Development of a Virtualized Computing Environment for LHC Analyses on Standard Batch Systems and Measurement of the Inclusive Jet Cross-Section with the CMS experiment at 7 TeV

Zur Erlangung des akademisches Grades eines
DOKTORS DER NATURWISSENSCHAFTEN
von der Fakultät für Physik des
Karlsruher Instituts für Technologie genehmigte

DISSERTATION

von

Dipl. Phys. Oliver Oberst
aus Rastatt

Mündliche Prüfung: 21. Januar 2011

Referent: Prof. Dr. G. Quast
Institut für Experimentelle Kernphysik

Korreferent: Dr. habil. M. Kunze
Steinbuch Centre for Computing

meiner Familie

Contents

Introduction **1**

1 Quantum Chromodynamics - The Theory of the Strong Interaction **5**
1.1 The Standard Model of Elementary Particle Physics 6
1.2 Cross Sections 7
1.3 Quantum Chromodynamics 9
1.3.1 The QCD Lagrangian 10
1.3.2 The Running Coupling 11
1.3.3 Hadrons 12
1.3.4 Deep Inelastic Scattering 13
1.3.5 QCD Improved Parton Model 15
1.4 From Quarks to Hadrons 17
1.4.1 Parton Shower 19
1.4.2 Hadronisation 21
1.5 Jet Algorithms 23
1.5.1 Collinear Safety 24
1.5.2 Infrared Safety 26
1.5.3 Combinatorial Algorithms 26

2 The CMS Experiment **29**
2.1 The Large Hadron Collider 29
2.2 The Compact Muon Solenoid 32
2.2.1 Tracking System 32
2.2.2 The Silicon Pixel Detector 32
2.2.3 The Silicon Strip Detector 35

2.2.4 Calorimeter System 35
2.2.5 The Electromagnetic Calorimeter (ECAL) 35
2.2.6 The Hadronic Calorimeter (HCAL) 36
2.2.7 Superconducting Solenoid 37
2.2.8 Muon Chambers 37
2.2.9 Data Acquisition and Trigger 38
2.3 Grid Computing 38

3 Virtualisation in High Throughput Computing and Grid Environments 41
3.1 Virtualization 42
3.1.1 Definition of Virtualization 42
3.1.2 Virtualization on the x86 Architecture 43
3.2 Virtualization in High Performance and High Throughput Computing . . . 47
3.2.1 ViBatch - Virtualization of Worker Nodes within Common Batch Systems 49
3.3 Summary 56

4 Measurment of the Double Differential Inclusive Jet Cross Section at 7 TeV with CMS 59
4.1 The Observable 60
4.1.1 Binning 60
4.2 Jet Reconstruction 61
4.2.1 Jet Energy Scale 62
4.2.2 Jet Energy Resolution 64
4.3 Event and Jet Selection 66
4.3.1 CMS Data Taking Runs and Luminosity Sections 66
4.3.2 Trigger Requirements 67
4.3.3 Vertex Requirements 68
4.3.4 Missing Transverse Energy Selection 69
4.3.5 Jet Selection 69
4.3.6 Summary 71
4.4 The Inclusive Jet Spectrum 72
4.4.1 Unsmearing the Spectrum 73
4.5 Systematic Experimental Uncertainties 74
4.5.1 Jet Energy Scale Uncertainty 74
4.5.2 Jet Energy Resolution Uncertainty 76
4.5.3 Luminosity 76
4.6 Theoretical Predictions 76
4.6.1 Next-to-leading Order Cross-Section Calculations 77

4.6.2 NP Correction Factors and Uncertainties 78
4.6.3 Summary 78
4.7 Final Comparison of Theory Calculations and Data 82
4.8 Conclusions 82

5 Conclusions 85

A Appendix - Theoretical Details 87
A.1 Gell-Mann Matrices 87

B Appendix - Computing 89
B.1 Batch Systems - Resource Managers and Schedulers for Computing Clusters 89
B.1.1 Resource Manager 89
B.1.2 Scheduler 89
B.1.3 Pre and Post Execution Scripting 90
B.2 ViBatch - Configuration 90
B.2.1 Libvirt Preparation 92
B.2.2 Maui/Torque Setup 93
B.2.3 ViBatch Prologue Script 94
B.3 IC1 - Institutes Cluster at the SCC 99
B.4 Grid Services 100

C Appendix - Experimental Details 105
C.1 Datasets 105
C.2 Run and Luminosity Sections 106
C.3 Jet p_T Binning 107
C.4 Software 107
C.4.1 CMSSW 107
C.4.2 JUEZ 108
C.5 Official CMS Results 110

List of Figures 111

List of Tables 113

Bibliography 115

Introduction

To study the smallest known constituents of matter and their interactions, particle physics experiments employ energy densities comparable to the environment in the early universe shortly after the big bang. By exploring the properties of the fundamental forces in the universe and the building blocks of matter, physics cannot answer *why* the universe exists but *how* the universe developed to the current state.

The current descriptions of the laws of nature are very detailed and can precisely explain many experimentally observed processes. Over the last decades, the Standard Model of Particle Physics evolved to a consistent theory explaining the fundamental interactions between the elementary particles. Its predictions have been tested successfully in numerous experiments. Despite the accurate agreement of theoretical predictions from the Standard Model and experimental findings, the theory relies on several free parameters which cannot be determined from first principles. These parameters have to be determined experimentally and still open questions like the origin of mass and the quantum mechanical explanation of gravity have to be resolved.

The Large Hadron Collider (LHC) and the associated experiments at the Conseil Europèen pour la Recherche Nucléaire (CERN) are the next milestone in particle physics on the way to a more fundamental understanding of the properties of our universe.

The protons colliding in the experiments of the LHC are compounds of fundamental particles, the quarks and gluons. The high energetic interactions between them, described by quantum chromodynamics, are the basis for all subsequent physics processes, to be studied at CERN during the next decades. A detailed knowledge of the proton substructure described by the parton density functions (PDFs) is crucial for gathering a deeper understanding of the observations in the hard proton-proton collision processes. The deflected quarks and gluons form the deep inelastic scattering will hadronize to colour-neutral streams of particles, generally referred to as jets. The basics of quantum chromodynamics with its phenomenological models and the conception of jets are discussed in chapter 1 and the basic features of the LHC and the CMS experiment are depicted in chapter 2.

The comparison of measurements, applied on the vast amount of data the LHC produces, with the theoretical calculations require the availability of a complex and powerful computing infrastructure. The access to the data and the computing resources is provided by grid technologies. Thus, computing centres around the globe have to interoperate with each other using well defined grid interfaces. The compatibility to the Worldwide LHC Computing Grid (WLCG) via these interfaces relies on a consolidation to a basic computing infrastructure with a predefined operating system and software setup. These requirements are not always easy to provide for small grid entities like the IEKP which generally do not provide a large amount of computing resources, hosted on their own. Additionally, to ensure the flawless operation of the experiment software and therefore the validity of the gathered results, LHC physicists are also restricted to use a special software for their analyses with further constraints to the used computing environment. Both, the participation within the WLCG as well as the general complex compatibility issues coming with experiment software typically avoid for small research groups the use of local computing resources, which are shared between several different institutions.

By using virtualization technology, the requested homogeneous, grid and experiment compatible computing environment can be abstracted form the software environment on the underlying computing resources. With this abstraction it is possible to provide each user group its desired computing environment for e.g. providing grid computing nodes or experiment specific software setups within virtual appliances. Therefore the concept of virtualized worker nodes in standard batch system was developed and implemented within the scope of this theses and is presented in Chapter 3.

The observable for the measurement of the jet production rate at a specific luminosity is the double-differential inclusive jet cross-section (differential in rapidity and transverse momentum). Due to the high jet production rate, the determination of this cross-section is one of the first possible measurements at the LHC, providing deeper insights in the configuration of the PDFs and allowing the extraction of the strong coupling constant α_s, one of the free parameters of the Standard Model of Particle Physics. With the new energy scale available at the LHC, the validity of extrapolations from experiments with lower collision energies like LEP and Tevatron can be tested by comparing the recent next-to-leading order perturbative QCD predictions to the measured jet cross-section. During the commissioning of a new collider, the setup and conditions of the experiments change rapidly. A precise understanding of these conditions during the analyses is vital for a proper measurement. Chapter 4 describes the first measurement of the double-differential inclusive jet cross-section using the calorimeter system of the CMS detector with promising results in the comparison with the theory.

Quantum Chromodynamics - The Theory of the Strong Interaction

In the early days of Nuclear Physics, the binding energy of the nucleus had to be explained by a new type of interaction between protons and neutrons. Scattering experiments showed that the force of the interaction is very strong but covers only extremely short ranges. Yukawa introduced a massive boson as mediating quantum for this force, the so-called *meson*. In the 1940s these force carriers of the strong interaction, the π-mesons, were discovered in cosmic rays and first accelerator experiments. Surprisingly the experiments demonstrated, that there are a lot more particles to be found as results of nuclear interactions in a wide variety of masses and lifetimes. The decays of the long-lived particles were accounted to the weak interaction, while particles with very short lifetimes decay via the strong force.

By the mid 1960s, Nuclear Physics experiments evolved to higher energies and particle physicists were faced with a *particle zoo* containing a vast amount of different particles. However, with the growing reach in energy and precision enough information had been gathered and was interpreted independently by Gell-Mann and Ne'eman [1] in an ansatz, where the known hadrons are classified as multiplets of the unitary Lie group SU(3). The particle zoo could now be sorted in a kind of periodic table of elementary particles. The discoveries of particles like the Ω^-, predicted by this new framework, in 1963 at Brookhaven supported the classification of hadrons using a group theory approach.

The higher dimensional representations of the SU(3) became occupied by more and more discovered particles. However, the fact that the fundamental ones remained empty lead to doubts about the theory. Finally, Gell-Mann and Zweig [2] postulated independently without any experimental evidence that the fundamental representations of the SU(3) should also have its respective particles called *quarks*. These quarks are the elementary particles of which all hadronic matter is made of and indications of the quarks confined to dimensions of hadronic matter have been first observed in 1968 at the SLAC[1] in deep inelastic scattering experiments.

Today the high energetic interactions of hadrons are studied in collider experiments at the Tevatron and the Large Hadron Collider (LHC). Quantum chromodynamics (QCD) and the Glashow-Salam-Weinberg-(GSW)-Theory [3–5], a unification of quantum electrodynamics and the weak interaction, form the mathematical framework for an internally consistent theory, the Standard Model of Particle Physics. Except gravity it describes all interactions between the fundamental particles. In the following a short introduction into the Standard Model of Particles will be followed by a more detailed summary of Quantum Chromodynamics with focus on the particularities of hadron collider physics like the LHC. Both is given in detail in textbooks like [6, 7]).

1.1 The Standard Model of Elementary Particle Physics

In the past century, beginning with Ernest Rutherford's famous gold foil experiment, physicists advanced deeper into the structure of matter. The higher the energy of the probing particles in a scattering experiment the smaller the structures which can be investigated as the wavelength of a particle is inverse proportional to its relativistic energy [8]. Particle accelerators reached centre-of-mass energies, enabling them to probe the sub-structure of nucleons. It turned out that QCD and GSW, combined in the Standard Model of Particle Physics, are able to explain and even predict the existence of the fundamental particles and their interactions except gravity.

fermions	generation 1 2 3	electromagnetic charge [units of e^+]	colour charge	spin J
quarks	u c t	$+2/3$	r,b,g	$1/2$
	d s b	$-1/3$	r,b,g	$1/2$
leptons	ν_e ν_μ ν_τ	0	-	$1/2$
	e μ τ	-1	-	$1/2$

Table 1.1: *Fundamental particles of the Standard Model*

Within the Standard Model there are twelve fundamental particles representing the building blocks of all known matter in the universe (Tab. 1.1). Each of these fundamental particles has

---

[1] Stanford Linear Accelerator Collider

interaction	exchanged charge	gauge boson	gauge boson mass (GeV/c^2)
strong	colour	8 gluons (g)	0
electromagnetic	electric charge	photon (γ)	0
weak	weak charge	$W^{\pm}$,Z^0	$\approx$ 100

Table 1.2: *Fundamental Interactions and the respecting gauge vector bosons of the Standard Model*

its own anti-particle with the same mass but opposite electric and colour charge as well as an opposite component of the weak isospin. The particles are arranged in three particle generations, out of which only the first one builds up stable matter. Matter made of fundamental particles of the second and third generations is generally unstable and exists only temporarily after its production in natural phenomena like supernovae or in experiments within particle colliders, where the required energy densities are available.
The interactions between the elementary particles are described by the Standard Model of Particle Physics as an exchange of dedicated gauge vector bosons as listed in Tab. 1.2.

The experimental observations of the $W^{\pm}$ and the Z^0 bosons, the gluon, the top quark and the charm quark and the measurements of their properties and the comparison with the theoretical predictions lead to increasing confidence in the Standard Model.
However, there are several free parameters to be determined by experiments:

- The masses of the leptons and the quarks.
- The weak mixing angle.
- The still unknown mass of the Higgs boson.
- The coupling constants α_{em} and α_s of the electromagnetic and the strong force and the Fermi coupling constant G_F.
- The transition matrix elements of the electroweak interaction, contained in the CKM[2] matrix.

1.2 Cross Sections

Since Rutherford's gold foil experiment, the standard tool to probe the structure of matter is the scattering of probe and target entities. To compare experimental results with the theory

---

[2] Cabibbo-Kobayashi-Maskawa matrix

predictions, the cross-section of a physics process the fundamental observable. It is defined as the interaction rate per target particle W normalized to the incoming flux Φ of probe particles:

$$\sigma = \frac{W}{\Phi} \tag{1.1}$$

The commonly used unit of the cross-section is the *barn*, where 1 b = 10^{-28} m^2. The theoretically predicted cross-section for a specific physics process can be compared to the measurement and deviations from the theory will then be reflected by a different counting rate for this process. In collider experiments the cross-section is used differentially for varying phase space quantities such as the transverse momentum p_T and the pseudo-rapidity[3] η or the rapidity[4] y.

By using the quantum-mechanical transition matrix elements $|\mathcal{M}_{fi}|$

$$\mathcal{M}_{fi} = \langle \Psi_f | \mathcal{H}_{int} | \Psi_i \rangle. \tag{1.2}$$

which are derived from the interaction Hamiltonian and the energy density within the available phase space p_f, the interaction rate can be calculated with Fermi's Golden Rule [9]

$$W = \frac{2\pi}{\hbar} |\mathcal{M}_{fi}| p_f. \tag{1.3}$$

To estimate the number of produced events, and therefore the outgoing particle flux, the machine properties of the colliders have to be incorporated. This is done via the Luminosity $\mathcal{L}$, a quantity depending on beam parameters like protons per bunch and total number of bunches, the beam optics, and the energy of the beam. During an LHC run, the instantaneous luminosity is experimentally determined to estimate the cross-section by using:

$$\sigma = \frac{W}{\mathcal{L}}. \tag{1.4}$$

According to the Optical Theorem, the total proton-proton cross-section is related to the elastic forward scattering amplitude $f_{el}(0)$. Forward detectors, operating close to the beam pipe, like the TOTEM experiment [10], can measure the elastic and inelastic collision rates (N_{el}, N_{inel}). Combined with the differential elastic rate dN_{el}/dt at a squared four momentum transfer $t = 0$ the total cross section and the luminosity can be derived:

$$\sigma_{tot} = \frac{16\pi}{1+\rho^2} \cdot \frac{dN_{el}/dt|_{t=0}}{N_{el}+N_{inel}}, \mathcal{L} = \frac{1+\rho^2}{16\pi} \cdot \frac{N_{el}+N_{inel}}{dN_{el}/dt|_{t=0}}, \text{ with } \rho = \frac{\Re[f_{el}(0)]}{\Im[f_{el}(0)]}. \tag{1.5}$$

---

[3] $\eta = -\ln\left[\tan\frac{\Theta}{2}\right] = \frac{1}{2}\ln\frac{|\vec{p}|+p_z}{|\vec{p}|-p_z}$

[4] $y = \frac{1}{2}\ln\frac{E+p_z}{E-p_z}$, p_z is the momentum component along the beam axis.

In the beginning of a collider experiment ρ has to be calculated from theory, but can later be determined experimentally by distinguishing the hadronic and the Coulomb contributions to the elastic scattering cross-section in interference measurements.
Another method to derive the luminosity or to cross check the results from the forward detector measurement is to use precisely measured process signatures like the Drell-Yan muon pair production. Via

$$\int \mathcal{L} dt = \frac{N_{DY}}{\sigma_{DY}} \tag{1.6}$$

and the very precise knowledge of the Drell-Yan muon pair production cross-section σ_{DY} the luminosity can be estimated.

1.3 Quantum Chromodynamics

By steadily increasing the collision energy in particle accelerators during the last half of the twentieth century, the gathered experimental evidence for a substructure of the proton, resulting in the quark parton model (later described in detail in section 1.3.5), the new particles and their interactions had to be explained by a new theory. The theory of QCD describes both, the existence of hadrons made of quarks and gluons and, in a reduced form, the attractive binding force between nuclei.
The discovery of particles like the Δ^{++}, a bound state of three up quarks with parallel spin, required the introduction of a new quantum number to fulfil Pauli's exclusion principle. This quantum number had to provide three additional possible quantum states, to allow an anti-symmetric wave function for a bound state of three identical quarks with parallel spin. Analogous to the additive colour model, QCD introduces three colours (*red*,*green*,*blue*) and their respective anti-colours. Each bound state has to be colour neutral (*white*). This can be achieved by combining three quarks (or anti-quarks) of different (anti-)colour resulting in (anti-)baryons, or by a combination of one quark and one anti-quark (with the corresponding anti-colour), the Mesons.
The combination of the 3×3 different colour states results in a colourless singlet, which is not contributing to any interactions, and an octet representing the SU(3), as shown in Table 1.3. These octet states are the exchange bosons mediating the strong interaction and are named gluons.
The number of colours is not determined from within the theory. Measurements of the decay widths of processes like the $\pi^0 \to \gamma\gamma$ depend on the number of possible quark colour states N_c [11]:

$$\Gamma(\pi^0 \to \gamma\gamma) = 7.63 \text{ eV} \left(\frac{N_c}{3}\right)^2 \tag{1.7}$$

The measurement of 7.84 ± 0.56 eV indicates $N_c \approx 3$.
By estimating the ratio between multi-hadron and myon pair production in electron-positron annihilation, two very similar processes, the relative coupling strengths can be compared and the number of quark colours can be distinguished:

$$R_\gamma = \frac{\sigma(e^+e^- \to \text{hadrons})}{\sigma(e^+e^- \to \mu^+\mu^-)} = N_c \sum_q e_q^2 = N_c \frac{11}{9} \tag{1.8}$$

The measurements yield $N_c \approx 3.2$. The deviation of 0.2 can be accounted for by higher order QCD corrections.

Table 1.3: *Possible representation of the colour singlet and octet of the SU(3) symmetry.*

Symmetry	representation							
Singlet	$\sqrt{1/3}\,(r\bar{r} + g\bar{g} + b\bar{b})$							
Octet	$r\bar{g}$	$r\bar{b}$	$g\bar{b}$	$g\bar{r}$	$b\bar{r}$	$b\bar{g}$	$\sqrt{1/2}\,(r\bar{r} - g\bar{g})$	$\sqrt{1/6}\,(r\bar{r} + g\bar{g} - 2b\bar{b})$

1.3.1 The QCD Lagrangian

Analogous to QED, QCD was formalized as a quantum field theory, more specifically a Yang-Mills gauge theory. Implying the summation over repeated indices, the QCD Lagrangian can be written as

$$\mathcal{L}_{QCD} = \sum_q \bar{q}_i (i\gamma^\mu D_\mu - m_q)_{ij} q_j - \frac{1}{4} F^a_{\mu\nu} F^{a\,\mu\nu} \tag{1.9}$$

whereas the field strength tensor $F^a_{\mu\nu}$ and the covariant derivatives D_μ are given by:

$$\begin{aligned} F^a_{\mu\nu} &= \partial_\mu A^a_\nu - \partial_\nu A^a_\mu - g f^{abc} A^b_\mu A^c_\nu && (1.10) \\ (D_\mu)_{ij} &= \delta_{ij}\partial_\mu + i g_s T^a_{ij} A^a_\mu && (1.11) \\ (m_q)_{ij} &= m_q \delta_{ij} && (1.12) \end{aligned}$$

The A^a_ν represent the gluon fields, g_s the gauge coupling, f^{abc} the structure constants and T^a_{ij} the generators of the gauge symmetry SU(3) group. The only free parameters in this theory are the coupling constant g_s and the fermionic masses. The gauge coupling translates into the strong coupling constant as $\alpha_s = \frac{g_s^2}{4\pi}$. As mentioned the experimental findings implied the assumption, that the underlying symmetry of a Yang-Mills theory is represented by the SU(3). The eight Lie group generators for the SU(3) can be written as $T^a = \frac{\lambda_a}{2}$ with the Gell-Mann matrices λ_a (See Appendix A.1). For details on the structure constants and group theory see appropriate text books. The appearance of the coupling along with the propagators of quarks and gluons in equations 1.10 and 1.11 shows, that there are three fundamental QCD processes which can be observed, the gluon radiation and three and four gluon vertices, as shown in Fig. 1.1.

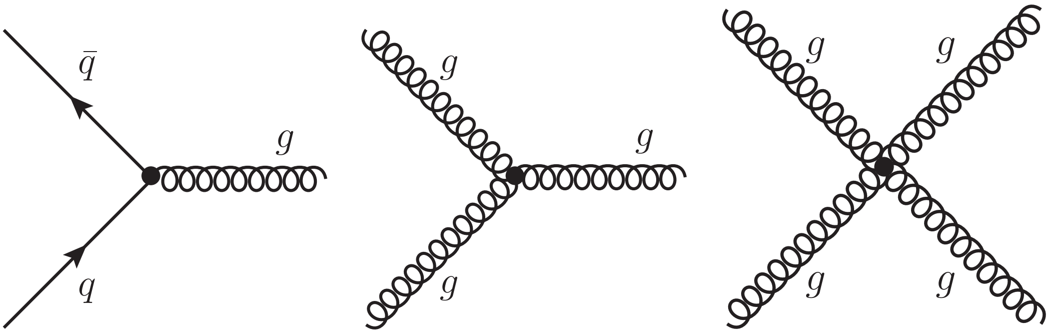

Figure 1.1: *Feynman graphs of the basic QCD interactions*

1.3.2 The Running Coupling

The coupling constant describes the interaction between two particles. However, the constants in quantum field theory depend on the momentum transfer Q^2. In QED the dependence is very small, while in QCD it is very strong due to the gluon self coupling. In both cases, the contributions of vacuum fluctuations result in a screening of the respective interaction. For higher Q^2, the coupling constant increases due to the fact that the effective charge of the participating particles gets larger, as the distance between them decreases. However, in case of the strong interaction gluons can fluctuate into gluons which leads to an anti-screening of the strong coupling. For gluons the anti-screening prevails over the screening effect and therefore the α_s decreases for higher Q^2.

In the mathematical description the strong coupling is represented by a renormalization group equation

$$\frac{\partial \ln \alpha_s(Q^2)}{\partial \ln Q^2} = \frac{\beta(\alpha_s(Q^2))}{\alpha_s(Q^2)}, \tag{1.13}$$

which can be written as

$$\beta(\alpha_s(Q^2)) = Q^2 \frac{\partial \alpha_s(Q^2)}{\partial Q^2}. \tag{1.14}$$

This defines the mass independent β function where one can perform an expansion in α_s and an evolution in terms of Q^2:

$$\beta(\alpha_s(Q^2)) = -\frac{\beta_0}{4\pi}\alpha_s^2(Q^2) - \frac{\beta_1}{8\pi^2}\alpha_s^3(Q^2) + \mathcal{O}(\alpha_s^4(Q^2)) \tag{1.15}$$

The coefficients β_i can be calculated by using specific schemes (e.g. the $\overline{MS}$ scheme). With the number of quark flavours n_f for a given Q^2, β_0 can be calculated by using one-loop QCD theory:

$$\beta_0 = 11 - \frac{2}{3}n_\mathrm{f}. \tag{1.16}$$

Now, the one-loop solution for α_s reads

$$\alpha_s(Q^2) = \frac{\alpha_s(Q_0^2)}{1 + \alpha_s(Q_0^2)\frac{\beta_0}{4\pi}\ln\frac{Q^2}{Q_0^2}}, \tag{1.17}$$

with Q_0^2 being the energy scale, used in the renormalization. This function is steadily falling with increasing Q^2 for the non-Abelson SU(3)-β function being negative ($\forall\ n_f < 17$). Currently it is common to use the mass of the neutral Z-Boson M_Z as energy scale during the renormalization. According to the particle data group [12] the 2010 world average is:

$$\alpha_s(M_Z^2) = 0.1184 \pm 0.0007 \tag{1.18}$$

In Fig. 1.2 the different measurements extrapolated to $Q_0 = M_Z$ are given in detail on the left side. On the right hand side of this figure, the measurements are shown at their respective scale. Here, the running behaviour of α_s shows that the strength of the interaction decreases with increasing energy scale. This fact is known as *asymptotic freedom* and allows to treat quarks as quasi free particles at high enough interaction energies, and can be well described by using perturbative QCD.

The special feature of QCD, the self-coupling of the gluons, results not only in the asymptotic freedom. When quarks are separated, the energy in the colour field between the quark pair increases. At a certain threshold, enough energy is gathered in the field to create a new colourless quark anti-quark pair. Due to this behaviour, the so-called *confinement*, free quarks are never observed.

1.3.3 Hadrons

Due to confinement, up, down, charm, strange and bottom quarks *hadronize* (see section 1.4.2) if separated from their partners and therefore are confined within colourless singlets

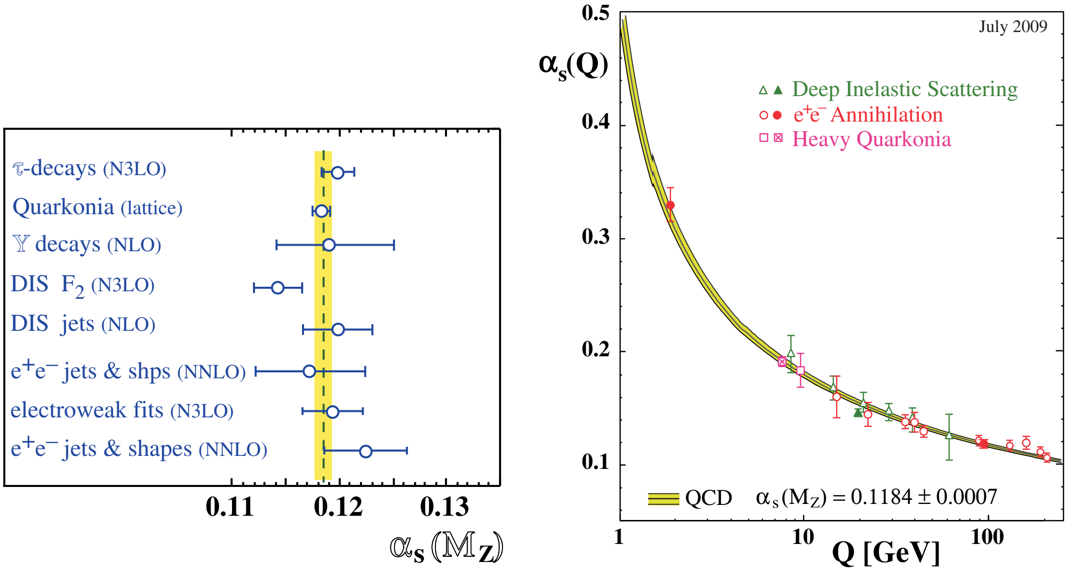

Figure 1.2: *World average of α_s is shown on the left as a summary of various measurements extrapolated to $Q^2 = M_Z$. The running of α_s according to measurements is shown on the right. The yellow band marks the $\pm 1\sigma$ limit. From [13].*

called *hadrons*. The top quark instead decays very fast and has no time to hadronize. At the moment, two different hadron sub-groups are known, the mesons and the baryons. By adding up the properties of the constituent quarks, following the conservation laws, the effective properties of the hadron can be constructed. According to the particle data group about one hundred different hadrons have been observed and their properties have been measured by studying their decay products. The mesons and baryons made of heavy quarks are very unstable, resulting in typical lifetimes between $10^{-7}s$ and $< 10^{-20}s$, depending if they decay electromagnetic, weak or via the strong interaction. As long as bound within a nucleus, proton and neutron are stable and are therefore the basic building blocks of all stable matter in the universe. As free neutrons decay with a half-time of about 15 minutes into a proton an electron and an anti-electron-neutrino, protons are the ideal choice for studying baryons in detail. By using electron probes with a sufficient energy ($\approx 5\,\text{GeV}$) in so-called deep inelastic scattering (DIS) experiments, the proton substructure can be revealed.

1.3.4 Deep Inelastic Scattering

In deep inelastic electron-proton scattering experiments, the high energetic electron exchanges a virtual photon with the proton. The proton is destroyed and short-living excited hadrons are produced as sketched in Fig. 1.3. The deep inelastic scattering process deviates from the elastic scattering by the momentum fraction which the point-like particle ,called *parton*, interacting with the photon is carrying described by the *Bjorken scale*

$$x := \frac{Q^2}{2Pq} = \frac{Q^2}{2M\nu} \tag{1.19}$$

with the momentum transfer Q^2 between photon and proton, the mass of the proton M and the Lorentz invariant energy transfer (or excitement energy) $\nu = \frac{Pq}{M} = E - E'$ in a specific reference system. Within the calculation of the cross-section for the excited states in the DIS, the mass of the resonance expressed by ν has to be considered:

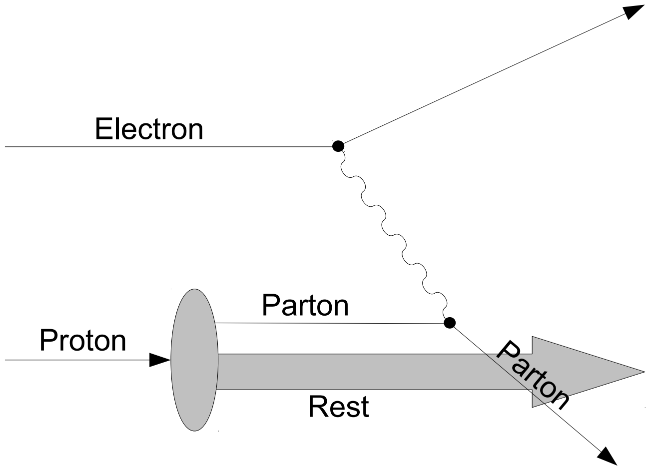

Figure 1.3: *Schematic representation of deep inelastic electron-proton scattering.*

$$\begin{aligned} \frac{\mathrm{d}^2\sigma}{\mathrm{d}\Omega \mathrm{d}E'} &= \left(\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}\right) \cdot \left[W_2(Q^2,\nu) + 2W_1(Q^2,\nu)\tan^2\frac{\theta}{2}\right] && (1.20) \\ W^2c^4 &= P'^2 = (P+q)^2 = M^2c^4 + 2Pq + q^2 = M^2c^4 + 2Mc^2\nu - Q^2 && (1.21) \end{aligned}$$

By using the Bjorken scale, the form factors W_1 and W_2 can be translated to:

$$\begin{aligned} F_1(x,Q^2) &= Mc^2W_1(Q^2,\nu) && (1.22) \\ F_2(x,Q^2) &= \nu W_2(Q^2,\nu). && (1.23) \end{aligned}$$

Using this, the cross-section of the electron-proton DIS reads as

$$\frac{\mathrm{d}^2\sigma}{\mathrm{d}x\mathrm{d}\hat{y}} = \frac{4\pi\alpha^2(s-M^2)}{Q^4} \cdot \left[(1-\hat{y})F_2(x) + \hat{y}^2 xF_1(x) - \frac{M^2}{s-M^2}x\hat{y}F_2(x)\right] \tag{1.24}$$

with the inelasticity $\hat{y} = \nu/E$.
In this model the structure functions can be redefined by absorbing the target mass M and the energy transfer ν to be

$$\begin{aligned} F_1(x) &= \frac{1}{2}\sum_i e_i^2 f_i(x) && (1.25) \\ F_2(x) &= \sum_i e_i^2 f_i(x). && (1.26) \end{aligned}$$

with the functions $f_i(x)$ which give the probability that the quark i carries the momentum fraction x. These functions are called *parton distribution functions* (PDFs) and F_1 measures the parton density and F_2 describes the momentum density, both weighted with the coupling strength to the photon probe.
During early electron-proton experiments in the late 60s of the last century it was found, that the scaling behaviour of the measured structure functions became independent from Q^2 [14] which finally lead to a pre-QCD quark parton model [15, 16]. In this quasi-classical model the leptons scatter on the independent partons by the exchange of a vector boson. The *Callan-Gross-relation* [17]

$$2xF_1(x) = F_2(x) \tag{1.27}$$

is valid only for fermionic particles. DIS experiments therefore confirmed the quarks to be fermions. By adding quantum numbers to the partons, the *constituent quark model* was formulated. The fact, that the parton masses are negligible for high Q^2 implies, that the Bjorken scale can be interpreted to be the momentum fraction of the proton carried by the interacting quark. Measurements of the proton momentum fractions of the quarks showed, that only about half of the momentum is carried by them. This was the experimental hint to the existence of gluons as additional constituents in the proton.

1.3.5 QCD Improved Parton Model

QCD and and the improved constituent quark model together explain the independence of the proton constituents in DIS experiments by the asymptotic freedom and the missing transverse momentum fraction in the hadron is accounted to the gluons. The previously mentioned PDFs in Eq. 1.25 and Eq. 1.26 of the quark-parton model are independent of the scale Q. To explain the quantum theoretical features of the hadron structure, Altarelli used perturbative QCD (pQCD) [18]. As a result the short range interactions of the specific parton and the large scale distance interactions from the hadron can be separated. Through this separation, the PDFs became scale dependent, e.g. for high Q^2 where the quark might be extracted

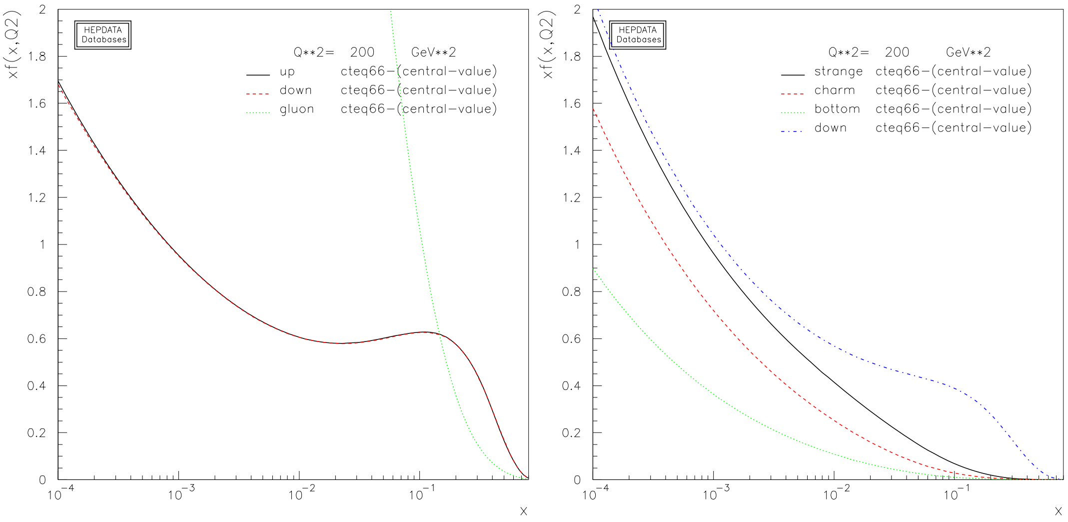

Figure 1.4: *CTEQ6.6 parton momentum distributions for a scale of* $Q^2 = 200$ *GeV. Derived with the online tool of the Durham university [23]. A large fraction of the proton momentum is carried by low-*x *gluons (green line on the left).*

from the proton, its momentum fraction differs from is momentum when it interacts with the photon, as it may have radiated gluons. Dorkshitzer, Gribov, Lipatov, Altarelli and Parisi introduced the DGLAP equations [19–21], which describe the evolution of the PDFs with Q^2 and Bjorken x. Unfortunately, it is not possible to derive PDFs from first principles using perturbative methods. However the PDFs retrieved from measurements can be extrapolated to other scales by using the DGLAP equations [22]:

$$\frac{\partial q_i(x,\mu^2)}{\partial \log \mu^2} = \frac{\alpha_s}{2\pi}\int_x^1 \frac{dz}{z}\left(P_{q_iq_j}(z,\alpha_s)q_j(\frac{x}{z},\mu^2) + P_{q_ig}(z,\alpha_s)g(\frac{x}{z},\mu^2)\right) \quad (1.28)$$

$$\frac{\partial q_i(x,\mu^2)}{\partial \log \mu^2} = \frac{\alpha_s}{2\pi}\int_x^1 \frac{dz}{z}\left(P_{gq_j}(z,\alpha_s)q_j(\frac{x}{z},\mu^2) + P_{gg}(z,\alpha_s)g(\frac{x}{z},\mu^2)\right). \quad (1.29)$$

Here, z is the momentum fraction of the parton after radiation and the *splitting functions* $P_{ab}(z,\alpha_s(\mu^2))$ describe the possibility of radiating a gluon. The expansion in powers of α_s of these splitting functions reads:

$$P_{ab}(x,\alpha_s(\mu^2)) = P^0_{ab}(x) + \frac{\alpha_s}{2\pi}P^1_{ab}(x) + ... \quad (1.30)$$

The PDFs are continuously improved integrating new measurements into global fits performed by collaborations, e.g. MSTW or CTEQ [24]. Fig. 1.4 shows an excerpt of the recent CTEQ6.6 PDF set, plotted by using the online tool of the Durham university [23]. On the left hand side it becomes obvious, that a large fraction of the proton momentum is carried by

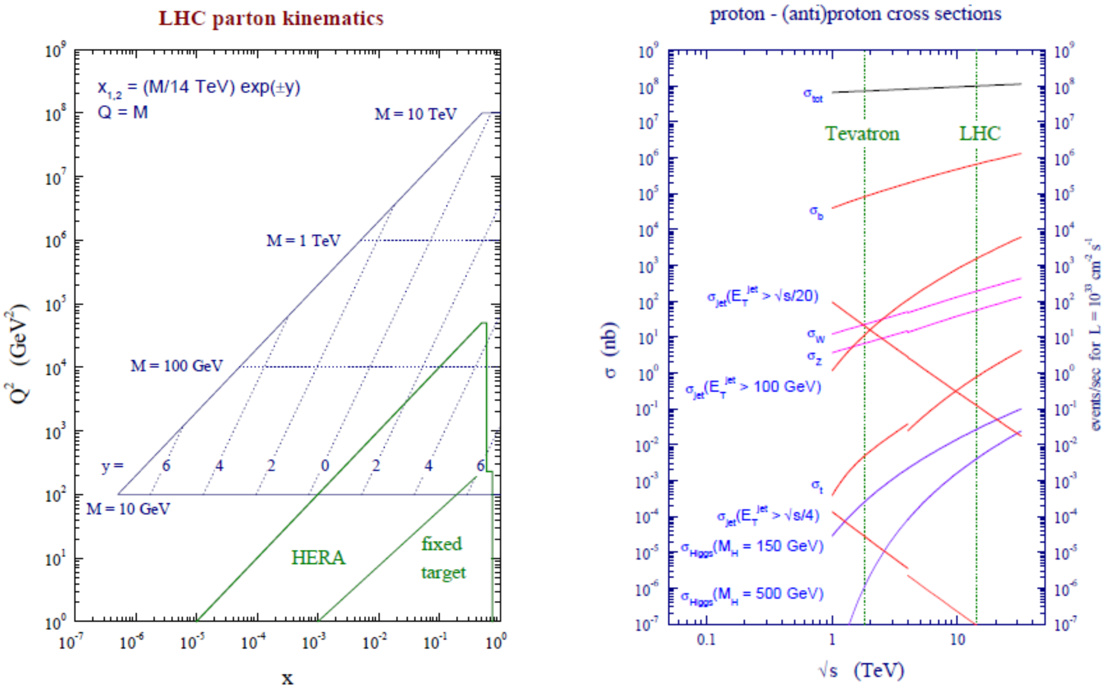

Figure 1.5: *Parton kinematics and cross-sections at Tevatron and the LHC. Taken from [25].*

low-x gluons.

1.4 From Quarks to Hadrons

The process from the initial hard interaction to the final particles, measured in the detector, should be theoretically calculable using QCD and GSW. For energies greater than about 4 GeV the QCD estimations can be done using perturbative methods, which are, unfortunately, very computing intensive. Therefore in most cases the leading order (LO) or next-to-leading order (NLO) approximations are used. Very few of the processes can be studied with higher order. All these calculations represent the process for the actual hard interaction, but due to confinement, which does not allow coloured objects like quarks or gluons to be separated from each other by distances larger than about 1 fm, the products of the initial hard interaction decay collinearly, while the energy invested in the colour field between these partons is used up to create quark anti-quark pairs from the vacuum until all the partons have been integrated into colourless hadrons as already pointed out before. These collinear streams of hadrons are named jets. To emulate higher order processes and to describe the production of the particle jets, additional theoretical frameworks are needed. For processes,

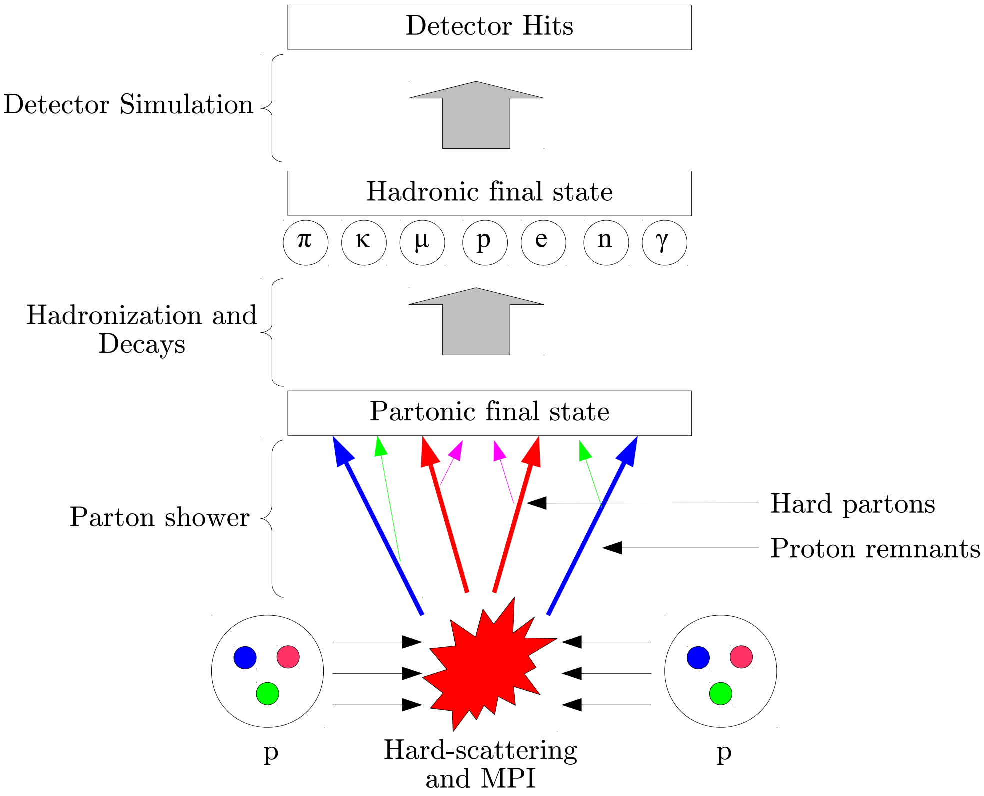

Figure 1.6: *Event simulation of multi purpose generators.*

where the energy of the particles is high enough, the parton shower model describes the collinear decays. Below the energy limit where perturbative methods can be used, phenomenological hadronisation models have been tuned to describe experimental data. These models have been applied to the Monte Carlo event generation procedure as shown in figure 1.6. At the LHC, each initial process for the produced particles is a strong interaction, and these calculations are therefore of major interest. The collimated streams of hadrons carry the information about the initial strong interaction. For that reason, all particles which are supposed to come from the same hard parton are grouped together into jets by special algorithms. These should be able to work on different types of input at different levels of either Monte Carlo generated particles or reconstructed detector hits. To assure this, the input objects must have comparable properties. Generally, they are described by four-vector containing a three dimensional direction information and an energy component. Both can be derived for the respective object by calculating the particle properties or from calorimeter measurements. The jet algorithms have then to measure the distances between objects and then to define a separation between them or they have to be combined via a special recombination scheme.

Different types of jet algorithms are in common use: Cone type algorithms [26] cluster the objects together that are close in angle around a high-energetic seed, and subsequent clustering algorithms like the k_T algorithm [27], which iteratively combine objects that have the smallest distance of all possible pairwise combinations. Throughout this work the anti-k_T algorithm has been used and is described in section 1.5.3.

1.4.1 Parton Shower

In the 1970s and 1980s the parton-shower approach for the description of jet radiation has been developed. By the use of the parton showering process a few partons produced in the hard interaction at a high energy scale can be related to partons at an energy scale close to Λ_QCD via the DGLAP equations (eq. 1.29). Using a universal non-perturbative model the transition from the hard partons to the experimentally observed hadrons can then be calculated analytically or numerically following and evolution equation. The solution of this evolution equation can be transformed into a Sudakov form factor [28] given in (1.31), which represents the probability of evolving from a higher scale to a lower scale without the emission of a gluon greater than a given value.

$$\Delta(t) = \exp\left[-\int_{t_0}^{t}\frac{dt'}{t'}\int\frac{dz}{z}\frac{\alpha_s}{2\pi}\frac{f(x/z,t)}{f(x,t)}\right] \tag{1.31}$$

Parton showering in principle introduces two new scales, one for the initial-state parton showering, where the evolution proceeds backwards from the hard scale t of the process to the cutoff scale t_0, with the Sudakov form factors being weighted by the parton distribution functions at the relevant scales, and one for the shower in the final state. In the final state the Sudakov form factor has a similar form but without the PDF weighting. With the introduction of the Sudakov form factor, all the effects of soft and collinear gluon emission are resumed. Thus, the predictions are well-defined even in these regions. By applying this method, the non-singular contributions of large energy, wide angle gluon emissions are omitted even though the soft and collinear regions are the dominant effect as they are logarithmically enhanced.

The probability functions provided via the Sudakov form factors are used in Monte-Carlo Generators to simulate the chain of sequential radiations. Within the parton showering process, successive values of an evolution variable t, a momentum fraction z, and an azimuthal angle ϕ are generated, along with the flavours of the partons emitted during the showering. Different choices for the evolution variable t is are made within the various implementations in PYTHIA [29, 30], HERWIG [31] or HERWIG++ [32]. Additionally, these Monte-Carlo Generators apply different methods to describe an angular ordering of the emission which represents an attempt to simulate more precisely those higher order contributions that are enhanced due to soft gluon emission (colour coherence).

While Fixed-order calculations explicitly account for colour coherence, parton shower Monte-Carlo models including colour-flow information are only an approximation. This so-called *leading-log-approximation* cancels all but the leading terms of the perturbative expansion due to the cut-off parameter t_0 . With increasing angle the emitted energy decreases in a logarithmic way. *Thin* jets emitt most gluons in the forward direction, and are therefore only less affected by this cut. As mentioned, wide angle emissions are not correctly described by this method. Thus, additional models are implemented in the generators to fill the respective phase space.

Several new approaches [33] are now available to describe higher jet multiplicities via the parton shower models. By attaching parton shower models to multijet matrix-elements (tree level) or NLO loop calculations the problem of double counting arises. Certain contributions occur twice. Once in the higher orders of perturbation or in the extended matrix element, and additionally they are emulated in the higher orders by the parton shower model. Two different approaches cope with this problem. The CKKW (Catani, Krauss, Kuhn, Webber [34]) method exploits a k_T algorithm (see section 1.5.3) on the higher multiplicity matrix level calculations in order to eliminate infrared divergences, and to apply an additional re-weighting of the decay histories. The MLM (M.L. Mangano) [35] prescription, a similar method, identifies the double-counted contributions by using a cone based algorithm.

1.4.2 Hadronisation

Below the interaction energy limit $t_0 \approx 1 - 4$ GeV within the decay chain, α_s becomes too large and parton shower calculations are no longer applicable. Therefore, hadronisation or fragmentation models are applied to describe the production of *final state* hadrons, thus cancelling the dependence on t_0. It is assumed, that the hadronisation process, operating at low momentum transfers, should not affect the large scale perturbatively determined topology of the event. Especially in the context of precision measurements, the non-perturbative effects cannot be neglected. To estimate the theoretical uncertainties, founded in the absence of an exact theory describing the hadronisation process, one has to consider several of the available hadronisation models. The implementations of the models are tuned to describe the data of former and current experiments in a constant process and the extrapolations are steadily tested within comparisons with actual measurements [36]. The two models used within the scope of this work are:

- Cluster Fragmentation:
  Implemented in the simulation program HERWIG and HERWIG++, the currently best known cluster fragmentation model [37] is based on the pre-confinement property of the angular-ordered parton shower. It takes the remnants of the preceding parton shower and builds colour singlet clusters of $q\bar{q}$-pairs. These are then decayed into hadrons following flavour conservation. Light quark anti-quark pairs are produced by splitting the gluons isotropically which remained at the end of the parton shower process. Then, they are bound together to colour singlet states with neighbouring quark anti-quark pairs with the also remaining quarks from the parton showers. These singlets are formed into clusters, with the momentum given by the sum of the constituent partons. According to the Pre-confinement, pairs of colour-connected neighbouring particles have the tendency to be arranged within limited extension in both coordinate and momentum space. Thus, the formed clusters have a mass distribution that falls rapidly with growing masses and is asymptotically independent of the overall energy scale Q^2 and the hard interaction. Typical masses of a few GeV suggest that the clusters can be associated with *super-resonances* which decay independent of each another into hadron resonances. This is realised by ignoring spin correlations which leads mainly to an isotropic two body decay of a cluster into two hadrons. The constituent flavours are conserved an additional quark anti-quark pair is popped from the vacuum to fill the empty spaces during the decay. These new quarks are created to grant a correct colour flow. In order to fulfil the respective conservation laws spin, momenta, and masses are adjusted accordingly. In cases where the clusters are too light for a two body decay, they are decayed into a single hadron. The spillover mo-

menta are redistributed among neighbouring clusters. For the case where the mass of a cluster is too large and therefore an isotropic decay is no longer applicable, an anisotropic fission mode is used to split the clusters into two lighter ones by popping a quark anti-quark pair from the vacuum. This is repeated recursively until the clusters are allowed to decay into final hadrons via the isotropic method. The cluster content of heavy quark flavours like charm or bottom quarks is fixed by an additional model, to reflect multiplicities observed in experiments. According to probability functions derived from experimentally determined branching ratios as they are provided by the PDG the flavours of the final particles are assigned.

- Lund/String Fragmentation:
  The Lund/String fragmentation model was initially developed at the University of Lund [30]. It is one of the most popular string hadronisation models and it is implemented in the multi purpose generator PYTHIA. The motivation for this model is given by the properties of the colour field between two coloured particles which loose energy to the colour field between them, as they move apart. Due to self-interactions this field is supposed to collapse into a string-like configuration with a uniform energy per unit length, or string tension of about $\kappa \approx 1\ \mathrm{GeV/fm} \approx 0.2\ \mathrm{GeV}^2$, which corresponds to a linear confining potential like the QCD confinement for quarks. The transverse size of a string, is therefore negligible. This allows the dynamics to be described by a massless, one-dimensional, relativistic string possessing no transverse excitations. The equations of motion for the string result in a so-called yo-yo mode, where a massless quark anti-quark pair which forms the endpoints of the string oscillates repeatedly outwards and inwards at the speed of light, passing through each other while transferring energy to and from the string. Colourless string segments are formed between neighbouring partons of the parton shower output. Each segment terminates on a quark and an anti-quark. So-called *kinks*, which represent gluons, are an additional solution to the string model. The stronger the kink, the harder is the radiated gluon. Therefore, soft and hard radiation of gluons are included in this model. Following certain flavour and spin selection rules hadrons are produced as describe by the iterative spontaneous quark anti-quark production in the intense colour field, which breaks the string more and more apart.

  The pair production invokes quantum mechanical tunnelling. Without a common mass or transverse momentum the q and $\bar{q}$ can classically be created at one point and then pulled apart by the field. In contrast to that the quark and anti-quark must classically be produced at a certain distance if they have a mass and/or transverse momentum, so that the field energy between them can be transformed into the sum of the two transverse masses $\mathrm{m}_{\perp}$. In order to sustain local flavour conservation the quarks may be created quantum mechanically in one point, and then tunnel out to the classically allowed region. The tunnelling probability is given as

$$\exp\left(-\frac{\pi \mathrm{m}_\perp^2}{\kappa}\right) = \exp\left(-\frac{\pi \mathrm{m}^2}{\kappa}\right) \exp\left(-\frac{\pi \mathrm{p}_\perp^2}{\kappa}\right) \tag{1.32}$$

in terms of $\mathrm{m}_\perp$ of the quark and anti-quark.

By the model described above in principle only mesons can be produced. However, also baryon production is described, as the q or $\bar{q}$ can also be a diquark system. Via the tunnelling model the quark flavour is chosen while suppressing heavy quarks as indicated in eq. 1.32. The amount of the longitudinal momentum of the produced hadron is determined by the Lund symmetric fragmentation function. Finally, a choice for the properties of the formed hadrons has to be made. I an attempt to model the supposed dynamics via flavour and spin selection rules the Hadrons are formed. Therefore quite a lot of parameters need adjustment, so that the produced state is compatible with observations, as done in the so-called tuning. Additional features are build into the baryon production like the handling of diquarks, producing qq' $\bar{q}\bar{q}'$ pairs instead of $q\bar{q}$, tunnelling from the vacuum. In the diquark model, the baryon and anti-baryon are always produced as nearest neighbours along the string. Additionally, a *popcorn* model exists creating in the simple case one additional meson between the baryons produced by diquarks and multiple mesons in the advanced case. If the energy of the string does not suffice to utilise the above mentioned methods, they are treated as clusters, similar to the cluster model described previously.

1.5 Jet Algorithms

For distances larger than about one Fermi the confinement does not allow coloured objects like quarks or gluons to be separated from each other. The energy invested in the colour field between these partons is used up to create quark anti-quark pairs from the vacuum until all these partons have been integrated into colourless hadrons. The resulting collimated streams of hadrons, typically named jets, are observed in a detector. However, their origin are a small number of partons which had been created in the hard collision. In order gain deeper understandings of the physics behind the hard collision process, the observed particles in the detector have to be linked back to the originating partons from the hard process. Therefore, algorithms group particles that are supposed to come from the same hard parton into a jet. The infrared and collinear emissions happen on a random basis. It has to be assured that their occurrence should not change the final observable. Thus, the clustering procedure of the jet algorithm should be robust against such random influences to definitely point to the initial parton direction. Collinear and infrared sensitive situations are depicted in sections 1.5.1 and 1.5.2. The algorithms can be run on different levels of calculation, generation or reconstruction as the objects used as input to the algorithms are defined to have comparable properties. For this purpose, a four-vector-like character, comprising a three dimensional direction and an energy component, is chosen. These quantities can be derived either from calculated particle properties or from measurements within different detector systems by using location and amount of the energy deposits. In the case of calorimeter objects, which are usually constructed to be mass-less are combined from different calorimeter cells into so-called *towers* before they are passed to the jet algorithm. Jet input or the final state jets will be massive four-vectors, depending on the recombination schemes which are used during tower building or during the jet clustering. Jet algorithms require the following ingredients to cluster jets from their input:

- A distance measure which is needed to define the separation between objects.
- A decision procedure whether objects are to be combined.
- A recombination scheme, which explains how to combine the input objects.

Within the CMS collaboration and also throughout this work straight forward four-vector addition is used, although there exist several different recombination schemes.

1.5.1 Collinear Safety

The collinear approximation could be used to calculate high energetic parton showers as jet evolution is usually happening in a small cone. This evolution contains for instance gluon radiation collinear to a high energetic parton. For the definition of jets, which has to connect the hadronic final state to the initial partonic process, such additional collinear radiation may not lead to a different interpretation. The definition of collinear safety is that all effects due to collinear splitting during calculation, hadronisation, or during measurements within the detector may not change the output of the algorithm. That means that the two situations pictured in Figure 1.7 should always produce one single jet. Algorithms which produce zero or two jets, in case of the upper left picture, are not collinear safe which depicts the problem of thresholds in seeded algorithms as follows: If both seeds are below the threshold - although they might together be large enough to be considered as a seed and thus also producing a jet - no jet might be produced. Additionally the collinear radiation leads to ambiguities, whether one or two jets will be created, which is also an unwanted feature.

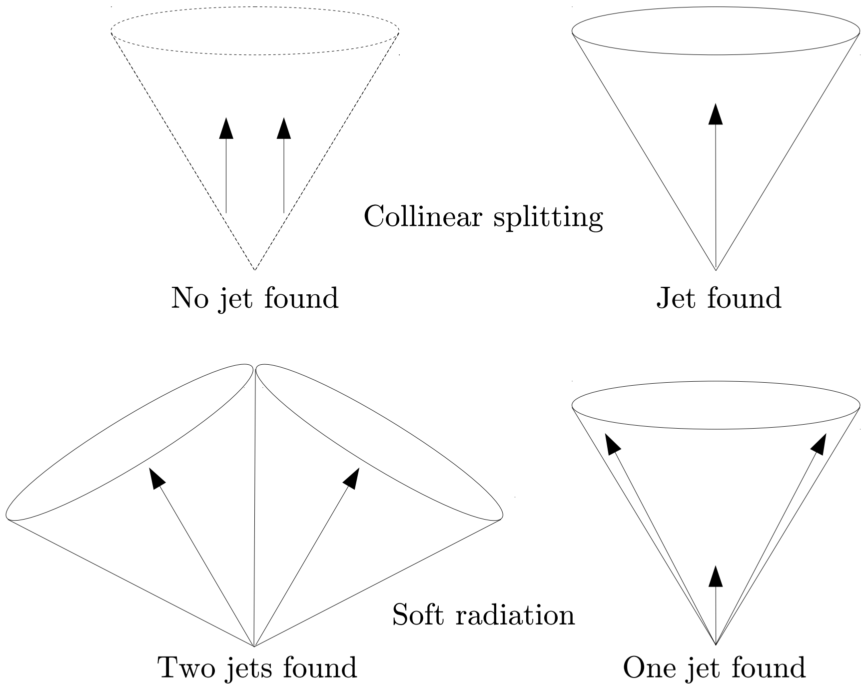

Figure 1.7: *Examples for collinear (upper) and infrared (safety) of jet algorithms.*

This can happen due to the splitting of the jet energy into two neighbouring parts of the calorimeter system. In case of NLO calculations, collinear unsafety leads to singularities which spoil the convergence of the MC integration.

1.5.2 Infrared Safety

During the part of the processes described by parton showers or hadronisation soft gluon radiations are also occurring. The emitted gluons have only a small amount of energy. Infrared safe algorithms should not be affected by this soft gluon radiation. This feature is intrinsically fulfilled within combinatorial algorithms like the k_{T} algorithms (see section 1.5.3) because the resolution parameter already manages that lower energetic parts are easily merged with higher energetic contributions, while the cone type algorithms are not intrinsically safe of such a behaviour. With a soft gluon being emitted in between two cone jets the energy overlap of the two initial cones can be large enough which might lead to a situation where the algorithms merge the two cones into one, as depicted in Figure 1.7. Clustering jets around the midpoint seeds should avoid this unwanted behaviour. This is implemented within the midpoint cone algorithm [38]. However, this is only temporarily solution for the infrared unsafety problem because an additional second radiation would already taint this solution. Therefore, only a seedless cone algorithm, like SISCone [39] or combinatorial algorithms, which are presented in the following, are safe to all orders in perturbation theory.

1.5.3 Combinatorial Algorithms

Combinatorial jet algorithms are not using a fixed jet geometry. The particles are clustered according to their orientation in four-vector space, which leads to collinear and infrared safe behaviour. The distances between all particles in an event have to be calculated at every step of the clustering, leading to a vast amount of computing overhead. The first combinatorial jet algorithms performance scaled with the clustered particles N in the order of $\mathcal{O}(N^3)$. By the use of Voronoi diagrams [40] this can be reduced to $\mathcal{O}(N^2)$ or even $\mathcal{O}(N\ ln(N))$ within the FastJet implementation [41].

The Inclusive $\mathrm{k_T}$ Algorithm

The inclusive $\mathrm{k_T}$ algorithm relies on the distance measures

$$
\begin{aligned}
d_{ij} &= min(p_{\mathrm{T},i}^2, p_{\mathrm{T},j}^2)\frac{\Delta R_{ij}^2}{R^2} \quad \text{with} \quad \Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2 && (1.33)\\
d_{iB} &= p_{\mathrm{T},i}^2 && (1.34)
\end{aligned}
$$

with the distance d_{ij} between two particles i and j in four-vector space and the distance d_{iB} between the particle i and the beam axis. The *resolution parameter* R fixes the jet size. The jet clustering is arranged in several different steps. Fist, the distances d_{ij} and d_{iB} are calculated and the minimum values of all d_{ij} and d_{iB} are determined. If the absolute minimum is given by a d_{ij}, i and j are combined to a single new object and the algorithm returns to the first step, removing i and$j$ from the list of inputs. If the distance between a particle and the beam d_{iB} is the smallest, the object is listed within final state jets and removed from the list of inputs. This procedure is repeated until no particles or objects are left in the event.

The $\mathrm{k_T}$ algorithm clusters all particles in the event in a jets, which can lead to jets with very small transverse momentum. Within experiments the limit in het p_T for $\mathrm{k_T}$ algorithm is given by the energy resolution of the detector components. In Monte Carlo simulations, it is convenient to only consider jets above a certain p_T threshold. Formula 1.33, indicates that the $\mathrm{k_T}$ algorithm first merges soft particles before it assigns them to a hard particle which is an advantage in studies of soft QCD. However, the fact that the $\mathrm{k_T}$ algorithm does not provide a fixed jet shape renders it less usable for many analyses.

The anti-$\mathrm{k_T}$ Algorithm

The anti-$\mathrm{k_T}$ algorithm [42] is very similar to the inclusive $\mathrm{k_T}$ algorithm. It uses a different distance measure which is given as

$$
\begin{aligned}
d_{ij} &= min(p_{\mathrm{T},i}^{-2}, p_{\mathrm{T},j}^{-2})\frac{\Delta R_{ij}^2}{R^2} \quad \text{with} \quad \Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2 && (1.35)\\
d_{iB} &= p_{\mathrm{T},i}^{-2} && (1.36)
\end{aligned}
$$

Therefore the anti-k_T algorithm reverses the clustering order as it first clusters soft particles to the closest hard particles, leading to a well defined almost perfectly circular jet shape for well isolated jets.

For jets that are found very close to each other with very different transverse momenta, the harder jet will be a circle, while the softer one will be crescent-shaped. In case of similar transverse momenta, they are either divided along a straight line or merged if their axes are close to each other. The intrinsic infrared and collinear safety and the fixed jet shape as well as a good scaling behaviour in high multiplicity events makes the anti-k_T algorithm the most desirable choice for most jet analyses.

The CMS Experiment

2.1 The Large Hadron Collider

Situated northwest of Geneva in Switzerland, on the border to France, the particle physics laboratory CERN is currently running the worlds largest particle collider, the Large Hadron Collider (LHC). With reaching a proton-proton collision energy of $\sqrt{s} = 2.36$ TeV in December 2009 and then $\sqrt{s} = 7$ TeV in March 2010 it has become the most energetic one as well. The 27 km circumference tunnel of the Large Electron-Positron Collider (LEP) is used to house the LHC and its four experiments - The Compact Muon Solenoid (CMS) [43], ATLAS[1] [44], ALICE[2] [45], LHCb[3] [46]. CMS and ATLAS are general purpose detectors, expected to explore physics at the TeV scale. Their main aims are to discover the Higgs boson and to find evidence of physics beyond the standard model, such as super symmetry or extra dimensions, and to upgrade the measurements of the free parameters of the standard model such as the running strong coupling α_S.

To keep the protons circulating in the LHC tunnel within two stable beams about 5000 superconducting niobium-titanium coils are assembled in dipole magnets producing a magnetic field of about 8 Tesla. The LHC began its operation in 2008 with a start-up beam energy of $450\,$GeV. During a magnet test for beam operation at $5\,$TeV in September 2008 a superconducting interconnection between two magnets failed, leading to a resistive voltage of 1V at 9kA. The resulting electric arc punctuated the helium enclosure of the cooling system. Several magnets were damaged and had to be repaired or replaced. To avoid further inci-

---

[1] A Toroidal LHC ApparatuS

[2] A Large Ion Collider Experiment

[3] Large Hadron Collider beauty experiment

dents, it was decided to restart the LHC with a centre-of-mass energy of $\sqrt{s} = 7$ TeV for the first physics operation in 2010. The design performance is expected to be reached after the technical shutdown which is scheduled for the end of 2011. Then it will collide beams of 2808 bunches of about 1.15×10^{11} protons each at a collision energy of $\sqrt{s} = 14$ TeV leading to a luminosity of 10^{34} cm^{-2}s^{-1}. Additionally to proton-proton collisions the LHC will collide lead nuclei with an energy of 5.52 TeV. Due to its high energy, the quark gluon plasma which is expected to be produced during this lead collisions will give scientists the possibility to study the state of the early universe. A sketch of the LHC underground structure can be found in Fig. 2.1.

The protons are injected into the LHC with an energy of 450 GeV which they reach in the Super Proton Synchrotron (SPS). Niobium sputtered cavities then accelerate them to the desired energy. The previously mentioned dipole magnets bend the protons trajectory to keep them on the track, while quadrupole and other special magnets are focusing the beam.

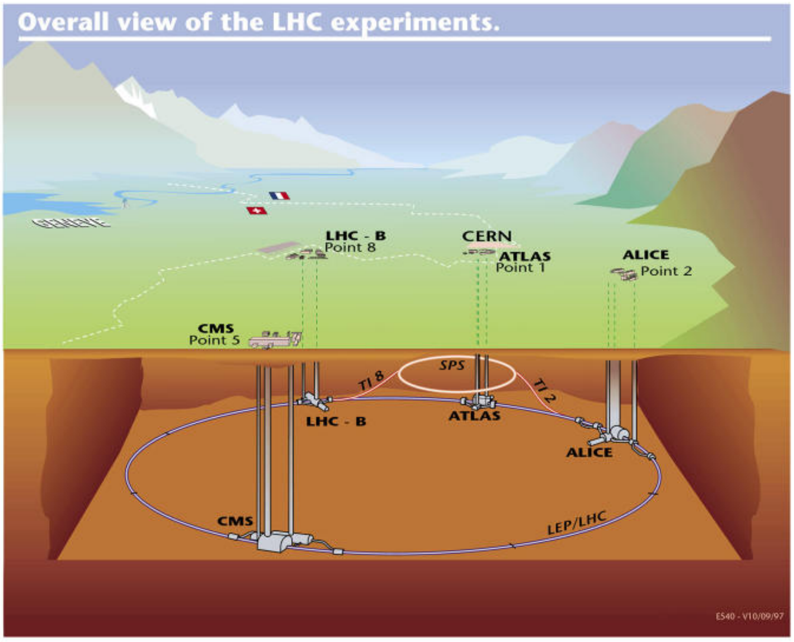

Figure 2.1: *Schematic overview of the Large Hadron Collider LHC at CERN with the interaction points at the four detectors ALICE, ATLAS, CMS and LHCb [47].*

The event rate $\dot{N}_{\text{event}}$ of a given process with the cross-section σ_{process} is given as

$$\dot{N}_{\text{event}} = \mathcal{L} \cdot \sigma_{\text{process}}. \tag{2.1}$$

Machine Parameter	Unit	Nominal Value
Luminosity $\mathcal{L}$	$\mathrm{cm}^{-2}\mathrm{s}^{-1}$	10^{34}
Number of bunches n_b	1	2808
Protons per bunch N_b	1	10×10^{11}
RMS bunch length σ_z	cm	7.95
RMS bunch transverse width σ^*	m	16.7
Relativistic gamma factor γ_r	1	7460
Revolution frequency f_rev	kHz	11.2
Bunch crossing frequency f_cross	MHz	40
Crossing angle at interaction point θ_c	rad	285
Beta function β^*	m	0.55
Transverse beam emittance ϵ_n	m	3.75
Luminosity lifetime τ_L	h	15

Table 2.1: *LHC parameters at nominal proton-proton running conditions. Taken from [48]*

For a collider like the LHC, the luminosity is connected to a number of beam parameters via:

$$\mathcal{L} = \frac{N_\mathrm{b}^2 n_\mathrm{b} f_\mathrm{rev} \gamma_\mathrm{r}}{4\pi \epsilon_\mathrm{n} \beta^*} F, \tag{2.2}$$

where N_b is the number of particles per bunch, n_b is the number of bunches per beam, γ_r is the relativistic gamma factor, f_rev the revolution frequency, ϵ_n the normalized transverse beam emittance, β^* the value of the betatron function at the collision point. The geometric luminosity reduction F is connected to the beam crossing angle at the interaction point:

$$F = \left(1 + \left(\frac{\theta_\mathrm{c}\sigma_z}{2\sigma^*}\right)^2\right)^{-1/2}. \tag{2.3}$$

Here, θ_c is the full crossing angle, σ_z is the root mean squared of the bunch lengths and σ^* the root mean squared of the transverse beam size. Table 2.1 gives an overview of the estimated LHC design parameters for the beams crossing within the CMS detector at a centre-of-mass energy of $\sqrt{s} = 14$ TeV.

2.2 The Compact Muon Solenoid

CMS is one of the largest international scientific collaborations in history. Approximately 3500 scientists from 200 institutes in 37 countries are involved in the commissioning and maintenance of the detector as well as the realisation of physics analyses. The CMS detector with its overall length of 21.5 m and a diameter of 15 m has a total weight of 12500 t. The silicon tracking system and the electromagnetic and hadronic calorimeters are placed within a 13 m long superconducting solenoid, which provides a magnetic field of about 4 Tesla. The muon chambers are embedded in the iron return yoke providing a magnetic field of about 1.8 Tesla. In the following an overview of the different detector parts is given referring to the very detailed description of the CMS experiment in [43].

2.2.1 Tracking System

The CMS tracking system, placed in the magnetic field of the solenoid, measures the tracks of charged particles. Their momenta can be calculated from the curvature of their trajectories. The tracking system covers a volume given by a cylinder of 5.8 m in length and 2.8 m in diameter which corresponds to a pseudo-rapidity range of $|\eta| < 2.5$. Particles passing the tracking system deposit energy within sequent detector layers. These different *tracker hits* are combined by a pattern recognition algorithm, which forms particle trajectories. Therefore, this algorithm decides which of the possible particle trajectories are most probably originating form the same particles and declares them as particle tracks. As the innermost layer of the detector is very close to the beam pipe to precisely determine the position of decay vertices of particles, the material used for the tracker has to be radiation hard. To master these challenges, CMS employs an all-silicon approach divided into two subsystems, the silicon pixel detector and the silicon strip detector, which are explained in the following. An overview of the tracker system is given in Fig. 2.4.

2.2.2 The Silicon Pixel Detector

The silicon pixel detector consists of three cylindrical hybrid pixel layers which are arranged at radii of 4.4 cm 7.3 cm and 10.2 cm around the interaction point and complemented by two disks of pixel modules on each side. It delivers three high precision space points on each charged particle trajectory. In total the pixel detector provides 66 million pixels and covers and area of about 1 m^2. With a pixel size of 100×150 μm^2, a spatial resolution in the range of about 15 μm to 20 μm can be achieved.

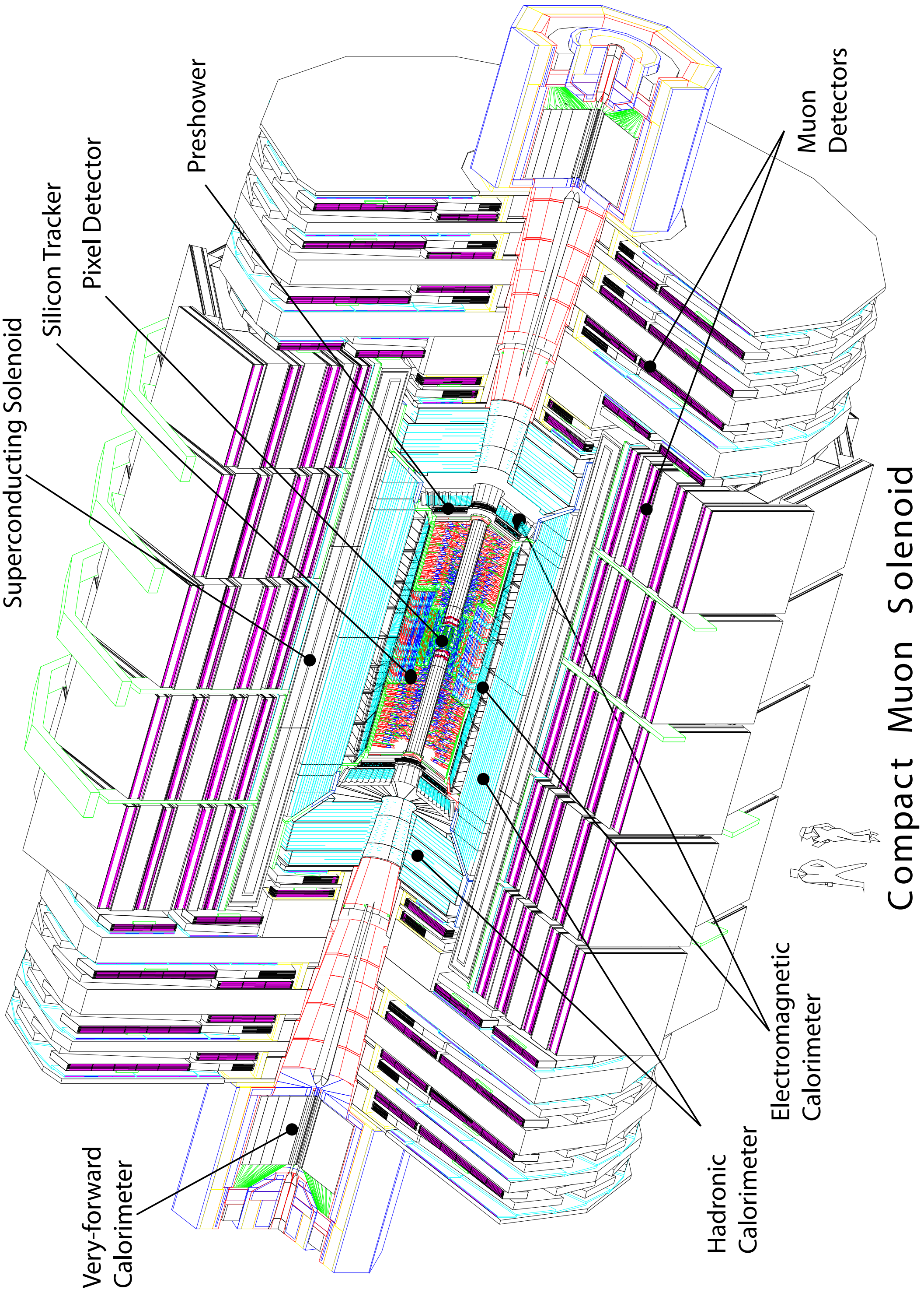

Figure 2.2: *Overview of the CMS detector.*

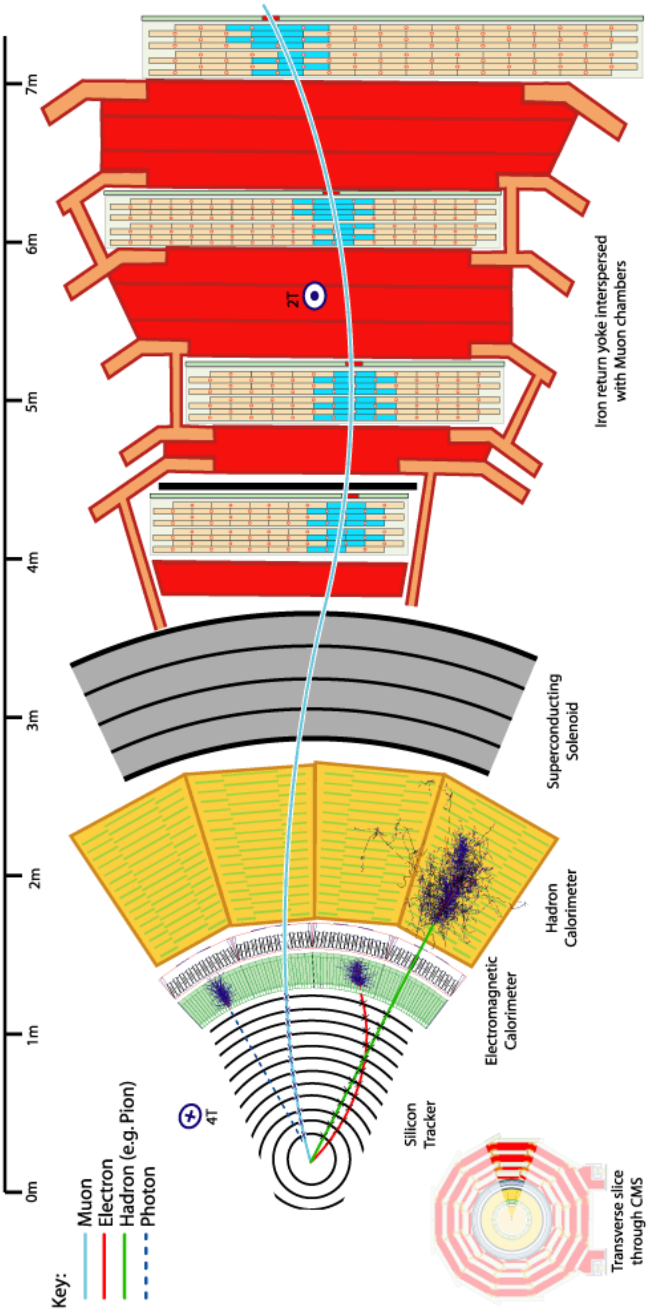

Figure 2.3: *Profile of the CMS detector with the tracks of an electron, a photon, a hadron (e. g. pion) and a muon. The electron and the photon deposit their energy in the electromagnetic calorimeter and are stopped there. The hadron passes this region of the detector without losing much of its kinetic energy and reaches the hadron calorimeter, where it is stopped. Only muons are able to pass the absorber material of the superconducting solenoid and can be detected in the muon chambers.*

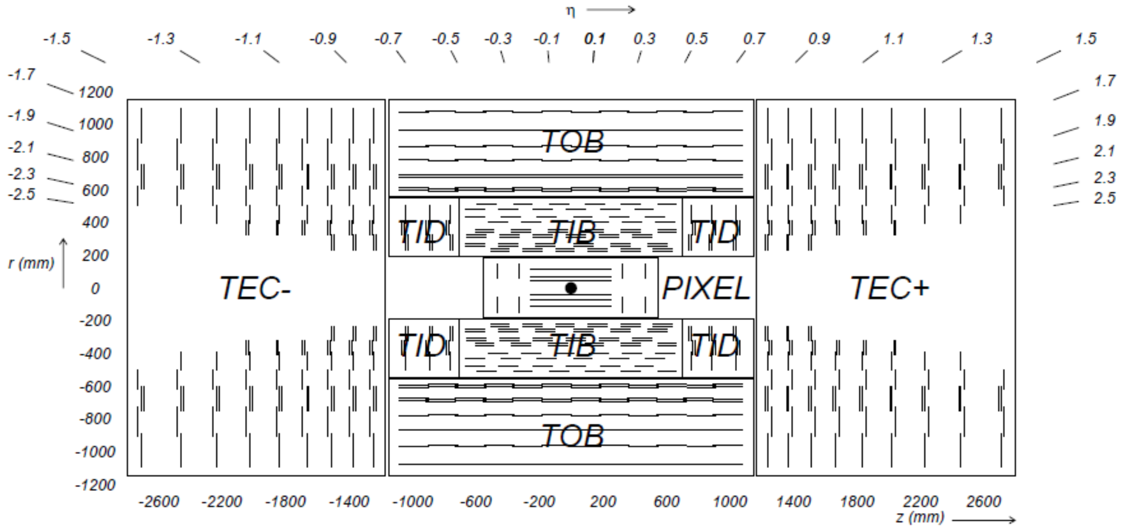

Figure 2.4: *Overview of the CMS silicon detector with Tracker Inner Barrel (TIB), Tracker Inner Disc (TID), Tracker Outer Barrel (TOB) and Tracker EndCap (TEC). From [49].*

2.2.3 The Silicon Strip Detector

The silicon strip detector consists of four inner (TIB) and six outer barrel (TOB) layers assembled in shells. Layer 1 and 2 are double-sided in each of the two barrel regions, whereas the remaining ones are only single-sided. The inner barrel detector is completed by two endcaps, each composed of three small discs. In total, the 15400 modules cover an area of 210 m^2 with up to 9.6 million silicon strips in total.

2.2.4 Calorimeter System

The detector components following the tracking system are the calorimeters. Here, the total-absorption method is used to determine the kinetic energy of an incident particle.

2.2.5 The Electromagnetic Calorimeter (ECAL)

Electrons, positrons and photons create an electromagnetic shower, a cascade of photons and electron-positron pairs, in the calorimeter material. As the incident particle can also pass energy to the electrons of the calorimeter material via Compton scattering and the photo-electric effect, the energy deposit in the electromagnetic calorimeter is proportional to the

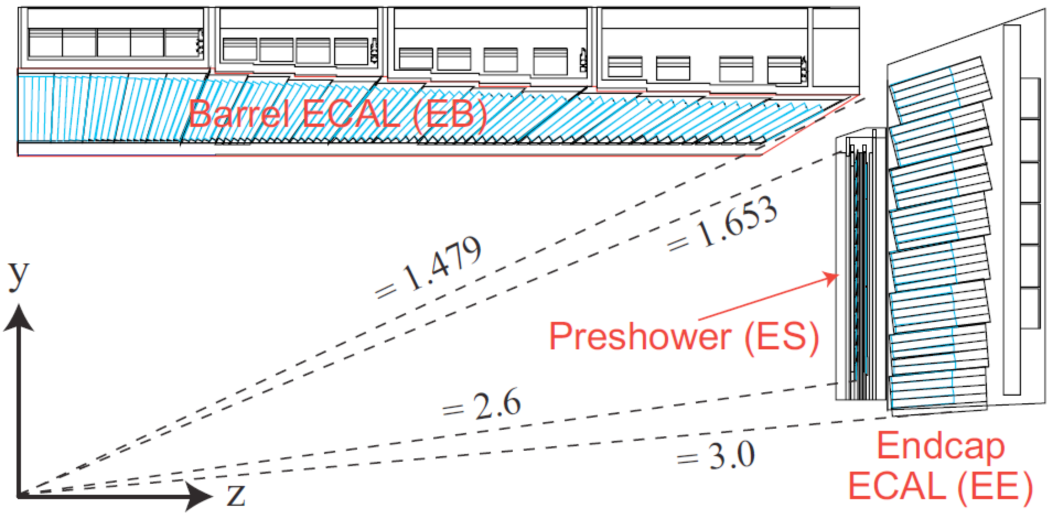

Figure 2.5: *Overview of the layout of the electromagnetic calorimeter of the CMS experiment. Taken from [50].*

kinetic energy of the incident particle. This first calorimeter layer of the CMS detector consists of more than 80000 scintillating lead-tungsten ($PbWO_4$) crystals, 61200 in the barrel area and 21523 in the endcaps. The high density of 8.28 g/cm^3 and the large number of electrons per atom lead to a short radiation length ($X_0 = 0.89$ cm) and Molière length (2.2 cm) for electrons[4] and photons. The dimensions of the electromagnetic calorimeter are designed in a way that all these particles are completely stopped in this detector component. This compact calorimeter provides a high energy resolution which is in particular useful for the measurement of photon energies. The CMS ECAL is sketched in Fig. 2.5

2.2.6 The Hadronic Calorimeter (HCAL)

Hadrons will pass the ECAL without losing much of their energy as a result of their larger radiation length. These particles are stopped in the hadronic calorimeter (Fig. 2.6)where they create hadronic showers. This cascade results from the interaction of the incident particle with the nuclei of the typically heavy material and consists mainly of pions, K-mesons, protons, neutrons and fragments of nuclei. To cope the high penetrating power of hadrons, the hadronic calorimeter consists of 50 mm thick copper absorber plates interleaved with 4 mm thick scintillator tiles read out by wavelength shifting fibres. In order to achieve a

---

[4] From here on, the term electron refers to both, electrons and positrons. It is explicitly indicated if exclusively electrons are meant.

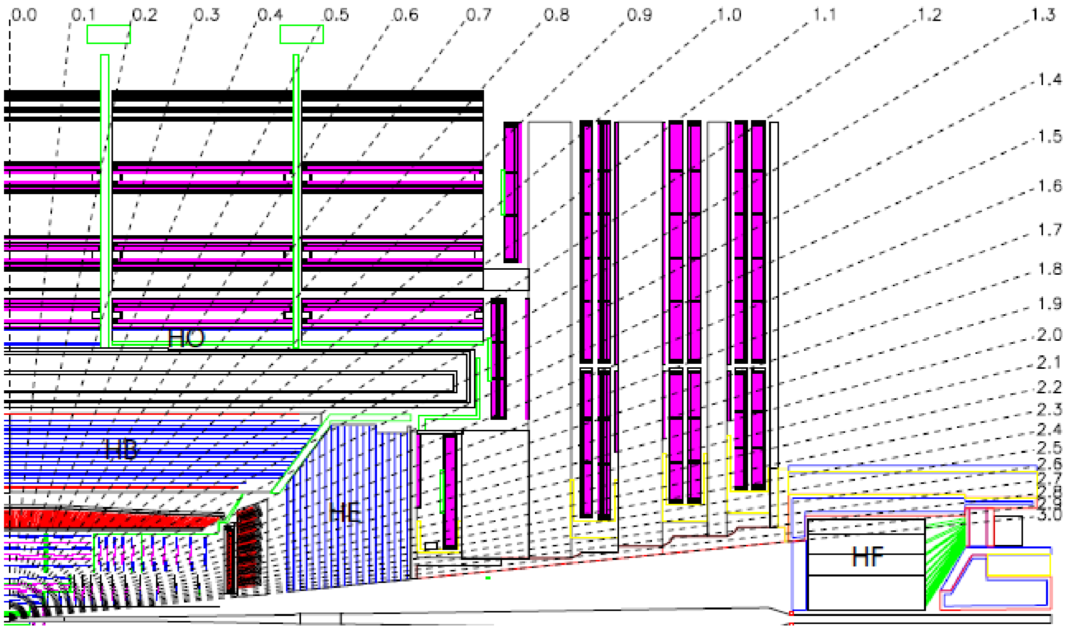

Figure 2.6: *Slice through the CMS hadronic calorimeter. It consists of the Hadronic Barrel (HB), the Hadronic Endcap (HE), the Hadronic Outer (HO) detector and the Hadronic Forward (HF) detector. Taken from [49].*

good resolution for missing energy, a calorimeter coverage up to a pseudo-rapidity $|\eta| = 5$ is needed. For this purpose, the Hadronic Forward (HF) calorimeters are placed at 11.2 m from the interaction point extend the pseudo-rapidity coverage down to $|\eta| = 5.2$.

2.2.7 Superconducting Solenoid

The solenoid is about 13 m long and has an inner diameter of 5.9 m. It generates a magnetic field of up to 4 Tesla in the inner region. The flux generated by the coil is returned via a 1.5 m thick saturated iron yoke. With its 2.5 GJ of stored energy, up to now it is the largest superconducting magnet in the world.

2.2.8 Muon Chambers

A precise identification and measurement of muons is one of the primary tasks of CMS. The sensors in the muon chambers are interleaved with the iron return yoke plates and placed behind thick layers of absorber material. This guarantees that only muons reach this part of the detector, where three types of gaseous muon sensors are installed: Drift Tubes (DT) in the central barrel region, Cathode Strip Chambers (CSC) in the endcap region and Resistive Parallel Plate Chambers (RPC) in both, the barrel and the endcaps. The latter are very fast and thus well designed for Level-1 triggering, whereas the slower DT and CSC sensors provide a precise measurement of momentum and position of the muon.

2.2.9 Data Acquisition and Trigger

At the CMS detector, the proton bunches will cross each other 40 million times each second, leading to an interaction rate of 10^9 Hz. To reduce the output data rate to a manageable rate of 100-400 Hz, a sophisticated trigger system is used.

The Level-1 trigger (L1) is very fast and designed to reduce the rate to about 100 kHz. Special hardware processors are looking for signs of an interesting event in each piece of the detector, for example a group of calorimeter cells with a high energy deposition. While the Level-1 trigger is deciding to accept or reject an event, the high resolution data is kept in the memory pipelines of the front-end electronics. If accepted, the event is passed to the High Level Trigger system (HLT). Here, information from different detector pieces is combined. This processing will take place on commodity processors and could reach up to 1 s of processing time per event. In this step, the rate is reduced to 150 Hz and the remaining data will be stored for offline processing.

2.3 Grid Computing

After the data reduction through the trigger system is passed, still 400 MB per second have to be stored and analysed for the CMS experiment during operation of the LHC. Facing the computing challenges to distribute, reprocess and to analyse the acquired data, the LHC collaborations share their computing resources within the the Worldwide LHC Computing Grid (WLCG). It is organised in a tiered structure as shown in figure 2.7.

Different tasks are assigned to the Tiers. The Tier 0 centre at CERN receives the data output directly from the Data Acquisition (DAQ) System from the experiments and distributes it to the Tier 1 centres.

The Tier 1 centres dispose of large mass storage systems including tape systems and computing power. They reprocess the data and allocate it to their attached Tier 2 centres. The main focus of Tier 2 centres is to allow users to perform physics analyses and the production of simulated data. They need an indispensable fast broadband network connection to their

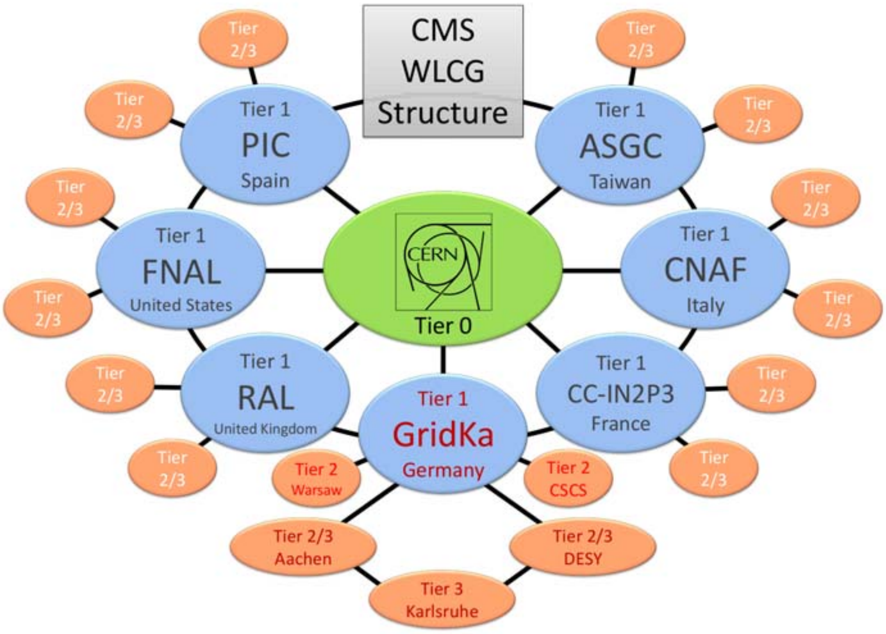

Figure 2.7: *The tiered structure of the WLCG.*

Tier 1 centre to make the simulated events available at the Tier 1 site and to receive a specific data set for ongoing analysis without the need to store them over a longer period. There are also Tier 3 centres which consist of smaller computing clusters maintained by university groups. These are connected to the Tier 1 and 2 centres and are an additional layer for the user analyses, like the IEKP Tier 3 centre. Tier 0,1 and 2 have to provide a 24/7 service.

The whole WLCG infrastructure relies on different services at the different sites. These software services (see appendix B.4 for details on the services) together with the interfaces for the users are represented by the grid middleware gLite. The WLCG sites have to provide a standardised software environment to prevent incompatibility problems with the experiment software or grid middleware and the operating systems (OS) used at the sites. Therefore, CMS grid-sites have the constraint to use Scientific Linux [51] as operating system on all service an computing machines.

Virtualisation in High Throughput Computing and Grid Environments

During the past years, computer virtualization evolved to one of the central IT infrastructure components. The roots of virtualization reach back into the 1960s, the era of large mainframes. With an increasing amount of data which had to be processed in parallel by a growing number of users per machine it turned out, that the workflow on these mainframes could be optimised by running multiple operating system instances virtualized on one mainframe. These virtual environments were scheduled cleverly to give one virtual instance the possibility of waiting for the needed data while the processing units were assigned to other virtual machines which already had their data accessible. Such a first implementation of virtualization was introduced by IBM on the 360/40 mainframes.

The introduction of the personal computer (PC) with the Altair 8800 in 1975 and the Apple II in 1977 and finally the IBM PC in 1981 marked a turning point. Now, one PC per user provided all basic functionalities with possible access to mainframes if required. Therefore, virtualization was essentially abandoned in favour of client-server applications .

IBM's PC x86 architecture prevailed and evolved to the standard architecture not only on the desktops but also in large scale High Performance Computing (HPC) and High Throughput Computing (HTC) facilities, as it turned out to be very flexible and cost effective to build computing clusters out of off-the-shelf components.

Today's cluster and server infrastructures are typically overdesigned to cope with expected peak loads. The compound of a software service and a hardware machine is inflexible and prevents the usage of idle hardware by other software services. Thus, the old concept of virtualization re-appeared at the beginning of the 21st century and became a new information technology (IT) standard. It even enabled the development of a completely new computing

concept, *cloud computing*.
This chapter provides a general introduction in virtualization in the first part, followed by a description of current computing infrastructures as used by the High Energy Physics community. A motivation is given why virtualization is needed not only for the infrastructure service consolidation but also in the fields of HPC and HTC. A concept of dynamically integrating virtualized computing worker nodes into standard batch systems was developed and will be explained and discussed in detail. It allows each user group of a shared computing cluster to use a custom computing environment within a dynamically dedicated cluster partition.

3.1 Virtualization

Virtualization, specifically hardware virtualization (also referred as platform virtualization), abstracts the computing platform by hiding the underlying physical hardware resource from the user. There are several reasons for hardware virtualization, which are:

- Compatibility: Provide applications with the needed operating system in virtual machines without altering the host system.
- Load-Balancing: Better overall utilization of the hardware resources by shifting virtual machines between available resources.
- High-Availability: By migrating or re-starting virtual machines on different hardware resources downtimes due to maintenance or hardware failures can be minimized.
- OS-Evaluation and Testing: Possibility to test new systems and setups without touching the production system or providing additional testing hardware.
- Encapsulation: Limit the risks for the host system to be infected by malware and non-disclosure issues by encapsulating the users and the applications in virtual environments.

3.1.1 Definition of Virtualization

One of the first precise definitions of hardware virtualization was given in 1973 by Gerald J. Popek and Robert P. Goldberg [52], by defining architecture requirements for virtualization. According to them, a virtual machine has to be an "*efficient*, *isolated duplicate of the real machine*". The virtual machine monitor (VMM) software will therefore act as a manager of

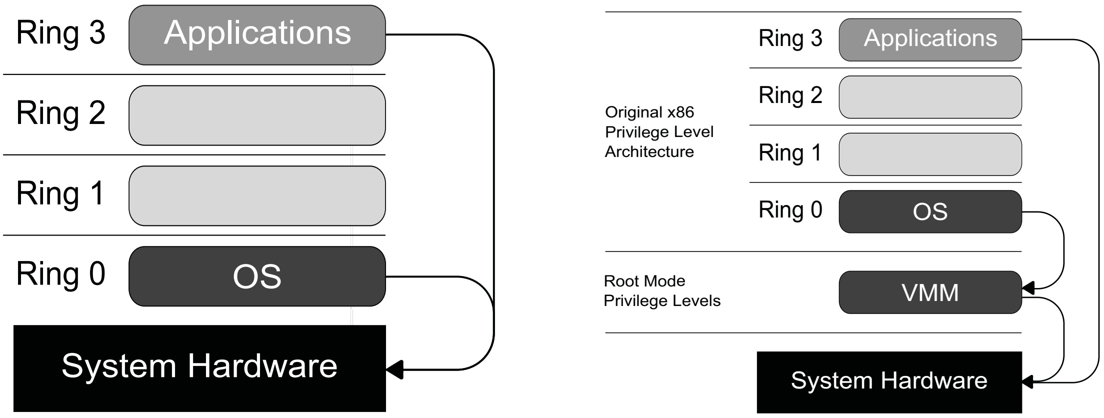

Figure 3.1: *Left: x86 privilege levels. Full hardware access is granted only in Ring 0, while user applications have to run in the restricted Ring 3 level. Right: Virtualization extended x86 architecture. Virtualization support provided by the hardware. The x86 execution levels are shifted above a new privilege level. The Virtual Machine Monitor is managing the different operating system guests by running in this new full privileged environment. Taken from [53]*

the hardware resources by providing the virtual machines with "*essentially identical environments to the original machine*" and, very important, the programs or systems running in these environments should show only minor performance decreases. For that purpose the VMM has the full control of the hardware resources. Hereby, the two related technologies, virtualization and emulation, were clearly distinguished, as the required efficiency for virtualization is only reachable if a statistically dominant subset of the virtual processor instructions are executed without any software intervention directly by the real processor.

3.1.2 Virtualization on the x86 Architecture

The technical standard of today's infrastructures is the x86 based architecture. x86 based operating systems are designed to run directly on the hardware, encapsulating different tasks in hierarchically organized privilege levels, called Ring 0,1,2 and 3, with well defined hardware and inter-process access rights. While Ring 0 provides direct access to the hardware with full privileges, user processes are running in the Ring 3 environment. This strictly organized hierarchical scheme is sketched in Fig. 3.1 on the left hand side. To provide the possibility of running several operating systems in parallel on the same hardware host through virtualization it requires an additional privilege level by either extending the x86 architecture itself as shown on the right hand side in Fig. 3.1, or an additional software layer. There are several different virtualization techniques, using both approaches as described in detail in the next sections.

Full Virtualization

Full virtualization provides a specific environment which is a complete simulation of the underlying hardware to the virtual machines. The operating system encapsulated in the virtual machines has indirect access to the hardware through the abstraction VMM layer, the hypervisor. This hypervisor acts as manager and translator between the hardware and the virtual machines. On a fully virtualized virtual machine, the operating system has not to be modified to match any requirements arising from the applied virtualization solution. The encapsulated OS communicates with the simulated hardware infrastructure like BIOS, CPU, network and peripheral devices though its native drivers. The term full virtualization is diffusely defined: It evolved over the time as different products claimed to provide full virtualization. However real full virtualization, similar to the one applied already in the 1960s on IBM's mainframes, is now finally possible by exploiting the previous mentioned x86 virtualization extensions, AMD-V and Intel VT-x as presented in the next section.
Providing full virtualization without hardware support was possible by techniques such as ,for instance, the binary translation [54] used by VMware, which translates the virtual machine instructions by modifying them before they are passed to the central processing unit (CPU).

Full virtualization software:

- Kernel-based Virtual Machine (KVM) [55]:
  KVM provides full virtualization on platforms with hardware virtualization extensions like AMD-V and Intel VT-x. It is capable of mixing full and paravirtualization by using the VirtIO [56] framework. In this case specific hardware devices are paravirtualized to increase performance. Find more details on KVM and VirtIO in section 3.2.1.

- VMware [57] Server (former GSX) / VMware Player:
  These are the free products of VMware using full virtualisation running the hypervisor as application in the host OS (Windows or Linux).

- VMware ESX:
  It is the high end product of VMware. To avoid performance losings VMware combined host OS and hypervisor in a minimalist system one has to run on certified hardware only. It has the ability to run several certified operating systems.

- Others: Parallels Desktop/Workstation [58], Adeos [59], Mac-on-Linux [60], Win4BSD [61]

x86 Virtualization Extensions

Full virtualization on the x86 architecture was completely based on software techniques like the VMware binary translation as previously mentioned and the virtualization software developers were faced with very complex issues. With the growing market demands for virtualization, AMD and Intel both therefore introduced new processor extensions to the x86 architecture. The first generation implemented new privilege instructions which eased the development of virtualization software but showed no performance improvement as hardware virtualization assistance for the memory management unit was still missing. AMD and Intel had a very similar approach as both were inspired by the IBM's experiences and developments in virtualization. In both implementations the processor finally acquired a new mode, the new ring level which was needed by the VMM (as seen in Fig. 3.1).

Paravirtualization

In the paravirtualization approach,the hardware access is directly passed to the VMs by the VMM. This is done by certain *hooks* in the hypervisor system which are used by the virtual machines to access the hardware. Therefore, a modified version of the virtual machine operating system has to be installed which can handle the specific hooks from the specific paravirtualization software in use. If the kernel of the considered OS has no virtualization support included, this task is not trivial. In fact, the Linux Kernel already supports virtualization and this will be expanded in near future.

In figure 3.2, one can compare the two virtualisation approaches. Examples are:

- XEN [62]: Developed by the University of Cambridge, this Open Source Virtual Machine Monitor (VMM) supports Linux and FreeBSD. Xen today supports both full and paravirtualization. In the first case, one needs to have the hardware virtualization support, in the second case host and quest kernels have to be patched with Xen specific code. Find more details on Xen in section 3.2.1. .
- User Mode Linux (UML) [63]: UML is already part of the official Linux kernel releases. It is a kernel flavour which allows to run complete Linux kernels as application processes inside of operating Linux systems without affecting their configuration or stability.
- Virtuozzo [64]: Virtuozzo is a commercial product specialised on server virtualisation.

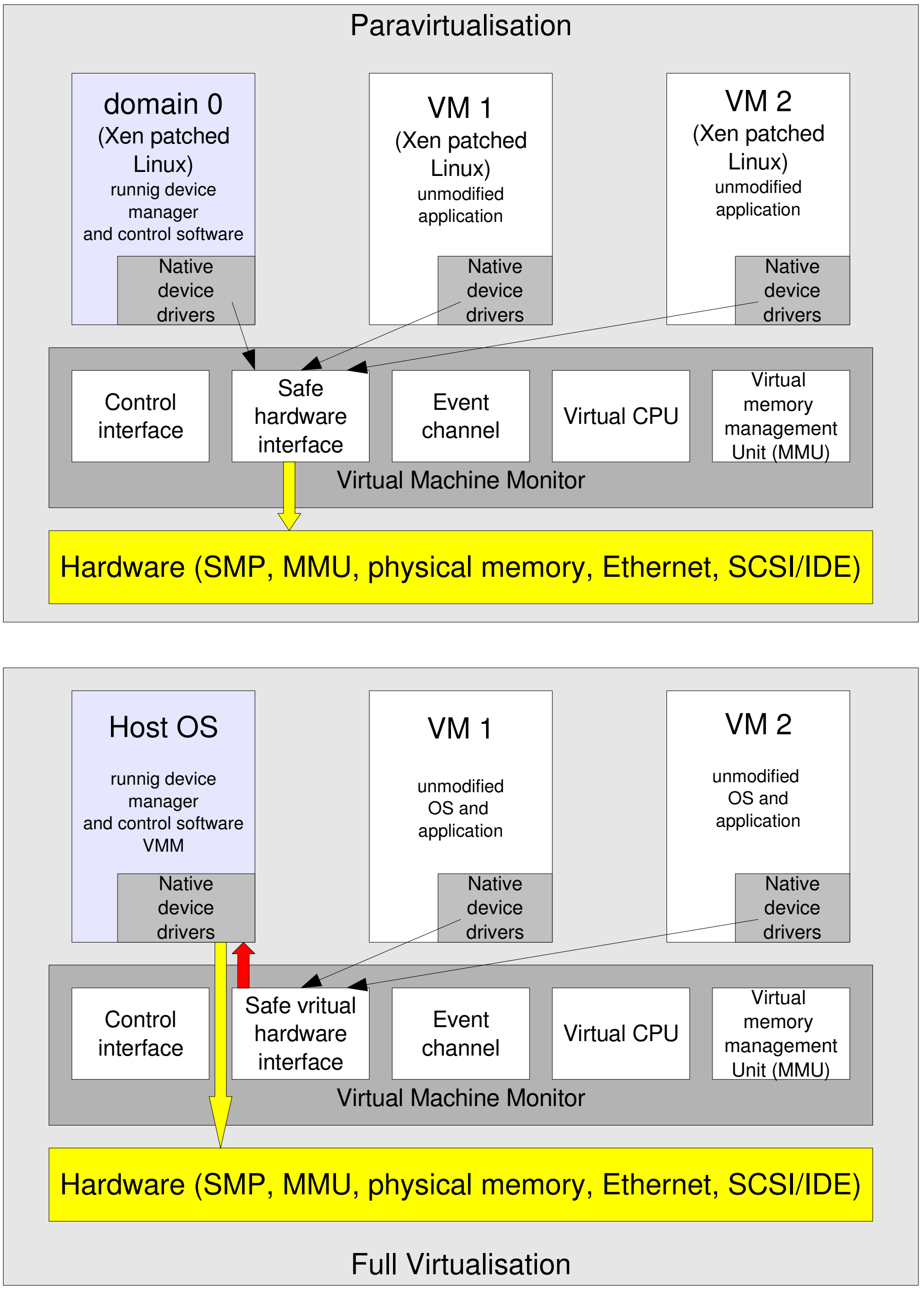

Figure 3.2: *Paravirtualization (top) and full virtualisation (bottom).*

Other Virtualization Techniques

Additionally to full and paravirtualization, there are several other techniques which can be accounted to one of the previous mentioned methods, or an emulation approach. Operating System Containers, for example, provide well encapsulated virtual operating systems by replicating the kernel data structure with a minor performance overhead but a strong dependence to the used kernel of the host system.

3.2 Virtualization in High Performance and High Throughput Computing

As pointed out in chapter 2, typical High Energy Physics experiments produce a vast amount of data to be analysed by thousands of physicists from hundreds of countries distributed all over the world. The ≈ 450 MB/s data streaming out of the detector after hardware and software trigger mechanisms have to be recorded and stored redundantly and reprocessed several times with the newest calibrations and alignment data before they can be split up into data subsets for the final physics analyses. Additionally to the real data recorded by the experiments, about the same amount of simulated Monte Carlo data samples have to be generated, stored, reprocessed and analysed to compare the theoretical predictions used for the Monte Carlo simulations with the real measurements. These data processing and analysis work flows are applied at several hundreds of computing centres hosted by universities and research centres spread all over the globe, providing parts of their computing infrastructure to their respective experiment community. Together with several dedicated computing centres, the computing and storage abilities of all participating entities are combined in the WLCG (see section 2.3).

By sharing such a large scale collective analysis infrastructure, today's experiments in HEP have consolidated the used software environments to only a small number of operating system flavours, validated for their experiment and grid middleware software. At research facilities like CERN or larger universities or institutions, computing resource providers tend to have a consolidated operating systems landscape to minimize their administrative overhead. In cases where a cluster is shared between different communities with different requirements for their computing or security setup, one solution would be to statically divide the shared computing clusters into separated sub-clusters where one would work against the previously mentioned tentative consolidation of the OS landscape. Additionally, no opportunistic distribution of the load between the user communities can be achieved, resulting in a poor overall utilization efficiency of the provided hardware.

Thus, a preferred approach is to make the batch system aware of virtualization and to provide each community its favoured operating system and software setup in virtual machines. Here, the scheduler has full flexibility resulting in a better overall efficiency of the resources. There are two different possibilities to virtualize the available cluster worker nodes. On one

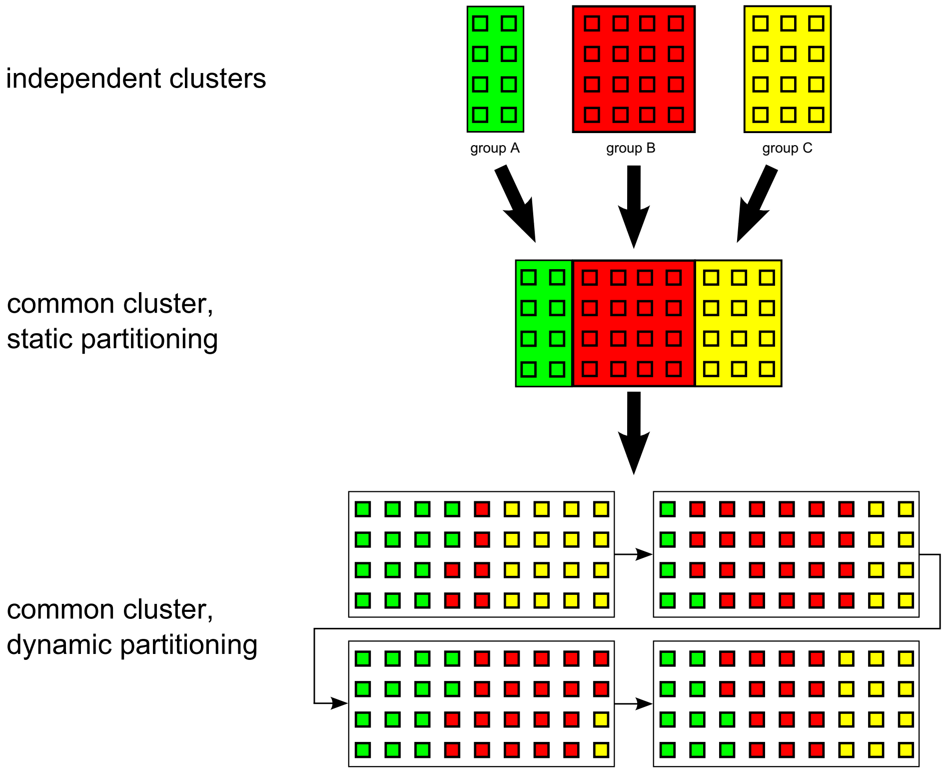

Figure 3.3: *Three possibilities to run HPC clusters for providing different software environments. First possibility at the top shows the most trivial solution where each user group runs on a dedicated cluster with each having a homogeneous setup. In the two other scenarios, the hardware infrastructure is shared among all user groups. In the static partitioned case, each group gets a dedicated subset of computing nodes where their specific environment is installed. The third possibility is to have the environments dynamically assigned to each group depending on their utilization of the resources.*

side, there is the site perspective, where one has to run on a system with a set-up which does not cope the needs of the users in security, privacy and compatibility aspects. On the other side, there is the grid perspective, where one wants to provide the grid users the possibility to choose between different environments for their jobs. In both cases, one wants to assure the simplest and most transparent approach possible to ease maintenance and management of the whole system. The virtualization layer should be completely hidden from the users, so that they only have to choose the environment they want to be served with. The virtualization of worker nodes results in a dynamic partitioning of a cluster as seen in figure 3.3. All concurrently running virtual machines of the same type hereby represent a partition of the cluster. In contrast to a statically partitioned cluster, these dynamic partitions are able to cover idle times and peak demands of the different user groups and therefore assure an opportunistic distribution of the load on the resource, optimising the overall utilization efficiency of the cluster.

3.2.1 ViBatch - Virtualization of Worker Nodes within Common Batch Systems

To dynamically provide a subset of a computing infrastructure with a specific working environment in terms of operating system and installed software, two different possibilities are conceivable: either re-install the hardware nodes via some automated installation mechanism, which is not feasible on a job by job basis, or virtualize the computing nodes where the grade of dynamic partitioning is adjustable (job by job or time limited partitioning of a cluster). Several projects over the last years exploiting virtualization in HPC clusters have been implemented and tested. Most of them need batch systems which are aware of virtual machines or require a modification of the resource manager and batch job scheduler in order to provide management functionality for the virtual machines. Such are e.g. Magrathea [65] or Xen Grid Engine [66] and our first approach for virtualizing worker nodes [67]. As already depicted, all these implementations need a change of the used batch system software or additional software daemons or a combination of both. Within the scope of this work, the main aim in the development of a new cluster virtualization software was to use common, existing software tools for the virtualization layer and to make only small or even no changes to the used batch system to provide portability and to confine the dependence on a specific base installation of the cluster infrastructure. This leading thought was implemented within ViBatch [68], a system described in the following which makes use of standard batch system abilities and a common virtualization layer enabling the use of a wide variety of virtualization systems. An introduction to batch systems, and some examples of their setups are given in appendix B.1.

A major functionality of current standard batch systems, the so-called prologue and epilogue scripts (see Appendix B.1.3) appeared to be in the main focus of the implementation. The standard task of these scripts is to prepare and cleanup the computing nodes before and after a job was executed. In ViBatch these scripts contain the whole virtualization step as discussed in the following.

Choice of Virtualization Technique

As already discussed above, there are special requirements for a virtualization technique in a dynamically partitioned HTC cluster:

- Support of different commodity operating systems.
- Acceptable overhead in performance.
- Easy installation and maintenance of virtual machines.

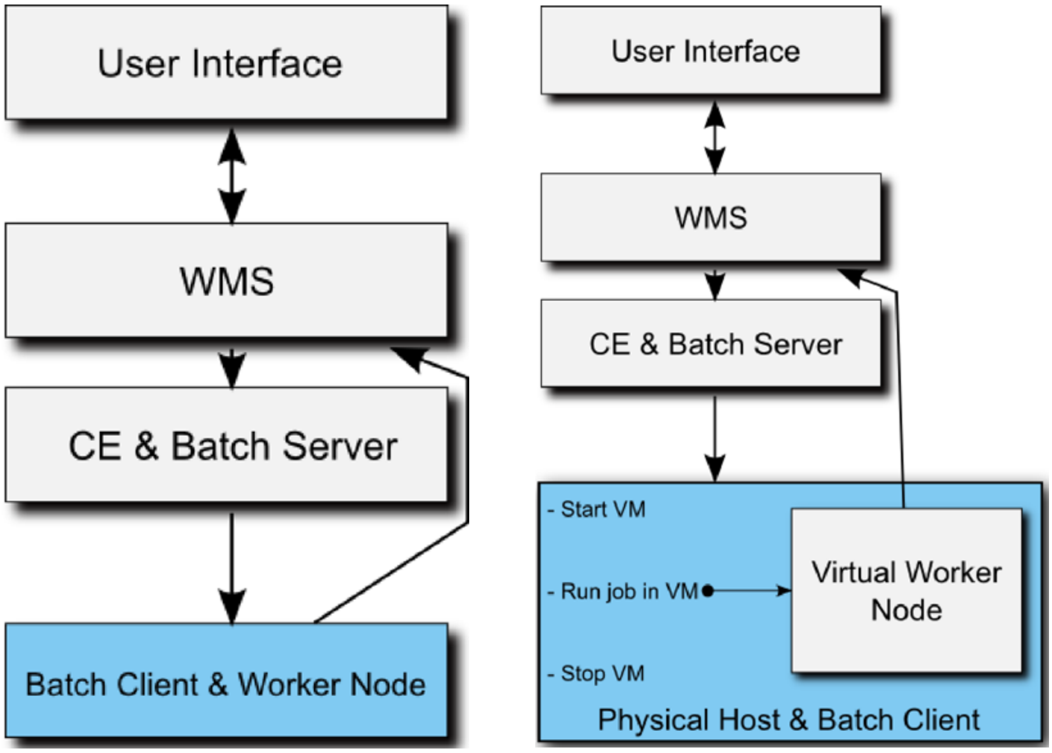

Figure 3.4: *Encapsulating a job in a virtual machine. By using prologue and epilogue scripts of the batch system, the virtual machine is not a property of the node. It is an extension to the original job, a so-called wrap-job.*

- Isolation of virtual machines from the host system.
- Controlling of virtual machines using scripts.
- Acceptable additional costs through possible required software licences.

Xen and KVM are our candidates of choice for such a partitioning. While Xen was the standard for years in common Linux distributions, KVM is getting more and more important due to its direct implementation in the Linux kernel.

The two hypervisors have a different virtualization approach (see Fig. 3.5). While Xen uses the common approach where the VMM is responsible for the scheduling and memory management as well as the hardware access of the virtual machines, KVM falls back on the standard abilities of the Linux kernel. It runs a VM as standard Linux process. In contrast to Xen, KVM relies on the hardware virtualization features of the CPU to reach its full performance in contrast to the emulation mode in which it is able to fall back.

The performance overhead is acceptable in both cases and a common API, `libvirt`, is available which provides command line and programming interfaces to ease the implemen-

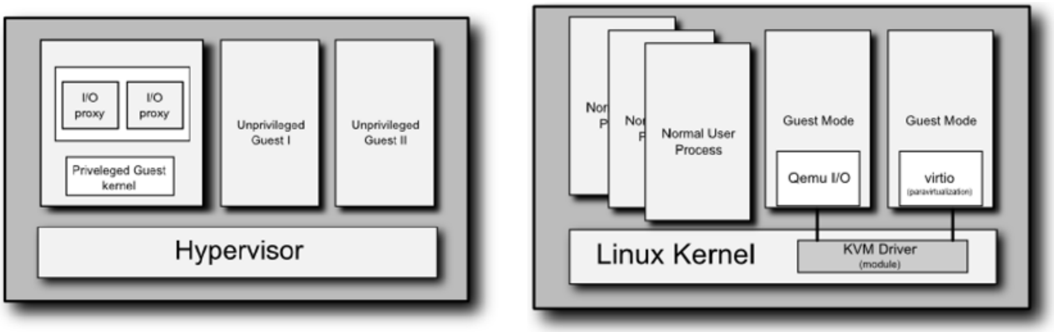

Figure 3.5: *Common virtualization model using a hypervisor on the left hand side and the Linux kernel as a hypervisor on the right. The Linux kernel schedules and manages the memory of the VMs. The hardware access is provided by the KVM kernel module and supports emulation and paravirtualization. KVM relies on hardware virtualization support of the CPU.*

tation of the partitioning framework.

Using `libvirt` to provide XEN and KVM compatibility

As virtualization on Linux systems got more and more used during the last years, many different virtualization solutions are made available by the Open Source community, each with its own configuration and management tools. To enable compatibility between environments with different virtualization systems, `libvirt` provides a common API for management and configuration of virtual machines. Xen and KVM are supported and therefore it was decided to base the implementation of the ViBatch concept on `libvirt`. This allows to decouple the dynamic virtualization infrastructure from the virtualization layer and thus to switch easily between Xen and KVM and also to benefit from functions and interfaces `libvirt` provides.

By using `libvirt` each VM is defined by a uniform XML configuration file, no matter which virtualization infrastructure is in use. Thus, only the Hypervisor configuration parameter has to be adopted. An example for a `libvirt` XML configuration file as used in ViBatch can be found in appendix B.2.1.

Performance Considerations

To estimate the performance impact of the virtualization layer, several standard benchmarking tools are available. However, each is specialised on testing only a specific part of the system. To have an overall impression of the performance loss for typical HEP applications, which use both, heavy I/O and CPU, detailed comparisons between native and virtualized

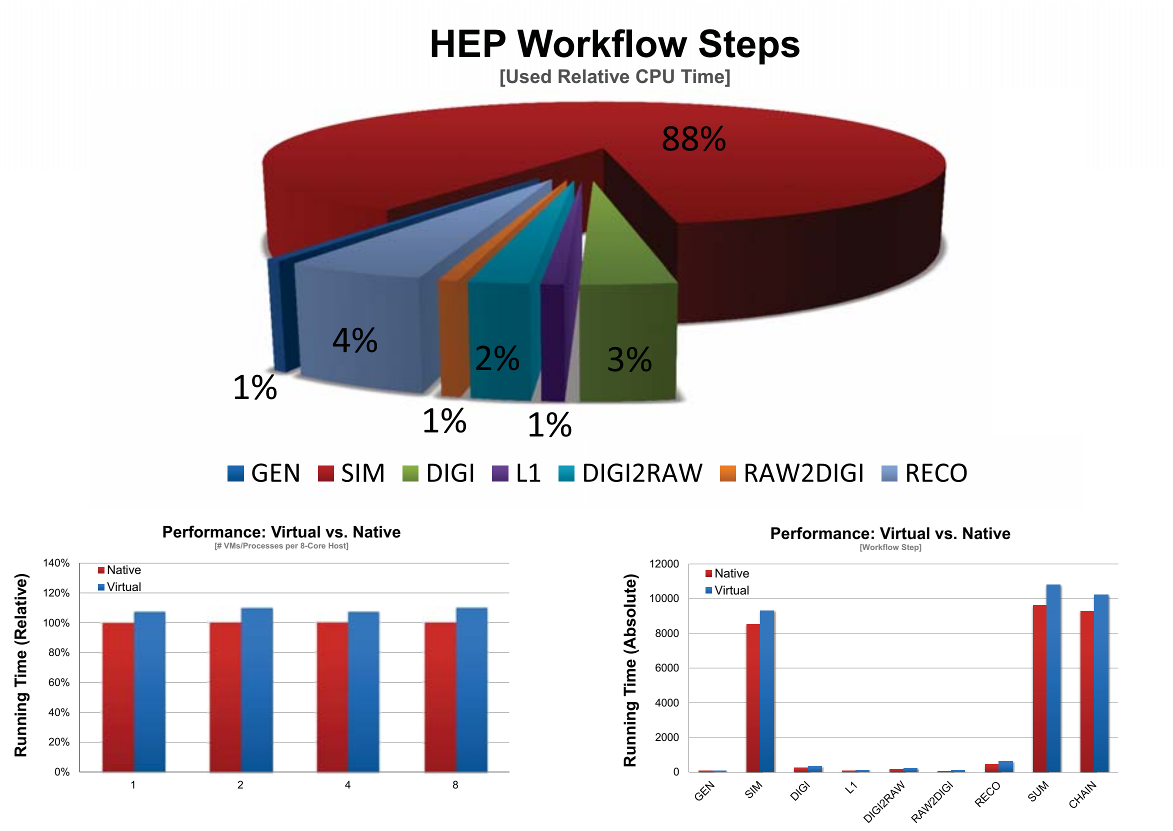

Figure 3.6: *VMBench Results: Fractions of the different analysis steps for a common CMSSW analysis chain (top). Running time comparison between a CMSSW analysis running on the host machine (red) and within a virtual machine (blue) for different numbers of parallel running tasks on one host (lower left). Absolute running time on the host (red) and the VM (blue) for the different analysis steps (lower right). Taken from [69]*

systems can be obtained by using experiment specific software and HEP specific analysis workflows. Former performance tests using CMSSW have shown a performance impact for Xen of a few percent (see [67]). To have a more detailed impression on the performance loss due to KVM virtualization, which is now used within ViBatch, a benchmark suite called VMBench [69] was executed. VMBench exploits a common CMSSW analysis job and adds several monitoring modules, the *watchers*. These modules enable different features to estimate for example the CPU or memory usage of individual analysis steps configured in the CMSSW analysis chain. These steps differ in their resource consumption. Their overall fraction of the required runtime in an analysis job is shown in Fig. 3.6 as well as the overall comparison results. The details on the CMSSW analysis steps can be found in appendix C.4.1.

Preparation of the Virtualized Worker Nodes

A single user or user group can either request a certain software environment including already installed and configured computing software or prepare their virtual machine template on their own.

The virtual worker nodes will be started on the hardware nodes automatically, as discussed in 3.2.1. A very important set-up step is to dynamically configure the network of each individual virtual machine. This is done via DHCP[1]. Almost all other configuration issues like the connection to a network or a cluster file system, can be completed within the virtual machine template before their initial preparation.

Encapsulation of Computing Jobs into Virtual Cluster Partitions using Prologue and Epilogue Scripting Functionality

As mentioned above, a common functionality of batch or workload management systems are the prologue and epilogue scripting functionality which may be run before or after a job execution on the specific worker node. The standard usage of those scripts is the preparation or clean-up of the nodes. In ViBatch, this technique is used to encapsulate the original job, which has to be run within a virtualized worker node, in a so-called “wrap-job”.

The cluster used for the development of ViBatch is divided in two partitions. Some of the worker nodes are virtualized in contrast to the nodes in the basic partition of the cluster that consists of the hardware nodes. Apparently the wrap-jobs are only needed for virtualized partitions. In figure 3.7 the work-flow of the two different job types is sketched.

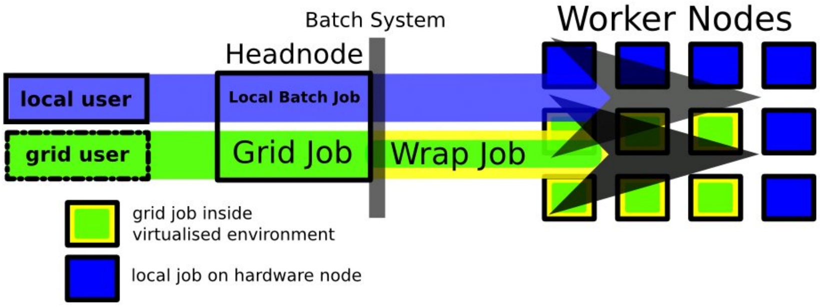

Figure 3.7: *Workflow of the two different job types. Local jobs are executed on the hardware hosts, Grid jobs are wrapped and passed to a VM. Taken from [70]*

Jobs submitted to the basic partition are executed as usual. In contrast to that, jobs that

---

[1] Dynamic Host Configuration Protocol

require a different partition are encapsulated by the prologue and epilogue functionality of the batch system. The prologue script starts the virtual machine on the designated hardware worker node as part of the job. As soon as the virtual machine is started, the original job is passed to the virtual worker node using an `ssh`[2] remote shell.

After the user job completes successfully (or if the user application has failed), the wrap-job will take care of the disposal of the virtual machine. If instead the job is interrupted in its execution by a system signal from the batch system, the termination of the virtual machine will be carried out via the clean-up routines within the batch system's epilogue script. The virtual machine will be instantly destroyed, as a clear shut-down is not necessary. The detailed ViBatch workflow as depicted in Fig. 3.8 can be summarised to:

1. A user submits a job to a batch system. He decides if the job should run on a virtual worker node or the native host OS by submitting to an appropriate queue which needs to be set up on the batch server. Due to that it is easily possible to mix up virtual and native worker nodes on the same cluster.
2. If a virtual queue is selected, the batch system executes the prologue script at the beginning of each job.
3. The prologue script prepares the the virtual machine image by cloning the VM from a provided template on the local worker node hard disk.
4. Modify template to accept the actual user job later on. Currently, a non-password protected and user specific public ssh key is copied to the authorized keys file on the VM.
5. Start the virtual machine via the libvirt API. Hand over a proper MAC address for the virtual network interface to allow individual network setup via DHCP.
6. At the end of the booting process, the VM creates a lock-file via an init script on the local or cluster file system
7. The prologue script checks for this lock-file to guarantee a completely booted VM.
8. The actual user job is piped via ssh to the VM.
9. The user jobs is executed inside the VM
10. After the job has finished and the job output was returned to the user, the epilogue script is executed.

---

[2] Secure Shell Protocol

11. The VM is shut down and destroyed.

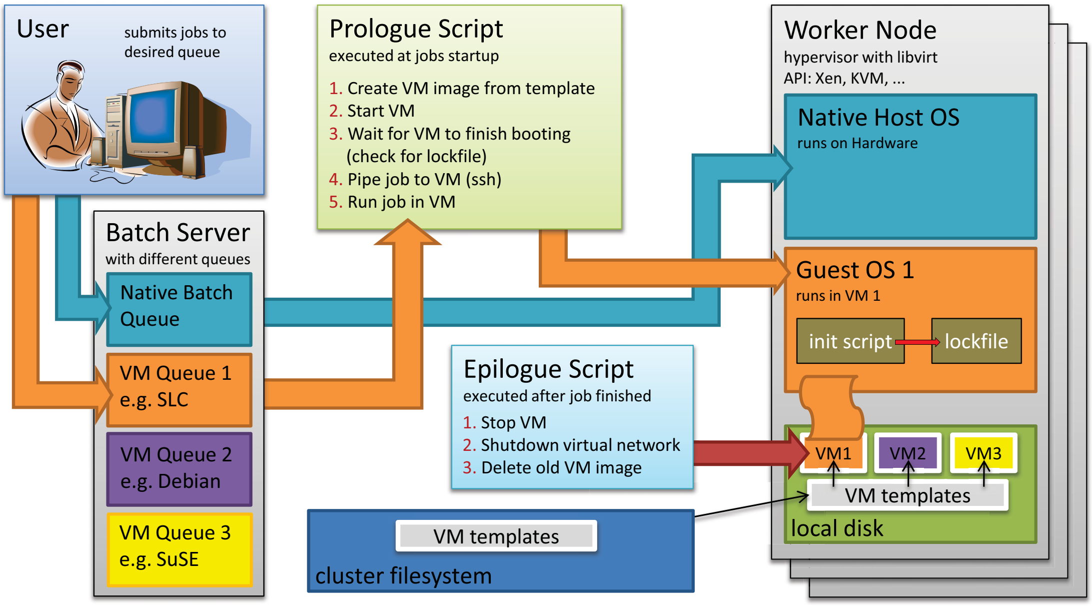

Figure 3.8: *Schematic overview of the ViBatch concept: portable to any batch system with pro- and epilogue scripting functionality, independent from the underlying hypervisor, lightweight setup, transparent to the user, allows a mixed batch system setup with native and virtual worker nodes. Taken from [69]*

The start-up time of a virtualized worker node depends on the OS started within the VM. In most cases it takes less than 30 seconds. This is negligible compared to the mean job execution time of standard HEP workflows. A more detailed documentation and the source code of ViBatch can be found at [71]

Test Environment and Evaluation

The implementation and testing of the ViBatch system was tested on a 1600 core shared computing cluster hosted at the Steinbuch Centre for Computing (SCC) at the Karlsruhe Institute of Technology (KIT), the Institutes Cluster (IC1). A detailed description of the IC1, its infrastructure and setup, can be found in appendix B.3. During the development phase of ViBatch four hardware nodes of the cluster with eight CPU cores each were reserved for virtual machine queues. As only small changes were needed in the base configuration of the batch server, the production batch system could be used in parallel for the development and testing of ViBatch and ongoing physics analyses jobs running on the native worker node hardware. An estimated amount of about one hundred thousand jobs were launched during the different development phases and in the final testing about fifty thousand test jobs were executed successfully.

At the time of writing this thesis, preparations for running ViBatch in production mode, by opening the queues for all users and a final large scale test including all 1600 cores of the IC1 cluster were ongoing.

Similarities to Cloud Computing

During the last couple of years, *Cloud computing*, a new omnipresent buzzword captured the IT sector. Without a precise definition, although a vast amount of interpretations exist, Cloud computing can be described as the umbrella term for the provisioning of IT platforms, infrastructures and applications as services. A broad overview of the topic is provided in [72]. While standard Cloud computing solutions provide applications and platforms as services, the provision of High Performance Computing environments is a quite new facet. Although ViBatch is a locally applied cluster virtualization technique it provides the basic requirements, to be classified as *HPC as a Service* (HPCaaS) private Cloud software. By dynamically requesting public Cloud resources from a third party provider and interfacing these resources to the local batch system and ViBatch, the local computing cluster can be extended to address peak loads. The required software aspects for such a solution, e.g. running a local batch system with Cloud worker nodes, are evaluated and developed the IEKP resulting in a tool called ROCID [73]. An assembly of ROCID and ViBatch is part of further planned studies within the IEKP computing concepts.

3.3 Summary

Providing an encapsulated and user defined environment on a shared computing cluster infrastructure is a key feature for current High Performance and High Throughput Computing resource providers. As a result, the virtualization of worker nodes in batch systems became very popular. Although, several commercial batch systems nowadays implement the use of virtualized worker nodes, the idea of using basic, already existing, low level and therefore wide spread functionalities lead to a highly portable solution. As it is intrinsically highly flexible in the choice of the used virtualization layer and batch system solution exploiting the prologue and epilogue batch system functionality, an intervention in and a modification of the used software tools is not necessary.

ViBatch is implemented in a given batch infrastructure just by minor configuration changes such as adding virtual machine queues and enabling pro- and epilogue scripting functionality. The development and evaluation could be applied within the production system on the IC1 cluster at the SCC without major disturbance of the provided services. The full production operation will be launched after the last full scalability test by using all 1600 cores of the cluster, which is currently in preparation. Future plans involve the usage of public Cloud resources via ”cloudburst“ techniques. Such a concept will allow to intercept peak loads on the IC1 and future clusters, and will drastically improve the analysis throughputs for all local physicist.

Measurment of the Double Differential Inclusive Jet Cross Section at 7 TeV with CMS

The first task of a new collider and the associated experiments is to check measurements which are early practicable, for compatibility with extrapolations using the Standard Model from measurements of other experiments like LEP and Tevatron. One of these analyses is, due to the high jet production rate, the measurement of the inclusive double-differential jet cross-section. In the high energetic proton-proton collisions of the LHC exceeds the jet production rate of the Tevatron by several orders of magnitude. Although the LHC operation is at a very early stage, the inclusive jet cross-sections analysis presented in this work gives a first insight in the comparison of the new data with the latest theoretical predictions.

During the first data taking with the CMS experiment, the experiment setup and conditions change rapidly, and a precise knowledge of these conditions during the analysis is vital for a proper measurement. This will be discussed in detail. Findings of this work were contributed to the inclusive jet measurement by CMS which are presented in [74]. A detailed overview of the used and also partially developed analysis software within this work is given in appendix C.4.

rapidity	CMS Detector region
$0 \leq \lvert y \rvert < 0.5$	central barrel
$0.5 \leq \lvert y \rvert < 1.0$	central barrel
$1.0 \leq \lvert y \rvert < 1.5$	central barrel
$1.5 \leq \lvert y \rvert < 2.0$	barrel-endcap transition
$2.0 \leq \lvert y \rvert < 2.5$	endcap
$2.5 \leq \lvert y \rvert < 3.0$	endcap

Table 4.1: *Rapidity bins and the corresponding detector regions.*

4.1 The Observable

After cleaning up the gathered data from non-collision events and noise jets the inclusive jet cross-section is measured double-differentially in bins of jet transverse momentum p_{T} and rapidity y. It is defined as:

$$\frac{d^2\sigma}{dp_{\mathrm{T}}dy} = \frac{C_{\mathrm{res}}}{\mathcal{L} \cdot \varepsilon} \cdot \frac{N_{\mathrm{jets}}}{\Delta p_{\mathrm{T}} \cdot \Delta y} \tag{4.1}$$

where:

- N_{jets} is the number of jets counted in a bin,
- $\mathcal{L}$ is the integrated luminosity,
- ε is the efficiency of the event and jet clean-up,
- C_{res} is the resolution unsmearing correction factor,
- Δp_{T} and Δy are the transverse momentum and rapidity bin sizes respectively.

4.1.1 Binning

The binning in absolute rapidity $|y|$ has been defined according to the CMS detector geometry. The bin sizes are $\Delta y = 0.5$ and the inclusive jet cross section in the following is measured in six bins up to $|y| = 3.0$ as presented in Tab. 4.1.

The binning in calorimeter jet p_{T} is chosen to be compatible with the jet energy resolution of the CMS detector. The bin size should be roughly proportional to the jet energy resolution at the centre of the respective jet p_{T} bin. The high jet p_{T} bins are merged, that at least five jets are expected within the bin, based on the NLO prediction. The complete table of the used jet p_{T} bins can be found in appendix C.2

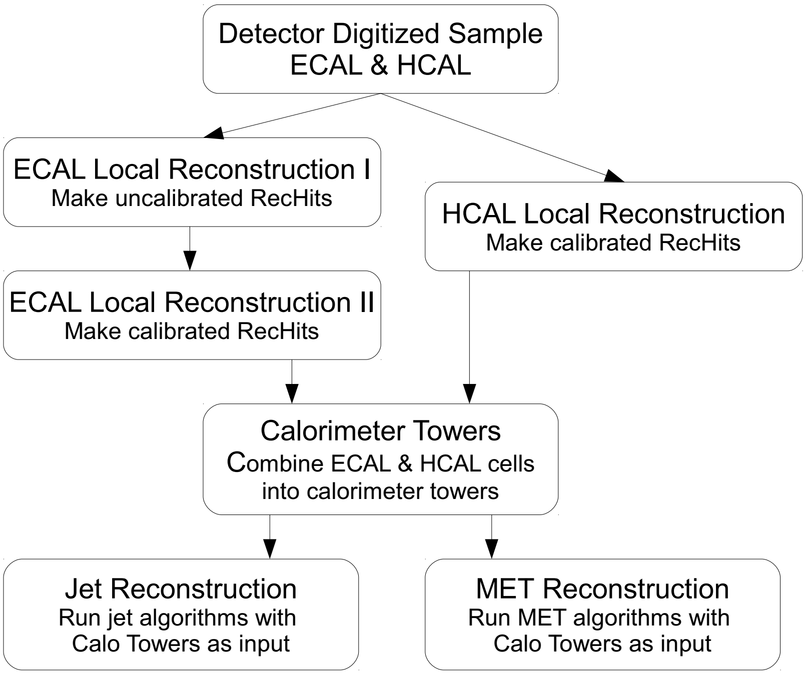

Figure 4.1: *Calorimeter jet reconstruction within CMS. This figure shows the steps to produce jets and missing transverse energy (MET) objects from the non-linear response to the ADC counts in the calorimeters cells.*

4.2 Jet Reconstruction

The calorimeter jets used in this analysis are reconstructed following the chain presented in Fig. 4.1. Several steps are required to derive jets found by the different jet algorithms from the ADCs[1] output of the calorimeter. The ADC counts are converted back to energy per calorimeter cell, the so-called *hits*. The hits from 5×5 ECAL-crystals and a single HCAL cell are combined into projective towers which correspond to the HCAL granularity. Although the HCAL-ECAL geometry not projective within the endcap, calorimeter towers are also formed. The derived four-vectors of the calorimeter towers are then used as input for the jet finding algorithms.

To measure the double differential inclusive jet cross-section, this analysis uses the collinear and infrared safe anti-k_T algorithm with a radius of R $= 0.5$, which is the current standard within the CMS Collaboration for this algorithm.

---

[1] Analogue-to-Digital Converters

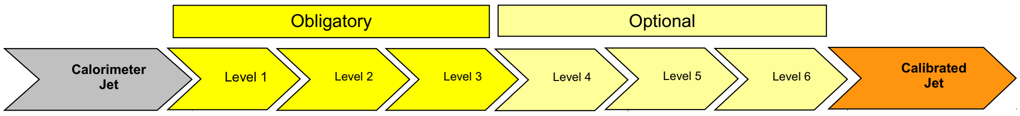

Figure 4.2: *Factorized jet energy corrections within CMSSW. The first three correction levels are obligatory.*

4.2.1 Jet Energy Scale

The jet energy from reconstructed jets has to be corrected for a multitude of effects. CMS therefore uses a factorized jet energy correction (JEC) approach [75] as pointed out in Fig. 4.2. The correction levels are:

- Level 1: *Offset* correction [76] - corrects for additional jet energy due to electronic noise and pile-up as well as the energy loss from calorimeter thresholds.
- Level 2: *Relative* correction - equalises the jet response[2] in pseudo-rapidity.
- Level 3: *Absolute* correction - restores jet response of the transverse momentum on average to unity as a function of p_{T}.
- Level 4: Electromagnetic energy fraction (EMF) corrections.
- Level 5: Jet flavour corrections - corrects discrepancies in the jet response due to the different jet flavours at particle level.
- Level 6: Underlying Event corrections - corrects jets to subtract additional energy originated in multiple parton interactions MPI
- Level 7: Parton jet corrections - corrects the jets to parton level.

These corrections are determined with several data driven methods like the di-jet balancing [77] for the relative correction or the absolute correction derived from γ+jet [78] or Z+jet [79] balancing. At the time of this analysis, the corrections factors were derived from simulation, as there was not enough collision data to measure the different correction functions and the respective uncertainties. After applying the relative corrections on the calorimeter jets, a residual discrepancy in the jet energy response, relative to the central region, is observed in the data compared to the simulation. This deviation is linear in the pseudo-rapidity η and rises up to $5\,\%$ in the highest pseudo-rapidity regions. Therefore an exclusive di-jet sample was used to measure this discrepancy and to provide additional corrections factors, which were then applied to the data after the level 2 and level 3 corrections [80].

---

[2] $R_{\text{jet}} = \frac{p_{\mathrm{T,particle\ level\ jet}}}{p_{\mathrm{T,calorimeter\ jet}}}$

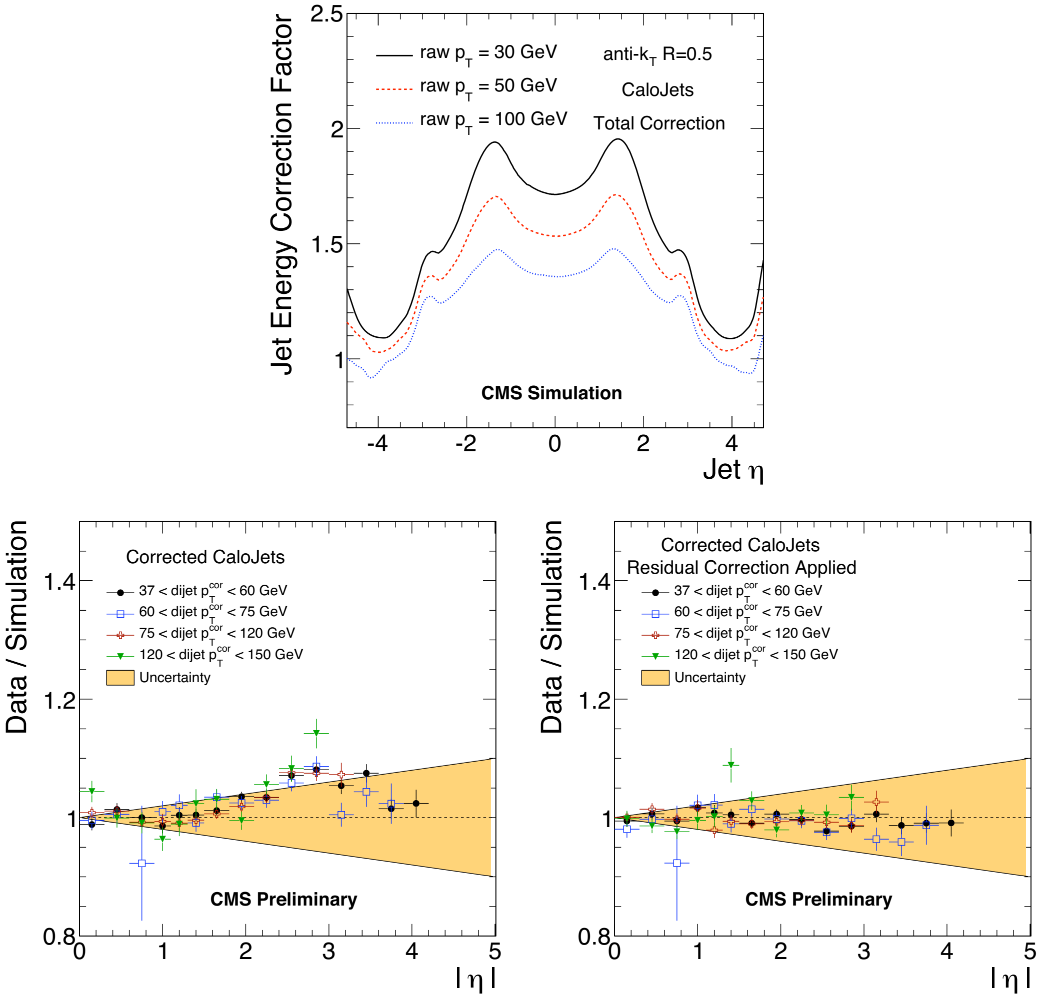

Figure 4.3: *Applied absolute JEC (upper) and closure for level 2 and level 3 corrected calorimeter jets before (lower left) and after applying the additional residual corrections. From [80]*

The advantage of such factorized jet energy scale corrections, is that the different corrections and their uncertainties can be determined and understood almost independently and therefore give a more detailed insight into the detector behaviour and physical side effects.
The jet energy scale (JES) is one of the main sources for systematic experimental uncertainties within the inclusive jets cross-section measurement and will be discussed in detail in section 4.5.1.

4.2.2 Jet Energy Resolution

After applying the JES corrections, a second important detector effect has to be taken into account for the inclusive jets measurement. The resolution of the calorimeter response for single particles is finite, resulting in a finite width of the jet energy response. This effect has an impact on the steadily falling double-differential inclusive jet p_{T} spectrum, as it is responsible migration of jets into the wrong jet p_{T} bins.
Analogous to the JES correction determinations, the jet energy resolutions can be evaluated in two different ways. In the first place, the resolution can be estimated by a Monte Carlo truth analysis. Additionally, the di-jet asymmetry method provides the measurement of the jet energy resolution as soon as enough data is available. Both methods are presented in [80].
In the MC truth analysis, the jet energy response is determined by associating generated particle jets to fully simulated, reconstructed and JES corrected ones. Through a matching procedure a jet pair is formed. A particle jet is associated to the closest reconstructed jet within η - ϕ space via

$$\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta y)^2}. \tag{4.2}$$

Only jet pairs with $\Delta R < 0.2$ were then used to derive the difference in p_{T} of the matched jet pairs. The choice of the matching distance ΔR impacts only at low-p_{T} on the widths of the response distributions, but is negligible for a jet $p_{\mathrm{T}} > 40$ GeV. Additionally only the two matched pairs with the highest particle jet p_{T} are considered. In Fig. 4.4 the resulting jet p_{T} response distribution $p_{\mathrm{T}}/p_{\mathrm{T}}^{REF}$ from [80] is shown for calorimeter jets in $|\eta| < 0.5$ and within a particle level jet p_{T} range of 230 GeV $< p_{\mathrm{T}}^{REF} < 300$ GeV.
The MC truth jet resolution distribution can be approximately described by the σ of a Gaussian fit to the central values of the jet response distribution. The non-Gaussian tails can be better modelled by a double-sided Crystal Ball fit, which describes the low and high tails by a power law. Both fits are displayed in Fig. 4.4.

The current studies of the JER have shown, that for the double-differential inclusive jet measurement the distortion of the steadily falling jet p_{T} spectrum due to JER effects, e.g. the non-Gaussian tails of the p_{T} response, can be neglected compared to the large uncertainties

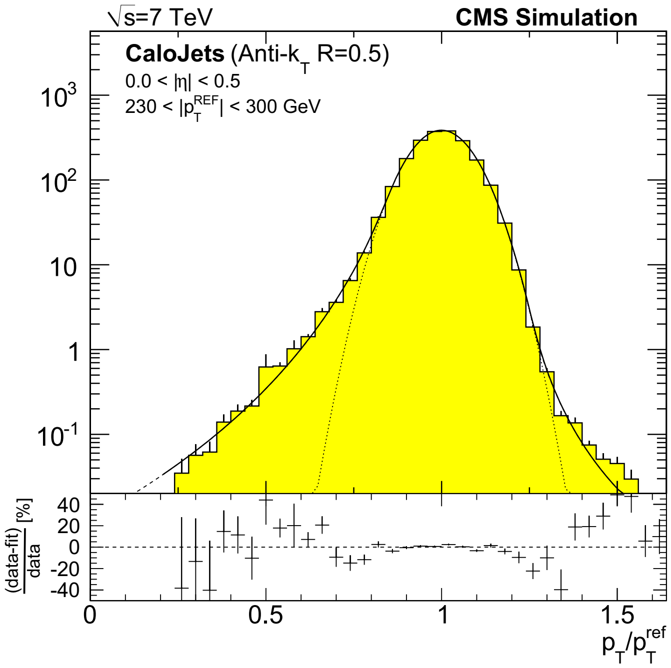

Figure 4.4: *Jet energy response distribution for a specific $|\eta|$ and jet p_T range with Gaussian and double-sided Crystal Ball fits. The residuals in percent for the Crystal Ball fit is shown at the bottom. From [80]*

arising from the inaccurate knowledge of the JES. During the unfolding procedure of the measured double-differential inclusive jet cross-section, which is described in section 4.4.1, it was therefore sufficient to describe the resolution by the Gaussian core.

4.3 Event and Jet Selection

The data recorded by the CMS detector consist not only of collision events. Electronic noise and energy deposits within the detector from particles not produced by proton-proton collisions at the interaction point, e.g. collisions of protons and residual gas molecules within the beam pipe, can add non-signal data. Thus, several selection steps have to be passed by a CMS event in data to be added to the actual measurement.

To ease the workflow of processing the data during the analyses, the events are separated in data subsets sorted by several different trigger patterns. During the start-up of the CMS data taking at 7 TeV these trigger requirements were very loose, to enable the determination of trigger efficiencies and to gather the highest possible amount of collision data. The data analysed here, were recorded from May to July 2010 with a detailed table of the used datasets given in appendix C.1. The following sections give an overview of the different event selection criteria which were applied within this analysis.

4.3.1 CMS Data Taking Runs and Luminosity Sections

The data taking periods of the CMS detector are called *runs*. A run consists of multiple luminosity sections, which is a predefined period of data taking where the instantaneous luminosity can be considered constant. This separation is essential for the bookkeeping of the integrated luminosity. The LHC beam orbit gaps every 90 μs provide natural boundaries for the luminosity sections. This gap triggers the periodic resynchronisation of the CMS front end electronics. At a start of a data taking run, the first orbit with valid physics data (FVO) is distributed to all DAQ applications. Beginning with the FVO, a luminosity section is then defined as a fixed amount of subsequent beam orbits. The luminosity section size is predefined within the DAQ configuration and can be adjusted to match requirements of the detector readout electronics.

If during a data taking run the CMS detector status changes, e.g. a detector sub-system had to be shutdown due to hardware failures, the respective lumi-segments or the entire run is rendered unusable for physics analyses. The data streams are still written and stored within the data sub-sets. However the valid luminosity sections and runs are recorded within special JavaScript Object Notation (JSON) files, provided by the CMS Data Quality Monitoring (DQM) team. These files can then be used within the physics analyses to blacklist luminosity sections or entire runs which are not qualified for physics analyses. The integrated luminosity of the used events can then be calculated by querying a central luminosity database for the used luminosity sections and runs. The JSON files used for this analysis can be found in appendix C.2.

4.3.2 Trigger Requirements

The datasets used in this analysis contain events which triggered the lowest single jet trigger, implying that at least one jet is present within each event stored in the datasets. Two additional trigger requirements were used in the inclusive jets analysis. The first selection depends on Level 1 (L1)triggers and purges the events for beam halo effects, originated from interactions of the beam with residual gas molecules in the beam pipe. The second selection is based on the high level trigger (HLT), which selects events with a specific single jet trigger at a certain jet p_T threshold.

- L1 trigger requirements: The so-called *beam halo veto* is given when one of the following L1 technical trigger bits 36,37,38,39 has fired. These events are discarded.
- HLT requirements: The `HLTJet30U` trigger must have fired. It selects events with at least one calorimeter jet with an uncorrected $p_\mathrm{T} > 30$ GeV.

During the data taking period covered by this analysis no additional pre-scaling factor had to be applied to the `HLTJet30U` spectrum as all triggered events were stored within the datasets. With increasing luminosity the HLT jet trigger for jets with low p_T will get a prescale factor, such that the specific HLT triggered events will be randomly dropped to reduce the overall data stream from the detector. Therefore future measurements of the double-differential inclusive jet cross section will have to combine the different trigger streams with different pre-scale factors into a smooth, continuous jet p_T spectrum.
The turn-on point for a jet trigger is defined as the JES corrected jet p_T value at which the trigger efficiency is at least $99\,\%$ with respect to the preceding HLT single jet trigger. Fig. 4.5 shows the turn-on curve in the central rapidity region. The turn-on curve can be described by an error function[3]:

$$\mathrm{erf}(u) = \frac{2}{\sqrt{\pi}} \int_0^u e^{-t^2} dt, \tag{4.3}$$

with the argument $u = ax + b$. This defines the turn-on point as the x, where $\mathrm{erf}(u) = 0.99$. According to the turn-on curve jets with a JES corrected p_T of above ≈ 50 GeV can be used

---

[3] also called the Gauss error function or probability integral

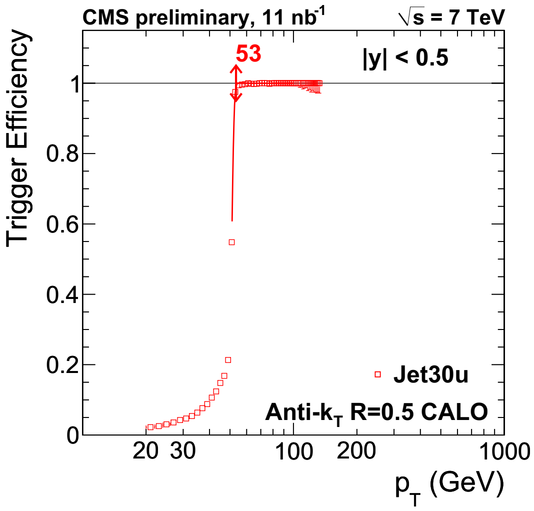

Figure 4.5: *Trigger turn-on curve for the central rapidity bin from [74]. The estimated turn-on point, where the trigger is more than 99 % efficient,is marked*

in the analysis. In this analysis the jet p_{T} spectrum is starting at 64 GeV. The applied trigger selection ensures that the efficiency ϵ in Eq. 4.1 is close to unity.

4.3.3 Vertex Requirements

The events passing the trigger requirement have to have a primary vertex (PV) which is consistent with the measured transverse position of the beam, the so-called *beam spot*. The primary vertex is reconstructed from particle tracks by sophisticated algorithms, described in [81]. The following selection criteria for the vertex are applied:

- The event must have at least one reconstructed primary vertex.
- The z-coordinate of the primary vertex is confined in the luminous region $|z_{\mathrm{PV}}| < 15\,\mathrm{cm}$.
- The position of the PV has to be determined by using at least five reconstructed particle tracks.
- The radial distance to the beam spot must be less than $0.15\,\mathrm{cm}$.

This primary vertex selection rejects beam related non-collision noise events very efficiently, which maintains a close to $100\,\%$ event selection efficiency ($\epsilon \approx 1$ in Eq. 4.1).

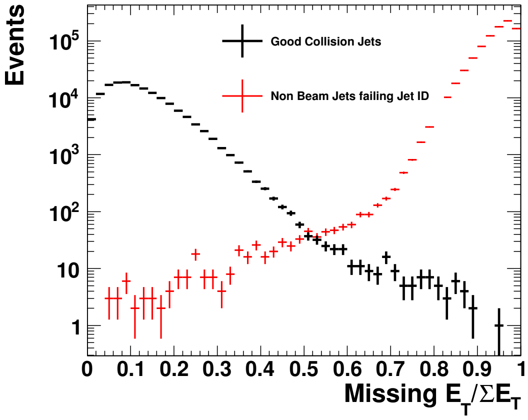

Figure 4.6: *MET/$\sum E_{\mathrm{T}}$ for collision and non-collision events. [82]*

4.3.4 Missing Transverse Energy Selection

Spikes in calorimeter cells due to electronic noise or cosmic ray showers will produce a vast amount of artificial transverse energy E_{T} in an event without being balanced by any physical object within the detector, resulting in the so-called missing E_{T} (MET). By comparing MET and the scalar sum of all transversal energies ($\sum E_{\mathrm{T}}$) in an event, one gets the relative missing E_{T} (MET/$\sum E_{\mathrm{T}}$). Due to the finite jet resolution and the imprecise JES corrections, collision events feature non-zero but never the less low values of MET/$\sum E_{\mathrm{T}}$ while spike events are extremely unbalanced in the transversal plane leading to high values of MET/$\sum E_{\mathrm{T}}$. A comparison of jets from collisions and jets from non-collision events is given in Fig. 4.6. The figure shows the efficiency of this event selection cut and its rejection power as a function of the MET/$\sum E_{\mathrm{T}}$ cut value. Below a MET/$\sum E_{\mathrm{T}} \approx 0.5$ more non-collision events than good collision events are rejected. In order to keep the efficiency factor ϵ from Eq. 4.1 consistent with unity, events with MET/$\sum E_{\mathrm{T}} > 0.6$ were selected according to efficiency studies shown in Fig. 4.7

4.3.5 Jet Selection

In addition to the event based clean-up selections, a per jet clean-up was applied to further reduce the impact of calorimeter noise and therefore to reduce the unphysical energy deposits in the calorimeters. Several quantities of the HCAL and ECAL cells which are clustered into a jet are stored within the data and are combined in the so-called CMS *JetID* [83].

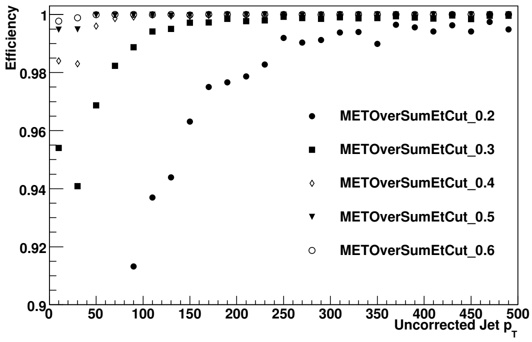

Figure 4.7: *MET/$\sum E_\mathrm{T}$ selection efficiency for different selection values as a function of uncorrected jet p_T. [82]*

JetID Variables

In order to asses the likelihood of a jet originating from the interaction of the proton-proton beam collision, the variables listed below have been considered:

- f_EM: Fraction of the jet energy (ECAL+HCAL) measured by the ECAL cells.
- N_towers: Total number of calorimeter towers which are clustered into a jet.
- $N_{\mathrm{cells}^{90}}$: The minimum number of calorimeter cells clustered into a jet containing $90\ \%$ of the jet energy.
- $\sigma_{\phi\phi}$: The root mean squared of the transverse energy weighted ϕ distribution of the calorimeter towers clustered into a jet.

The majority of these variables are an indicator, if the jet was found in more than one subdetector (e.g. the jet energy is distributed among the electromagnetic and hadronic calorimeter cells).
Additional more hardware related variables are part of the JetID:

- f_HPD: Energy fraction contributed by the hottest (highest energy) hybrid photo-diode (HPD) readout.
- f_RBX: Energy fraction contributed by the hottest readout box (RBX), which combines several HPDs.
- R_HF: $(E_\mathrm{S} E_\mathrm{L})/(E_\mathrm{S} + E_\mathrm{L})$, E_S and E_L are the energies measured in the short and long fibres of the hadronic forward calorimeters respectively.

These quantities are based on the shape of the cluster of the energy deposits in a single detector sub-system and are expected to be circular in the $\eta \times \phi$ plane for physical jets. In contrast to that, electronic noise will produce distinct patterns over a number of readout channels.
This analysis uses the *loose* form of the JetID clean-up requirements, which mainly addresses spikes in the HCAL resulting in fake jets and uses a subset of the previous itemized variables:

- $f_{\text{EM}} < 0.01$: At least $1\,\%$ of the total measured jet energy is contained in the ECAL (clears noisy HCAL cells).
- $N_{\text{cells}^{90}} > 1$: If $90\,\%$ of the jet energy is contained in a single calorimeter hit, it is discarded.
- f_{HPD} & $f_{\text{RBX}} < 0.98$: If $98\,\%$ of the jet energy is contributed by a single photo-diode or readout box, it is discarded.

The JetID and the missing transversal energy selection extremely superimpose each other. However both are considered necessary to reject noise efficiently. As the jet selection only rejects noise jets the efficiency factor ϵ still can be assumed to be close to unity.

4.3.6 Summary

To separate non-collision or noisy events from clean proton-proton beam collision events the following selections were applied:

1. Beam halo veto: none of the L1 technical bits 36, 37, 38 or 39 have fired.
2. The HLTJet30U (Uncorrected jet $p_{\text{T}} > 30$ GeV) trigger must have fired.
3. Primary vertex within $|z_{\text{PV}}| < 15\text{cm}$, number of tracks used for the vertex fit: $\text{nDOF} > 4$, $\rho < 2\text{cm}$
4. MET/$\sum E_{\text{T}} < 0.6$

Additionally the jets in the selected events had to pass the JetID quality criteria. Tab. 4.2 lists the event numbers which pass the respective selections. Finally, all selections combined provide about one million events, which can be used within the measurement of the double-differential jet cross section.

Selection criteria	number of Events
Total events in the dataset	33611455
JSON selection of certified runs	5090481
Beam halo veto	3887201
Vertex and MET/$\sum E_{\mathrm{T}}$ selection	3855545
Events which triggered HLTJet30U	1341387

Table 4.2: *Impact of the different event selection cuts. The number of each cut include only the JSON selection and the respective cut itself.*

4.4 The Inclusive Jet Spectrum

In the early phase of CMS data taking, the inclusive jet p_{T} spectrum can be acquired by just selecting the lowest un-prescaled HLT. By using the HLTJet30U trigger and starting sufficiently above the trigger turn-on at a jet p_{T} of 64 GeV the inclusive jet p_{T} spectra was constructed as it is given in Fig. 4.8. The spectrum is normalized to the bin widths in p_{T} and $|y|$ and to the estimated integrated luminosity of 60 nb^{-1}. The high p_{T} bins are merged, such that at least 5 jets are expected in each bin according to the NLO calculation.

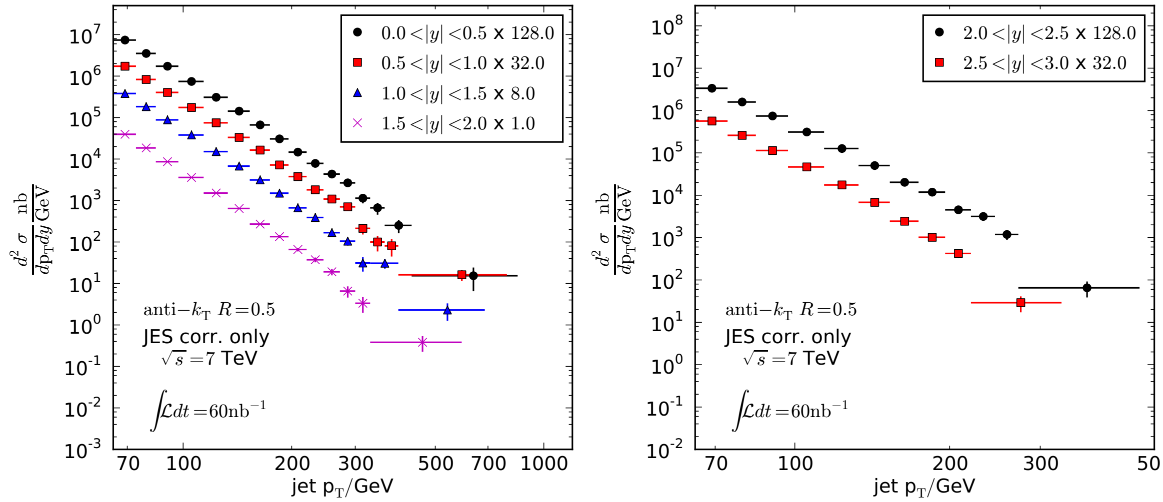

Figure 4.8: *Measured and JES corrected inclusive jet cross section as a function of jet p_{T} for different rapidity ranges.*

4.4.1 Unsmearing the Spectrum

The measured JES-calibrated jet p_T spectra are corrected for resolution effects using an *ansatz* [84,85]: it is assumed that the true jet p_T spectrum can be modelled by a parametrised function of jet transverse momentum,

$$f(p_\mathrm{T}) = N \cdot p_\mathrm{T}^{-\alpha} \cdot \left(1 - \frac{2\cosh(y_{min})p_\mathrm{T}}{\sqrt{s}}\right)^{\beta} \exp(-\gamma p_\mathrm{T}) \tag{4.4}$$

The ansatz is based on phenomenological fits with the NLO theory, whereas the power terms $p_\mathrm{T}^{-\alpha}$ and $\left(1 - \frac{2\cosh(y_{min})p_\mathrm{T}}{\sqrt{s}}\right)^{\beta}$ are associated with the hard particle production and the suppression effect at the edges of the parton distribution function phase space, respectively. The value of β is fixed to $\beta = 8$ because of the low kinematic range accessible in the current analysis. The ansatz function is then smeared using a Gaussian approximation

$$R(p'_\mathrm{T}, p_\mathrm{T}) = \frac{1}{\sqrt{2\pi}\sigma(p'_\mathrm{T})} \exp\left[-\frac{(p'_\mathrm{T} - p_\mathrm{T})^2}{2\sigma^2(p'_\mathrm{T})}\right] , \tag{4.5}$$

where $\sigma(p_\mathrm{T})$ is a parametrization of the relative σ dependence of p_T (Equation 4.4.1):

$$\frac{\sigma(p_\mathrm{T})}{p_\mathrm{T}} = \sqrt{\mathrm{sqn}(N) \cdot \left(\frac{N}{p_\mathrm{T}}\right)^2 + S^2 \cdot p_\mathrm{T}^{m-1} + C^2} \, . \tag{4.6}$$

The convolution of the jet p_T function in Eq 4.4.1 and the detector resolution results in the measured cross-section:

$$F(p_\mathrm{T}) = \int_0^\infty f(p'_\mathrm{T}) R(p'_\mathrm{T}, p_\mathrm{T}) dp'_\mathrm{T} \tag{4.7}$$

During the unsmearing process, the measured jet p_T spectrum is fitted with the ansatz function and the multiplicative unsmearing correction factors C_res from Eq. 4.1 are calculated via:

$$C_\mathrm{res,bin} = \frac{\int_\mathrm{bin} f(p_\mathrm{T}) dp_\mathrm{T}}{\int_\mathrm{bin} F(p_\mathrm{T}) dp_\mathrm{T}} \, . \tag{4.8}$$

The measured cross sections in Fig. 4.8 is fitted successfully with the smeared ansatz function as indicated in the fit quality plot on the left in Fig. 4.9). The derived unsmearing factors for the different rapidity bins are shown on the right side. Applying these factors to the JES-calibrated spectra results in the final double-differential inclusive jet cross-section spectra given in Fig. 4.10.

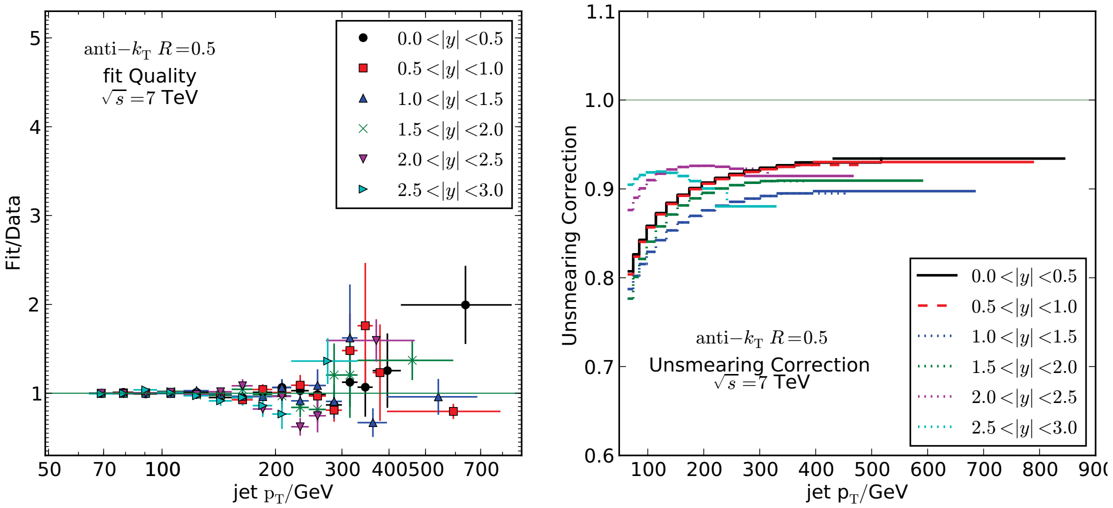

Figure 4.9: *Unsmearing fit quality (left) as the fractional difference between the fit function and the data points with statistical error bars. Unsmearing correction factors (right) derived by the unsmearing procedure.*

4.5 Systematic Experimental Uncertainties

The dominant systematic uncertainties in the measurement of the inclusive jet cross-section are due to the imprecise knowledge of the jet energy scale, the jet energy resolution and the uncertainty on the total integrated luminosity, summarized in Fig. 4.11.

4.5.1 Jet Energy Scale Uncertainty

The small integrated luminosity available for this analysis prevented a precise measurement of the JES and the respective uncertainty by using data driven methods. However, by following official guidelines of the CMS collaboration, several facts allow to understand the quality of the JES corrections derived from simulation:

- Comparisons of simulation and measurements of the π^0 mass peak set a limit for the ECAL energy scale uncertainty. It is estimated to be better than $1\,\%$ [86].
- By using isolated hadrons, the response in the HCAL to high p_T tracks of single pions only deviates to a few percent between data and simulation [86, 87].

These two facts together indicate that the uncertainty of the JES for calorimeter jets can be evaluated to be $10\,\%$.

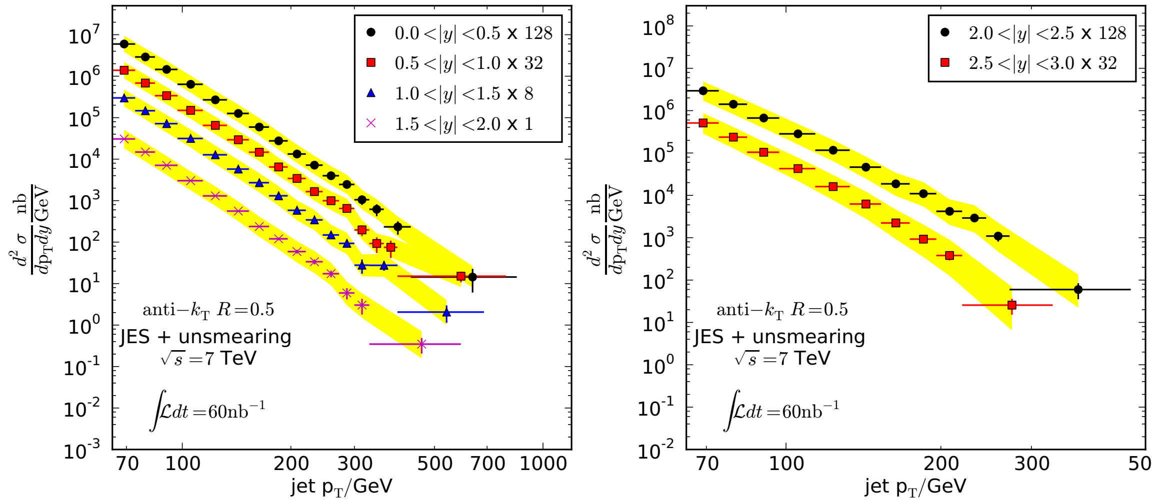

Figure 4.10: *The final unsmeared cross-section spectra for the central (left) and the outer (right) rapidity bins. The yellow band describes the systematic experimental uncertainties. The statistical uncertainties are attached to the data points.*

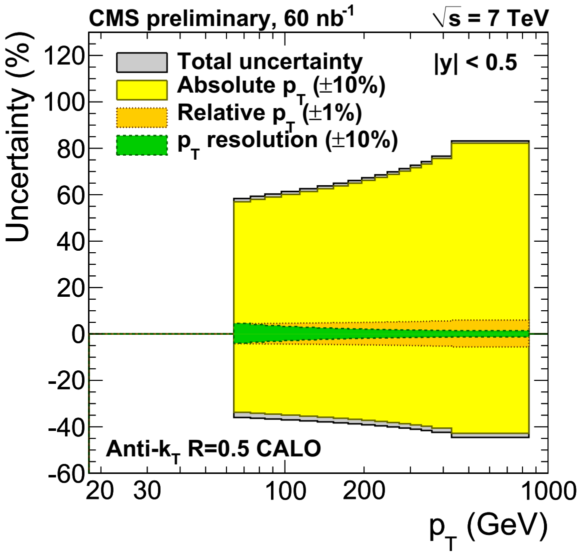

Figure 4.11: *Fractional experimental systematic uncertainties for the central rapidity bin. The grey band for the total systematic uncertainty includes an estimated 11 % uncertainty from the integrated luminosity measurement. The JES uncertainties are split into an absolute and a relative part. Plot taken from [74].*

4.5.2 Jet Energy Resolution Uncertainty

The quality of the unsmearing of the measured inclusive jet cross-section is strongly connected to a precise knowledge of the jet p_{T} resolution. However, precise quantitative estimates of the jet energy resolution were also not yet gained by data driven methods like the asymmetry measurements of the dijet balancing. Analogous to the estimation of the JES uncertainties, experimental evidence exists which leads to the conclusion, that the jet energy resolution uncertainties, derived from simulation, are under control:

- Comparisons of simulation and measurements of the π^0 width set a limit for the ECAL energy scale uncertainty. It is estimated to be better than $3\,\%$ [86].
- Meson and baryon resonances measured within the CMS tracker and their measured widths are compatible with the simulation to a very high precision - in many cases better than $1\,\%$ [88].
- Test beam data show an agreement between the measured hadron energy resolution and the simulation of about $10\,\%$.

With the listed facts, a $10\,\%$ uncertainty of the jet energy resolution can be assumed for calorimeter jets.

4.5.3 Luminosity

Within CMS the luminosity of the proton-proton collisions in the interaction point is measured with the two forward calorimeters (HF). The HF covers the pseudo-rapidity range between $3 < |\eta| < 5$. Additionally, multiple tracks can be referred to vertices within the interaction region, which provides an additional luminosity measurement. Both measurements are found to be compatible with each other. The precise measurements of the LHC beam currents and the transverse size of the beams yield an $11\,\%$ accuracy on the absolute calibration of the luminosity determinations [89]. This accuracy for the calibration can be directly translated in an $11\,\%$ uncertainty on the luminosity normalisation of the inclusive jet cross-section spectra.

4.6 Theoretical Predictions

The inclusive jet cross sections are currently described best by the next-to-leading order perturbative calculations in QCD. These calculations rely on different inputs. In proton-

proton collisions the parton distribution has to be taken into account as well as the strong coupling α_s. This section will give a brief introduction into the used theoretical predictions and the associated uncertainties which are used for comparison with data.

4.6.1 Next-to-leading Order Cross-Section Calculations

Perturbative QCD theoretical predictions for the double-differential inclusive jet cross section at next-to-leading order were calculated using NLOJET++ [90] within the fastNLO [91, 92] framework using the CTEQ-6.6 PDF sets [93]. The NLO predictions are generally an approximation to the full theory. Therefore, they feature several uncertainties:

- **Scale Uncertainty:**
  The cross-section for hadron-hadron collisions to order n in the strong coupling can be written as:

  $$\sigma_{\mathrm{h}_1,\mathrm{h}_2} = \sum_{k=0}^{n} \alpha_s^k(\mu_R) \sum_{\text{flavour i}} \sum_{\text{flavour j}} c_{i,j,k}(\mu_R, \mu_F) \times f_i^{\mathrm{h}_1}(x_1, \mu_F) \times f_j^{\mathrm{h}_2}(x_2, \mu_F) \; , \quad (4.9)$$

  depending on α_s, perturbative coefficients $c_{i,j,k}$, the parton density functions $f_i^{\mathrm{h}_1}(x_1, \mu_F)$, $f_j^{\mathrm{h}_2}(x_2, \mu_F)$ of the respective hadrons h_1 and h_2, the momentum fractions x_1 and x_2, and the renormalization and factorisation scale μ_R and μ_F.
  To evaluate the quality of the approximation to order n in α_s two different ways are possible. Either, the full result is known or at least the next term in the perturbation chain to next-to-next-to-leading (NNLO) can be computed (see [94]). Both are not possible for the inclusive jet cross-section. Instead, scale variations of μ_R and μ_F are used which enter Eq. 4.9 through $\alpha_s(\mu_R)$ due to the renormalization of ultraviolet divergences and the PDFs $f_i(x, \mu_F)$ due to the factorisation in long-distance hadronic physics and short distance hard scatters. Within the full theory, the dependence on the scales would be exactly cancelled via the μ_R and μ_F dependence of the renormalization coefficients $c_{i,j,k}$. As the perturbation series is interrupted at NLO, a residual discrepancy remains, which dependence on the missing orders and can therefore be interpreted as a degree of the approximation quality.
  In this case, the resulting scale uncertainty is estimated by comparing the central cross-section value ($\mu_R = \mu_F = p_{\mathrm{T,jet}}$) with the ones for $\mu_R = \mu_F = p_{\mathrm{T,jet}}/2$ and $\mu_R = \mu_F = 2 \cdot p_{\mathrm{T,jet}}$.

- **PDF uncertainties:**
  A second source for uncertainties in Eq. 4.9, are originated in the different fits for the

PDFs. The estimated PDF variations are available within the LHAPDF package [95, 96]. Within this analysis the measurements are compared to NLO predictions using the CETQ6.6 PDF fits and the corresponding uncertainties. Following the proposal of the PDF4LHC working group [97] the impact of other global PDF fits [98–100] was investigated. Fig. 4.14 shows a comparison of the latest PDF4LHC with the CTEQ6.6 NLO calculations with the corresponding uncertainty bands. The largest differences of about $6\,\%$ between the central values of the CTEQ6.6 and the combined PDF4LHC cross-section predictions are found to be in the region for jets with a p_{T} of about 100-250 GeV and can be neglected considering the large experimental uncertainties of the measurement.

4.6.2 NP Correction Factors and Uncertainties

The measured quantities within collider experiments are usually corrected for detector effects to reach the particle level (see Chap. 1.4) and can then be compared to Monte Carlo simulated events using the theoretical calculations. Unfortunately, the next-to-leading order calculations of the inclusive jet cross-section, in contrast to other processes, are not available in the Monte-Carlo generator MC@NLO [90, 101]. To compare the measured distribution with theory predictions from the NLO calculations additional corrections for phenomenological non-perturbative effects [102] have to be applied to the NLO theory curves. To estimate these non-perturbative (NP) effects the to leading order Monte Carlo generators Pythia and Herwig++ with their different models for multiple parton interactions and hadronisation are taken into account and the NP correction is assumed to be the mean of these two calculations. As an estimate for the uncertainty associated with the NP corrections half of the difference between the predictions from Pythia and Herwig++ is used. For low jet p_{T} these differences can be up to $100\,\%$. The applied NP corrections factors to the NLO curve are shown in Fig. 4.12.

4.6.3 Summary

The NLO predictions had to be corrected for non-perturbative (NP) effects such as hadronisation and multiple parton interactions. These NP corrections were estimated by comparing parton-level predictions for the inclusive jet cross section with the corresponding prediction at particle level after harmonization and multiple parton interactions obtained by using Herwig++ and Pythia. The variation of the different PDFs in the CTEQ-6.6 set and the choice

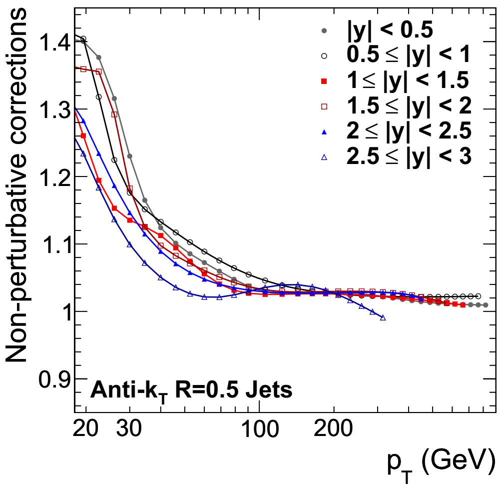

Figure 4.12: *Non-perturbative correction factors as applied on the NLO calculations for the anti-k_T jet algorithm. The factors were determined by using the mean of the individual corrections from Pythia and Herwig++. From [74].*

of renormalization and factorization scales are also accounted for in the uncertainty estimation. Fig. 4.13 gives summary of the derived systematic theoretical uncertainties. For low jet p_T (< 50 GeV) the large differences in the predictions of Pythia and Herwig++ result in a dominance of the NP uncertainties while the PDF uncertainties are dominant for high jet p_T, in the central region. In the forward region, the uncertainties of the NP dominate also at high jet p_T.

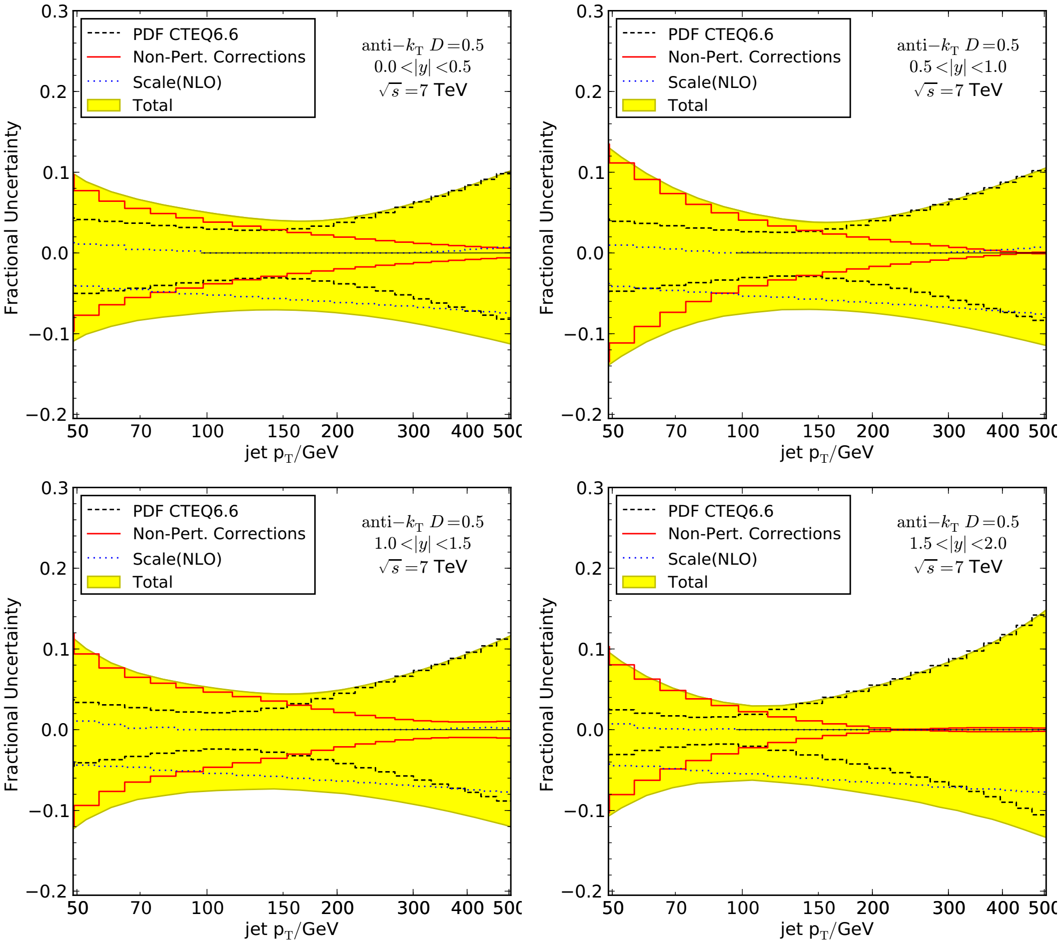

Figure 4.13: *Fractional theoretical uncertainties for the central four rapidity bins.*

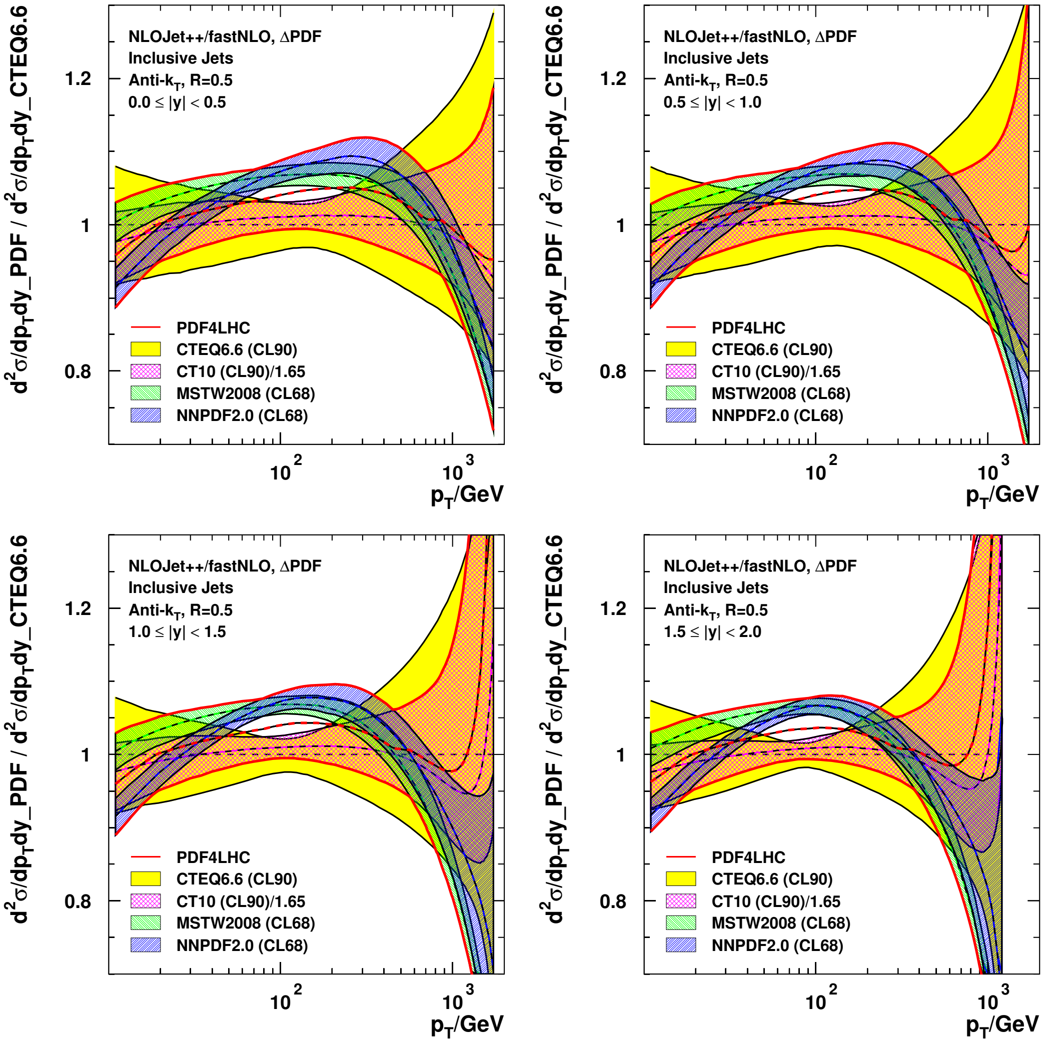

Figure 4.14: *Latest NLO predictions and fractional uncertainties for the central four rapidity bins as requested by PDF4LHC compared to the calculation using CTEQ6.6. The dashed red line depicts the combined PDF4LHC cross-section prediction and the red line encloses the 68 % confidence level band. The largest discrepancy of the central cross-section values of about* 6 % *are negligible compared to the experimental uncertainties in the measurement. Provided by [103].*

4.7 Final Comparison of Theory Calculations and Data

The measured and for JES and JER corrected cross-section spectra for jets clustered with the anti-k_T algorithm for a total integrated luminosity of $60\ \mathrm{nb}^{-1}$ are presented in Fig 4.15 together with the non-perturbative corrected NLO prediction curves. The yellow bands around the data points depict the systematic uncertainties due to experimental effects. For a detailed comparison, the ratios between the measured cross-section spectra and the theory curves in the different rapidity bins are shown in Fig. 4.16. The theoretical predictions agree with the results from the measurement to within about $20\,\%$ over most of the considered jet p_T and rapidity ranges. The fact, that the theory calculations systematically predict a higher double-differential jet cross-section than the measured one can be accounted to the imprecise experimental luminosity estimation and a therefore resulting bias within the normalisation of the spectra.

4.8 Conclusions

An initial measurement of the double-differential jet cross-section with early CMS data from proton-proton collisions at $7\ \mathrm{TeV}$ for an integrated luminosity of $60\ \mathrm{nb}^{-1}$ was presented. The used jets were reconstructed from calorimetric energy depositions with anti-k_T algorithm. With a sophisticated event and jet selection, non-collision events and noise could be reduced very efficient without loosing *real* physical jets. This first inclusive jet analysis already covered a jet transverse momentum range from $64\ \mathrm{GeV}$ up to $800\ \mathrm{GeV}$ in the central rapidity region. Within the uncertainties, the measurement shows a nice confirmation of QCD theory already at this early state of the LHC physics operation. The experimental uncertainties and the observed offset are expected to be reduced as soon as data driven JES estimations are available and the luminosity measurement will be improved with the growing understanding of the CMS Detector and the LHC beam attributes. This work contributed to the officially published results by the CMS collaboration (See appendix C.5).

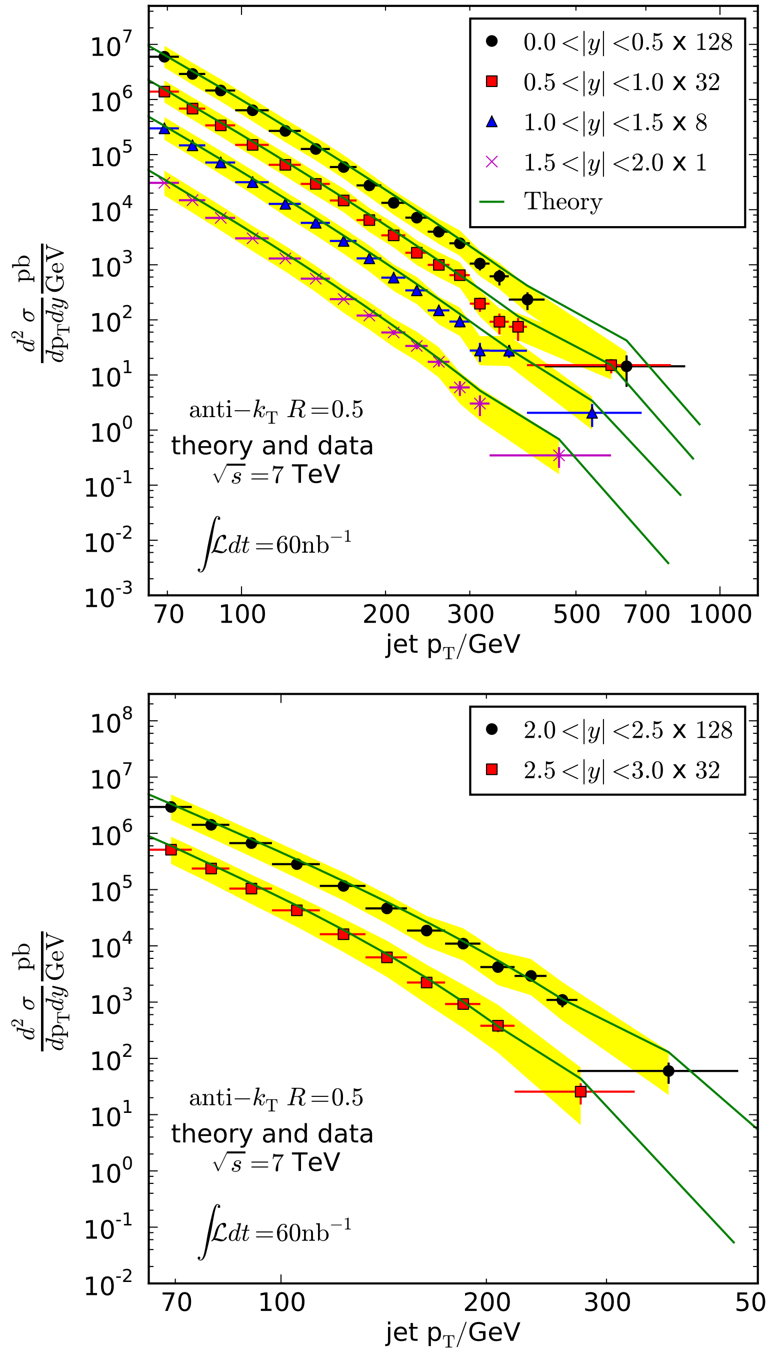

Figure 4.15: *Double-differential jet cross section for jets with the anti-k_T algorithm for $60\ nb^{-1}$ as predicted by the pQCD+NP theory (red line) and measured (dots with statistical errorbars). The yellow bands depict the systematic experimental uncertainties.*

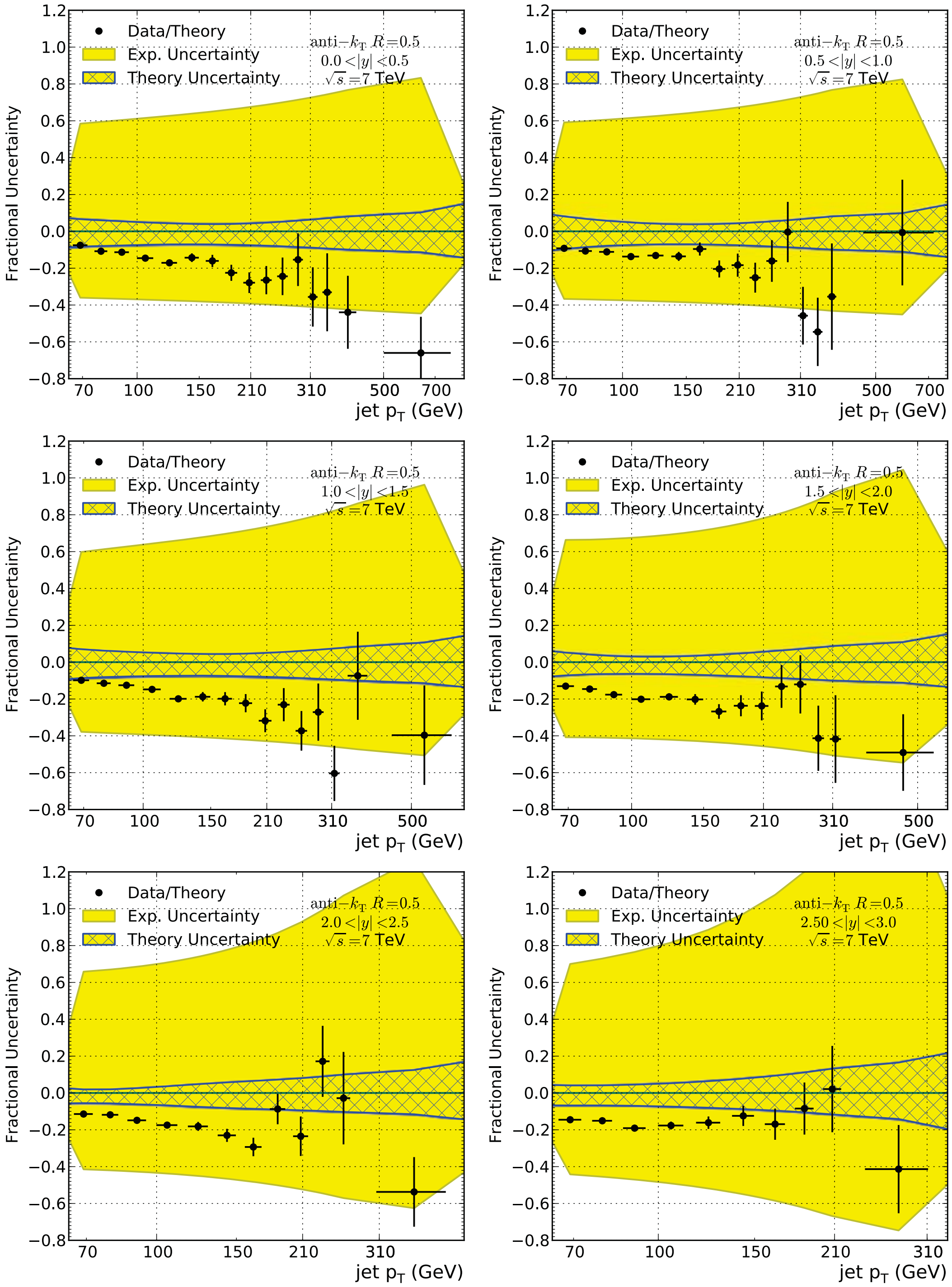

Figure 4.16: *Comparison of theory and measurement in ratio plots. Within the investigated phase space measurement and theory predictions agree within about* 20 % *of the cross-section value, which is within the range of the statistic and systematic uncertainties.*

Conclusions

With the beginning of the LHC operation at a centre-of-mass energy of 7 TeV in March 2010, CERN entered a new phase in the exploration of the fundamental forces and particles described by the Standard Model of Particle Physics and the search for new physics. At the LHC, processes of quantum chromodynamics have by far the largest cross-section. Therefore, the high production rate of jets is one of the first accessible observable to test and improve the commissioning of the detector and to make first comparisons with theoretical predictions and measured quantities.

To analyse the LHC data, a vast amount of computing resources is required to provide a reliable and powerful analysis environment to the LHC collaborations. To cope with these challenges the usage of the latest available technologies and even the development of new computing techniques is inevitable. The current LHC computing concept covers the data distribution and reprocessing requirements by employing grid technologies. However, the usage of platform virtualization and cloud computing would enable a transition to provide flexible computing resources independent of the underlying infrastructure at the different LHC computing centres. In this regard, providing an encapsulated and a user defined environment on a shared computing cluster infrastructure is a key feature for current High Performance Computing resource providers.
The usage of virtualization techniques to dynamically partition a cluster into independent sub-clusters was implemented within this work. The developed software, called ViBatch, uses standard tools, which are provided by most of the current cluster batch systems. By using these standard tools in a refined setup, the analysis jobs of various user groups are encapsulated within virtual machines which provide the desired computing environment. Future plans involve the usage of public cloud resources via cloudburst techniques which will en-

able the combination of grid and cloud techniques. Such a scenario decouples the complete computing infrastructure from hard and software constraints resulting in a dynamic High Performance Computing environment almost independent of local resources.

In this work, a pioneering measurement of the inclusive jet cross-section - double-differential in jet transverse momentum and rapidity - with first LHC data for collisions with a centre-of-mass energy of 7 TeV and a comparison with perturbative QCD calculations was presented. A central aspect of such an analysis using early data of a collider experiment in its commissioning phase, is a detailed study of the conditions at the time of data taking. With an integrated luminosity of $60\,\mathrm{nb}^{-1}$ collected by June 2010 with the CMS detector, the measured inclusive jet cross-section covers a transverse momentum range of jets up to 0.7 TeV which is already comparable to the reach of the Tevatron.
Taking the theoretical and experimental uncertainties into account, it is observed that already at this early state, the used predictions from theory describe the measured inclusive jet cross-section well within the investigated phase space. With increasing statistical precision and an accurate estimation of the jet energy scale and jet energy resolution uncertainties through data driven methods, it will soon be possible to perform first fits of α_s, to constrain the parton density functions, and to observe or exclude possible deviations from the Standard Model expected in scenarios like the compositeness of quarks.

Appendix - Theoretical Details

A.1 Gell-Mann Matrices

The Gell-Mann matrices λ_a are one possible representation of the fundamental generators of the special unitary group SU(3):

$$\begin{aligned}
\lambda_1 &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda_2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \\
\lambda_4 &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \lambda_5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix}, \\
\lambda_6 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \lambda_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, \quad \lambda_8 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix} \frac{1}{\sqrt{3}}
\end{aligned} \tag{A.1}$$

The structure constants $f^{abc} = f_{abc}$ are defined by the commutator relations

$$\left[T^a, T^b\right] = i f^{abc} T^c \tag{A.2}$$

They are totally antisymmetric ($f_{bac} = f_{acb} = -f_{abc}$) within their indices and the non-zero values out of the total $8 \cdot 8 \cdot 8 = 512$ are

$$\begin{aligned}
f_{123} &= 1 \\
f_{458,678} &= \sqrt{3}/2 \\
f_{147,165,246,257,345,376} &= 1/2
\end{aligned} \tag{A.3}$$

Appendix - Computing

B.1 Batch Systems - Resource Managers and Schedulers for Computing Clusters

A typical batch system can be distinguished into two different parts, the resource manager and the scheduler.

B.1.1 Resource Manager

The resource manager controls the nodes and the jobs in a cluster. It provides the basic functionality to start, stop, hold, cancel and monitor the jobs. It also keeps track of the used resources by the jobs and enforces the compliance with the job parameters like runtime or memory consumption.

B.1.2 Scheduler

The scheduler communicates with the resources manager and provides an ordering of the jobs via several configurable aspects. This job queue is then passed back to the resource manager. The decision scheme for the job queuing might include the following parameters:

- Compute resource availability
- Job priority (calculated by user configurable algorithms)

- Number of jobs allowed for the specific user or the specific resources.

B.1.3 Pre and Post Execution Scripting

A standard function in batch systems, is the ability to prepare and cleanup the computing nodes before a job is executed with so-called prologue and epilogue scripts.

B.2 ViBatch - Configuration

The basic configuration of ViBatch is centralised in one file with the following appearance:

```
# ViBatch installation directory (has to be accessible from all WNs)
VIBATCH_INSTALL_PATH="/pfs/data/software/kit/bd00/virt/vibatch"

# ViBatch config file used on the WNs (must be accessible from all WNs)
VIBATCH_CONFIG_FILE=${VIBATCH_INSTALL_PATH}"/config/vibatch_setup.cfg"

# Directory to store ViBatch settings and configs on the WNs
# (e.g. libvirt xml configuration files)
# Could be same as globally available VIBATCH_INSTALL_PATH or
# some local storage space on the WNs
VIBATCH_CONFIG_PATH_WN="/scratch/virt/config/libvirt"

# Lockfile directory on the WNs (has to be accessible from all WNs).
# It is used to store a lock file which indicates that a VM has started successfully.
VIBATCH_LOCKFILE_PATH_WN=${VIBATCH_INSTALL_PATH}"/lock"

# Alive directory on the WNs (has to be accessible from all WNs).
# A VM creates a file there, as soon as it is ready to accept connections.
VIBATCH_ALIVE_PATH_WN=${VIBATCH_INSTALL_PATH}"/alive"

# The parent of the user’s home directory
VIBATCH_HOME_DIR="/home/ws"

# Directory where ViBatch generates the public ssh keys to access the VMs
VIBATCH_PUBKEY_PATH_WN=${VIBATCH_INSTALL_PATH}"/generation/pubkeys"

# Directory where ViBatch generates the user environment files
VIBATCH_USERENV_PATH_WN=${VIBATCH_INSTALL_PATH}"/generation/environments"

# DEPRECATED!
# Filename for the public keypair generated for user access
#VIBATCH_PUBKEY_FILENAME=vibatch_user.rsa

# The ssh key which is required for logging in to a VM.
# This key belongs to the root user, should be without password.
VIBATCH_VMKEY=${VIBATCH_INSTALL_PATH}"/config/keys/id_rsa_vibatch"

# Time a VM has to boot before it is regarded as failure
VIBATCH_TIMEOUT_VMBOOT="200"

##### CONFIG: DEBUG TOOLS ##################################################################

# A directory to store debug and status output of ViBatch.
# Can be turned on by un-commenting the line below.
# !Note: This feature is only for development purposes and not
# supported and recommended in a production setup of ViBatch!
VIBATCH_DEBUG_PATH_WN=${VIBATCH_INSTALL_PATH}"/status"

# The logging level for debugging. ( HIGH | MEDIUM | LOW)
VIBATCH_LOG_LEVEL=HIGH

##### CONFIG: VIBATCH CLUSTER AND VM CONFIG ################################################
```

```
# Directory which contains the source image template (has to be accessible from all WNs)
VIBATCH_IMAGE_PATH_SOURCE="/pfs/data/software/kit/bd00/virt/images_templates"

# Source image names (separated by spaces if more than one)
# The images need to be named in the following schema:
# <vm_queue_name>_template.raw
# For the <vm_queue_name> have a look at
# the variable VIBATCH_VIRT_QUEUES below.
VIBATCH_IMAGE_NAMES_SOURCE="VM_SLC5_WN_template.raw"

# Directory to contain the VM image templates on the WNs
VIBATCH_IMAGE_PATH_DEST="/scratch/virt/images"

# Workernodes where ViBatch should be installed (separated by spaces)
VIBATCH_WN="ic1n025 ic1n026 ic1n027 ic1n028"

# Batch server where ViBatch should be installed
VIBATCH_SERVER="ic-pbs"

# Number of VMs to deploy on each WN (a good guess
# is to take the number of cores).
VIBATCH_VM_SLOTS=8

# Maximum memory per VM
VIBATCH_VM_MAX_MEM=1900000

# Memory after initial VM start
VIBATCH_VM_MEM=1900000

# Number of CPU cores per VM
VIBATCH_VM_CORES=1

# Prefix for the generated VMs
VIBATCH_VM_SEED="ic-wn"

# Prefix for the generated MAC addresses
VIBATCH_VM_MAC_SEED="52:54:02"

# Prefix name for the workernodes (e.g. without a running number at the end)
VIBATCH_WN_SEED="ic1n"

# The XML template file
VIBATCH_XML_TMPL=${VIBATCH_INSTALL_PATH}"/config/templates/libvirt-wn-template.xml"

# ssh options
VIBATCH_SSH_OPTIONS="-o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no"

##### CONFIG: VIBATCH BATCH SYSTEM CONFIG ################################################


# Batch system to be used on the server (capital letters, TORQUE )
VIBATCH_BATCHSYSTEM="TORQUE"

# Path to batch system config
VIBATCH_BATCH_ENVIRONMENT="/var/spool/torque/pbs_environment"

# Path to the place where the batch systems stores the jobs
VIBATCH_BATCH_JOBS="/var/spool/torque/mom_priv/jobs"

# Path to the directory where links to the pro/epilogues scripts are kept
VIBATCH_BATCH_SCRIPTS="/var/spool/torque/mom_priv"

# Specify Virtual Queues here (separated by spaces)
VIBATCH_VIRT_QUEUES="VM_SLC5"

##### CONFIG: VIBATCH HYPERVISOR #########################################################

# Which hypervisor is used? Currently tested only with KVM, but should work as well
# with all other hypervisors compatible to libvirt
VIBATCH_HYPERVISOR="qemu"

# The location of the QEMU-IMG binary
VIBATCH_QEMU_IMG="/usr/local/bin/qemu-img"

# Location of service binary (distribution dependent)
VIBATCH_SYSTEM_SERVICE="/sbin/service"
```

```
# The libvirt virsh executable
VIBATCH_VIRSH_EXE="/usr/bin/virsh"

# The libvirt daemon
VIBATCH_LVD_EXE="/usr/local/sbin/libvirtd"


# The file where libvirtd stores its PID, this is essential to be
# able to recover from e.g. unclean shutdowns
VIBATCH_LVD_PID="/usr/local/var/run/libvirtd.pid"

# Arguments passed to libvirtd
# ! only needed when starting libvirtd by hand
VIBATCH_LVD_ARGS="-d --listen"
```

B.2.1 Libvirt Preparation

Libvirt uses XML files to configure the virtual machines. All service and worker node VMs are configured in that way. As an example the configuration of the central administration VM looks like the following:

```
<domain type=’kvm’>
  <name>ic-adm</name>
  <uuid>4dea2413-1d22-d3f3-2516-782e98a23fa0</uuid>
  <memory>2048000</memory>
  <currentMemory>2048000</currentMemory>
  <vcpu>1</vcpu>
  <os>
    <type arch=’x86_64’ machine=’pc-0.12’>hvm</type>
    <boot dev=’hd’/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <clock offset=’utc’/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <emulator>/usr/local/bin/qemu-system-x86_64</emulator>
    <disk type=’file’ device=’disk’>
      <source file=’/software/kit/bd00/virt/images/Deb500_64bit_ic-adm.raw’/>
      <target dev=’hda’ bus=’ide’/>
      <address type=’drive’ controller=’0’ bus=’0’ unit=’0’/>
    </disk>
    <controller type=’ide’ index=’0’>
      <address type=’pci’ domain=’0x0000’ bus=’0x00’ slot=’0x01’ function=’0x1’/>
    </controller>
    <interface type=’bridge’>
      <mac address=’52:54:00:12:00:01’/>
      <source bridge=’eth0’/>
      <script path=’/mnt/data/software/kit/bd00/scripts/kvmifup/kvmifup’/>
      <model type=’e1000’/>
      <address type=’pci’ domain=’0x0000’ bus=’0x00’ slot=’0x04’ function=’0x0’/>
    </interface>
    <interface type=’bridge’>
      <mac address=’SOMEMAC’/>
      <source bridge=’eth3’/>
      <script path=’/pfs/data/software/kit/bd00/scripts/kvmifup/kvmifup’/>
      <model type=’e1000’/>
      <address type=’pci’ domain=’0x0000’ bus=’0x00’ slot=’0x05’ function=’0x0’/>
    </interface>
    <input type=’mouse’ bus=’ps2’/>
    <graphics type=’vnc’ port=’-1’ autoport=’yes’ listen=’127.0.0.1’ keymap=’en-us’ passwd=’SOMEPASSWORD’/>
    <video>
      <model type=’cirrus’ vram=’9216’ heads=’1’/>
      <address type=’pci’ domain=’0x0000’ bus=’0x00’ slot=’0x02’ function=’0x0’/>
    </video>
```

```
  </devices>
</domain>
```

B.2.2 Maui/Torque Setup

Excerpt of the Torque setup to enable queues with virtual machines:

```
#
# Example for a native queue "short".
# All native queues need to have
# resources_default.neednodes set to "GENERIC"
#
create queue short
set queue short queue_type = Execution
set queue short Priority = 30
set queue short max_queuable = 3000
set queue short max_running = 130
set queue short resources_max.cput = 01:00:00
set queue short resources_max.walltime = 03:05:00
set queue short resources_default.nice = 0
set queue short acl_group_enable = True
set queue short acl_groups = bd00
set queue short max_user_run = 90
set queue short resources_default.neednodes = GENERIC
set queue short enabled = True
set queue short started = True

#
# Example to create and define a virtual queue "VM_DEB5"
# All virtualized queues need to have
# resources_default.neednodes set to "VMHOST"
#
create queue VM_DEB5
set queue VM_DEB5 queue_type = Execution
set queue VM_DEB5 Priority = 30
set queue VM_DEB5 max_queuable = 3000
set queue VM_DEB5 max_running = 32
set queue VM_DEB5 resources_max.cput = 01:00:00
```

```
set queue VM_DEB5 resources_max.walltime = 03:05:00
set queue VM_DEB5 resources_default.nice = 0
set queue VM_DEB5 acl_group_enable = True
set queue VM_DEB5 acl_groups = bd00
set queue VM_DEB5 max_user_run = 32
set queue VM_DEB5 resources_default.neednodes = VMHOST
set queue VM_DEB5 enabled = True
set queue VM_DEB5 started = True
```

B.2.3 ViBatch Prologue Script

The main functionality of ViBatch is encapsulated within the prologue and epilogue scripts presented here.

Prologue

```
#!/bin/bash
# This script is executed by the main prologue script if the
# job is submitted to a virtual queue.
# It needs to get the following values:
#
# argv[1]       job id
# argv[2]       job execution user name
# argv[3]       job execution group name
# argv[4]       job execution queue
# Map arguments to local script variables
JOBID=$1
USERN=$2
USERG=$3
QUEUE=$4


# Some cached variable
VB_WN_HOSTNAME=$(hostname)
VB_EXEC_TIME=$(date)

# Function to gather information about the job and the host it is running on
function vbfnJobinfo() {
vbfnLog 1 ""
vbfnLog 1 "[VIBATCH-PROLOGUE] Summary:"
vbfnLog 1 "Job and Host Info Summary:"
vbfnLog 1 "--------------------------"
vbfnLog 1 ""
vbfnLog 1 "   Execution time:           ${VB_EXEC_TIME}"
vbfnLog 1 ""
        vbfnLog 1 "   Running on host:          ${VB_WN_HOSTNAME}"
vbfnLog 1 "   Kernel version of host:   $(uname -a)"
vbfnLog 1 "   Uptime of host:           $(uptime)"
vbfnLog 1 ""
vbfnLog 1 "   ViBatch Install Path:     ${VIBATCH_INSTALL_PATH}"
vbfnLog 1 "   ViBatch Config File:      ${VIBATCH_CONFIG_FILE}"
vbfnLog 1 "   ViBatch Local WN Path:    ${VIBATCH_CONFIG_PATH_WN}"
vbfnLog 1 ""
        vbfnLog 1 "   ViBatch JobID:            ${JOBID}"
vbfnLog 1 "   Job Execution Username:   ${USERN}"
vbfnLog 1 "   Job Execution Groupname:  ${USERG}"
        vbfnLog 1 "   Queue:                    ${QUEUE}"
        vbfnLog 1 "   Executed Script:          $0"
vbfnLog 1 ""
vbfnLog 1 "   Return Code:              $1"
```

```
vbfnLog 1 "   Return Message:           $2"
}

# Function to display error output if something went wrong
# Sets the exit code of the script at the end
# !! maybe replace/expand this with a more general function in functions.sh
function vbfnExit() {
    if [ $1 -ne 0 ]; then
vbfnLog 1 ""
vbfnLog 1 "[VIBATCH-PROLOGUE] Exit Code Information:"
vbfnLog 1 "The ViBatch script has exited with a non-zero return code:"
vbfnLog 1 "----------------------------------------------------------"
vbfnLog 1 ""
vbfnLog 1 "   Script Exit Code: $1"
vbfnLog 1 ""
vbfnLog 1 "Error Code Description:"
vbfnLog 1 "-----------------------"
vbfnLog 1 "   10 Script not called with proper arguments or config error"
vbfnLog 1 "   11 Directory for the lock files not valid"
vbfnLog 1 "   12 Directory for templates not valid"
vbfnLog 1 "   13 Template file not found"
vbfnLog 1 "   14 VM ssh key not found"
vbfnLog 1 "   15 Virsh executable not found"
vbfnLog 1 "   20 Could not source function file"
vbfnLog 1 "   21 Could not source hooks file"
vbfnLog 1 "   22 Could not source nodes file"
vbfnLog 1 "   30 Could not connect to libvirt daemon"
vbfnLog 1 "   31 Could not create lockfile"
vbfnLog 1 "   32 Could not create overlay image from template"
vbfnLog 1 "   33 libvirt daemon was not able to start VM"
vbfnLog 1 "   34 VM did not come up to accept connections within time"
vbfnLog 1 "   35 Could not prepare VM"
vbfnLog 1 "   40 No free VM slot on worker node"
vbfnJobinfo $1 "$2"

        vbfnLog 2 "EXIT PROLOGUE: $1 $2"

exit $1
    else
        vbfnJobinfo $1 "No problems detected"
        exit $1
    fi
}


# Include the main ViBatch function file
FILE_FUNCTIONS=$(dirname $0)/../lib/functions.sh
source ${FILE_FUNCTIONS}
if [ "$?" -ne "0" ]; then vbfnExit 20 "Could not source functions file: ${FILE_FUNCTIONS}"; fi

# Check if all arguments are passed to script
if [ ! "${JOBID}" -o ! "${USERN}" -o ! "${USERG}" -o ! "${QUEUE}" ]; then
vbfnExit 10 "Not all required commandline script arguments given"
fi

# Include the ViBatch hook file; it can be used to define
# specific tasks which are executed after a specific action
# was triggered within ViBatch scripts
FILE_HOOKS=$(dirname $0)/../config/vibatch_hooks.cfg
source ${FILE_HOOKS}
if [ "$?" -ne "0" ]; then vbfnExit 21 "Could not source hooks file: ${FILE_HOOKS}"; fi

# Include the ViBatch node configuration file, this
# is used to set up individual settings depending on
# the hostname on which it is executed.
FILE_NODES=$(dirname $0)/../config/vibatch_nodes.cfg
source ${FILE_NODES}
if [ "$?" -ne "0" ]; then vbfnExit 22 "Could not source nodes file: ${FILE_NODES}"; fi

# Execute global and hostspecific (if available) setup, defined in the
# ${FILE_NODES} configuration file
vbfnCall vbfnGlobal
vbfnCall vbfnLocal ${VB_WN_HOSTNAME}


if [ -n ${DEBUGLOGDIR} ] ; then
    export VIBATCH_DEBUGLOGFILE=${DEBUGLOGDIR}/${JOBID}.log
    touch ${VIBATCH_DEBUGLOGFILE}
```

```
    chmod 766 ${VIBATCH_DEBUGLOGFILE}
    vbfnLog 1 "[VIBATCH-PROLOGUE] Created logfile ${VIBATCH_DEBUGLOGFILE}"
fi


# Let’s write some debug statements, as well to the debug directory if enabled
vbfnLog 1 "[VIBATCH-PROLOGUE] Starting ViBatch job [${JOBID}] on WN [${HOSTNAME}]"



# Check if some required variables are set correctly
[[ -f ${VIRSH_EXE} ]] || ( vbfnExit 15 "Could not find virsh executable: ${VIRSH_EXE}" )
[[ -d ${LOCKDIR} ]] || ( vbfnExit 11 "Lockfile directory does not exist: ${LOCKDIR}" )
[[ -d ${TEMPLATEDIR} ]] || ( vbfnExit 12 "Could not find template directory: ${TEMPLATEDIR}" )
[[ -f ${TEMPLATEDIR}/${QUEUE}_WN_template.raw ]] || ( vbfnExit 13 "Could not find template file: \\
${TEMPLATEDIR}/${QUEUE}_WN_template.raw" )
[[ -f ${VMKEY} ]] || ( vbfnExit 14 "Root SSH VM key for authentication not found: ${VMKEY}" )

# Check if a VM is free, should be the case since otherwise
# the batch system should not have sent the job on that hostname
vbfnLog 3 "[VIBATCH-PROLOGUE] Check if a VM slot is free."

THE_VM=""
vbfnGetFreeVM
if [ "$?" -ne "0" ]; then vbfnExit 40 "No free VM slot found on WN, please check your batch system configuration!"; fi
vbfnLog 1 "[VIBATCH-PROLOGUE] [$JOBID] selected free VM: [${THE_VM}]"

# Lets lock the machine, that no other jobs besides this one gets submitted to it
CLEANUP_MODE=NORMAL
vbfnLockVM ${THE_VM} ${JOBID}
if [ "$?" -ne "0" ]; then vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}; vbfnExit 31  \\
"Could not create lock files in directory: [${LOCKDIR}]"; fi
vbfnLog 1 "[VIBATCH-PROLOGUE] Lock files created for VM [${THE_VM}] and job [${JOBID}]"

# Just to be sure machine is really dead
CLEANUP_MODE=KEEPLOCK
vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}

# Create a VM image to be usable
vbfnCreateImage ${THE_VM} ${QUEUE}
if [[ $? -ne 0 ]]; then
vbfnLog 1 "[VIBATCH-PROLOGUE] Error while creating overlay image from template."
CLEANUP_MODE=NORMAL
vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}
vbfnExit 32 "[VIBATCH-PROLOGUE] Could not create overlay image: [${THE_VM}] on host [${VB_WN_HOSTNAME}] for job [${JOBID}]"
else
vbfnLog 1 "[VIBATCH-PROLOGUE] Overlay image sucessfully created."
fi

# Check if LVD is alive and accepting connections.
# If not, lets try to restart it.
vbfnIsLVDAlive
LVDAlive=$?
vbfnLog 3 "[VIBATCH-PROLOGUE] isLVDAlive delivered $LVDAlive (1 = true, 0 = false)"
counter=0
while [ "${LVDAlive}" -ne 1 -a "${counter}" -le 5 ] ; do
vbfnLog 1 "[VIBATCH-PROLOGUE] Restarting libvirtd. Try number (${counter}/5)..."
(( counter ++ ))
vbfnRestartLVD
vbfnIsLVDAlive; LDVAlive=$?
done


if [ "${LVDAlive}" -ne 1 ] ; then
vbfnLog 1 "[VIBATCH-PROLOGUE] Libvirt daemon not accepting connections. Calling hook."

vbfnCall global_LVDdown ${VB_WN_HOSTNAME} ${THE_VM} ${JOBID} # let’s vbfnCall a hook
CLEANUP_MODE=NORMAL
vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}
vbfnExit 30 "[VIBATCH-PROLOGUE] Libvirt daemon down: [${VM}] on host [${VB_WN_HOSTNAME}]
for job [${JOBID}]" "$0" "$VB_WN_HOSTNAME" "$VM" "$JOBID"
else
vbfnLog 1 "[VIBATCH-PROLOGUE] Ok, libvirtd is up and responding. Time to bring virtual machine up."
fi

# Ok, libvirt is up and running. Now start THE_VM
${VIRSH_EXE} start ${THE_VM} 2>>${VIBATCH_DEBUGLOGFILE}
if [ "$?" -ne 0 ]; then
```

```
vbfnLog 1 "[VIBATCH-PROLOGUE] libvirt reported an error while starting ${THE_VM}."
CLEANUP_MODE=NORMAL
vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}
vbfnExit 33 "[VIBATCH-PROLOGUE] libvirtd could not start the virtual machine:
    [${THE_VM}] on host [${VB_WN_HOSTNAME}] for job [${JOBID}]" "$0" "$VB_WN_HOSTNAME" "$THE_VM" "$JOBID"
else
vbfnLog 1 "[VIBATCH-PROLOGUE] libvirt reported machine is running."
vbfnLog 2 "[$JOBID] started [$THE_VM]"
fi


# The virtual machine  will need a while to be up. Let's wait for it
vbfnWaitForVM $THE_VM $VMTIMEOUT
waitRet=$?
if [ ${waitRet} -ne 0 ]; then
        if [ ${waitRet} -eq 1 ]; then
    vbfnLog 1 "[VIBATCH-PROLOGUE] VM [${THE_VM}] timed out. Cancel prologue script."
    vbfnCall global_THE_VM_timeout $VB_WN_HOSTNAME $THE_VM $JOBID $TIMEOUT
    CLEANUP_MODE=NORMAL
    vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}
    vbfnExit 34 "[VIBATCH-PROLOGUE] VM timeout: [${THE_VM}] on host [${VB_WN_HOSTNAME}]
for job [${JOBID}]" "$0" "$VB_WN_HOSTNAME" "$THE_VM" "$JOBID"
elif [ ${waitRet} -eq 2 ]; then
    vbfnLog 1 "[VIBATCH-PROLOGUE] VM [${THE_VM}] could not be started. Cancel prologue script."
    vbfnCall global_THE_VM_timeout $VB_WN_HOSTNAME $THE_VM $JOBID $TIMEOUT
    CLEANUP_MODE=NORMAL
    vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}
    vbfnExit 34 "[VIBATCH-PROLOGUE] VM [${THE_VM}] could not be started because the lockfile has been deleted!"
fi
else
vbfnLog 1 "[VIBATCH-PROLOGUE] Machine is ready to accept connections."
fi


# Copy the job file to our virtual machine
vbfnPipeJobToVM ${THE_VM} ${JOBID}
if [ $? -ne 0 ]; then
vbfnLog 1 "[VIBATCH-PROLOGUE] Could not copy job to THE_VM."
#CLEANUP_MODE=NORMAL
#vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}
#vbfnExit 35 "[VIBATCH-PROLOGUE] Could not prepare THE_VM: [${THE_VM}] on host [${VB_WN_HOSTNAME}] for job [${JOBID}]"
else
vbfnLog 1 "[VIBATCH-PROLOGUE] Virtual Machine ${THE_VM} prepared. Job script copied sucessfully."
fi

# Create a user's account on the VM
vbfnPrepareUserOnVM ${THE_VM} ${USERN} ${USERG} ${JOBID}
if [ $? -ne 0 ]; then
vbfnLog 1 "[VIBATCH-PROLOGUE] Could not create user on the VM."
CLEANUP_MODE=NORMAL
vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}
vbfnExit 35 "[VIBATCH-PROLOGUE] Could not prepare User on VM: [$USERN]:[$USERG] on \
 VM [${THE_VM}] on host [${VB_WN_HOSTNAME}] \
 for job [${JOBID}]" "$0" "$VB_WN_HOSTNAME" "$THE_VM" "$JOBID"
else
vbfnLog 1 "[VIBATCH-PROLOGUE] Access granted for user [$USERN]:[$USERG] on VM [$THE_VM]\
   for JOBID [${JOBID}] on Host [$VB_WN_HOSTNAME]."
fi


# All done. Leaving prologue script.
vbfnLog 1 "[VIBATCH-PROLOGUE] Now exitting PROLOGUE."
vbfnLog 1 "[VIBATCH-PROLOGUE] PROLOGUE ERROR CODE: 0"
vbfnCall global_prologue_win ${VB_WN_HOSTNAME} ${THE_VM} ${JOBID}
vbfnExit 0 "[VIBATCH-PROLOGUE] Prologue exits: [${THE_VM}] on host [${VB_WN_HOSTNAME}]\
            for job [${JOBID}]" "$0" "$VB_WN_HOSTNAME" "$THE_VM" "$JOBID"
```

Epilogue

```
#!/bin/bash
# This script is executed by the main epilogue script.
# It needs to get the following values:
#
# argv[1]       job id
# argv[2]       job execution user name
```

```
# argv[3]       job execution group name
# argv[4]       job execution queue


# Map arguments to local script variables, but USERN and USERG are not needed
JOBID=$1
USERN=$2
USERG=$3
QUEUE=$4


# Function to gather information about the job and the host it is running on
function vbfnJobinfo() {
vbfnLog 1 ""
vbfnLog 1 "[VIBATCH-EPILOGUE] Summary:"
vbfnLog 1 "Job and Host Info Summary:"
vbfnLog 1 "-------------------------"
vbfnLog 1 ""
vbfnLog 1 "   Execution time:           $(date)"
vbfnLog 1 ""
        vbfnLog 1 "   Running on host:          $(hostname)"
vbfnLog 1 "   Kernel version of host:   $(uname -a)"
vbfnLog 1 "   Uptime of host:           $(uptime)"
vbfnLog 1 ""
vbfnLog 1 "   ViBatch Install Path:     ${VIBATCH_INSTALL_PATH}"
vbfnLog 1 "   ViBatch Config File:      ${VIBATCH_CONFIG_FILE}"
vbfnLog 1 "   ViBatch Local WN Path:    ${VIBATCH_CONFIG_PATH_WN}"
vbfnLog 1 ""
        vbfnLog 1 "   ViBatch JobID:            ${JOBID}"
vbfnLog 1 "   Job Execution Username:   ${USERN}"
vbfnLog 1 "   Job Execution Groupname:  ${USERG}"
        vbfnLog 1 "   Queue:                    ${QUEUE}"
        vbfnLog 1 "   Executed Script:          $0"
vbfnLog 1 ""
vbfnLog 1 "   Return Code:              $1"
vbfnLog 1 "   Return Message:           $2"
}



# Function to display error output if something went wrong
# Sets the exit code of the script at the end
# Error Codes: 1? = Variables / Environment ; 2? config error; 3? other
# Caution: The corresponding function with the same name exists already in the epilogue script
# !! maybe replace/expand this with a more general function in functions.sh
function vbfnExit() {
    if [ $1 -ne 0 ]; then
vbfnLog 1 ""
vbfnLog 1 "[VIBATCH-EPILOGUE] Exit Code Information:"
vbfnLog 1 "The ViBatch script has exited with a non-zero return code:"
vbfnLog 1 "----------------------------------------------------------"
vbfnLog 1 ""
vbfnLog 1 "   Script Exit Code: $1"
vbfnLog 1 ""
vbfnLog 1 "Error Code Description:"
vbfnLog 1 "----------------------"
vbfnLog 1 "   10 Script not called with proper arguments or config error"
vbfnLog 1 "   11 Could not find Virtual Machine related to JobId"
vbfnLog 1 "   20 Could not source function file"
vbfnLog 1 "   21 Could not source hooks file"
vbfnLog 1 "   22 Could not source nodes file"
vbfnJobinfo $1 "$2"

        vbfnLog 2 "EXIT EPILOGUE $1 $2"
exit $1
    else
        vbfnJobinfo $1 "No problems detected"
        exit $1
    fi
}


# Include the main ViBatch function file
FILE_FUNCTIONS=$(dirname $0)/../lib/functions.sh
source ${FILE_FUNCTIONS}
if [ "$?" -ne "0" ]; then vbfnExit 20 "Could not source functions file: ${FILE_FUNCTIONS}"; fi

# Cached Variables
```

```
VB_WN_HOSTNAME=$(hostname)
VB_EXEC_TIME=$(date)

# Check if all arguments are passed to script
if [ ! "${JOBID}" -o ! "${QUEUE}" ]; then
vbfnExit 10 "Not all required commandline script arguments given"
fi


# Include the ViBatch hook file; it can be used to define
# specific tasks which are executed after a specific action
# was triggered within ViBatch scripts
FILE_HOOKS=$(dirname $0)/../config/vibatch_hooks.cfg
source ${FILE_HOOKS}
if [ "$?" -ne "0" ]; then vbfnExit 21 "Could not source hooks file: ${FILE_HOOKS}"; fi

# Include the ViBatch node configuration file, this
# is used to set up individual settings depending on
# the hostname on which it is executed.
FILE_NODES=$(dirname $0)/../config/vibatch_nodes.cfg
source ${FILE_NODES}
if [ "$?" -ne "0" ]; then vbfnExit 22 "Could not source nodes file: ${FILE_NODES}"; fi

# Execute global and hostspecific (if available) setup e.g. several variables like LOCKDIR, defined in the
# ${FILE_NODES} configuration file
vbfnCall vbfnGlobal
vbfnCall vbfnLocal ${VB_WN_HOSTNAME}



[[ -f ${VIRSH_EXE} ]] || ( vbfnExit 15 "Could not find virsh executable: ${VIRSH_EXE}" )
[[ -d ${LOCKDIR} ]] || ( vbfnExit 11 "Lock directory does not exist: ${LOCKDIR}" )


if [ -n ${DEBUGLOGDIR} ] ; then
    export VIBATCH_DEBUGLOGFILE=${DEBUGLOGDIR}/${JOBID}.log
fi

THE_VM=""
vbfnGetVMByJob ${JOBID}
if [ $? -ne 0 ] || [ -z ${THE_VM} ]; then
vbfnLog 1 "[VIBATCH-EPILOGUE] Can not findout what VM is used for job [${JOBID}] on host [$VB_WN_HOSTNAME]. Aborting."
vbfnCall global_no_VM_found $VB_WN_HOSTNAME $JOBID "$VMLIST"
CLEANUP_MODE=BYJOBID
vbfnCleanup ${CLEANUP_MODE} "NO_VM" ${JOBID}
vbfnExit 11 "Could not find VM for job [${JOBID}]"
else
vbfnLog 1 "[VIBATCH-EPILOGUE] VM [$THE_VM] used by job [${JOBID}]."
fi


# Cleanup residuals and artefacts of the Virtual Machine
CLEANUP_MODE=NORMAL
vbfnCleanup ${CLEANUP_MODE} ${THE_VM} ${JOBID}


# All done. Leaving epilogue script.
vbfnLog 1 "[VIBATCH-EPILOGUE] Now exitting EPILOGUE..."
vbfnLog 1 "[VIBATCH-EPILOGUE] EPILOGUE ERROR CODE: 0"
vbfnCall global_epilogue_win $VB_WN_HOSTNAME $VM $JOBID # Oh a hook would be cool...
vbfnExit 0 "[VIBATCH-EPILOGUE] Epilogue exits: [${THE_VM}] on host [${VB_WN_HOSTNAME}]
for machine [${VM}] for job [${JOBID}]." "$0" "$VB_WN_HOSTNAME" "$VM" "$JOBID"
```

B.3 IC1 - Institutes Cluster at the SCC

Properties of the IC1, the testbed for ViBatch:

- Shared between nine different KIT department

- Two user group partitions:
  - HPC partition: parallel computing (MPI)
  - HEP Partition: Serial computing, High Throughput Computing (HTC)
- Technical Details:
  - 200 compute nodes with 2 x Intel Quadcore Xeon yielding a 17.5 TFlop peak performance
  - Lustre cluster file system 420 TB in total
  - OS: Suse Linux Enterprise 11.0Sp2

B.4 Grid Services

To enable users to access the data and the computing resources distributed by the computing centres participating in the WLCG, special software services have been developed and are steadily improve d. Several publicly funded projects were founded in the past to meet this challenge. The main aim of these projects was to build a so called ”middleware” system. The middleware should grant the users the access to the grid resources as an abstracted layer, enabling the users to work on the grid without the need to know and control all underlying services. There are some special requirements for such a grid environment. The most important are security concerns, heterogeneous hardware, and different software at the Tier centres. To reduce complications to a minimum, the Tier centres have to deploy the Grid services on Scientific Linux [51] machines. For grid security, three major tasks have to be accomplished: The Grid user should have access to all grid sites and resources needed, but the large number of computing and storage nodes with their broadband internet connection must not be open to abuse. At present, the existing WLCG software package is merged with gLite, the EGEE (Enabling Grids for E-sciencE) [104] middleware. The Grid services provided by it are described in the following.

Grid Wide Services

Grid Wide Services are deployed for the whole WLCG, and are therefore not installed on each site.

- **VO Server/ VOMS Server:**
  The VO/VOMS server is the registry office of a Virtual Organisation. On this LDAP[1] server, a list of all users belonging to a VO and their role and permissions are stored. The VO server only distinguishes two kinds of members: normal grid users and the software managers. This structure of user permissions is changed now and the successor of the VO server, the VOMS server, is able to handle a more precise management of user permissions and allows users memberships in different VOs. With this server, the user is able to select its role and VO when initialising the grid-proxy, a temporary electronic passport for the user.

- **Replica Location Service:**
  When a file is copied to the grid, the Replica Location Service (RLS) gives it a unique identifier, the GUID. All replicas of that file have this same identifier. There is also the possibility to s et Logical File Names (LFN) as synonyms for the GUID since this long identification string consists of arbitrary numbers and characters and is therefore not easy to remember. The RLS exists only once per VO and maps the logical file names to the physical file names via the GUID. Therefore it is one of the most critical services for data management in a Virtual Organisation.

- **Information Systems:**
  To manage the information on available resources and users on the grid, a multi-layer information service is established:

  - Grid Resource Information Service (GRIS):
    This service runs locally on the portal machines to the grid of each site. It publishes site specific information via the GLUE schema[2], e. g. the installed software or the number of free CPUs.
  - Grid Information Index Server (GIIS):
    This server collects the information of the local GRIS via LDAP.
  - Berkeley Database Information Index (BDII):
    The BDII is the central collection point for all information. It gathers information from all site GIIS via LDAP.

- **Resource Broker:**
  The Resource Broker (RB) is the intelligence of the grid, as it uses information from the Information System to distribute incoming job requests in an efficient manner on

---

[1] Lightweight Directory Access Protocol

[2] The Grid Laboratory for a Uniform Environment (GLUE) is a schema for information description

accessible resources.
After having received the job request of a user, the Resource Broker scans the request for its requirements. In a next step, it is searching through the Information System for sites that offer the resources the user demands for. Having found a matching site, the RB's Job Submission Service (JSS) will submit the job and all files the job needs to the corresponding site. Now, the Logging and Bookkeeping Service (LB) controls the status of the job. This information is also accessible for the user. When the job is finished, its output is transferred to the Resource Broker and stored there until the user retrieves it. A RB is able to accommodate multiple Virtual Organisations. Also a VO can have access to multiple Resource Brokers.

Site Wide Services

- **User Interface:**
  The User Interface (UI) is the user's access point to the grid for the user. Several programs are available on it, e. g. for submitting a job to the grid, getting information on the job status or retrieving the output of the job. The User Interface is also the portal for file access. The user can copy and register files as well as replicate existing files to other destinations. The authentication of the user is realised by the grid-proxy which has to be initialised before working on the grid.

- **Computing Element and Worker Node:**
  The Computing Element (CE) is the portal to the local batch system of a site. It is not - as the name suggests - the place where jobs are computed; these are the Worker Nodes (WN) of the batch system. The main reason to put this component in between the user and the worker node is simple: The CE offers a layer of abstraction, i. e. the user does not have to deal with different batch systems on different sites.

  When a job is submitted to the CE, the middleware software checks within the grid-mapfile[3], whether the user, authenticated by the grid-proxy, is allowed to run that job on this site or not. Furthermore, this file contains the information on the local account, to wich the user has to be mapped. At present there are three possibilities for this mapping:

  - The user does not have a local account (standard):
    The job is mapped on a mapping account. These are numbered accounts of the type VO-Name plus a three digits number, e. g. cms001.

---

[3] /etc/grid-security/grid-mapfile

  - The user does have a local account:
    It is the local site manager who decides whether the job is mapped to the user's local account or a mapping account.
  - The user is the software manager of the VO:
    On each site, a specials software directory is accessible via an environmental variable[4]. The VO software manager, who is mapped on an account with special privileges, is authorised to install software in directories where other members of the VO only read access.

  In the current release of WLCG, an improved management of permissions in the form of VOMS is available.

  After receiving the job, the CE submits it via the local batch system to the worker node, where the job actually runs. The Computing Element copies the input data for the job (Input Sandbox) to a special directory on the worker node, which is a subdirectory of the login directory. The worker node has access to this directory as well as to files, stored on the grid and the web. To access these files, several protocols are installed on the worker nodes, e. g. gridftp and wget. After the job is finished, the CE sends the job output (Output Sandbox) back to the Resource Broker.

- **Storage Element:**
  Like the Computing Element, the Storage Element is only a gateway to the local storage system. Since it is a Globus Gridftp service, it supports by default the gsiftp protocol. Other protocols like srm can be added by the local site manager. Since LCG version 2.4.0, also dCache [105] is provided. At smaller sites, only disk space is available, whereas larger sites like GridKA offer disk space and tape archival.

---

[4] $VO_CMS_SW_DIR for users of the CMS VO

Appendix - Experimental Details

C.1 Datasets

Run	est. Lum. (μb^{-1})	number of Events	Dataset name
135149	1278	23579429	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135445	1239	26964280	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135521	581	2013140	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135523	339	1138445	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135525	699	2376332	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135528	2024	6938279	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135534	102	353289	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135535	288	1002560	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135575	1403	4917330	/MinimumBias/Commissioning10-PromptReco-v9/RECO
135735	272	956941	/MinimumBias/Commissioning10-PromptReco-v9/RECO
subtotal	8225		
136033	837	150277	/JetMETTau/Run2010A-PromptReco-v1/RECO
136035	171	30986	/JetMETTau/Run2010A-PromptReco-v1/RECO
136066	1053	190031	/JetMETTau/Run2010A-PromptReco-v1/RECO
136082	717	126319	/JetMETTau/Run2010A-PromptReco-v1/RECO
136088	937	55112	/JetMETTau/Run2010A-PromptReco-v2/RECO
136100	3861	148970	/JetMETTau/Run2010A-PromptReco-v2/RECO
subtotal	7576		
total	60.30 nb^{-1}		

Table C.1: *7TeV Datasets analyzed with estimated luminosity and recorded number of events.*

C.2 Run and Luminosity Sections

Certified runs and luminosity sections for physics analyses are distributed via files using the JavaScript Object Notation (JSON) format. The numbers in quotes represent the run number followed by the valid luminosity section ranges. The following JSON files were used in the analysis:

- June, 14th Re-Reco JSON

```
{
"135149": [[297, 337], [339, 754], [756, 932], [934, 937], [942, 993], [995, 1031], [1033, 1098],
 [1102, 1808], [1811, 2269], [2274, 2524], [2528, 2713], [2715, 3098], [3100, 3102], [3105, 3179],
 [3182, 3303], [3305, 3381]],
"135175": [[55, 545], [548, 561], [563, 790], [792, 1046]],
"135521": [[60, 108], [110, 359], [361, 440], [442, 488]],
"135523": [[1, 64], [66, 109], [113, 124], [126, 211]],
"135525": [[1, 3], [6, 143], [145, 275], [293, 381], [384, 435], [437, 452]],
"135528": [[1, 91], [94, 95], [98, 142], [145, 147], [149, 308], [310, 454], [456, 606], [608, 609],
[611, 770], [773, 776], [779, 813], [816, 912], [915, 924], [926, 1082], [1084, 1213], [1215, 1436]],
"135535": [[75, 167], [169, 232]],
"135575": [[2, 210], [213, 241], [243, 264], [266, 381], [384, 638], [645, 1161], [1163, 1253]],
"135735": [[31, 42], [44, 149], [151, 234], [236, 320]],
"136066": [[181, 297], [299, 319], [321, 321], [323, 336], [338, 348], [350, 366], [368, 529], [532, 544],
[546, 595], [597, 719], [721, 1004], [1006, 1184]],
"136082": [[1, 173], [175, 422], [477, 504], [506, 506]],
"136100": [[1, 94]],
"136119": [[1, 36]],
"137027": [[98, 146], [149, 162], [165, 187]],
"137028": [[1, 107], [111, 157], [163, 169], [171, 180], [184, 200], [203, 209], [211, 232], [234, 274],
[276, 338], [341, 475], [478, 484]]
}
```

- PromtReco JSON:

```
{
"138571": [[1, 13]],
"138572": [[1, 213]],
"138737": [[1, 66]],
"138738": [[1, 10]],
"138742": [[1, 20], [22, 38]],
"138744": [[1, 25]],
"138745": [[1, 10]],
"138746": [[1, 130]],
"138747": [[1, 71], [73, 131]],
"138750": [[1, 46], [49, 208], [210, 623], [626, 710]],
"138751": [[1, 110], [112, 147]],
"138919": [[62, 150]],
"138920": [[1, 55]],
"138924": [[1, 62]],
"138937": [[1, 26]],
"138939": [[1, 26]],
"139020": [[227, 316], [319, 617]],
"139098": [[1, 92], [94, 121], [123, 160], [162, 179]],
"139100": [[1, 13], [15, 15], [17, 19], [21, 102], [104, 104], [106, 205], [207, 307]],
"139103": [[7, 154], [157, 414], [416, 416], [418, 449]],
"139364": [[1, 48], [51, 73]],
"139365": [[1, 9], [12, 14], [16, 82], [84, 84], [86, 112], [114, 166], [168, 179], [181, 250]],
"139370": [[1, 175], [177, 254], [257, 445], [447, 468], [472, 504], [506, 562], [564, 581], [583, 619], [622, 645]],
"139372": [[20, 46], [48, 173], [175, 194]],
"139375": [[1, 47]],
"139399": [[75, 107], [109, 125]],
"139407": [[1, 976], [978, 1074], [1076, 1264]],
"139457": [[18, 73]],
"139458": [[1, 395]],
"139459": [[1, 55]]
}
```

C.3 Jet p_{T} Binning

The binning in jet p_{T} is chosen to approximately follow the assumed resolution of the CMS calorimeter and is given in Tab. C.2. Note that within the analysis the high jet p_{T} bins are merged as described in 4.1.1.

Lower limits in jet p_{T} [GeV]									
64	66	69	71	74	76	79	81	84	87
90	94	97	101	105	109	114	118	123	128
133	138	143	148	153	158	163	168	174	179
185	190	196	202	208	214	220	226	232	238
245	251	258	265	272	279	286	293	300	307
315	322	330	338	346	354	362	370	378	386
395	403	412	421	430	439	449	458	468	477
487	497	507	517	527	537	548	559	570	581
592	603	615	626	638	650	662	674	686	698
711	724	737	750	763	776	790	804	818	832
846	860	875	890	905	920	935	951	967	983
999	1015	1032	1049	1066	1083	1101	1118	1136	1154

Table C.2: *Lower limits of the jet p_{T} bins used in this analysis. The bin-width is chosen to be roughly compatible with the jet energy resolution at the respective bin centre.*

C.4 Software

A broad variety of software tools have been developed to analyse the huge amount of data today's very complex High Energy Physics experiments produce. To enable a flawless exchange and reuse of developed algorithms and programs, the LHC experiments each rely on their specific well defined and centrally managed software frameworks. Within CMS most tools are programmed using the object oriented C++ programming language and the whole framework is compatible with or includes other open source programs, like `ROOT`.

C.4.1 CMSSW

All tools and the well as the definition and implementation of the data structure within CMS experiment data is packaged within the CMS Software (CMSSW) framework. The central data model is based on the physics event. A stored event consists of the recorded raw data and conditions from a single triggered bunch crossing for both, data and MC simulation. In the simulated data the MC truth information is also saved. Several levels of data files according to the respective depth of the executed CMSSW analysis chain, which consists of the following steps:

- GEN: Physics process calculation from theory.
- SIM: Simulation of particle interactions with detector material.
- DIGI: Simulation of the electronic readout of the detector.
- L1: Simulation of the low-level collision event selection.
- DIGI2RAW: Matching between simulated and real detector response.
- RAW2DIGI: Conversion from real to simulated detector response.
- RECO: High level physics object reconstruction (e.g. Jets,Tracks)

The central intention of the CMS framework is to provide a data reconstruction and data analysis framework implemented in a modularised approach and steered by configuration files written in the Python [106] scripting language. The framework has a common interface to several different MC generators to provide a wide variety of simulated theoretical predictions.

C.4.2 JUEZ

Although the CMSSW framework offers all tools required by physics analyses, it has a major drawback. The computing infrastructure used to run analyses has to have a CMSSW framework installation. This is only feasible on compatible operating systems like Scientific Linux 4. To enable a fast turnaround in the development of analysis code and production of results even on private desktop and notebook systems, the events considered for the analysis can be translated from the CMSSW file formate to a reduced format which stores only required information for the analysis. This process is called *skimming*. The resulting skims can then be run on multiple times in the course of the development of an analysis, yielding for example bare ROOT files, containing only histograms. In a joint effort with other ongoing analyses at the IEKP, a data format called JUEZ and the corresponding analysis tools based on ROOT were developed. The JUEZ format consists of the following data structures:

- KAGenParticle: Derived from ROOT TParticle, contains also generator level information such as mother-daughter relations.
- KAJetID: Contains members and classes for both CaloJet and PFJet identification.
- KAL1: Level-1 trigger information.
- KAParticle: Class for generalized particles: Can contain muons as well as jets and offers special functionality such as jet area information.
- KATrack: Contains reconstructed tracks.
- KAVertex: Vertex information.

C.5 Official CMS Results

The results of this work contributed to the official CMS publication [74] for the inclusive jet cross-section measurement as shown in Figure C.1.

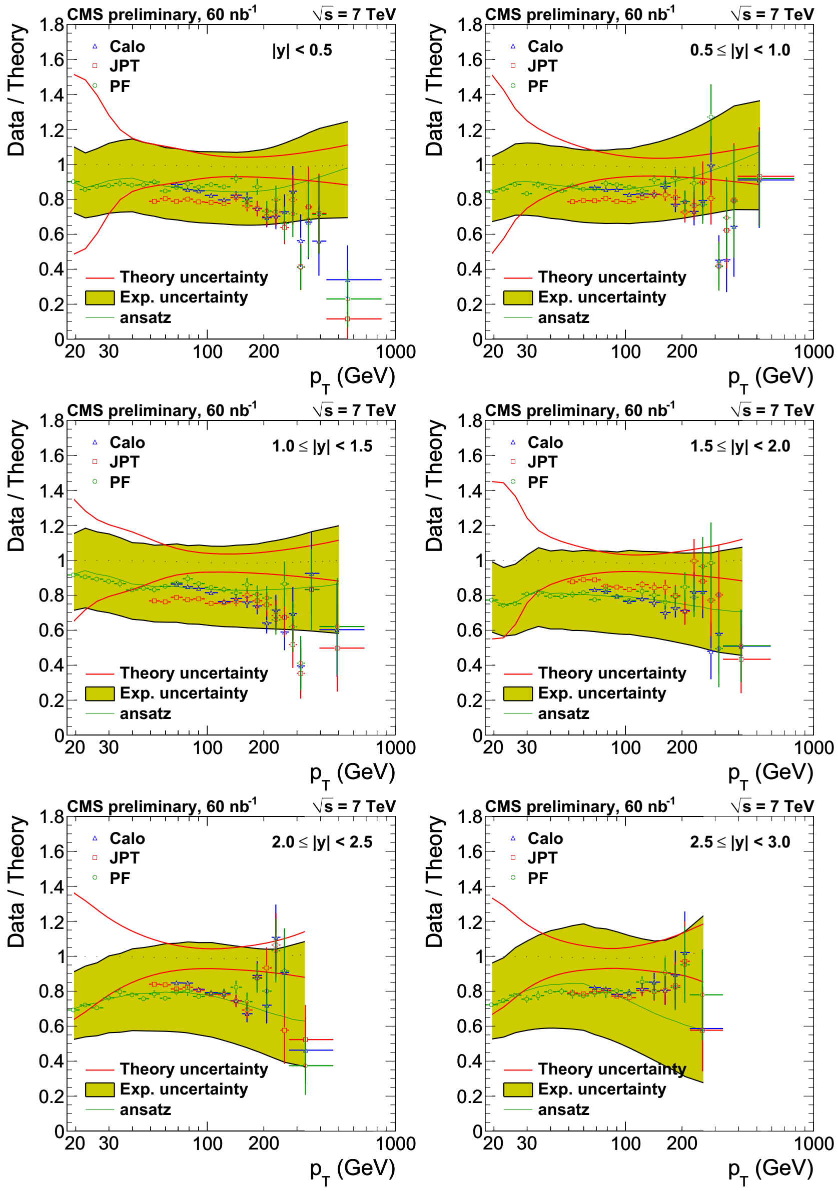

Figure C.1: *Comparison between theory predictions and three different jet types used with the CMS for inclusive jet cross-section measurements. All three different jet types agree well with each other within the experimental uncertainties as well as with the theoretical NLO calculation. From [74]*

List of Figures

1.1 The basic QCD interactions Feynman graphs 11
1.2 World average of α_s at $M(Z^0)$ and $\alpha_s(Q)$ relation 13
1.3 Deep Inelastic Scattering 14
1.4 CTEQ6.6 PDFs for $Q^2 = 200$ GeV 16
1.5 Parton kinematics and cross-sections at Tevatron and the LHC 17
1.6 Event simulation of multi purpose generators 18
1.7 Infrared and collinear safety of jet algorithms 25

2.1 The Large Hadron Collider LHC at CERN 30
2.2 The CMS detector 33
2.3 Profile of the CMS detector 34
2.4 Overview of the CMS silicon tracker 35
2.5 Overview of the ECAL layout of CMS 36
2.6 Slice through the CMS HCAL 37
2.7 The tiered structure of the WLCG. 39

3.1 x86 Architecture with and without VM support 43
3.2 Paravirtualization (top) and full virtualisation (bottom). 46
3.3 Typical HPC cluster partitioning 48
3.4 Prologue and Epilogue: Virtual Machine as a Job 50
3.5 Linux Kernel as a Hypervisor (KVM) 51
3.6 VMBench Results 52
3.7 Virtualized job workflow 53
3.8 The ViBatch concept. 55

4.1 Calorimeter Jet Reconstruction within CMS 61

4.2 Factorized jet energy corrections within CMSSW 62
4.3 Jet energy correction Factors 63
4.4 Jet p_T resolution from MC truth 65
4.5 HLT Trigger turn-on curve 68
4.6 MET/$\sum E_\mathrm{T}$ for collision and non-collision events. 69
4.7 MET/$\sum E_\mathrm{T}$ selection efficiency. 70
4.8 The inclusive jet cross-section spectra 72
4.9 Unsmearing fit quality and correction factors 74
4.10 Final unsmeared inclusive jet cross-section spectra 75
4.11 Systematic experimental uncertainties 75
4.12 Non-Perturbative Correction Factors. 79
4.13 Theory Uncertainties 80
4.14 NLO Uncertainties according to PDF4LHC 81
4.15 Theory and Data spectra 83
4.16 Comparison of theory and measurement 84

C.1 Official CMS Results 110

List of Tables

1.1 Fundamental particles 6
1.2 Fundamental Interactions 7
1.3 Colour representation of quarks 10

2.1 LHC machine parameters at nominal running 31

4.1 Rapidity bins and the corresponding detector regions. 60
4.2 Event selection numbers 72

C.1 7TeV Datasets analyzed with estimated luminosity and recorded number of events. 105
C.2 Lower limits of the jet p_{T} bins 107

Bibliography

[1] M. Gell-Mann and Y. Ne'eman, *The eightfold way*. Frontiers in Physics. Benjamin, New York, NY, 1964. 5

[2] G. Zweig, *An su$_3$ model for strong interaction symmetry and its breaking; part ii*, . 5

[3] S. L. Glashow, *Partial Symmetries of Weak Interactions*, *Nucl. Phys.* **22** (1961) 579–588. 6

[4] A. Salam, *8th Nobel Symposium on Elementary Particle Physics: Relativistic Groups and Analyticity*, . 6

[5] S. Weinberg, *A Model of Leptons*, *Phys. Rev. Lett.* **19** (1967) 1264–1266. 6

[6] D. J. Griffiths, *Introduction to Elementary Particles*. John Wiley & Sons, Inc., 1987. 6

[7] C. S. B. Povh, K. Rith and F. Zetsche, *Teilchen und Kerne*. Springer Verlag, 2004. 6

[8] L. de Broglie, *Recherches sur la thorie des quanta*, *Ann. Phys.* **3** (1925) 22. 6

[9] P. A. M. Dirac, *The quantum theory of the emission and absorption of radiation*, *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **114** (1927), no. 767 pp. 243–265. 8

[10] TOTEM Collaboration, *The TOTEM experiment at the CERN Large Hadron Collider*, *JINST* **3** (2008) S08007. 8

[11] G. Dissertori, I. G. Knowles, and M. Schmelling, *Quantum Chromodynamics: High Energy Experiments and Theory*. Oxford University Press, 2002. 9

[12] K. Nakamura and P. D. Group, *Review of particle physics*, *Journal of Physics G: Nuclear and Particle Physics* **37** (2010), no. 7A 075021. 12

[13] S. Bethke, *The 2009 World Average of* $\alpha_s(M_Z)$, *Eur. Phys. J.* **C64** (2009) 689, [`0908.1135`]. 13

[14] M. Breidenbach, J. I. Friedman, H. W. Kendall, E. D. Bloom, D. H. Coward, H. DeStaebler, J. Drees, L. W. Mo, and R. E. Taylor, *Observed behavior of highly inelastic electron-proton scattering*, *Phys. Rev. Lett.* **23** (Oct, 1969) 935–939. 15

[15] J. D. Bjorken and E. A. Paschos, *Inelastic electron-proton and* γ*-proton scattering and the structure of the nucleon*, *Phys. Rev.* **185** (Sep, 1969) 1975–1982. 15

[16] R. P. Feynman, *Photon-hadron interactions [by] R. P. Feynman.* W. A. Benjamin, Reading, Mass.,, 1972. 15

[17] C. G. Callan and D. J. Gross, *High-energy electroproduction and the constitution of the electric current*, *Phys. Rev. Lett.* **22** (Jan, 1969) 156–159. 15

[18] G. Altarelli, *Partons in quantum chromodynamics*, *Phys. Rept.* **81** (1982) 1–129. 15

[19] Y. L. Dokshitzer, *Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics*, *Sov. Phys. JETP* **46** (1977) 641–653. 16

[20] V. N. Gribov and L. N. Lipatov, *Deep inelastic e p scattering in perturbation theory*, *Sov. J. Nucl. Phys.* **15** (1972) 438–450. 16

[21] G. Altarelli and G. Parisi, *Asymptotic Freedom in Parton Language*, *Nucl. Phys.* **B126** (1977) 298. 16

[22] J. C. Collins and D. E. Soper, *The Theorems of Perturbative QCD*, *Ann. Rev. Nucl. Part. Sci.* **37** (1987) 383–409. 16

[23] Online tool for pdf-plotting, Durham University `http://durpdg.dur.ac.uk/hepdata/pdf3.html/`. 16

[24] J. F. Owens, *Parton distributions in hadrons: The CTEQ program for the global analysis of parton distributions*, . Prepared for Workshop on Deep Inelastic Scattering and QCD (DIS 95), Paris, France, 24-28 Apr 1995. 16

[25] S. Catani, *Aspects of QCD, from the Tevatron to the LHC*, `hep-ph/0005233`. 17

[26] G. Sterman and S. Weinberg, *Jets from Quantum Chromodynamics*, *Phys. Rev. Lett.* **39** (1977) 1436. 19

[27] S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock, and B. R. Webber, *New clustering algorithm for multi - jet cross-sections in e+ e- annihilation*, *Phys. Lett.* **B269** (1991) 432. 19

[28] J. M. Campbell, J. W. Huston, and W. J. Stirling, *Hard interactions of quarks and gluons: A primer for LHC physics*, *Rept. Prog. Phys.* **70** (2007) 89, [`hep-ph/0611148`]. 19

[29] T. Sjostrand, S. Mrenna, and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **05** (2006) 026, [`hep-ph/0603175`]. 20

[30] T. Sjostrand, S. Mrenna, and P. Z. Skands, *A Brief Introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852, [`0710.3820`]. 20, 22

[31] G. Marchesini *et. al.*, *HERWIG: A Monte Carlo event generator for simulating hadron emission reactions with interfering gluons. Version 5.1 - April 1991*, *Comput. Phys. Commun.* **67** (1992) 465. 20

[32] M. Bahr *et. al.*, *Herwig++ Physics and Manual*, *Eur. Phys. J.* **C58** (2008) 639, [`0803.0883`]. 20

[33] D. A. Kosower, *QCD at the dawn of the LHC era*, *AIP Conf. Proc.* **842** (2006) 189, [`hep-ph/0604015`]. 20

[34] S. Catani, F. Krauss, B. R. Webber, and R. Kuhn, *Qcd matrix elements + parton showers*, *Journal of High Energy Physics* **2001** (2001), no. 11 063. 20

[35] R. Frederix, *Recent developments in madgraph/madevent v4*, *Nuclear Physics B - Proceedings Supplements* **183** (2008) 285 – 289. Proceedings of the 9th DESY Workshop on Elementary Particle Theory. 20

[36] CMS Collaboration, *Measurement of the underlying event activity with the jet area/median approach at 0.9 tev*, *CMS Physics Analysis Summary* **CMS-QCD-PAS-10-005** (2010). 21

[37] B. R. Webber, *A qcd model for jet fragmentation including soft gluon interference*, *Nuclear Physics B* **238** (1984), no. 3 492 – 528. 21

[38] G. C. Blazey *et. al.*, *Run II jet physics*, `hep-ex/0005012`. 26

[39] G. P. Salam, *A Practical Seedless Infrared Safe Cone Algorithm*, `0705.2696`. 26

[40] M. Cacciari and G. P. Salam, *Dispelling the N^3 myth for the k_t jet-finder*, *Phys. Lett.* **B641** (2006) 57, [`hep-ph/0512210`]. 26

[41] M. Cacciari, G. Salam, and G. Soyez, "FastJet." `http://fastjet.fr`, 2006. Web Page. 26

[42] G. Soyez, *The SISCone and anti-kt jet algorithms*, `0807.0021`. 27

[43] **CMS** Collaboration, R. Adolphi *et. al.*, *The CMS experiment at the CERN LHC*, *JINST* **3** (2008) S08004. 29, 32

[44] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, *JINST* **3** (2008) S08003. 29

[45] ALICE Collaboration, *The ALICE experiment at the CERN LHC*, *JINST* **3** (2008) S08002. 29

[46] LHCb Collaboration, *The LHCb Detector at the LHC*, *JINST* **3** (2008) S08005. 29

[47] J.-L. Caron, "Overall view of lhc experiments.. vue d'ensemble des experiences du lhc.." AC Collection. Legacy of AC. Pictures from 1992 to 2002., May, 1998. 30

[48] L. Evans, (ed.) and P. Bryant, (ed.), *LHC Machine*, *JINST* **3** (2008) S08001. 31

[49] CMS Collaboration, *The CMS experiment at the CERN LHC*, *JINST* **3** (2008) S08004. 35, 37

[50] CMS Collaboration, *CMS Physics Technical Design Report, Volume I: Detector Performance and Software*, vol. LHCC-2006-001 of *Technical Design Report CMS*. CERN, Geneva, 2006. 36

[51] Scientific Linux
`https://www.scientificlinux.org/`. 39, 100

[52] G. J. Popek and R. P. Goldberg, *Formal requirements for virtualizable third generation architectures*, *Commun. ACM* **17** (July, 1974) 412–421. 42

[53] V. Buege, *Virtualisation of Grid Resources and Prospects of the Measurement of Z Boson Production in Association with Jets at the LHC*. Karlsruhe Institute of Technology, 2008. 43

[54] R. L. Sites, A. Chernoff, M. B. Kirk, M. P. Marks, and S. G. Robinson, *Binary translation*, *Commun. ACM* **36** (February, 1993) 69–81. 44

[55] Kernel Based Virtual Machine
`http://www.linux-kvm.org/page/Main_Page`. 44

[56] Virtio: An I/O virtualization framework for Linux
`http://www.ibm.com/developerworks/linux/library/l-virtio/`. 44

[57] VMware - Virtualisation Software
`http://www.vmware.com/`. 44

[58] Parallels Desktop/Workstation Virtualisation
`http://www.parallels.com/`. 44

[59] The Adeos Project
`http://home.gna.org/adeos/`. 44

[60] The Mac-on-Linux Project
`http://mac-on-linux.sourceforge.net`. 44

[61] Win4BSD
`http://www.win4bsd.com`. 44

[62] The Xen virtual machine monitor
`http://www.cl.cam.ac.uk/research/srg/netos/xen/`. 45

[63] The User-mode Linux Kernel
`http://user-mode-linux.sourceforge.net/`. 45

[64] Virtuozzo - a Virtualisation Software by SWSOFT
`http://www.swsoft.com/en/virtuozzo/`. 45

[65] M. Ruda, J. Denemark, and L. Matyska, *Scheduling virtual grids: the magrathea system*, in *Proceedings of the 2nd international workshop on Virtualization technology in distributed computing*, VTDC '07, (New York, NY, USA), pp. 7:1–7:7, ACM, 2007. 49

[66] N. Fallenbeck, H.-J. Picht, M. Smith, and B. Freisleben, *Xen and the art of cluster scheduling*, in *Virtualization Technology in Distributed Computing, 2006. VTDC 2006. First International Workshop on*, pp. 4 –4, November, 2006. 49

[67] O. Oberst and others., *Virtualizing a batch queuing system at a university grid center*, *LECTURE NOTES IN COMPUTER SCIENCE* **2006, ISSU 4331** (2006) 397–406. 49, 52

[68] T. Hauth, A. Scheurer, and G. Quast, *Dynamic Extensions of Batch Systems with Cloud Resources*, *Presented at CHEP2010 - To be published* (2010). 49

[69] O. Oberst, A. Scheurer, and G. Quast, *Virtualized Batch Worker Nodes: Conception and Integration in HPC Environments*, *Presented at CHEP2010 - To be published* (2010). 52, 55

[70] B. Klein, *Anwendung von Virtualisierungstechniken auf Grid-Ressourcen und Rekonstruktion schwerer Resonanzen aus Zerfllen in Quark- und Gluon-Paare mit dem CMS-Detektor am LHC*, Master's thesis, Institut fuer Experimentelle Kernphysik, Universitaet Karlsruhe, 2008. IEKP-KA/2008-23. 53

[71] ViBatch Code and Documentation `https://ekptrac.physik.uni-karlsruhe.de/trac/BatchVirt/`. 55

[72] C. Baun, M. Kunze, J. Nimis, and S. Tai, *Cloud Computing Web-basierte dynamische IT-Services*. Springer, 2010. 56

[73] T. Hauth, A. Scheurer, and G. Quast, *Dynamic Extensions of Batch Systems with Cloud Resources*, *Presented at CHEP2010 - To be published* (2010). 56

[74] CMS Collaboration, *Measurement of the Inclusive Jet Cross Section in pp Collisions at 7 TeV*, *CMS Physics Analysis Summary* **CMS-PAS-QCD-10-011** (2010). 59, 68, 75, 79, 110

[75] CMS Collaboration, *Plans for Jet Energy Corrections at CMS*, *CMS Physics Analysis Summary* **CMS-PAS-JME-07-002** (2008). 62

[76] CMS Collaboration, *Offset Energy Correction for Cone Jets*, *CMS Physics Analysis Summary* **CMS-PAS-JME-09-003** (2009). 62

[77] CMS Collaboration, *Determination of the Relative Jet Energy Scale at CMS from Dijet Balance*, *CMS Physics Analysis Summary* **CMS-PAS-JME-08-003** (2009). 62

[78] P. Gupta, B. C. Choudhary, S. Chatterji, and S. Bhattacharya, *Study of direct photon plus jet production in CMS experiment at $\sqrt{s}$ = 14-TeV*, *Eur. Phys. J.* **C53** (2008) 49. 62

[79] CMS Collaboration, *Determination of jet energy scale using $Z \to e^+e^- +$ jet p_T balance and procedure for combining data driven corrections*, *CMS Physics Analysis Summary* **CMS-PAS-JME-08-005** (2009). 62

[80] CMS Collaboration, *Jet performance in pp collisions at 7 tev*, *CMS Physics Analysis Summary* **CMS-PAS-JME-10-003** (2010). 62, 63, 64, 65

[81] W. Erdmann, *Vertex reconstruction at the cms experiment*, *Journal of Physics: Conference Series* **110** (2008), no. 9 092009. 68

[82] A. Heister, P. Lawson, K. Kousouris, D. Mason, R. Harris, L. Lebolo, O. Oberst, A. Oehler, K. Rabbertz, F. Stober, D. Duggan, and K. Rose, *Inclusive Jet Pt Spectrum with Calorimeter Jets at 7 TeV*, *CMS Analysis Note* (2010). 69, 70

[83] CMS Collaboration, *Calorimeter jet quality criteria for the first cms collision data*, . 69

[84] **D0** Collaboration, B. Abbott *et. al.*, *Inclusive jet production in $p\bar{p}$ collisions*, *Phys. Rev. Lett.* **86** (2001) 1707, [`hep-ex/0011036`]. 73

[85] **CDF** Collaboration, A. A. Affolder *et. al.*, *Measurement of the inclusive jet cross section in $\bar{p}p$ collisions at $\sqrt{s} = 1.8$ TeV*, *Phys. Rev.* **D64** (2001) 032001, [`hep-ph/0102074`]. 73

[86] CMS Collaboration, *Commissioning of the Particle-flow Event Reconstruction with the first LHC collisions recorded in the CMS detector*, *CMS Physics Analysis Summary* **CMS-PAS-PFT-10-001** (2010). 74, 76

[87] T. Hansen and P. Schleper, *Single Hadron Response of the CMS Calorimeter. oai:cds.cern.ch:1308725*. PhD thesis, Hamburg U., Hamburg, 2010. Presented on 2010. 74

[88] CMS Collaboration, *Tracking and Vertexing Results from First Collisions*, *CMS Physics Analysis Summary* **CMS-PAS-TRK-10-001** (2010). 76

[89] CMS Collaboration, *Measurement of cms luminosity*, *CMS Physics Analysis Summary* **CMS-PAS-EWK-10-004** (2010). 76

[90] Z. Nagy, *Three-jet cross sections in hadron hadron collisions at next-to-leading order*, *Phys. Rev. Lett.* **88** (2002) 122003, [`hep-ph/0110315`]. 77, 78

[91] T. Kluge, K. Rabbertz, and M. Wobisch, *Fast pQCD calculations for PDF fits*, in *14th International Workshop on Deep Inelastic Scattering (DIS 2006), 20-24 Apr 2006*, (Tsukuba, Japan), p. 483, Apr., 2006. `hep-ph/0609285`. 77

[92] T. Kluge, K. Rabbertz, and M. Wobisch, “fastNLO — fast pQCD calculations for hadron-induced processes.” `http://hepforge.cedar.ac.uk/fastnlo`, 2005. Web Page. 77

[93] P. M. Nadolsky *et. al.*, *Implications of CTEQ global analysis for collider observables*, *Phys. Rev.* **D78** (2008) 013004, [`0802.0007`]. 77

[94] **Particle Data Group** Collaboration, C. Amsler *et. al.*, *Review of Particle Physics*, *Phys. Lett.* **B667** (2008) 1. 77

[95] M. R. Whalley, D. Bourilkov, and R. C. Group, “LHAPDF - the Les Houches Accord PDF Interface.” `http://hepforge.cedar.ac.uk/lhapdf`, 2005. Web Page. 78

[96] M. R. Whalley, D. Bourilkov, and R. C. Group, *The Les Houches Accord PDFs (LHAPDF) and Lhaglue*, `hep-ph/0508110`. 78

[97] **PDF4LHC** Collaboration, J. Huston, *PDFs for the LHC*, *PoS* **DIS2010** (2010) 036. 78

[98] A. D. Martin, W. J. Stirling, R. S. Thorne, and G. Watt, *Parton distributions for the LHC*, `0901.0002`. 78

[99] R. D. Ball *et. al.*, *A first unbiased global NLO determination of parton distributions and their uncertainties*, `1002.4407`. 78

[100] H.-L. Lai *et. al.*, *New parton distributions for collider physics*, *Phys. Rev.* **D82** (2010) 074024, [`1007.2241`]. 78

[101] S. Frixione and B. R. Webber, *Matching NLO QCD computations and parton shower simulations*, *JHEP* **06** (2002) 029, [`hep-ph/0204244`]. 78

[102] M. Heinrich *et. al.*, *Non-perturbative Corrections to Inclusive Jet Spectra at 10 TeV*, *CMS Analysis Note* (2009). 78

[103] K. Rabbertz, “Comparison between nlo calculations..” Kindly provided through private communication., December, 2010. 81

[104] Enabling Grids for E-sciencE
`http://public.eu-egee.org/`. 100

[105] dCache
`http://dcache.desy.de/`. 103

[106] G. V. Rossum, *The Python Language Reference Manual*. Network Theory Ltd., September, 2003. 108

Mein größter Dank gebührt Herrn Professor Günter Quast und Herrn Dr. Marcel Kunze, für die Betreuung dieser Arbeit. Nicht nur die Geduld und ihr Einsatzt für die Mitglieder ihrer Arbeitsgruppen sondern auch die Bereitschaft, ihr unerschöpfliches Wissen zu teilen haben maßgeblichen Anteil am erfolgreichen Anfertigen dieser Arbeit.
Besonders danke ich ebenfalls Dr. Armin Scheurer, Dr. Volker Büge, Dr. Klaus Rabbertz und Dr. Andreas Oehler, ohne welche diese Arbeit in dieser From nicht möglich gewesen wäre. Ihre Kommentare, Anleitungen und ihre Motivation haben mich enorm vorangebracht.

Vielen Dank für die hervorragende Zusammenarbeit in freundschaftlicher Atmosphäre an Dr. Michael Heinrich, Dr. Danilo Piparo von denen ich unwahrscheinlich viel lernen konnte und mit denen ich auch sehr viel Zeit außerhalb der Teilchenphysik verbringen konnte.

Danke auch an alle Kollegen mit denen ich das Büro teilte und die auf die eine oder andere Art zu dieser Arbeit beitrugen, sei es durch technische Unterstützung oder durch Korrekturlesen. Im einzelnen sind das Christophe Saout, Kristov Hackstein, Joram Berger, Fred Stober, Thomas Hauth, Manuel Zeise, David Kernert.

Natürlich möchte ich auch allen Mitgliedern des EKP und des SCC danken, die ich hier nicht namentlich genannt habe, für die Bereitstellung und den Unterhalt der Computerinfrastruktur und die tolle Atmosphäre.
Besonderen Dank gebührt auch allen Kollegen in der CMS Kollaboration sowie dem KIT für die Bereitstellung der finanziellen Mittel.

Ganz besonders möchte ich meiner Familie und meiner Freundin danken, die mich stets unterstützt haben. Außerdem natürlich meinen Freunden danke für die gute Zeit.