ProcEff (bottom) for the same workflows. On the x-axis, the first number is representing the workflow, listed in Listing A.6. The second one is the normalization factor. Over the bars, the total CPU efficiency of each site for the according workflow is given.

Table 5.6.: Results for the evaluation of the aggregated corrected processing efficiency, $\mathrm{P}_{\mathrm{agg}}^{\mathrm{corr}}(p)$ (Eq. (5.9)), of all sites for the initial phase of the integration (January 2022 until end of June 2023). Only workflows of type *Production** and *Processing** are considered. For more robust results, it is required that each workflow had at least 10 jobs running at every evaluated site. The p factors were derived according to the introduced method (see Section 5.6.2 and Fig. A.48) for the period under investigation, with TOpAS as a baseline (see Table 5.4). The first row describes the average deviation from the optimum per workflow for every site, extracted from Fig. 5.15. On the second row, the result with proper workflow weights according to their CommittedCoreHr* share is given, extracted from Fig. A.52.

Measure	GridKa	RWTH	TOpAS	HoreKa	RWTH-HPC
Average Deviation	20.8%	8.9%	14.1%	50.7%	22.2%
Average Deviation (weighted)	17.4%	3.5%	16.6%	46.6%	24.9%

each site is working, the entirety of all workflows must be examined – 313 common workflows in total for the investigated time period. For this, the distribution of the aggregated, corrected processing efficiencies, $\mathrm{P}_{\mathrm{agg}}^{\mathrm{corr}}$, of all workflows, depicted in the upper part of Fig. 5.15, is analyzed. The resulting values are expressed as the average deviation from the optimum per site in Table 5.6, along with the results for the same evaluation when additionally adding workflow weights.

With those values is additionally considered that different workflows account for a different proportion of the total compute time and the results are weighted accordingly. Since the threshold is set rather small to get decent statistics, this helps to mitigate outliers and derive a more robust result with the analysis. The according distribution and box plot is presented in Fig. A.52. But as visible from the table and the comparison of the box plots, in this case, the difference is rather small and the conclusions drawn below do not change.

From the histograms in Fig. 5.15, it is clearly visible that the Tier-2 in Aachen (RWTH) is overall performing best in terms of processed events per invested core hour. For around 50 % of the workflows, the site had the highest processing efficiency, as shown in the separated 1.0-bin next to the distribution.

Furthermore, the figure shows significant differences in the distributions between HoreKa and the other sites. The processing efficiency for the HPC center is distributed nearly constant over the full range. This is reflected in a mean of 0.49 and the inter-quartile range spanning symmetrically a total of around 50 % of the full range. Such a result is clearly indicating issues with the integration. However, it seems to be strongly workflow dependent. About 50 % of all workflows perform significantly worse, while the other half is comparable with the other sites. The reasons for such a behavior can be manifold and will be investigated in the next chapter.

CLAIX/RWTH-HPC, the other NHR center, is competing well. This can be clearly observed in the box plot (bottom). Mean and box are comparable to the WLCG sites. Only the tail of the distribution is longer for the HPC center. This is mainly caused by considering the job failures (see § in Table 5.5), which alter the result for the site. The same distribution without accounting for failures, provided in Fig. A.53, does not

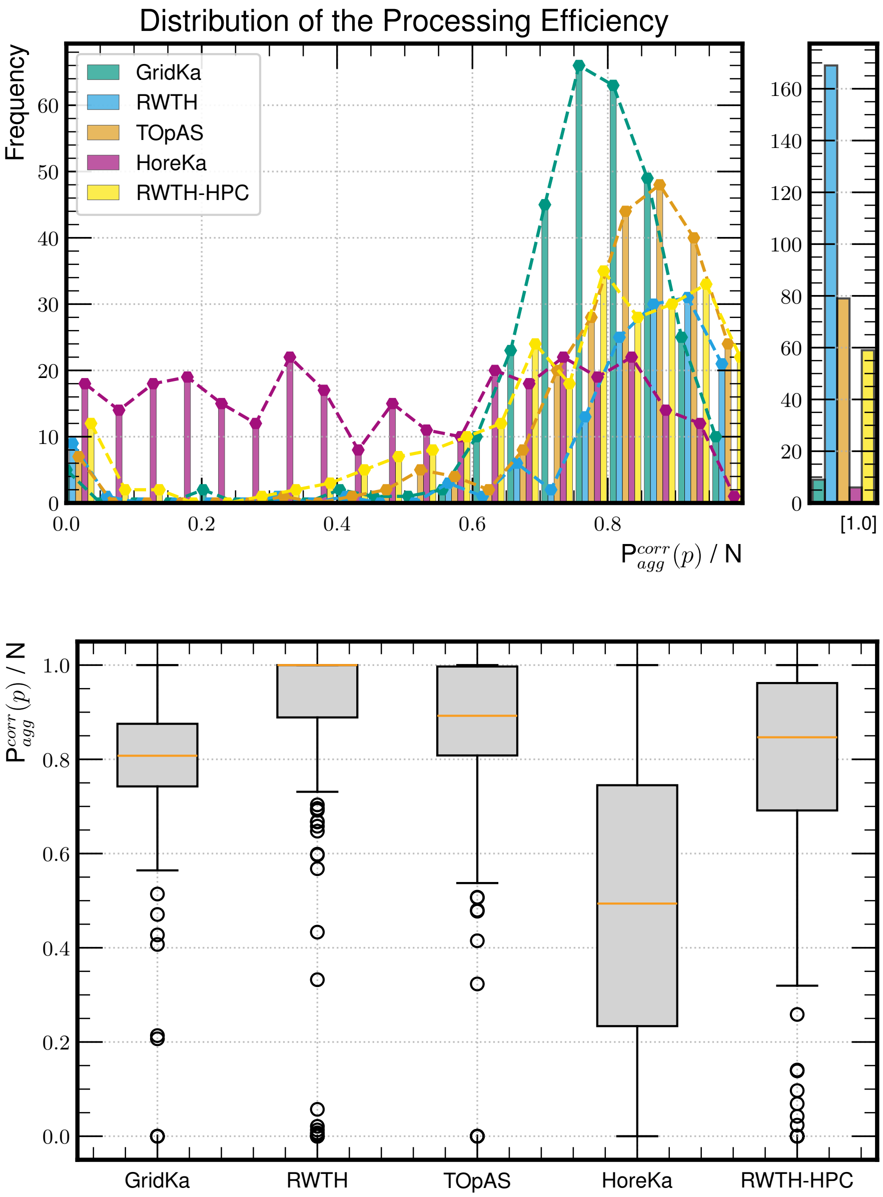


Figure 5.15.: Distribution of the corrected processing efficiencies aggregated per workflow for all reviewed sites. For a better illustration, the 1.0 bin is shown separately, which depicts the balance of the best results per workflow between the sites. The according box plots are depicted below.
Means of the distributions are given in Table 5.6 (as average deviation from the optimum). With the orange bars, the medians of the distributions are given. HoreKa is here the clear outlier. The boxes indicate the full inter-quartile range, spanning form the 25th percentile to the 75th percentile, containing the middle 50 % of the values. This area indicates the central tendency of the distribution, as well as the spread. While CLAIX/RWTH-HPC is competing well with the WLCG sites, HoreKa has a significantly wider spread. The whiskers span from the minimum to the maximum value within 1.5 times the inter-quartile range. This area describes the range of values in the distribution that are still considered typical. For HoreKa, it goes over the full width, clearly indicating problems – which is also reflected in the flat distribution in the top figure. The circles represent outliers of the distribution that are not contained within the range of typical values. In Fig. A.52, different workflow weights are additionally considered.

show the pronounced tail. However, despite the highest failure rate of about 40 %, the final outcome is still comparable – which again shows, why the classic quantities are not fully conclusive!

To summarize, this evaluation has demonstrated that the HPC centers are in principle able to compete with the classic grid sites, as demonstrated with CLAIX/RWTH-HPC. While performance and efficiency are comparable – or even better – for the NHR site, reliability still falls short. This is no ultimate game-breaker, since the resource is providing a similar throughput of events per core hour in comparison to the WLCG sites, but it is still unfortunate, as it could be better.

In case of HoreKa, the analysis has indicated that there seem to be underlying problems with the integration. While around half of the workflows were performing well – or at least comparable to the WLCG sites – inefficiencies of the other half degraded the overall performance of the resource during the initial phase of the integration. However, this must not lead to the conclusion that the integration of the HPC center has failed. Because even if it looks at first glance as if the site is performing worse, this is only in relation to the possible optimum. The total outcome of HoreKa is still comparable with the other sites! Here, the site clearly profits a lot from the high performance of HPC resources. This is shown in Fig. A.55, where no weighting with the pure performance (p) is done – and also holds with the failure rate considered (Fig. A.57). From the experiment's perspective, the integration is therefore performing on a very comparable level to the WLCG grid sites. But of course, it falls definitely short behind the expectations and self-set goals.

From the sustainability point of view, rating the integration is therefore very challenging. On the one hand, a large fraction of the workflows performed significantly worse than possible, leading to a waste of compute performance and potential. Because if the integration would have worked better, much more could have been achieved with the HPC resources. And the responsible usage of the given resources by the HEP community also demands that the utilization has to be better. But on the other hand, one could also argue that the overall climate impact is nevertheless reduced. Because a comparable outcome was achieved per invested core hours, which is beneficial under the well-justified assumption that the large-scale HPC resources are more energy efficient and sustainable in comparison to several smaller computing centers.

In the big picture, the result is therefore not ideal, but does not represent a major loss either. Especially, when additionally considering that HoreKa was used as an opportunistic resource and the HEP jobs were occupying idle resources. As argued in Section 5.4, even the inefficient utilization with a small usable outcome can be more sustainable than not using the resources at all. Only if the resource is poorly used when it could be used more efficiently by someone else does this manifest in a de facto loss.

Nevertheless, the study has shown that research and development for further improvements are suggested to enhance reliability and efficiency for eventually exploiting the full potential of the HPC resources. Also from the resource provider's perspective, who insists on an efficient utilization. Because for the pledged integration of the NHR resources, of course, higher demands and standards will apply.

5.6.4. Summary

The evaluation of the initial phase of the opportunistic integration of HoreKa from 2022 to June 2023 has brought many interesting things to light. First of all, it proved the enhanced informative value of the newly introduced measures for the evaluation of a grid site. While an analysis based solely on the classic quantities – CPU efficiency and failure rate – is not entirely conclusive and can sometimes be misleading, the enhanced measures provide a more expressive and reliable rating. Furthermore, they also cover various additional aspects that cannot be captured using the traditional metrics.

The application of the new measures to the opportunistically integrated NHR centers has shown that they could not entirely compete with the classic grid sites in terms of reliability and efficiency during the initial phase of the integration. While the official WLCG sites, GridKa and the Tier-2 center at RWTH Aachen, as well as TOpAS, the Tier-3 at KIT, performed extraordinarily well, the HPC resources fell behind in some aspects. Especially in terms of the Total Loss, which quantifies the impact of failures and inefficiencies. For the WLCG sites, the loss of compute time was overall only between 10 and 20 % – which also marks the optimal target for the future pledged integration. An even more efficient utilization than the dedicated sites is rather unrealistic. In case of the NHR centers, the according evaluation showed that the overall loss sums up to nearly 50 % of the invested compute power without any usable output. However, despite the actual cause for failures and inefficiencies could not be derived directly from the monitoring data, this analysis has also shown that the problems for both NHR centers were different. The result must therefore not be used as an argument against the utilization of HPC resources in general, because each center can compete in one of the classical quantities with the dedicated WLCG sites. A comparable utilization therefore must be possible in the future.

In summary, the initial testing phase can still be seen as a great success because of the following reasons:

Firstly, it has shown that the NHR resources are usable for HEP research and the centers can already compete with the classic grid sites in some aspects without any further optimization during the pilot phase. Secondly, even from the sustainability point of view the initial phase can be reviewed positively. It has to be kept in mind that the opportunistic integration is only utilizing unused resources. The requirements on performance and sustainability are therefore reduced, as explained above. Hence, the statement of the loss can be turned around. At least about 50 % of the compute time was converted into productive outcome for the progress of science, which would otherwise have been lost.

This argument, however, does not count anymore with the pledged integration, which requires the sites to perform better to avoid an actual waste of resources. From the WLCG perspective, as well as the resource provider's point of view, a smaller loss is absolutely desired. This is exactly where the the next positive point comes into play:

Thirdly, the opportunistic integration of the NHR centers allowed to collect and analyze monitoring data which is extremely valuable as basis for the future enhancement of the utilization of such untypical grid resources. Because only by identifying problems, a solution resulting in an improvement can be developed. The in-depth analysis based on the different quantities gave hints to different problems that will be addressed later.

And lastly, the evaluation of the processing efficiencies has shown that the actual throughput of the HPC centers was still comparable to the official grid sites, despite the reduced efficiency and reliability. The analysis therefore gives a good impression on the enormous potential of the NHR resources. If utilized better, they can be extremely beneficial to cope with the expectedly rising demand for compute power in HL-LHC era. This can therefore be seen as clear confirmation that it is reasonable to stick to the future strategy and invest in research and development in order to utilize the full potential of the HPC resources.
Possible solutions for workflow and efficiency optimizations – and therefore a better utilization of HPC centers – will be addressed in the next chapter.

6. Optimizations for the Utilization of HPC Resources

> “It’s quite simple, just follow the dotted line," the Planmaker explained. "Don’t let any bad idea lead you astray. Don’t let them persuade you to take a short cut or take one yourself. Life is a winding path. One sometimes has to make detours. That’s my humble opinion, anyway.”
>
> (The 13½ Lives of Captain Bluebear – Walter Moers)

In the previous chapter, the idea and concept behind the integration of HPC centers to provide parts of the national contributions of compute power to the WLCG were presented. Additionally, the challenges and limitations of such an endeavor were discussed and quantities for a meaningful evaluation were introduced.

The subsequent evaluation of the integration of HoreKa and CLAIX as opportunistic resources has shown that the two NHR centers offer enormous potential, but cannot yet be ideally used for HEP workloads. They still lag behind traditional grid sites in terms of reliability and the efficient utilization. These discrepancies, however, are due to various reasons. While CLAIX/RWTH-HPC mainly suffered from reliability issues, HoreKa struggled with a rather inefficient utilization of the resource. Optimizing the usage of these kind of resources is the designated goal for the future of HEP computing in Germany to be prepared for the HL-LHC era, but represents ongoing challenges.

The first part of this chapter investigates the problems and limitations leading to the fallback of the HPC resources behind the dedicated WLCG sites. Here, the focus lies on HoreKa, as the reasons for the increased failure rate at CLAIX cannot be derived solely from the accessible payload monitoring, as described before.

The section after is dedicated to concepts that could enhance the situation. Based on the findings, adequate measures and optimizations are described that may be capable of helping to achieve the objectives described in Section 5.4. This includes conceptual ideas for a better utilization of resources in general and a practical, XRootD-based approach for improving the efficiency of the HPC center.

In the last part, a prototype implementation for workflow and efficiency optimizations at HoreKa is presented and evaluated. Subsequently, conclusions for the pledged integration of NHR resources are drawn based on the findings.

6.1. Identification of Bottlenecks and Limitations

The identification of problems and possible underlying technical constraints is the basis for improvements. While the observation and identification of limitations is relatively easy, tracking down the reasons is often hindered by operational hurdles – especially on HPC centers. The limited monitoring capabilities, especially on hardware level, in combination with a full production system increase the complexity enormously. In case of HoreKa, for example, this is problematic, because the resource is mainly lacking behind in terms of CPU efficiency. The number one reason for inefficiencies are typically remote transfer and data access limitations, including the overhead for the software provisioning, as described in Section 4.2.4.

When having a look at the hardware setup of HoreKa, depicted in Fig. 4.23, the most probable limitation can be identified in the connectivity. Every worker node only offers a 1 Gbit/s link to the Internet, which has to handle all the remote transfers. When considering that twelve 8-core job pilots can accept CMS jobs per node, and potentially gigabytes of remote data are necessary for a single job, this can easily become a bottleneck. This assumption is also in line with the comparison between HoreKa and CLAIX/RWTH-HPC. The NHR center in Aachen provides 10 Gbit/s per node and does not have any problems with inefficient job processing, as shown in Fig. 5.10.

A challenge is, how to actually investigate and prove this well-founded hypothesis solely based on independent job monitoring data? At HoreKa, no I/O monitoring of the in- and outgoing connections is allowed for the worker nodes[1]. This makes the situation considerably more difficult, because as for every connection, both partners can be the limiting factor. Especially when data is accessed at the other side of the world, the transfer rates are often low. It is therefore hardly possible to differentiate, e.g., between the saturation of a network link and a slow remote server without this information.

Additionally, not every loss of efficiency can be accounted to that. Inefficient software and also misconfiguration can lead to similar effects. For example, it can sometimes happen that an 8-core pilot is only occupied by one single core job – maybe because of over-booking or due to the out-running of pilots, as described in Section 5.5.2. These effects degrade the CPU efficiency without being I/O limited.

The problem is that all effects are mixing up and interfering and a disentanglement is impossible on job level – or at least not feasible. This also means, a filtering after e.g. the `Workflow*`, the `CMS_JobType*`, or the number of utilized cores cannot exclude the interference effects. Or in general: In a complex production system, like the WLCG, it can never be guaranteed that the result is not altered through jobs running in parallel that are not within the selection, but e.g. anyway saturate the link of a server.

In case of HoreKa, the tendency is very clear, but to actually track down the underlying problem is a very complicated endeavor with the limited monitoring capabilities on such HPC centers. The best way to find reliable indicators is therefore to check the correlations of relevant quantities and compare them with other sites. Based on the in-depth analyses of the integration during the initial phase (January 2022 to end of June 2023) from the previous chapter, it is possible to pinpoint areas where the

[1] Only internal InfiniBand monitoring is permitted and provided by the central job monitoring system of the cluster.

integration falls short.

The most natural thing to check first is the correlation between the CPU efficiency and the fraction of the read time from the total runtime. If the two quantities are not correlated, an I/O limitation is unlikely. A strong correlation, in turn, would mean that possible degradations of the CPU efficiency are likely to be directly related to the transfer of remote data.

For TOpAS and HoreKa, clear differences are visible, as depicted in Fig. 6.1. When investigating the distribution for TOpAS (top), the CPU efficiency is going down linear with the read time fraction and the behavior is in line with the amount of transferred data, as shown in Fig. B.10. Nevertheless, also a significant fraction of jobs is not I/O bound but inefficient (below the maximum values in the diagonal). These jobs do not have a large read time fraction and therefore suffer from other – most probably – intrinsic inefficiencies, which again indicates that several effects are overlapping. With a Pearson correlation factor of around

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \cdot \sqrt{\sum(y_i - \bar{y})^2}} = -0.4\,, \tag{6.1}$$

this indicates that the site has overall no strong I/O limitation, which is also reflected in the CPU efficiency above 80 %. Some jobs may still be I/O limited, but as mentioned before, this can also be caused by the remote servers.

For HoreKa (bottom), a stronger correlation is visible in Fig. 6.1. The correlation factor of $R_{xy} = -0.6$ indicates that I/O has a bigger influence on the overall CPU efficiency of the site. But even more importantly, the fraction of the read time from the total core hours ranges up to 80 %. This is a clear indication that I/O limitation makes up a significant part of the CPU efficiency reduction. Especially with the on average smaller amount of transferred data in comparison to TOpAS, this supports the statement. Additional information on the average read times and data transfers is given in Table B.3. But from the monitoring data on job level without additional information, the actual impact is impossible to derive. Additionally, there are still many jobs with a small read time fraction that are nevertheless inefficient. This means that there must be multiple effects overlaying. From the comparison, however, it can be derived that this is not an exclusive problem of HoreKa, because for TOpAS, the same is visible. The most probable explanation are external factors not directly related to the site, like intrinsically inefficient jobs that are not I/O bound, or failing jobs that e.g. run in a timeout.

Ultimately, when reviewing and comparing the correlation per workflow, significant differences can be observed. The comparison is given in Fig. 6.2. While TOpAS has some workflows that show a strong correlation of the CPU efficiency to the read time fraction, most of them do not indicate an I/O limitation[2]. At HoreKa, most of the workflows show a strong correlation of CPU efficiency and read time fraction, indicating that many workflows are I/O bound.

In conclusion, there are several indications that the CPU efficiency at HoreKa is suffering from data access bottlenecks due to the comparably slow external connectivity of the worker nodes. To prepare the site for the pledged integration, the investigation of mitigation strategies is therefore an important step forward to improve the utilization and make the HPC site ready for the future.

[2] Some even show a negative correlation, which is implausible and therefore neglected in the evaluation. Most probably, this is caused by a faulty reporting.

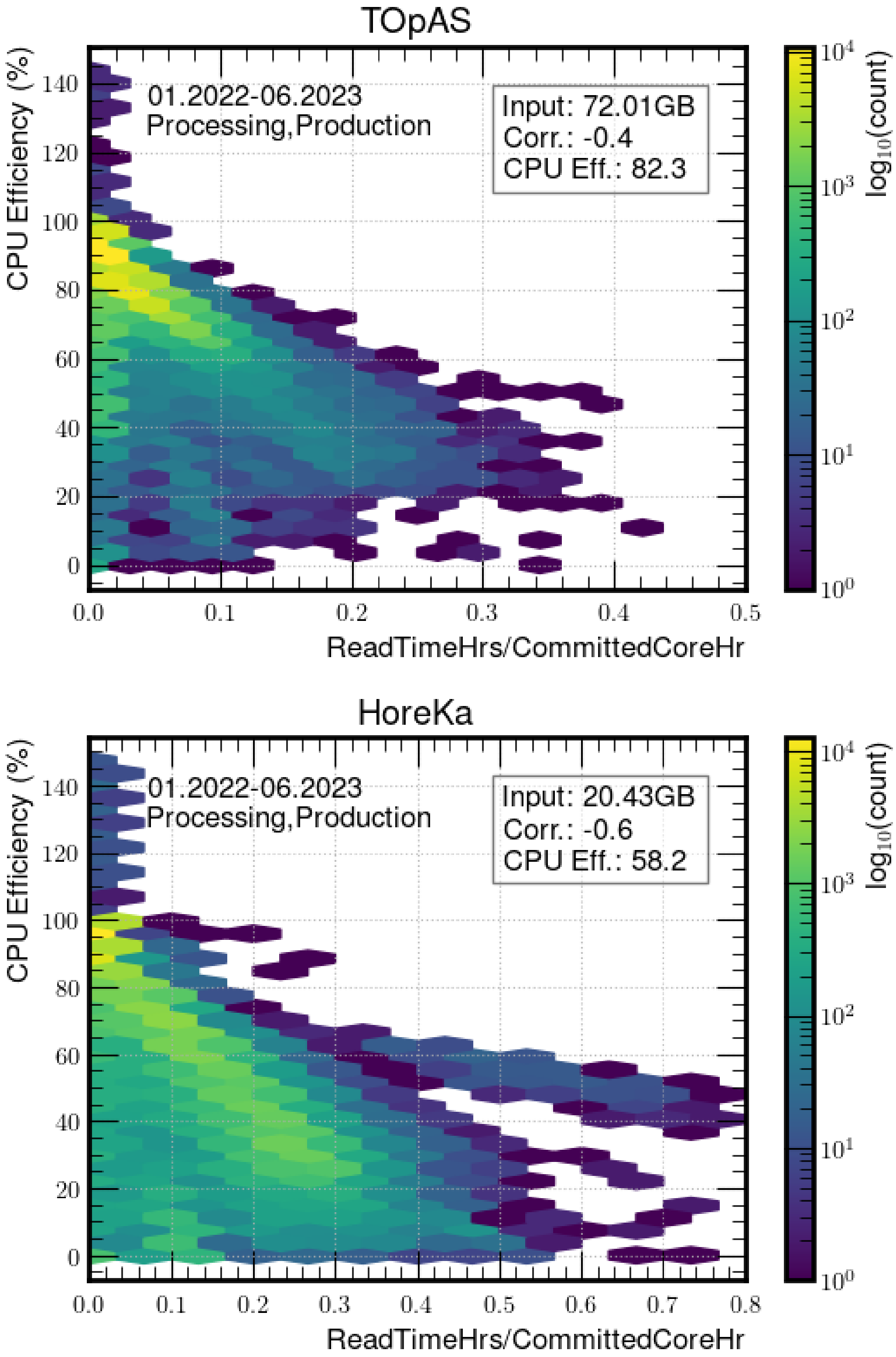


Figure 6.1.: Correlation between the CPU efficiency and the fraction of the read time from the total job time. For HoreKa, the correlation is stronger, indicating a bigger I/O limitation. However, below the diagonal line of a perfect correlation, still a significant amount of jobs can be found. This indicates that there are multiple reasons for inefficiencies. The same is presented for the Aachen sites in Fig. B.9.

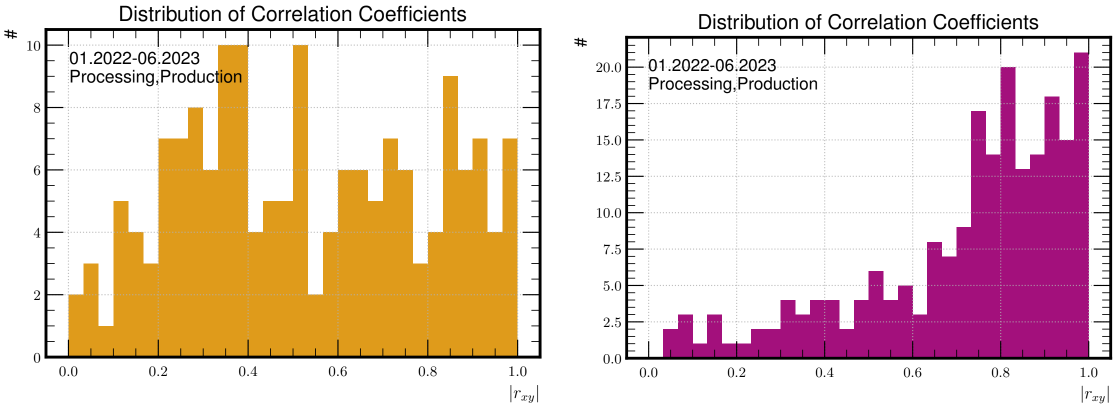


Figure 6.2.: These histograms show the distribution of the correlation factors (Eq. (6.1)) according to Fig. 6.1, but derived per workflow running at TOpAS (left) and HoreKa (right). A difference is clearly visible. While for HoreKa, a significant fraction of the workflows shows a strong correlation, which indicates an I/O limitation, TOpAS has a more equal distribution.
Note: For TOpAS and HoreKa, 49 and 18 workflows respectively did not provide a plausible result and are excluded from the evaluation.

6.2. Optimization Strategies

The results from the evaluation of the initial phase and the investigation of bottlenecks have shown that HoreKa cannot compete in terms of efficiency and reliability with the dedicated grid sites. At the same time, even higher expectations are attached to the pledged integration of the HPC centers, when they are supposed to make a part of the national WLCG contributions. Because on the one hand, the performance potential of the NHR centers is extremely valuable to be flexible and prepared for a future with a strongly rising demand for compute resources. And on the other hand, it will be very important to minimize the waste of compute power in the future. Not only to provide sufficient resource for the growing demand, but also with regard to sustainability and the responsibility of the scientific community. Investigating possible strategies for an enhanced utilization is therefore essential for the HEP community, as well as for the resources providers. Consequently, improving and optimizing the utilization of HoreKa is an important task and the basis for the success of the future computing strategy. Only if the site is performing comparably well, a migration from the dedicated university Tier-2 centers is well justified.

A good starting point for an optimization is the I/O limitation as discovered in the previous analysis of the integration. However, this directly reveals a downside of the transition. Because naturally, if the data processing is I/O bound, the logical consequence would be an upgrade of the external links. In case of the NHR centers, this is of course not possible, as there is no direct influence on the hardware. It is therefore not possible to simply increase the bandwidth per worker node to avoid data access bottlenecks. Instead, other, less intrusive ways must be found to improve how well an HPC site can be used for HEP workloads.

Some purely software-based options that do not require an in-depth intervention are discussed in the following.

6.2.1. Prefetching of Data

One possibility to mitigate a data access bottleneck and enhance the efficiency of a resource would be to prefetch the required data. When data is constantly transferred to the HPC site in advance and jobs only start running when everything is locally available, the efficiency of the resources could benefit. This is mainly the concept which the ATLAS Collaboration uses for its computing model. With Rucio, the data could be virtually placed at the HPC site and jobs start running, when everything is available.

Even if this approach can definitely be beneficial, there are several arguments against using it for CMS data processing in the grid: First of all, CMSSW, the CMS software framework, is tailored to optimize data access. The production jobs heavily use the streaming capability of XRootD, meaning they open a file remotely and only access the parts of it that are required. By this, the overall amount of data that needs to be transferred is greatly reduced. If the whole files were to be prefetched before starting a job, the total transferred data would accordingly rise significantly. This is, of course, not ideal when a site is already I/O limited.

Additionally, another problem arises in this context. The NHR centers will mostly be used for MC production and processing, like the university Tier-2 centers, which requires PREMIX data. By reusing the background event data instead of additionally simulating it, CPU power is saved – however at the cost of additional data transfers. Combining this optimization with prefetching unfortunately becomes unfeasible due to two reasons. Firstly, a grid job typically accesses – sometimes only small – parts of several PREMIX files. As a consequence, much more data must be transferred in advance than what is used in the end. And secondly, which data is required is decided just-in-time. This means, to actually use prefetching for the MC production would require the entire dataset to be kept available. However, this is unfavorable for multiple reasons. The PREMIX datasets can be of the order of petabytes and several different production campaigns can run at the same time. Consequently, several petabytes of storage would be necessary to benefit from the mechanism. On top, these large amount of data would need to be transferred in advance, which again requires a high bandwidth to work well.

What makes the situation even more challenging is that one grid site usually only gets a small fraction of all jobs from a production campaign. This means that chances are high that many of the prefetched files are never used, which makes it inefficient – storage and data transfer wise.

To summarize, although prefetching can be advantageous in principle, it is not suitable for mitigating the observed limitations for CMS workflows. The concept is not compatible with the computing model and would introduce additional challenges and inefficiencies. To fully benefit from such an approach, it would require significant changes in the CMS workflow management and scheduling infrastructure, eventually excluding it from being a feasible option.

6.2.2. Caching and Cache-Aware Scheduling

A typical approach when it comes to the mitigation of data access limitations is caching. It is a broadly used concept that saves exabyte of data transfers in the Internet every year. In HEP computing, it is also already widely used. CVMFS, for example, highly relies on caching. The principle is rather simple: The transferred data is stored on a local, volatile filesystem and can be accessed from there, if required again. This is absolutely useful, not only for the accelerated transfer, but also because it spares the remote data servers. The concept can also be very useful for data processing in a complex computing infrastructure such as the WLCG with globally distributed data to reduce the overall load and optimize data access. However, several things need to be considered.

In general, for optimizing data access by caching, the most relevant measure is the cache hit rate. Only if a decent amount of data is actually required again the overall system can profit from maintaining a local cache. This directly describes a first significant limitation of the concept when it is used for HEP data processing: Overall, it is rather unlikely that two production jobs are requiring the same data within a reasonable period of time. The fact that CMS jobs are already optimized to transfer only the data that is actually needed makes it even more difficult. Because when only parts of a file are cached, this further reduces the probability of achieving a cache hit. One way to optimize the cache hit rate in this sense could be to transfer slightly more of a file than what actually is required to increase the chance of making data available that may be used later. For PREMIX files, this approach could work out. However, it has to be tuned very carefully, especially when I/O is already limiting. Otherwise it could have the opposite effect and saturate the links even more.

Additionally, it has to be considered that in this case also more storage space is required. In order to be able to use the concept effectively, the cache size must be selected appropriately. However, cache sizes are usually limited and cannot be chosen freely – in case of HoreKa 250 TB are available. This is in principle already a decent size for a cache[3], but it always has to be set in relation to the processed data. Because if e.g. PREMIX data is cached, this makes up only 25 % of a full dataset[4]. The expected cache hit rate is therefore rather small and there is a relatively quick risk that cached data will be overwritten before it can be used again. In addition, different campaigns are often running in parallel at a grid site, which significantly increases the total amount of data in the pipeline and further increases this risk. Unfortunately, the turnover rate of a cache can only be optimized with a larger storage or selective caching decisions. If e.g. only smaller analysis datasets are cached, the turnover rate is way lower as the cache size is better proportioned to the absolute amount of data. Therefore, a valid option for the optimization could be to exclude large datasets and types, like PREMIX or RAW, and selectively cache only files that are known to be accessed regularly[5]. This can be for example fully pre-processed datasets (typically NanoAOD, in case of CMS) that are investigated in end-user analyses. Such a restriction was already successfully tested in the past, see Ref. [188]. With them, a decent fraction can be cached and the chances are high that another analysis user job requests the same data. Of course, the benefit ultimately scales with the fraction of *Analysis** jobs running at a cite that is utilizing such a system.

[3] Of course, this depends on the circumstances, such as the number of available cores. More information can be found, e.g., in Ref. [188].

[4] Remark: Not all PREMIX datasets are petabytes in size. But some are and this should be seen as an example.

[5] But it also has to be kept in mind that the information on what data is popular at the moment has to be integrated dynamically into the system, which is an additional, non-trivial effort.

A software-based alternative to increase the cache hit rate could be the implementation of a data-aware scheduling mechanism that also considers the currently cached data. In principle, CMS is already taking data locality into account for the scheduling decisions. But this considers only datasets that are stored longer term at a certain site. What makes things even more complicated is that HoreKa is only a sub-site of GridKa and does not provide any permanent storage. As a result, there is no data stored permanently that could be used for a scheduling decision based on data-awareness. Instead, the scheduling decision is limited to the main site (GridKa). From there, a job is routed over the Cloud CEs without considering the data locality beyond the Tier-1 level.

Accordingly, the general idea behind a cache-aware scheduling approach is to extend this concept to additionally consider short-term data at volatile caches. If jobs that require the same data could be actively directed to the same site where the file is potentially already cached, the cache hit rate would increase. However, there are a few technical difficulties and challenges to overcome in order to achieve this. At first, knowledge about the locally cached data is necessary. But directly tracking everything, e.g. in a database, is not feasible and does not scale horizontally. In addition, when many jobs are running, the cache state can change fast, making the scheduling even more complicated. What is also problematic in case of CMS jobs is that only parts of files are transferred – and therefore cached – by default. Additionally tracking which blocks of a certain file are available makes the endeavor totally unfeasible, while caching the whole files instead is again less efficient.

One way around this hurdles could be the approach of Virtual Placement [189, 190], as utilized by the ATLAS collaboration. The concept is based on XCache and was developed to enable the use of storage-less sites and caches in the ATLAS computing model, which is heavily relying on prefetching. Because on default, the ATLAS jobs are only scheduled when the required data is available, which, of course, cannot be reconciled with the concept of a volatile cache. If a site does not provide an addressable storage, the standard scheduling mechanism does not work as intended. Accordingly, a mechanism is needed that makes the jobs believe that the required files are available at a certain site, even if this is not the case in reality. As a solution, the Virtual Placement was introduced, which uses Rucio to virtually place files in a cache to e.g. serve a storage-less HPC center with grid jobs. The data is then not necessarily available locally, but is provided dynamically by XCache in case of doubt. If the data is accessed again, the local cache can serve the files.

Such a concept could also be useful to enable a cache-aware scheduling to improve the performance of an I/O limited site like HoreKa. The advantage is that the actual status of the cache not necessarily has to be tracked, because if a file is there only virtually, it is simply fetched from remote. But this also means that only after a certain fill state of the caches, a real benefit can be expected. Alternatively, a prefilling of a cache with popular datasets could be considered. This would increase the cache hit rate and relieve the network links in production if a file can be served from a local cache. Additionally, other jobs without locally available files then can benefit from the freed-up bandwidth. But again, this would require adequate jobs and data types that actually require the same files, so most likely *Analysis** jobs. At the moment of writing, HoreKa was only executing *Production** and *Processing** tasks and the concept was therefore not beneficial. On top, significant changes in the CMS data and workflow management would be required to actually deploy such a concept.

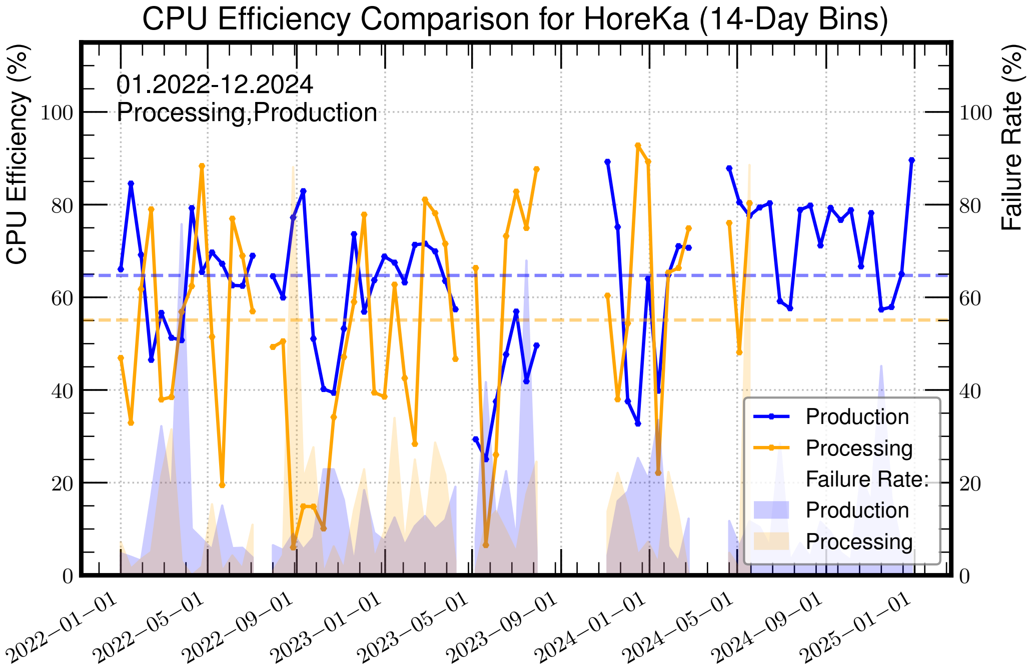


Figure 6.3.: CPU Efficiency and failure rate for *Processing** and *Production** jobs at HoreKa. The dashed lines indicated the total values for the entire period. In many intervals, they are very comparable. But the overall CPU efficiency is significantly better for *Production** jobs. Often, but not always, the CPU efficiency and the failure rate tend to be correlated.

Nevertheless, the idea is definitely interesting for the future of the German HEP computing infrastructure. Especially, when additional NHR centers may contribute to the German WLCG infrastructure. With a network of distributed caches at the HPC centers – or alternatively at the former Tier-2 centers – the foreseen data lake model can be ideally supplemented. Then additional redundancy could be created by virtual replica of datasets, which additionally relieves the official WLCG storage servers at KIT and DESY.
Under current circumstances, however, an approach for optimizing the integration of HoreKa based on caching and cache-aware scheduling cannot be implemented in the short term for the reasons mentioned.

6.2.3. Adaption of the Job Mix

Even if the various jobs in a computing grid are aligned as closely as possible in terms of runtime and resource requirements, there can still be significant variations between different job types. This is shown exemplary in Fig. 6.3 for CMS *Production** and *Processing** jobs that were running at HoreKa. While being often comparable in terms of efficiency and failure rate, the overall result reveals a total deviation of nearly 10 % in total CPU efficiency. From this arises a further opportunity for a rather simple optimization strategy.
An intelligent job selection can help to make the best possible use of available resources. For example, if an I/O limitation of a site with high compute performance, such as HoreKa, is known, it does not make sense to send *Merge** jobs which re-

quire little computing power but transfer large amounts of data. At the same time, if performance-hungry event generation tasks that do not require input data are sent to a site with less performant worker nodes but great bandwidth, this is no optimal usage of the given resources. Therefore, with a proper job selection per site, a common optimum can be found. Of course, this does not mean the solution that corresponds to a local optimum, i.e. rejecting all inefficient jobs so that the own site is in a better position. If everyone would do that, the system as a whole would suffer greatly. Instead, the entire (national) computing infrastructure needs to be considered to achieve an optimal utilization of the available resources.

The ideal concept would therefore be that the jobs are selected according to the capabilities of each site in the entire German WLCG infrastructure. In the end, it could even be useful to view the entirety of the German contributions across all experiments as a joint computing compound. Because the best possible and most sustainable result can only be achieved if the system as a whole is optimized.

However, the same applies here as for the caching approach: It would be a great solution for the future! But at the moment this is only possible to a limited extent. Because currently, no distinction is made between data intensive jobs and compute intensive jobs or workflows in the computing model and workflow management. A selection can only be based on job types. The problem with this is that different campaigns of the same type, however, still can be significantly different. In order to be able to fully incorporate this concept into the scheduling decisions to e.g. find a national optimum of the resource usage, adjustments to the submission infrastructure and CMS workflow management would be required.

For now, excluding the most data intensive jobs and only allow *Production** and *Processing** workflows at the NHR center is the only thing that can be used directly.

6.2.4. XBuffer – XRootD-Based Data Access Bottleneck Mitigation

Another software-based option to mitigate data access bottlenecks is based on special features of XRootD, the default framework for data access in the WLCG. In addition to the caching functionality of XCache, the tool also allows proxying of traffic, as previously described in Fig. 4.11. An XRootD Proxy can be used to get through a firewall, but also data access can be redirected. This process is visualized in Fig. 6.4. Instead of directly sending a request to the GridKa redirector as entry point to the data federation (dashed, brown arrow), and transfer the data directly (brown arrows), a CMS job addresses the Proxy to access a file (dashed, purple arrow). The Proxy then becomes a client itself and sends the request (dashed, green arrow) for data as usual. As the next step, the requested file is located by the redirectors (dashed, black arrows) and the location is reported to the Proxy. In opposite to a XRootD Redirector, which only provides the client with the storage location of a desired file (see Fig. 4.13), the XRootD Proxy actually contacts the remote server, executes the transfer (green arrows), and finally serves the requested data to the initial client over the local network (blue arrow). Additionally, the transferred data can also be saved on the local filesystems to realize an XRootD Caching Proxy.

XRootD's proxy feature therefore can be exploited to optimize the utilization of NHR sites, as it helps to get closer to the original computing concept of HPC centers. Because typically, large HPC workflows require little external data, but rather generate large amounts of data or conduct complex computations. In case that external data

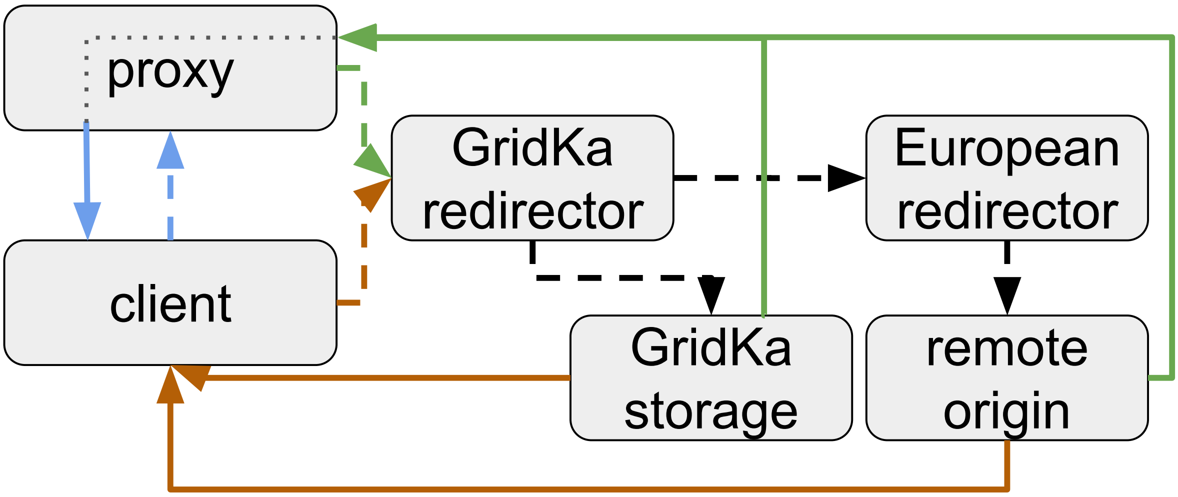


Figure 6.4.: Concept for data-access bottleneck mitigation based on XRootD. Normally (brown path), a client sends its data access request over its own external link to a redirector within the data federation, which locates the file. The client then accesses the file on the remote server itself and transfers the required data. With an XRootD (Caching) Proxy deployed on a faster connected data transfer node of an HPC cluster, this data access process can be optimized. Instead of using the initial path (brown), the client directs its request to the proxy server (blue, dashed). The Proxy then contacts the data federation and executes the transfer (green, dashed path). Finally, the XRootD Proxy serves the files to the initial client over the internal network (green and blue, solid).

is indeed required, it is usually transferred to the cluster in advance. For this, such centers usually provide dedicated data transfer nodes that have a faster external connectivity and can access the local parallel file systems to store the required data. As a result, the – often comparably small – external links per worker node are not needed for the main data access, but e.g. only for the pulling of containers or reporting of intermediate results to an external database. CMS workflows, in turn, fetch the data on the fly during processing, which utilizes the (possibly slower) external links. But thanks to the proxy feature of XRootD, it is possible to use the system analogously for HEP computing tasks.

By deploying such a proxy on the login or data transfer nodes of a cluster, their faster connectivity can be used to mitigate data access bottlenecks. In addition, the previously described concept of caching (Section 6.2.2) can be incorporated, if e.g. local storage or a share on the parallel filesystem of the cluster is available. When the proxy also cache the transferred data, it is typically referred to as an XRootD Caching Proxy. However, as discussed above, whether or not caching is useful depends strongly on the circumstances.

Alternatively, even if no adequate cache hit rate is expected, deploying the XRootD Proxy with the additional caching features can still be beneficial. Because XCache can supplement the optimization strategy by the possibility to prefetch (Section 6.2.1) data during a transfer. This means that the XRootD Caching Proxy simply fetches some additional blocks of files in advance that are not yet required and saves them locally. If they are then requested at a later point in time, it can serve them from the local cache. The functionality therefore corresponds more to a buffer than to a classic cache – hence the self-chosen name: XBuffer.

In summary, it can be said that this is a promising concept for optimizing the utilization of an HPC center as a grid site, as it combines the previously described strategies to achieve the best possible outcome. But when deploying such an XBuffer, care should be taken when deciding how much data is prefetched. Because the loading of (currently) unnecessary data in advance is only beneficial if it does not interfere with normal data transfers. If too much additional data is fetched and the link of the transfer node is saturated, all running jobs suffer from the reduction in available bandwidth. The prefetch factor therefore has to be chosen carefully and is important to be monitored and adapted, if necessary.

An XCache, as well as an XBuffer were deployed and tested at HoreKa, which is described in the next sections.

6.3. Prototype Deployment at HoreKa

The previously introduced XBuffer concept, as well as a full XCache or XRootD Proxy, have the potential to align the HEP computing concept with the HPC mode of operation. With this, it can help to make the most of the integration of HPC centers into the WLCG, especially if they suffer from an I/O limitation, like HoreKa. By utilizing a data transfer or login node of the cluster, which typically come with a better connectivity, the system can increase the available bandwidth for HEP workflows that require just-in-time data access during processing. XBuffer can therefore be

an important building block[6] for the more efficient use of national scientific HPC centers within the NHR initiative as part of the future German HEP computing strategy.
The prerequisites to deploy the different XRootD-based concepts on an HPC cluster are listed in the next part.

6.3.1. Prerequisites and Implementation

An implementation of the concept has the following general prerequisites:

- **Edge Server with Good Connectivity:** Access to a data transfer or login node with a faster connection to the Internet is essential for the concept. The more bandwidth the better. This, of course, includes the right to use an adequate share of the node for the foreseen purpose. Additionally, things like a Firewall need to be considered to reduce the risk of additional limitations.

- **User namespaces and Apptainer:** The developed concept is fully containerized and can easily deployed with Apptainer, without any other strict requirements. Additionally beneficial, however, could be the availability of `cgroups` v2, which enables additional features of Apptainer, such as verbose statistics. Even if rather unlikely to be available, network `namespaces` can simplify the deployment and enable more advanced versions of the setup. The same applies for the possibility of a bare-metal deployment of the software. If this is not given, like at HoreKa, the prototype presented here is relying on Apptainer for the virtualization, but of course, similar containerization technologies can be used as well for the deployment.

- **Storage:** If the caching and/or prefetching feature of XCache is to be used, local storage space is required. This is not necessarily a parallel filesystem, like IBM Spectrum Scale/GPFS, accessible from every node of the cluster, but such a storage can be beneficial for further optimizations (see later in Section 6.4).

- **External Management and Monitoring Server (Optional):** The prototype as deployed at HoreKa has two different operation modes: It can run as a self-sufficient standalone system based on `systemd` or managed from an external server. If the second mode is chosen, such a server is required from which the according node can be accessed. In addition, the prototype setup can provide additional monitoring data. If enabled, external databases (InfluxDB and OpenSearch) are required.

All these conditions are met for HoreKa. To deploy the XRootD (Caching) Proxy, a login node of the cluster is used, as visualized in Fig. 6.5. These nodes are equipped with an 50 Gbit/s external link to the Internet (green), which is significantly faster than the 1 Gbit/s connection of the worker nodes (orange). Hence, from the pure theoretical perspective, the proxy node can serve up to 50 workers. In practice, this will probably be less, since also others may want to use the nodes as well. Internally, all nodes are interconnected via fast IPoIB (blue), just like the IBM Spectrum

[6] Along with the other discussed optimization strategies – if possible to utilize. For CMS, the standalone implementation of the strategies is unfortunately not possible at the time of writing this thesis.

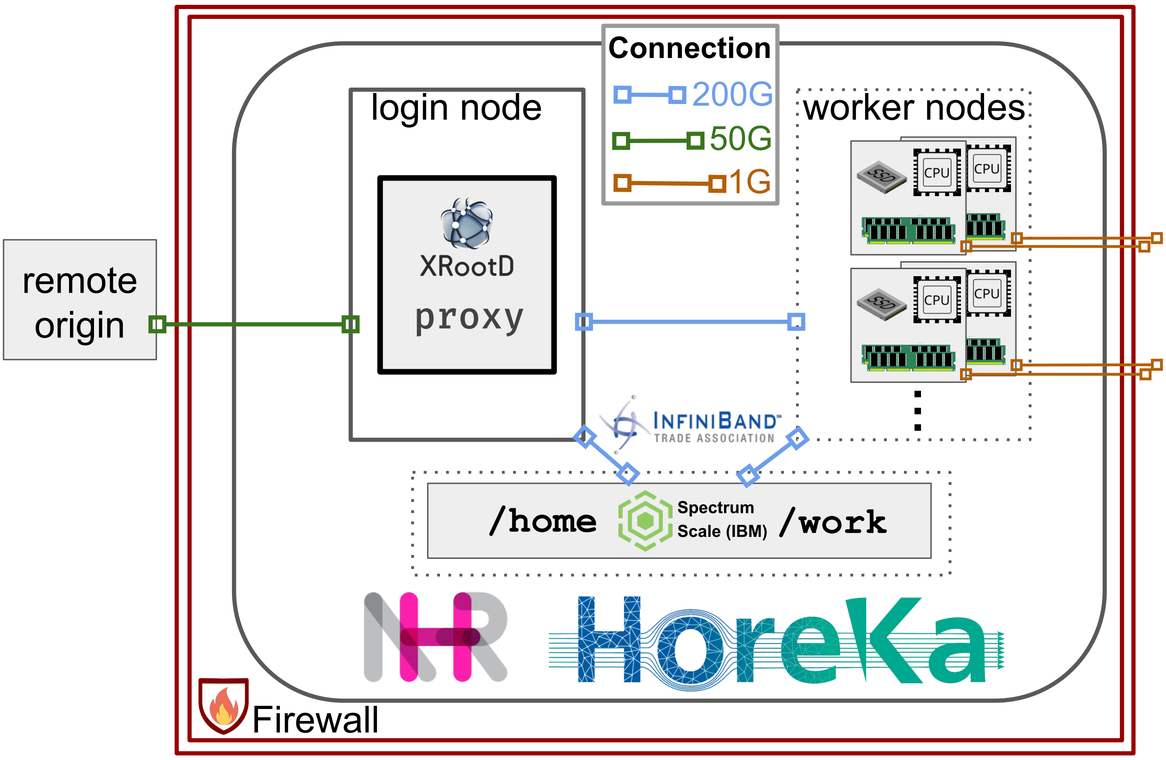


Figure 6.5.: Overview of HoreKa with an XRootD Proxy (as a placeholder) deployed on a login node of the cluster. The setup can be configured to be a full XCache (Caching Proxy), an XBuffer (see Section 6.2.4), or a default XRootD Proxy server. For the internal connections, the cluster provides fast InfiniBand. The incoming and outgoing connectivity is as visualized. To use the caching features, the setup can use a share on the parallel filesystem of the cluster.

Scale/GPFS, which supports RDMA-based communication over InfiniBand and is accessible from every node of the cluster.

For setting up the prototype at HoreKa, a bare-metal installation is not possible, but a fully containerized setup based on Apptainer instances is used. As operational mode, the standalone version was chosen. Here, deployment and automation of the setup are realized as `systemd` (user) services.

The integration of the HPC cluster into the WLCG is realized with COBalD/TARDIS, as described in Section 4.5.3. Allocated nodes are made available as part of an OBS, see Fig. 4.21, and can accept grid jobs via the Cloud CEs located at GridKa. This is beneficial, as optimizations like the job type selection (Section 6.2.3) can be realized through the HTCondor configuration for the cluster. However, the use of the XRootD-based concept in all presented forms in general is not limited to an integration in this way.

When utilizing the XRootD-based concepts for data access bottleneck mitigation, the file access process changes. The resulting, modified file access process with an XRootD Proxy is detailed in Fig. A.59. For the deployment of XCache and XBuffer, the same is presented in Fig. A.60. Additional details on setup and configuration are provided in Section A.5.1.

During the second part of the pilot phase for the integration of HoreKa, all different versions of the concept were tested and optimized. A full evaluation is presented in Section 6.3.3 to find out, if the system can help to optimize the utilization of the HPC center.

But first, another benefit of the concept is presented: additional Monitoring.

6.3.2. Additional Monitoring Capabilities

One of the most limiting constraints when running HEP workflows on HPC is the restricted monitoring. As already mentioned in a previous section, this makes the identification of bottlenecks more complicated and also prevents the standard monitoring workflows for a grid site that usually help to identify problems fast. A sub-site like HoreKa is also not monitored by the monitoring services running at its main site (GridKa). As a consequence, mostly no low-level monitoring information are available in this case, which can lead to difficulties with regard to reliable operation of such centers. Therefore, self-made solutions are required.

Fortunately, the above presented, XRootD-based concept can also help here. Because in addition to the standard payload monitoring data from HTCondor, an XRootD Caching Proxy provides own monitoring information. This includes details on transfers, cached files and how often they are accessed, and other valuable information that supplement the picture and can help to identify problems and limitations or optimize the overall performance. Additionally, if cgroups v2 are enabled on the host, Apptainer instance statistics can also be helpful as they provide information on more basic metrics like storage and memory consumption and even transferred data volume, if network userspaces are available.

Collecting this additional information and combining it with the standard CMS payload monitoring in a meta-monitoring system, such as HappyFace4, can help to get a great overview of the integrated HPC site. It can complement the job monitoring data by more basic information on the data transfers and help to identify problems,

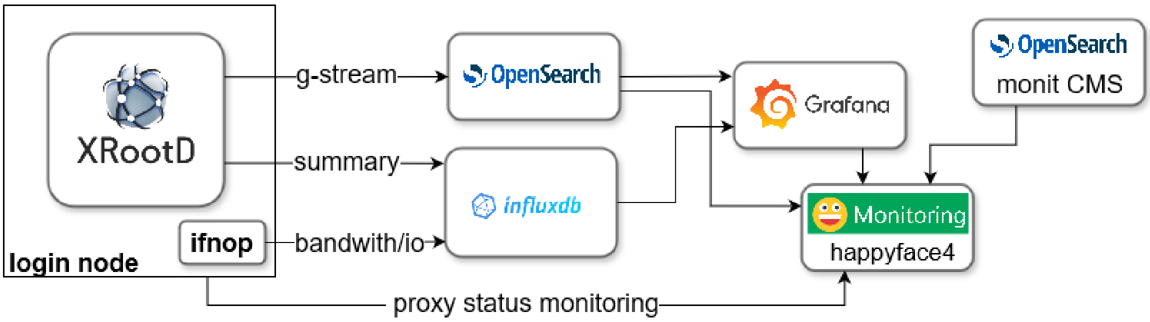


Figure 6.6.: A possible monitoring stack employed for the prototype. Thanks to the XRootD (Caching) Proxy, additional information can be collected. The so-called g-stream monitoring stream of XRootD provides in-depth information on transfers and can be digested and pushed e.g. to OpenSearch. The summary monitoring of the tool and additional I/O monitoring – if available (here realized with *ifnop*, a self-made tool) – can be pushed as a time series to InfluxDB. The collected data can then be presented in Grafana and eventually joined with official job monitoring data in a meta-monitoring tool, such as HappyFace4. Grafana and HappyFace4 both are additionally able to provide alerts in case of problems, greatly simplifying the life of the responsible personnel.

even beyond the own responsibility[7]. In terms of a reliable operation of such a center as a grid site, this is extremely valuable, as it can be reacted fast on problems. An exemplary, monitoring stack including a self-made I/O monitoring of the proxy node is presented in Fig. 6.6.[8]
In conclusion, the presented concepts not only help to optimize the performance, but also increase the monitoring capabilities, which is essential to maintain a reliable grid site. This makes the XBuffer (and all other versions) even interesting if no I/O limitation is present or no direct benefit of the concept on the performance can be observed[9]. Because especially for HPC, monitoring is typically limited and every extension can help.

6.3.3. Evaluation of the Prototype

The described prototype setup was implemented and tested at HoreKa starting end of June 2023. It was refined over the year and a fully functioning Proof of Concept was deployed in the beginning of November of the same year. Since then, different configurations and optimizations were tested. This is indicated by the different background coloring in Fig. 6.7, which shows again the classic quantities, CPU efficiency and failure rate, to get a first impression of the integration. RWTH-HPC is excluded in the following evaluation, because the center was not used over the full period and is therefore not comparable.
During the first period (light blue), an XRootD Caching Proxy was deployed on the

[7] During the development and testing of the concept, I was able to identify and report multiple problems in the grid unrelated to HoreKa, from which the whole CMS Collaboration have profited.

[8] A fully functioning example Docker stack for setting up the monitoring accordingly is provided in Ref. [191].

[9] In such a case, an XRootD Proxy server could be deployed on each node and provide detailed, individual information on the data transfers.

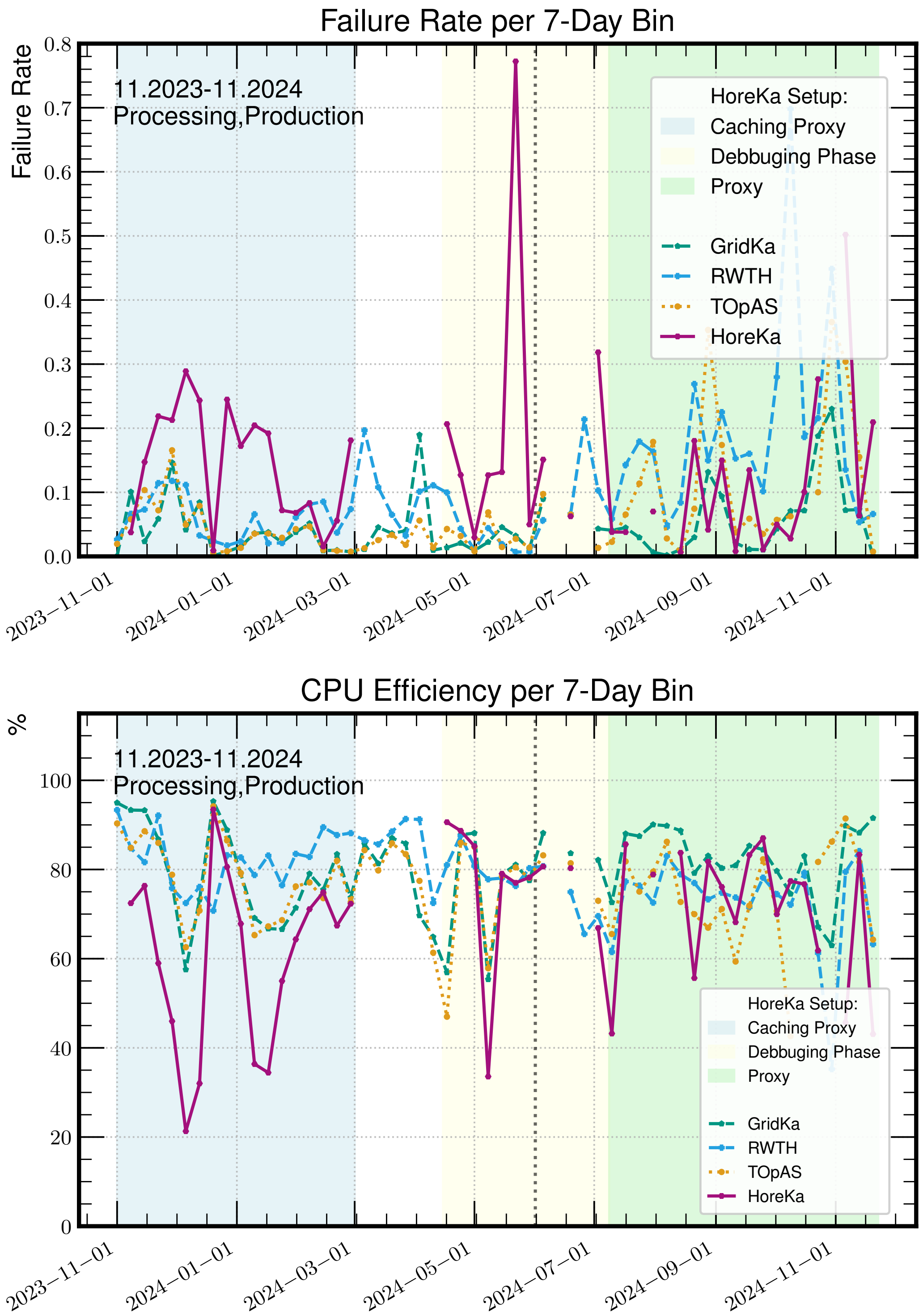


Figure 6.7.: CPU efficiency and failure rate for all sites during the second part of the pilot phase. With the three background colors, different periods with different configurations are indicated. The top figure shows the failure rate for HoreKa and all other sites that are used for the direct comparison. Below, the total CPU efficiency per time bin is shown. During the initial Caching Proxy Phase (blue), the HPC center was in parts still significantly lacking behind the WLCG sites and TOpAS. The Debugging Phase (yellow) was mainly used for reviewing and optimizing the concept and is not entirely representable. Ultimately, during the Proxy Phase (green), where the caching functionality of XRootD was not used, HoreKa could compete well with the other sites. The dotted vertical line in May 2024 indicates the time when *Processing** jobs were excluded from HoreKa.

login node of the cluster which actively used the caching functionality provided by XRootD, see Section 6.2.4. Due to different issues, this initial setup was shut down end of February 2024. The main reason was a bug in the caching procedure in XRootD, unfortunately making it impossible to further cache data on a storage system with a fixed quota, like on HoreKa. As the top part of Fig. 6.7 shows, the failure rate was significantly higher during the first phase and the CPU efficiency (bottom) suffered strongly from caching issues. Even though the I/O limitation could be reduced by the setup, as visible in Fig. 6.8, the overall efficiency increased only by a few percent compared to the vanilla integration during the initial phase.
One of the reasons for this is the handling of failing transfers. Usually, CMSSW is then trying another origin to get the data that is required for the job. When using an XRootD Caching Proxy, this functionality is impaired. In case of a failing transfer, it takes over the retry and the client – the job initially requesting the data from the proxy – is not properly informed by the Caching Proxy about what is happening. The consequence is that XRootD tries to find an alternative source and retries after some time. This waiting time, however, is entirely CPU bad-put, since no data is transferred and nothing is processed. And eventually, if the connection cannot be restored and no alternative source can be found by the proxy, the job fails – but after a longer time due to the timeouts[10] of XRootD.
Additionally, this leads to another, closely related issue influencing the failure rate with the setup deployed. Because using an XRootD Proxy unfortunately disrupts the retry logic of CMSSW. The aforementioned retry mechanism is also triggered when a job is affected by a failing data transfer via the Proxy. But the difference is that the job sees the Proxy as the only origin. The consequence is that on retry, CMSSW kills the job because no additional source could be found that was not tried yet. A possible mitigation for this issue is presented in Section 6.4.1.
This unfortunate interplay of the two software frameworks therefore means that the number of possible retries is de facto reduced to 0 for jobs using the proxy, while default jobs have another attempt to find a working remote server – which of course can prevent job failures in some cases. As a consequence, the error rate increases for HoreKa, which is clearly visible in Fig. 6.7. The effect is particularly pronounced in times when the CMS AAA data federation has issues. This e.g. happend in November 2024. It should be mentioned that neither the site nor one of the two frameworks alone is responsible for the comparably high failure rate with the setup deployed. This is simply a non-ideal constellation which will hopefully improve in the future with further developments.
An additional thing to consider is that the prototype deployed at HoreKa was under constant development during the entire pilot phase. Especially during the period marked in yellow in Fig. 6.7, called 'Debugging Phase'. Within this time span, different solutions for the caching bug as well as different optimizations for the setup in general were tested. When it comes to evaluating how well HoreKa was performing, this has to be kept in mind. The problem is that XRootD does not support a reconfiguration without a restart. As a consequence, running jobs that are using the Caching Proxy on the login node are typically failing when the setup is restarted. Of course, this was avoided when possible. But it regularly happend that jobs needed to be killed due to necessary configuration changes – not only during the

[10] This can in principle be mitigated and optimized with a tweaking of the XRootD configuration for a use-case as decried here, but due the aforementioned bug, this was not possible within this thesis.

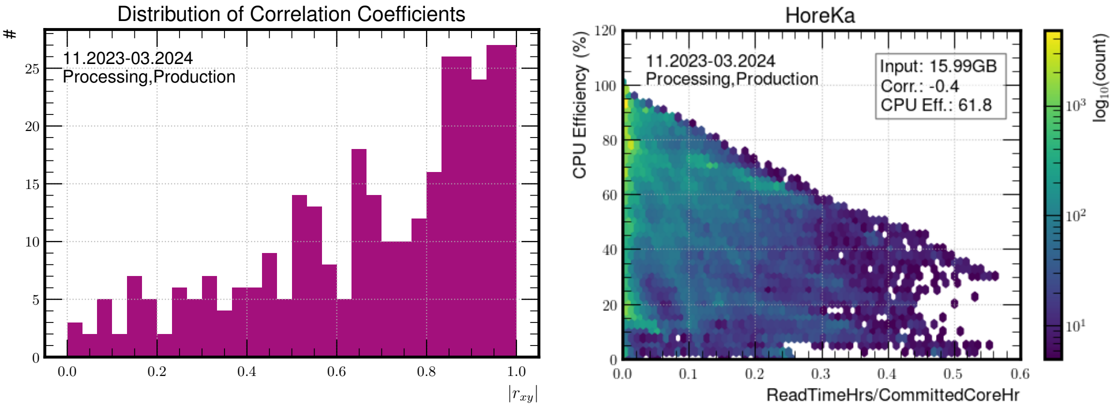


Figure 6.8.: On the left side, the distribution of the correlation factors between the CPU efficiency and the read time fraction is shown for the first optimization phase from November 2023 to March 2024. The I/O limitation is reduced in comparison to the initial phase of the integration (see Fig. 6.2), which is indicated by a smaller correlation factor of $r_{xy} = 0.35$. But as the peak around 1 in the distribution of the correlation factors per workflow indicates, some seem to be still I/O bound. On the right, the correlation of all jobs is shown. In opposite to the initial phase, no jobs surpass 60 %, which is another indication that the I/O limitation is reduced by the prototype setup.

Debugging Phase. This is therefore contributing to the failure rate at HoreKa for the entire pilot integration. However, the actual impact is hard to quantify, but should be over the full period only in the single-digit percent range. For the comparison, it is not considered, because the impact cannot reliably be determined. It just has to be kept in mind that the final deployment is expected to perform overall slightly better on the long run.

In general, this is the downside of the development within production systems. Currently, there is no possibility to get standardized test jobs to a certain site without being in full production mode. With such a mechanism, different configurations could be tested without interfering with the production work done at a site. But currently, there is no other way to optimize the integration. The CMS computing and submission infrastructure does not provide a simple way to get test jobs sent to a certain site[11]. On top, HoreKa is only a sub-site of GridKa. Therefore, it currently cannot be addressed directly[12] without additional tricks. These failures and inefficiencies need to be accepted to a certain degree to develop concepts for a reliable and efficient utilization of such resources in the upcoming period with fixed pledges.

Overall, the result is nevertheless remarkably good for the Debugging Phase – particularly when considering the additional failures. This is clearly visible in Fig. 6.7. Additionally, the direct comparisons to TOpAS and the Tier-2 center in Aachen (RWTH) in terms of CPU efficiency and failure rate are depicted in Fig. 6.9. Especially when reviewing the period in a more granular binning, as depicted in Fig. A.64,

[11] One option could in principle be HammerCloud tests. But they do not scale and would additionally require a development of dedicated benchmark jobs representative for HEP workloads.

[12] A possible way was found in collaboration with the WMAgent team. However, an implementation of the necessary changes was beyond the scope of this work.

it is clearly visible that HoreKa is able to compete well with the other sites in most of the bins. This clearly shows that the prototype setup (without caching) is able to significantly improve the performance of the HPC site.

During the third phase (green) in Fig. 6.7, a further improvement for HoreKa is visible. Caching was disabled over the entire period and only the proxying feature of XRootD is employed. This results in less timeout wait times and an overall smaller failure rate, hence the overall better result. Additionally, the site configuration was optimized[13] in the end of May (dotted, vertical line) from which HoreKa clearly profited in the third phase. This optimization – in combination with the prototype configured as an XRootD Proxy – mitigated the I/O limitation efficiently ($|r_{xy}| = 0.27$, as depicted in Fig. A.65).

Even though the total CPU efficiency is not ideal with only around 70 %, when directly compared to the WLCG sites and TOpAS, the HPC center is close to the set goal of a comparable performance and reliability. Only a small fraction of all time bins is significantly below the target region, underlined by the more granular comparison in Fig. A.64. They often correlate with a high failure rate and can be partly accounted to restarts of the setup, as previously described.

This evaluation has shown that, with the XRootD (Caching) Proxy setup employed in the second and third phases, the NHR center can clearly compete with the dedicated WLCG sites in terms of the classical quantities. As long as the data federation works reliably, the XRootD-based concept for the optimization is therefore very successful and a comparable reliability and efficiency can be achieved.

To get an even more complete picture, the newly introduced quantities are evaluated for the third part of the prototype phase (beginning of July 2024 until end of November 2024) in the following. Based on the corrected CPU efficiency, the loss of compute, and the Processing Efficiency, an even more conclusive statement can be made as to whether HoreKa can be used just as well as the dedicated grid resources for HEP workflows.

Corrected CPU Efficiency: By correcting the CPU efficiency, as described in Section 5.5.1.6, a more realistic picture of the actual job efficiency can be obtained. As depicted in Fig. A.66, the corrected CPU efficiency for HoreKa is comparable or even better, as the HPC center is clearly outperforming the Tier-2 center in Aachen over large parts of the third phase. In the last few weeks is additionally visible that the CMS data federation had problems. All compared sites had high failure rates, but for HoreKa, the impact is the highest, as explained above. This is why the setup was deactivated end of November. Nevertheless, the result is very promising.

Loss of Compute: To realistically consider the actual impact of inefficiencies and job failures on the integration, the Compute Time Loss (Section 5.5.1.7), Inefficiency Loss (Section 5.5.1.8), and Total Loss (Section 5.5.1.9) are evaluated. The results for HoreKa with the XRootD Proxy setup running are listed together with the other sites in Table 6.1. A comparison of the Total Loss over time for all investigated sites is additionally depicted in Fig. 6.10. The compositions of the individual losses for each site are presented in Fig. A.67.

[13] A legacy feature called *lazy download* was disabled, which reduced the fetched amount of data per job.

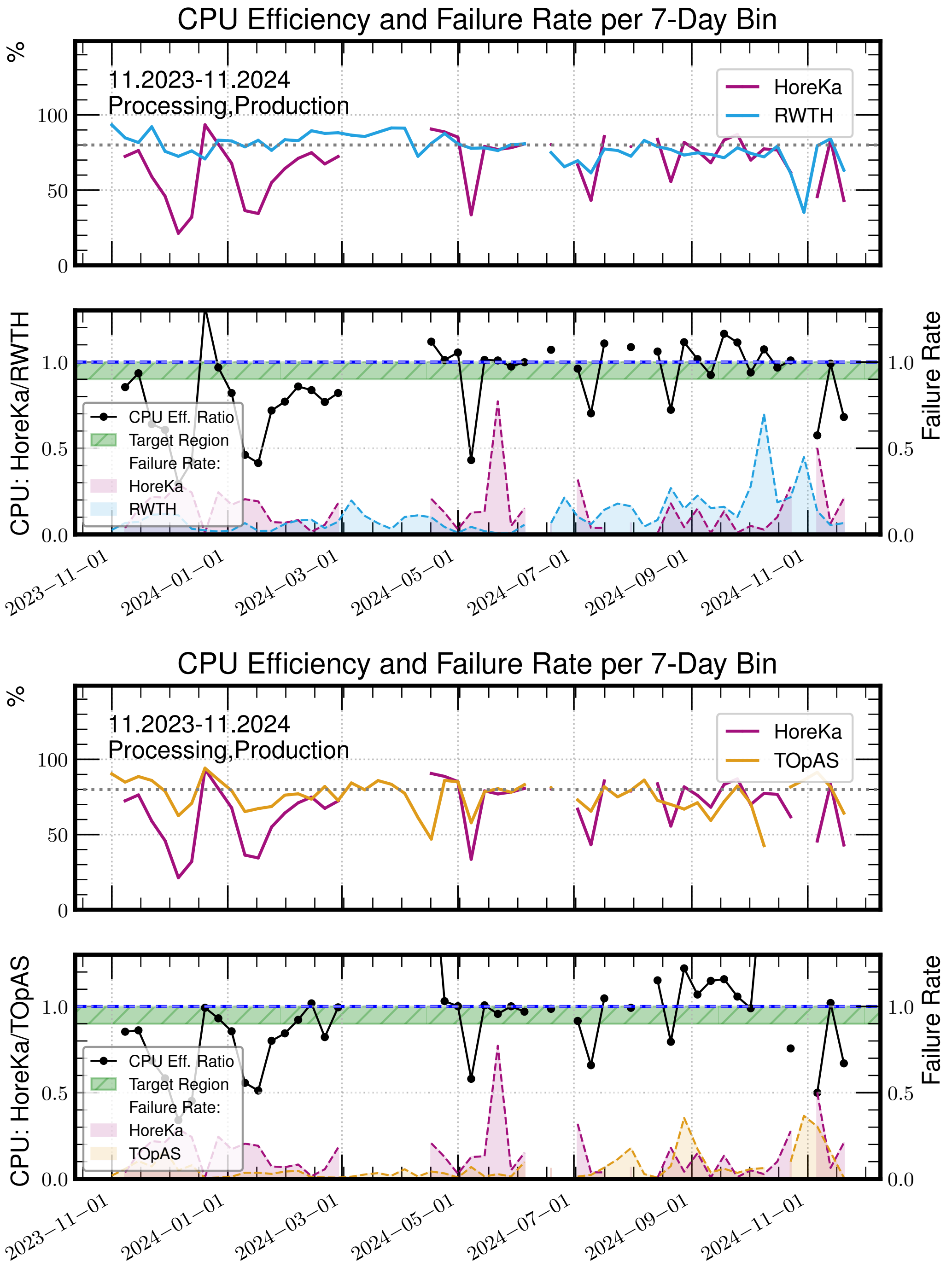


Figure 6.9.: CPU efficiency and failure rate of HoreKa in direct comparison to the Tier-2 in Aachen (top) and the Tier-3 at KIT (bottom). As both figures show, during the second and third phase, the CPU efficiency is mostly in the targeted region – meaning it is comparable to the according site. Partly, HoreKa is even performing better. Also in terms of reliability, the HPC center is mostly showing a similar failure rate as the WLCG sites. The comparison to GridKa is additionally provided in Fig. A.63.

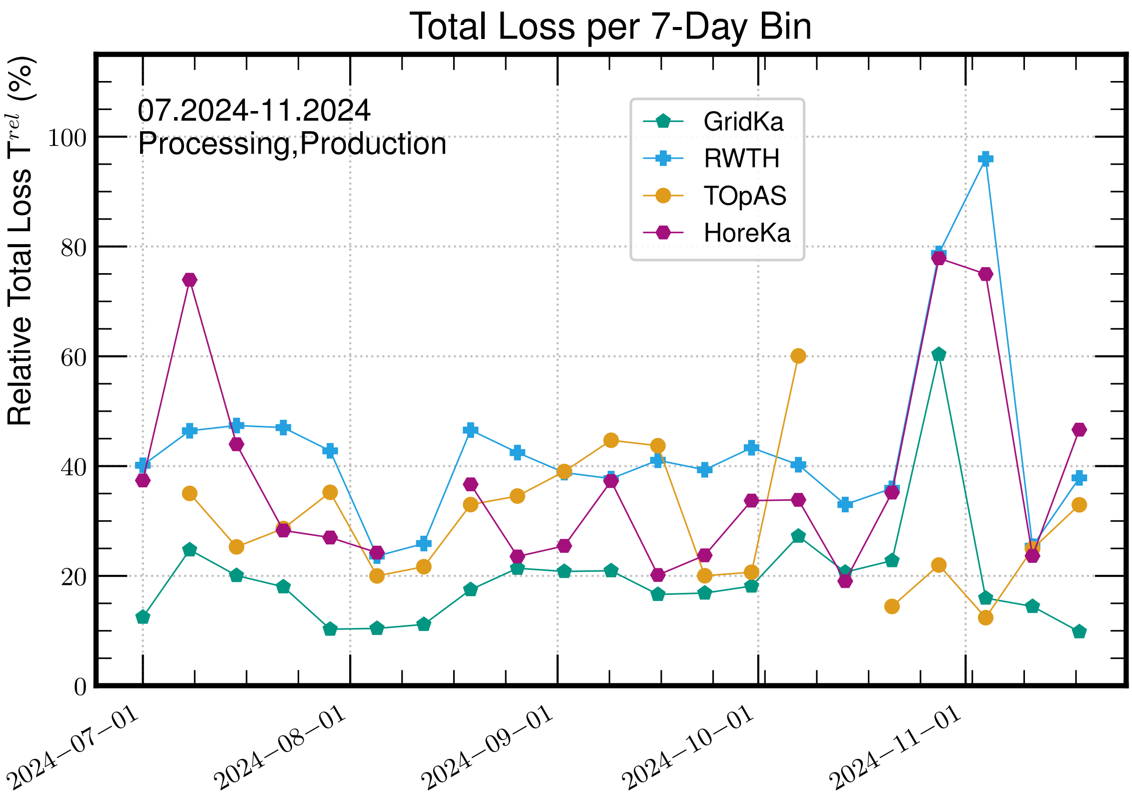


Figure 6.10.: Total Loss for all sites during the Proxy Phase. With the XRootD Proxy setup running and an optimized site configuration, HoreKa is competing very well. Only the Tier-1 center is performing significantly better than the others. The individual compositions of the Losses for all sites are provided in Fig. A.67. Overall, this comparison shows that the NHR site can now be utilized similarly to the traditional grid sites for processing HEP workflows.

The first thing to notice is that the deployed optimizations at HoreKa led to a more sustainable utilization of the HPC site, indicated by a reduction in Total Loss. In comparison to the initial phase of the integration, the fraction of invested core hours without a productive outcome could be reduced by about 10 %[14]. As the distribution of the Total Loss over time in Fig. 6.10 shows, the site was even comparable to TOpAS in big parts of the period. Hence, HoreKa not only improved, but now has an even lower relative Total Loss for the entire period than the Tier-2 center in Aachen (see † in Table 6.1).

This result is a great advancement, as HoreKa was clearly lacking behind the traditional grid sites during the initial phase of the integration. With the prototype setup and the described optimizations in place, the NHR center is now even performing comparable to GridKa in parts of the time period. This is remarkable, as the WLCG Tier-1 sets the bar very high with a relative Total Loss of only 20.2 % over the entire phase.

In summary, this analysis shows that the HPC centers are generally able to compete

[14] Here, it has to be mentioned again that the comparison with the former period is indicative but not entirely conclusive as the running workflows are of course different over time and can, e.g., have an intrinsically different efficiency. The review against the other sites during the same period is therefore more meaningful – but reflects the same result.

with the traditional grid sites. The similar – or even better – utilization is a strong argument for the future German HEP computing strategy, as it shows that contributing the pledges with shares on HPC centers can be just as reliable and even more sustainable than before. But it also has to be mentioned that the result in general is still not ideal. A waste of about 38 % of the invested compute power is lacking behind the own expectations, especially in terms of sustainability and the responsible utilization of the NHR resources. But as shown with this analysis, the reasons are likely beyond the scope of the resource providers, as the results of the sites are comparable. For an even better utilization of the given resources the experiments are in demand to optimize their workflows and infrastructure.

Lastly, this result not only shows the value of the research and development effort for a more sustainable integration of HoreKa, but also proves the usefulness of the newly introduced quantities. By simply looking at the CPU efficiency and the failure rate, it is not possible to make a definitive statement about the performance of the two sites. One could even draw the conclusion that the HPC center is performing worse than the Tier-2 center in Aachen due to the lower CPU efficiency. However, as the Total Loss shows, the picture is reversed when all available information is taken into account, and HoreKa is even performing slightly better.

Processing Efficiency: As the last part of the evaluation, the processing efficiency is reviewed again. For the initial phase, HoreKa was falling behind the WLCG sites and TOpAS, as shown in see Fig. 5.15 and Table 5.6. However, the NHR center was still competitive in terms of processed events per core hour. This was mainly due to the excellent pure performance, which compensated for the inefficiency and failures.

In the third phase with the optimizations and the XRootD Proxy in place, the picture changes. Now, HoreKa has caught up in terms of corrected processing efficiency ($\mathrm{P}_{\mathrm{agg}}^{\mathrm{corr}}(p)$, Eq. (5.9)), as clearly visible in Fig. 6.11. The HPC center is equal or better than the WLCG sites (see ‡ in Table 6.1). When additionally considering the workflow weights, the distribution changes but the overall result is very similar, as observable in the box plot in Fig. A.68.

The according analysis for the corrected but unweighted processing efficiency, $\mathrm{P}_{\mathrm{agg}}^{\mathrm{corr}}$, is depicted in Fig. 6.12. Thanks to a comparable reliability and efficiency, HoreKa is now clearly dominating all sites by far when the pure performance is neglected ($p = 1$). This is supported by the fact that in only 7.2 % of the workflows, another site had a higher number of processed events per invested core hour[15] (see § in Table 6.1). Together with the assumption that a core hour at a large HPC center is more sustainable than at smaller Tier-2 centers, this is an absolute gain!

Summary: To summarize, the XRootD-based approach improves the integration of the HPC center significantly. During the first phase with the XRootD Caching Proxy deployed, HoreKa particularly suffered from the increased failure rate and the waiting and timeout times[16]. The result was therefore not satisfying and further improvements for the prototype were tested. During the beginning of the second phase, the setup was then modified and optimized to only use the proxy functionality

[15] Based on the median, HoreKa processed around 20 % more events in total per core hour.

[16] As discussed, also the caching employed during the first phase can be beneficial, e.g. for *Analysis** jobs – if working properly.

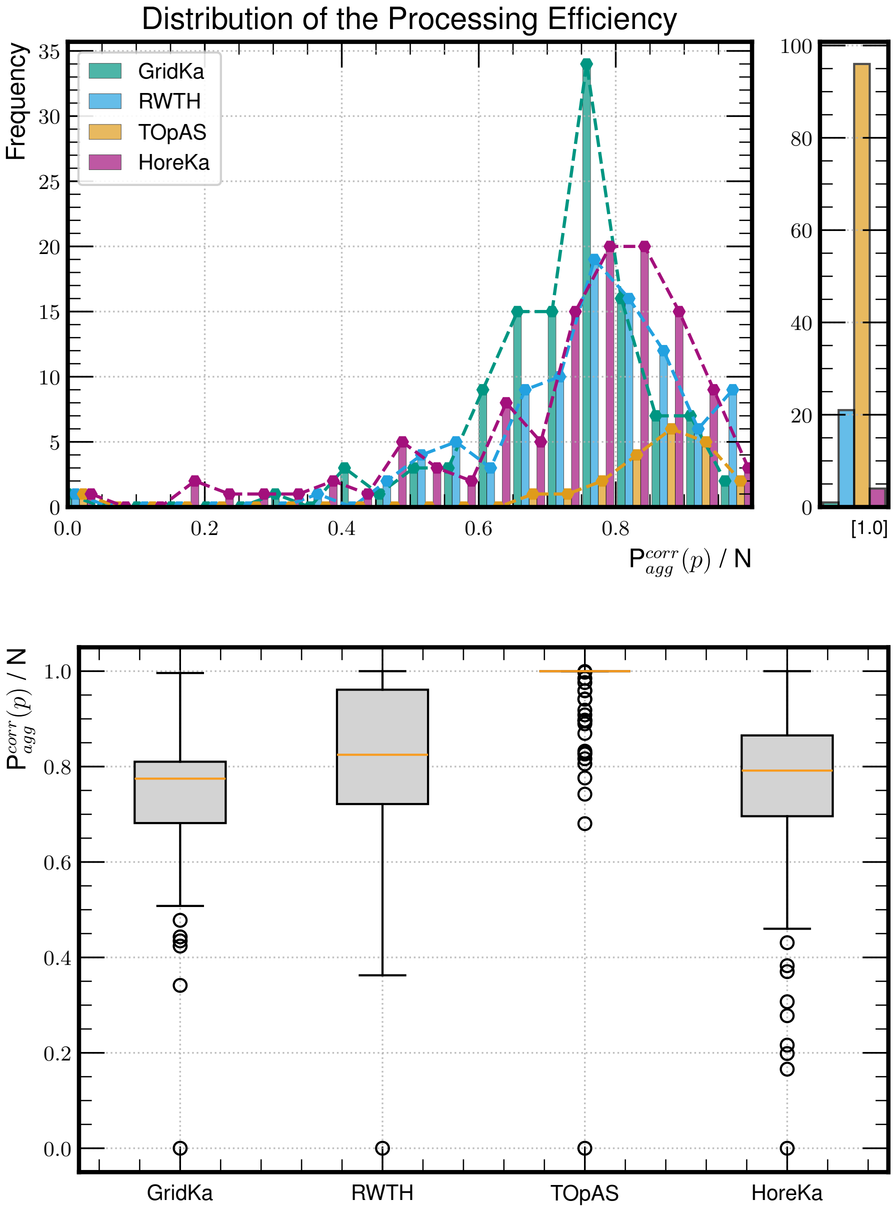


Figure 6.11.: Corrected processing efficiencies per workflow for all reviewed sites. At HoreKa, the XRootD Proxy setup (without caching) was deployed. For a better illustration, the 1.0 bin is again shown separately, like in the evaluation of the initial phase (Fig. 5.15). The according box plot is depicted below. It can be concluded from the distribution that HoreKa works just as well as the other sites with the optimizations employed. In Table 6.1, the average deviation from the optimum (1 - mean) reflects this, too. The box plot shows that – despite some outliers – HoreKa is now performing very comparable to the WLCG sites. In Fig. A.68, different workflow weights are additionally considered. The overall result, however, does not change significantly.

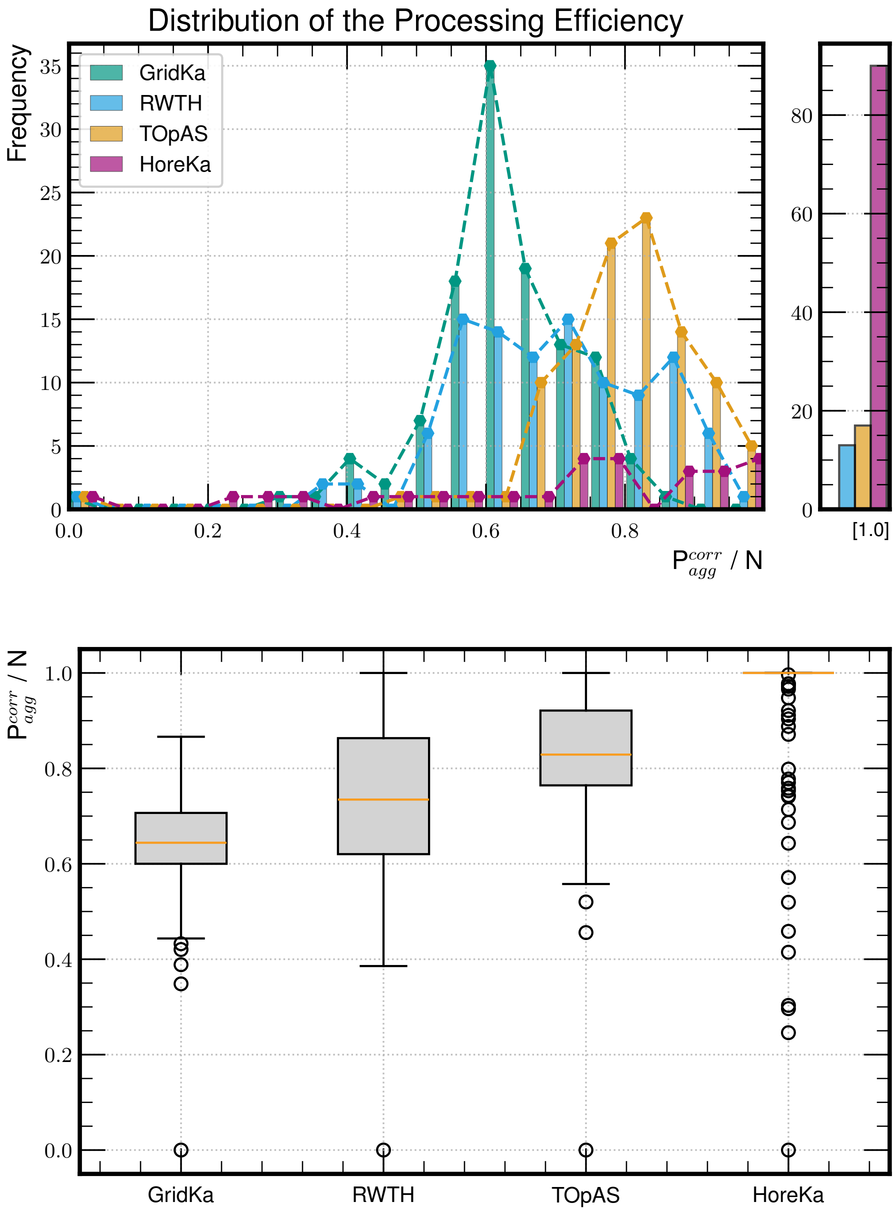


Figure 6.12.: Distribution of the corrected but unweighted processing efficiencies per workflow for all reviewed sites. Here, the performance factor p of the sites is neglected so that the result directly represents the total processed events per core hour. The according box plots are depicted below. With the XRootD Proxy setup and optimizations, HoreKa is extremely dominating the other sites. In over 90 % of the cases, the HPC center was processing the most events per core hour. Considering the workflow weights does not change the overall result.

Table 6.1.: Total statistics for the third period from July 2024 until end of November 2024. In this time, the prototype was running at HoreKa in the Proxy mode (without caching). The statistics for the entire second phase are given in Table A.7. For the performance rating, the p factors for 2024 in Table 5.4 are used. In comparison to the initial phase (Table 5.5), HoreKa was able to catch up, while the other sites have deteriorated.

Measure	GridKa	RWTH	TOpAS	HoreKa
Number of Jobs	1.31M	700K	47K	86K
$\text{CpuEff}_{\text{tot}}$ (Eq. (5.1))	81.6%	73.2%	75.1%	68.5%
Failure Rate (Eq. (5.8))	6.3%	20.4%	9.5%	12.9%
$\text{CpuEff}_{\text{tot}}^{\text{corr}}$ (Eq. (5.10))	79.8%	60.1%	69.9%	61.4%
CommittedCoreHr*	27.1M	8.05M	488K	1.12M
Inefficiency Loss I_{agg} (Eq. (5.17))	4.10M	1.47M	108K	262K
Inefficiency Loss $\text{I}_{\text{agg}}^{\text{rel}}$ (Eq. (5.18))	15.1%	18.3%	22.1%	23.3%
ComputeTimeLoss C_{agg} (Eq. (5.14))	1.36M	1.74M	39K	171K
ComputeTimeLoss $\text{C}_{\text{agg}}^{\text{rel}}$(Eq. (5.13))	5.0%	21.6%	8.0%	15.2%
Total Loss T^{rel} (Eq. (5.21))	20.2%	39.9%†	30.1%	38.6%†
p ratio (relative to TOpAS)	0.98	0.94	1	0.67
Calculated p factor (HS23 Equivalent)	13.78	14.36	13.5	20.15
$\text{P}_{\text{agg}}^{\text{corr}}(p)$ (Eq. (5.9))				
Median	0.77	0.82	1.0	0.79
Average Deviation from Optimum	25.5%	18.7%‡	3.1%	18.7%‡
$\text{P}_{\text{agg}}^{\text{corr}}$ (Eq. (5.9) with $p = 1$)				
Median	0.64	0.73	0.83	1.0
Average Deviation from Optimum	35.7%	26.3%	17.2%	7.2%§

of XRootD, leading to a significant improvement of the performance and a notable reduction of the I/O limitation. Additionally, further optimizations were tested and implemented during this period, such as the exclusion of *Processing** jobs at HoreKa and the deactivation of the *lazy download* feature in the site configuration. The impact of these steps is not easy to disentangle from the other optimizations, but led – together with the prototype setup running at the HPC site – to a comparable performance of the NHR center during the third part of the pilot phase. While still some issues persisted, the overall improvement was remarkable. HoreKa was competitive to the WLCG sites in terms of reliability and efficiency, and especially with the actual event throughput per core hour, the HPC center outperformed the grid sites by far.

From this can be concluded that HPC centers can generally be utilized as well as the traditional grid sites for HEP workflows with appropriate improvements and optimizations, such as the developed XBuffer approach. It also gives an outlook on the promising future of cooperation between the HEP community and the NHR centers within the new HEP computing strategy. Remaining weak points which were revealed during the in-depth analysis of the integration and optimizations are discussed in the following section alongside further suggestions for an even better utilization of the HPC resources.

6.4. Further Improvements

The evaluation has shown that the proposed setup in principle works fine – as long as the data federation is in good condition. If this is not the case, new challenges arise from the fact that the setup with an XRootD Proxy server reduces the available sources to only the proxy. Here, it is again to be mentioned that the prototype setup is not the reason for the problems in the first place. But it is contributing to the issues by reducing the number of possible retries, when data access failures are happening, which is rather unsatisfactory. And since a reliable and efficient operation of such an HPC site is the desired goal, different additional optimizations to the concept can be beneficial in the future.

Some options are discussed below.

6.4.1. Advanced Redirector Setup

During the evaluation of the deployed prototype at HoreKa it was observed that the unfavorable interplay between XRootD and CMSSW is causing a significant increase in job failures in comparison to the dedicated WLCG sites. To prevent the proxy server from being blacklisted as the only available source in the event of a failed data transfer, a more complex concept was developed to circumvent this issue. Since the problem itself cannot be resolved easily as it would require aligned changes in several software frameworks[17], the greatest opportunity for improvement is treating the symptoms.

The most naive approach for this is therefore to add additional proxies to the system. This tricks CMSSW into giving the transfer another try by asking one of the other available proxies. Such a behavior of the system can be realized with another feature

[17] Which would also expectedly take a very long time until such changes are reaching production.

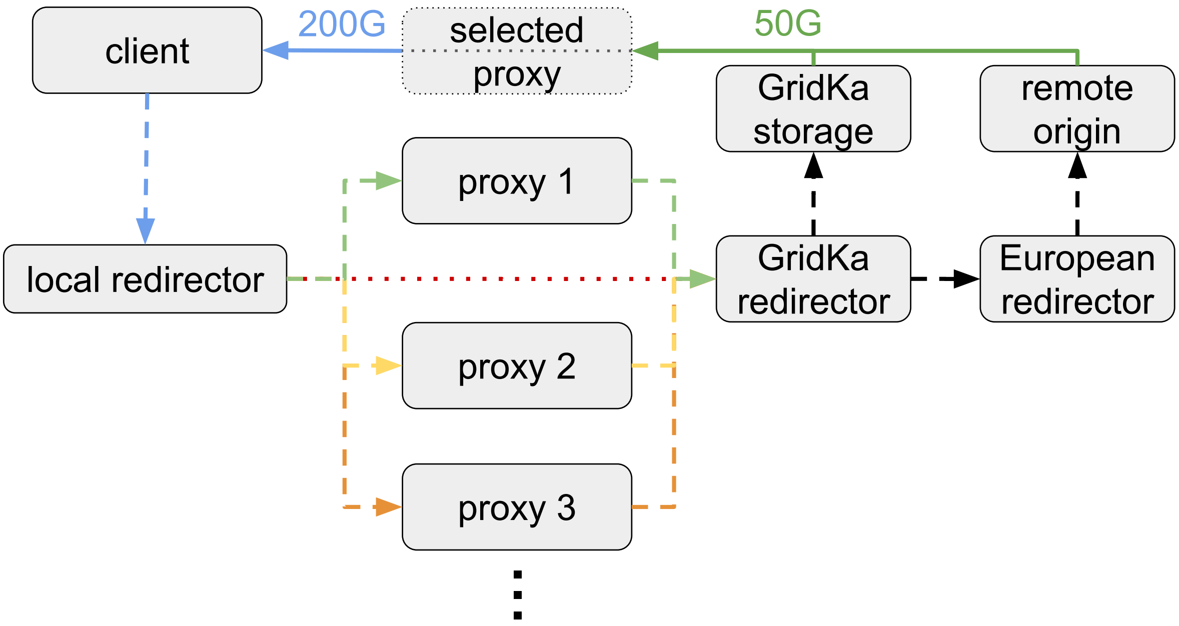


Figure 6.13.: The advanced concept utilizes multiple XRootD (Caching) Proxies that are clustered with a local redirector. In principle, the basic idea and functionality stays the same, as described in Fig. 6.4. But instead of addressing one of the proxies directly for a data transfer request, it is sent to the local XRootD Redirector (dashed, blue). This then selects one of the available proxies (dashed, green/yellow/orange), which contacts the data federation and carries out the transfer as in the basic concept. With this simulation of multiple origins, the problematic blacklisting of a single Proxy by CMSSW is resolved and the retry mechanism can work as intended, while still benefiting from the increased bandwidth of the login node. Additionally, it is imaginable to register the local Redirector also at a regional Redirector – the one at GridKa in this example. In case of a problem with the proxies, it can then redirect immediately to the data federation, bypassing the proxy setup to prevent failures (doted, red line).

offered by XRootD: Multiple proxies can be set up to form a cluster by adding a Proxy Manager, which is in this case essentially functioning as a local[18] XRootD Redirector. In general, the concept's intention is more for the identification and reporting of the origin of a file and additionally for a load balancing mechanism of a federation of clustered data servers. But it can also be exploited for the present use-case.

The initial prototype concept, as presented in Fig. 6.4, can be enhanced by adding a local Redirector and additional XRootD Proxies. The advanced system is visualized in Fig. 6.13. In this constellation, instead of addressing the XRootD Proxy directly, the Redirector is asked for a file. This circumvents the problematic behavior with the disabling of the only available source. The first, second, and third option are indicated in the figure in green, yellow, and orange, respectively. It then finds the best fitting XRootD (Caching) Proxy – which is a little bit daft in this setup as everything runs on the same physical node – and forwards the request. In a multi-node setup[19],

[18] Here, local indicates that the Redirector is only accessible by local jobs from within the cluster.

[19] When such a setup is required depends on the on-premise setup of each cluster. Of course, if

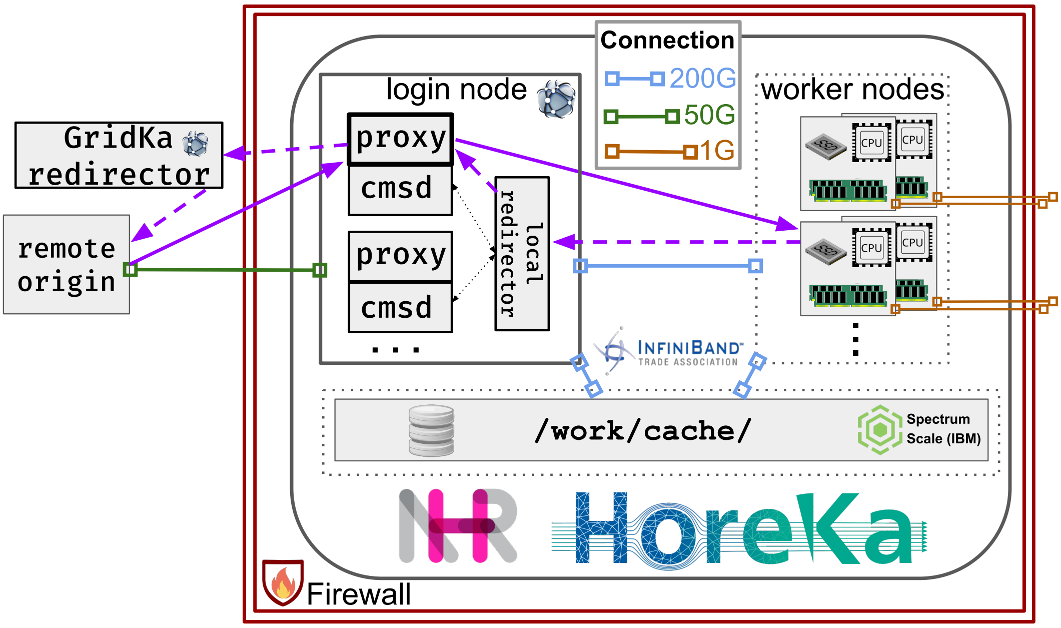


Figure 6.14.: Technical implementation concept for the advanced prototype. With multiple XRootD (Caching) Proxies (and their cluster management service daemons – cmsd), the aforementioned issues with failing transfers can be mitigated. The number of Proxies should be chosen according to the load, but at least three are recommended. Especially, if multiple nodes for the setup are available, this extension considerably increases the possibilities and potential of the concept.

however, where the different XRootD servers run on different physical hosts, this is an additional benefit. Then, the setup even automatically implements a proper load balancing between the available proxies without any further effort and an increased total bandwidth can be achieved. The rest of the data access process is the same as for the simple approach. A possible technical implementation of this advanced setup for HoreKa is shown in Fig. 6.14.

Additionally, as backup if problems with the setup occur, one could also think of extending the concept by integrating an external XRootD Redirector as well, e.g. the GridKa Redirector in case of HoreKa. This is indicated by the red-dotted line in Fig. 6.13. Such an extension would allow to bypass the Proxies, which e.g. could be useful if there are ongoing problems within the global data federation to avoid a further increase of the failure rate due to timeout times of the transfers via the Proxy. But this would require to add a small delay to the external Redirector so that the local Redirector prefers the Proxies to avoid bypassing the setup when there are no problems. Additionally, it adds further complexity to the system, particularly with parts that one typically does not have influence on (the external Redirector). Therefore, it has to be considered carefully, if this is desired as a fallback solution. Eventually, the optimal constellation is also depending on the policies and permissions. In case of HoreKa, for example, such an integration is not possible at the time of writing this thesis as it would require to open the firewall for registering with the external XRootD Redirector.

In summary, with this more advanced setup, the problematic interplay observed

several 1000 jobs run at the same time, a single XRootD (Caching) Proxy can become a bottleneck again. This has to be monitored carefully and extended as needed.

between the frameworks can be mitigated. Additionally, with a multi-node setup, the potential can further be increased to gain even more bandwidth and redundancy. However, the actual implementation depends on various conditions and must be adapted to each individual case. At HoreKa, for example, a deployment is currently not possible, because only one node is available and the security policies prevent a deployment of the advanced Redirector setup[20]. A simulated, working example for the advanced prototype setup is provided as a Docker stack in Ref. [191].

6.4.2. Dedicated Transfer Nodes

Another alternative for optimizing the presented prototype would be to deploy the setup on external, dedicated transfer nodes. As part of the future German HEP computing strategy, this could be servers at the collocated university Tier-2 sites, or in case of HoreKa, at GridKa. This is a valid option for NHR centers, or HPC resources in general, that do not allow the deployment of services on edge nodes of the cluster, but would implement a connection of such a node to the internal cluster network[21]. In case of HoreKa and GridKa, the two clusters are located in two buildings next to each other that are already interconnected. Therefore, deploying the setup on a server at the Tier-1 center and connecting it to the InfiniBand network of the HPC cluster would be a great option for future improvements. This would have essentially two advantages. At first, the strong restrictions and limitations to deploy such a complex system as a user on a cluster with stacked containers would be omitted, as on the external, self-managed node, everything could be set up bare-metal as `root`. This would simplify deployment and reduce overheads, as well as enabling more detailed, low-level monitoring. Additionally, necessary hardware upgrades could be decided independently of the NHR center. And as a second benefit, a dedicated transfer node at an official WLCG site could provide a direct connection to LHCONE. As a result, data access would be optimized further, as such an adaption would allow transfers via the WLCG networks instead of the public Internet, avoiding losses caused by multiple firewalls.

But of course, this would be a great concession on the side of the resource providers. For HoreKa, such an upgrade is being considered for the successor to the current cluster.

6.4.3. Performance Optimizations

In addition to adapting the physical resources for the prototype setup depending on the load, it is also possible to carry out a fine-tuning of the setup. XRootD provides a wide range of possibilities and configurations that can have a significant influence on the performance and reliability. It was, e.g., found that the reduction of timeout times can make a significant difference in terms of CPU efficiency, as it reduces the CPU bad-put. Another example is the adjustment of the prefetching according to the number of expected parallel jobs and the available bandwidth of the used edge node. When a high number of jobs is prefetching 100 blocks in advance, the total amount of transferred data increases significantly. At a certain point, this can, of

[20] Network namespaces are disabled, leading to problems when registering the proxies at the local Redirector, as they all share the same hostname

[21] Of course, this heavily depends on the individual security policies

course, lead to a saturation of the external link, affecting all running jobs. Therefore, it is important to choose a configuration fitting to the circumstances.

One big problem here is that, as already discussed in Section 6.3.3, a fine-tuning while running in full production mode is complicated. On the one hand, the current version of XRootD (v5.7.3) does not allow reconfiguration during operation, but always requires a restart. This can cause job failures or else make the optimization process very cumbersome and inefficient. On the other, all running jobs can be significantly different. This makes a controlled adjustment and optimization very difficult.

An alternative option could again be to utilize HammerCloud tests to run dedicated benchmarks with the same workload for different configurations. However, the problem with this is that some limitations may only occur due to high loads and the inevitable interference between different jobs on the same node or cluster. This can never be correctly detected and reflected by individual, independent test jobs under laboratory conditions. To simplify the tuning process, the experiments would have to provide dedicated testing, debugging, and benchmarking workflows. For now, adaptions of the configuration should be based on careful observations of the setup and be reviewed regularly, because as shown with this work, optimizations matching the circumstances can make a real difference.

6.5. Conclusion

For the upcoming HEP computing strategy in Germany to be successful, it is very important that the scientific HPC centers within the NHR Alliance, such as HoreKa at KIT, can be used reliably and efficient. The transition to these resources represents a fundamental change in the German computing infrastructure and brings several challenges and limitations with it, as described in Section 5.3. Of these, the different computing models of dedicated WLCG resources and HPC centers as well as the changing requirements in comparison to the opportunistic integration are particularly noteworthy. While expectations on the utilization of the NHR centers were comparably low during the first part of the pilot phase (January 2022 until June 2023), as discussed in Section 5.4, this will change drastically with the official, pledged integration to provide a part of the German WLCG contributions. Especially, a reliable and sustainable provisioning of compute power is demanded. The university Tier-2 centers, which are to be replaced by the NHR centers in the long run, can be seen as the target that has to be achieved.

To assess whether this goal is in reach, meaningful and representative measures for the evaluation of a grid site were introduced in Section 5.5.1. Based on these measures, the two opportunistically integrated NHR centers, HoreKa and CLAIX, located at KIT and RWTH Aachen, respectively, were analyzed in Section 5.6.3. This comprehensive evaluation has shown that although the HPC centers lag behind the classic grid sites[22] in some aspects, they also offer an enormous potential. Particularly HoreKa provided a comparable event throughput per invested core hour although having a significantly less CPU efficiency, as the processing efficiency evaluation in Section 5.6.3 showed.

In order to better exploit the high performance potential of the resource, the causes

[22] GridKa, the university Tier-2 in Aachen, and TOpAS, the Tier-3 at KIT.

were analyzed in detail in Section 6.1 and mainly an I/O limitation was identified. Based on the findings, different mitigation strategies were discussed in Section 6.2 and combined to introduce an XRootD-based approach for the mitigation of data access bottlenecks when utilizing HPC clusters. Three different variations were presented in Section 6.2.4: XCache, XBuffer, and a simple XRootD Proxy. Verbose testing of the various concepts at HoreKa during the second part of the pilot phase of its integration has revealed additional weaknesses and challenges, for which further optimizations are proposed in Section 6.4.

But what is more important is that the analysis in Section 6.3.3 has shown that the NHR center is able to compete with the classic WLCG sites when the prototype is used and different optimizations are in place. Because this is the basis for the success of the future strategy. The results compared to RWTH and TOpAS are definitely promising for the possibility to achieve the self-set goals of a sustainable and responsible utilization of the NHR resources. Furthermore, the gradual increase in pledges served by the HPC centers offers the opportunity to make even better use of the potential of these resources through further research and development until the start of the HL-LHC era.

In conclusion, we are confident that HPC centers like HoreKa are capable of making a reliable and important contribution to the WLCG in the future, while being more sustainable than many small, individual centers.

7. Summary and Outlook

> The great thing in this world is not so much where we are, but in what direction we are moving.
>
> (Oliver W. Holmes)

This thesis has addressed two closely connected challenges of modern High Energy Physics: achieving unprecedented precision in jet measurements at the CMS experiment through sophisticated, compute-intensive Jet Energy Corrections, and developing strategies for the sustainable utilization of HPC centers in the future to provide sufficient compute resources for the research in the high-precision era of the (HL-)LHC.

The thesis presents the first full derivation of the absolute residual Jet Energy Corrections for Run 2 Ultra Legacy data recorded with the CMS detector. The tools and methods were carefully adapted in close collaboration with the CMS-JERC group to meet the requirements and recommendations for the latest reprocessing of CMS Run 2 data. With an in-depth analysis of the result it was shown that the derived corrections minimize discrepancies between data and simulation. The resulting accuracy is essential for high-precision QCD studies and for detecting even the smallest deviations from the SM that could indicate new physics.

Furthermore, contributions were made toward the future of Jet Energy Corrections at CMS through a synchronization process with the successor group responsible for Run 3. By aligning the legacy MiniAOD-based frameworks with the new Run 3 toolset, a complete knowledge transfer was achieved. This synchronization process not only validated and refined the Run 2 calibrations but also established a robust foundation for Run 3, enabling the CMS Collaboration to leverage the proven capabilities of the established tools alongside the performance benefits of the modern frameworks and data types. This advancement is an important step in addressing the ever-increasing demand for computational resources.

The second part of this thesis focuses on this challenge. With the WLCG as the backbone of modern particle physics research, the HEP community is well positioned. However, to meet the increased requirements and expected demand for computing resources in the future, a new HEP computing strategy was decided in Germany. This strategy, with a special focus on the utilization of HPC centers within the NHR Alliance for the pledged contribution to the WLCG, was discussed together with the resulting challenges and limitations this absolute novelty comes with. Furthermore, new measures and methods for a meaningful evaluation of grid sites were introduced that provide a more detailed overview of how well the investigated resources are performing. By utilizing these measures, the opportunistic integration of HoreKa was analyzed and bottlenecks and limitations – mainly an I/O limitation in combination with a lack of monitoring capabilities – were identified. Based on these findings, different optimization approaches, such as prefetching, caching and cache-aware job scheduling, and an intelligent job selection were discussed. While most of them are not feasible to resolve the observed issues as stand-alone solutions, parts of the concepts were incorporated into a new, XRootD-based approach for the mitigation of data access bottlenecks on such HPC centers – called XBuffer. This concept – in case of HoreKa – utilizes login nodes of the cluster with a faster external connectivity to enhance the data transfer capabilities and can optionally employ caching and prefetching mechanisms. With the prototype setup in place, together with an optimized site configuration, it was ultimately shown that the HoreKa HPC cluster is able to compete with the traditional WLCG sites. For further improvements, additional weak points of the setup – mainly in terms of reliability – were identified and possible solutions discussed to enable the full exploitation of the potential of the NHR resources in the future.

The results of this thesis are therefore highly encouraging. They not only support the viability of the future HEP computing strategy but also lay a strong foundation for further research and development to achieve an even more sustainable utilization of HPC resource in the future. The promising performance of the XRootD-based Proxy prototype setup has shown that a deeper exploring of the concept is worthwhile, as a full exploitation of the enormous performance potential of the NHR centers will be very beneficial for the HEP community. Future work in this direction could focus on fine-tuning of parameters and optimizations, the evaluation of advanced setups for a more robust utilization, and caching strategies supporting the aversed data lake model. In combinations with advancements in scheduling, one could even consider optimizations on a workflow-basis for, e.g., ensuring that resource-intensive tasks are allocated to the most capable sites.

Another potential benefit of the new strategy could be an on-demand provisioning of GPUs and accelerators in the future through the integration of such HPC clusters. This flexibility, coupled with reduced maintenance and ecological advantages, paves the way for more sustainable computing practices. Going even two steps further, this could especially be beneficial when the experiment boundaries may be overcome in computing on a national level. A joint utilization of such resources will strengthen experiment and interdisciplinary collaborations, offering even more opportunities for an optimization of the national HEP computing infrastructure. And joint, continued research and development efforts can extract even more potential from existing resources, reducing the need for additional hardware investments while enhancing overall efficiency and sustainability.

Furthermore, a refinement of the newly introduced evaluation measures has the potential to lead to a 'Standard Model of HEP Grid Site Evaluation' – inspired by particle physics – enabling a more realistic and conclusive assessment of grid sites, with enhanced comparability and a special focus on sustainability. Such a framework would provide a well-founded basis for future strategic decisions and help ensure a more optimized and sustainable HEP computing.

In conclusion, by combining contributions to precision physics measurements with cutting-edge, sustainable computing developments, this work not only contributes to the understanding of fundamental interactions in High-Energy Physics but also confirms the future HPC strategy in Germany. Based on the findings, it was shown that HPC centers, such as HoreKa, are able to play an important role in HEP computing, providing a flexible, reliable, and sustainable foundation for future High-Energy Physics research.

Bibliography

[1] S. Chatrchyan et al. 'Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC'. In: *Phys. Lett. B* 716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235.

[2] G. Aad et al. 'Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC'. In: *Phys. Lett. B* 716 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020. arXiv: 1207.7214.

[3] V. Khachatryan et al. 'Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV'. In: *JINST* 12 (2017), P02014. DOI: 10.1088/1748-0221/12/02/P02014. arXiv: 1607.03663. URL: https://cds.cern.ch/record/2198719.

[4] M. E. Peskin and D. V. Schroeder. *An introduction to quantum field theory*. Boulder, Colorado: Westview, 1995. URL: https://cds.cern.ch/record/257493.

[5] S. Weinberg. *The Quantum Theory of Fields: Volume 1, Foundations*. Cambridge University Press, 2005. ISBN: 9781139643245.

[6] R. P. Feynman. 'Space-Time Approach to Quantum Electrodynamics'. In: *Phys. Rev.* 76 (6 Sept. 1949), pp. 769–789. DOI: 10.1103/PhysRev.76.769. URL: https://link.aps.org/doi/10.1103/PhysRev.76.769.

[7] D. J. Gross and F. Wilczek. 'Ultraviolet Behavior of Nonabelian Gauge Theories'. In: *Phys. Rev. Lett.* 30 (1973). Ed. by J. C. Taylor, pp. 1343–1346. DOI: 10.1103/PhysRevLett.30.1343.

[8] F. Wilczek. 'Nobel Lecture: Asymptotic freedom: From paradox to paradigm'. In: *Reviews of Modern Physics* 77.3 (Sept. 2005), pp. 857–870. ISSN: 1539-0756. DOI: 10.1103/revmodphys.77.857. URL: http://dx.doi.org/10.1103/RevModPhys.77.857.

[9] R. L. Workman et al. 'Review of Particle Physics'. In: *PTEP* 2022 (2022), p. 083C01. DOI: 10.1093/ptep/ptac097.

[10] C. Patrignani. 'Review of Particle Physics'. In: *Chinese Physics C* 40.10 (Oct. 2016), p. 100001. DOI: 10.1088/1674-1137/40/10/100001. URL: https://dx.doi.org/10.1088/1674-1137/40/10/100001.

[11] R. L. Jaffe. *Deep inelastic scattering with application to nuclear targets*. Tech. rep. Cambridge, MA: MIT. Cent. Theor. Phys., 1985. URL: https://cds.cern.ch/record/161757.

[12] Z Lalak et al. 'CTEQ parton distributions and flavor dependence of sea quarks'. In: *Phys. Lett. B* 304 (1993), pp. 159–166. DOI: 10.1016/0370-2693(93)91130-F. URL: https://cds.cern.ch/record/248068.

[13] B. Andersson et al. 'Parton Fragmentation and String Dynamics'. In: *Phys. Rept.* 97 (1983), pp. 31–145. DOI: 10.1016/0370-1573(83)90080-7.

[14] K. G. Wilson. 'Confinement of Quarks'. In: *Phys. Rev. D* 10 (1974). Ed. by J. C. Taylor, pp. 2445–2459. DOI: 10.1103/PhysRevD.10.2445.

[15] R.P. Feynman. *QED: The Strange Theory of Light and Matter*. Alix G. Mautner memorial lectures. Princeton University Press, 1985. ISBN: 9780691083889.

[16] R.P. Feynman and A. Zee. *QED: The Strange Theory of Light and Matter*. Alix G. Mautner memorial lectures. Princeton University Press, 2006. ISBN: 9780691125756.

[17] B.R. Webber. 'Fragmentation and Hadronization'. In: *Int. J. Mod. Phys. AS* 15S1 (2000), pp. 577–606. DOI: 10.1142/S0217751X00005334. URL: https://cds.cern.ch/record/419784.

[18] The CMS collaboration. 'Determination of jet energy calibration and transverse momentum resolution in CMS'. In: *Journal of Instrumentation* 6.11 (Nov. 2011), P11002–P11002. ISSN: 1748-0221. DOI: 10.1088/1748-0221/6/11/p11002. URL: http://dx.doi.org/10.1088/1748-0221/6/11/P11002.

[19] The CMS Collaboration. *Jet Energy Resolution in CMS at sqrt(s)=7 TeV*. Tech. rep. Geneva: CERN, 2011. URL: https://cds.cern.ch/record/1339945.

[20] The CMS Collaboration. *CMS: The computing project. Technical design report*. Technical design report. CMS. CERN, July 2005. URL: https://cds.cern.ch/record/838359.

[21] S. Chatrchyan et al. 'The CMS experiment at the CERN LHC'. In: *JINST* 3 (2008). Also published by CERN Geneva in 2010, S08004. DOI: 10.1088/1748-0221/3/08/S08004. URL: https://cds.cern.ch/record/1129810.

[22] E. Lopienska/CERN. *The CERN accelerator complex*. Accessed: 2025-02-02. webpage. URL: https://cds.cern.ch/images/CERN-GRAPHICS-2022-001-1.

[23] T. S. Pettersson and P. Lefèvre. *The Large Hadron Collider: conceptual design*. Tech. rep. 1995. DOI: 10.17181/CERN-AC-95-05-LHC. URL: https://cds.cern.ch/record/291782.

[24] R. Bruce et al. 'LHC Run 2: Results and challenges'. In: (2016), MOAM5P50. DOI: 10.18429/JACoW-HB2016-MOAM5P50. URL: http://cds.cern.ch/record/2201447.

[25] O. Aberle et al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2020. DOI: 10.23731/CYRM-2020-0010. URL: https://cds.cern.ch/record/2749422.

[26] A.M. Sirunyan et al. 'Particle-flow reconstruction and global event description with the CMS detector'. In: *Journal of Instrumentation* 12.10 (Oct. 2017), P10003–P10003. ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/10/p10003. URL: http://dx.doi.org/10.1088/1748-0221/12/10/P10003.

[27] D. Barney. 'CMS Slice'. In: (2015). URL: http://cds.cern.ch/record/2628641.

[28] CMS/CERN. *The CMS Detector*. Accessed: 2025-02-11. webpage. URL: https://cms.cern/detector.

[29] I. Neutelings. *CMS coordinate system*. Accessed: 2025-02-03. webpage. URL: https://tikz.net/axis3d_cms/.

[30] G. L. Bayatian et al. *CMS Physics: Technical Design Report Volume 1: Detector Performance and Software*. Technical design report. CMS. Geneva: CERN, 2006. URL: https://cds.cern.ch/record/922757.

[31] T. Sakuma and T. McCauley. 'Detector and Event Visualization with SketchUp at the CMS Experiment'. In: *J. Phys. Conf. Ser.* 513 (2014). Ed. by D. L. Groep and D. Bonacorsi, p. 022032. DOI: 10.1088/1742-6596/513/2/022032. arXiv: 1311.4942 [physics.ins-det]. URL: http://cds.cern.ch/record/2677903.

[32] W. Adam et al. 'The CMS Phase-1 Pixel Detector Upgrade'. In: *JINST* 16.02 (2021), P02027. DOI: 10.1088/1748-0221/16/02/P02027. arXiv: 2012.14304. URL: https://cds.cern.ch/record/2748381.

[33] V. Karimäki et al. *The CMS tracker system project: Technical Design Report*. Technical design report. CMS. Geneva: CERN, 1997. URL: https://cds.cern.ch/record/368412.

[34] The CMS Collaboration. *The CMS electromagnetic calorimeter project: Technical Design Report*. Technical design report. CMS. Geneva: CERN, 1997. URL: https://cds.cern.ch/record/349375.

[35] The CMS Collaboration. *The CMS hadron calorimeter project: Technical Design Report*. Technical design report. CMS. Geneva: CERN, 1997. URL: https://cds.cern.ch/record/357153.

[36] D. E. Groom, N. V. Mokhov and S. I. Striganov. 'Muon stopping power and range tables 10-MeV to 100-TeV'. In: *Atom. Data Nucl. Data Tabl.* 78 (2001), pp. 183–356. DOI: 10.1006/adnd.2001.0861.

[37] V. Khachatryan et al. 'The CMS trigger system'. In: *Journal of Instrumentation* 12.01 (Jan. 2017), P01020–P01020. ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/01/p01020. URL: http://dx.doi.org/10.1088/1748-0221/12/01/P01020.

[38] G. Parida. 'Run-3 Commissioning of CMS Online HLT reconstruction using GPUs'. In: *EPJ Web of Conf.* 295 (2024), p. 11020. DOI: 10.1051/epjconf/202429511020. URL: https://doi.org/10.1051/epjconf/202429511020.

[39] The CMS Collaboration. *WorkBook MiniAOD*. Accessed: 2025-02-13. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookMiniAOD.

[40] The CMS Collaboration. *The CMS NanoAOD data tier*. Accessed: 2025-02-13. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookNanoAOD.

[41] The CMS Collaboration. *Particle-Flow Event Reconstruction in CMS and Performance for Jets, Taus, and MET*. Tech. rep. Geneva: CERN, 2009. URL: https://cds.cern.ch/record/1194487.

[42] The CMS Collaboration. 'Performance of CMS muon reconstruction in pp collision events ats = 7TeV'. In: *Journal of Instrumentation* 7.10 (Oct. 2012), P10002–P10002. ISSN: 1748-0221. DOI: 10.1088/1748-0221/7/10/p10002. URL: http://dx.doi.org/10.1088/1748-0221/7/10/P10002.

[43] The CMS Collaboration. *Baseline muon selections for Run-II (CMS internal, rev. 59)*. Accessed: 2023-01-11. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/SWGuideMuonIdRun2.

[44] M. Cacciari, G. P. Salam and G. Soyez. 'The anti-ktjet clustering algorithm'. In: *Journal of High Energy Physics* 2008.04 (Apr. 2008), pp. 063–063. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2008/04/063. URL: http://dx.doi.org/10.1088/1126-6708/2008/04/063.

[45] Yu.L Dokshitzer et al. 'Better jet clustering algorithms'. In: *Journal of High Energy Physics* 1997.08 (Aug. 1997), pp. 001–001. ISSN: 1029-8479. DOI: 10.1088/1126-6708/1997/08/001. URL: http://dx.doi.org/10.1088/1126-6708/1997/08/001.

[46] The CMS Collaboration. 'Missing transverse energy performance of the CMS detector'. In: *Journal of Instrumentation* 6.09 (Nov. 2011), P09001–P09001. ISSN: 1748-0221. DOI: 10.1088/1748-0221/6/09/p09001. URL: http://dx.doi.org/10.1088/1748-0221/6/09/P09001.

[47] The CMS collaboration. 'Performance of the CMS missing transverse momentum reconstruction in pp data at s = 8 TeV'. In: *Journal of Instrumentation* 10.02 (Feb. 2015), P02006–P02006. ISSN: 1748-0221. DOI: 10.1088/1748-0221/10/02/p02006. URL: http://dx.doi.org/10.1088/1748-0221/10/02/P02006.

[48] The CMS Collaboration. *Pileup Jet Identification*. Tech. rep. Geneva: CERN, 2013. URL: https://cds.cern.ch/record/1581583.

[49] D. Bertolini et al. 'Pileup per particle identification'. In: *Journal of High Energy Physics* 2014.10 (Nov. 2014). ISSN: 1029-8479. DOI: 10.1007/jhep10(2014)059. URL: http://dx.doi.org/10.1007/JHEP10(2014)059.

[50] The CMS Collaboration. *WorkBook MiniAOD*. Accessed: 2025-02-14. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/////MissingETOptionalFiltersRun2.

[51] The CMS Collaboration. *Jet Energy Corrections (JEC) – CMS OpenData Guide*. Accessed: 2025-02-14. webpage. URL: https://cms-opendata-guide.web.cern.ch/analysis/systematics/objectsuncertain///////jetmetuncertain.

[52] C. Verstege. 'Measurement of the Triple-Differential Cross-Section of Z+Jet Production with the CMS Detector at 13 TeV'. MA thesis. Karlsruhe Institute of Technology (KIT), 2022. URL: https://publish.etp.kit.edu/record/22168.

[53] M. M. Horzela. 'Measurement of Triple-Differential Z+Jet Cross Sections with the CMS Detector at 13 TeV and Modelling of Large-Scale Distributed Computing Systems'. PhD thesis. KIT Karlsruhe, 2023. DOI: 10.5445/IR/1000165566. URL: https://publikationen.bibliothek.kit.edu/1000165566.

[54] The CMS Collaboration. *Jet Energy Corrections: Official Software Tools for applying JEC Corrections and Uncertainties (CMS internal)*. Accessed: 2025-02-14. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookJetEnergyCorrections.

[55] The CMS Collaboration. *Physics Objects: Jet Corrections – CMS OpenData Workshop 2022*. Accessed: 2025-02-14. webpage. URL: https://cms-opendata-workshop.github.io/workshop2022-lesson-physics-objects/08-jecjer/index.html.

[56] M. Cacciari and G. P. Salam. 'Pileup subtraction using jet areas'. In: *Physics Letters B* 659.1–2 (Jan. 2008), pp. 119–126. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2007.09.077. URL: http://dx.doi.org/10.1016/j.physletb.2007.09.077.

[57] The CMS Collaboration. 'Jet energy scale and resolution measurement with Run 2 Legacy Data Collected by CMS at 13 TeV'. In: (2021). URL: https://cds.cern.ch/record/2792322.

[58] G. Agarwal. *Jet Energy Scale and Resolution Measurements in CMS*. Tech. rep. 6 pages, 6 figures, Contribution to 41st International Conference on High Energy physics - ICHEP 2022, Accepted for publication in POS. 2022. DOI: 10.22323/1.414.0652. arXiv: 2301.02175. URL: https://cds.cern.ch/record/2847453.

[59] The CMS Collaboration/JERC group. *cms-jet/JECDatabase (GitHub)*. Accessed: 2023-01-12. webpage. URL: https://github.com/cms-jet/JECDatabase/.

[60] The CMS Collaboration. *Run-2 UltraLegacy Datasets for Analysis (CMS internal)*. Accessed: 2023-01-14. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMS/PdmVRun2LegacyAnalysis.

[61] The CMS Collaboration. *IOV and IOV Metadata In a Nutshell (CMS internal)*. Accessed: 2025-02-14. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic////////////////SWGuideIOVAndIOVMetaDataInANutshell.

[62] The CMS Collaboration. *Recommended Jet Energy Corrections and Uncertainties For Data and MC (CMS internal)*. Accessed: 2023-01-08. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/JECDataMC.

[63] KIT-CMS/R. Hofsaess. *Kappa – JEC Release (GitHub)*. Accessed: 2025-02-14. webpage. URL: https://github.com/KIT-CMS/Kappa/releases/tag/jec_final.

[64] The CMS Collaboration. *Noise Filter Recommendations for Run II & Run III (CMS internal)*. Accessed: 2023-01-08. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/MissingETOptionalFiltersRun2#Noise_Filter_Recommendations_for.

[65] The CMS Collaboration. *Jet Identification (CMS internal)*. Accessed: 2023-01-10. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMS/JetID.

[66] The CMS Collaboration. *Baseline muon selections for Run-II (CMS internal)*. Accessed: 2025-02-14. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/SWGuideMuonIdRun2.

[67] The CMS Collaboration. *Rochester corrections (CMS internal)*. Accessed: 2023-01-07. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/RochcorMuon.

[68] The CMS Collaboration. *E/gamma Energy Corrections – Residual Scale Smearing Corrections (CMS internal)*. Accessed: 2023-01-07. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/EgammaRunIIRecommendations#Residual_Scale_Smearing_Correcti.

[69] The CMS Collaboration. *Jet Energy Resolution (CMS internal)*. Accessed: 2023-01-12. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/JetResolution.

[70] The CMS Collaboration. *Jet energy scale uncertainty sources (CMS internal)*. Accessed: 2025-02-14. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMS/JECUncertaintySources#Jet_energy_scale_uncertainty_sou.

[71] The CMS Collaboration. *HLT Paths RunII List (CMS internal)*. Accessed: 2023-01-07. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/HLTPathsRunIIList.

[72] KIT-CMS/R. Hofsaess. *Excalibur – JEC Release (GitHub)*. Accessed: 2025-02-14. webpage. URL: https://github.com/KIT-CMS/Excalibur/releases/tag/jec_final.

[73] KIT-CMS/R. Hofsaess. *Artus – JEC Release (GitHub)*. Accessed: 2025-02-14. webpage. URL: https://github.com/KIT-CMS/Artus/releases/tag/jec_final.

[74] The CMS Collaboration. *Hot zone maps (CMS internal)*. Accessed: 2022-12-22. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMS/JetID#Hot_zone_maps.

[75] The CMS Collaboration/JERC group. *cms-jet/JECDatabase/jet_veto_maps (GitHub)*. Accessed: 2022-12-22. webpage. URL: https://github.com/cms-jet/JECDatabase/tree/master/jet_veto_maps.

[76] The CMS Collaboration. *Jet identification in high pile-up environment (PileupJetID) for Ultra Legacy Data (CMS internal)*. Accessed: 2023-01-10. webpage. URL: https://twiki.cern.ch/twiki/bin/viewauth/CMS/PileupJetIDUL.

[77] The CMS Collaboration. *CMS Luminosity – Public Results*. Accessed: 2025-02-02. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults#Run_2_charts_of_luminosity.

[78] The CMS Collaboration. *CMS Pile-up simulation – CMS OpenData*. Accessed: 2025-02-16. webpage. URL: https://opendata.cern.ch/docs/cms-guide-pileup-simulation.

[79] R. Hofsaess and M. Horzela. *L3Res Z+Jet Combination Files (CMS internal)*. Accessed: 2023-01-12. webpage. URL: https://gitlab.cern.ch/cms-jetmet/JERCProtoLab/-///////////commit/e6591330564e2230af894ab6858e7f57827056e1.

[80] Scikit-HEP Project. *scikit-hep/coffea (GitHub)*. Accessed: 2025-02-17. webpage. URL: https://github.com/scikit-hep/coffea.

[81] R. Hofsaess and M. Horzela. 'L3Res Z+Jet UL Update (talk)'. In: *JERC Working Group Meeting*. Jan. 2023. URL: https://indico.cern.ch/event/1460070/contributions/6175247/attachments/2957156/5200544/Monitoring_RHofsaess.pdf.

[82] CMS Offline Software and Computing. *CMS Phase-2 Computing Model: Update Document*. Tech. rep. CERN, 2022. URL: https://cds.cern.ch/record/2815292.

[83] E. Martelli. 'LHCOPN, LHCONE and evolution in research network (talk)'. In: *ATCF8 TIFR Mumbai*. Sept. 2024. URL: https://indico.cern.ch/event/1411901/.

[84] A. Kelleher. *Moore's Law — Now and in the Future*. Accessed: 2024-12-08. webpage. URL: https://www.intel.de/content/www/de/de/newsroom/opinion/moore-law-now-and-in-the-future.html.

[85] D. Britton, S. Campana, T. Boccali et al. 'WLCG Strategy 2024 - 2027'. In: *Zenodo* (2023). DOI: 10.5281/zenodo.12623280. URL: https://zenodo.org/records/12623280.

[86] I. Bird et al. *LHC computing Grid. Technical Design Report*. Technical design report. LCG. CERN, June 2005. ISBN: 9290832533, 9789290832539. URL: https://cds.cern.ch/record/840543.

[87] M. Aderholz et al. *Models of Networked Analysis at Regional Centres for LHC Experiments (MONARC). Phase 2 Report*. Tech. rep. CERN, Apr. 2000. URL: https://cds.cern.ch/record/510694.

[88] WLCG/CERN. *The WLCG*. Accessed: 2024-12-12. webpage. URL: https://home.cern/science/computing/grid.

[89] CRIC/WLCG. *WLCG pledges in CRIC*. Accessed: 2024-12-17. webpage. URL: http://wlcg-cric.cern.ch/core/pledge/list/.

[90] WLCG/CERN. *The WLCG Tiers*. Accessed: 2024-12-12. webpage. URL: https://wlcg-public.web.cern.ch/tiers.

[91] C. Heidecker et al. 'Boosting data-intensive end-user analyses: TOpAS — a dedicated high throughput cluster (talk)'. In: *ErUM-Data meeting*. Apr. 2020. URL: https://indico.physik.uni-muenchen.de/event/35/contributions/328/attachments/139/207/Erum-Data-meeting-2020-04-17-cheidecker.pdf.

[92] I. Bird et al. *Update of the Computing Models of the WLCG and the LHC Experiments*. Tech. rep. CERN, Apr. 2014. URL: https://cds.cern.ch/record/1695401.

[93] E. Martelli and S. Stancu. 'LHCOPN and LHCONE: Status and Future Evolution'. In: *Journal of Physics: Conference Series* 664 (May 2015), p. 6. DOI: 10.1088/1742-6596/664/5/052025. URL: https://iopscience.iop.org/article/10.1088/1742-6596/664/5/052025.

[94] E. Martelli. 'Evolving the LHCOPN and LHCONE networks to support HL-LHC computing requirements'. In: *EPJ Web of Conferences* 295 (May 2024). DOI: 10.1051/epjconf/202429507016. URL: https://www.epj-conferences.org/articles/epjconf/abs/2024/05/epjconf_chep2024_07016/epjconf_chep2024_07016.html.

[95] E. Martelli. 'LHCOPN update (talk)'. In: *LHCOPN–LHCONE meeting #53, IHEP Beijing*. Oct. 2024. URL: https://indico.cern.ch/event/1410638/.

[96] A. Barczyk and D. Foster. 'LHC Open Network Environment LHCONE (talk)'. In: *Software Computing Workshop, CERN*. Apr. 2011. URL: https://indico.cern.ch/event/119169/.

[97] E. Martelli. *LHCOPN map*. Status: 10/2024. Accessed: 2024-12-07. webpage. URL: https://twiki.cern.ch/twiki/bin/view/LHCOPN/OverallNetworkMaps.

[98] M. Lassnig, C. Wissing et al. 'WLCG/DOMA Data Challenge 2024: Final Report'. In: *Zenodo* (June 2024). DOI: 10.5281/zenodo.11444180. URL: https://zenodo.org/records/11444180.

[99] E. Martelli. 'Summary notes (talk)'. In: *LHCOPN-LHCONE meeting #52, INFN Catania*. Apr. 2024. URL: https://indico.cern.ch/event/1349135/.

[100] The HTCondor Team. *HTCondor*. latest. DOI: 10.5281/zenodo.2579447. URL: https://zenodo.org/records/14238973.

[101] T. Tannenbaum et al. 'Condor – A Distributed Job Scheduler'. In: *Beowulf Cluster Computing with Linux*. Ed. by T. Sterling. MIT Press, Oct. 2001. URL: https://chtc.cs.wisc.edu/doc/beowulf-chapter-rev1.pdf.

[102] M.J. Litzkow, M. Livny and M.W. Mutka. 'Condor - A Hunter of Idle Workstations'. In: *Computer Sciences Technical Report*. 730. Dec. 1987. URL: https://minds.wisconsin.edu/bitstream/handle/1793/58896/TR730.pdf.

[103] HTCondor. *HTCondor documentation*. Accessed: 2024-12-12. webpage. URL: https://htcondor.readthedocs.io/.

[104] B. Holzman et al. *HEPCloud, a New Paradigm for HEP Facilities: CMS Amazon Web Services Investigation*. 2017. arXiv: 1710.00100. URL: https://arxiv.org/abs/1710.00100.

[105] Wikipedia. *chroot*. Accessed: 2024-12-09. webpage. URL: https://en.wikipedia.org/wiki/Chroot.

[106] Linux. *namespaces – Linux manual page*. Accessed: 2024-12-08. webpage. URL: https://man7.org/linux/man-pages/man7/namespaces.7.html.

[107] P. Menage. *CGroups*. Accessed: 2024-12-10. webpage. URL: https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt.

[108] UnionFS project. *UnionFS*. Accessed: 2024-12-10. webpage. URL: https://unionfs.filesystems.org/.

[109] Docker Inc. *Docker*. Accessed: 2024-12-11. webpage. URL: https://www.docker.com/resources/what-container/.

[110] Docker Inc. *Docker*. Accessed: 2024-12-09. webpage. URL: https://www.docker.com/.

[111] The Kubernetes Authors / The Linux Foundation. *Kubernetes*. Accessed: 2024-12-11. webpage. URL: https://kubernetes.io/.

[112] Red Hat, Inc. *Red Hat OpenShift*. Accessed: 2024-12-11. webpage. URL: https://www.redhat.com/de/technologies/cloud-computing/openshift.

[113] Apptainer project. *Apptainer*. Accessed: 2024-12-10. webpage. URL: https://apptainer.org/.

[114] J. Blomer et al. 'Distributing LHC application software and conditions databases using the CernVM file system'. In: *Journal of Physics: Conference Series* 331.4 (Dec. 2011), p. 042003. DOI: 10.1088/1742-6596/331/4/042003. URL: https://dx.doi.org/10.1088/1742-6596/331/4/042003.

[115] J. Blomer et al. *CVMFS*. latest. DOI: doi.org/10.5281/zenodo.1010441. URL: https://zenodo.org/records/4114078.

[116] CernVM Team. *CVMFS documentation*. Accessed: 2024-12-09. webpage. URL: https://cvmfs.readthedocs.io/.

[117] D. Wessels et al. *squid-cache (GitHub)*. Accessed: 2024-12-09. webpage. URL: https://www.squid-cache.org/,%20https://github.com/squid-cache.

[118] CernVM Team. *cvmfsexec (GitHub)*. Accessed: 2024-12-11. webpage. URL: https://github.com/cvmfs/cvmfsexec.

[119] R. Bachmann et al. 'CERN's Run 3 Tape Infrastructure'. In: *HEPiX Spring*. Apr. 2022. URL: https://indico.cern.ch/event/1123214/.

[120] J. Leduc et al. 'CERN Tape Archive Run 3 Production Experience'. In: *EPJ Web of Conferences* 295 (May 2024), p. 01049. DOI: 10.1051/epjconf/202429501049. URL: https://www.epj-conferences.org/articles/epjconf/pdf/2024/05/epjconf_chep2024_01049.pdf.

[121] M. Barisits, T. Beermann, F. Berghaus et al. 'Rucio: Scientific Data Management'. In: *Computational Software in Big Science* 3.1 (2019), p. 11. DOI: 10.1007/s41781-019-0026-3. URL: https://doi.org/10.1007/s41781-019-0026-3.

[122] Rucio/CERN. *Rucio*. Accessed: 2024-12-11. webpage. URL: https://rucio.cern.ch/.

[123] A. Afaq et al. 'The CMS dataset bookkeeping service'. In: *Journal of Physics: Conference Series* 119.7 (July 2008), p. 072001. DOI: 10.1088/1742-6596/119/7/072001. URL: https://dx.doi.org/10.1088/1742-6596/119/7/072001.

[124] FTS Team/CERN. *File Transfer Service*. Accessed: 2024-12-08. webpage. URL: https://fts.web.cern.ch/fts.

[125] A. Dorigo et al. 'XROOTD – A highly scalable architecture for data access'. In: *WSEAS Transactions on Computers* 4 (Apr. 2005), pp. 348–353.

[126] A. Hanushevsky et al. *XRootD*. latest. DOI: 10.5281/zenodo.8101777.. URL: https://zenodo.org/records/14245021.

[127] The XRootD Project. *Official XRootD documentation*. Accessed: 2024-12-14. webpage. URL: https://xrootd.github.io/docs.html.

[128] A. Hanushevsky. 'XRootD Server Plug-ins'. In: *XRootD Workshop, Ljubljana*. Mar. 2023. URL: https://indico.cern.ch/event/875381/contributions/5305955/.

[129] A. Arora et al. *400Gbps benchmark of XRootD HTTP-TPC*. Dec. 2023. arXiv: 2312.12589. URL: https://arxiv.org/abs/2312.12589.

[130] B. Bockelman. 'Demonstrator Analysis 200 Gb/s (talk)'. In: *WLCG/HSF Workshop, DESY*. May 2024. URL: https://indico.cern.ch/event/1369601/contributions/5924000/.

[131] A. Held, B. Bockelman and S. Oksana. 'The 200 Gbps Challenge: Imagining HL-LHC analysis facilities (talk)'. In: *CHEP, Krakow*. Oct. 2024. URL: https://indico.cern.ch/event/1338689/contributions/6009824/.

[132] A. Aimar et al. 'MONIT: Monitoring the CERN Data Centres and the WLCG Infrastructure'. In: *EPJ Web of Conferences* 214 (Sept. 2019), p. 08031. DOI: 10.1051/epjconf/201921408031. URL: https://www.epj-conferences.org/articles/epjconf/pdf/2019/19/epjconf_chep2018_08031.pdf.

[133] MONIT/CERN. *MONIT*. Accessed: 2024-12-15. webpage. URL: https://monit.web.cern.ch/.

[134] MONIT/CERN. *Official MONIT documentation*. Accessed: 2024-12-15. webpage. URL: https://monit-docs.web.cern.ch/.

[135] OpenSearch Project. *OpenSearch and OpenSearch Dashboards*. Accessed: 2024-12-15. webpage. URL: https://opensearch.org/.

[136] Grafana Labs. *Grafana*. Accessed: 2024-12-15. webpage. URL: https://grafana.com/.

[137] E. Tejedor et al. 'Facilitating collaborative analysis in SWAN'. In: *EPJ Web of Conferences* 214 (2019), p. 07022. DOI: 10.1051/epjconf/201921407022. URL: https://www.epj-conferences.org/articles/epjconf/abs/2019/19/epjconf_chep2018_07022/epjconf_chep2018_07022.html.

[138] K. Shvachko et al. 'The Hadoop Distributed File System'. In: *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. 2010, pp. 1–10. DOI: 10.1109/MSST.2010.5496972.

[139] SWAN/CERN. *SWAN documentation*. Accessed: 2024-12-15. webpage. URL: https://swan.docs.cern.ch/.

[140] CMS CompOps. *CMS SAM test repository*. Accessed: 2024-12-15. webpage. URL: https://gitlab.cern.ch/etf/cmssam.

[141] CMS CompOps. *CMS SAM test overview*. Accessed: 2024-12-15. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/CompOpsSAMTests.

[142] CMS CompOps. *CMS site readiness reports*. Accessed: 2024-12-15. webpage. URL: https://cmssst.web.cern.ch/sitereadiness/report.html.

[143] D. van der Ster et al. 'HammerCloud: A Stress Testing System for Distributed Analysis'. In: *Journal of Physics Conference Series*. Vol. 331. IOP, Dec. 2011, p. 072036. DOI: 10.1088/1742-6596/331/7/072036. URL: https://iopscience.iop.org/article/10.1088/1742-6596/331/7/072036.

[144] WLCG HammerCloud/CERN. *CMS HammerCloud test overview*. Accessed: 2024-12-15. webpage. URL: https://hammercloud.cern.ch/hc/app/cms/.

[145] Rucio Team. *Rucio monitoring*. Accessed: 2024-12-15. webpage. URL: https://rucio.github.io/documentation/operator/monitoring/.

[146] OSG. *XRootD Monitoring Shoveler (GitHub)*. Accessed: 2024-12-15. webpage. URL: https://github.com/opensciencegrid/xrootd-monitoring-shoveler.

[147] K. Ellis. 'Shoveler XRootD monitoring'. In: *XRootD and FTS Workshop, STFC UK*. Sept. 2024. URL: https://indico.cern.ch/event/1386888/contributions/6093094/.

[148] G. Borja et al. ‘New XrootD Monitoring Implementation’. In: *EPJ Web of Conferences* 295 (May 2024), p. 01040. DOI: 10.1051/epjconf/202429501040. URL: https://www.epj-conferences.org/articles/epjconf/abs/2024/05/epjconf_chep2024_01040/epjconf_chep2024_01040.html.

[149] The CMS Collaboration. *dmwm/WMCore (GitHub)*. Accessed: 2024-12-08. webpage. URL: https://github.com/dmwm/WMCore.

[150] R. Hofsaess. *CMS LogMatching (GitHub)*. Accessed: 2024-12-15. webpage. URL: https://github.com/RHofsaess/logmatching.

[151] R. Hofsaess. ‘Making the Most of CMS Monitoring (talk)’. In: *Fall 2024 Offline Software and Computing Week*. Oct. 2024. URL: https://indico.cern.ch/event/1460070/contributions/6175247/attachments/2957156/5200544/Monitoring_RHofsaess.pdf.

[152] R. Hofsaess. *CMS LogCollect (GitHub)*. Accessed: 2024-12-15. webpage. URL: https://github.com/RHofsaess/logcollect.

[153] C. Winter, A. Gottmann et al. *HappyFace4 Meta-Monitoring (GitHub)*. Accessed: 2024-12-16. webpage. URL: https://github.com/HappyFaceMonitoring.

[154] C. Winter, A. Gottmann et al. *HappyFace4 documentation*. Accessed: 2024-12-16. webpage. URL: https://happyface.pages.etp.kit.edu/HappyFaceCore/.

[155] A. Anisenkov et al. ‘CRIC: Computing Resource Information Catalogue as a unified topology system for a large scale, heterogeneous and dynamic computing infrastructure’. In: *EPJ Web Conf.* 245 (2020), p. 03032. DOI: 10.1051/epjconf/202024503032. URL: https://cds.cern.ch/record/2758813.

[156] The HEPiX Benchmarking Group. *HS06*. Accessed: 2024-12-08. webpage. URL: http://w3.hepix.org/benchmarking/HS06.html.

[157] Standard Performance Evaluation Corporation. *SPEC CPU® 2006*. Accessed: 2024-12-08. webpage. URL: https://www.spec.org/cpu2006/.

[158] N. Szczepanek, D. Giordano et al. ‘HEP Benchmark Suite: Enhancing Efficiency and Sustainability in Worldwide LHC Computing Infrastructures (talk)’. In: *ACAT, Stony Brook*. Mar. 2024. URL: https://indico.cern.ch/event/1330797/contributions/5796505/.

[159] D. Giordano et al. *HEPScore: A new CPU benchmark for the WLCG*. 2023. arXiv: 2306.08118. URL: https://arxiv.org/abs/2306.08118.

[160] HEPiX Benchmarking Group. *HS23 Scores*. Accessed: 2024-12-12. webpage. URL: http://w3.hepix.org/benchmarking/scores_HS23.html.

[161] A. Coveney et al. *apel/apel*. latest. DOI: 10.5281/zenodo.1694524. URL: https://doi.org/10.5281/zenodo.1694524.

[162] A. Coveney et al. *APEL*. Accessed: 2024-12-08. webpage. URL: https://apel.github.io/.

[163] EGI. *EGI Accounting*. Accessed: 2024-12-12. webpage. URL: https://accounting.egi.eu/.

[164] M. Boehler et al. *AUDITOR*. 2024. DOI: 10.5281/zenodo.12653483. URL: https://zenodo.org/records/13239266.

[165] M. et al Boehler. 'AUDITOR: Accounting for opportunistic resources'. In: vol. 295. May 2024. DOI: 10.1051/epjconf/202429504008. URL: https://www.epj-conferences.org/articles/epjconf/abs/2024/05/epjconf_chep2024_04008/epjconf_chep2024_04008.html.

[166] M. Boehler et al. *ALU-Schumacher/AUDITOR (GitHub)*. Accessed: 2024-12-08. webpage. URL: https://github.com/ALU-Schumacher/AUDITOR.

[167] M. Fischer et al. *MatterMiners/cobald (latest)*. DOI: 10.5281/zenodo.1887872. URL: https://zenodo.org/records/8199049.

[168] M. Fischer et al. *COBalD documentation*. Accessed: 2024-12-20. webpage. URL: https://cobald.readthedocs.io/.

[169] M. Schnepf et al. 'Dynamic Integration and Management of Opportunistic Resources for HEP'. In: *EPJ Web of Conferences* 214 (Sept. 2019), p. 08009. DOI: 10.1051/epjconf/201921408009. URL: https://www.epj-conferences.org/articles/epjconf/abs/2019/19/epjconf_chep2018_08009/epjconf_chep2018_08009.html.

[170] M. Giffels et al. *MatterMiners/tardis*. latest. DOI: 10.5281/zenodo.2240605. URL: https://zenodo.org/records/14033952.

[171] M. Giffels et al. *COBalD/TARDIS documentation*. Accessed: 2024-12-20. webpage. URL: https://cobald-tardis.readthedocs.io.

[172] Gemeinsame Wissenschaftskonferenz (GWK). *Pressemitteilung 11/2020*. Accessed: 2024-12-23. Nov. 2020. URL: https://www.gwk-bonn.de/fileadmin/Redaktion/Dokumente/Pressemitteilungen/pm2020-11.pdf.

[173] SCC/KIT. *HoreKa*. Accessed: 2024-12-23. webpage. URL: https://www.scc.kit.edu/en/services/horeka.php.

[174] TOP500.org. *June 2024, TOP500 list*. Accessed: 2024-12-23. URL: https://top500.org/lists/top500/2024/06/.

[175] S. Wiebe and C. Könemann. *Press Release 038/2024*. Accessed: 2024-12-23. May 2024. URL: https://www.kit.edu/kit/english/pi_2024_038_kit-supercomputer-one-of-the-worlds-most-energy-efficient.php.

[176] SCC/KIT. *NHR@KIT: Application for computing time*. Accessed: 2024-12-23. webpage. URL: https://www.nhr.kit.edu/english/application.php.

[177] TOP500.org. *November 2024, TOP500 list*. Accessed: 2024-12-26. webpage. URL: https://top500.org/lists/top500/2024/11/.

[178] C. Acosta-Silva et al. 'Integration of the Barcelona Supercomputing Center for CMS computing: Towards large scale production'. In: *EPJ Web of Conferences* 295 (May 2024), p. 07027. DOI: 10.1051/epjconf/202429507027. URL: https://www.epj-conferences.org/articles/epjconf/abs/2024/05/epjconf_chep2024_07027/epjconf_chep2024_07027.html.

[179] R. F. von Cube. 'Dynamic Integration of Heterogeneous Computing Resources and Jet Energy Calibration for the CMS Experiment'. PhD thesis. Karlsruher Institut für Technologie (KIT), 2022. 83 pp. DOI: 10.5445/IR/1000151252.

[180] CMS Offline Software and Computing. *CMS Offline and Computing Public Results*. Accessed: 2024-12-26. webpage. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic//////CMSOfflineComputingResults.

[181] A. Mohapatra /CMS Monitoring. *CMS HPC usage (CMS internal)*. URL: https://twiki.cern.ch/twiki/bin/view/CMS/FacilitiesServicesMeetingHPCOpp.

[182] Komitee für Elementarteilchenphysik in Deutschland. *Perspektivpapier der Teilchenphysiker:innen in Deutschland*. Accessed: 2024-12-26. Mar. 2022. URL: https://www.ketweb.de/sites/site_ketweb/content/e199639/e312771/KET-Computing-Strategie-HL-LHC-final.pdf.

[183] NHR-Verein e.V. *Nationales Hochleistungsrechnen*. Accessed: 2024-12-23. wepage. URL: https://www.nhr-verein.de/en.

[184] P. Saidev et al. 'Goettingen - First performance test results of the ATLAS jobs on the integrated NHR Emmy HPC'. In: *Fidium Collaboration Meeting, Aachen*. Accessed: 2024-12-23. Oct. 2024. URL: https://indico.desy.de/event/46129/contributions/174407/.

[185] M. Schnepf. *Private communication – inofficial benchmarks*. 2024.

[186] A. Krull et al. 'Experience with ARM WNs at the WLCG Tier1 GridKa (Poster)'. In: *CHEP, Krakow*. 2024. URL: https://indico.cern.ch/event/1338689/contributions/6011886/.

[187] T. Kress. *Private communication*. 2024.

[188] J. Molina. 'Optimal XCache service for the CMS experiment in Spain'. In: *ACAT, Stony Brook*. Mar. 2024. URL: https://indico.cern.ch/event/1330797/contributions/5796642/.

[189] I. Vukotic. 'Scheduling with Virtual Placement (talk)'. In: *ATLAS Sites Jamboree and HPC strategy*. Mar. 2019. URL: https://indico.cern.ch/event/770307/#14-doma-access-caches.

[190] I. Vukotic. 'XCache experience Virtual Placement (and ServiceX) (talk)'. In: *XRootD and FTS Workshop, Ljubljana*. Mar. 2023. URL: https://indico.cern.ch/event/875381/contributions/5303591/.

[191] R. Hofsaess. *XBuffer (GitHub)*. Accessed: 2024-12-15. webpage. URL: https://github.com/RHofsaess/XBuffer.

[192] A. Hayrapetyan et al. 'Luminosity determination using Z boson production at the CMS experiment'. In: *The European Physical Journal C* 84.1 (Jan. 2024). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-023-12268-2. URL: http://dx.doi.org/10.1140/epjc/s10052-023-12268-2.

[193] W. Johnston. *LHCONE map*. Status: 10/2024. Accessed: 2024-12-07. webpage. URL: https://twiki.cern.ch/twiki/bin/view/LHCONE/LhcOneMaps.

[194] K. Ellis et al. 'WLCG Data Challenge 24: LHC experiment experiences (talk)'. In: *WLCG workshop, DEYS*. May 2024. URL: https://indico.cern.ch/event/1369601/contributions/5923582/attachments/2855546/4993849/DC24_LHCexps_WLCG.pdf.

[195] LHC/CERN. *LHC schedule*. Status: 11/2024. Accessed: 2024-12-28. URL: http://lhc-commissioning.web.cern.ch/schedule/LHC-long-term.htm.

A. Appendix

The details are not the details. They make the design.

(Charles Eames)

A.1. Additions to the Jet Energy Calibration

A.1.1. Datasets

Listing A.1: List of datasets utilized for the L3res corrections.

```
            ------------ 2016UL: ------------
Data: (GT: 106X_dataRun2_v35)
  /DoubleEG/Run2016B-ver1_HIPM_UL2016_MiniAODv2-v1/MINIAOD
  /DoubleEG/Run2016B-ver2_HIPM_UL2016_MiniAODv2-v3/MINIAOD
  /DoubleEG/Run2016(C|D|E|F)-HIPM_UL2016_MiniAODv2-v1/MINIAOD
  /DoubleEG/Run2016(F|G|H)-UL2016_MiniAODv2-v1/MINIAOD
  /DoubleMuon/Run2016B-ver(1|2)_HIPM_UL2016_MiniAODv2-v1/MINIAOD
  /DoubleMuon/Run2016(C|D|E|F)-HIPM_UL2016_MiniAODv2-v1/MINIAOD
  /DoubleMuon/Run2016(F|G|H)-UL2016_MiniAODv2-v1/MINIAOD
MC: (GT: MC 2016APV: 106X_mcRun2_asymptotic_preVFP_v11
         MC 2016: 106X_mcRun2_asymptotic_v17)
  /DYJetsToLL_M-50_TuneCP5_13TeV-amcatnloFXFX-pythia8/
      RunIISummer20UL16MiniAODv2_106X_mcRun2_asymptotic_v17-v1/
      MINIAODSIM
  /DYJetsToLL_M-50_TuneCP5_13TeV-amcatnloFXFX-pythia8/
      RunIISummer20UL16MiniAODAPVv2-106X_mcRun2_asymptotic_
      preVFP_v11-v1/MINIAODSIM

            ------------ 2017UL: ------------
Data: (GT: 106X_dataRun2_v35)
  /DoubleEG/Run2017B-UL2017_MiniAODv2-v1/MINIAOD
  /DoubleEG/Run2017C-UL2017_MiniAODv2-v2/MINIAOD
  /DoubleEG/Run2017D-UL2017_MiniAODv2-v1/MINIAOD
  /DoubleEG/Run2017E-UL2017_MiniAODv2-v1/MINIAOD
  /DoubleEG/Run2017F-UL2017_MiniAODv2-v2/MINIAOD
  /DoubleMuon/Run2017(B|C|D)-UL2017_MiniAODv2-v1/MINIAOD
  /DoubleMuon/Run2017E-UL2017_MiniAODv2-v2/MINIAOD
  /DoubleMuon/Run2017F-UL2017_MiniAODv2-v1/MINIAOD
MC: (GT: 106X_mc2017_realistic_v9)
  /DYJetsToLL_M-50_TuneCP5_13TeV-amcatnloFXFX-pythia8/
      RunIISummer20UL17MiniAODv2-106X_mc2017_realistic_v9-v2/
      MINIAODSIM

            ------------ 2018UL: ------------
```

```
Data: (GT: 106X_dataRun2_v37)
  /EGamma/Run2018(A|B|C)-UL2018_MiniAODv2-v1/MINIAOD
  /EGamma/Run2018D-UL2018_MiniAODv2-v2/MINIAOD
  /DoubleMuon/Run2018(A|B|C|D)-UL2018_MiniAODv2-v1/MINIAOD
MC: (GT: 106X_upgrade2018_realistic_v16_L1v1)
  /DYJetsToLL_M-50_TuneCP5_13TeV-amcatnloFXFX-pythia8/
      RunIISummer20UL18MiniAODv2-106X_upgrade2018_realistic_
      v16_L1v1-v2/MINIAODSIM
```

A.1.2. MET Filters

The filters listed below are the ones applied for the derivation of the L3 residual corrections in Section 3.1. Details and recommendations are provided here: [50].

2016UL:

- Flag_goodVertices
- Flag_globalSuperTightHalo2016Filter
- Flag_HBHENoiseFilter
- Flag_HBHENoiseIsoFilter
- Flag_EcalDeadCellTriggerPrimitiveFilter
- Flag_BadPFMuonFilter
- Flag_BadPFMuonDzFilter
- Flag_eeBadScFilter

2017&2018 UL:

- Flag_goodVertices
- Flag_globalSuperTightHalo2016Filter
- Flag_HBHENoiseFilter
- Flag_HBHENoiseIsoFilter
- Flag_EcalDeadCellTriggerPrimitiveFilter
- Flag_BadPFMuonFilter
- Flag_BadPFMuonDzFilter
- Flag_eeBadScFilter
- Flag_ecalBadCalibFilter

A.1.3. Trigger

Listing A.2: List of triggers utilized for the L3res corrections. All available triggers for Run 2 can be found in [71].

```
    ———————— 2016UL ————————
HLT_Ele23_Ele12_CaloIdL_TrackIdL_IsoVL_DZ
HLT_Mu17_TrkIsoVVL_Mu8_TrkIsoVVL_DZ (unprescaled)

    ———————— 2017UL ————————
HLT_Mu17_TrkIsoVVL_Mu8_TrkIsoVVL_DZ_Mass8
HLT_Ele23_Ele12_CaloIdL_TrackIdL_IsoVL

    ———————— 2018UL ————————
HLT_Mu17_TrkIsoVVL_Mu8_TrkIsoVVL_DZ_Mass8
HLT_Ele23_Ele12_CaloIdL_TrackIdL_IsoVL
```

A.1.4. Selections and IDs

Listing A.3: List of selections and IDs utilized for the L3res corrections.

```
    ———————— AK4 CHS jet collection ————————
JER Smearing:
  https://twiki.cern.ch/twiki/bin/viewauth/CMS/JetResolution
    2016UL (hybrid): Summer20UL16_JRV3
    2017UL (hybrid): Summer19UL17_JRV3
    2018UL (hybrid): Summer19UL18_JRV2

JEC: https://twiki.cern.ch/twiki/bin/viewauth/CMS/JECDataMC
    (L1, L1L2L3 and L1L2L3L2Res)
    2016UL: Summer19UL16(APV)_Run(BCD|EF|BCDEF|FGH)_V7
    2017UL: Summer19UL17_V6
    2018UL: Summer19UL18_V5

ID: https://twiki.cern.ch/twiki/bin/view/CMS/JetID13TeVUL
    2016UL: tightlepveto, Version: 2016UL
    2017UL: tightlepveto, Version: 2017/18UL
    2018UL: tightlepveto, Version: 2017/18UL

Lepton cleaning: Only the leptons from the reconstructed Z are
cleaned from the jet list to avoid removing leptons from decays,
etc, which could negatively influence the total energy of the
jets/event. However, since the tight lep veto is used, the
influence of misused leptons as jets should be prevented anyway.

    ———————— Lepton IDs and Isolation ————————
Muons:
  https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideMuonIdRun2
  ID: tight
  Isolation: PF, tight
Electrons:
   https://twiki.cern.ch/twiki/bin/viewauth/CMS/
      EgammaRunIIRecommendations
   ID: tight, cut-based VID
      2016UL: 'Fall17-94X-V2'
      2017UL: 'Fall17-94X-V2'
      2018UL: 'Fall17-94X-V2'
   Isolation: part of ID

      ———————— Hotzone Maps ————————
See: https://twiki.cern.ch/twiki/bin/viewauth/CMS/
         JECDataMC#Jet_veto_maps
The jetvetomap is applied on the leading jet.
   2016UL: Summer19UL16_V0/hotjets-UL16.root
      ('h2hot_ul16_plus_hbm2_hbp12_qie11', 'h2hot_mc')
   2017UL: Summer19UL17_V2/hotjets-UL17_v2.root
      ('h2hot_ul17_plus_hep17_plus_hbpw89')
   2018UL: Summer19UL18_V1/hotjets-UL18.root
      ('h2hot_ul18_plus_hem1516_and_hbp2m1')

      ———————— Pileup ————————
See: https://twiki.cern.ch/twiki/bin/view/CMS/PileupJSONFileforData
sha256sum of pileup_latest.txt:
2016: e4106987f1379eec3e9579855b70119265ecacf4c9049f4b71fd51b06d97865c
2017: 3c3aa282ca4f664e89118440e346b492bb39e486cdc75263b02b5e391c3c4582
2018: cbe619012ba8d04e0bcd7b555c4977b6a59cd198cf5f0eaceea60be7ed410829
```

Table A.1.: PF Jet ID Criteria for AK4CHS Jets (UL16).

PF Jet ID	$\lvert\eta\rvert \leq 2.4$	$2.4 < \lvert\eta\rvert \leq 2.7$	$2.7 < \lvert\eta\rvert \leq 3.0$	$3.0 < \lvert\eta\rvert \leq 5.0$
Neutral Hadron Fraction	< 0.90	< 0.90	< 0.90	> 0.2
Neutral EM Fraction	< 0.90	< 0.99	> 0 & < 0.99	< 0.9
Number of Constituents	> 1	-	-	-
Muon Fraction	< 0.80*	-	-	-
Charged Hadron Fraction	> 0	-	-	-
Charged Multiplicity	> 0	-	-	-
Charged EM Fraction	< 0.80*	-	-	-
Number of Neutral Particles	-	-	> 1	> 10

**for LepVeto*

Table A.2.: PF Jet ID Criteria for AK4CHS Jets (UL17 and UL18).

PF Jet ID	$\lvert\eta\rvert \leq 2.6$	$2.6 < \lvert\eta\rvert \leq 2.7$	$2.7 < \lvert\eta\rvert \leq 3.0$	$3.0 < \lvert\eta\rvert \leq 5.0$
Neutral Hadron Fraction	< 0.90	< 0.90	-	> 0.2
Neutral EM Fraction	< 0.90	< 0.99	> 0.01 & < 0.99	< 0.9
Number of Constituents	> 1	-	-	-
Muon Fraction	< 0.80*	< 0.80*	-	-
Charged Hadron Fraction	> 0	-	-	-
Charged Multiplicity	> 0	> 0	-	-
Charged EM Fraction	< 0.80 *	< 0.80*	-	-
Number of Neutral Particles	-	-	> 1	> 10

**for LepVeto*

A.1.5. Muon and Electron Corrections

Listing A.4: List of muon and electron corrections utilized for the L3res corrections.

```
                    ———————— Muon Channel ————————
Rochester Corrections:
  https://twiki.cern.ch/twiki/bin/viewauth/CMS/RochcorMuon
  2016UL: Run2.v5
    APV: RoccoR2016aUL.txt
    sha256sum=
cc49740b1b89418e52c8baa6ca30d27a5021650775896feae67c2a384d8d0748
    nonAPV: RoccoR2016bUL.txt
    sha256sum=
53b6e1a2f511775520ff40f87a8ed4540ab9aa6cbfcfda7795be81bc7af54544
  2017UL: Run2.v5 RoccoR2017UL.txt
    sha256sum=
c8ceda9435226773a8a9a418e5158753b6f1df4be773d113d4207422f6cd93b3
  2018UL: Run2.v5 RoccoR2018UL.txt
    sha256sum=
e4b48ef6146b9df97f9f3ae7d11d887f44a45c37fe6824627b0609dea44cc86e
    ———————— Electron Channel ————————
EGamma Residual Scale and Smearing Corrections:
  https://twiki.cern.ch/twiki/bin/viewauth/CMS/
  EgammaRunIIRecommendations#Residual_Scale_Smearing_Correcti
  2016UL, 2017UL, 2018UL: ecalTrkEnergyPostCorr
```

A.1.6. Z Selection and Cut Flow

Z Selection:

- Z+Jet balanced topology selection:
  - $|\Delta\Phi(\text{leadingjet}, Z) - \pi| < 0.44$ (BackToBack)
  - $\alpha = p_{\mathrm{T}}^{\text{secondjet}} / p_{\mathrm{T}}^{Z} < 1.0$
- Muon Channel:
  - $|\eta^{\mu} < 2.3|$
  - $p_{\mathrm{T}}^{\mu} > 20 GeV$
  - $p_{\mathrm{T}}^{\text{sub}} > 10 GeV$
- Electron Channel:
  - $|\eta^{e} < 2.4|$
  - $p_{\mathrm{T}}^{e} > 25 GeV$
  - $p_{\mathrm{T}}^{\text{sub}} > 15 GeV$
- Z Selection (Both Channels):
  - $p_{\mathrm{T}}^{Z} > 15 GeV$
  - $|m_{\text{reconstructed}}^{Z} - m_{\text{PDG}}^{Z}| < 20 GeV$

Listing A.5: Cut flow for the L3res corrections.

```
                 ———————— Muon Channel ————————
JsonFilter: filter by golden json files
HltFilter: apply triggers
ValidJetsFilter
LeadingJetPtCut: see above
MinNMuonsCut: 2
MaxNMuonsCut: 3
MuonPtCut: see above
MuonEtaCut: see above
ValidZCut: mass window cut
METFilter: list see below
BackToBackCut: 0.44
JetIDCut: last updated 06.2021
                 ———————— Electron Channel ————————
JsonFilter: sort out by golden json files
HltFilter: apply triggers
ValidJetsFilter
LeadingJetPtCut: see above
MinElectronsCountFilter: 2
MaxElectronsCountFilter: 3
ElectronPtCut: see above
ElectronEtaCut: see above
ValidZCut: mass window cut
METFilter: list see below
BackToBackCut: 0.44
JetIDCut: last updated 06.2021
```

A.1.7. Further Results and Control Plots

Here, the results for the other years of Run 2 are provided. As in 2018UL (see Section 3.3), all distributions shown in gray are normalized to unit area, allowing for a direct comparison of their shapes independent of their absolute yields.

A.1.7.1. 2017UL

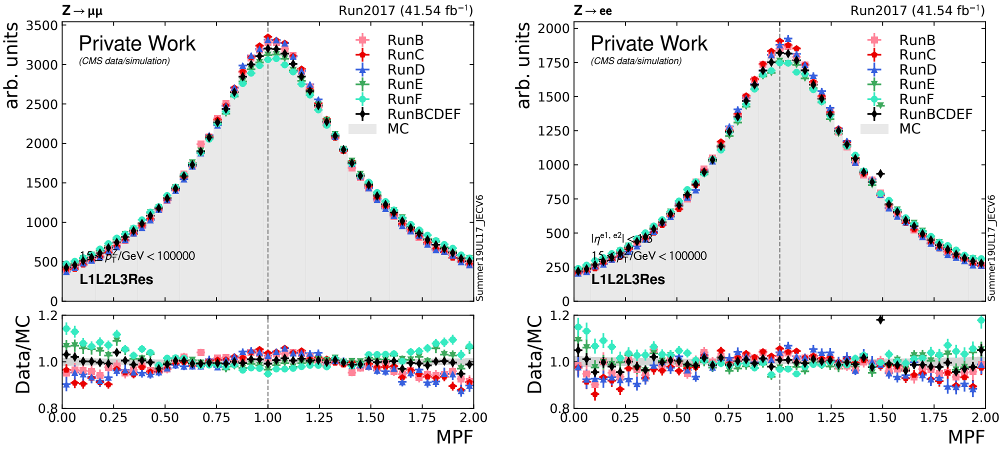


Figure A.1.: Comparison of the response derived with the MPF method for muons (left) and electrons (right).

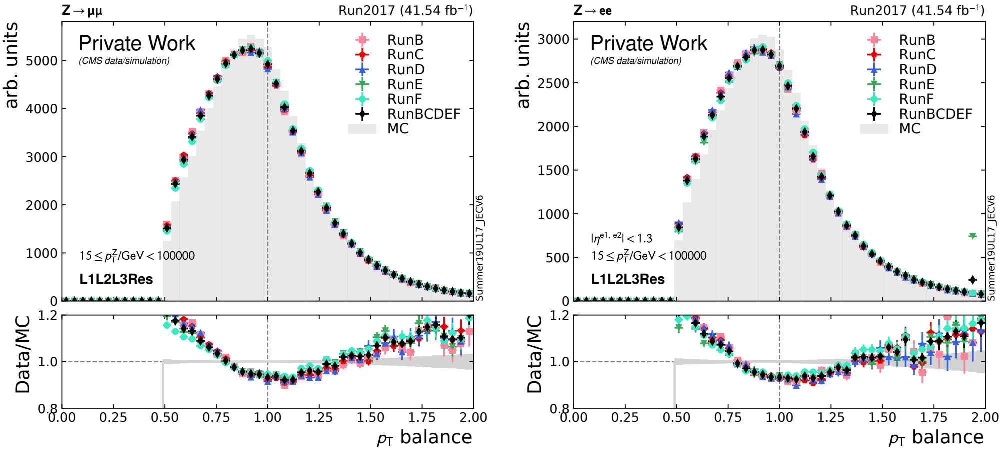


Figure A.2.: Comparison of the response derived with the direct p_{T} balance method for muons (left) and electrons (right).

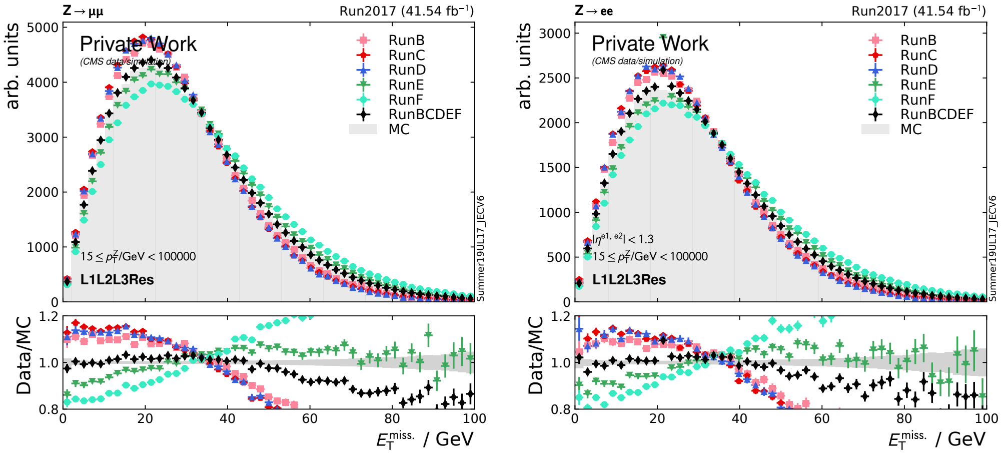


Figure A.3.: Missing transverse energy for the muon dataset (left) and the electron dataset (right).

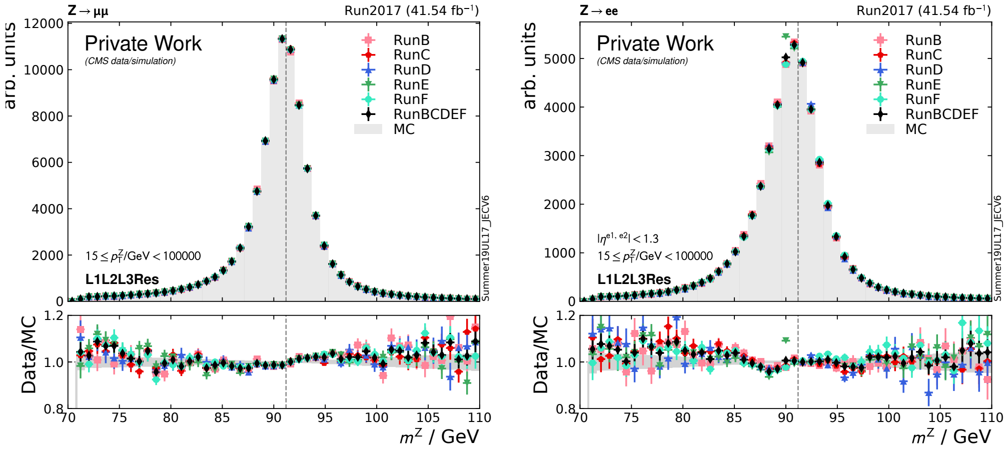


Figure A.4.: Z mass reconstructed from muons (left) and electrons (right).

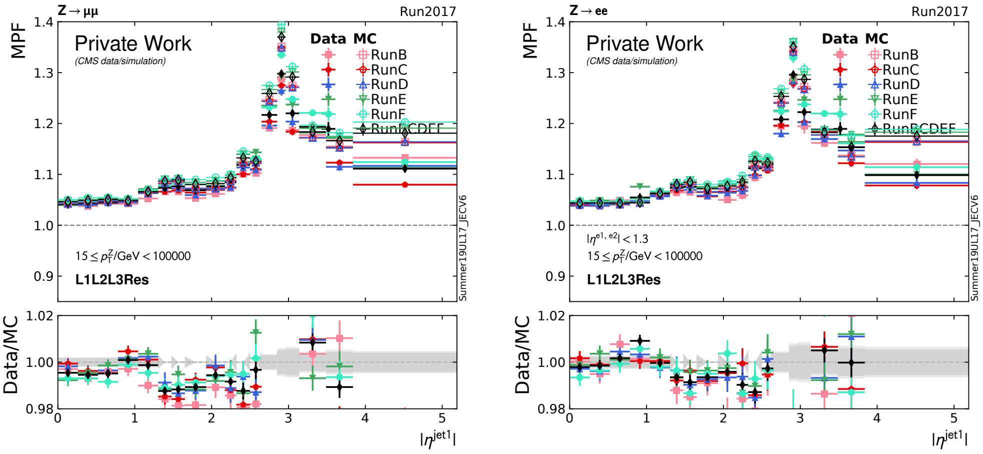


Figure A.5.: MPF for muons (left) and electrons (right) derived in dependence of the pseudorapidity of the leading jet. The agreement for the central detector region is for both channels remarkably good with the L3Res corrections in place.

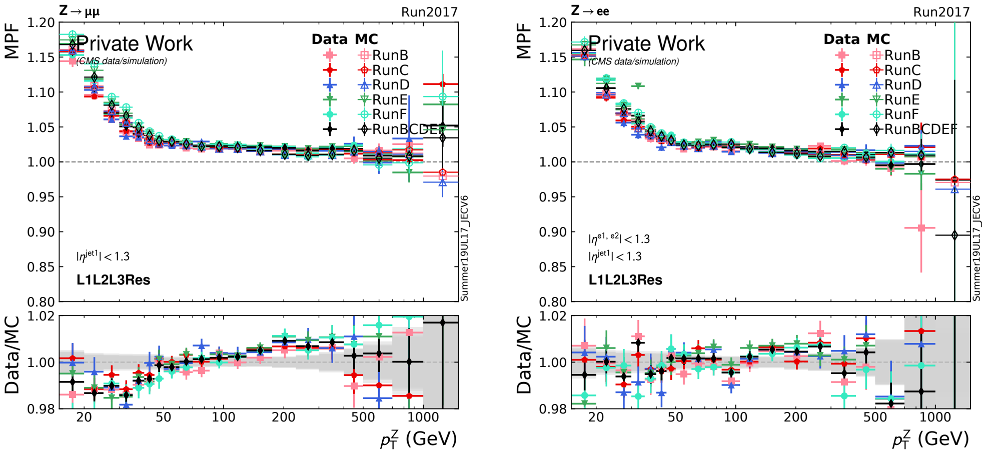


Figure A.6.: MPF for muons (left) and electrons (right) derived in dependence of the transverse momentum of the leading jet. Interestingly, the electron channel is even closer to the optimum than the muon channel.

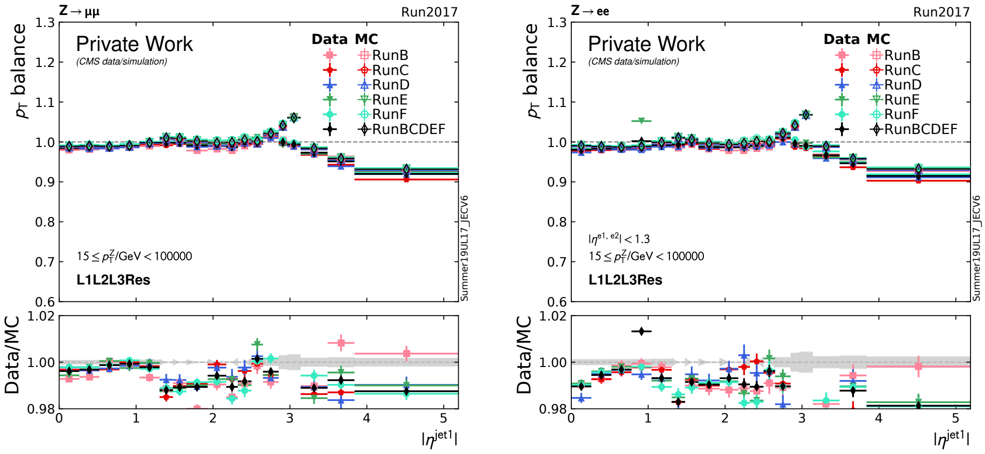


Figure A.7.: For 2017UL, the direct p_T balance method in dependence of η yields very comparable results to the MPF method in the muon channel. This indicates a very consistent calibration for this year.

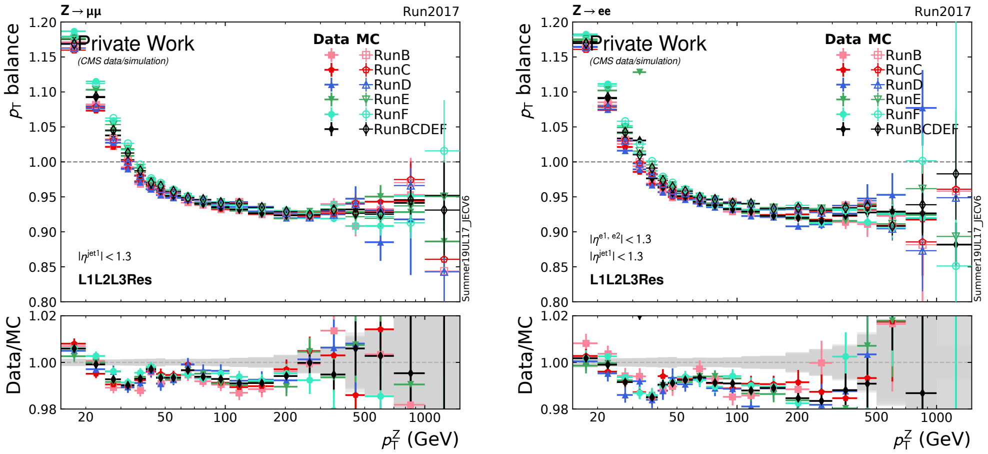


Figure A.8.: Just as for the derivation in dependence of η, the direct p_T balance method also yields very good results in dependence of p_T. For muons, the discrepancy between data and simulation is less than 1 % over the full p_T range.

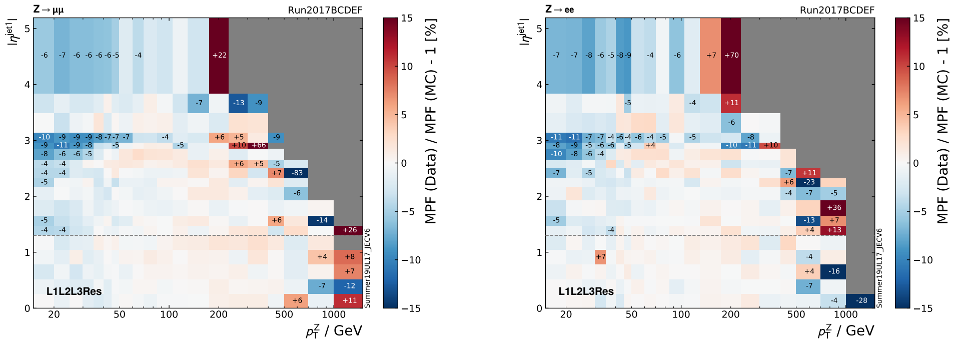


Figure A.9.: Double ratio for the MPF method. The result for both channels is comparably good.

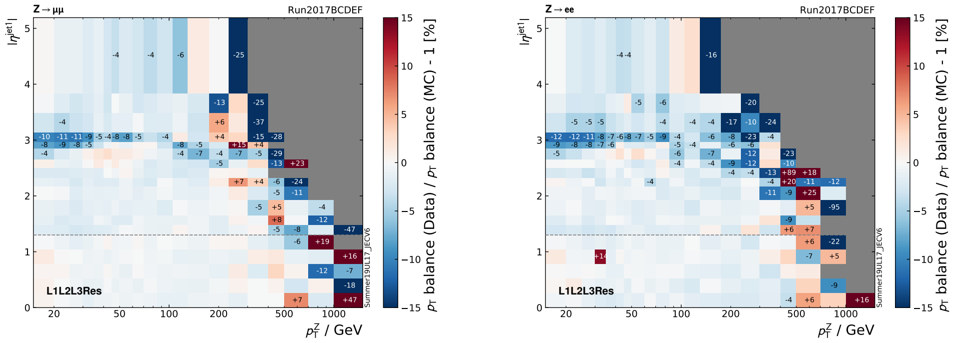


Figure A.10.: 2-dimensional response derived with the direct balance method. In comparison to Fig. A.25, this shows that the direct balance method is very comparable to the MPF method for 2017UL, which is a great cross-check and verifies the validity of the method. Especially in the low-p_T region, the DB method is even better here.

A.1.7.2. 2016UL

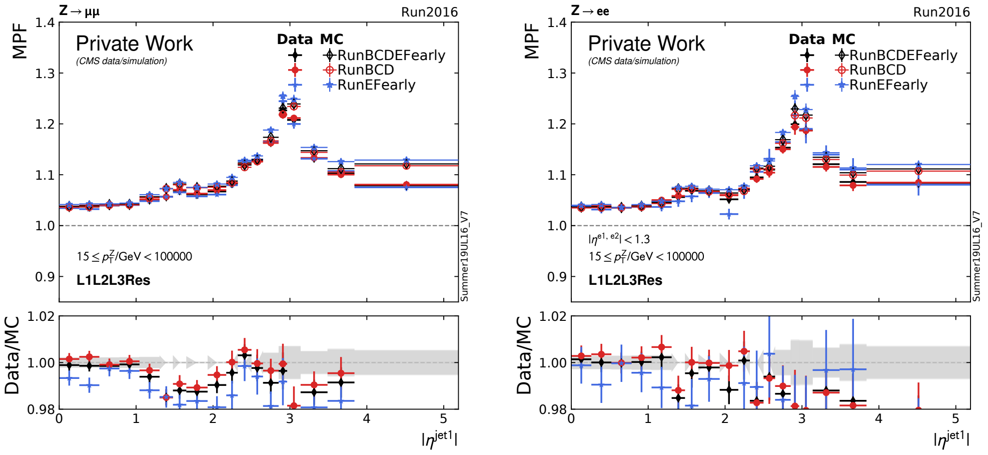


Figure A.11.: MPF for muons (left) and electrons (right) derived in dependence of the pseudorapidity of the leading jet. The agreement for the relevant region is for both channels nearly perfect with the L3Res corrections in place.

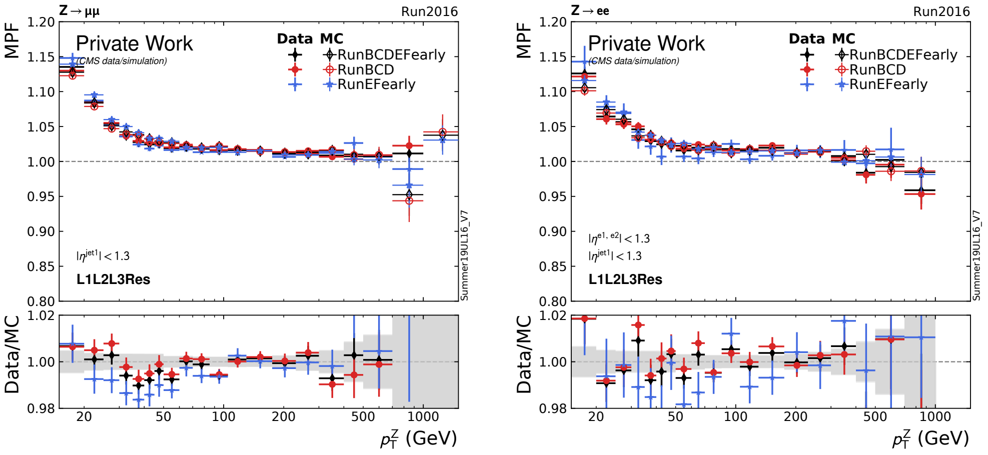


Figure A.12.: MPF for muons (left) and electrons (right) derived in dependence of the transverse momentum of the leading jet. The result for muons is decent. For electrons, the overall spread is larger.

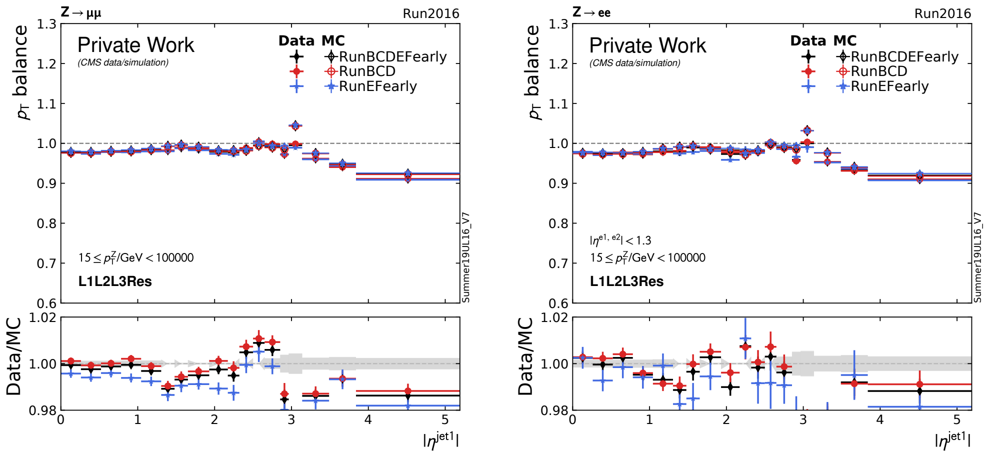


Figure A.13.: Also for 2016UL, the direct p_T balance method in dependence of η yields very comparable results to the MPF method in the muon channel. The electron channel is worsening in the range $0.8 < |\eta| < 1.7$.

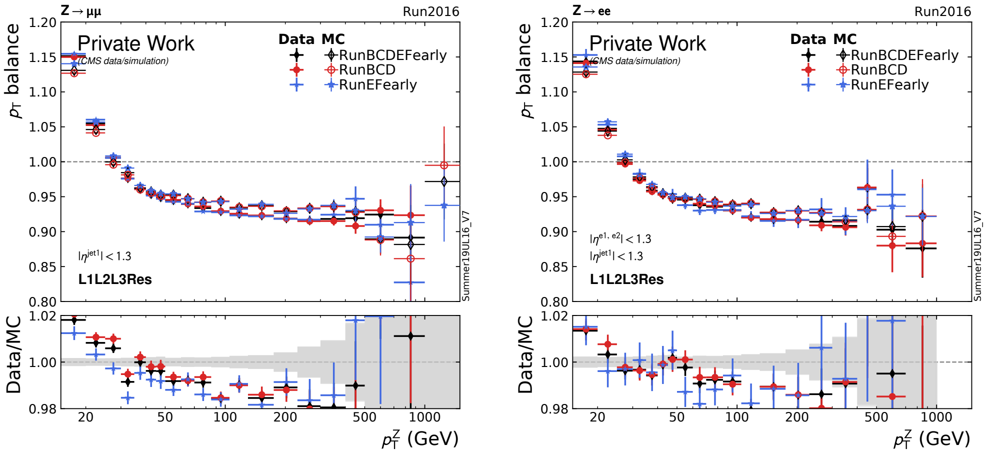


Figure A.14.: In contrast to 2017UL and 2018UL, the agreement is only in the range from 40 GeV to 80 GeV very good for both channels. In the other regions, it is lacking behind the MPF method.

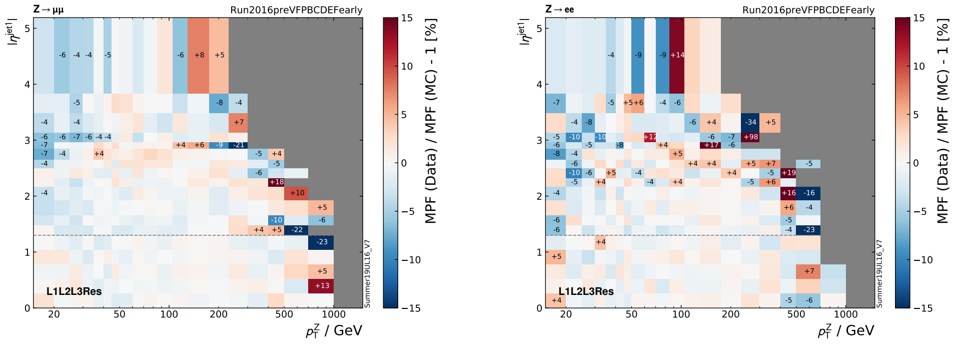


Figure A.15.: Double ratio for the MPF method. The result for both channels is very good in general. For low p_T values, the agreement in the muon channel is slightly better.

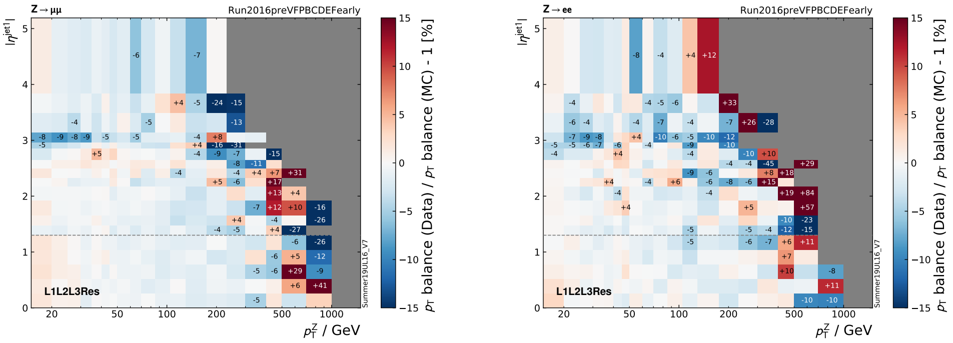


Figure A.16.: Double ratio derived with the direct balance method. In contrast to the other years, in this time period the response is slightly overestimated in the low-p_T region for both channels. The overall result is nevertheless very good in the relevant area.

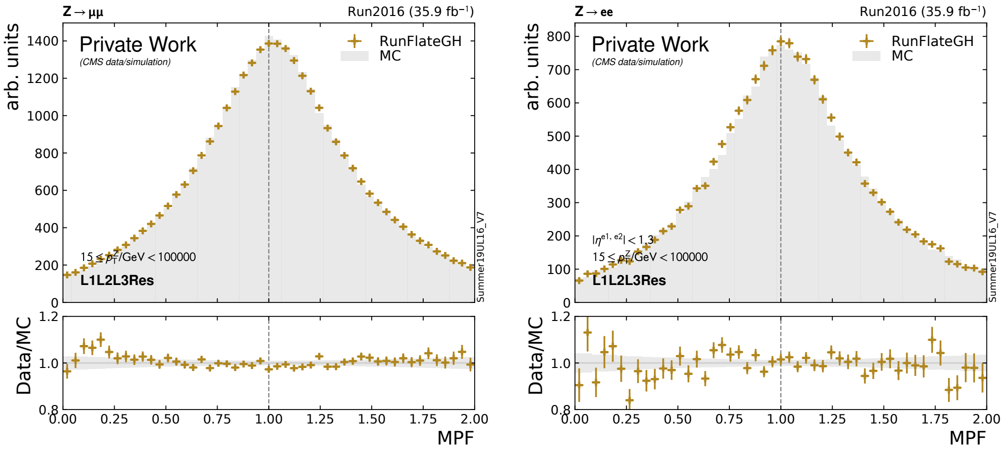


Figure A.17.: Comparison of the response derived with the MPF method for muons (left) and electrons (right). Especially the muon channel shows an extremely good agreement with the L3Res corrections applied.

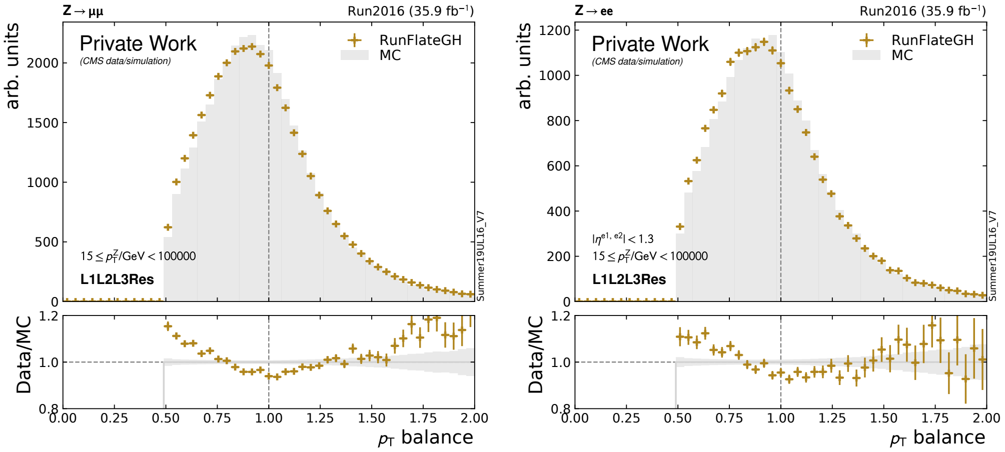


Figure A.18.: Comparison of the response derived with the direct p_{T} balance method for muons (left) and electrons (right). The discrepancies are significantly higher than for the MPF method.

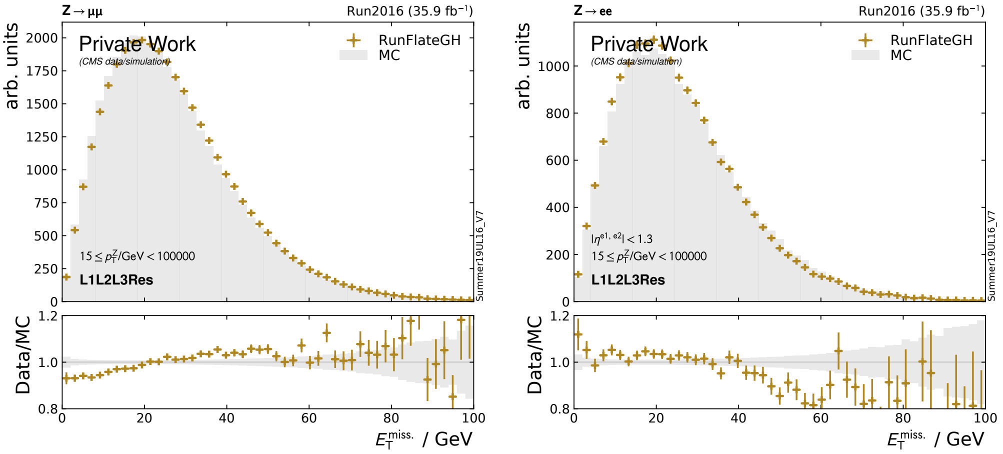


Figure A.19.: Missing transverse energy for the muon dataset (left) and the electron dataset (right). In the most relevant region – above 15 GeV – the missing transverse momentum is well modeled for both sub-sets.

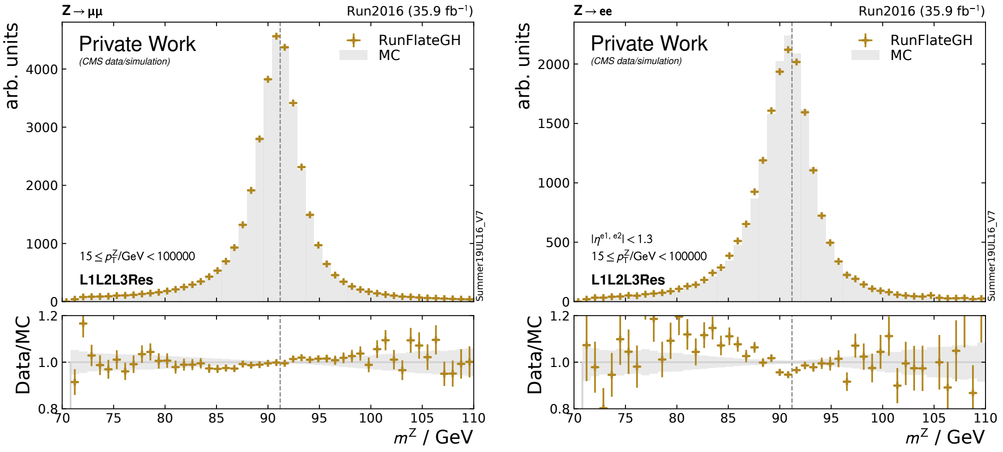


Figure A.20.: Z mass for 2016UL reconstructed from muons (left) and electrons (right). While the agreement for muons is overall good, in the electron channel significant derivations are observable, especially in the tails.

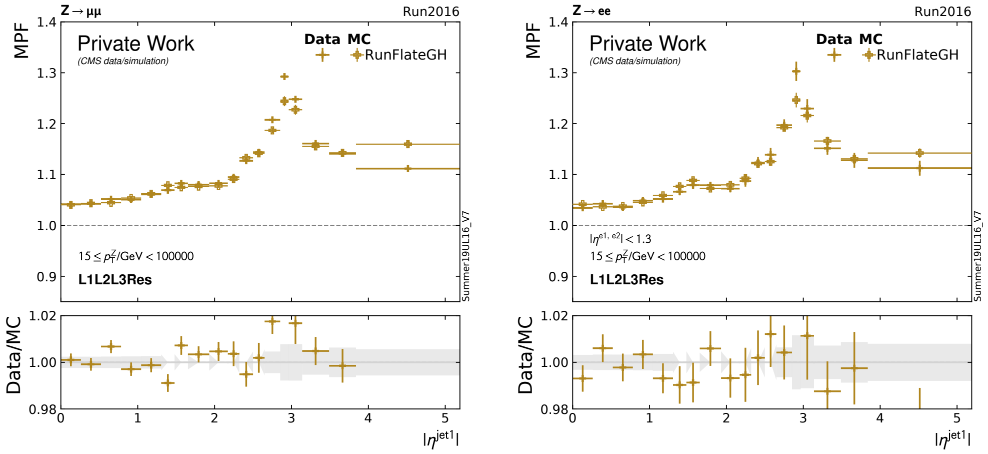


Figure A.21.: MPF for muons (left) and electrons (right) derived in dependence of the pseudorapidity of the leading jet. The agreement for the muon channel is very good up to $|\eta| = 2.5$. For the muon channel, the spread is slightly bigger but the agreement is still decent with the L3Res corrections in place.

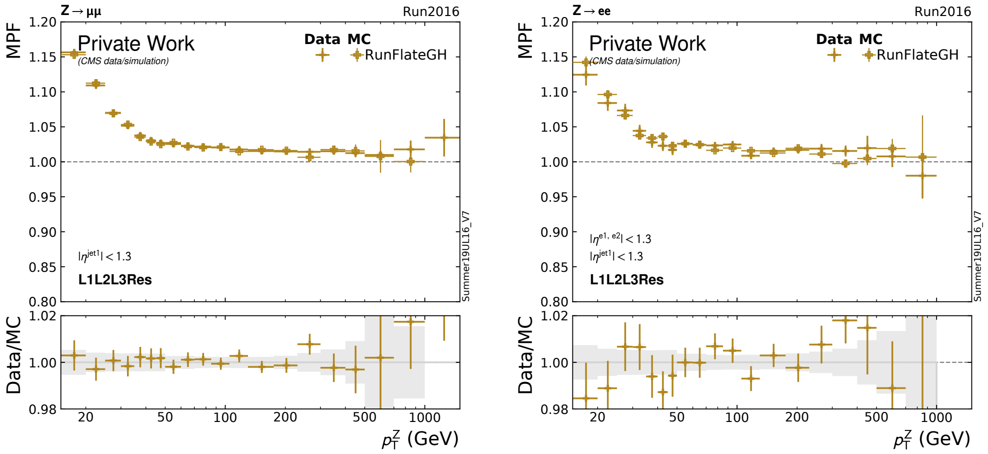


Figure A.22.: MPF for muons (left) and electrons (right) derived in dependence of the transverse momentum of the leading jet. The result for the muon channel is very close to being perfectly calibrated. For electrons, the result shows some discrepancies for $p_T < 50$ GeV and $p_T > 250$ GeV.

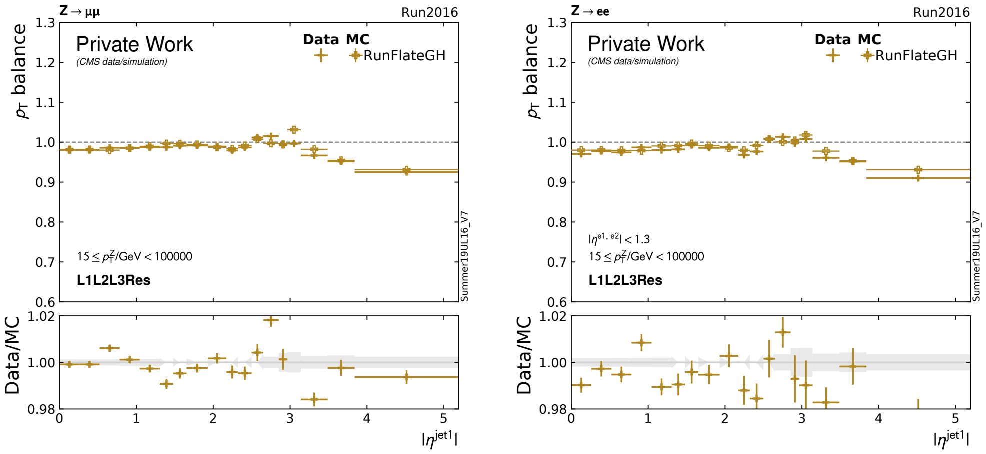


Figure A.23.: For 2016UL, the direct p_T balance method in dependence of η yields very good results for muons as well. Up to $|\eta| = 2.5$, the data/MC discrepancy is blow 1 %. For electrons, the agreement is slightly worse but still decent in the relevant region.

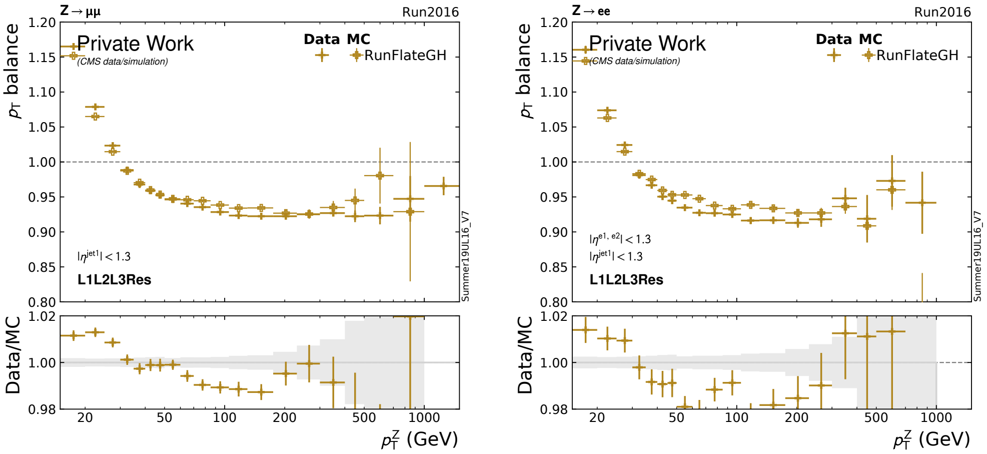


Figure A.24.: Evaluated in dependence of p_T, the direct balance method shows significantly worse results than the MPF method for both channels.

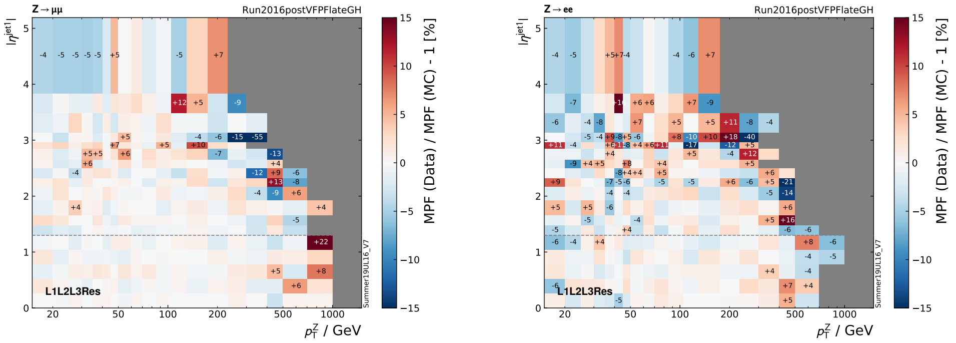


Figure A.25.: Double ratio for the MPF method. The MPF method yields overall very good results in the muon channel. In the electron channel, the discrepancies are higher in the low and high p_T range.

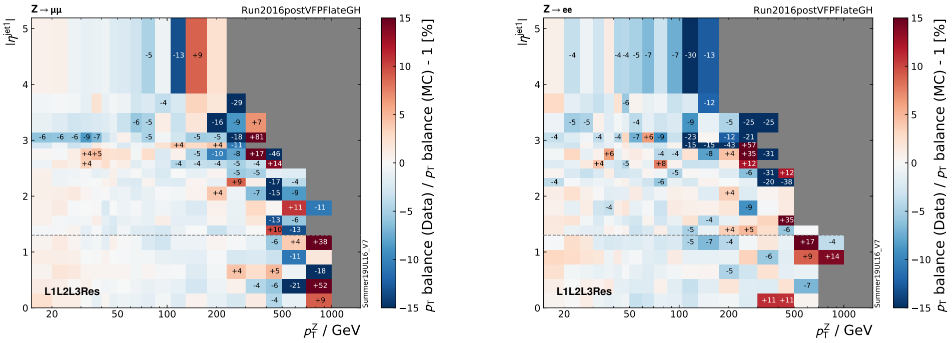


Figure A.26.: Double ratio derived with the direct balance method. Just as for the first half of 2016UL, the response is slightly overestimated in the low-p_{T} region for both channels when using the p_{T} balance method. The overall result is nevertheless decent.

A.2. Monitoring Data Preparation Details

A.2.1. Variable Selection

The HTCondor monitoring index for CMS is very verbose. Sometimes, this is great, as demonstrated in the next section (Section A.2.3). But it can become problematic when variables describe the same quantity, but are not consistent with their values. Relevant for this work are especially two of the most important quantities: `CoreHr*` and `CommittedCoreHr`.

As a first step for the data preparation and validation, the most reasonable and reliable one has to be chosen. For this, the distributions of the variables from jobs that ran at HoreKa in 2022 are shown in Fig. A.27. From this can be indicated that the `WallClockHr*`, and with it the `CoreHr*`, do not provide a solid representation of reality. Because a significant part of the events seems to have a runtime greater than the maximum lifetime of the pilots (20 h), which is impossible. As a consequence, the CPU efficiency is significantly degraded for those jobs that exceed the maximum runtime. The same happens also in the other direction, which is shown below. The reasons are not deducible.

In conclusion, the `CommittedWallClockHr*` directly reported from CMSSW is the more reasonable variable, as it is strictly capped at 20 h. The `WallClockHr*` should only be used, if the other one is not available, and only after a careful plausibility check!

The second problem that has to be discussed before the actual data preparation stands in direct relation with the first one: The complex software frameworks and monitoring systems cannot guarantee 100 % reliable data. This applies not only for the `WallClockHr*`, but for all quantities. Sometimes, hick-ups, failures, or bugs happen. If e.g. a grid job runs into a segmentation violation, it cannot be guaranteed that the reported data is still valid. As a consequence, it is very important to validate the used data and to identify unreasonable outliers. Because errors in the basic quantities can lead to significant deviations and alter the results of an evaluation, as shown with the following example.

In Fig. A.28, the CPU efficiency distribution of jobs exceeding the theoretical maximum of 800 % (assuming no additional boost of the CPUs) is presented for four different sites. This CPU efficiency is calculated with `CoreHr*` as the denominator, which is the default reported as `CpuEff*` in the CMS HTCondor monitoring. The selected jobs have an average CPU efficiency (see Eq. (5.3)) of 21 773 %, and all together a total efficiency (Eq. (5.2)) of 1421 %. The problem occurs at all four evaluated sites and is clearly a bug, since a CPU efficiency of up to 600 000 % is impossible. This is caused by a faulty[1] value of `CoreHr*` that is too low, distorting the `CpuEff*` reported by HTCondor. Why this happens, however, is unclear. But it is definitely not a site-related problem, since it occurs on 4 sites over 3 years in total.

To resolve this problem, the CPU efficiency should be calculated with the `CommittedCoreHr*` as the denominator – in line with the findings from above. The result is presented in Fig. A.30. While still some jobs are faulty, the majority is fixed and the distribution looks ways more reasonable. The jobs that are still faulty are all *Analysis** test jobs and will be sorted out anyway (see *Job Type Selection* below).

[1] Additionally, very often the same low value can be found which also indicates a bug or problems with reporting.

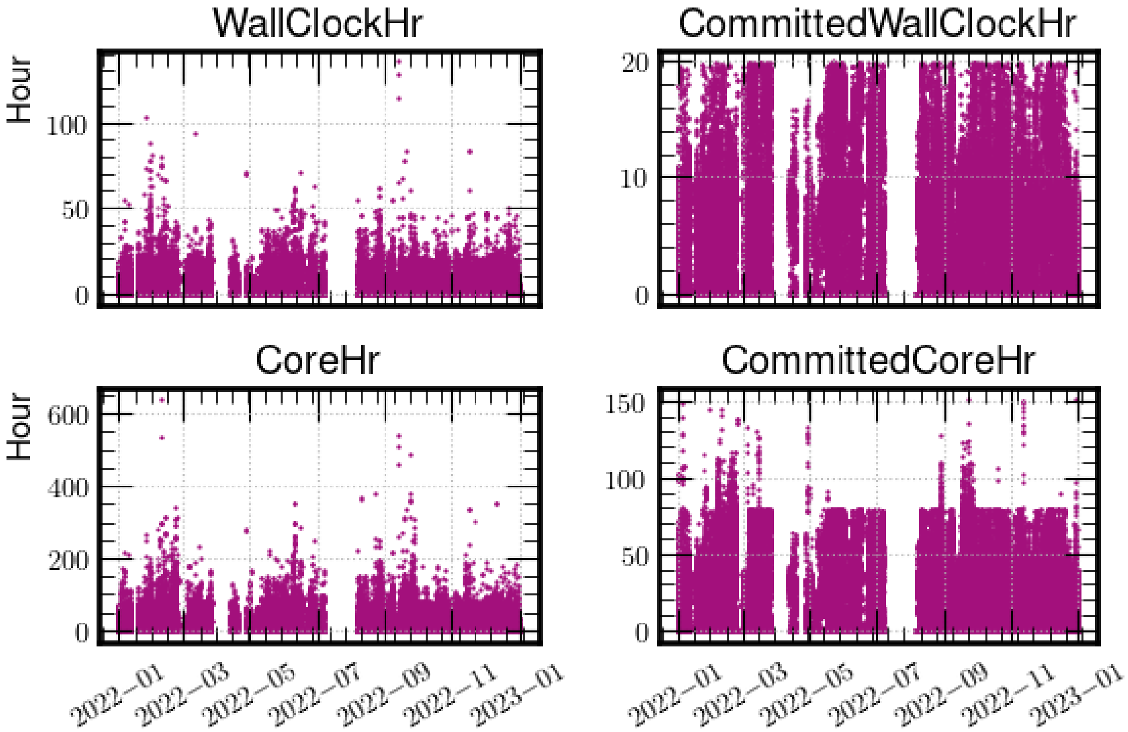


Figure A.27.: In the HTCondor payload monitoring of CMS, some variables are available in different versions. The most relevant one is the runtime of a grid job, present in the monitoring data as `WallClockHr*` and `CommittedWallClockHr*`. Often, they are the same. But sometimes, they are completely inconsistent. Since the `WallClockHr*` regularly exceeds the maximum pilot runtime of 20 hours, as visible in the top left figure, the other quantity should be preferred.

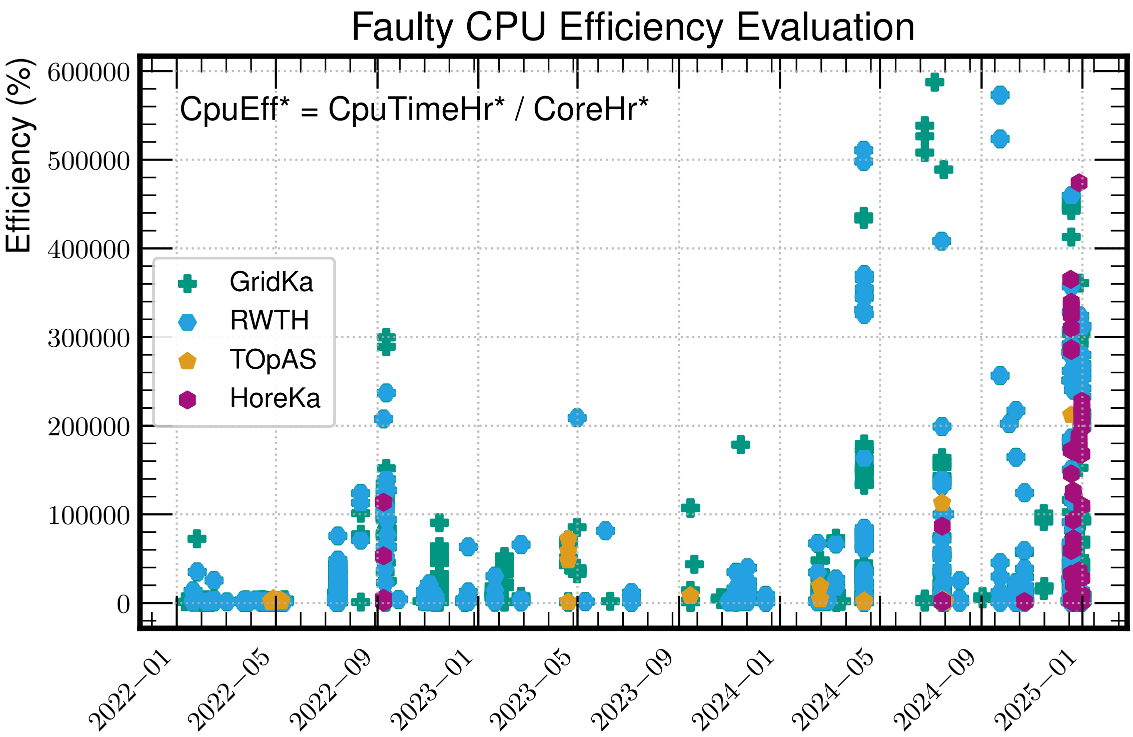


Figure A.28.: Efficiency distribution of all faulty jobs with a CPU efficiency above the theoretical maximum of 800 % (single-core job over-booked to a full 8-core pilot). The problem occurs at all evaluated sites: HoreKa, TOpAS, GridKa, and RWTH, the Tier-2 at university of Aachen. Such extreme CPU efficiencies are clearly an error.

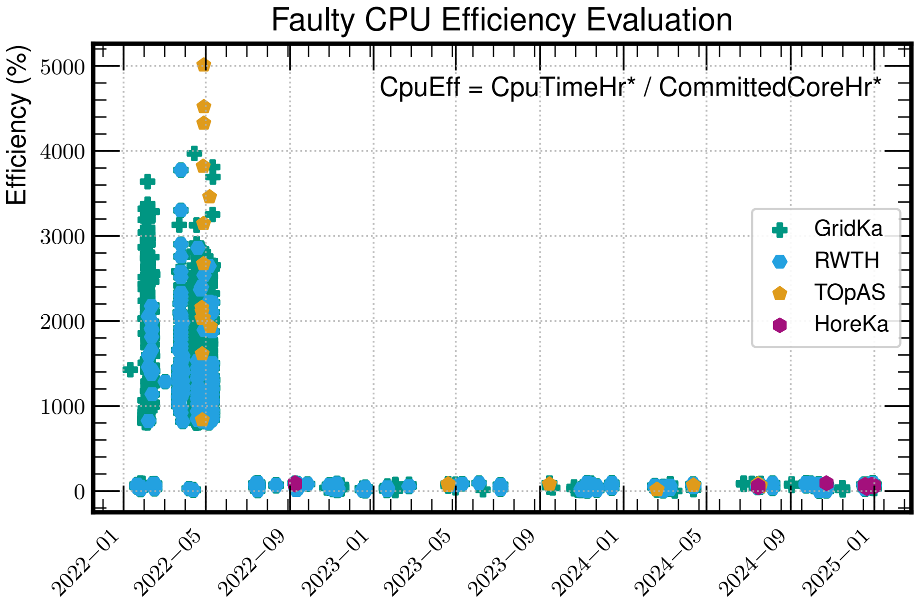


Figure A.29.: This figure shows the full distribution of the same faulty jobs as Fig. A.28, but with the CPU efficiency calculated with the `CommittedCoreHr*` instead of `CoreHr*`. The remaining faulty jobs are all *Analysis** test jobs (not to be confused with real *Analysis** jobs!) that are anyway not representative and can be safely neglected for the evaluation of a grid site.

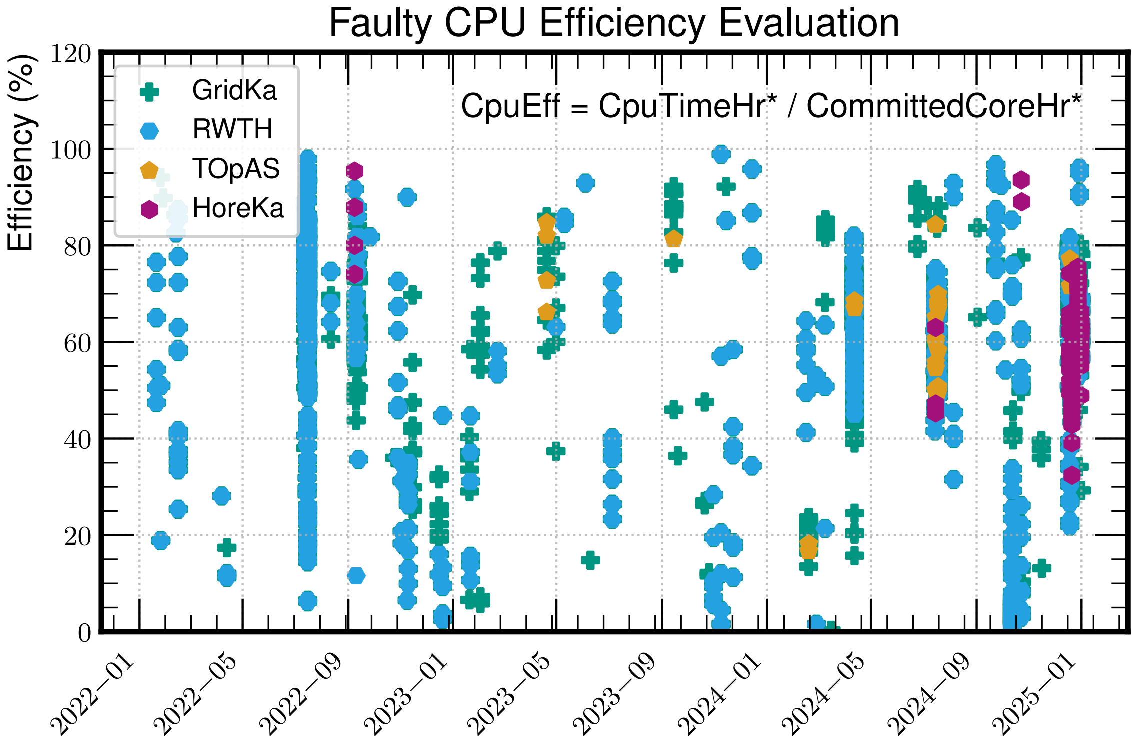


Figure A.30.: Efficiency distribution of the same faulty jobs but with the CPU efficiency calculated with the `CommittedCoreHr*` instead of `CoreHr*`. This calculation clearly provides more reasonable results and shows that a proper data selection is crucial for a reliable evaluation based on the payload monitoring data.
Remark: 1116 jobs are still faulty, however, the issue is not as pronounced. This happens, where the calculation yields the same result as both reported values for the core hours are equal. Those jobs are all *Analysis** test jobs that will be neglected anyway for the further analyses (see the paragraph *Job Type Selection* below). The full distribution is given in Fig. A.29.

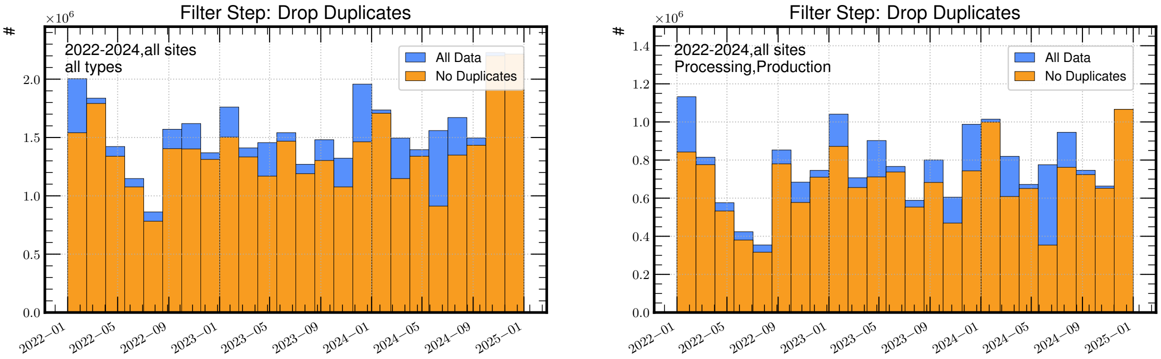


Figure A.31.: The figures show the extent of duplicates in the monitoring data from 2022 to 2024. In blue, the total number of jobs per time bin is depicted. Orange excludes the duplicates. On the left, all job types are shown together for all investigated sites: HoreKa, TOpAS, GridKa (T1_DE_KIT), and Aachen (T2_DE_RWTH). Right, the same is shown for the most relevant types only: *Production** and *Processing**. Other job types are separately shown in Fig. A.33. In general, no real systematic can be found from the basic investigation.

The lessons learned by the examples are firstly, that the payload monitoring data for CMS jobs is not always totally reliable and requires a careful evaluation. Secondly, the 'Committed' quantities always seem to be more reliable and less error prone. The CPU efficiency and other evaluation measures should therefore be calculated according to Eq. (5.1), using `CommittedCoreHr*`, instead of `CoreHr*`, which seems to yield more stable and reasonable results. And lastly, even if the total impact of the examples is rather small, this shows, how important a proper data validation and preparation is for a realistic and consistent evaluation.

A.2.2. Analysis of Duplicates

The monitoring data from the HTCondor index can contain duplicates, which are identical data points in all variables. They do not exist for every entry and there are no obvious systematics, as shown in Fig. A.31. This is no site-specific issue, as demonstrated with the according comparison for HoreKa only, presented in Fig. A.32. A comparison between the two depicted distributions per figure, each for all job types (left), and *Processing** and *Production** jobs only (right), indicates that the effect is also not directly coupled to the job types. All others are shown in Fig. A.33 for HoreKa (left) and all sites accumulated (right).

One possible explanation could be the way the monitoring data is accessed. Maybe the querying process of the data with `pyspark` from HDFS, which may does not handle the structured data from the monitoring index entirely correct, causes the duplicates. Alternatively, it has to be attributed to inadequacies of the monitoring infrastructure. A more in-depth evaluation of the reasons, however, within the complex monitoring systems of the WLCG production environment is beyond the scope of this thesis.

Fortunately, a correction of the issue is fairly simple after it was identified. To correct the dataset for the site evaluation, it is safe and sufficient to simply keep only one data point and exclude all duplicates, since they are entirely identical. This can be

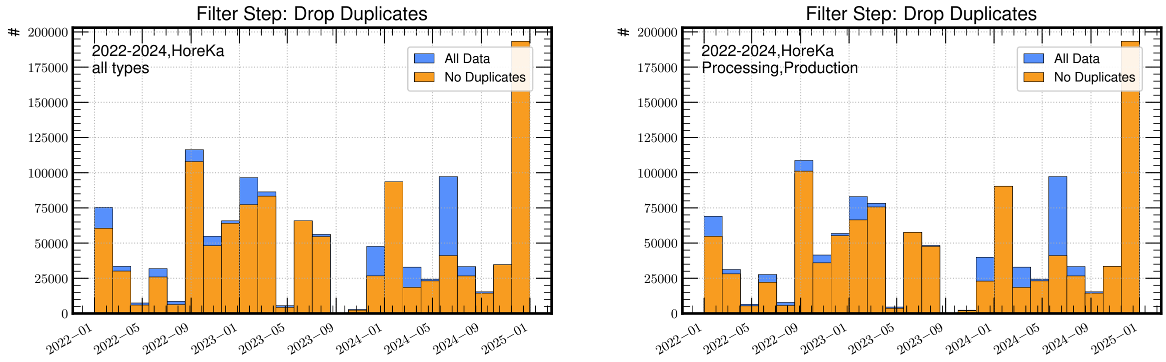


Figure A.32.: The figures show the histograms of jobs that ran at HoreKa from 2022 to 2024. Analog to Fig. A.31, all job types are presented left, and the most representative ones for a site evaluation on the right. In general, again no systematics are visible and the distributions do not allow to draw conclusions on the reasons.

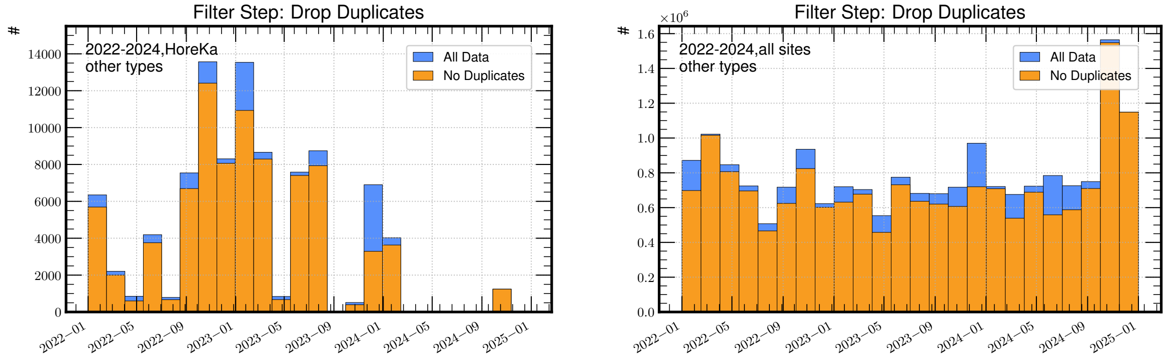


Figure A.33.: Here, the distributions of job types other than *Production** and *Processing** at HoreKa (left), and all sites together (right), is presented. This underlines that the effect is neither job type dependent, nor restricted to a certain site. What is noticeable is that there are significantly fewer of them. However, the issue still occurs.

done by filtering them based on the `GlobalJobId*` that has to be unique.

A.2.3. Data Correction, Cleaning, and Filtering

With the following data correction, cleaning, and filtering, it is ensured that all quantities that are relevant to evaluate the measures introduced in Section 5.5.1 can be derived from the monitoring data for HoreKa. The list of the most important quantities that has to be validated is therefore oriented on the relevance for the further analyses of a resource and includes: `CoreHr`*, `CommittedCoreHr`*, `WallClockHr`*, `CommittedWallClockHr`*, `CpuTimeHr`*, and `ChirpCMSSWTotalCPU`*. Furthermore, a `CompletionDate`* must exist and `WMAgent_SubTaskName`*, the name of the sub-task of a workflow the job is part of, must be set to allow a matching of common workflows at different sites later on. Lastly, `KEvents` should be none-zero when a *Production** job is successful, since it is rather unlikely that a MC simulation does not yield any output when functioning correctly.

At first, events are identified, where `ChirpCMSSWTotalCPU`*, the total consumed CPU time in seconds reported from CMSSW, is not set and the `CpuTimeHr`* is zero, too. This can happen, when a job fails and the data is not correctly reported. For HoreKa, around 0.25 % of all jobs are affected that need to be filtered out. Because such a case is not recoverable, as no data on the consumed CPU time of the jobs is available. The same applies, if `CommittedCoreHr`* and `CoreHr` are zero. Then, the monitoring data is inconclusive about the runtimes. The consequence is that the affected jobs are unusable for most evaluations, no matter if failed or successful. In principle, they can still be considered for the failure rate, but including them increases it by only 0.2 %. Since it anyway does not correctly reflect the impact and the total `CommittedCoreHr` of these jobs sum up to less than 1000 core hours, it is justified to simply disregard them.

If only the `CpuTimeHr`* is not set, it can be calculated from the value reported from CMSSW (`ChirpCMSSWTotalCPU`*) as the next step. For the HoreKa data from 2022 to 2024, this this is a minor issue. Nevertheless, to be as comparable as possible, the events can be recovered with this.

The next quantities to be validated are the `CommittedWallClockHr`* and the `WallClockHr`* which where already addressed in the example above. As discussed, the `CommittedWallClockHr`* is always to be preferred. Only if it is zero or unreasonable[2], but the `WallClockHr`* is not, the event can be recovered by using it instead as a replacement. This should be again fine in first order approximation, as as long as the time is realistic. The same applies for the `CommittedCoreHr`* and `CoreHr`, as they simply reflect the former two quantities multiplied by the number of `RequestCpus`*. These overlaps can be utilized for recovering faulty jobs as well, but only if the values are reasonable. If all the quantities are zero, but the job was successful, something is definitely wrong and it has to be filtered out.

When recovering jobs, it is important to validate that no unrealistic runtimes were taken over from the `WallClockHr`*. But also in general, it is important to make sure that only reasonable events are considered. For this, two additional boundaries for the `CommittedWallClockHr`* can be introduced. At first, an upper limit for a realistic runtime needs to be defined. This limit, of course, depends on the site, its integration, and the configuration. For HoreKa, the maximum wall time of a job is ultimately limited by the runtime of the COBalD/TARDIS drones (20 h). It is impossible that a job can run longer and can be therefore be considered as a

[2] If e.g. the `CommittedCoreHr`* is significantly shorter than the `CpuTimeHr`*.

reasonable upper boundary.

A reasonable lower limit is more difficult to find. Because depending on the job type, short runtimes can still be correct. For this work, it was decided that 3 min is reasonable as a lower limit for a successful *Production** job. Correctness can no longer be guaranteed below this level. The same applies for such jobs, which were successful, but `KEvents*` is zero. They also should be reviewed and maybe filtered, since a successful *Production** job should always yield an outcome.

For failed jobs, those two filters are not applied, because they can in principle also be valid when failing faster and do not provide an output.

Finally, the `CompletionDate*` and `WMAgent_SubTaskName*` both should be set, since they are required for more detailed (comparison) analyses and a time-dependent evaluation of the site performance and efficiency. Especially the first point is important, because over time, things can obviously change. To evaluate and reflect this, a time resolution of the quantities is crucial.

The entire employed correction chain is provided in Table A.3.

In summary, the available payload monitoring data for CMS is mostly reliable, but not perfect – as expected in such a complex environment. Therefore, a consistent data preparation, selection, and validation is crucial for comparable results. Best comparability can be achieved when carefully filtering erroneous, unreasonable, or duplicated data and correct faulty reports, if possible, to avoid a deterioration of the results.

Table A.3.: Filtering and correction chain for HoreKa. The first column describes the conditions with which the data is masked or corrected. Job type filtering removes 68.6 % of all jobs at HoreKa from 2022 to 2024. They contribute around 17 % of the total committed compute time, but are not representative for the performance of a grid site. Of the whole dataset, around 13 % are duplicates. The total process filters out around 72 % of the entire monitoring dataset to provide a representative basis for the evaluation of a grid site.

Correction step	Description
– Filtering, if: –	
multiple identical entries of a data point exist	Drop duplicates
Job type selection: CMS_JobType != Processing* OR Production*	Filter out unwanted job types
(ChirpCMSSWTotalCPU* == 0) AND (CpuTimeHr* == 0)	No CPU time info
(CoreHr* == 0) & (CommittedCoreHr* == 0)	No runtime info
(CommittedWallClockHr* < 3/60) OR (CommittedWallClockHr* > 20)	Unreasonable runtimes
Site* == selected site	Select site (e.g. T1_DE_KIT or T2_DE_RWTH)
MachineAttrCMSSubSiteName0* == selected sub-site	e.g. KIT-HOREKA, KIT-T3, or RWTH-HPC
– Corrections–	
(ChirpCMSSWTotalCPU* ! = 0) AND (CpuTimeHr* == 0)	Derive CPU time from CMSSW
(CommittedCoreHr* == 0) AND (CoreHr* ! = 0)	Derive from CoreHr*, if reasonable
(CommittedCoreHr* < CpuTimeHr*) AND (CoreHr* is reasonable)	Replace faulty CommittedCoreHr* by CoreHr*
– Optional –	
(CompletionDate* > X) AND (CompletionDate* < Y)	Restriction to time interval [X,Y]
WMAgent_SubTaskName* == selection	Selection of certain workflows
CPUModel == selection	Restriction to certain CPU models
...	...

A.3. Data-Driven Performance Value Estimation

Here, a new, data-driven method is presented that allows to estimate the performance potential of a grid resource without direct machine access. While official sites typically use HS23 benchmarks for this, those scores are not always available. E.g. for opportunistic resources, there is nothing like CRIC, where one can find the performance potential of a given resource. Therefore, a direct performance comparison is complicated to impossible.

To overcome this problem, a method was developed that solely uses the available payload monitoring data from CMS to estimate the relative performance of a grid resource in comparison to a known one. As described in Eq. (5.28), the idea of the method is to derive a ratio of the performance factors from a comparison of the direct processing efficiencies (Eq. (5.4)) for different sites. This is possible under the assumption that the same job from the same workflow on the same hardware should yield an identical throughput under the condition that the job was successful and running fully efficient. Because in this case, the p ratio should realistically reflect the difference in the performance of the two sites.

Only comparing one job, however, is not sufficient to obtain reliable results. Two grid jobs from the same workflow, by design, do roughly the same work within a similar runtime. But sometimes, on the one hand, not all simulations are similar complex, and on the other hand, different remote data may be accessed, leading to unavoidable differences. As a consequence, it will never be possible to achieve perfect results with this method and certain compromises must always be made in terms of the accuracy of the result. But this must be accepted, as there is no other option solely based on the available monitoring. A first step to stabilize the result, is to use all jobs of a workflow that meet the previously described requirements to calculate the ratio. And second, to refine the results even more, all workflows fulfilling the conditions can be included in the evaluation, instead of just one. Based on sufficient statistics, the averaging effect then ensures good and reliable results, as shown later.

The full method for deriving the performance factor is now exemplary presented for HoreKa. As described, a well-known baseline is required to achieve conclusive results. Several other sites are available for comparing the pure performance, from which TOpAS is chosen. The opportunistic Tier-3 at KIT is a good choice because, even if the circumstances are not identical, both resources are integrated into the WLCG in the same way with COBalD/TARDIS within the same OBS. This leads to many overlapping workflows that are distributed between the two resources. Additionally, TOpAS is a very reliable site with a high efficiency and low failure rate, which is also beneficial for the evaluation. As an R&D resource for GridKa, it is furthermore benchmarked with the official toolset (HS23 score: 13.5 [185]).

As a first step, the data must now be carefully selected. Since only similar jobs are conclusive, the monitoring data must be filter for workflows that ran on both sites. The second condition is that those workflows ran highly efficient. Each and every job that is considered for the evaluation therefore is required to have a CPU efficiency of at least 98.5 %. In addition, outliers are filtered out to make the results more stable. For this, the most comparable jobs are selected by restricting the interval to the range with the most overlap. It is derived by finding the jobs of a certain workflow with the minimal and maximal CPU efficiency for each site. Then, the greater minimum and the smaller maximum are selected. Jobs that are finally considered for the calculation must lie within the according CPU efficiency interval extended by 10 % in both directions: [minimum-10% , maximum+10%]. This sets the area under consideration to the maximum match. Additionally, only workflows with at least five jobs on each site that fulfill the demanded requirements are selected, which is necessary to avoid (statistical) outliers. Extreme and unreasonable outliers must be disregarded manually.

The second step is to calculate the aggregated processing efficiency (Eq. (5.7)), implicitly including the performance factor p, for each considered workflow and site. For the evaluation, all data from 2022 to 2024 is considered to get the biggest possible statistics. In principle, the performance factor of course can (and does) change over time, e.g. when new hardware is commissioned. For HoreKa, however, the hardware did not change since the commissioning, which is typical for HPC resources. This is simplifying the estimation. Concerning the other resources, a more granular timely resolution could be beneficial for more accurate results. However, the introduction and availability of new worker nodes is typically not linked to an annual rhythm, but generally does not follow any precisely defined cycles. Instead, hardware is procured when funding is available and the market situation permits it. The simple limitation to each individual year is therefore reducing the statistics in the first place without an expected greater benefit. Nevertheless, it is a useful consistency check for the newly introduced method, when the years are in approximate agreement. Furthermore, for sites where the hardware significantly changed, this can be considered by using the individual results for the according time periods, if the statistic is sufficient.

In Fig. A.34 and Fig. A.35, the results for four different workflows are shown that met the requirements and were executed at HoreKa (hexagons) and TOpAS (x markers). The figures show the processing efficiency and the CPU efficiency as a control plot, each. The derived p ratio results from the calculation of the individual aggregated processing efficiencies, as described in Eq. (5.28).

A good indicator that the selected workflows in Fig. A.34 are meaningful is that the processing efficiency is very constant per CPU model. This supports the idea of the method. When the processing efficiency is not constant – within a reasonable tolerance – a discarding of the workflow should be considered. Because this indicates that the primary assumptions for the method are not fulfilled. Examples are given and explained in Fig. A.36. Here, additional effects that are not correctly taken into account seem to influence the processing and change the results.

In addition, it is important that both distributions are consistent in the performance ratios of individual CPU models. This is not always the case, as shown in the bottom part of Fig. A.36. Here, the oldest CPU model seems to have the highest performance, which is very unlikely. Such a workflow should therefore be manually excluded in order to mitigate the effects of inappropriate values on the result. The reasons for such cases can be manifold and usually, an investigation only based on the payload monitoring data is not sufficient to reveal them, as it e.g. does not take interference effects between jobs running on the same nodes into account.

From Fig. A.35, two more conclusions can be drawn. The top figure shows that the results can even be comparable for workflows with low statistics. They should be treated with caution, but can still be beneficial for the final result as they contribute to an averaging effect making the results more robust. The lower part of the figure shows that the monitoring data is not always complete. Some jobs cannot be assigned

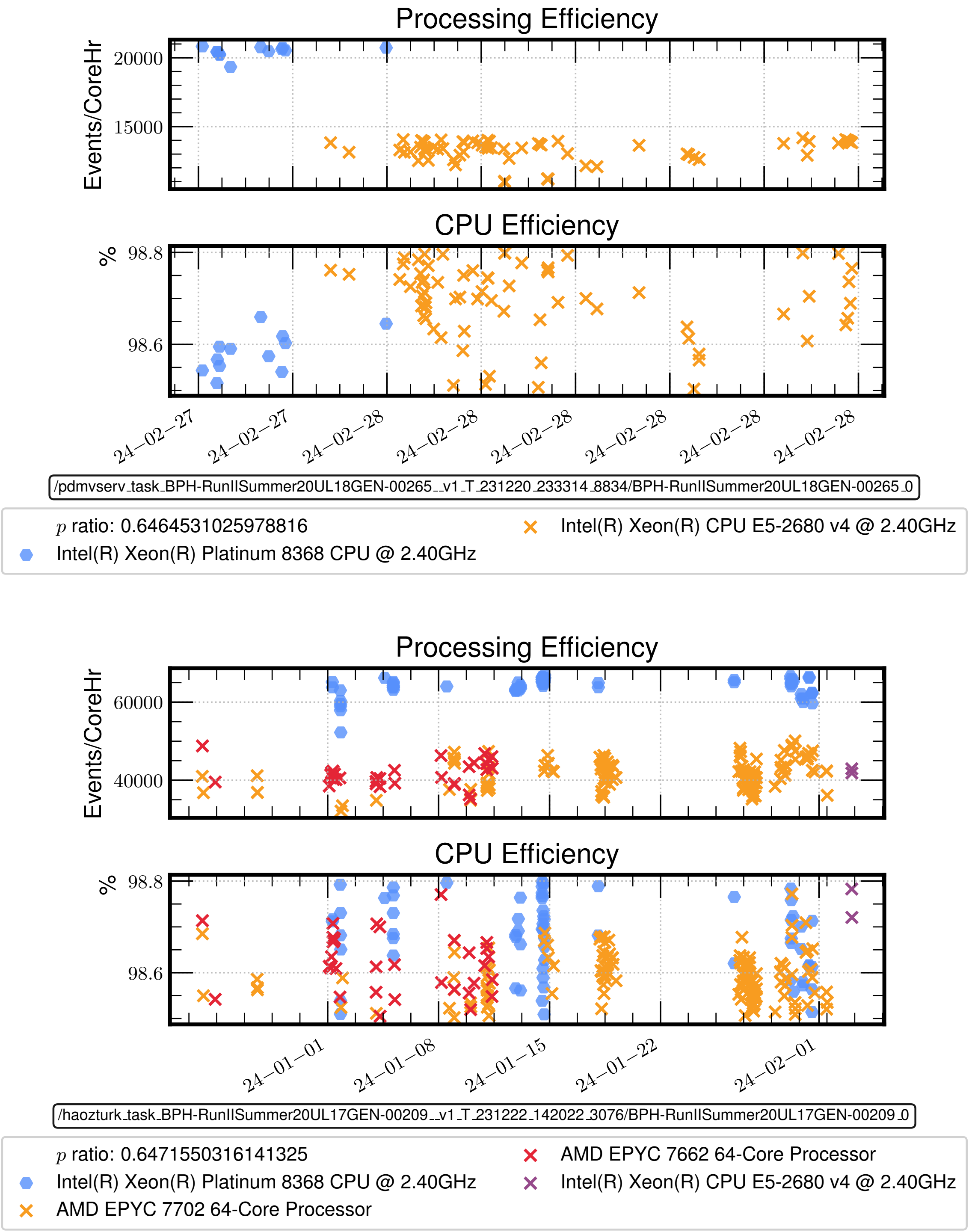


Figure A.34.: The figures show the estimation of the p factor ratios between TOpAS (x marker) and HoreKa (hexagons) for two different workflows based on the processing efficiency. Evaluation is limited to a very narrow CPU efficiency range near to full efficiency to achieve most comparability. Both workflows show that HoreKa has an enormous performance potential, between 30 % and 50 % higher than TOpAS. Since HoreKa employs HPC CPUs with a clear focus on performance, this is quite logical. Within each workflow, it is clearly visible that the processing efficiency per CPU model is nearly constant, which supports the principle of the method.

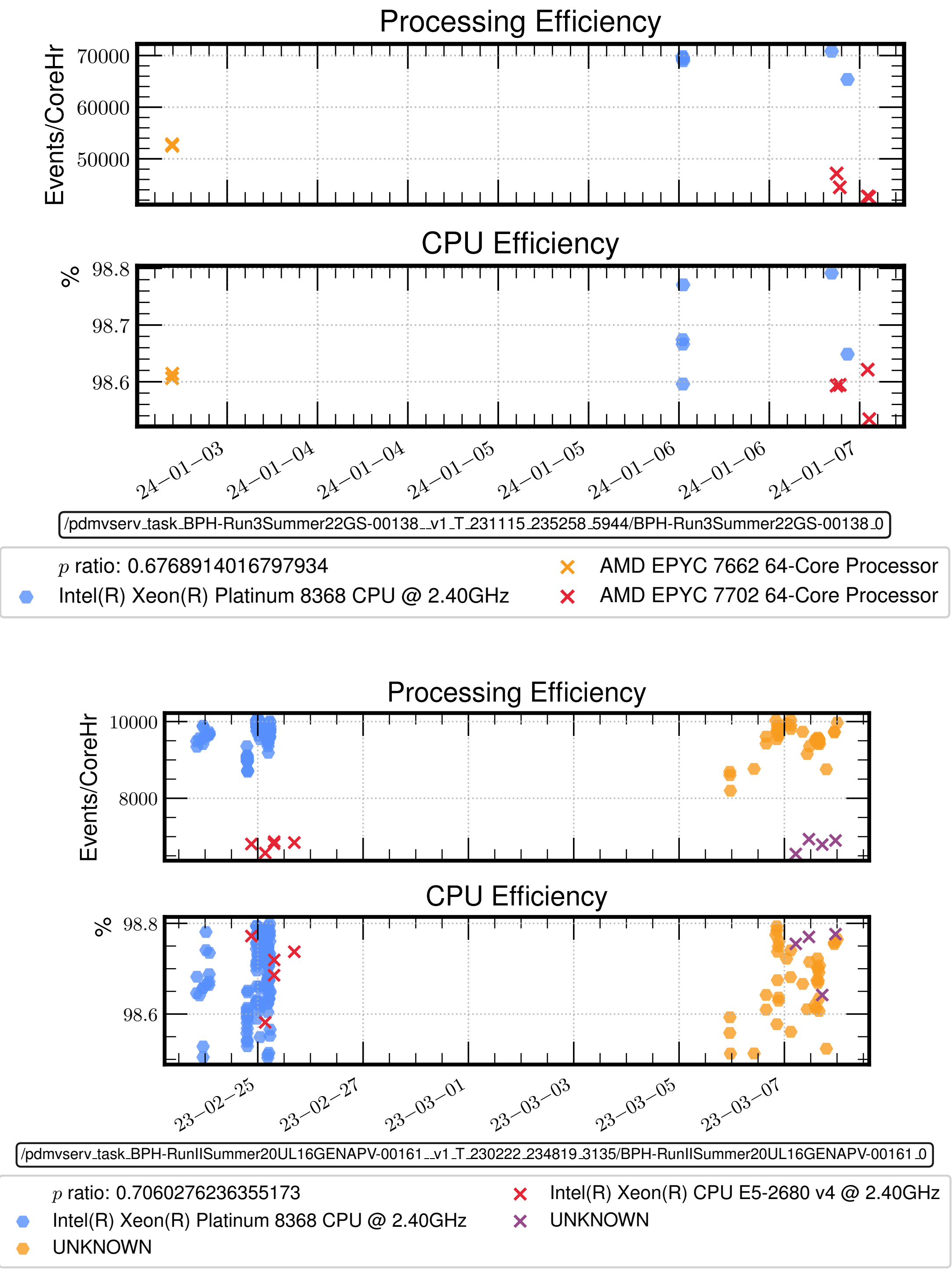


Figure A.35.: The top distribution shows that the results are even consistent with only six events per site. The bottom figure, however, shows that the monitoring data is not always totally correct. From the time separation, it can be concluded that there was a problem with reporting when the workflow occurred for the second time. The exact reasons can no longer be determined in retrospect. Nevertheless, due to the very similar processing efficiency, the UNKNOWN events can be assigned with relative certainty to the according CPU model.

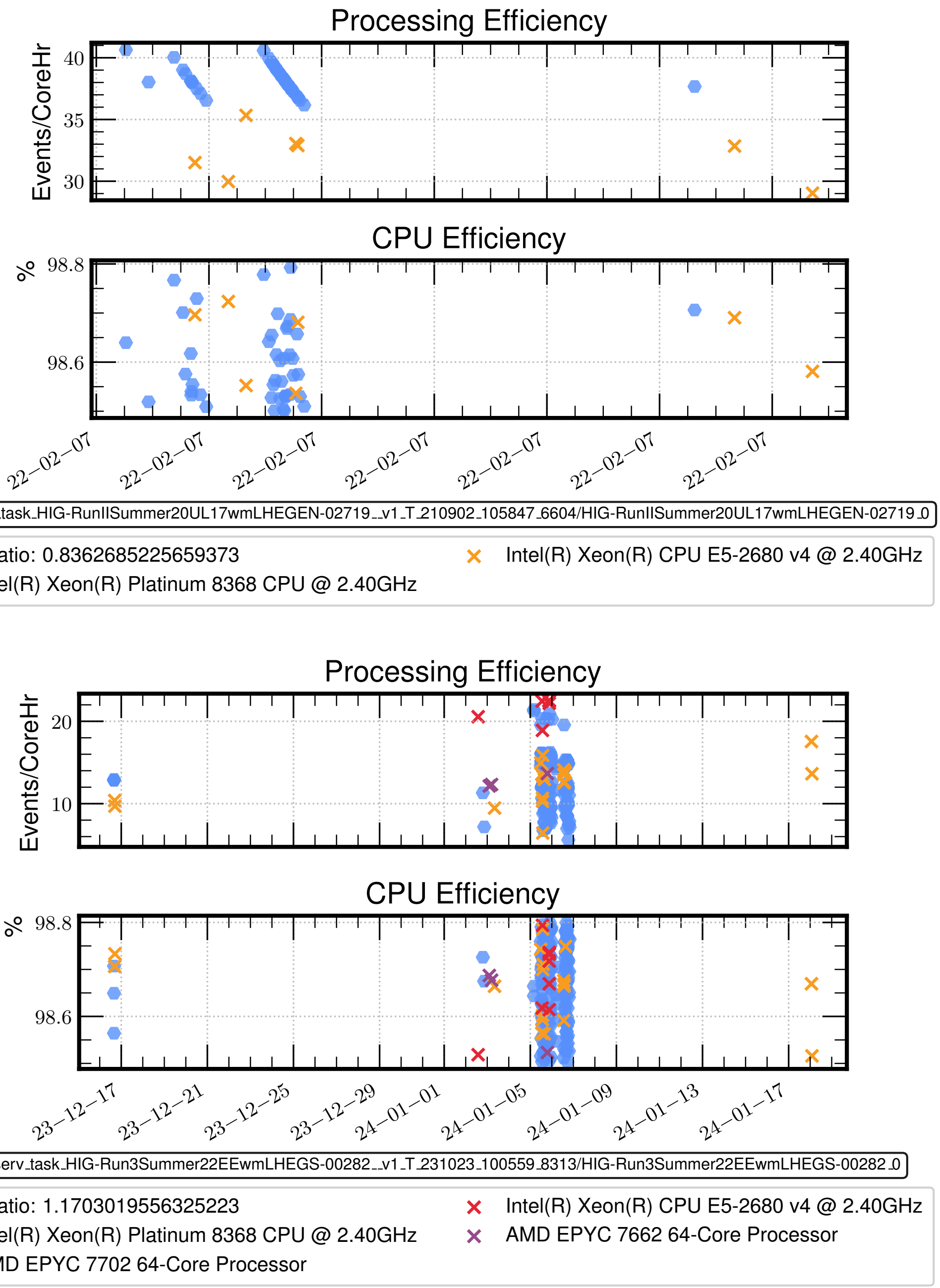


Figure A.36.: Both figures show workflows with strange, unreasonable behavior observed at HoreKa (hexagon) and TOpAS (x). They were both sorted out for the final evaluation. The top distribution shows a clear degrading effect of the processing efficiency at HoreKa. The second one is not representative for the p estimation, as the spread of the processing efficiency is two large. Only with a rather constant processing efficiency, the results can be reliably considered. On top, the oldest CPU yields the best result, which also indicates discrepancies for the workflow. This can happen for multiple reasons that are not reflected in the monitoring data, like e.g. interference effects between running jobs.
In conclusion, the disregarding is well-satisfied.

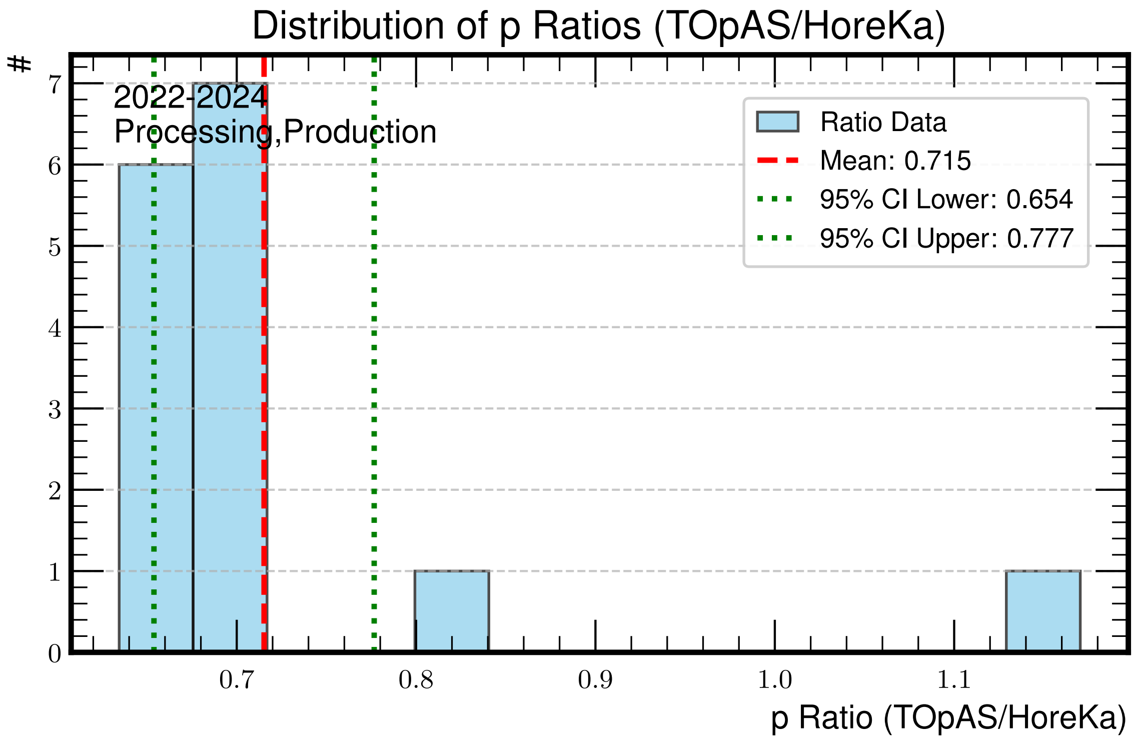


Figure A.37.: This figure shows the full evaluation with both outliers that were removed in Fig. 5.9. Even though the removal is well-justified, keeping them would also change the result by only around 3 % in total. This underlines that a very reliable performance factor estimation is possible with the newly introduced method.

to a certain CPU model (UNKNOWN). The reasons are unclear, but most certainly reporting issues, as it seems. Since the utilized CPU is not essential to evaluate the overall performance, this workflow still can be used, as long as it is consistent. Furthermore, for this specific example, the UNKNOWN jobs are in good agreement with the jobs where the CPU model is known. They can therefore be assigned to the same models with a high degree of certainty. Especially at HoreKa, where only one CPU model is deployed, this is even trivial.

Finally, the last step is to evaluate the most probable p factor from the individual results. The distribution of the ratios of all selected workflows at HoreKa and TOpAS is shown in Fig. A.37. For evaluating the final result, the two outliers (see Fig. A.36) were sorted out manually, leading to the distribution presented in Fig. 5.9. Even though the effect is only in the single-digit percent range, the workflows yield unreasonable results and are accordingly disregarded.

For all other evaluated sites, the p factor ratios are estimated in the same way. All associated distributions are provided in Section A.3.1. The final results in relation to TOpAS are provided in the upper part Table A.4. Not always a sufficient number of common workflows is available to use the method fully reliable. These cases are indicated in parentheses. However, even in those cases, the results are mostly consistent.

The final ratios that are used for further analyses are provided in the last column of the table that contains the estimated values for the full time period (2022-2024).

Table A.4.: The upper part of the table shows the same p ratios as Table 5.3. Parentheses indicate the cases where only a small number of common workflows meeting the requirements to evaluate the performance ratio are available. Overall, the results are very consistent. The last column contains the values that are eventually used for further analyses. In addition, the ratios between the other sites and HoreKa are derived which serve as consistency checks for the method.

Sites	2022	2023	2024	2022-2024
$p_{\text{TOpAS}}/p_{\text{HoreKa}}$	0.66 ± 0.02	(0.69 ± 0.01)	0.67 ± 0.02	0.68 ± 0.01
$p_{\text{TOpAS}}/p_{\text{GridKa}}$	1.05 ± 0.05	(1.01 ± 0.18)	0.98 ± 0.02	1.06 ± 0.05
$p_{\text{TOpAS}}/p_{\text{RWTH}}$	1.33 ± 0.07	(1.20 ± 0.13)	0.94 ± 0.05	1.09 ± 0.05
$p_{\text{TOpAS}}/p_{\text{RWTH-HPC}}$	(0.85 ± 0.03)	(0.88 ± 0.02)	(0.76 ± 0.02)	0.82 ± 0.02
Consistency Check:				
$p_{\text{GridKa}}/p_{\text{HoreKa}}$	0.65 ± 0.03	(0.84)	0.65 ± 0.04	0.67 ± 0.03
$p_{\text{RWTH}}/p_{\text{HoreKa}}$	0.52 ± 0.03	0.53 ± 0.02	0.70 ± 0.05	0.55 ± 0.03
$p_{\text{RWTH-HPC}}/p_{\text{HoreKa}}$	0.81 ± 0.08	(0.79 ± 0.01)	(0.81)	0.78 ± 0.004

Because a hard split between the years may also split workflows running in both years, which may reduce statistics within each year. In the worst case so much that the workflow has to be neglected, while it may could be used for the estimation when fully considered. The effect can be observed for RWTH-HPC, where the statistics for each year are very low but the full period is sufficient to provide a trustworthy result. But anyway, all results are consistent. Therefore, using the full period provides more robust results – as long as the hardware does not change too drastically. This condition is satisfied[3] for all reviewed sites, but RWTH.

Only for the Tier-2, the values change significantly over time, as shown in Table A.4. This can be explained by the increasing share of more performant hardware commissioned over time at the site. While in 2021, the site provided an officially benchmarked performance value of around 10 [89], the value for 2024 is significantly higher (HS23 score: 14.41 [89]). In such a case, the method of course yields a way better result when considering the more granular factors over time – if sufficient statistics are available for the estimation. However, for evaluations of the grid site over the full time period, the value for the full time period can still be seen as a decent approximation, as it is averaging out the higher (2022) and lower (2024) performance ratios for RWTH in comparison to TOpAS.

The lower part of Table A.4 additionally provides the results for the performance factor ratio estimation between the other sites and HoreKa, which serve as consistency checks. When comparing the values, they show good agreements.

Finally, from the p ratios also absolute performance ratings can be derived when compared to a well-known baseline. This can additionally serve as a consistency check. In the top part of Table A.5, absolute values equivalent to HS23 are calculated from the performance ratios between the sites and TOpAS in combination with the HS23 score of TOpAS.

The agreement for HoreKa, GridKa, and CLAIX (RWTH-HPC) is extraordinary good. The result for the Tier-2 center at RWTH Aachen is more off, caused by the effects explained above. In this case, a nearly perfect agreement is achieved when

[3] This can be confirmed by the rather constant p ratios, as well as the used CPUs in the monitoring data, which do not change significantly.

Table A.5.: The table shows the HS23 scores derived with the described method and TOpAS (bold) as the absolute basis for comparison. The official values for GridKa and RWTH are taken from CRIC. In parentheses, HS23 values are provided which were privately derived with the official benchmark suite [185]. The calculated HS23 equivalents are using the results for the entire period from 2022-2024, see Table A.4. The bottom part presents the obtained values based on the performance in relation to HoreKa, which can be seen as an additional cross-check.

Site	Official HS23	HS23 Equivalent
TOpAS	(**13.5**) [185]	baseline
HoreKa	(20.2) [185]	20.15
GridKa	12.75 [89]	12.74
RWTH	14.4 [89]	12.39
RWTH (2024)	14.4 [89]	14.36
RWTH-HPC	(16.7) [187]	16.46

Consistency Check:

Site vs. HoreKa	p	Calculated	Deviation to Official
GridKa	0.67	13.54	5.8 %
RWTH	0.55	11.11	22.8 %
RWTH-HPC	0.78	15.76	5.6 %
RWTH(2024)	0.7	14.14	1.8 %

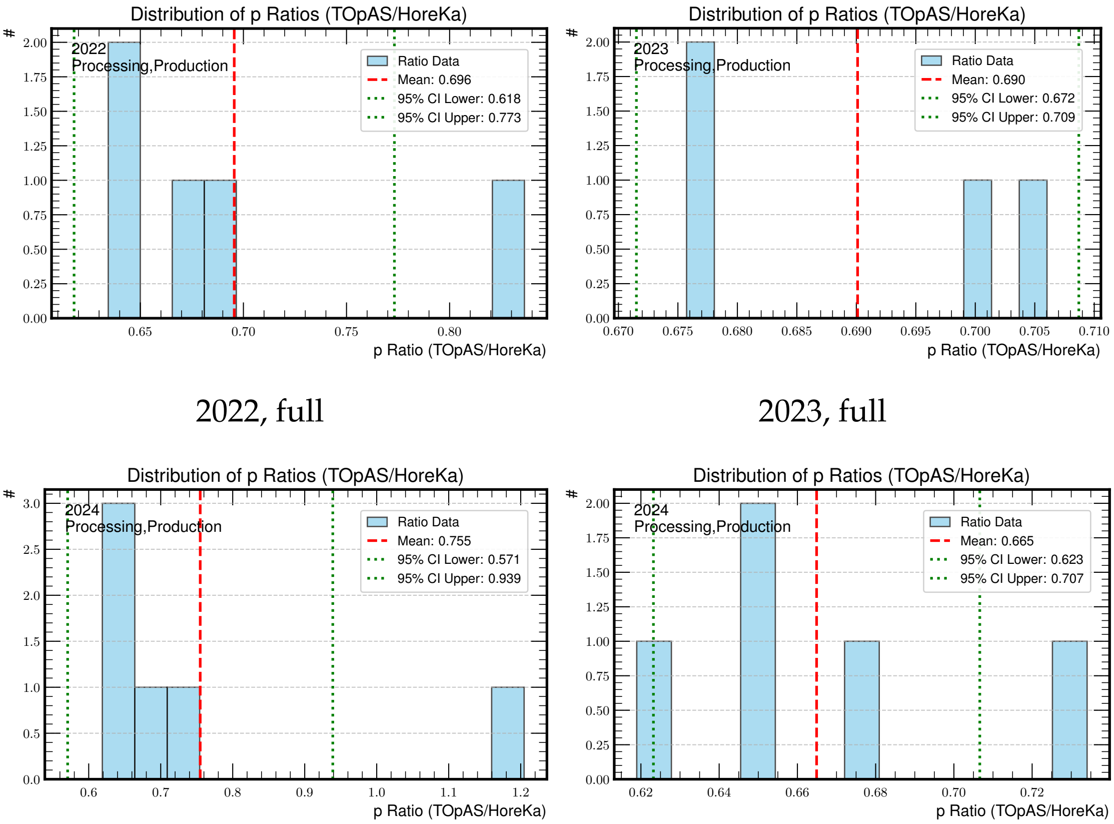
using the p ratio for 2024, instead of the averaged value for the full time period, as shown in the table.

The bottom part of it provides an additional consistency check. It uses the benchmarked performance factor for HoreKa (20.2) to calculate the values for the other sites based on their performance ratio with HoreKa. As the relative deviation shows, the results differ between 2 and 5 % (RWTH excluded), which is a remarkable result for the data-driven method.

In conclusion, the newly introduced, data-driven method for the pure performance evaluation of a grid site based on the production monitoring data is in very good agreement with official benchmarks. The findings show that consistent results with an uncertainty of around 5 % can be achieved. The obvious disadvantage, however, is that the method can of course only be used retrospectively. But for the evaluation of (opportunistic) grid resources without official benchmark values, the method presents a plausible alternative.

The best results can be achieved with the maximum statistics. For GridKa, TOpAS, and HoreKa, the values are also very consistent for the different years. Only for the Tier-2 center in Aachen the deviation is larger, which is mainly caused by the changing hardware setup, reducing the accuracy over multiple years.

A.3.1. P Ratio Estimation for All Sites



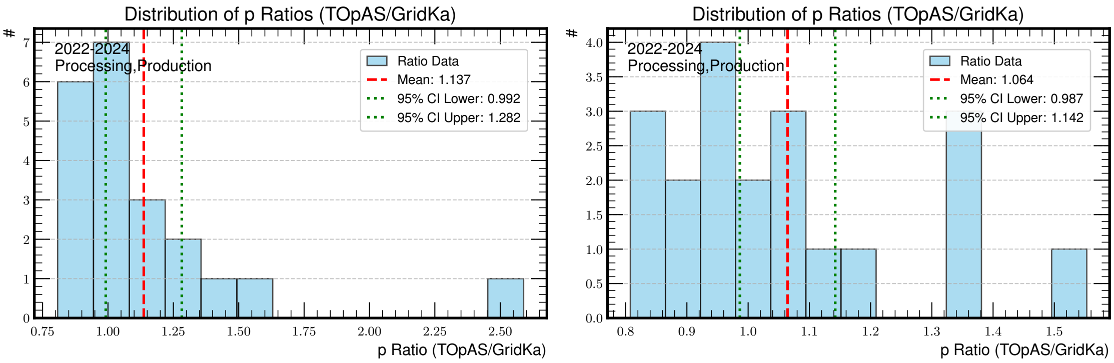
2022, full

2023, full

2024, full

2024, one outlier removed

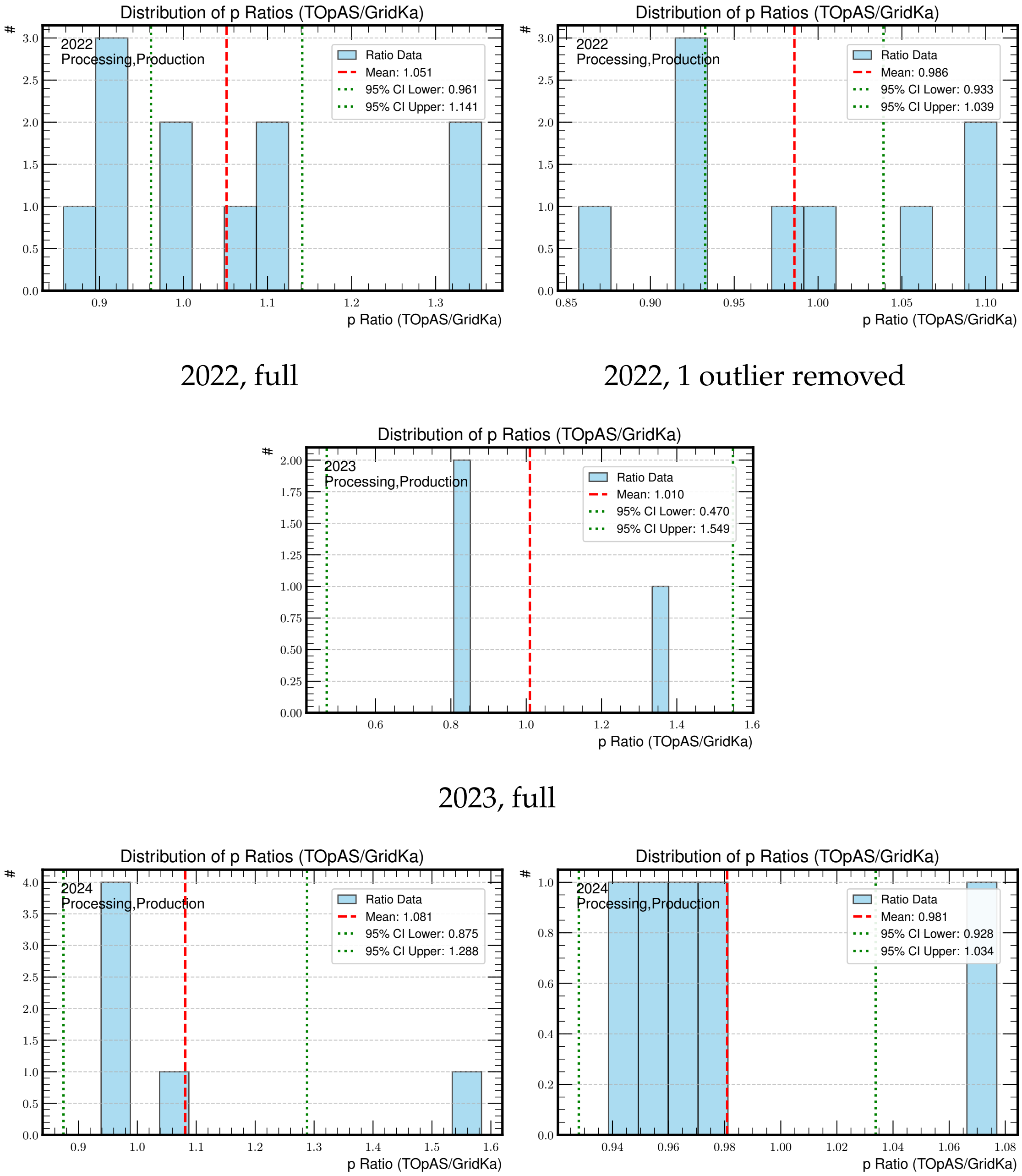
Figure A.38.: Overview of the individual results for the relative performance estimation between TOpAS and HoreKa.



2022-2024, full

2022-2024, one outlier removed

Figure A.39.: Overview of the results for the relative performance estimation between TOpAS and GridKa (I).



2022, full

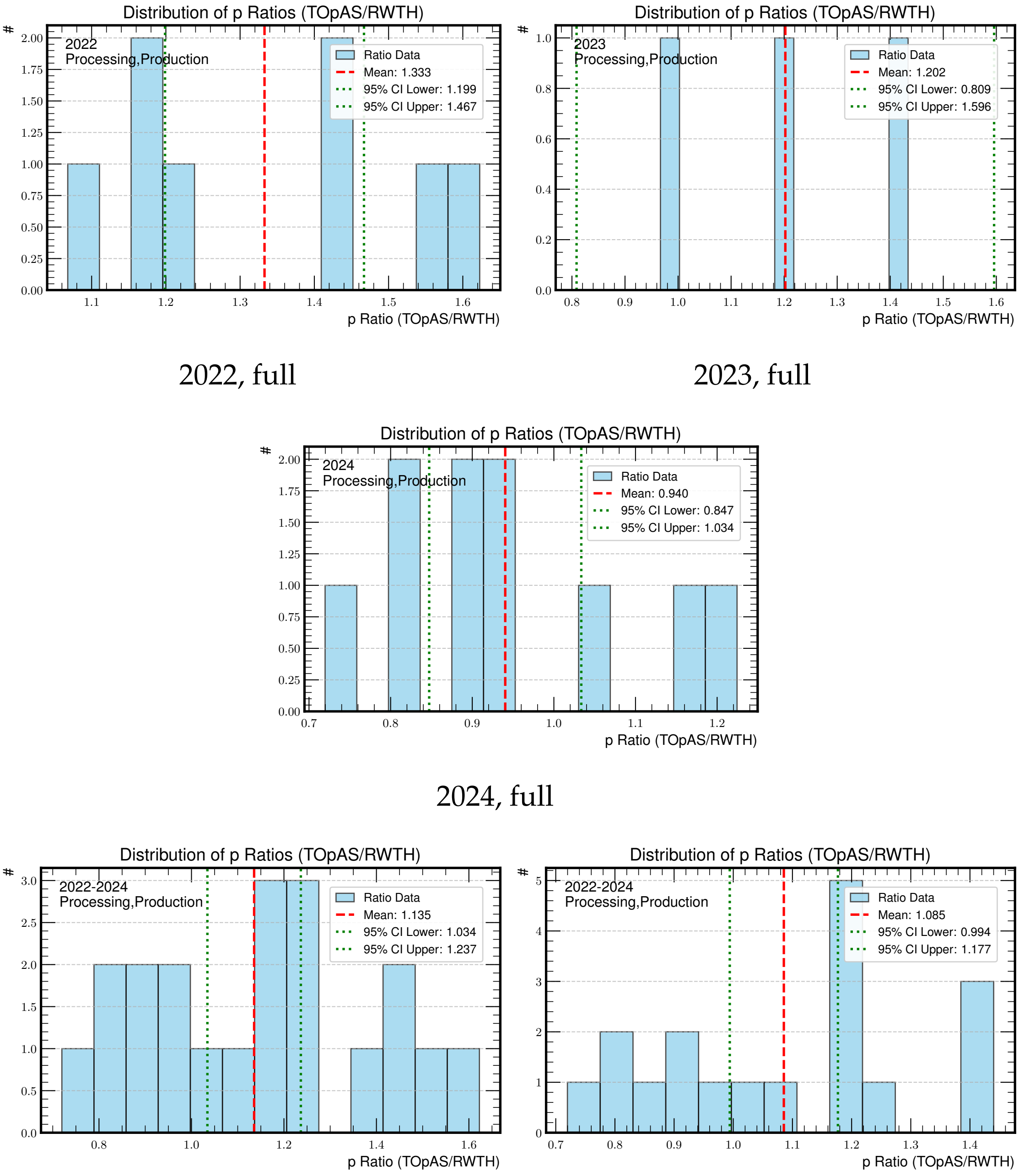
2022, 1 outlier removed

2023, full

2024, full

2024, one outlier removed

Figure A.40.: Overview of the results for the relative performance estimation between TOpAS and GridKa (II).



2022, full

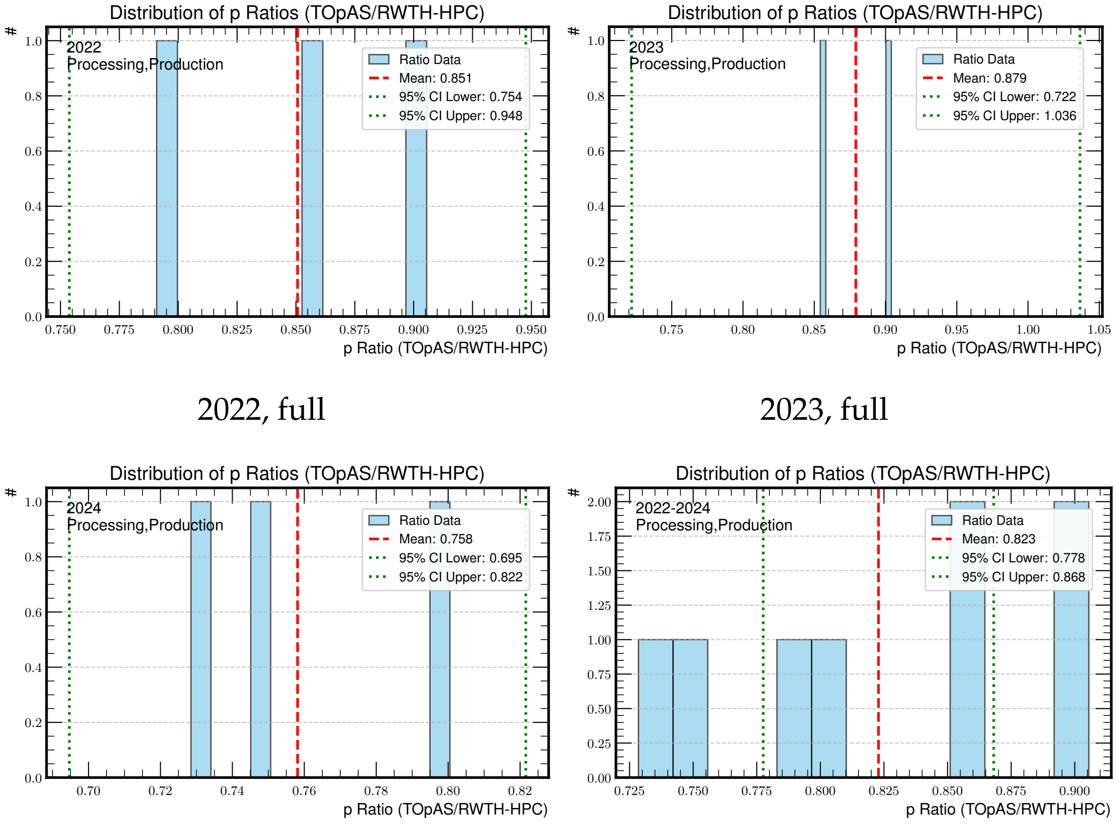
2023, full

2024, full

2022-2024, full

2022-2024, two outliers removed

Figure A.41.: Overview of the results for the relative performance estimation between TOpAS and the Tier-2 (RWTH).



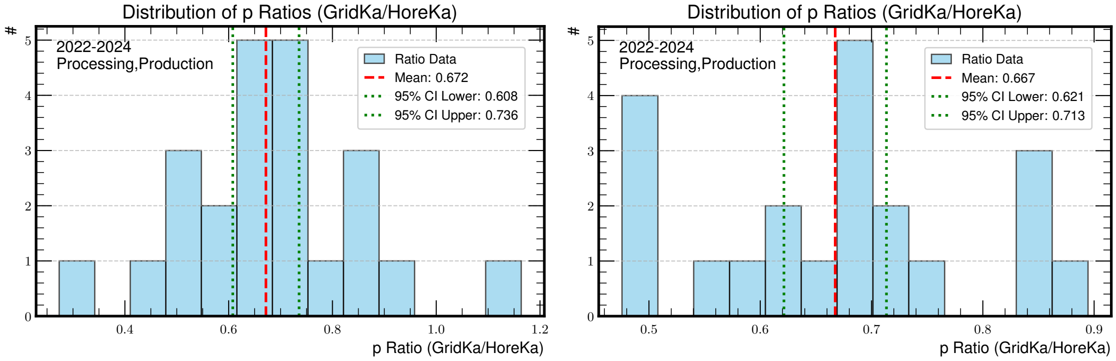
2022, full

2023, full

2024, full

2022-2024, full

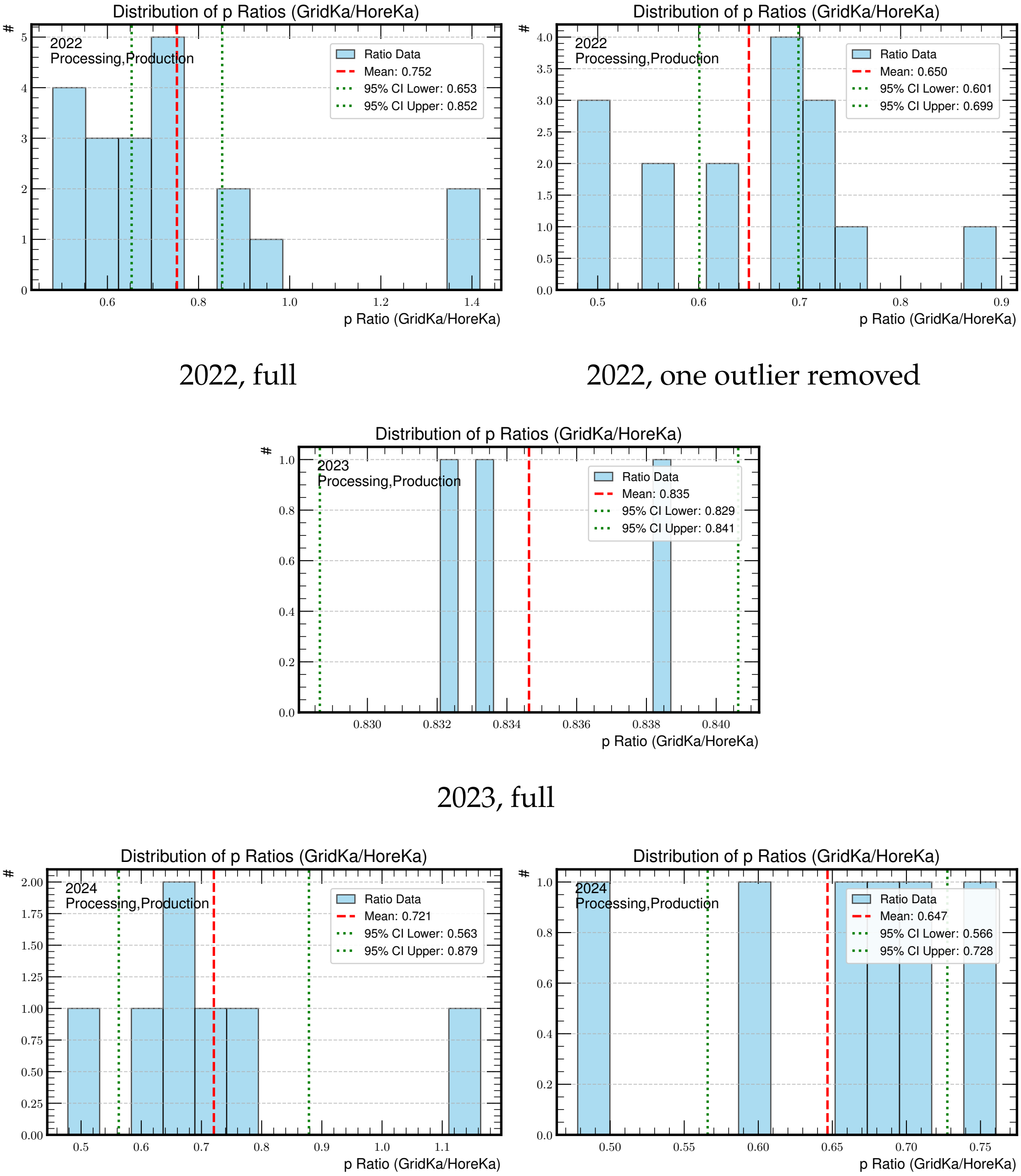
Figure A.42.: Overview of the individual results for the relative performance estimation between TOpAS and CLAIX (RWTH-HPC).



2022-2024, full

2022-2024, two outliers removed

Figure A.43.: Overview of the results for the relative performance estimation between GridKa and HoreKa (I).



2022, full

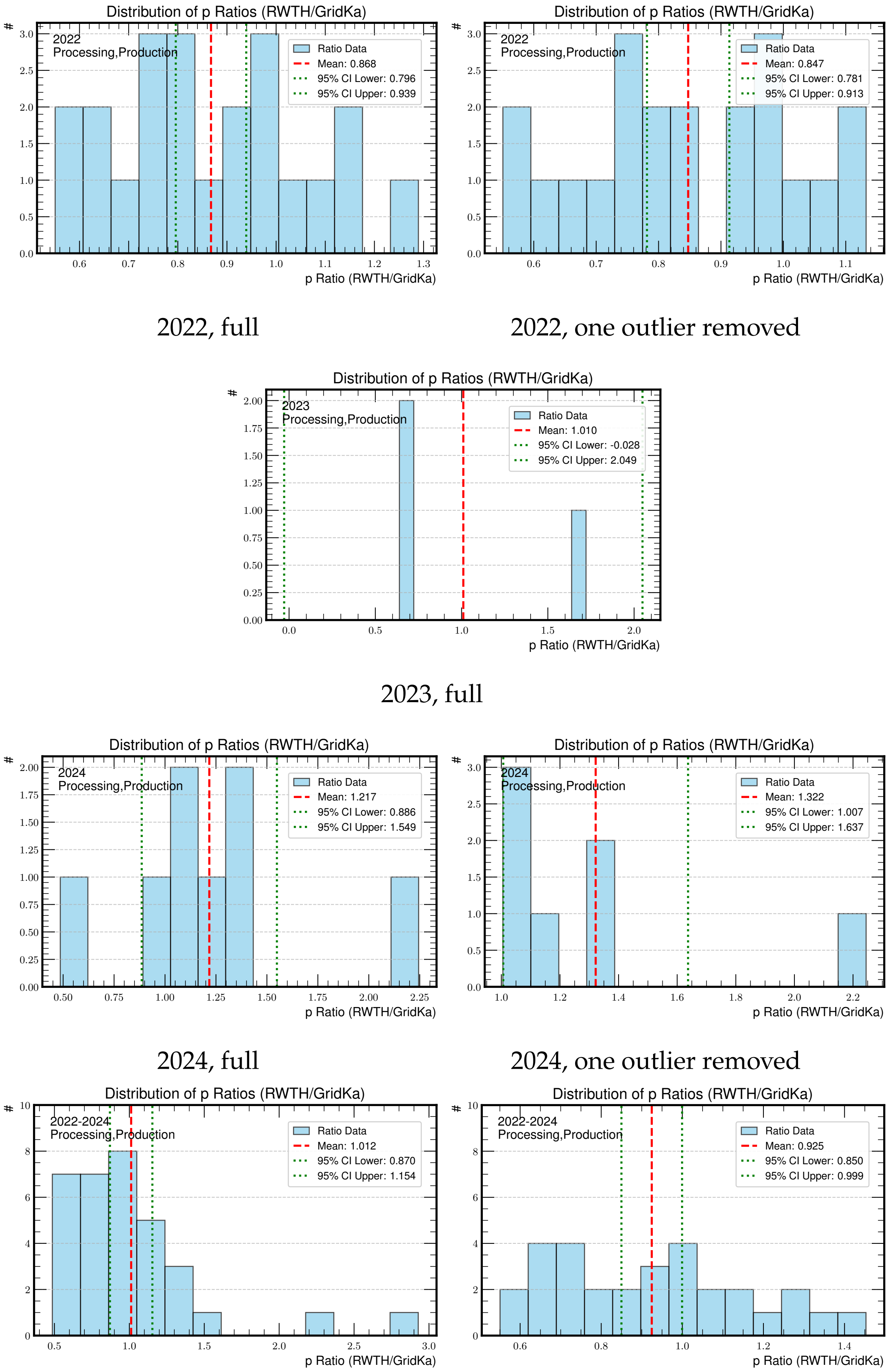
2022, one outlier removed

2023, full

2024, full

2024, one outlier removed

Figure A.44.: Overview of the results for the relative performance estimation between GridKa and HoreKa (II).



2022, full

2022, one outlier removed

2023, full

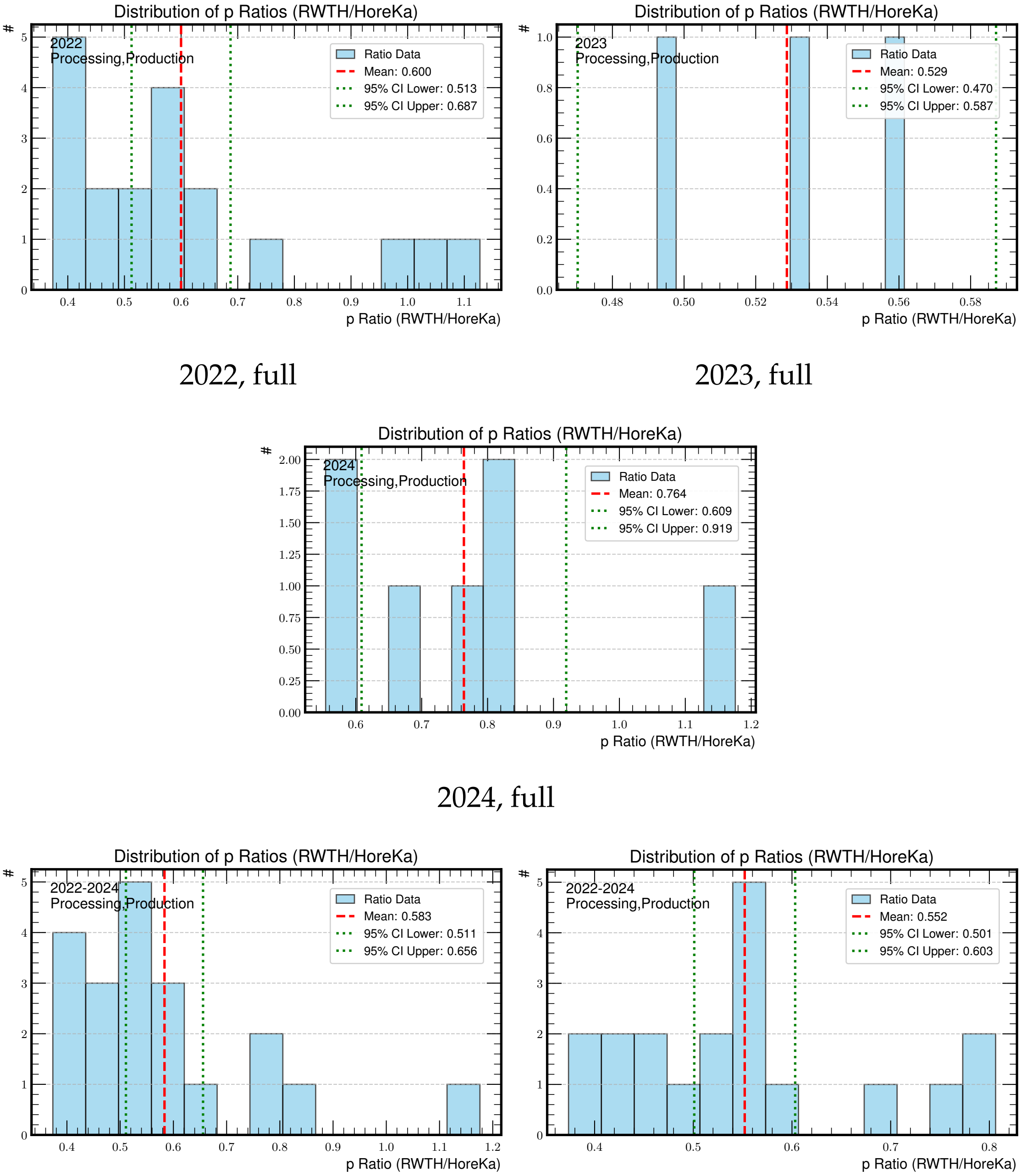
2024, full

2024, one outlier removed

2022-2024, full

2022-2024, two outliers removed

Figure A.45.: Overview of the results for the relative performance estimation between RWTH and GridKa.



2022, full

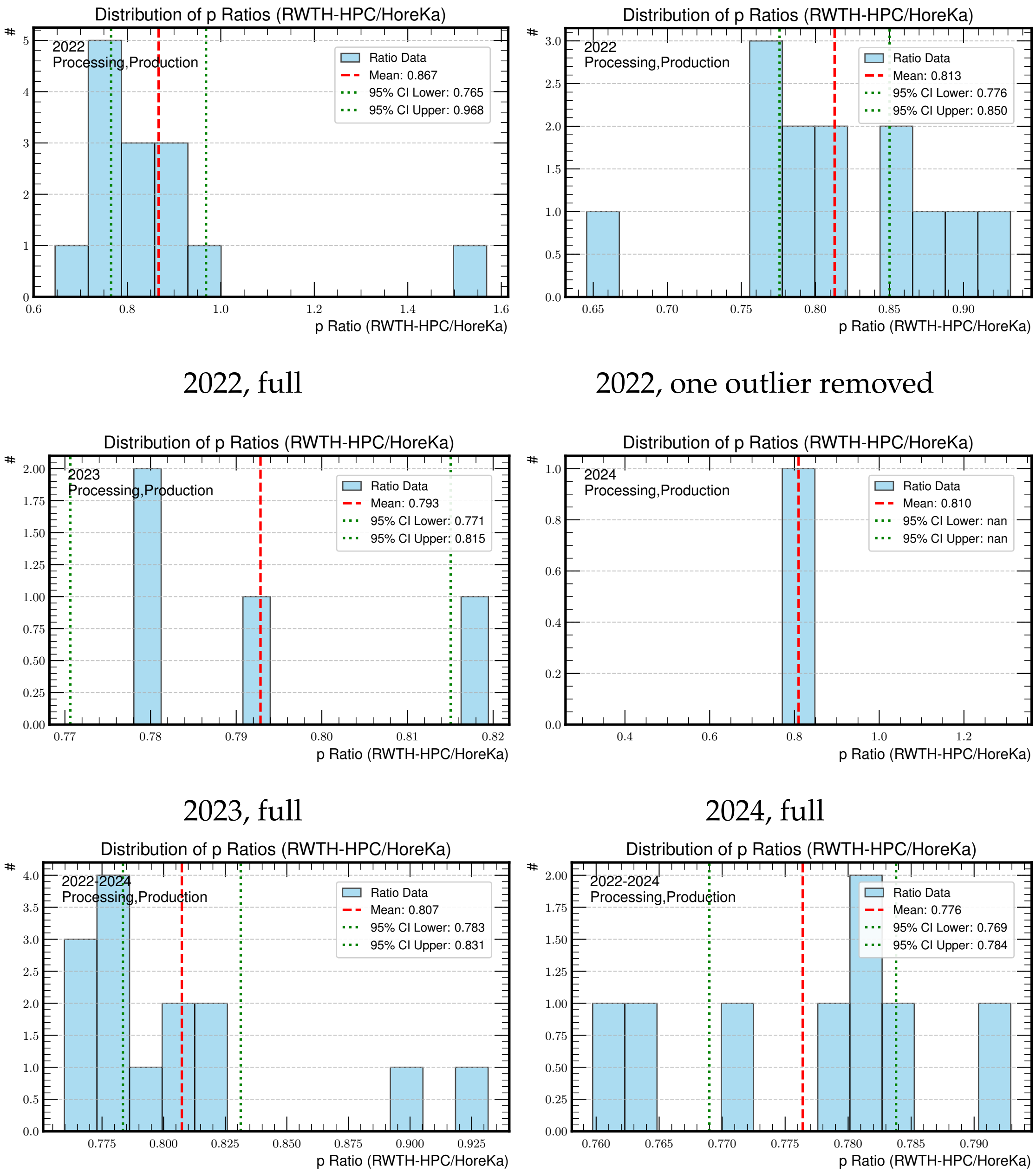
2023, full

2024, full

2022-2024, full

2022-2024, one outlier removed

Figure A.46.: Overview of the results for the relative performance estimation between RWTH and HoreKa.



2022, full

2022, one outlier removed

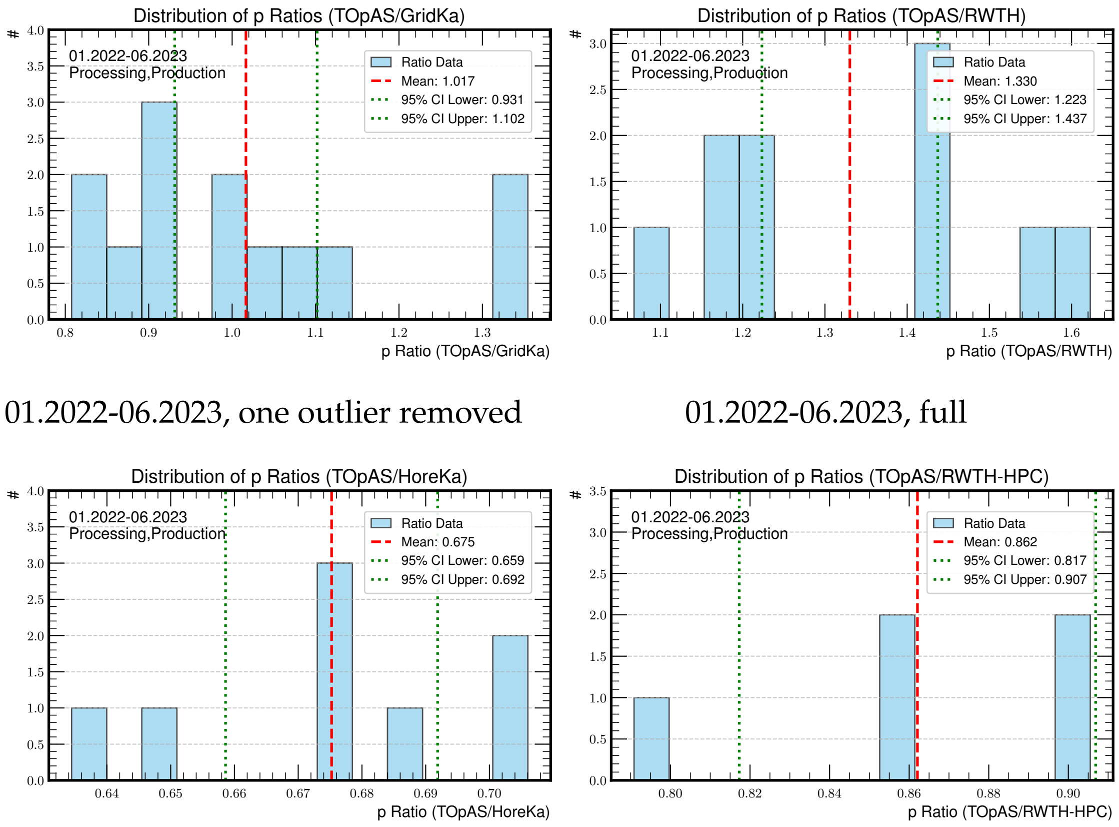
2023, full

2024, full

2022-2024, full

2022-2024, two outliers removed

Figure A.47.: Overview of the results for the relative performance estimation between CLAIX/RWTH-HPC and HoreKa.



01.2022-06.2023, one outlier removed

01.2022-06.2023, full

01.2022-06.2023, one outlier removed

01.2022-06.2023, full

Figure A.48.: Overview of the p ratio evaluations for the initial phase. The values are used in Table 5.5 for the relative performance estimation between TOpAS and the other sites.

Table A.6.: Additional filter steps to evaluate a more comparable processing efficiency. T describes the threshold for the minimal number of jobs from a workflow to be considered. N is the remaining number of workflows that are used for the evaluation. The impact of the steps is given with the changes in the total CommittedCoreHr* per site. A threshold of T=10 is selected as the relative reduction is the most similar. For more than 10, the statistics for RWTH-HPC drop too much.

Filter Step	T = 1, N=1611		T = 10, N=313		T = 50, N=83	
Site/CommittedCoreHr*	Tot	Rel	Tot	Rel	Tot	Rel
GridKa	35.15M	–	26.53M	75.5%	17.1M	48.7%
RWTH	19.71M	–	15.18M	77.0%	10.4M	52.5%
TOpAS	1.61M	–	1.30M	80.9%	970K	60.4%
HoreKa	1.79M	–	1.35M	75.6%	951K	53.1%
RWTH-HPC	3.75M	–	2.84M	75.8%	1.05M	28.1%

A.4. Additions to the Evaluation of the Initial Phase

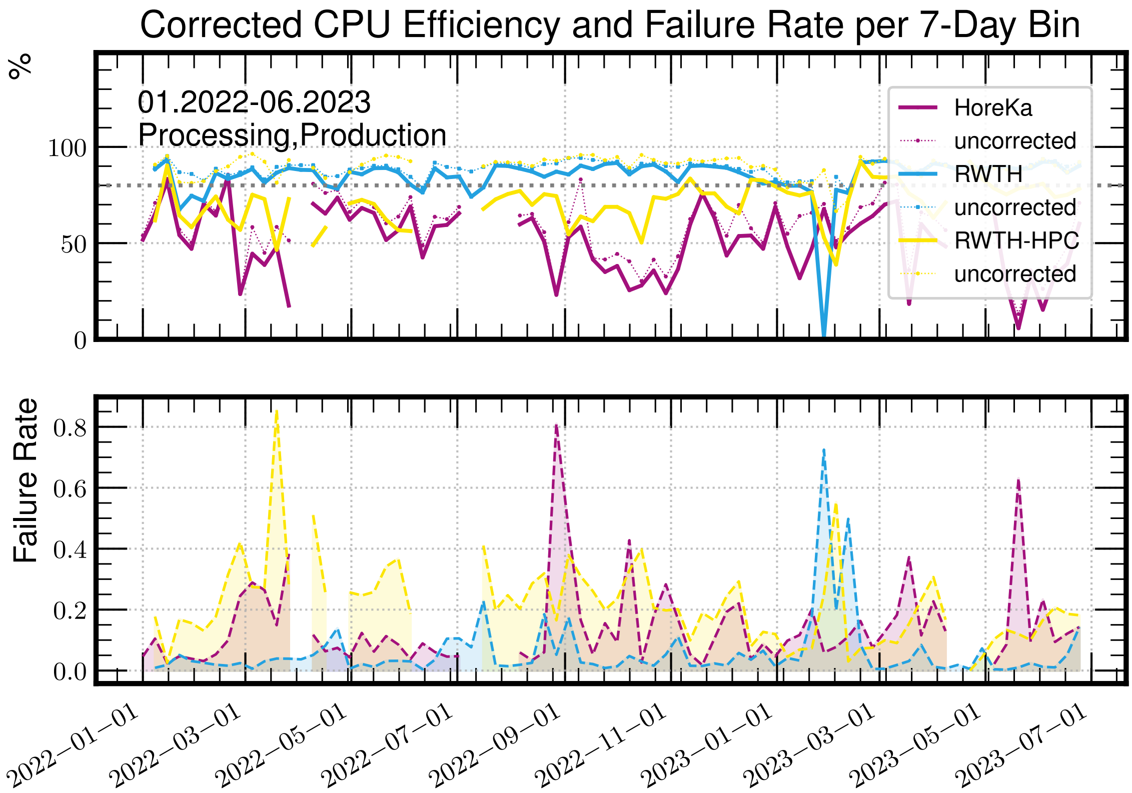


Figure A.49.: The figure shows the total corrected CPU Efficiency distribution of all investigated sites during the initial phase of the integration. This quantity takes the failure rate into account, as described in Section 5.5.1.6. The result is therefore better reflecting reality, since the impact of job failures is considered. Consequently, CLAIX, the site with the highest failure rate, degrades the most.

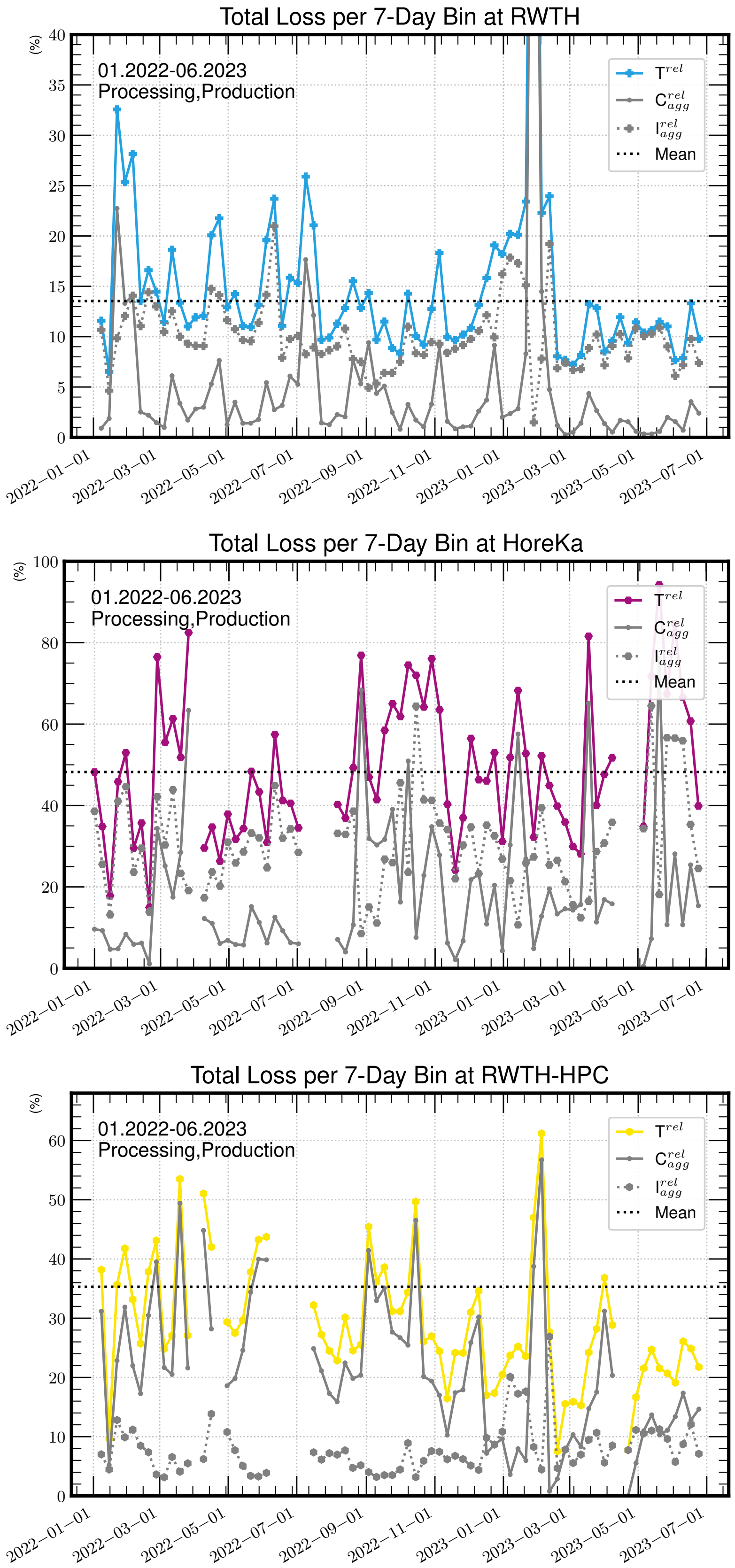


Figure A.50.: The composition of the Total Loss is shown for RWTH, HoreKa, and RWTH-HPC. All other sites are included in Fig. 5.13. The distributions show that WLCG site performs significantly better, as on average only around 15 % of the invested compute time is lost. For the two NHR centers, the value is two to three times higher. While HoreKa is mainly suffering from inefficiencies, CLAIX looses a remarkable part due to the high rate of failures.

Listing A.6: List of workflows analyzed in Fig. 5.14.

```
1: /pdmvserv_task_TOP—RunIISummer20UL17wmLHEGEN—00008__v1_T_210714_135143_8539/TOP—
RunIISummer20UL17wmLHEGEN—00008_0
2: /pdmvserv_task_HIG—RunIISummer20UL17wmLHEGEN—00983__v1_T_210713_174807_1455/HIG—
RunIISummer20UL17wmLHEGEN—00983_0
3: /cmsunified_task_HIG—RunIISummer20UL17wmLHEGEN—00979__v1_T_210713_204128_7793/HIG—
RunIISummer20UL17wmLHEGEN—00979_0
4: /pdmvserv_task_TOP—RunIISummer20UL17wmLHEGEN—00008__v1_T_210714_135143_8539/TOP—
RunIISummer20UL17wmLHEGEN—00008_0/TOP—RunIISummer20UL17SIM—00265_0
5: /cmsunified_task_HIG—RunIISummer20UL18wmLHEGEN—00503__v1_T_210713_202044_1283/HIG—
RunIISummer20UL18wmLHEGEN—00503_0
6: /pdmvserv_task_TOP—RunIISummer20UL18wmLHEGEN—00021__v1_T_210714_135651_5829/TOP—
RunIISummer20UL18wmLHEGEN—00021_0
7: /pdmvserv_task_TOP—RunIISummer20UL18wmLHEGEN—00021__v1_T_210714_135651_5829/TOP—
RunIISummer20UL18wmLHEGEN—00021_0/TOP—RunIISummer20UL18SIM—00264_0/TOP—
RunIISummer20UL18DIGIPremix—00264_0
```

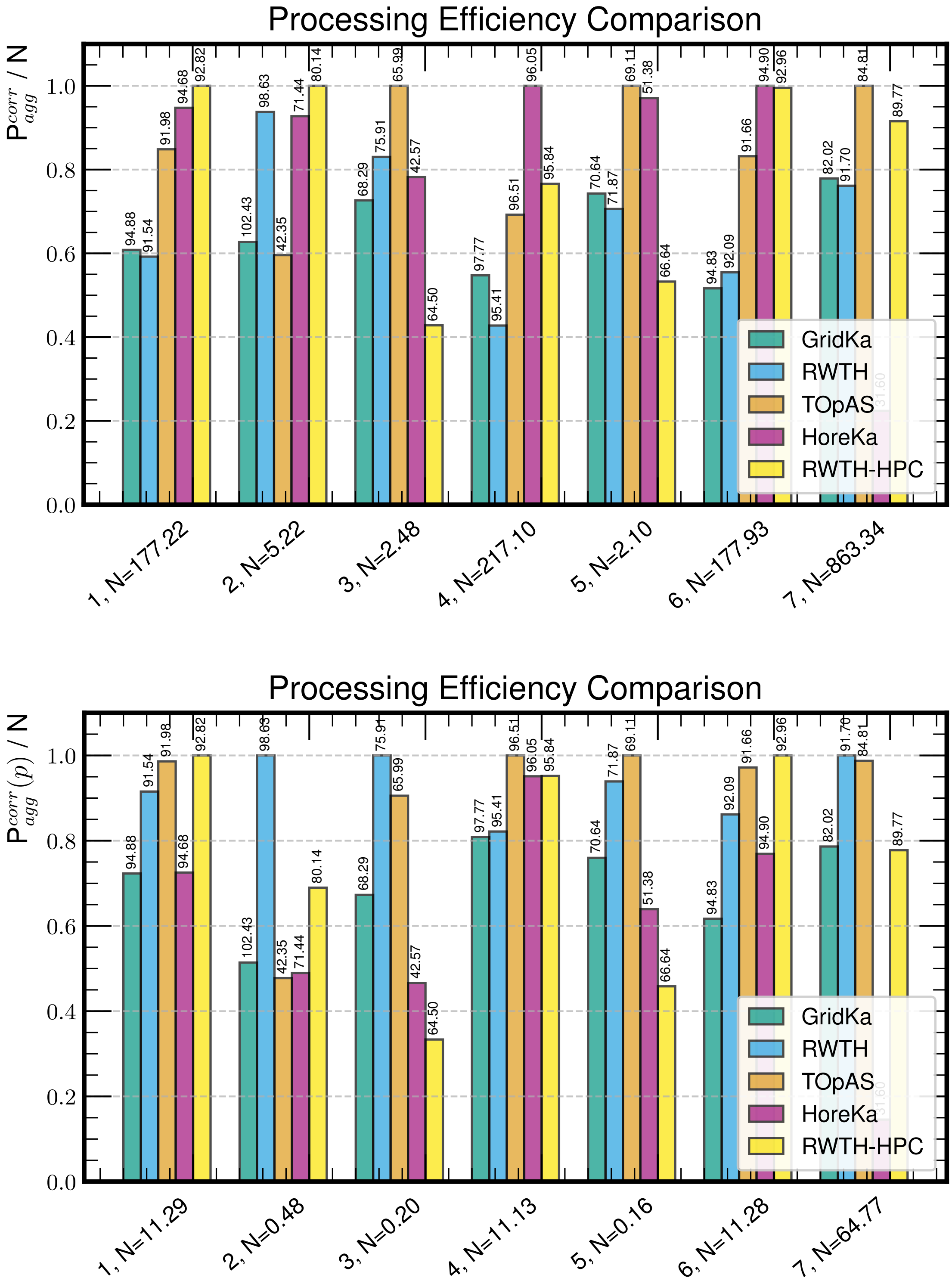


Figure A.51.: This figure shows the same comparison as Fig. 5.14, but displays the corrected quantities. As expected, the results for the HPC centers is more significantly reduced due their comparably higher failure rates.

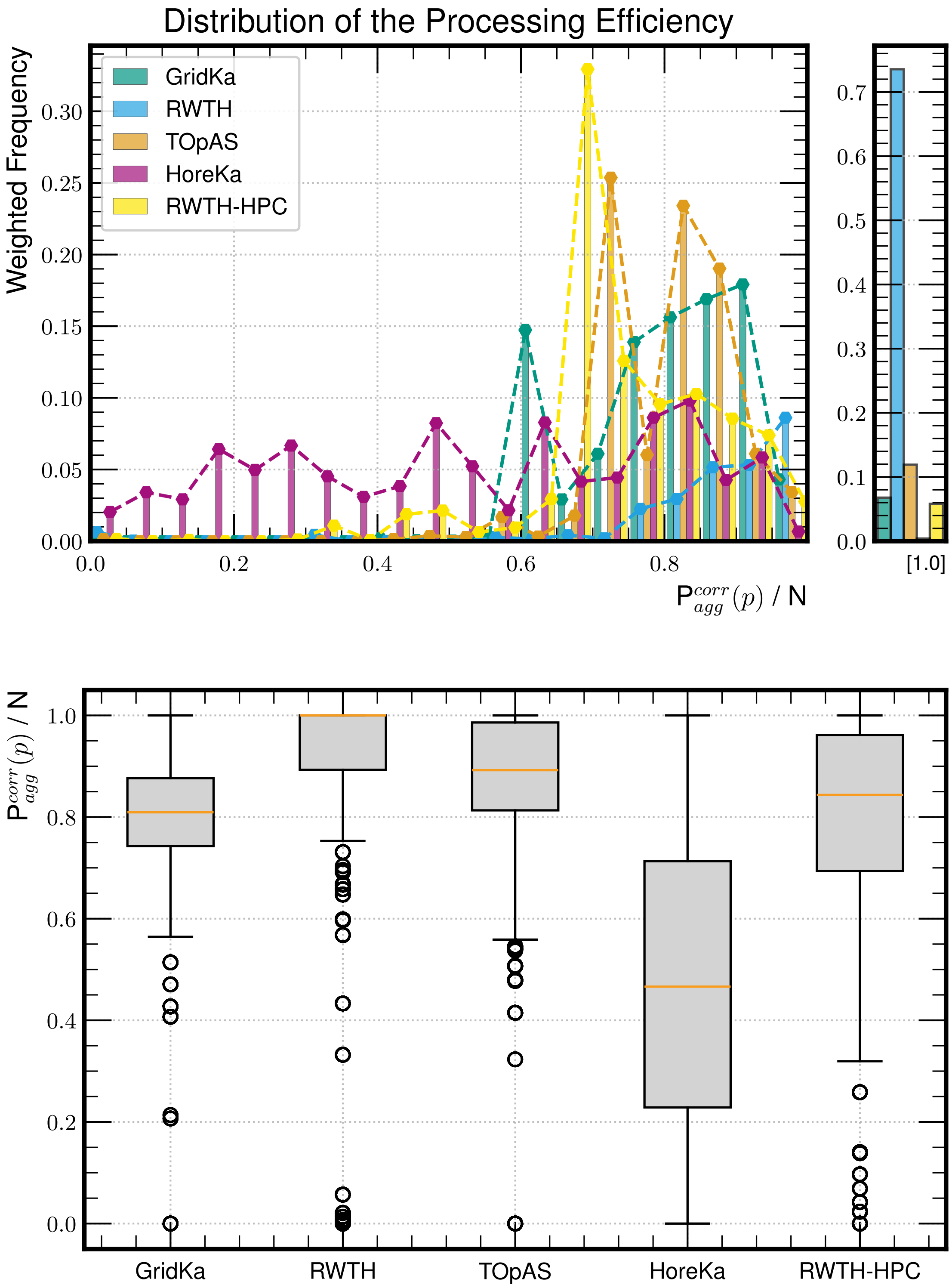


Figure A.52.: This distributions results from Fig. 5.15 by additionally incorporating the share of the total committed core hours of each workflow as a workflow weight. The differences are small in this case, as written in Table 5.5.

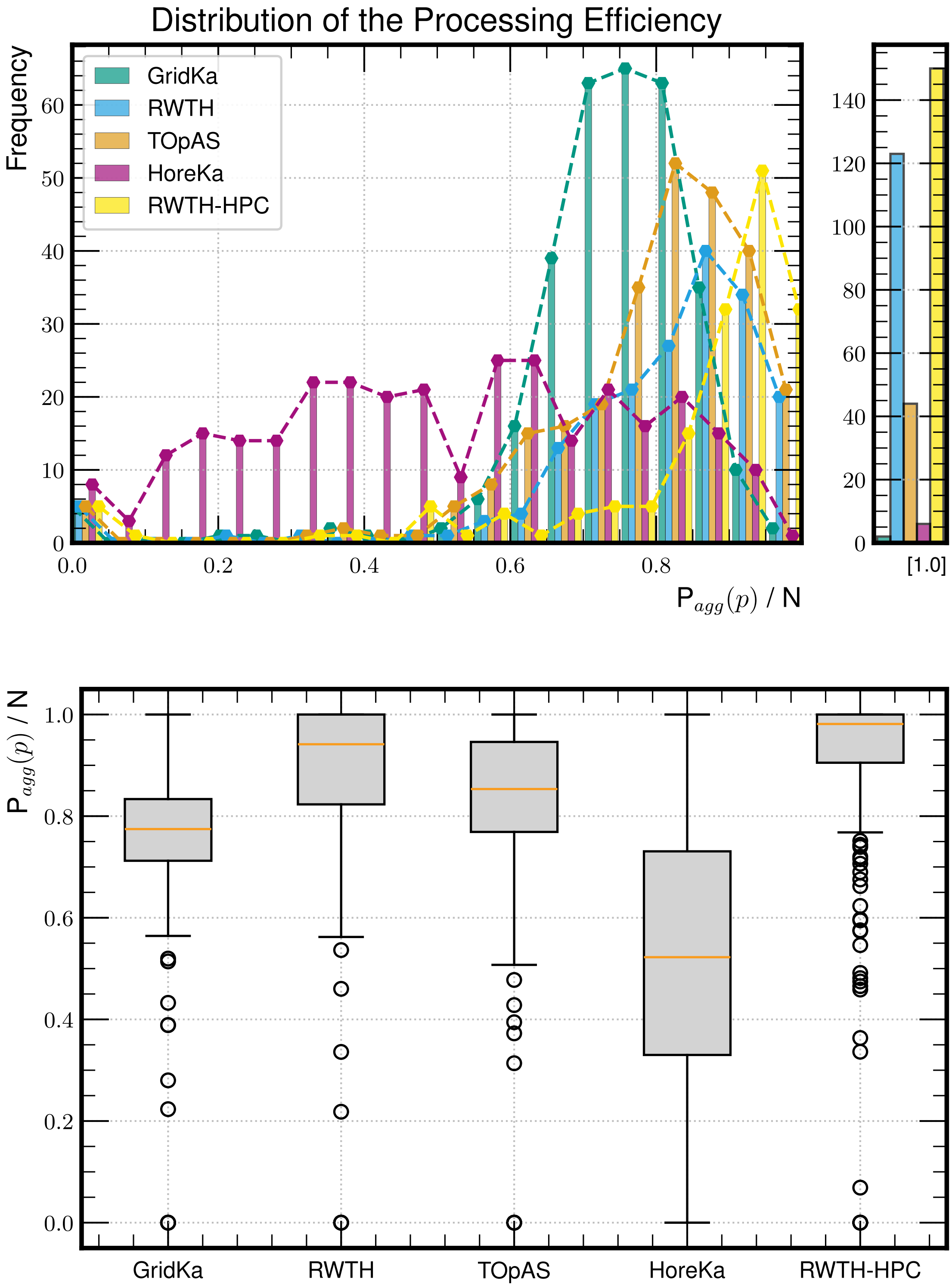


Figure A.53.: This version of the figure shows the aggregated but uncorrected processing efficiency with taking the performance factor of the sites into account ($P_{agg}(p)$). Here, the tails for RWTH-HPC are clearly reduced in comparison to Fig. 5.15. This indicates that the reason for the tails is the high failure rate of the center.
The distribution including the workflow weights is only marginally different, why it is not shown here.

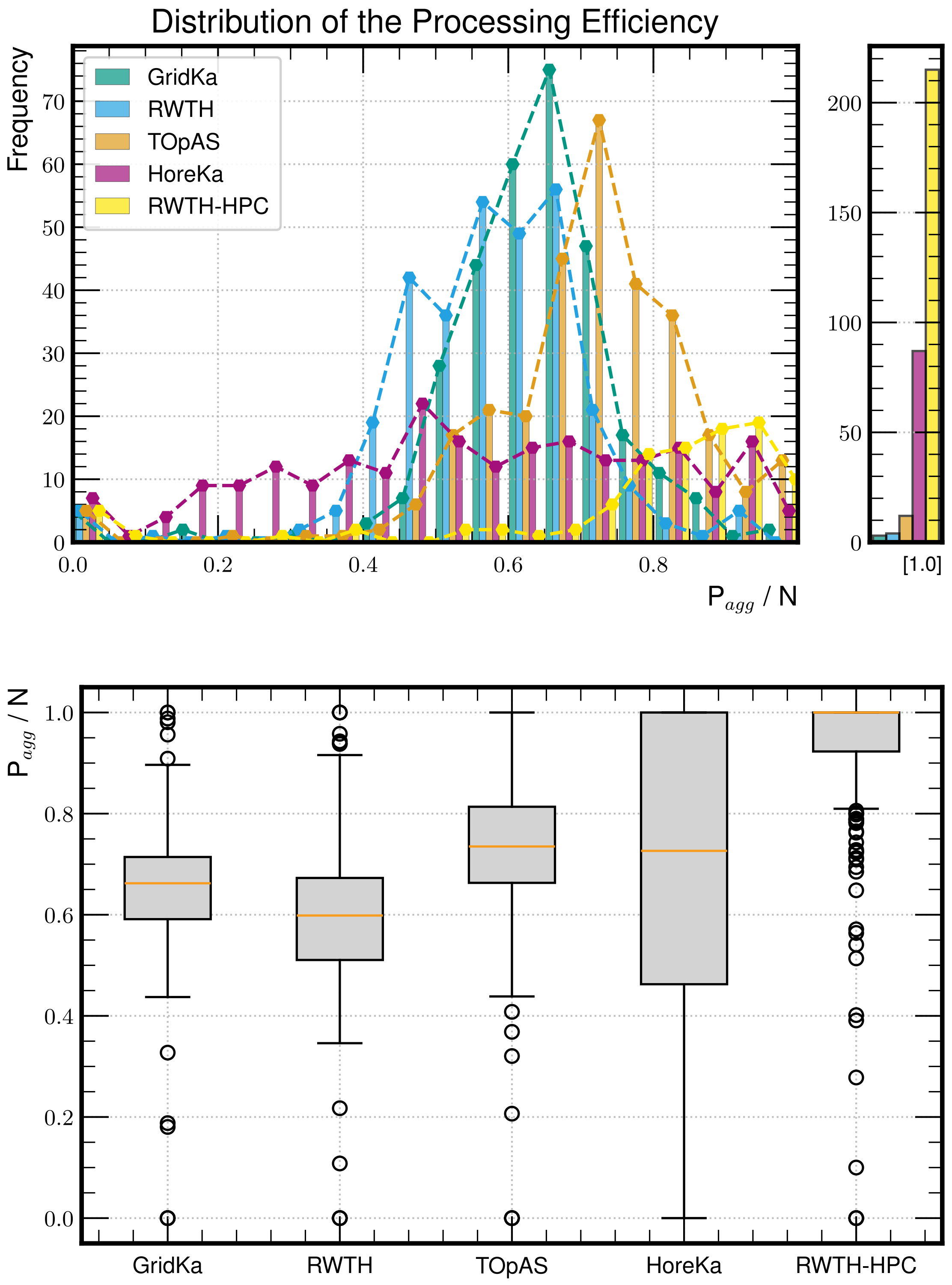


Figure A.54.: Here, the distributions are shown without considering the pure performance of a site (p) and without correcting for failures, nor workflow weights. This plain vanilla version of the processing efficiency represents simply the total processing throughput of the sites and is helpful to rate the potential of a resource. It clearly shows that the NHR centers are dominating the field. HoreKa has an overall broader spread, meaning that inefficiencies degrade the result. But it is still comparable to the WLCG sites.
The impact of workflow weights on the distribution is presented in Fig. A.55. In Fig. A.56, the same distribution is shown when the failure rate is considered additionally.

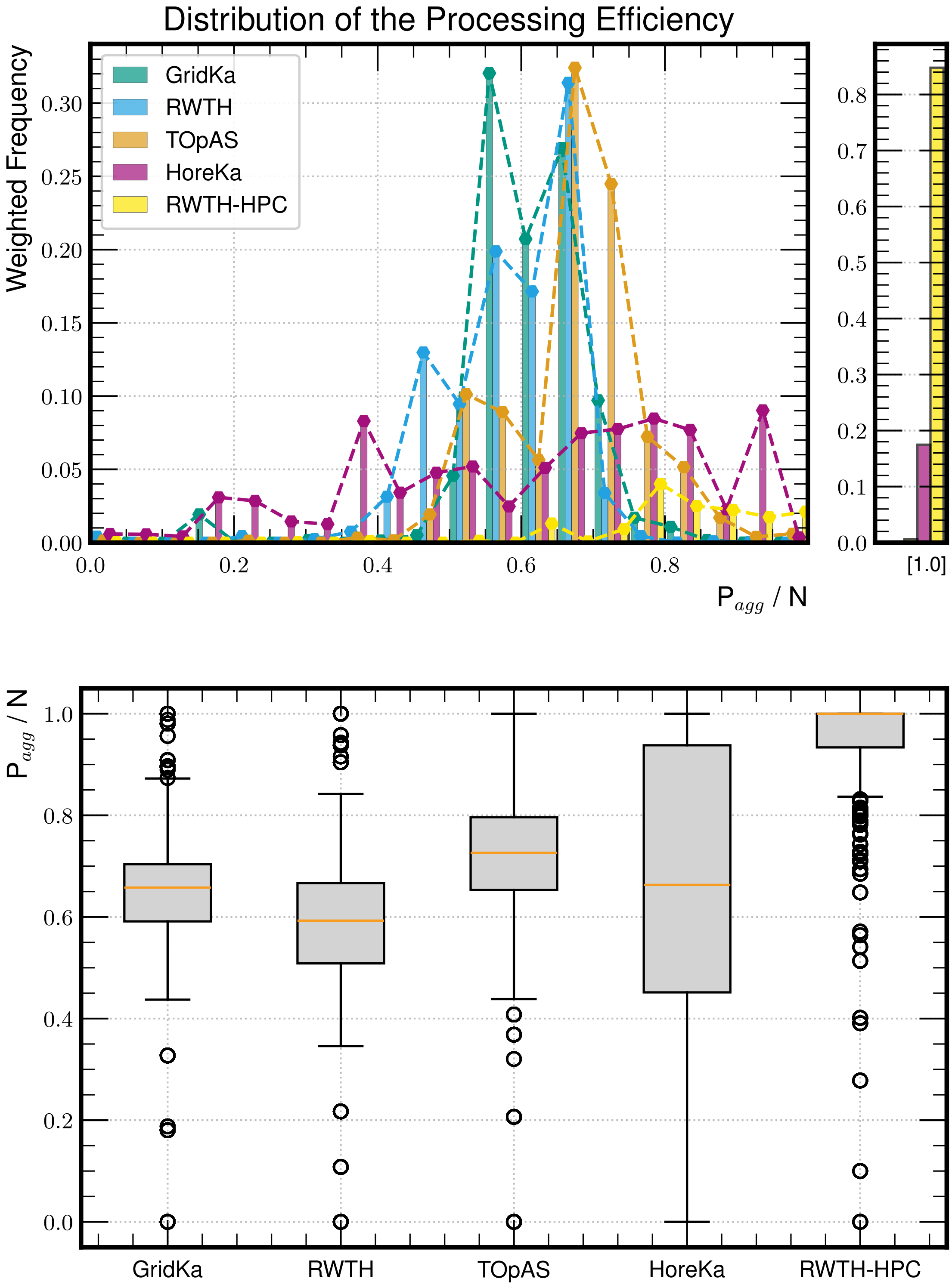


Figure A.55.: Distribution of the processing efficiencies in the same constellation as Fig. A.54, but with workflow weights applied. The differences are again marginal.

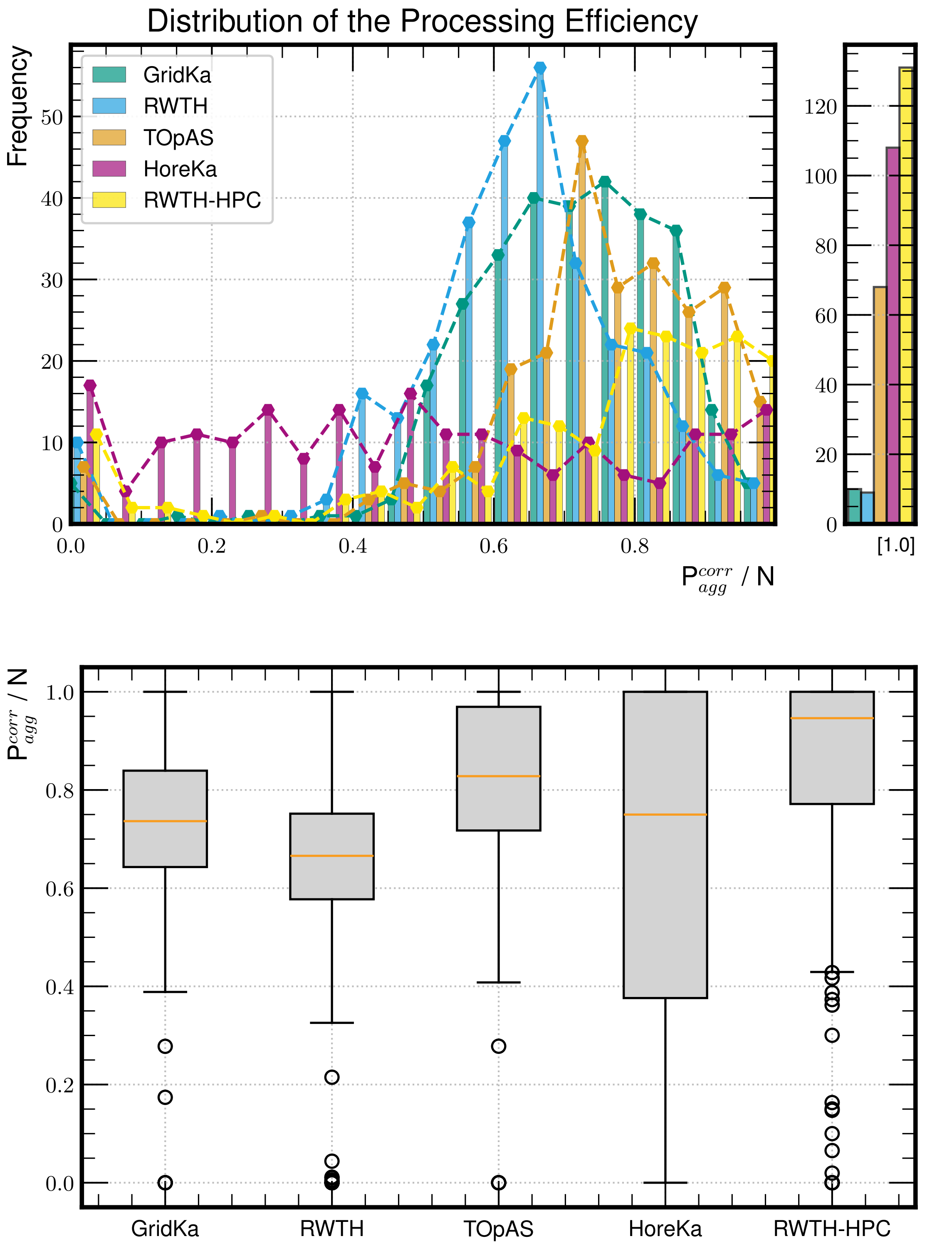


Figure A.56.: The figure shows the corrected processing efficiency, but without incorporating the pure performance factor p in the calculation. This figure therefore describes the definitive outcome the sites provided. The two NHR centers are clearly leading. HoreKa made up ground on CLAIX, which can be mainly accounted to the significantly higher failure rate at the latter, which is considered here. The overall performance of it is nevertheless still degraded by the long. significant tail to lower processing efficiencies. In Fig. A.57, the same constellation is shown with additional workflow weights.

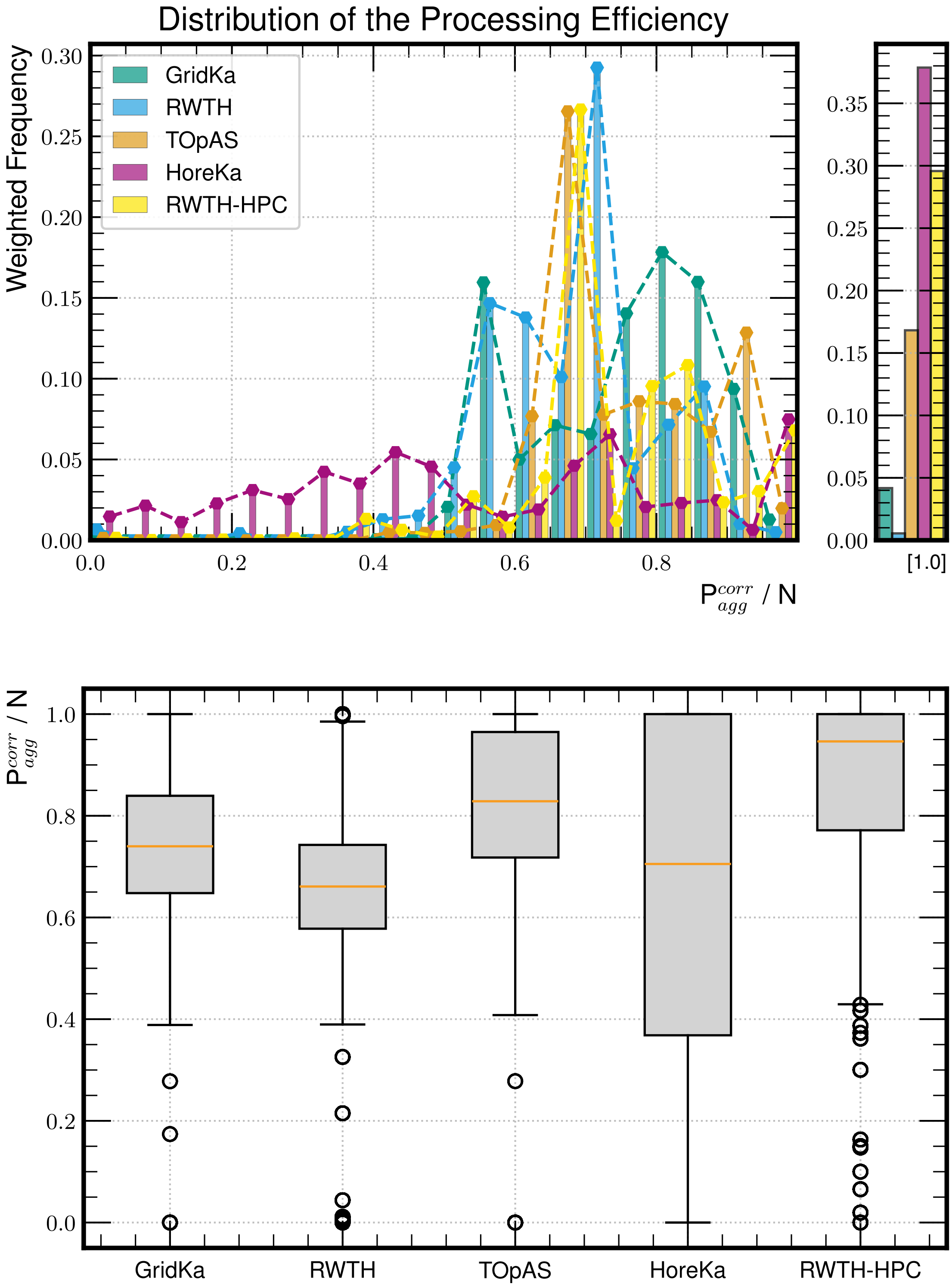


Figure A.57.: Analog distribution to Fig. A.56, but with workflow weights applied. Here, HoreKa is even outperforming CLAIX/RWTH-HPC in the 1.0-bin, showing that the workflow weights can make a difference.

A.5. Additions to the HoreKa Proof of Concept

A.5.1. Configuration and Setup

A repository with all required images, configurations, systemd units, a full monitoring stack, and further tools for development and tesing is provided in Ref. [191].

A.5.1.1. XRootD Configurations

Here, a minimal XRootD configuration for the presented prototype, including verbose monitoring, is presented. The additional summary monitoring is independent of the proxy configuration and uses an external tool called mpxstats as part of the XRootD software framework (see https://xrootd.web.cern.ch/doc/man/mpxstats.8.html).

Listing A.7: Basic XRootD config.

```
###########
# Exports #
###########
all.export /root:/
all.export /xroot:/

##### Network #####
xrd.port 1094
xrd.allow host *
# pss.inetmode {v4 | v6 } # default is v6

##### Forwarding #####
# With this configuration, the client defines the destination
pss.origin =
# Alternative: direct mode to the GridKa Redirector
# pss.origin = cmsxrootd-test.gridka.de

##### Filesystem #####
ofs.osslib libXrdPss.so

##### Caching #####
pss.cachelib default
oss.localroot /cache
pfc.ram 64g
pfc.diskusage 0.9 0.95 files 150t 180t 210t
pfc.prefetch 10
pfc.blocksize 2m # This has to be adapted to the underlying
# filesystem
# pss.dca world # Direct cache access (dca)

### Optimizations: ###
# Reduce very long waiting times which are pretty bad for CpuEff
pss.setopt ConnectTimeout 100s # default: 120s
pss.setopt ReconnectWait 30s #default: 1800
```

```
##### Security: #####
# Remark: In production we currently do not employ additional
# authentication and authorization features of XRootD, since
# these are in principle already covered by the workflow
# management systems


##### Monitoring #####
### Summary Monitoring ###
# +++++ ATTENTION +++++
# xrd.report monit.domain.edu:9931 every 5s all sync cache
# DNS resolving to v6 can lead to problems if the port forwarding
# etc is not configured properly (e.g. host only listening on IPv4)
xrd.report 123.45.678.910:9931 every 30s all sync cache

### Detailed g-stream Monitoring (json) ###
xrootd.mongstream all use send json fullhdr 123.45.678.910:9932

### Full Detailed Monitoring ###
xrootd.monitor all flush io 60s ident 5m mbuff 8k rbuff 4k \\
  rnums 3 window 10s dest files iov info user 123.45.678.910:9933

##### Debug Options #####
# maximum — very spammy
xrootd.debug all
xrootd.log all
xrd.trace debug
pss.setopt DebugLevel 5 # To debug connections to the fedration
# (5 Dump, 4 Debug, 3 Error, 2 Warning, 1 Info)
pss.trace info
pfc.trace dump
```

A.5.1.2. Siteconf vs XrdClProxyPlugin

For using the XRootD (Caching) Proxy with official production jobs, essentially two possible ways to set this up are available. The first option, as intended by XRootD, is to use the `XrdClProxyPlugin`. This plugin allows to address a proxy server with every transfer by simply adding the proxy address to the remote file request of the client. An example configuration is printed in Listing A.8. As an alternative, the proxy address can be set in the SITECONFIG centrally. By this, every official CMS job is always appending the proxy to the address of the remote origin.

Listing A.8: Basic XrdClProxyPlugin config.

```
url=root://*
lib =libXrdClProxyPlugin-5.so
enable=true
XROOT_PROXY=root://123.45.678.910:1094//
#XROOT_PROXY_EXCL_DOMAINS=<add domains to exclude, if desired>
```

Both ways have different benefits and disadvantages. While the plugin approach is more flexible – it can in principle be set per pilot by mounting config and plugin into the container – it unfortunately has two significant disadvantages: First, it is used for read and write attempts. As a consequence, the stage-out of job logs stops working, because the write attempt is then sent from the proxy to EOS and not from the job, hence with different credentials. Unless the X509 proxy utilized by the XRootD Proxy has write permission (which is rather unlikely), this is a major issue, since no proper credential forwarding is supported (yet). Second, if the XRootD Proxy itself has problems, there is no way to circumvent the proxy server and, for example, contact the redirector directly.

Therefore, it was decided to set the XRootD Proxy's address directly as prefix in the site configuration. The downside of this approach is that the entire site configuration has to be changed to disable the prototype, which is only applied for new nodes/job pilots. Nevertheless, the intended fallback mechanism can be employed and the failure rate does not increase further when the Proxy itself has issues. This behavior is depicted in Fig. A.58. While for now this approach is preferable, in the future – especially with tokens – a credential forwarding may be possible, which would make the plugin approach more feasible due to its more flexible concept in general.

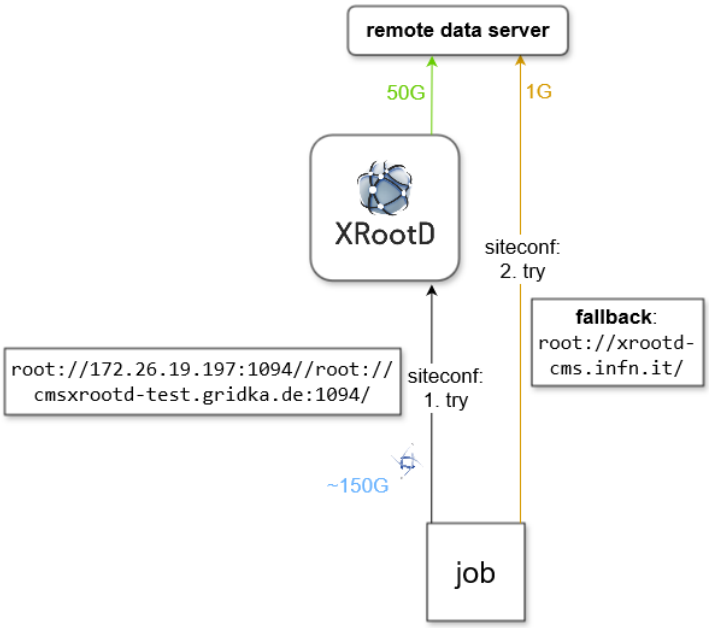


Figure A.58.: Visualization of the client and proxy behavior when the XRootD (Caching) Proxy is configured via the site's general configuration (SITECONF). In this case, the first try is addressed to the proxy server. If it fails, the fallback is triggered which sends the request directly to the European Redirector.

A.5.1.3. Security Considerations

Especially on HPC centers, security is extremely important. When running the presented prototype, however, we decided against the additional authentication and authorization features of XRootD. This decision is well-justified since despite other users of the HPC cluster could contact the XRootD (Caching) Proxy in principle, they would anyway need a valid grid proxy (voms proxy) or authentication token to identify. Therefore, the an explicit authentication process is not necessary. However, in Listing A.9, an exemplary configuration is provided.

Listing A.9: Exemplary configuration for authentication and authorization.

```
# Load the security plugin
xrootd.seclib libXrdSec.so

# Enable
ofs.authorize 1

# Tokens
sec.protocol /usr/lib64 ztn
# GSI
sec.protocol /usr/lib64 gsi -d:1 -crl:0 -vomsfun:default \\
  -vomsat:extract -gmapopt:nomap

# Host Certificates/TLS
xrd.tlsca certdir /etc/grid-security/certificates
xrd.tls /<path-to-cert.pem> /<path-to-key.pem>
xrootd.tls capable all

ofs.authlib ++ libXrdAccSciTokens.so config=/etc/xrootd/scitokens.conf
ofs.authlib ++ libXrdMacaroons.so

# Logging
scitokens.trace all
acc.audit deny grant
```

A.5.2. Data Access via the Prototype

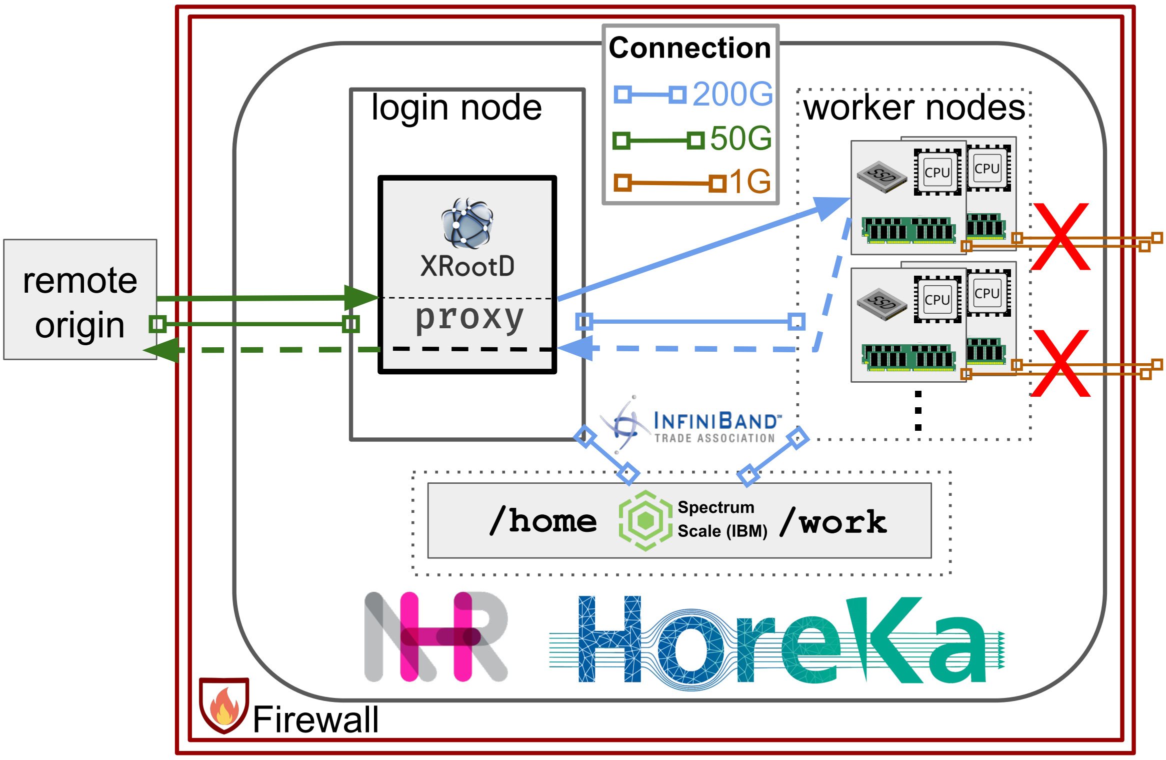


Figure A.59.: The figure shows schematically the data access process with an XRootD Proxy deployed on the login node of the cluster. Instead of fetching the data with the own external links (brown with red X), a client speaks to the proxy server on the login node via InfiniBand (dashed, blue arrow). The proxy then takes over the client's role and requests and transfers the data from a remote origin (green arrows). As the last step, all transferred data is provided to the initial client by the proxy via the internal networks (blue arrow).

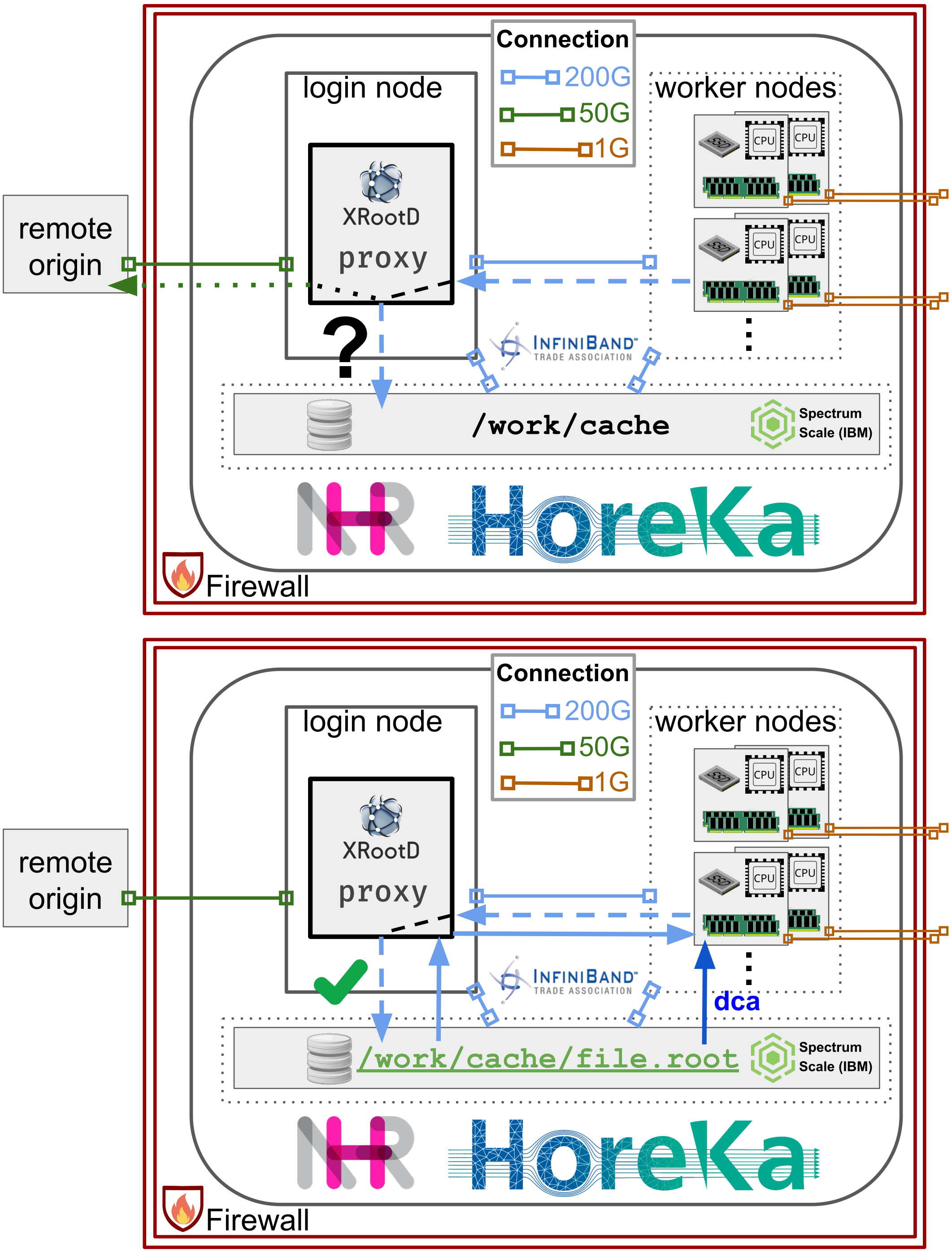


Figure A.60.: Data access via XCache/XBuffer. The same applies as in Fig. A.59. But additionally, the caching feature is enabled – optionally with prefetching. In this case, the XRootD Caching Proxy utilizes the parallel filesystem of the cluster. When a client requests a file from the XRootD server on the login node, it first checks if the file is available on the local storage, indicated with the question mark in the top figure. If not, the file is fetched from remote (dashed, green arrow) and the transferred data is cached on-the-fly (dashed, blue arrow) while the proxy provides it to the client via the internal network. In case a requested block is available (bottom), it is served from the local storage via the fast internal network (light blue arrow). Additionally, if a file is fully available on the cache, it can also be served via the direct cache access feature of XRootD (blue arrow). This allows the client to fetch the file directly from the storage without the detour via the cache. It was observed that this can accelerate the transfers by a factor of 100.

A.6. Evaluation of the Second Phase

A.6.1. Total Statistics for the second Phase

Table A.7.: Total statistics for the entire second part of the pilot phase (November 2023 until end of 2024). Here, all workflows of type *Production** and *Processing** are considered. RWTH-HPC did only run test workflows during most of this period and is therefore excluded from the evaluation. In comparison to the initial phase (see Table 5.5), HoreKa was able to catch up, while the other sites have deteriorated.

Measure	GridKa	RWTH	TOpAS	HoreKa
Number of Jobs	3.49M	2.41M	201K	463K
Fraction of *Processing**	34.1%	12.6%	33.1%	10.0%
Fraction of *Production**	65.9%	87.4%	66.9%	90.0%
$\text{CpuEff}_{\text{tot}}$ (Eq. (5.1))	79.5%	78.6%	76.3%	68.5%
$\text{CpuEff}_{\text{tot}}$ (*Processing**)	88.3%	82.3%	78.0%	68.7%
$\text{CpuEff}_{\text{tot}}$ (*Production**)	75.1%	78.4%	76.0%	68.4%
Failure Rate (Eq. (5.8))	5.8%	14.9%	6.5%	17.6%
$\text{CpuEff}_{\text{tot}}^{\text{corr}}$ (Eq. (5.10))	77.6%	71.2%	73.7%	61.2%
$\text{CpuEff}_{\text{tot}}^{\text{corr}}$ (*Processing**)	86.9%	73.6%	72.7%	58.0%
$\text{CpuEff}_{\text{tot}}^{\text{corr}}$ (*Production**)	72.8%	71.1%	73.8%	61.5%
CommittedCoreHr*	76.6M	31.9M	2.41M	5.05M
Inefficiency Loss I_{agg} (Eq. (5.17))	14.0M	5.62M	531K	1.23M
Inefficiency Loss $\text{I}_{\text{agg}}^{\text{rel}}$ (Eq. (5.18))	18.3%	17.6%	22.1%	24.3%
ComputeTimeLoss C_{agg} (Eq. (5.14))	3.18M	3.55M	102K	729K
ComputeTimeLoss $\text{C}_{\text{agg}}^{\text{rel}}$(Eq. (5.13))	4.1%	11.2%	4.3%	14.5%
Total Loss T^{rel} (Eq. (5.21))	22.4%	28.8%	26.4%	38.8%

A.6.2. Classic Quantities

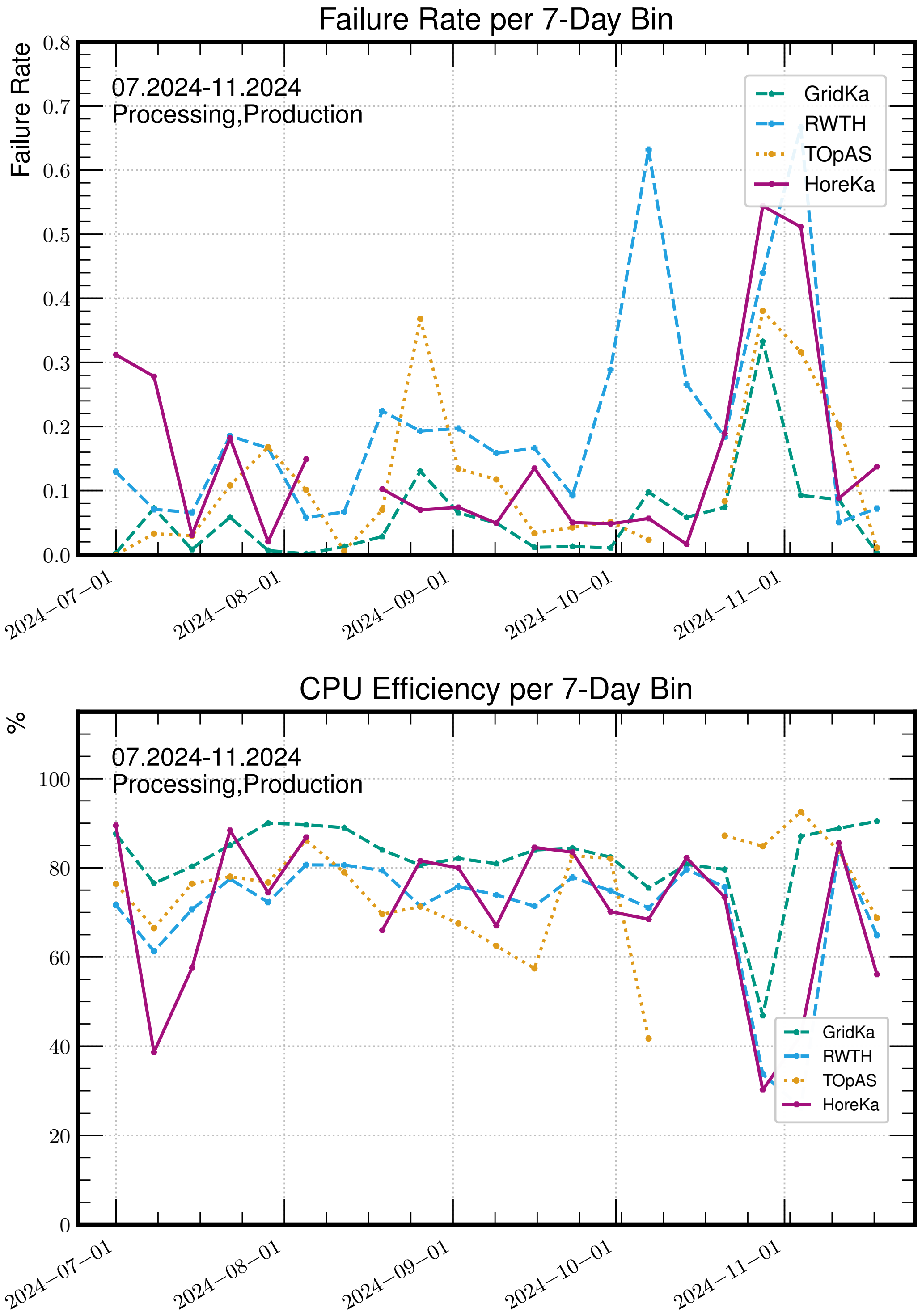


Figure A.61.: Failure rate and CPU efficiency for all sites during the third part of the pilot phase. Within this period, the prototype at HoreKa was deployed as an XRootD Proxy without the caching features. With this configuration, HoreKa is very comparable to the other sites. A more granular overview of the same time interval is given in Fig. A.62.

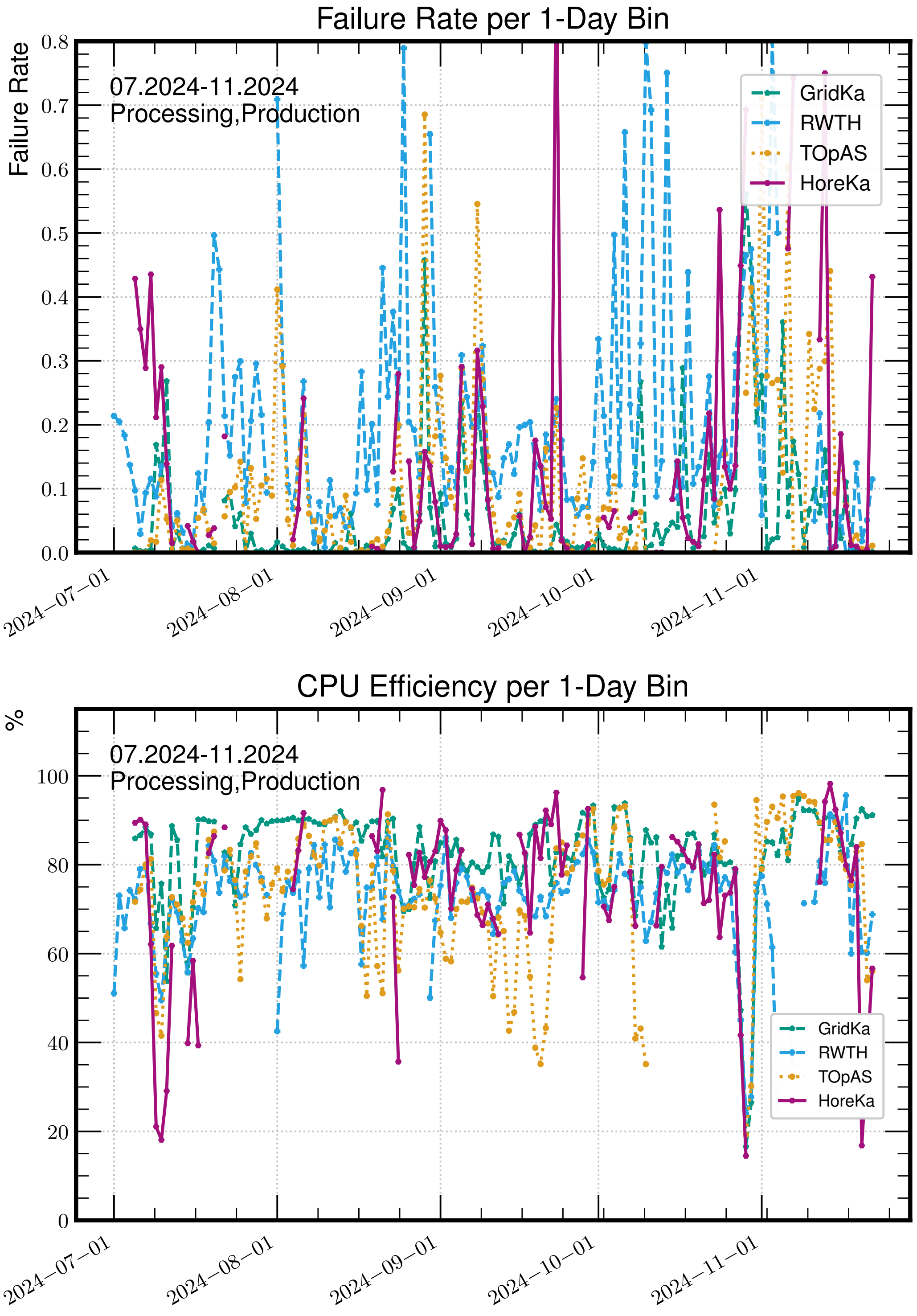


Figure A.62.: The figures show the same as Fig. A.61, but with 1-day bins instead to provide a more fine-granular overview. Most of the time, the HPC center is performing absolutely comparable.

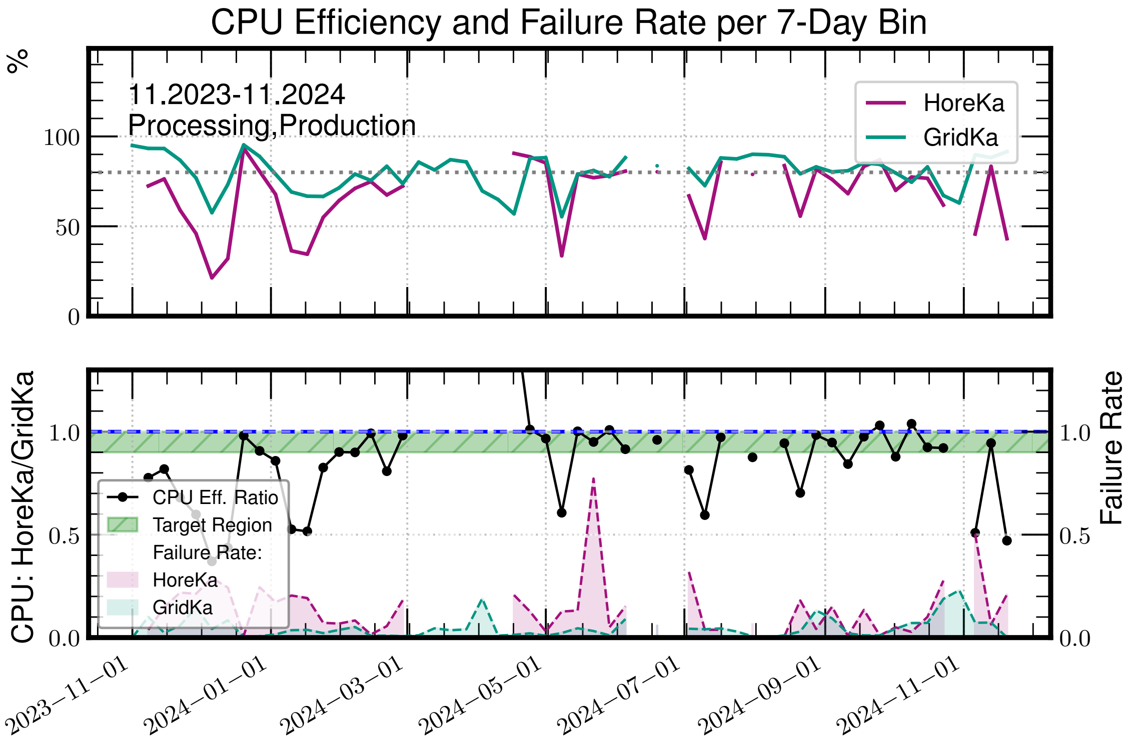


Figure A.63.: Direct CPU efficiency and failure rate comparison with GridKa. Most parts of the second and the third phase are very comparable.

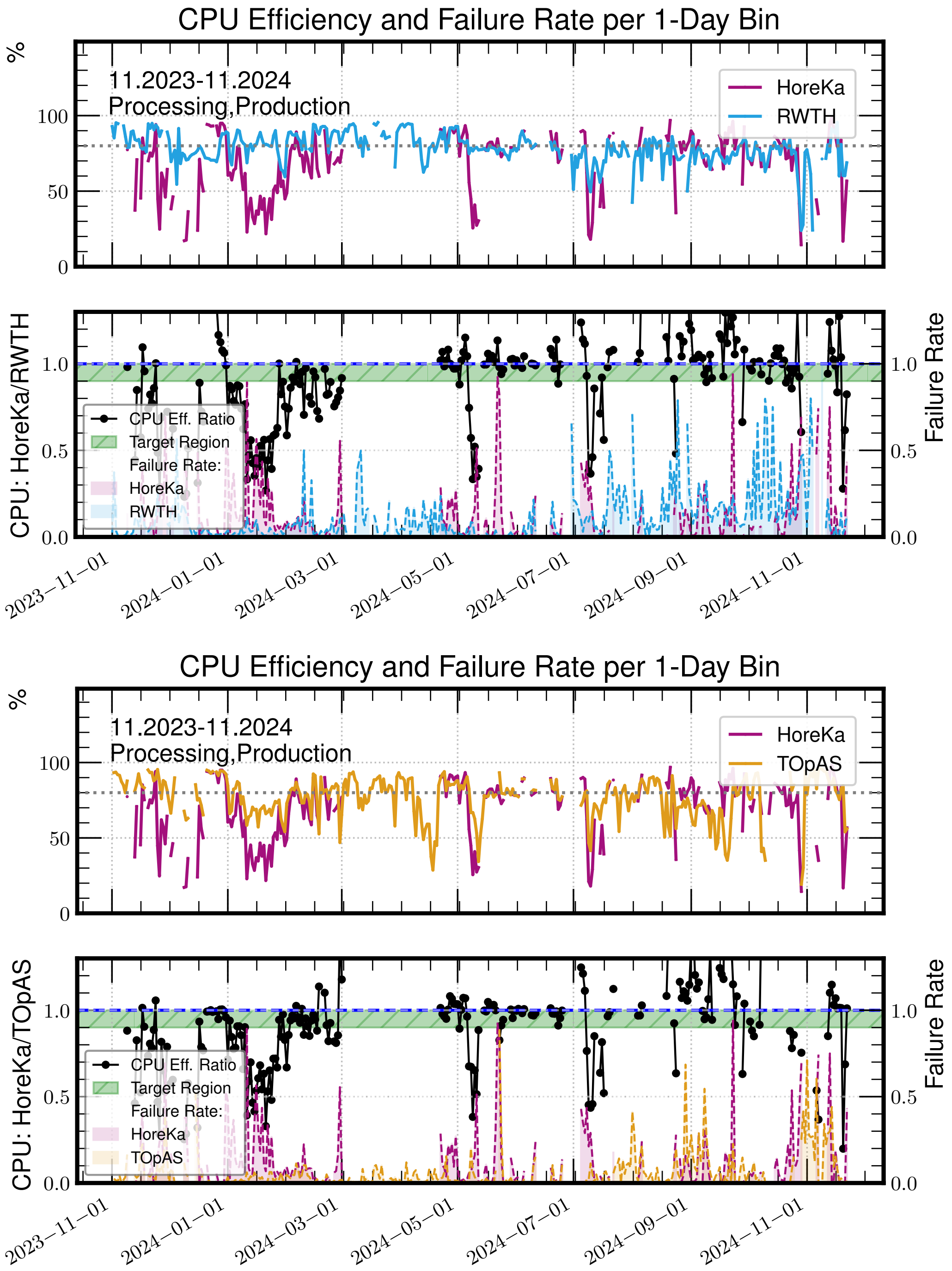


Figure A.64.: CPU efficiency and failure rate of HoreKa in direct comparison to the Tier-2 in Aachen (top) and the Tier-3 at KIT (bottom) with a time resolution of one day. As both figures show, during the second and third phase, the CPU efficiency is mostly in the targeted region. Only some few days are significantly below. Overall, the performance of HoreKa is very comparable.

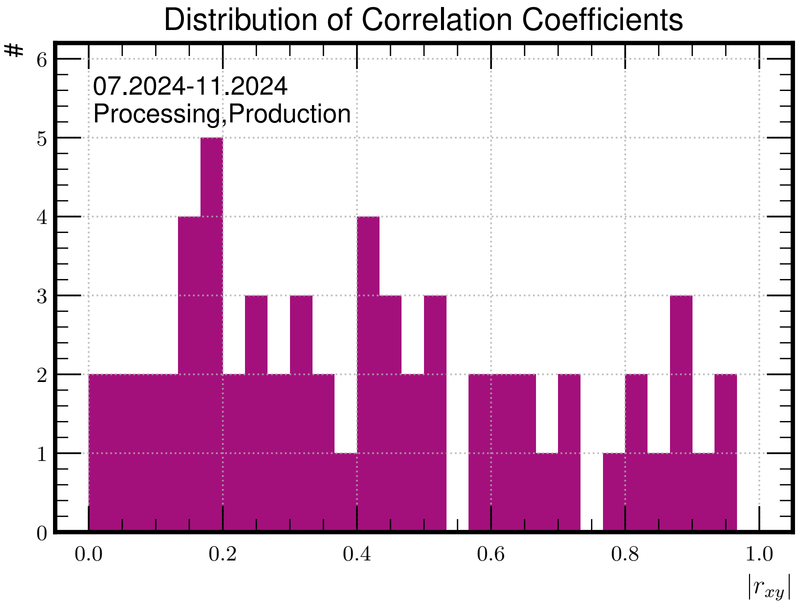


Figure A.65.: Correlation factors per workflow for the Proxy Phase. With a value of $|r_{xy}| = 0.27$, no direct correlation between the read time fraction and the CPU efficiency is observed anymore. This indicates that the optimizations together with the prototype setup efficiently mitigated the data access bottleneck at HoreKa.

A.6.3. New Measures

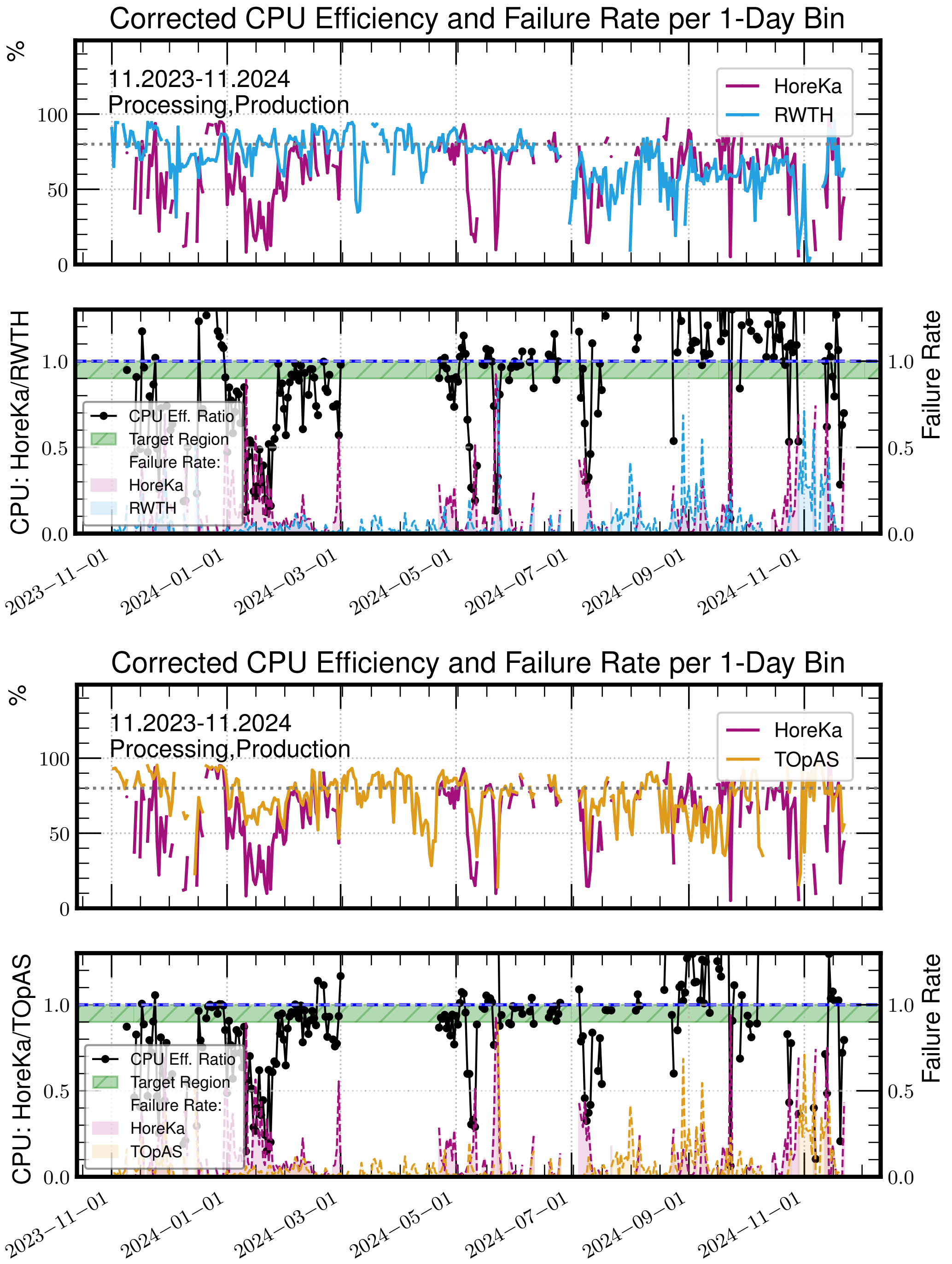


Figure A.66.: Here, the same is presented as in Fig. A.64, but with the corrected CPU efficiency instead. Since the failure rates are comparable, the impact on the ratio between the sites is small. Again, during the second and third phase, the CPU efficiency is mostly in the target region. Only some few days are significantly below. Overall, the performance of HoreKa is very comparable.

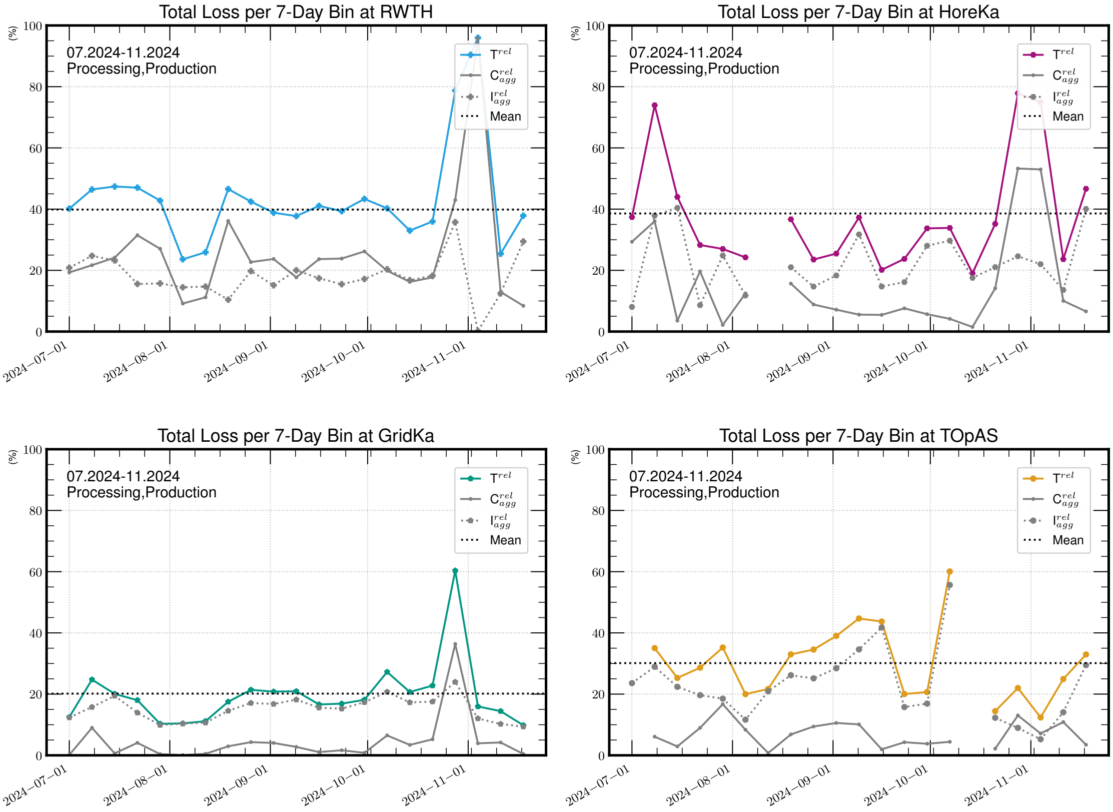


Figure A.67.: Overview of the individual compositions of the relative Total Loss per site. As the comparison to all other sites shows, the utilization of HoreKa improved and is now very competitive. Even in the beginning of November, when the CMS AAA data federation had problems, the impact of the job failures at the HPC center is high, but comparable to the Tier-2 in Aachen.

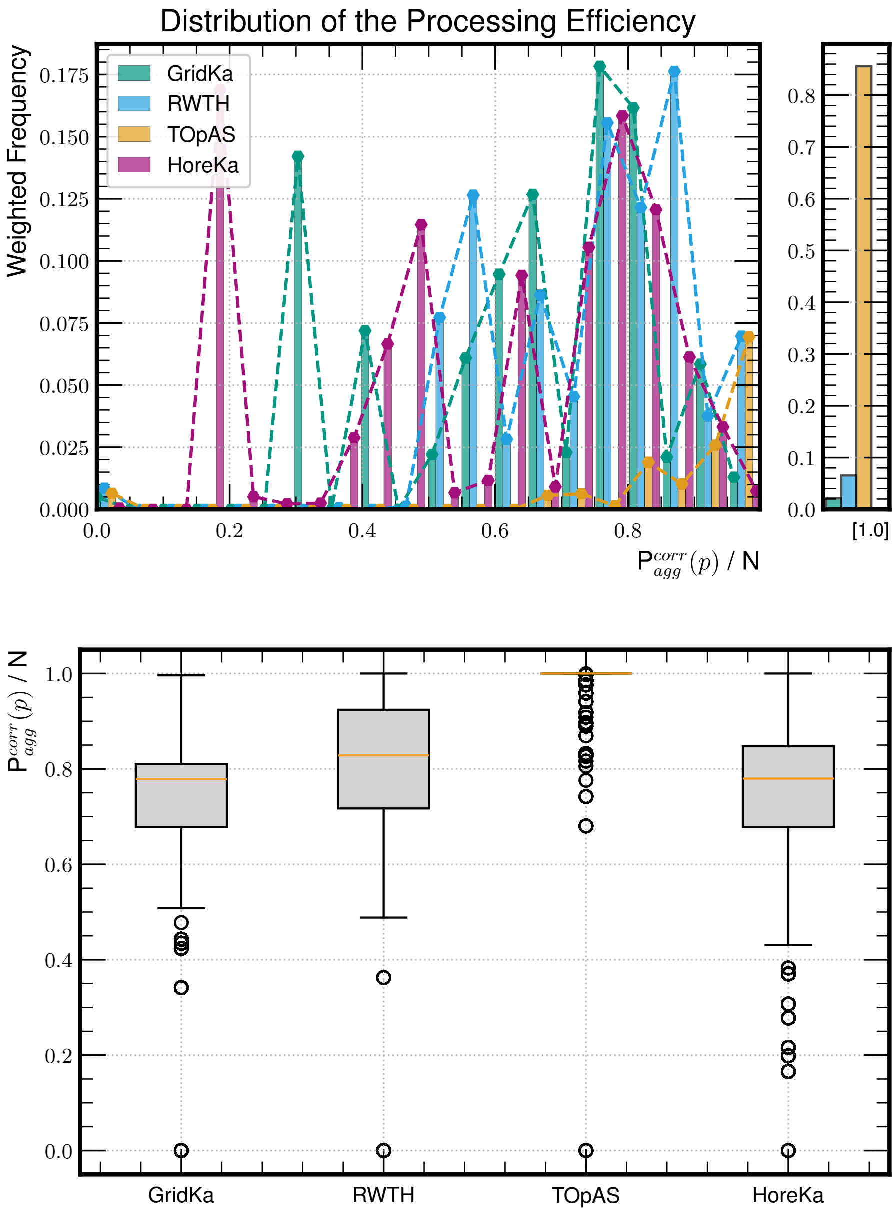


Figure A.68.: Distribution of the corrected processing efficiencies aggregated per workflow for all reviewed sites. Additionally, workflow weights are applied. The according box plots are depicted below.
With the additional weights, the distribution of the processing efficiencies per workflow slightly changes, but the overall result is very similar, as visible in the box plot. HoreKa is performing very comparable to the WLCG sites with the Proxy setup enabled.
In Fig. 6.11, the unweighted versions are shown.

B. Supporting materials

Without data, you're just another person with an opinion.

(W. Edwards Deming)

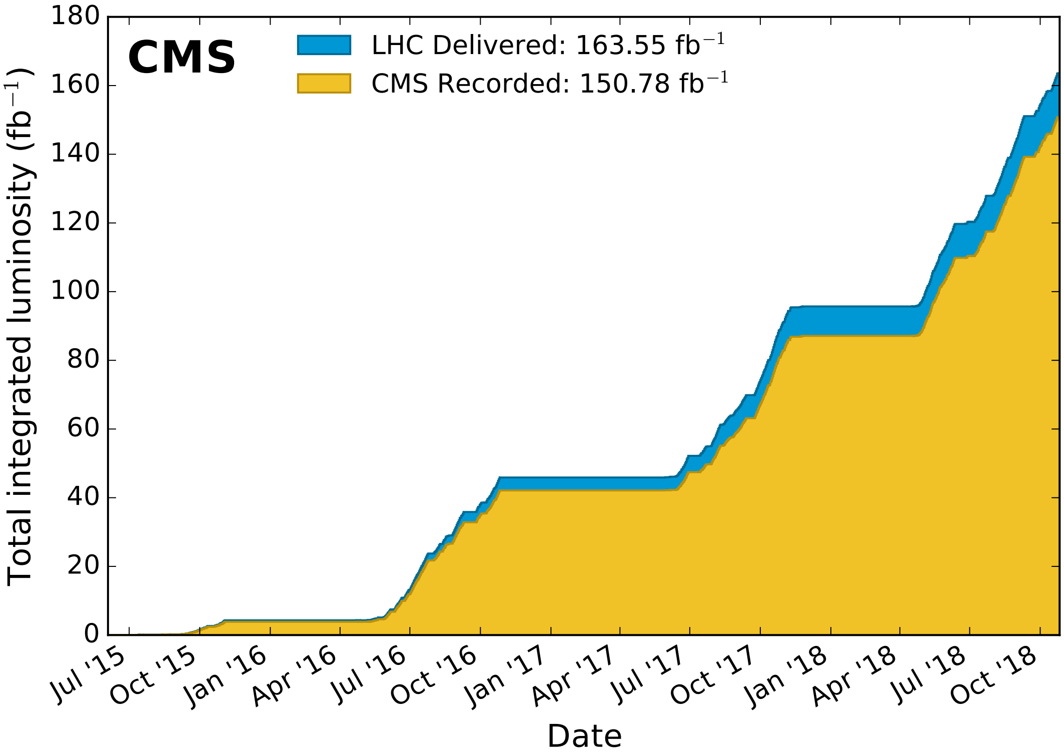


Figure B.1.: Total delivered integrated luminosity by the LHC in comparison to the recorded data at CMS for Run 2 (13 TeV center-of-mass energy). The figure only includes proton-proton collisions. Within this thesis, the entire dataset for Run 2 was evaluated. For further details, see e.g. [192].
Source: [77]

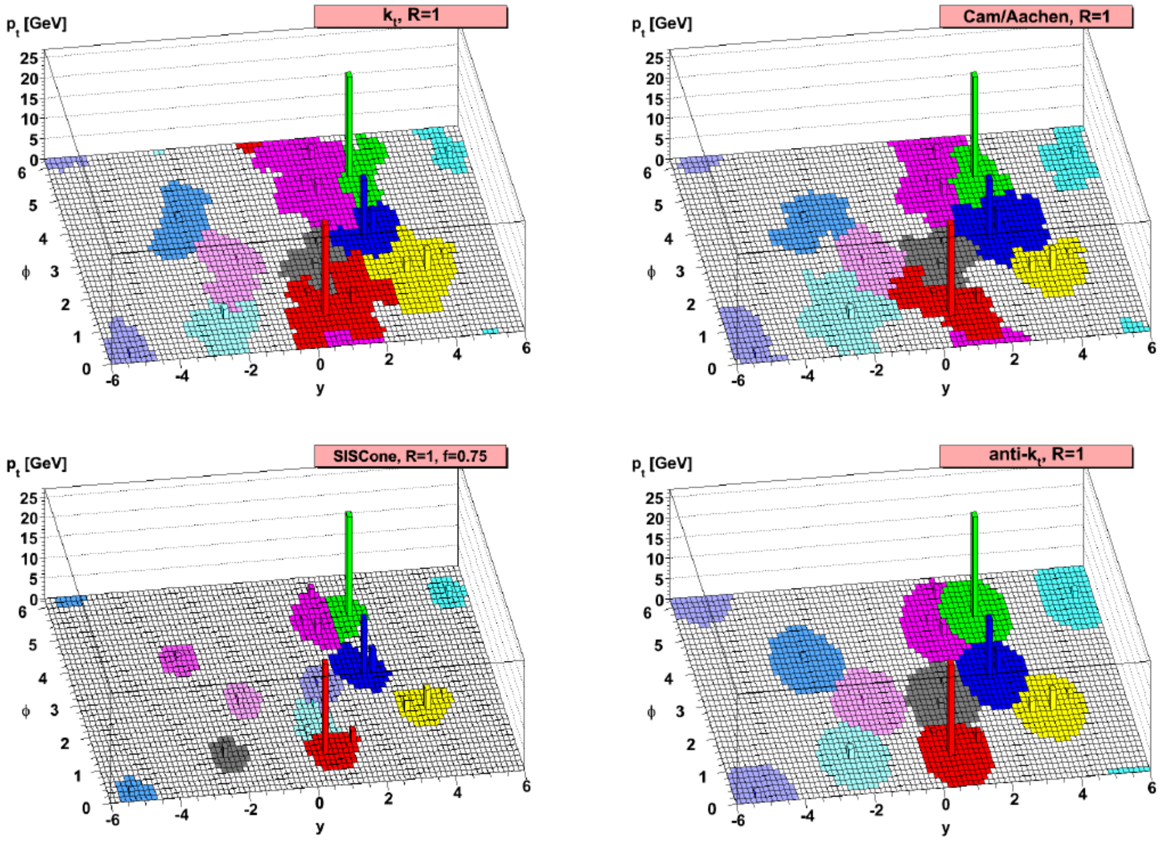


Figure B.2.: The figure compares different jet clustering algorithms. As clearly visible, the anti-k_t algorithm provides a good separation and very regular, cone-shaped jets centered around the most energetic particle.
Source: [44]

Table B.1.: The default XRootD plugins with their purpose. The most relevant ones for this work are highlighted. For the full reference, see [127]. Available plugins that are shipped with the default installation can be listed with `xrdpinls`.

acc	Access control (authorization)
cms	**Cluster Management Services**
dig	The digFS built-in file system
frm	File Residency Manager
http	HTTP protocol
ofs	**Open File System: coordination of acc, cms, and oss components**
oss	**Open Storage System: file system implementation**
pfc	**Proxy File Cache: implementation of caching**
pss	**Proxy Storage Service: specialized oss plugin for caching**
sec	Security and authentication
xrd	Extended Request Daemon
xrootd	The xrootd protocol implementation
all	Applies the directive to all of the above components.

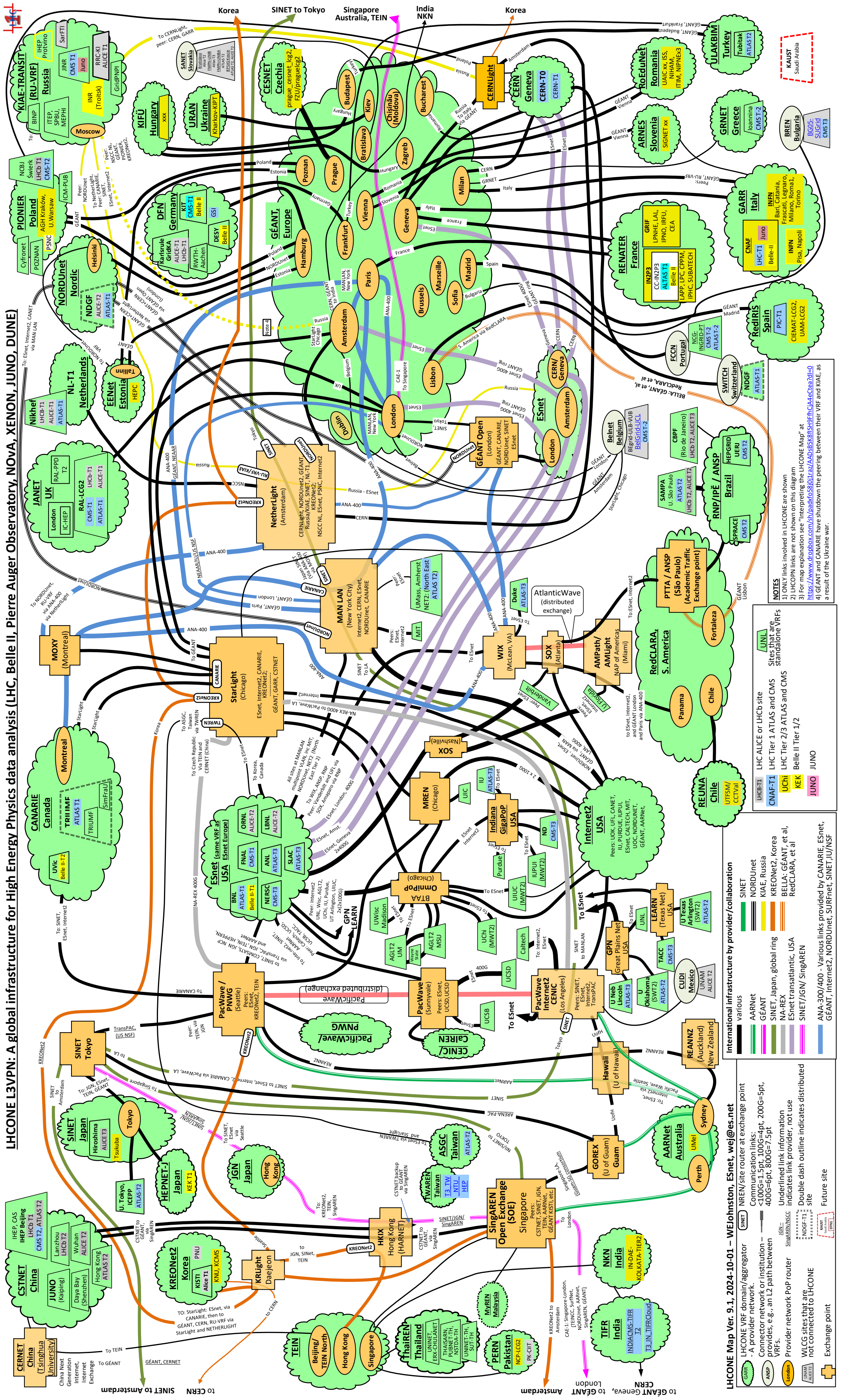

LHCONE L3VPN: A global infrastructure for High Energy Physics data analysis (LHC, Belle II, Pierre Auger Observatory, NOvA, XENON, JUNO, DUNE)
LHCONE Map Ver. 9.1, 2024-10-01 – WEJohnston, ESnet, wej@es.net
GÉANT, Europe
ESnet, USA
CANARIE, Canada
Internet2, USA
StarLight (Chicago)
MAN LAN (New York City)
WIX (McLean, VA)
NetherLight (Amsterdam)
CERNLight
CERN, Geneva
SINET, Japan
KREONet2, Korea
CSTNET, China
TEIN
AARNet, Australia
RedCLARA, S. America
RNP/IPÊ / ANSP, Brazil
REUNA, Chile
NORDUnet, Nordic
GARR, Italy
RENATER, France
DFN, Germany
JANET, UK
RedIRIS, Spain
PIONIER, Poland
Russia
SingAREN Open Exchange (SOE) Singapore
HKIX Hong Kong (HARNET)
PacificWave / PNWG (Seattle)
PacWave (Sunnyvale)
AMPath/AMLight NAP of Americas (Miami)
AtlanticWave (distributed exchange)
PacificWave (distributed exchange)
International infrastructure by provider/collaboration
NOTES
1) ONLY links involved in LHCONE are shown
2) LHCOPN links are not shown on this diagram
3) For map explanation see "Interpreting the LHCONE Map" at https://www.dropbox.com/sh/padk0s8j0j1rav/AAD4b5K8fSq9rfhCA4eCtea?dl=0
4) GÉANT and CANARIE have shutdown the peering between their VRF and KIAE, as a result of the Ukraine war.

Figure B.3.: The LHCONE network as of October 2024 [193].

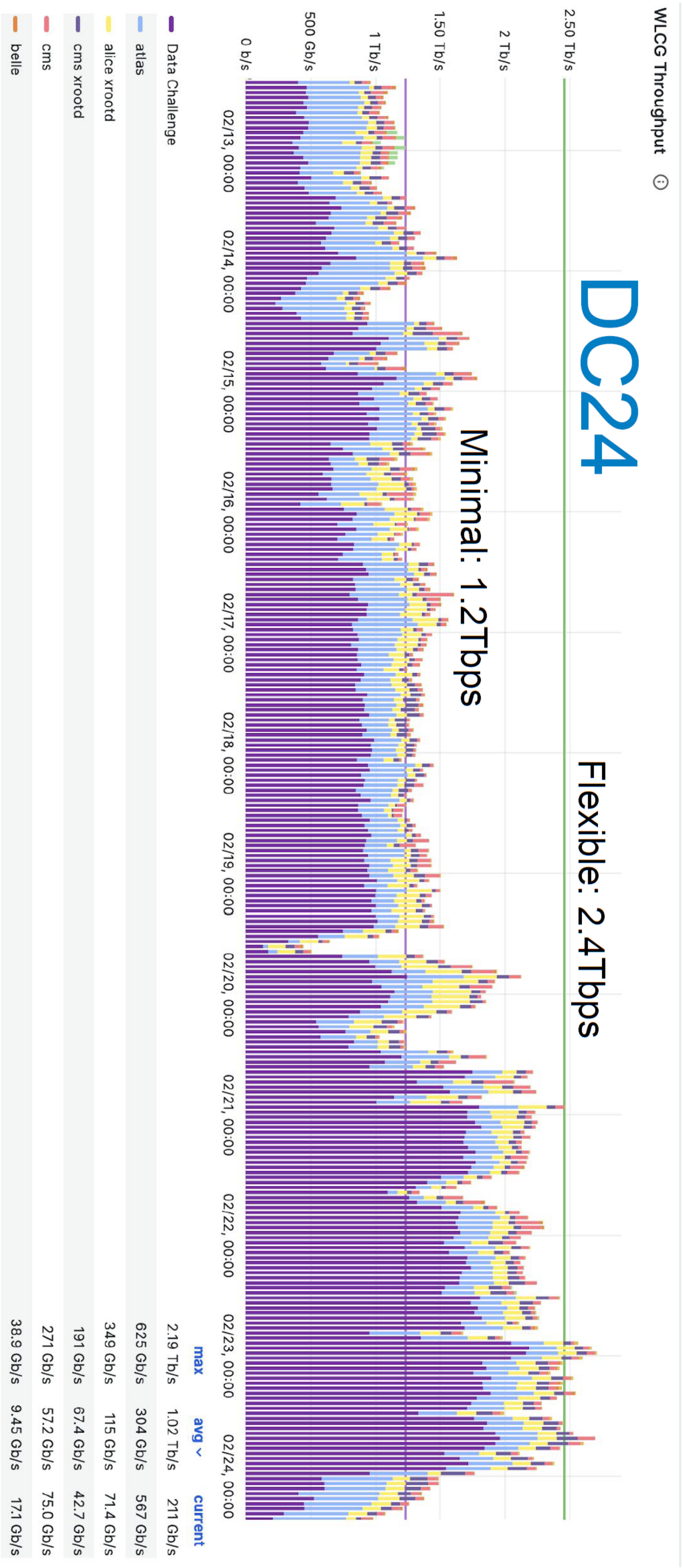
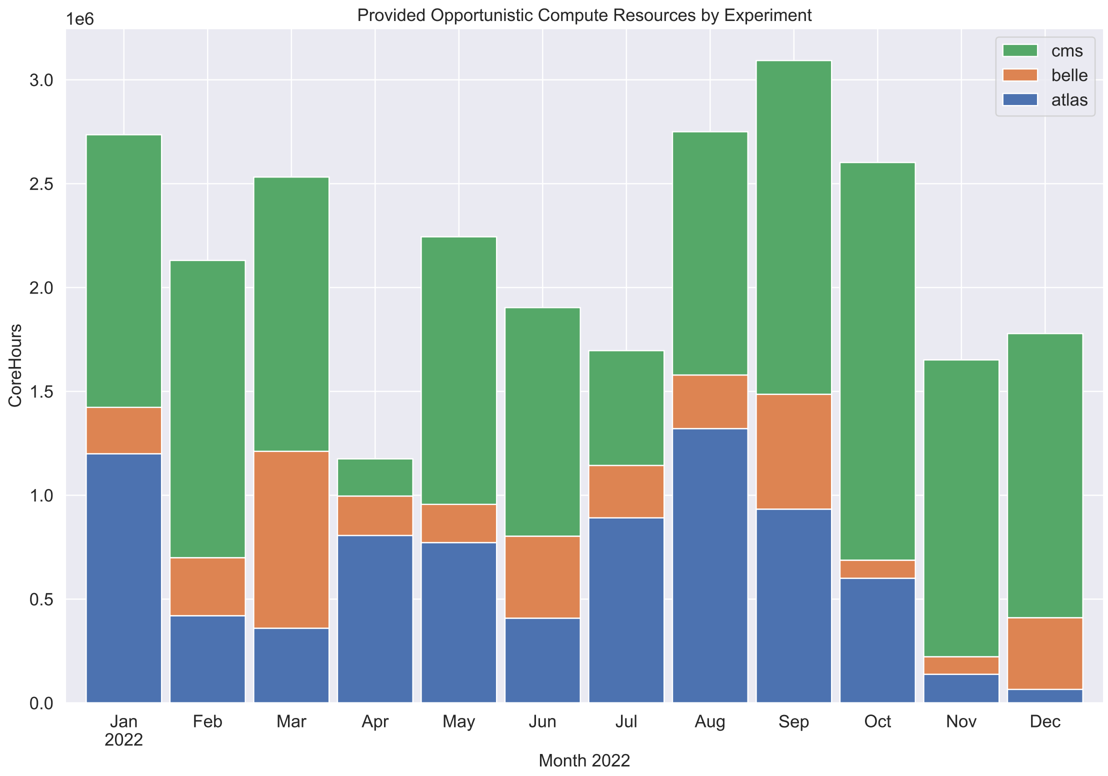


Figure B.4.: Results of the Data Challenges 2024. The minimal target was fully achieved. The flexible model was partly reached. Overall, the challenge is considered a success and shows that, particularly in terms of network, the is on a good way. *Taken from*: [194]



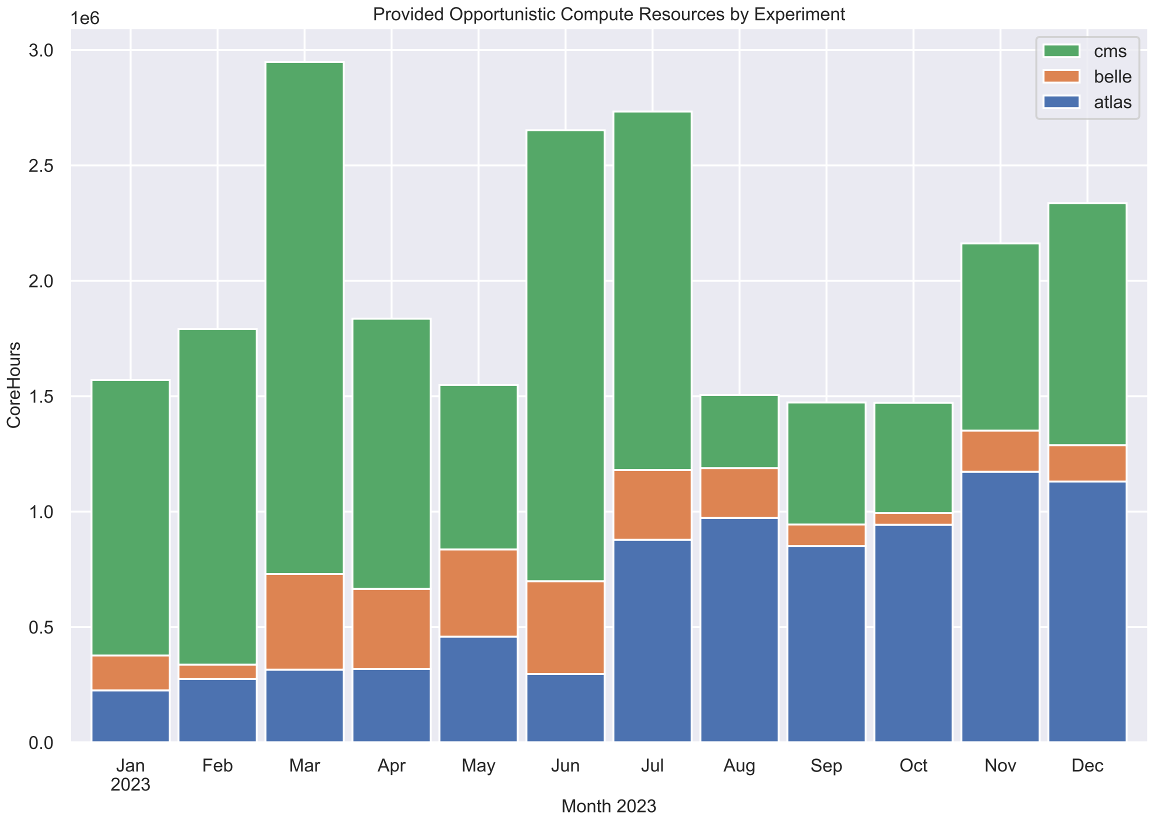


Figure B.5.: Opportunistic resources in Germany by experiment for the years 2022 and 2023. For both years, more than 1.5 million CPU core hours per month were provided on average to the experiments. The contribution in 2024 is shown in Fig. 4.19. *Plots provided by: Dr. Manuel Giffels*

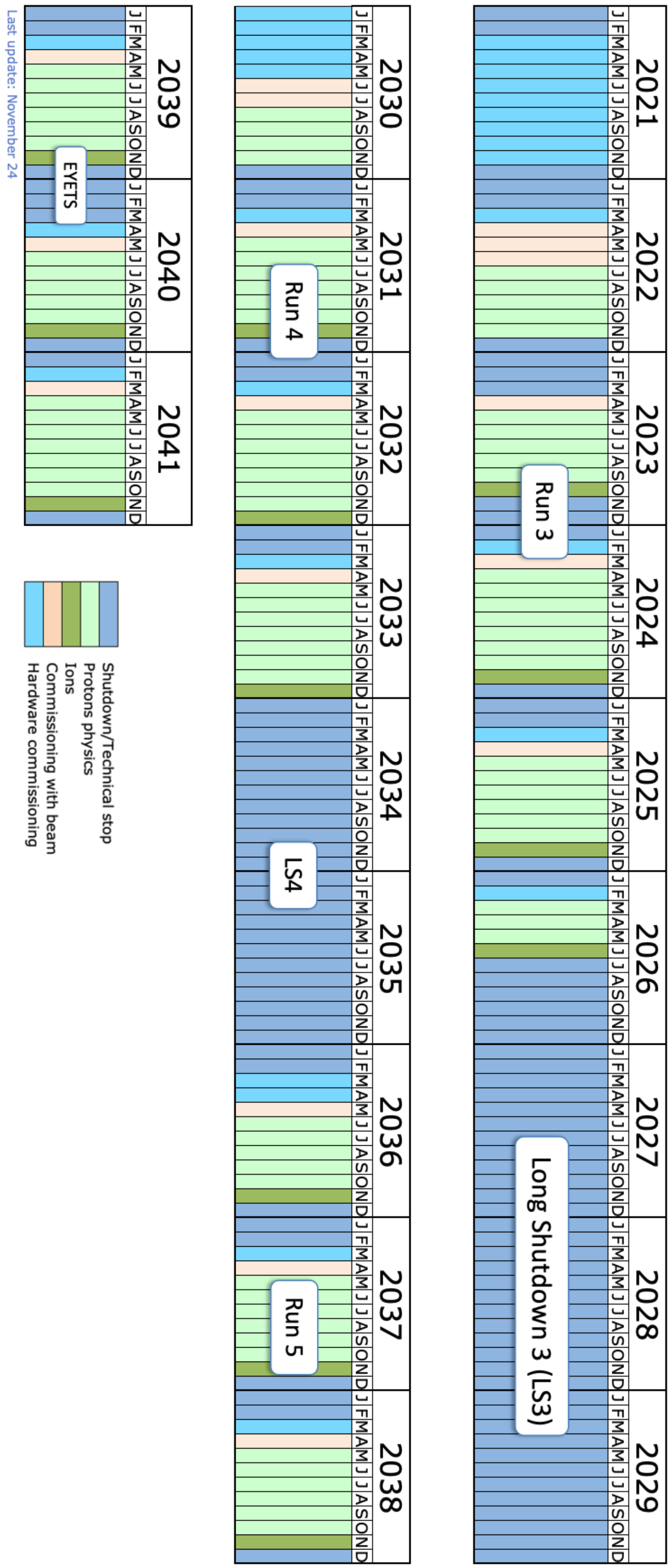
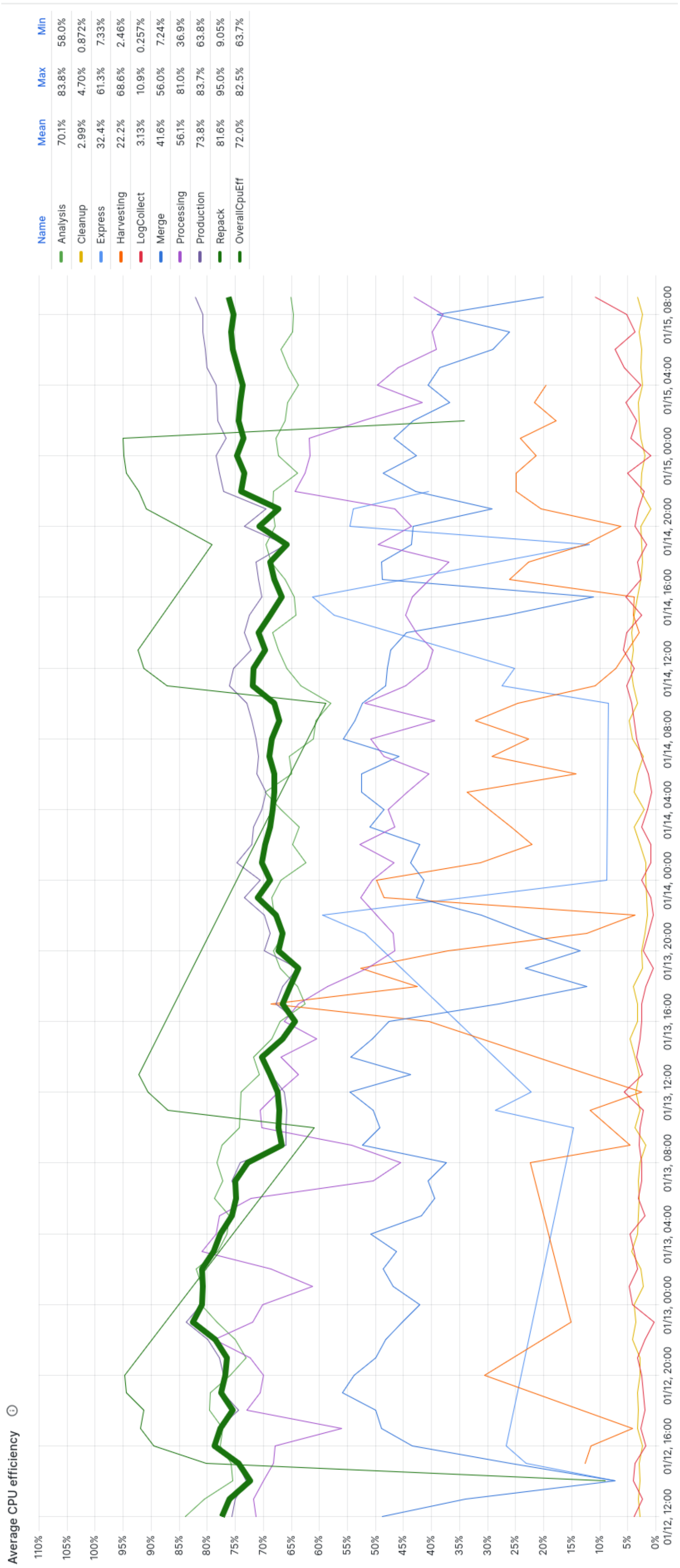


Figure B.6.: The planned LHC schedule for the next decades, as of November 2024.
Source: [195]



Name	Mean	Max	Min
Analysis	70.1%	83.8%	58.0%
Cleanup	2.99%	4.70%	0.872%
Express	32.4%	61.3%	7.33%
Harvesting	22.2%	68.6%	2.46%
LogCollect	3.13%	10.9%	0.257%
Merge	41.6%	56.0%	7.24%
Processing	56.1%	81.0%	36.9%
Production	73.8%	83.7%	63.8%
Repack	81.6%	95.0%	9.05%
OverallCpuEff	72.0%	82.5%	63.7%

Figure B.7.: The average CPU efficiency of CMS jobs in the global pool categorized by job types. The overview shows that rarely an efficiency above 80 % is achieved. *Source*: CMS MONIT/Grafana

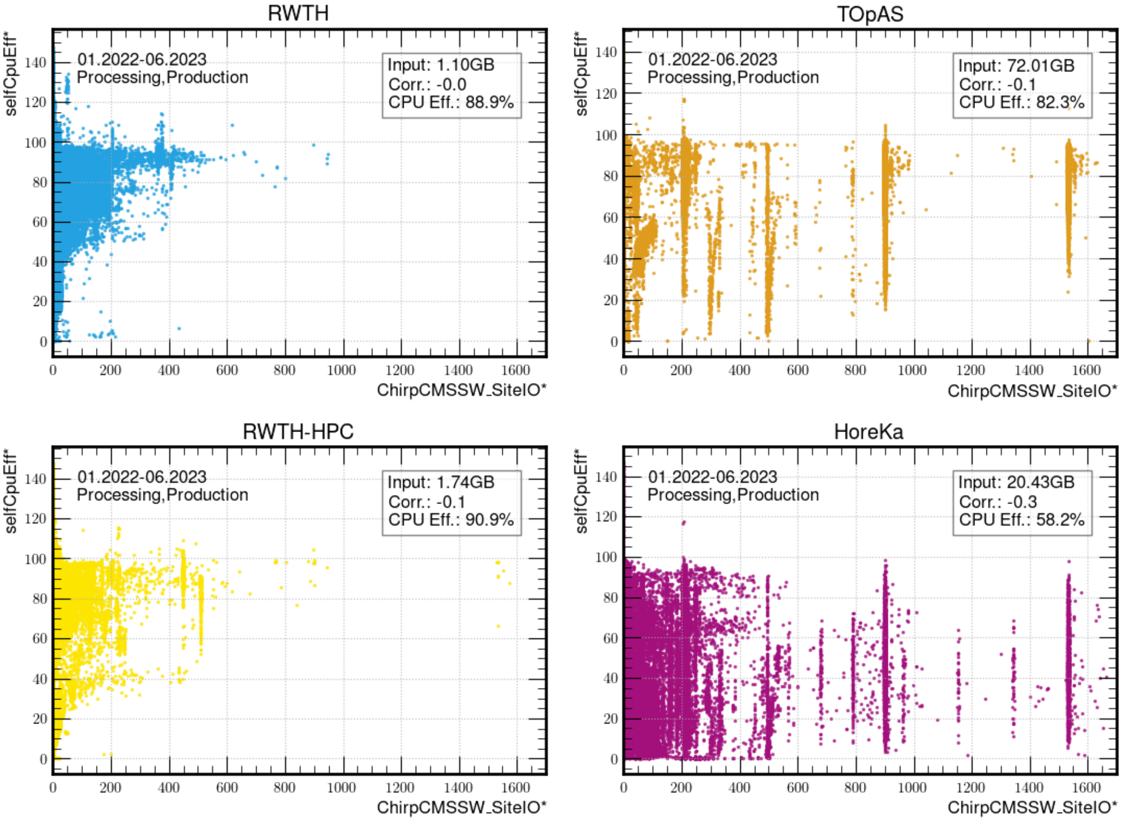


Figure B.8.: Correlation between the CPU efficiency and remote input data per site. For the comparison, only common workflows are considered that were executed at all investigated sites to create a better comparability. What is clearly noticeable is that first, no direct correlation exists for all sites, and second, there are significant differences between the Aachen and KIT sites. HoreKa and TOpAS seem to transfer significantly more data from remote (Input). GridKa looks very similar to TOpAS and is therefore neglected here.

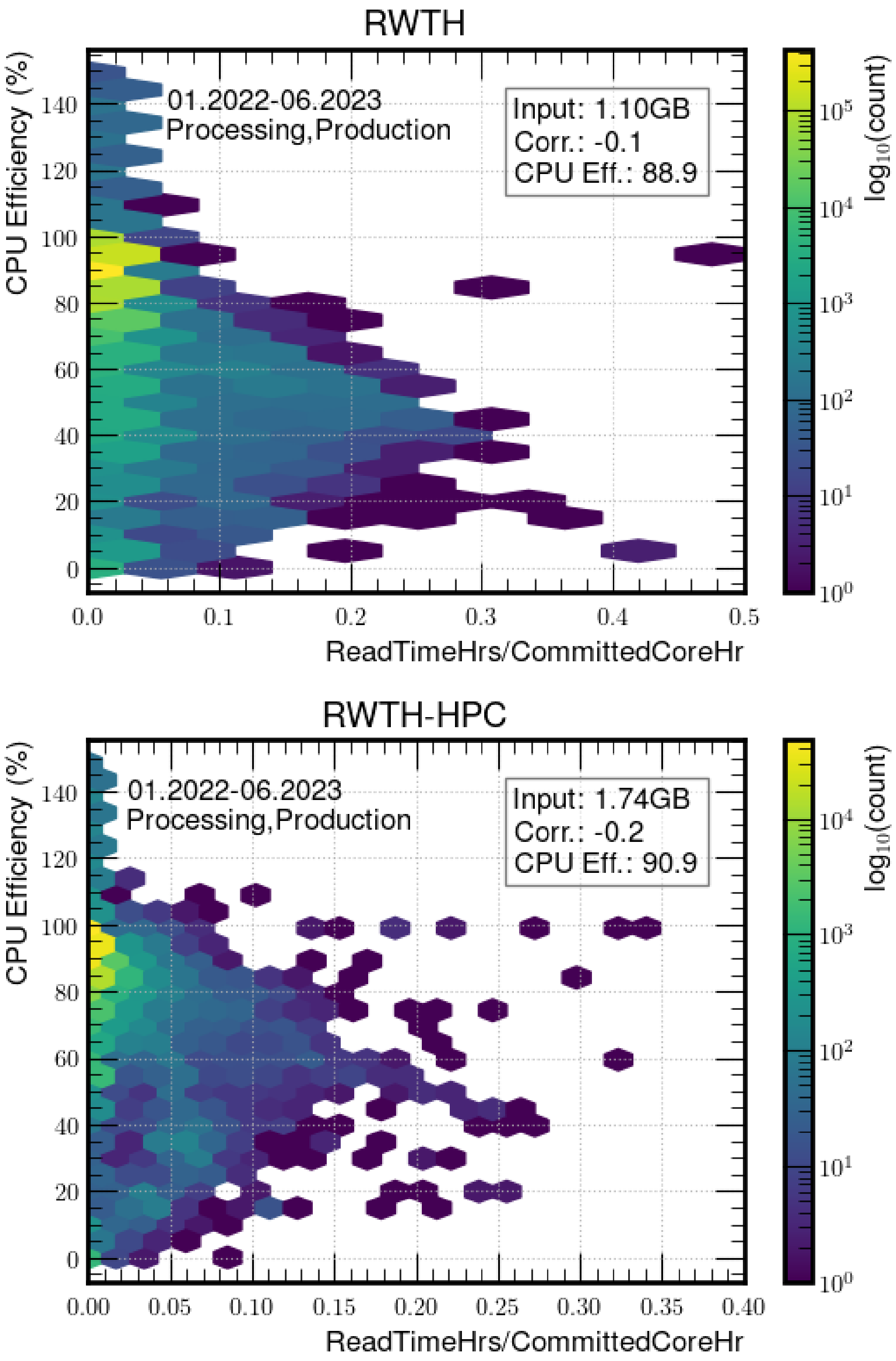


Figure B.9.: For RWTH and RWTH-HPC, the transferred amount of data is significantly less for the same workflows like investigated in Fig. 6.1. A possible explanation could be that the data was (partly) stored there and is therefore not accounted as remote data in the monitoring.

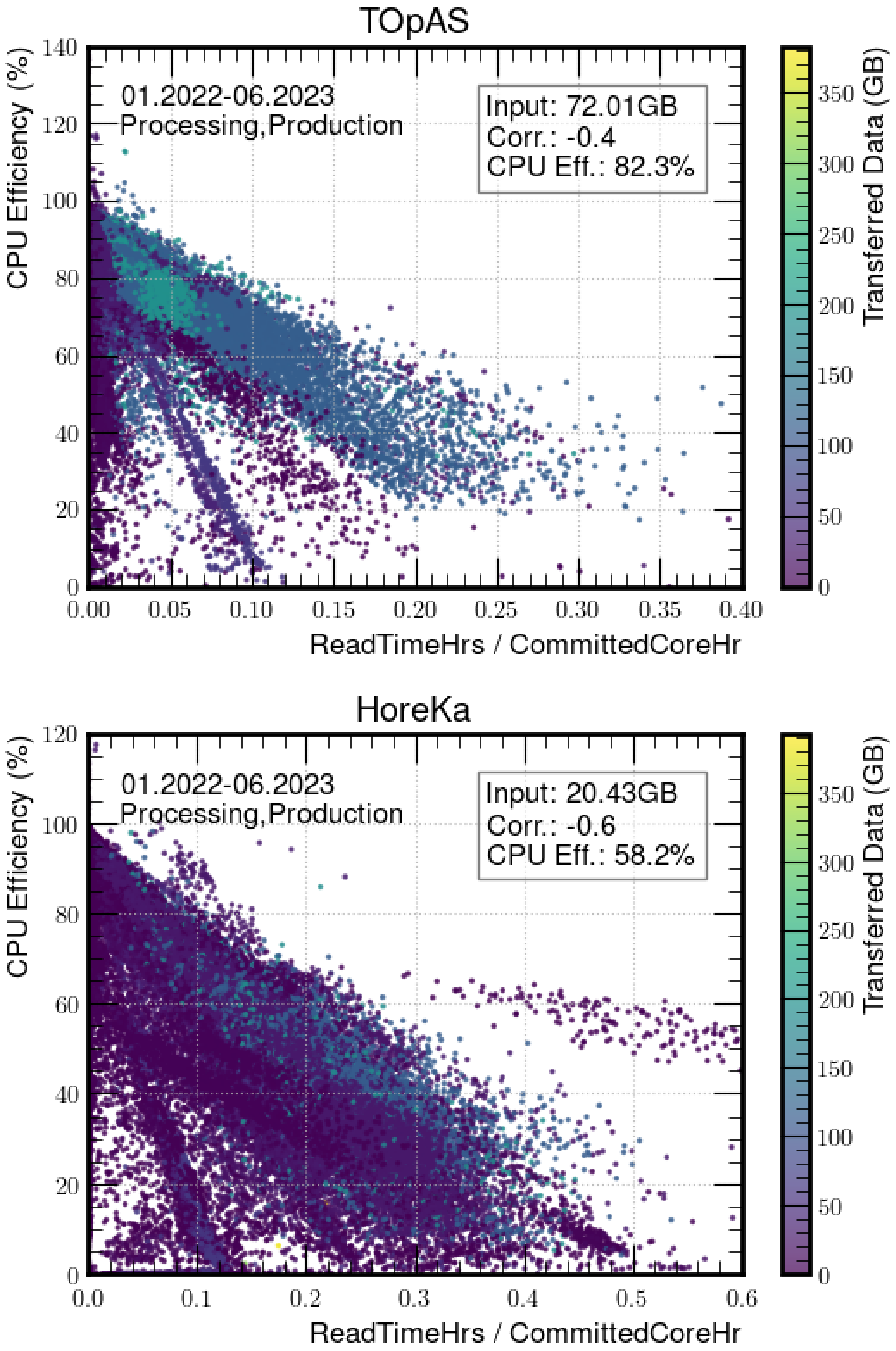


Figure B.10.: Correlation between the read time fraction and the CPU efficiency of jobs running at TOpAS (top) and HoreKa (bottom). TOpAS has overall more transferred data but still a smaller correlation. This indicates that HoreKa is significantly more I/O limited.

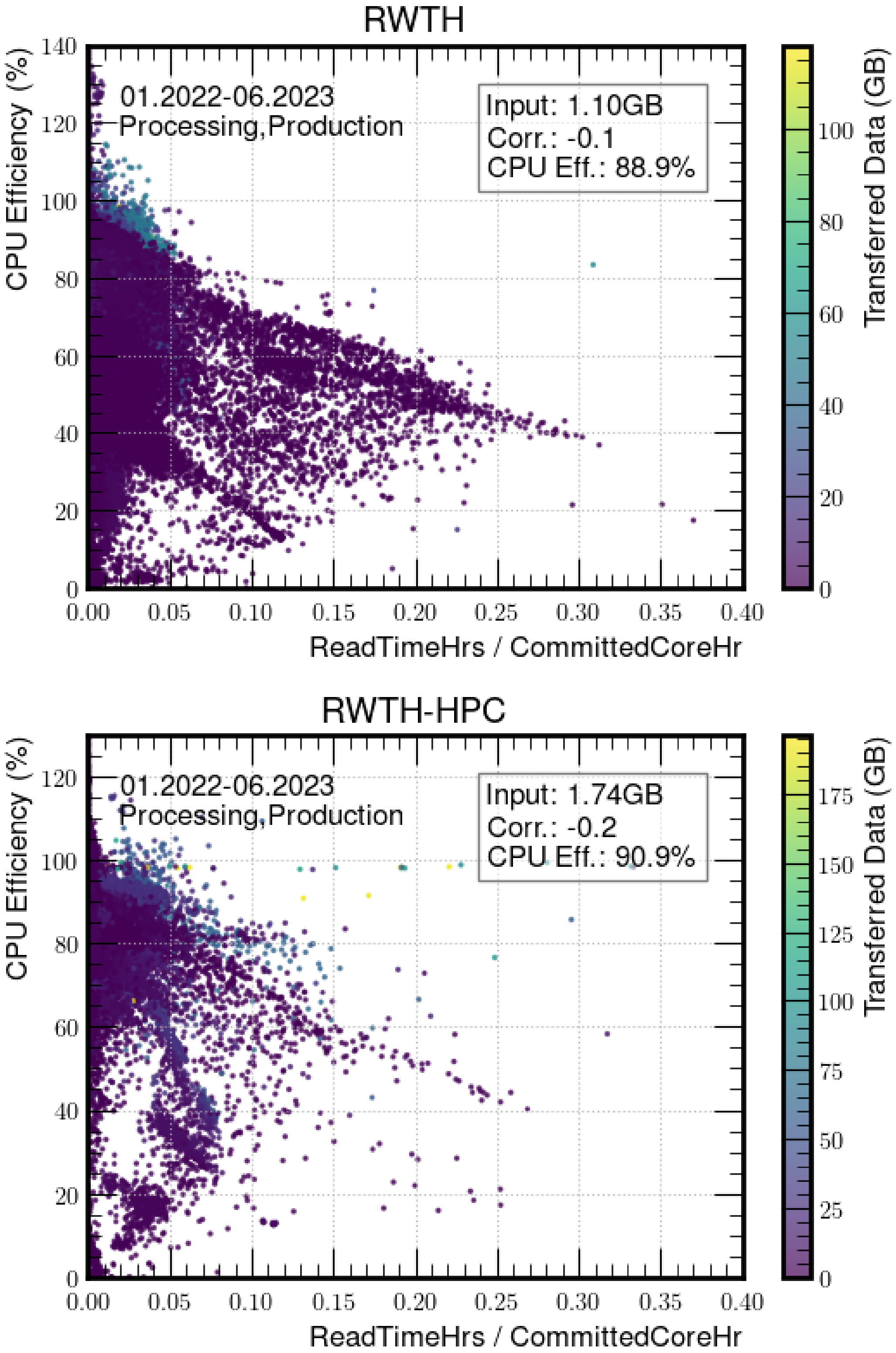


Figure B.11.: Analog distribution to Fig. B.10 but for the RWTH Tier-2 and CLAIX, the other NHR center. Remarkable is that both sites transfer significantly less remote data and therefore do not show a I/O limitation.

Table B.2.: Overview of all relevant CMS monitoring variables from the HTCondor monitoring index. They are mainly relevant for evaluating the integration of a grid resource (see Section 5.5).

Name	Definition	Description
ChirpCMSSW_SiteIO	$\sum$ all remote I/O	Remote transferred data
ChirpCMSSWTotalCPU	CPU time reported by CMSSW	in Seconds
CMS_JobType	CMS job types	Production, Processing, Analysis
CommittedCoreHr	CommittedWallClockHr x JobCpus	Total invested CPU time
CommittedWallClockHr	job runtime	Total accounted job runtime
CpuEff	$\frac{\text{CpuTimeHr}}{\text{CoreHr}}$	CPU usage ratio
CpuTimeHr	$\sum_{\text{cores}} \text{CpuTime}$	Time the CPU was not idle
EventRate	$\frac{\text{KEvents}}{\text{CommittedCoreHr} \times 3600}$	Per second
JobFailed	Flag for failed jobs	Type: boolean
CMS_JobType	Type of workflow	'Processing', 'Production'
KEvents	Number of events	Per Job
RequestCpus	Number of used CPUs	–
Type	General job classes	'production', 'analysis', 'test'
Workflow	Collection of similar CMS jobs	–

Table B.3.: The tables shows additional I/O-related information for all investigated sites for the initial phase of the integration. HoreKa has clearly the longest I/O times, indicating a limitation of the bandwidth.

Measure	GridKa	RWTH	TOpAS	HoreKa	RWTH-HPC
Number of Jobs	5.04M	1.57M	262K	504K	418K
Fraction of *Processing**	19.7%	8.2%	27.0%	42.0%	6.4%
Fraction of *Production**	80.3%	91.8%	73.0%	58.0%	93.6%
$\text{CpuEff}_{\text{tot}}$ (Eq. (5.1))	83.7%	89.1%	82.8%	58.0%	89.0%
$\text{CpuEff}_{\text{tot}}$ (*Processing**)	81.7%	82.6%	81.2%	48.6%	86.4%
$\text{CpuEff}_{\text{tot}}$ (*Production**)	83.9%	89.5%	83.1%	61.8%	89.1%
Failure Rate (Eq. (5.8))	8.3%	4.3%	5.6%	13.9%	40.1%
Remote Data per Job:					
*Processing**	17.4GB	2.1GB	11.4GB	6.9GB	3.2GB
*Production**	73.5GB	11.6GB	64.2GB	21.9GB	8.5GB
Average Runtime:					
*Processing**	2.75h	3.07h	1.59h	2.05h	2.91h
*Production**	6.57h	6.92h	4.62h	4.61h	4.82h
Average Readtime:					
*Processing**	0.22h	0.11h	0.08h	0.75h	0.15h
*Production**	0.70h	0.03h	0.34h	1.61h	0.06h
Readtime Fraction:					
*Processing**	1.9%	0.7%	1.1%	8.5%	1.1%
*Production**	3.1%	0.1%	2.0%	10.3%	0.3%

MISCELLANEOUS

Acknowledgments

Zunächst gilt mein herzlicher Dank meinen beiden Professoren, Günter Quast und Achim Streit, die mir diese Promotion ermöglicht haben. Ihr Vertrauen und ihre Förderung haben mir den Weg in die wissenschaftliche Welt geebnet.
Ein besonderer Dank gebührt meinem Betreuer, Manuel Giffels, sowie meinem Kollegen Nils Faltermann, die mir immer mit Rat und Tat zur Seite standen und auch abseits der Arbeit für unvergessliche Momente gesorgt haben. Gleiches gilt für meine Büro-Nachbarn, Cedric Verstege und Alessandro Brusamolino, die den Alltag nicht nur bereichert, sondern zu einer echten Freude gemacht haben. Außerdem möchte ich mich bei Maximilian Horzela und Florian von Cube bedanken, die mich während der Corona-Pandemie unter erschwerten Bedingungen herzlich in die Gruppe aufgenommen und unterstützt haben. Und natürlich danke ich auch allen Kolleginnen und Kollegen am ETP und SCC, die meine Promotion zu einer tollen Zeit gemacht haben. Was oft als rein beruflicher Austausch begann, hat sich schnell zu wertvollen Freundschaften entwickelt.
Vielen Dank an euch alle für die fachliche und persönliche Unterstützung und ganz allgemein die sehr schöne Zeit am ETP!
Weiterhin möchte ich meiner Familie von Herzen danken. Von Anfang an haben sie an mich geglaubt, mir immer den Rücken freigehalten und mir das alles ermöglicht. Nur dank ihrer Unterstützung bin ich so weit gekommen. Ebenso danke ich meinen Freunden, die mir in guten wie in herausfordernden Zeiten zur Seite standen und deren Gesellschaft und Unterstützung mir sehr viel bedeutet. Besonders dankbar bin ich auch meiner Freundin Deborah, die immer für mich da war, mich unterstützt und mir in allen Lebenslagen beigestanden hat.
Allen, die mich auf meinem Weg begleitet haben, spreche ich meinen tief empfundenen Dank aus. Ohne Euch wäre diese Arbeit nicht möglich gewesen.

Glossary

AAA AAA stands for Any Data, Anytime, Anywhere and is the global data federation concept of CMS relying on a hierarchical network of XRootD Redirectors. It enables each member to access the experiment's data seamlessly and efficiently, regardless of their location or that of the data. 74, 75, 76, 77, 81, 121, 170, 272

APEL The Accounting Processor for Event Logs toolkit is an accounting system for the WLCG that collects, processes, and publishes accounting data of grid jobs. 84, 86

Apptainer Apptainer (former Singularity) is a container framework that is particularly popular in HEP computing and HPC as it does not require extensive permissions to run, what is a strict requirement in the WLCG/ grid computing context. 65, 66, 94, 95, 165, 167

ATLAS ATLAS is one of the general-purpose experiments at the LHC and was able to discover the Higgs boson together with the CMS experiment in 2012. 7, 8, 9, 49, 57, 69, 75, 87, 88, 89, 93, 101, 109, 112, 113, 114, 158, 160

AUDITOR The **A**ccoUnting **D**ata handl**I**ng **T**oolbox for **O**pportunistic **R**esources is a framework that allows the building of advanced accounting pipelines. It was initiated to enable a proper accounting for (opportunistic) sub-sites within WLCG. 85, 86

CE Compute Elements are site interfaces that allow job submission systems to communicate with computing resources in a distributed environment like the WLCG. For CMS, HTCondor CEs are used for the job submission infrastructure and act as gateways between the workflow management and the computing resources. With regard to opportunistic resources, they are especially important to enable seamless integration of grid workflows into local batch systems, e.g. with an OBS. 79, 87, 89, 94, 101, 103, 160, 167

CERN The Conseil européen pour la recherche nucléaire (CERN, eng.: European Organization for Nuclear Research), is the European Organization for Nuclear Research and operates the world's largest particle physics laboratory, where scientists use advanced particle accelerators like the Large Hadron Collider to explore the fundamental building blocks and forces of nature. 1, 7, 53, 54, 57, 62, 68, 70, 77, 78, 80, 97, 107, 112

CHS Charged Hadron Subtraction is a mitigation strategy for pileup contribution. It uses tracking information to remove charged particles that supposedly originate from pileup vertices, ensuring that only particles from the primary interaction are included in the jet reconstruction. 16

Cloud Computing Cloud Computing is a distributed computing paradigm that provides on-demand access to a shared pool of resources, typically over the internet. It offers scalable and elastic services, mainly targeting business and consumer applications with a pay-as-you-go pricing model . 49, 62, 63, 113

CMS The Compact Muon Solenoid is one of the general-purpose experiments at the LHC. 1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 21, 25, 27, 28, 29, 30, 31, 43, 44, 47, 49, 51, 52, 57, 58, 59, 60, 63, 68, 69, 73, 74, 75, 76, 77, 79, 81, 89, 93, 94, 98, 99, 101, 107, 109, 112, 113, 116, 117, 118, 121, 132, 133, 136, 154, 158, 159, 160, 161, 162, 164, 165, 167, 170, 171, 172, 185, 221, 222, 225, 229, 231, 260, 272, 275, 286

CMSSW CMSSW, abbreviation of CMS Software, is the offline software framework of the CMS experiment. It comprises the reconstruction, simulation, and analysis of particle collision. 75, 76, 77, 79, 81, 158, 170, 179, 180, 221, 228

COBalD COBalD is a software framework for the balancing of opportunistic resources. It allows to dynamically scale the resource requests depending on the current demand. 89, 90, 93, 94, 95, 97, 108, 109, 111, 113, 131, 132, 167, 228, 231

condition database Condition databases describe specialized databases used in HEP to store time-dependent or configuration and calibration data of the detector that is e.g. necessary for running reconstructions and simulations. 66

containerization Containerization is a virtualization technology that allows applications to be packaged with all their dependencies into isolated, lightweight environments, called containers, ensuring consistency across different host systems and architectures. Today, they are especially important for portability and reproducibility of software and widely used within the WLCG. 49, 64, 65, 67, 110

CRIC The Computing Resource Information Catalog is a centralized database that provides detailed information about the availability, capabilities, and status of the computing and storage resources across the WLCG, including the commitments made by the different sites. 69, 84, 101, 137, 231, 238

CVMFS CernVM-FS is a global, read-only network filesystem based on http and designed for distributing software for HEP and other experiments. It is crucial for providing standardized and validated software environments enabling distributed scientific computing across grid, HPC, and cloud infrastructures. 64, 66, 67, 79, 87, 94, 101, 158

Data Challenges The WLCG Data Challenges are a series of network and storage challenges to prepare for the HL-LHC data taking. The goal for 2024 was 25% of the expected HL-LHC requirement. Further challenges are planned every two years until the start of the HL-LHC. 57, 278

DBS CMS's dataset bookkeeping service is a meta data catalog and query system containing meta data about datasets, including their provenance, specifications, and processing history. 69

dCache dCache is a storage solution for large clusters that can serve as a storage backend as well as a middleware layer for disk and tape cluster management. It works with so-called pools that logically divide a cluster into separate spaces, e.g. for different VOs. 79, 101

DESY DESY is one of the largest German research facility and the second largest WLCG site in Germany. It has two locations, Hamburg and Zeuthen, with different focuses. As an Helmoltz Center, like KIT, it is the backbone of the German WLCG infrastructure. 99, 101, 104, 107, 114, 161

DM Dark Matter is a hypothetical, elusive, and non-luminous form of matter which interacts very weakly with ordinary matter. Mainly astrophysical observations strongly indicate its existence, e.g. through gravitational effects, which suggest that most of the Universe's mass consists of dark matter. Today, large areas of modern (high-energy) dedicate significant efforts to uncovering its properties and underlying nature. 16

Docker Docker is a containerization technology and was one of the first available solutions that made containers very popular. Today, many alternatives became available, like podman or apptainer, that especially fit better the HEP use-case, as they require less permissions. But docker is still often used as a generic trademark for containerization, something like google for web searches. 64, 65, 168, 182

ECAL The Electromagnetic Calorimeter (ECAL) is a sub-detector made of lead tungstate crystals which measure the energy and position of electromagnetically interacting particles (mainly electrons and photons) by absorbing their energy and converting it into scintillation light. 11, 14

EGI The European Grid Infrastructure is a federation of distributed storage and computing resources that connects research institutions across Europe. Today, it is not entirely focused on classical grid computing anymore, but still contributes to the field with software tools (middleware) and support for resource providers and users in several scientific fields. 53, 86

EOS EOS is a storage technology written mostly in C/C++ that is actively developed and used at CERN since 2011. XRootD is the core technology providing a remote access protocol, but also other access methods are available: a POSIX-like FUSE client, http, and WebDav. More information: https://github.com/cern-eos/eos and https://eos-web.web.cern.ch. 70, 73, 81

FSR Final State Radiation (FSR) refers to the radiation – typically gluons or photons – emitted by the outgoing particles after the primary collision event. 21

FTS The File Transfer Service can be seen as the executing backend of Rucio, as it actually transfers the data across the various storage endpoints in the WLCG as determined in the Rucio rules. This includes the transfers to the tape systems as well as the stage-out and replication of tape data. 70, 75, 80

FUSE FUSE is a unix kernel modul that allows to create virtual filesystems in user space without any privileges. 66

Grafana Grafana is a monitoring platform capable of querying and incorporating several various data sources, such as OpenSearch, Influx, Prometheus, and others. The open-source tool is especially useful for the visualization of time-series data with customizable, interactive dashboards and advanced alerting mechanisms. 59, 78, 80, 82

Grid Computing Grid computing is a distributed computing paradigm, proposed by Ian Foster and Carl Kesselman, where individual, often geographically dispersed resource providers are loosely coupled to collaborate on complex (computing) tasks. The naming is inspired by the power grid that has an analog philosophy of sharing of resources, like processing power and storage, provided over a standardized point of entry (like the sockets for the power grid). 49, 52, 62, 79, 106, 109, 112

GridKa GridKa, formerly known as FZK-LCG2, is the German WLCG Tier-1 center located at KIT Campus North. It is a multi-VO facility providing several PBs of storage and multiple ten thousands of CPUs to the LHC community. Additionally, it integrates opportunistic resources, e.g. in form of TOpAS, the local Tier-3, or HoreKa, the scientific HPC cluster at KIT. 68, 77, 79, 82, 83, 89, 93, 94, 99, 101, 103, 107, 137, 138, 151, 160, 162, 167, 171, 173, 174, 180, 181, 182, 183, 223, 225, 231, 237, 238, 239, 240, 242, 243, 244, 268, 282

HammerCloud HammerCloud tests are performance tests for large-scale (grid) infrastructures used to tests and validate the performance computing resources at various sites. For probing, they run typical experiment workloads. 79, 171, 183

HappyFace4 HappyFace4 is a meta-monitoring tool developed at KIT which allows the integration and unification of arbitrary monitoring sources. The visualization of all available information on one overview page helps to identify problems fast and trigger according solutions. 82, 83, 95, 168

HCAL The Hadron Calorimeter (HCAL) is the sub-detector surrounding the ECAL. Designed as a brass-based sampling calorimeter, it has a higher stopping power so that it is able to measure the energy and position of hadronically interacting particles by absorbing their energy in the dense absorber material and converting the resulting showers into detectable light. 11, 14

HDFS The Hadoop Distributed File System (HDFS) is a scalable, distributed file system across clusters of commodity hardware within the Apache Hadoop ecosystem designed to manage and store large datasets. With advanced features like streaming of datasets, it is widely used, often in combination with Spark. 78, 225

HEP HEP is the commonly used abbreviation for High Energy Physics, including all particle physics related fields, from the smallest to the largest scales, experimental and theoretical, up to computing. However, there is no explicit definition and HEP in the context of this thesis mainly addresses the high energy elementary particle physics within the LHC context. 2, 3, 4, 7, 42, 47, 49, 51, 52, 53, 55, 56, 58, 59, 60, 61, 63, 65, 66, 70, 75, 79, 84, 91, 95, 97, 99, 104, 106, 108, 109, 110, 111, 113, 114, 115, 116, 120, 128, 130, 132, 134, 135, 138, 146, 150, 151, 153, 157, 158, 159, 161, 164, 165, 167, 171, 172, 174, 175, 179, 182, 183, 185

HEPCloud HEPCloud is a framework developed primarily at Fermilab to enable seamlessly access to heterogeneous computing resources, focusing on commercial and academic clouds, high-performance centers, and classic grid infrastructures. 62

HL-LHC The HL-LHC, where HL stands for High-Luminosity, describes the next phase of the Large Hadron Collider starting in the early 2030s. Designed for increased luminosity, allowing for more frequent particle collisions, it enables a deeper exploration of fundamental physics including high-precision measurements. 2, 7, 47, 49, 50, 52, 56, 58, 62, 63, 75, 97, 98, 102, 106, 107, 109, 110, 113, 116, 132, 152, 153, 184

HLT The High-Level Trigger (HLT) of the CMS experiment is a software-based event filtering system (in opposite to the hardware-based L1 trigger), which is running on a large computing farm located in the cavern next to the detector. It applies fast, simplified reconstruction algorithms to refine the selection based on the informative value of the events and reduces the rate of accepted events down to about 1 kHz (5 kHz for Run 3), which is tolerable for the readout and storage systems. 12, 29, 30

HoreKa The Hochleistungsrechner Karlsruhe is a high-performance computing (HPC) cluster located at KIT in Germany and part of the NHR compound. Its main purpose is to support federal research in various fields such as physics, chemistry, biology, climate science, and engineering. 2, 89, 91, 92, 93, 94, 95, 97, 98, 99, 108, 109, 113, 114, 117, 118, 130, 132, 134, 136, 137, 138, 140, 142, 143, 144, 146, 148, 149, 150, 153, 154, 155, 156, 157, 159, 160, 161, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 181, 182, 183, 185, 186, 221, 223, 225, 226, 228, 230, 231, 232, 233, 235, 236, 237, 238, 239, 242, 243, 245, 246, 249, 254, 256, 257, 265, 266, 269, 270, 271, 272, 273, 282, 284, 287

HPC High-Performance Computing summarizes the use of supercomputers and advanced computing techniques to perform complex calculations, simulations, and data processing at extremely high levels. It is widely used in industry and academic research. 2, 49, 51, 55, 59, 60, 61, 63, 64, 67, 86, 89, 90, 93, 95, 97, 98, 99, 104, 105, 106, 108, 109, 110, 111, 112, 113, 114, 116, 128, 130, 132, 134, 138, 139, 141, 143, 146, 148, 150, 151, 152, 153, 154, 155, 157, 160, 161, 163, 164, 165, 167, 169, 172, 173, 174, 175, 177, 179, 182, 183, 185, 186, 232, 233, 251, 262, 267, 272

HS06 HepSpec06 is a benchmarking tool that was widely used in HEP computing for providing a common performance rating of WLCG resources as a basis for proper accounting. It was replaced by HepScore23, a more HEP-workflow oriented tool. 84

HS23 HepScore23 is the modern successor for the HepSpec06 benchmarking suite. It is more oriented on modern HEP workloads and better takes multi-core applications into account. 84, 85, 101, 104, 117, 120, 128, 135, 137, 178, 231, 237, 238

HTC High-Throughput Computing refers to a distributed computing paradigm that focuses on maximizing the throughput of data rather than just parallel performance, aiming to enhance computational efficiency and speed in a widely distributed computing environment. It is the dominant concept within WLCG, where processing large volumes of data and handling numerous tasks reflect the real-world demands of the compute model. 58, 60

HTCondor HTCondor is a high-throughput workload management system that efficiently schedules and executes distributed computing tasks across heterogeneous resources, forming a key component of the data processing infrastructure for the WLCG. 58, 59, 60, 61, 62, 81, 87, 89, 94, 109, 112, 117, 118, 167, 221, 222, 225, 286

IBM Spectrum Scale/GPFS IBM Spectrum Scale is an enterprise software-defined storage solution that is based on GPFS, providing scalable and high-performance data management across multiple locations and environments. GPFS (General Parallel File System) is a high-performance, scalable file system developed by IBM for managing large volumes of data across distributed storage environments. In technical slang, it is often used as a synonym for the IBM spectrum scale solution. 91, 165

InfiniBand InfiniBand is a high performance, low latency interconnect, commonly used in HPC environments. It supports Remote Direct Memory Access (RDMA) and is the basis of data transfers in multi-node setups. 93, 154, 166, 167, 182, 263

IoV An Interval of Validity (IOV) in CMS typically describes a range of run numbers during which a specific set of calibration, alignment, or other condition data is considered valid for data reconstruction and analysis. 28, 31, 32

IPoIB IP over InfiniBand is a network interface implementation that allows IP protocol based communication over InfiniBand. Typically used for high-bandwidth, low-latency networking for data centers and high-performance computing environments. 91, 92, 165

ISR Initial State Radiation (ISR) describes the emission of (mostly) quarks or gluons from the incoming particles before the hard scattering event in a collision. 21

JEC Jet Energy Calibration is the process of applying corrections to the raw energy measurements of particle jets in order to account for pileup, detector effects, energy losses, and systematic biases, ensuring that the reconstructed jet energy accurately reflects the true particle-level energy. 1, 3, 6, 7, 13, 14, 15, 17, 18, 25, 27, 28, 30, 47, 49

JER At CMS, Jet Energy Resolution (JER) quantifies the precision with which the detector measures the energy of jets produced in high-energy collisions, serving as a key indicator of its performance and calibration. JER corrections are derived based on both simulation and data-driven techniques and help to mitigate inherent detector limitations, ensuring a better matching with the true jet energies. 6, 7, 21, 25, 28

JES Within CMS, the Jet Energy Scale describes the calibration that adjusts the measured energy of jets to accurately reflect the true energy of the originating particles by correcting for detector response and other effects. 6, 18

jet A jet is a cone-shaped, collimated cascade of particles typically emerging from high-energy quarks or gluons. Jets make up a significant part of the study of QCD and serve as important signatures that allow the investigation of the underlying strong force. 1, 3, 4, 5, 6, 7, 14, 15, 16, 17, 18, 19, 30, 276

Jet Energy Correction Jet Energy Corrections (not to be confused with Jet Energy Calibration) describe the sequential steps applied for correcting the measured jet energies to reliably reflect the true energies, which is important for a wide range of physics analyses. 4, 6, 16, 17, 18, 23, 24, 27, 37, 47, 185

Jupyter Jupyter is an open-source framework and platform for interactive, web-based coding, data analysis, and in-place visualization including (markdown) text in so-called notebooks. 78

KIT The Karlsruhe Institute of Technology is a German university together with a Helmoltz research center located 10km north of Karlsruhe (Campus North). It hosts the German WLCG Tier-1 site, GridKa, as well as several additional scientific (HPC) clusters, like HoreKa or BwUniCluster2.0. 2, 28, 42, 43, 45, 46, 47, 55, 67, 79, 82, 83, 86, 89, 100, 101, 104, 107, 108, 114, 132, 138, 151, 161, 173, 183, 231, 269, 282

Kubernetes Kubernetes, often abbreviated with K8s, is an open-source low-level container orchestration platform initially introduced by Google. Today, K8s is maintained by the Cloud Native Computing Foundation. It automates the deployment, scaling, and management of micro-services and containerized applications in general across multiple servers. By providing features like auto-deployment, self-healing, load balancing, and automatic and elastic scaling in distributed environments, K8s became an indispensable tool of the modern Internet. 62, 64, 89

L1 The Level-1 correction as part of the Jet Energy Corrections at CMS corrects for additional energy from pileup and underlying event by subtracting the estimated from the raw jet energy. This is typically done with the jet area method, heavily relying on MC simulations. 18, 19, 20

L2 The Level-2 correction as part of the Jet Energy Corrections at CMS adjusts variations in the response of the detector in dependence of η and ensures a uniform energy scale throughout the detector. 18, 20, 21, 23

L2L3 L2L3 describes the combined Level-2 and Level-3 corrections. 18, 21, 23, 25

L2L3Res The L2L3Res correction step effectively unifies the L2Res and L3Res corrections into a single residual calibration step that addresses any remaining discrepancies between data and simulation in η and $p-T$. 18, 25

L2Res The Level-2 relative residual correction as part of the Jet Energy Corrections at CMS is a data-driven method utilizing dijet events to minimize the remaining discrepancy between data and simulation in relative η bins after applying the L2 corrections. 21, 23, 31, 33, 34, 35, 36, 37, 38, 39, 40

L3 The Level-3 correction as part of the Jet Energy Corrections at CMS adjusts variations in the absolute energy scale of the jet energy measurements in terms of p_T and ensures that the measured energies reflect the true particle-level energies. 18, 20, 21, 23

L3Res The Level-3 absolute residual correction as part of the Jet Energy Corrections at CMS describes data-driven methods that fine-tune the absolute energy scale by minimizing differences between data and simulation using different comparison methods. 18, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39, 40, 42, 44, 47, 210, 213, 216, 218

LCG The LHC Computing Grid is a globally distributed computing infrastructure designed to process and analyze the vast amounts of data collected by the experiments at the LHC, enabling researchers around the world to share and utilize computing resources for data storage, processing, and analysis. After being full-scale available with the start of the LHC, the inclusion of worldwide resources led to the term worldwide LCG (WLCG) becoming established. 52, 53, 63

lfn A logical file name is the human-readable file path to identify a file in the global data management of CMS independent of its physical location. It forms the physical file name (pfn) together with the access protocol and the storage endpoint. By this, XRootD can be used (when addressing a Redirector) to automatically locate the file on a physical storage across the distributed grid. 75, 81

LHC The Large Hadron Collider is a circular particle collider based at CERN. At the time of writing, a maximum of 13.613.6 TeV total collision energy is reached during Run 3. 1, 3, 4, 7, 8, 11, 28, 49, 52, 53, 55, 56, 63, 66, 68, 70, 84, 91, 99, 101, 106, 110, 113, 127, 132, 185, 275, 280

LHCONE The LHC Open Network Environment is a private, dedicated network that is connecting the LHC Tier-0, Tier-1, and Tier-2 sites. It complements LHCOPN, which provides additional connectivity between the Tier-0 and the Tier-1s. 80, 87, 112, 182

LHCOPN The LHC Optical Private Network is one of the dedicated networks, alongside LHCONE, connecting the WLCG grid sites. It is an optical high-speed network and the backbone for the raw data redistribution between the Tier-0 and the Tier-1s. 56, 57, 80

Lustre Lustre is a high-performance, scalable distributed file system commonly used in large-scale computing environments, such as supercomputers and data centers, to manage and store vast amounts of data. *Glossary:* Lustre

MC Monte Carlo (MC) in the HEP context refers to computational techniques that use random sampling and statistical methods to model and analyze complex process, e.g. in simulations of particle collisions. They are especially important, when deterministic solutions are difficult or impossible to obtain. 7, 18, 19, 20, 21, 25, 28, 30, 31, 32, 34, 35, 36, 37, 38, 40, 47, 49, 51, 55, 69, 75, 107, 134, 158, 228

MET Missing Transverse Momentum – or Missing Transverse Energy – (MET) is calculated as the negative vector sum of the transverse momenta of all reconstructed PF candidates in a collision event. Figuratively speaking, it represents the imbalance in momentum in the plane perpendicular to the beam and indicates the presence of undetected particles. 16, 17, 23, 28, 30, 31, 32

MONARC The MONARC was a software framework designed for analyzing and managing the risks and performance of large-scale computing systems by simulating their behavior under various scenarios and optimizing their operation to find a feasible computing model for the LHC. 52, 53

MONIT CERN MONIT is the centralized monitoring service for CERN and the WLCG. It is used to collect, process, store, and visualize metrics with a comprehensive toolset, including, InfluxDB, OpenSearch, Grafana, Prometheus, and others. 59, 77, 78, 79, 80, 112, 281

Moore’s Law Moore’s Law is the observation that the number of transistors on a microchip doubles approximately every two years, leading to increased computing power and reduced costs. It is more of an observation and prediction than a real law, but it is so popular that it has driven the rapid advancement of technology, enabling faster, smaller, and more affordable compute resources. Rumors say that CPU manufacturers even set their R&D targets to align with the law. However, due to physical limits, mainly in form of pitches, manufacturing challenges, and the so-called power wall, the law is expected to at least slow down over the next decade. 51

MoU A Memorandum of Understanding is a formal agreement between the institutions forming the WLCG. It describes the roles, tasks, and responsibilities for resource providers and regulates the collaboration. 87, 110, 112

MPF The missing transverse momentum projection fraction method, commonly abbreviated with MPF, calibrates jets by projecting the missing transverse energy (MET) along the direction of a well-measured reference object (typically photons or Z bosons) to quantify the discrepancy between the expected energy response and the measured response of the recoiling hadronic system. 21, 22, 23, 31, 33, 34, 35, 36, 37, 38, 40, 41, 209, 210, 211, 212, 213, 214, 216, 218, 219

NHR National High-Performance Computing Alliance (Nationales Hochleistungsrechnen, NHR) in Germany is a consortium of nine university-based HPC centers providing advanced computational resources, support, and training to researchers in Germany. 91, 93, 98, 104, 105, 108, 109, 110, 111, 112, 113, 115, 116, 117, 132, 138, 139, 140, 143, 144, 146, 148, 150, 153, 157, 158, 161, 162, 165, 172, 174, 175, 179, 182, 183, 185, 186, 249, 254, 256, 285

OBS An Overlay Batch System dynamically integrates external and opportunistic resources, into a local batch system, creating a unified and scalable resource pool. It seamlessly expands or contracts the available compute capacity based on job queue demands, ensuring efficient utilization and cost optimization. 89, 90, 94, 113, 132, 167, 231

OpenSearch OpenSearch is an open-source fork of ElasticSearch by AWS designed for full-text search and therefore ideal for log analytics. With Dashboards (formerly Kibana), it provides a comprehensive online analytics tool for data visualization. 78, 80, 81, 165, 168

OpenShift OpenShift is an open-source container application platform built on Kubernetes that facilitates the deployment, management, and scaling of containers, providing operational and development tools for continuous integration, automation, and orchestration for cloud environments. 62, 64

OpenStack OpenStack is an open-source cloud computing platform that allows the creation and management of public and private clouds by providing tools for compute, storage, and networking resources. It is often used in large-scale data centers. 62, 89

origin An XRootD server that provides data in a distributed data federation is often called a (remote) origin. *Glossary:* origin, 72, 73, 77, 81, 170, 180, 263

OS Operating systems are software that manages hardware resources and provides user interfaces. It also provides the environments for running software applications. The kernel is the core component of the OS responsible for managing system resources and facilitating communication between hardware and software. 65

OSG The Open Science Grid is a distributed grid computing infrastructure in the US founded in 2004. It provides computing resources to support scientific research across multiple disciplines and supports and contributes greatly to the WLCG, e.g. with tools focusing on high-throughput computing. 53, 80

PDF Parton distribution functions (PDFs) are probability densities that describe how the momentum of a hadron is distributed among its constituents, often referred to as partons, at a given energy scale. 4

PF Particle Flow (PF) describes an algorithmic method employed by the CMS experiment that integrates information from all sub-detectors to reconstruct and identify each particle in an event, helping to disentangle the complex hadronic interactions and leading to enhanced precision in measuring the energy and momentum of jets and other physics objects. 13, 14, 15, 16, 17, 28

pfn The pfn is combined from the physical location of a file, the access protocol, and its lfn. pfn and lfn are mapped in the Rucio file catalog for allowing to locate all replicas of a certain file. 75, 77

PhEDEx CMS's Physics Experiment Data Export system (PhEDEx) was the old data management tool that was retired for Run III and replaced by Rucio. While

DBS handles the meta data, PhEDEx was responsible for the physical data placement across the WLCG storage infrastructure. 69

pileup Pileup refers to the overlapping of collisions during a single bunch-crossing (in-time) or between bunch-crossings (out-of-time). It introduces extra signals that complicate the reconstruction of events and needs a mitigation to provide reliable results. 9, 14, 16, 17, 30

pilot Pilot jobs are skeleton jobs that create a slot for actual experiment workflows. CMS, for example, submits empty 8-core pilots to the global HTCondor pool. When they are scheduled and occupy a resource, they report back to the central workflow management allowing to submit an actual payload with the CMS jop glide-in mechanism. This simplifies the resource provisioning and enables an efficient task scheduling managed by the experiments workflow management, independently of the the physical resources. 58, 60, 80, 89, 94, 130, 131, 154, 223, 260

PREMIX PREMIX is a dataset type within CMS that is used for storing background event data for MC simulations. Instead of simulating pileup and other noise in real-time for each event, the centrally provided PREMIX datasets are used to combine the available background stochastically with simulated signal process during event generation. By this, the computing demand of the simulations is significantly reduces. 69, 107, 134, 158, 159

PUPPI The PileUp Per Particle Identification approach is, alongside CHS, a mitigation strategy for pileup jets at CMS. It evaluates the spatial and momentum distribution of events together with tracking information to assign a weight to each particle, effectively down-weighting those likely originating from pileup, resulting in an enhanced reconstruction quality of jets. 16, 17

QCD Quantum Chromodynamics (QCD) describes the interaction of quarks and gluons via the strong force, mediated by a non-Abelian SU(3) gauge symmetry. A special feature is that the exchange particles carry a color charge themselves, which can lead to gluon self-interactions that produce the non-linear dynamics responsible for phenomena such as asymptotic freedom at short distances and confinement at long distances/low energies. 1, 3, 4, 5, 6, 18, 20, 47, 185

QED Quantum Electrodynamics (QED), whose modern formulation was originating from Richard Feynman's development of diagrammatic techniques to visualize the complex interaction processes in a simplistic way (Feynman Diagrams). It is the quantum field theory that describes how electrically charged particles interact via the exchange of photons. 5

ROOT The ROOT Data Analysis Framework is a software toolkit developed at CERN. It is used for data handling and implements the .root data format, which is the default for files in the HEP context. 70

Rucio Rudio is a policy-based data management system designed to handle Exabytes with billions of files. It ensures the efficient storage, replication, and access of the LHC data across the globally distributed WLCG storage infrastructures.

The name is inspired by Sancho Panza's reliable donkey in Miguel de Cervantes' Don Quixote. 69, 70, 75, 80, 114, 158, 160

Run A Run in the context of the LHC refers to the mayor data taking periods with long shutdowns in between. The LHC runs comprise: Run 1 (2010-2012), Run 2 (2015-2018), Run 3 (2022-2025 – planned) and Run 4 will be the future HL-LHC. The term is not to be confused with a run on experiment level, which describes a shorter data-taking period with a stable beam without interruption, allowing to associate specific collision conditions, e.g. beam energy or detector status, with the measured events for a proper analysis. 7, 9, 11, 12, 13, 25, 27, 28, 42, 43, 47, 106, 107, 185, 209

SAM The Site Availability Monitoring (SAM) tests are centrally triggered probes of the distributed computing infrastructure. They are designed for testing availability, reliability, and functionality of services and sites within the WLCG. 79, 80

SE Storage element is the term for storage resources that provide managed, distributed, and scalable storage for experiment data within the WLCG. 80

Slurm The Simple Linux Utility for Resource Management is an open-source workload manager and job scheduler designed for HPC systems. 58, 59, 60, 61, 89, 93, 95, 112

SM The Standard Model of particle physics is the currently most successful theory describing three of four known fundamental forces including all known elementary particles. However, in addition to the missing integration of the gravitational force, there is strong evidence that the theory is incomplete. Theories that go beyond that point are called *Beyond Standard Model* theories. 1, 3, 7, 47, 185

SWAN SWAN (Service for Web-based Analysis) is an online, cloud-based Jupyter service by CERN. It can be used for interactive data analysis and has an integration of Analytix, a Spark cluster at the CERN data center, and the HTCondor batch system for larger scale tasks. 78

TARDIS The Transparent Adaptive Resource Dynamic Integration System is a tool for the dynamic and transparent integration of different opportunistic resources into one overlay batch system. 62, 89, 90, 93, 94, 95, 97, 108, 109, 111, 113, 131, 132, 167, 228, 231

Tier A Tier describes a level within the hierarchical architecture of the WLCG. Ranging from Tier-0 to Tier-3, each Tier has a distinct task and according responsibilities. 54, 55, 56, 57, 59, 62, 68, 75, 78, 79, 86, 87, 88, 90, 99, 100, 101, 104, 106, 108, 109, 110, 112, 113, 116, 117, 132, 136, 138, 142, 143, 145, 148, 151, 157, 158, 160, 161, 171, 172, 173, 174, 175, 182, 183, 223, 231, 237, 238, 241, 269, 272, 285

TOpAS The Throughput Optimized Analysis System is a Tier-3 center within the WLCG located at KIT. It provides compute resources to the local institute and opportunistically to the WLCG and attached experiments. Furthermore, it is used as an R&D site of the German Tier-1, e.g. investigating the provisioning of GPUs over the grid. 55, 86, 89, 132, 136, 137, 138, 141, 144, 148, 151, 155, 157, 169, 171, 174, 175, 178, 183, 184, 223, 225, 231, 232, 233, 235, 236, 237, 238, 239, 240, 241, 242, 247, 282, 284

Ultra Legacy CMS Ultra Legacy (UL) datasets were (intentionally) the finally processed datasets of Run 2 ready for physics analyses. The idea was to reprocess all Run 2 data using the latest calibrations, alignments, and reconstruction algorithms to create a uniform, high-quality dataset for high-precision physics. 13, 27, 28, 29, 31, 32, 37, 42, 47

VM Virtual machines are full simulations of computers that run a complete operating system. They therefore provide a fully separated environment, including an own kernel, which is different in comparison to containers that share the host system's kernel. 65, 89, 95, 111, 112, 131

VO A Virtual Organization (VO) is a virtual representation of experiment or collaboration within the distributed grid computing infrastructure. It serves as a virtual entity in cyberspace, defining access policies, resource allocation, and data management for users and resources across multiple sites, enabling seamless collaboration and resource sharing within the WLCG. 62, 73, 84, 101, 103, 104, 107

WAN A Wide Area Network is a telecommunications network that spans a large geographic area, like the Internet or LHCONE, connecting multiple local area networks (LANs) and enabling communication over long distances. 70, 93

WebDAV Web Distributed Authoring and Versioning (WebDAV) is an HTTP extension that is used for transferring data all over the Internet. Its advantage is the high compatibility as an industry standard supporting efficient data transfers but not designed for streaming data like the xroot protocol. 70

WLCG The Worldwide LHC Computing Grid is the connection of over 160 international institutions providing compute resources and data storage for the collaborations affiliated with the Large Hadron Collider (LHC) experiments, enabling scientists to analyze the vast amount of particle collision data for high-precision physics. 2, 49, 51, 53, 54, 55, 56, 57, 58, 59, 60, 62, 64, 66, 67, 68, 69, 70, 71, 73, 75, 77, 78, 79, 82, 85, 86, 87, 88, 89, 93, 94, 95, 97, 98, 100, 101, 102, 103, 104, 105, 106, 108, 110, 111, 112, 113, 114, 116, 121, 128, 130, 132, 135, 138, 139, 140, 143, 144, 146, 148, 149, 151, 153, 154, 157, 159, 161, 162, 164, 167, 169, 172, 173, 174, 176, 179, 182, 183, 185, 186, 225, 231, 249, 254, 273

WMAgent WMAgent is part of the meta project WMCore (https://github.com/dmwm/WMCore), the workflow management framework of CMS, included in the data management and workflow management (DMWM) system. 81

XCache XCache refers to the caching feature of XRootD that locally stores frequently accessed data to reduce network load and accelerate data retrieval. 72, 73, 160, 162, 164, 165, 166, 167, 184

xroot The xroot protocol is a network protocol designed for data transfers in the HEP computing context. It is designed for fast, high-efficient, parallel, and asynchronous data access of huge files across multiple servers, while being highly scalable and reliable. One of its fundamental features is the ability to stream data efficiently enabling on-demand access to large-scale distributed datasets while limiting the transfer to what is actually required, which is a significant optimization. 70, 71

XRootD The eXtended ROOT Daemon, short XRootD, is a high-performance, distributed data storage and access framework implementing the xroot protocol. The server side is designed to handle large-scale data in distributed scientific computing environments with an automatic data detection mechanism. It is highly scalable, what is demonstrated with EOS. The client side is designed for efficient data access via the xroot protocol enabling distributed large-scale data analysis. 70, 71, 72, 73, 75, 76, 79, 80, 153, 158, 162, 163, 164, 165, 166, 167, 168, 169, 170, 172, 175, 180, 181, 182, 184, 186, 258, 260, 261, 262, 263, 264

XRootD Caching Proxy An XRootD Caching Proxy is an XRootD Proxy that additionally can cache transfers on a local filesystem using the xrd.pfc (proxy file cache) directive. It is often referred to as XCache. This can be useful for distributed sites without own pledged storage but capacities that can be used for (temporarily) storing files locally. The caching may reduce remote transfers, spare file servers, and increase the overall efficiency of data processing that is heavily relying on remote data, if a decent cache hit rate is given. *Glossary:* XRootD Caching Proxy, 73, 162, 164, 167, 168, 170, 175, 264

XRootD Manager A component of XRootD that coordinates independent, distributed data servers, localizes data in the complex infrastructure, and manages the distribution of accesses across an XRootD-based data federation including proper load-balancing on locality optimizations. *Glossary:* XRootD Manager, 73

XRootD Proxy An XRootD Proxy is an XRootD server that redirects requests to a certain destination. This may be used for fronting a cluster behind a firewall or to redirect traffic internally, where the destination is fixed (direct mode). Alternatively, it can be used to freely forward client's requests to their desired destination (forwarding mode), which for example can be a Redirector. *Glossary:* XRootD Proxy, 72, 73, 163, 164, 170, 172, 174, 175, 176, 177, 179, 180, 184, 260, 266

XRootD Redirector Colloquial synonym for an XRootD Manager. *Glossary:* XRootD Redirector, 72, 73, 74, 77, 79, 162, 180, 181

Z+Jet Z+Jet events are high-energy collision events in which a Z boson is produced in association with one or more jets. They can be used as a clean reference for calibrating jet energies due to the well-measured properties of the Z boson's decay products ($\mu^+\mu^-$ or e^+e^-) in a back-to-back topology. 2, 6, 18, 23, 24, 25, 27

Acronyms

AAA Any Data, Anytime, Anywhere. *Glossary:* AAA, 74

APEL Accounting Processor for Event Logs. *Glossary:apelg* , 84

AUDITOR AccoUnting Data handlIng Toolbox for Opportunistic Resources. *Glossary:auditorg* , 85

BSM Beyond Standard Model. *Glossary:* SM, 1

CE Compute Element. *Glossary:* CE, 101

CERN Conseil européen pour la recherche nucléaire. *Glossary:* CERN

CHS Charged Hadron Subtraction. *Glossary:* CHS, 16

CMS Compact Muon Solenoid. *Glossary:* CMS, 1

COBalD COBalD – the Opportunistic Balancing Daemon. *Glossary:cobaldg*

CRIC Computing Resource Information Catalog. *Glossary:* CRIC, 84

CVMFS CernVM File System. *Glossary:* CVMFS, 64

DBS dataset bookkeeping service. *Glossary:* DBS

DC Data Challenges. *Glossary:dcg* , 57

DESY Deutsches Elektronen-Synchrotron. *Glossary:* DESY

DM Dark Matter. *Glossary:* DM

ECAL Electromagnetic Calorimeter. *Glossary:* ECAL, 11

EGI European Grid Infrastructure. *Glossary:* EGI, 53

EOS EOS storage system. *Glossary:* EOS

FST Final State Radiation. *Glossary:* FSR, 21

FTS File Transfer Servive. *Glossary:* FTS, 70

FUSE Filesystem in Userspace. *Glossary:* FUSE

GPFS IBM Spectrum Scale/GPFS. *Glossary:* IBM Spectrum Scale/GPFS

HCAL Hadron Calorimeter. *Glossary:* HCAL, 11

HDFS Hadoop Distributed File System. *Glossary:hdfsg* , 78

HDFS Hadoop Distributed File System. *Glossary:hdfsg*

HEP High Energy Physics. *Glossary:* HEP, 2

HL-LHC High-Luminosity Large Hadron Collider. *Glossary:* HL-LHC, 2

HLT High-Level Trigger. *Glossary:* HLT, 12

HoreKa Hochleistungsrechner Karlsruhe. *Glossary:horekag* , 91

HPC High-Performance Computing. *Glossary:* HPC, 2

HS06 HepSpec06. *Glossary:hs06g* , 84

HS23 HepScore23. *Glossary:hs23g* , 84

HTC High-Throughput Computing. *Glossary:* HTC, 58

IoV Interval of Validity. *Glossary:iovg* , 28

IPoIB IP over InfiniBand. *Glossary:* IPoIB, 92

IST Initial State Radiation. *Glossary:* ISR, 21

JEC Jet Energy Calibration. *Glossary:* JEC, 1

JER Jet Energy Resolution. *Glossary:* JER, 6

JES Jet Energy Scale. *Glossary:* JES, 6

KIT Karlsruhe Institute of Technology. *Glossary:* KIT, 2

LCG LHC Computing Grid. *Glossary:* LCG, 52

lfn logical file name. *Glossary:* lfn, 75

LHC Large Hadron Collider. *Glossary:* LHC, 1

MC Monte Carlo, often short for Monte Carlo simulations. *Glossary:mcg* , 7

MET Missing Transverse Momentum. *Glossary:* MET, 16

MONARC High Energy Physics. *Glossary:* MONARC

MoU Memorandum of Understanding. *Glossary:* MoU, 87

MPF missing transverse momentum projection fraction. *Glossary:* MPF, 21

NHR National High-Performance Computing Alliance (Nationales Hochleistungsrechnen, NHR). *Glossary:* NHR, 91

OBS Overlay Batch System. *Glossary:obsg* , 89

OS Operating System. *Glossary:* OS

OSG Open Science Grid. *Glossary:* OSG, 53

PDF Parton distribution functions. *Glossary:* PDF, 4

PF Particle Flow. *Glossary:* PF, 13

pfn physical file name. *Glossary:* pfn, 75

PUPPI PileUp Per Particle Identification. *Glossary:* PUPPI, 16

QCD Quantum Chromodynamics. *Glossary:* QCD, 1

QED Quantum Electrodynamics. *Glossary:* QED, 5

ROOT ROOT framework. *Glossary:* ROOT

SE Storage Element. *Glossary:* SE, 80

Slurm Simple Linux Utility for Resource Management. *Glossary:slurmg* , 60

SM Standard Model of particle physics. *Glossary:* SM, 1

TARDIS Transparent Adaptive Resource Dynamic Integration System. *Glossary:tardisg*

TOpAS Throughput Optimized Analysis System. *Glossary:topasg* , 55

UL Ultra Legacy. *Glossary:* Ultra Legacy

VM Virtual Machine. *Glossary:* VM, 65

VO Virtual Organization. *Glossary:vog* , 62

WAN Wide Area Network. *Glossary:* WAN

WLCG Worldwide LHC Computing Grid. *Glossary:* WLCG, 2

XRootD eXtended ROOT Daemon. *Glossary:* XRootD, 70