Large-Scale Data Management and Analysis

Big Data in Science

1st Edition

LSDMA

Content

Editorial

Dear Readers,

welcome to the first edition of the LSDMA brochure with many interesting articles about the fascinating R&D by the partners in the LSDMA consortium. The field of Big Data is broad: some articles give in-depth details on specific activities, others give an overview on Big Data research topics.

LSDMA stands for “Large-Scale Data Management and Analysis” [1] and is a portfolio project funded by the German Helmholtz Association for a five year period. Under leadership of KIT, four Helmholtz centres (KIT, FZ Jülich, DESY, GSI), six universities (University of Hamburg, University of Ulm, Heidelberg University, HTW Berlin, TU Dresden and GU Frankfurt) and the German Climate Computing Centre (DKRZ) have joined to optimise data life cycles in selected scientific communities.

In our Data Life Cycle Labs (DLCLs), data experts perform joint R&D together with the scientific communities to optimise data management and analysis tools, processes and methods. Complementing the activities in the DLCLs, the Data Services Integration Team (DSIT) focuses on the development of generic tools and solutions, which are applied by several scientific communities. Examples are authentication, authorization, identity management, archiving or metadata.

LSDMA organises an annual, international symposium on “The Challenge of Big Data in Science” each autumn - more information about the next symposium can be found at our website http://www.helmholtz-lsdma.de. Furthermore, community forums, technical forums and PhD meetings bring together the LSDMA consortium partners with the scientific communities.

I would like to thank all authors who contributed to this brochure. They are responsible for the contents of the respective articles and are your first contacts for any questions or comments. On page 39, you find a list of all LSDMA subprojects and their leaders as well as contact information for the project.

I want to express our thanks to the German Helmholtz Association and the German Federal Ministry of Education and Research.

Have a nice time reading the brochure.

A. Streit

Prof. Dr. Achim Streit

Lead-PI of the LSDMA Helmholtz Portfolio Project

Big Data in Photon Science

Hermann Heßling - HTW Berlin

Scientific Motivation

In photon science, nano-structures of tiny samples are explored. An ultra-short X-ray flash propagates through a sample and generates a coherent diffraction image. The samples are destroyed due to the high intensity of every single flash. It is essential that the images are taken before the damage process sets in. Even from complex biological samples, e.g. proteins and viruses, high-resolution images can be obtained.

The data rate produced in photon science will grow rapidly during the next years, and the success of the upcoming experiments increasingly depends on handling and processing huge amounts of data. Current analysis tools in photon science are not designed for high data volumes and will not scale well to cope with the strongly increasing demands.

Efficient data reduction is a major topic in photon science. For pre-selecting events already during data-taking an on-detector vetoing has to be developed. The real-time data reduction has to be automated in a manner acceptable to photon scientists and the needs of ever-changing experiments: they are more dynamic than the LHC experiments at CERN.

Big Data Challenges

The setup at the Lineac Coherent Light Source (LCLS) is typical for experiments in photon science (see Figure 1). Due to limitations in preparing the beam that transports the samples, the laser pulses hit only a small fraction of the samples (approx. 5%). Nevertheless, every data frame is saved to disk and, in the subsequent workflow, “blank diffraction images” are identified and removed.

Early vetoing has not been employed in any photon science experiment to date: due to the fear of losing valuable data and because it has been technically feasible to save all data.

The upcoming new experiments will lead to a shift in paradigm: saving all data will no longer be feasible. Rapid experimental feedback and data analysis is critical for the experiments in photon science. At the European XFEL this is made challenging by a pulse repetition rate of up to 27,000 pulses per second, leading in turn to extremely high data rates and large data volumes. Without rapid data vetoing it will not be possible to take full advantage of the extremely intense, ultra-short pulses of laser light.

Events that make it through the initial event veto layers may be amenable to either rejection or reduction in size using parallel algorithms. This layer forms part of the European XFEL conceptual DAQ plan and is deemed critical for reducing the amount of stored data to manageable levels. However, details of the algorithms for weeding out events of interest and performing preliminary data reduction remain to be defined and implemented.

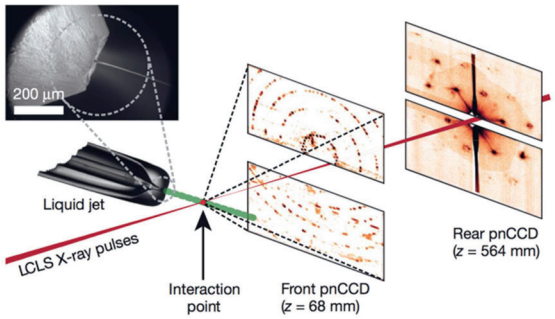

Figure 1 *In femtosecond nanocrystallography, Ultra-short X-ray pulses from a free electron laser hit tiny crystals. The samples are transported in a liquid or gas stream, perpendicular to the laser beam. The detector consists of two panels (equipped with 1024 x 1024 pixels each) and records coherent diffraction images at the rate of up to 200 Hz* [2; 3].

Experiments in photon science can be divided into various categories:

- Two-dimensional imaging of single objects,
- Three-dimensional imaging by analyzing ensembles of identical single objects, and
- Imaging of organic giants, such as proteins and viruses.

Each category demands its own data analysis solution. For all of these experiments it is necessary to separate useful hits from blank images. The Cheetah software [3], which will be part of the XFEL analysis workflow, is used to perform this task.

Identification of Blank Diffraction Images

In nanocrystallography, a laser flash propagates through a sample and is, thereby, broken into discrete sharp bright spots, the so-called Bragg peaks (see Figure 2). Roughly, the pattern structure of the Bragg peaks is related to the Fourier transform of the electron density of the crystallized object and the electron density, in turn, is given by the spatial distribution of the atoms in the unit cell of the object. Therefore, the locations, the intensities, and the widths of the Bragg peaks are essential for reconstructing an image of the sample. A reconstruction is made difficult as the detector records always a photon background and since

Bragg peaks not sufficiently bright, cannot be resolved.[1]

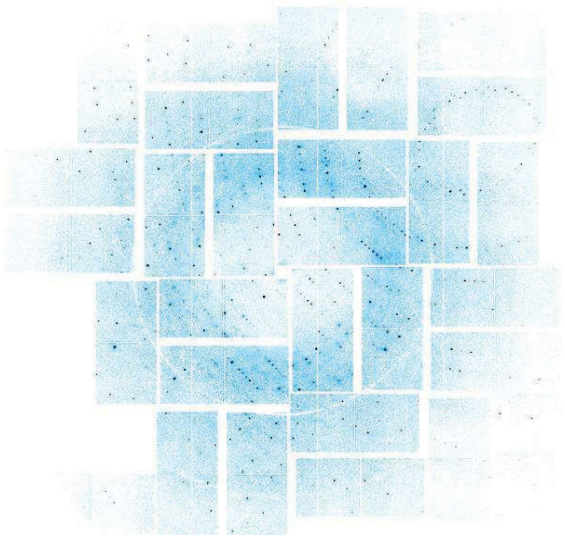

Figure 2 *Diffraction image from a single nano-crystal of the protein lysozyme taken at the LCLS* [4]. *The dark spots show the Bragg peaks. The "halo ring" around the center of the detector shows the photon background.*

Neural networks are used successfully for image recognition. Applying neural networks directly for exploring diffraction patterns is challenging. Firstly, the input from of the order of one million pixels has to be taken into account. Secondly, as the orientation of each single sample in the stream is not known, a priori, a huge training set of images and a large set of hidden neurons is needed to let the neural network learn all orientations of relevance. It should be noted that "deep neural networks" (built of many hidden layers of neurons) seem to have conceptual recognition problems as already an almost unnoticeable modification of input values may lead to misclassifications [5].

Concerning the problem of recognizing blank diffraction images the situation seems to be more comfortable since it is expectable that only a small amount of "relevant observables" is sufficient to determine whether or not a frame is useful in the subsequent analysis workflow. A natural approach is to reduce the large amount of information stored in the total set of pixels by applying coarse-graining methods.

In [6], LCLS data from two proteins and a virus were analyzed. It was shown that an identification of blank diffraction images is feasible provided the Bragg peaks are sufficiently brighter than the background. Three "relevant observables" were suggested and analyzed by a small neural network, see Figure 3. The results are shown in Figure 4. For the protein CatB a recognition rate of more than 90 % was achieved (after removing the photon background and taking the loss of intensity at large scattering angles into account[2]). The recognition rates of the other two probes were significantly smaller mostly because they showed almost no clear Bragg peaks.

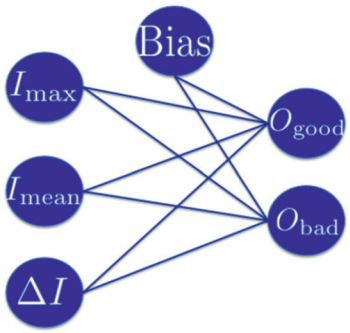

Figure 3 *Neural network with three input neurons, two output neurons and no hidden layer. The lines indicate weights that were determined from a training set of diffraction images with known properties (either useful or blank images). For a given diffraction image I_{max} denotes the maximum photon number out of all pixels, I_{mean} is the mean photon number from all pixel with a non-vanishing photon number, and ΔI the associated standard deviation. The value of the bias neuron is always set to one. A diffraction image is considered as recognized correctly if the output neurons obey the inequality $O_{good} > O_{bad}$.*

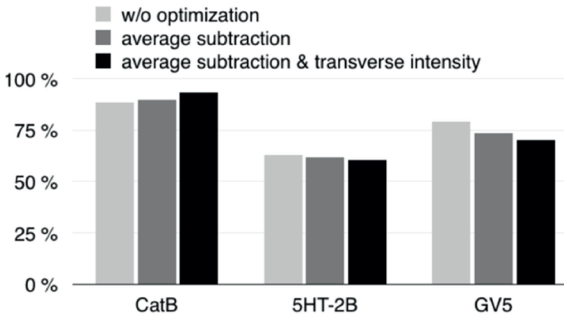

Figure 4 *Identification rates of two proteins (CatB, 5HT-2B) and a virus (GV5)* [6]. *The identification rate is determined from the sum of the truly recognized useful images and the truly recognized blank images.*

[1] In diffraction experiments, image reconstruction is especially difficult as only intensities are measured but not phases. The phases store a significant amount of information about the positions of the atoms. To solve this "phase problem", several methods are used, e.g. Patterson's autocorrelation method allows a determination of the relative positions of atoms.

[2] The photon number I_i of the i-th pixel is replaced by the *transverse intensity* $I_i \sin(\mu_i)$ where μ_i is the angle between the beam axis, the interaction point, and the location of the i-th pixel.

By incorporating the neural network into the online analysis framework it should be feasible to obtain a strong and effective reduction of the incoming data flood already during the data-taking period, at least for objects with clear Bragg peaks.

Cross Application Communication on NUMA

A detailed offline analysis of diffraction data in photon science can be done in parallel, at least in principal, as there are no correlations between different diffraction images. (The situation may change if time-series imaging is considered.) For improving significantly the speedup the analysis software, such as Cheetah [3], has to be ported on multi-core systems.

Large multi-core systems are equipped with non-uniform memory architectures (NUMA). Accessing memory associated to remote CPUs is a primary source of slowdown on NUMA. Different speeds on the interconnect links between the cores are also contributing significantly to the slowdown.

Cheetah [3] is used in X-ray diffraction for sorting data according to different criteria, and for rapid filtering of events to reduce significantly the data volume. Data processing in Cheetah is based on a multi-threaded architecture: a single thread reads the data of a diffraction image and passes them to a worker thread from a pool of worker threads. The worker threads are processing their data independently.

Cheetah does not scale if ported directly to a NUMA system, as can be extracted from the red curve of Figure 6.

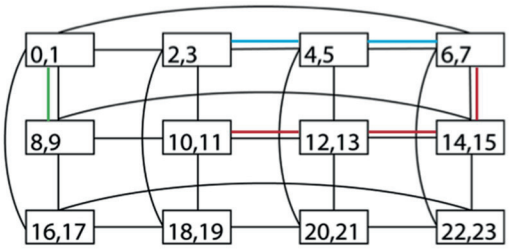

Figure 5 *SGI NUMA system with 144 cores [7]. The system consists of 24 CPUs equipped with 6 cores each. Every socket contains two CPUs. The topology of the connections between the sockets corresponds to a torus. The communication latency between sockets depends on the number of hops. The latency between neighboring sockets (green) is smaller than the latency over two hops (blue) or three hops (red). Each CPU has direct access to a memory of 32 GB (total memory: 768 GB).*

A bad scaling behavior is not untypical in the realm of Big Data. The resolution power of experimental devices and sensors is increasing and more and more data are collected. Often, the software for analyzing data was developed over years and is known for delivering reliable results. However, if it turns out that the flood of data can no longer be processed in a reasonable period of time, a decision has to be taken: is it efficient and effective to extend an existing sequential software by parallelization capabilities, or should a new software be developed from the scratch?

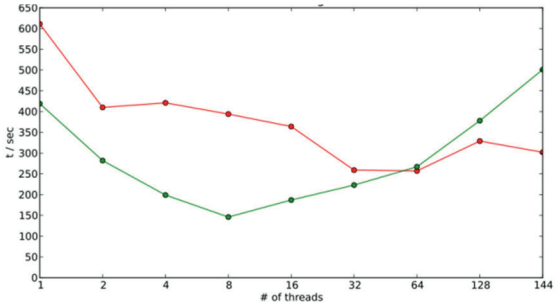

Figure 6 *Scaling of Cheetah* [7]. *The time t for processing 200 diffraction images (taken at LCLS) on a NUMA system (see Figure 5) is shown versus the number of worker threads. Red curve: no optimization, green curve: thread binding.*

The program code of Cheetah is written in C++ (and Python) and running on Linux. For managing threads, Cheetah uses the POSIX library PThread. PThread is steering the binding between the threads and the memory. However, the scheduling is not efficient for Cheetah (red curve, Figure 6) as the worker threads have to communicate quite often over multiple sockets to access their data on remote memory. The green curve of Fig. 5 indicates that the scaling of Cheetah may be improved considerably. It is obtained by binding the threads to the CPUs of the NUMA system. Linux provides a command line interface to a NUMA API that supports thread binding to CPUs and cores.

The speedup of Cheetah that can be extracted from Figure 6 seems to be improvable. The set of parameters with a significant influence on the speedup should be determined systematically. For example, the optimal number of threads to be pinned to a CPU depends critically on the amount of data stored in the diffraction images and on the available local memory of each CPU. Moreover, it seems to be feasible to incorporate a thread binding directly into Cheetah by only weak modifications of the program code. The library *libnuma* provides access to the NUMA API within C/C++ programs.

I/O performance Improvements Using Emerging Technologies

Konstantinos Chasapis, Michael Kuhn, Manuel F. Dolz - University of Hamburg

Scientific Motivation

The continuously increasing needs for data storage in HPC systems have emerged as one of the main obstacles that we have to face as we are moving towards exascale systems. Today, the largest HPC storage systems are in a range of multiple petabytes and suffer from many inefficiencies. Many scientific applications, including climate models, produce a vast amount of data and their performance is limited by the performance of the data storage subsystem.

To this end, the scientific community is working towards the performance improvement of input/output (I/O) operations. As part of this effort, in this work we evaluate the potential benefits of new hardware technologies that can be used in HPC storage systems.

Technologies

Traditionally, the performance gap between the CPU and the HDD is increasing. Starting from the previous decade, solid-state drives (SSDs) are being used to lower this gap. In contrast with HDDs, which are based on mechanical parts, SSDs are made entirely from electronics, which allows them to perform much faster. However, SSDs face two main drawbacks in comparison to HDDs. First is the cost factor since the price per gigabyte of SSDs is much higher than for HDDs; additionally, the durability of the SSDs is limited to a certain amount of write operations.

Moreover, to overcome the CPU clock rate wall and increase even more computing power within the same server, newer architectures have more cores and more CPUs. However, this does not come for free since new obstacles arise. The most important ones are: synchronization between different processes executing in the same CPU and non-uniform memory access (NUMA).

Use Cases

An important part of the data storage infrastructure is the metadata handling of the file system. The requirements of the metadata servers are different from the data storage servers. One of the main differences is the capacity needs: The size of the metadata is negligible in comparison to the actual data. For this specific use case, SSDs are a perfect fit and can be used as HDD replacements.

Results

In our evaluation we measure the effectiveness of SSDs and the implication of NUMA machines in Lustre's metadata server (MDS). In Figure 1 we can see the improvement using SSDs as the backend device of the MDS. We also include a configuration with RAM disk as the underlying storage device to indicate the practical maximum performance. In comparison to HDDs, SSDs deliver almost double the performance for file creations per second and four times more for the unlink operation. For the stat operation there is only a slight improvement because stat highly depends on other parts of the system.

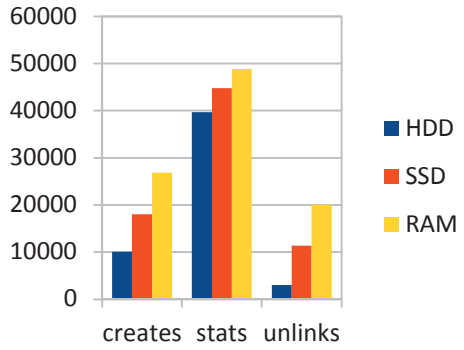

Figure 1 *Performance comparison of common metadata operations using HDD, SSD and RAM disk.*

Figure 2 illustrates the implication of NUMA machines on Lustre's MDS performance. The machine that we used for our experiments is equipped with four CPUs, 12 cores each. From our results we can observe that the performance improvement is limited to a single socket. We are currently carrying out a more extensive analysis to identify the limiting factors of the systems that prevent the performance scaling when using more than one socket.

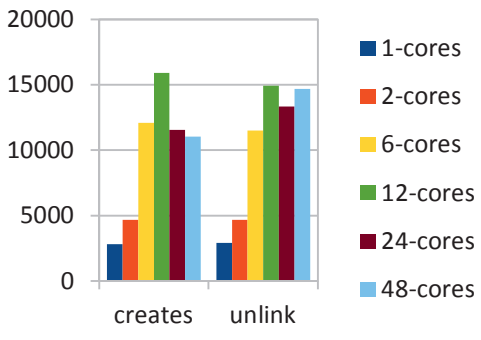

Figure 2 *Performance of metadata operations when using multiple sockets in a NUMA machine.*

KaHIP – Karlsruhe High Quality Partitioning

Yaroslav Akhremtsev, Peter Sanders, Sebastian Schlag, Christian Schulz - KIT

Due to many technical advances of the last decades, networks are used everywhere. Graphs can be used to model relationships in networks or other important data. The graph partitioning problem asks for a division of a graph's node set into k roughly equally sized blocks such that the number of edges that run between the blocks is minimized. For example, in parallel computing good partitionings of unstructured graphs are very valuable. In this area, graph partitioning is mostly used to partition the underlying graph model of computation and communication. Roughly speaking, nodes in this graph denote computation units, and edges represent communication. This graph needs to be partitioned such that there are few edges between the blocks (pieces). Figure 1 shows an example graph stemming from a finite element simulation which is partitioned into four blocks. Other important applications of graph partitioning include route planning, VLSI Design or solving sparse linear equation systems.

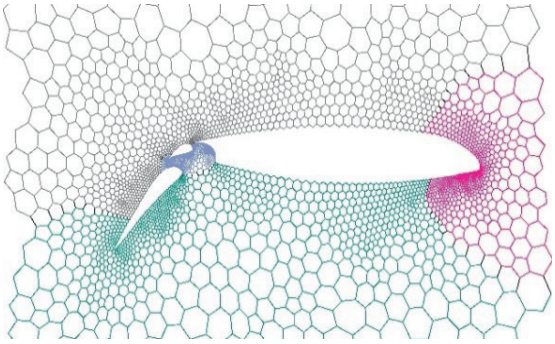

Figure 1 *A mesh that is partitioned into four blocks.*

It is well-known that the problem is NP-complete and that there is no approximation algorithm with a constant ratio factor for general graphs. Therefore mostly heuristic algorithms are used in practice. A successful heuristic for partitioning large graphs is the multilevel graph partitioning (MGP) approach where the graph is recursively contracted to achieve smaller graphs which should reflect the same structure as the input graph. After applying an initial partitioning algorithm to the smallest graph, the contraction is undone and a local search method is used at each level to improve the partitioning induced by the coarser level.

Although several successful multilevel partitioners have been developed in the last 16 years, we had the impression that certain aspects of the method are not well understood. This motivated us to make a fresh start putting all aspects of MGP on trial.

KaHIP – Karlsruhe High Quality Partitioning – is our family of graph partitioning programs that tackle the balanced graph partitioning problem. The framework implements many different algorithms. It includes a number of general purpose multilevel graph partitioning algorithms that use, among other techniques, flow-based methods and more-localized local searches to compute high quality partitions. KaHIP also includes a parallel evolutionary algorithm that is able to compute record setting solution quality in a couple of minutes for graphs of moderate size. Moreover, specialized techniques for different kinds of networks such as road networks or social networks are contained. Figure 2 illustrates the components of the KaHIP framework.

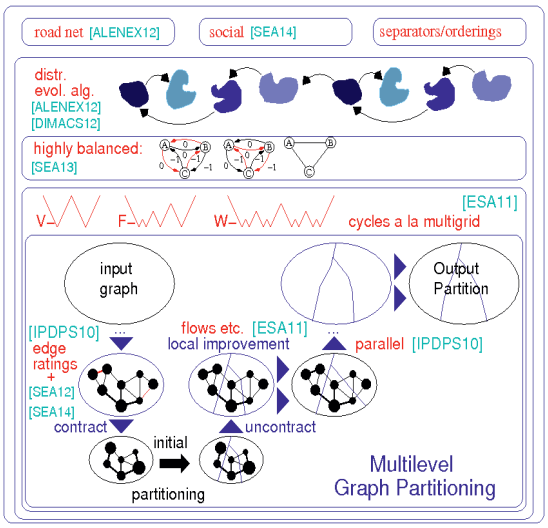

Figure 2 *Components of the KaHIP framework*

Parallel algorithms of the framework are more scalable and achieve higher quality than other state-of-the-art systems. For large complex networks the performance differences are very big. For example, our algorithm can partition a web graph with 3.3 billion edges in less than sixteen seconds using 512 cores of a high performance cluster while producing a high quality partition – none of the competing systems can handle this graph on our system.

KaHIP has been able to improve or reproduce the best known partitioning results in the well-known Walshaw Benchmark for almost all of the inputs using a short amount of time to create the partitions. Moreover, it scored most of the points in the graph partitioning subchallenge of the 10th DIMACS Implementation Challenge on Graph Partitioning and Graph Clustering.

KIT Data Manager: The Repository Architecture Enabling Cross-Disciplinary Research

Thomas Jejkal, Alexander Vondrous, Andreas Kopmann, Rainer Stotzka, Volker Hartmann - KIT

A repository is a managed location in which collections of digital data objects are registered preserved, made accessible and retrievable, and are curated. It is essential that data in a digital data object is accompanied by metadata describing the data contents and organization to enable their reuse in the future. Thus repositories are the mandatory building component for long-term archives.

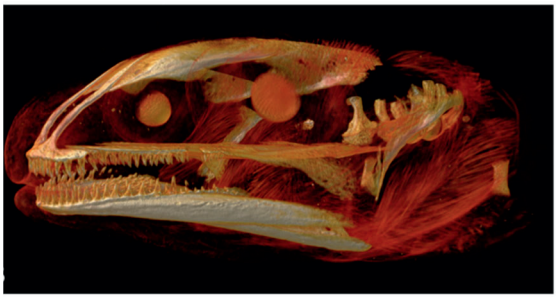

Figure 1 *Volume rendering of a newt larva imaged using fast synchrotron X-ray microtomography.*

In scientific imaging as in other data intensive scientific fields we observe a growing need to build up scientific repositories with various challenging requirements (for details see [8]): Ultra-fast synchrotron tomography produces several Petabytes of experimental and analysis data per year. The data structures are very heterogeneous requiring a flexible data organization.

Figure 2 *Scans of medieval manuscripts (left) and a high-resolution digital elevation model used for discovering unknown archaeological sites (right).*

In the field of humanities a huge variety of data exists that needs to be preserved and accessed over decades and centuries. Thus repositories in humanities require long-term interoperability and must survive mid-term technology changes.

Novel light sheet microscopes produce up to 16 TB of data per day that has to be ingested into a repository. Data ingest rates up to 1 GB/s are mandatory to free the local storage resources for uninterrupted scientific operation.

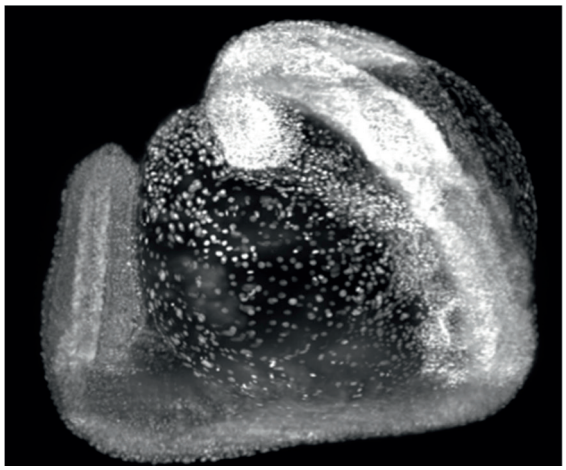

Figure 3 *Maximum intensity projection of an image stack depicting a zebrafish embryo at 24 hours post fertilization.*

Lightoptical nanoscopy produces datasets of up to 200 TB within one single measurement. These extreme large datasets generate novel challenges in handling, analysis and access. Furthermore nanoscopy is a novel imaging method and the interpretation of results is a challenging task. For that it is necessary to dynamically annotate the raw data images for experts to share and to compare their findings.

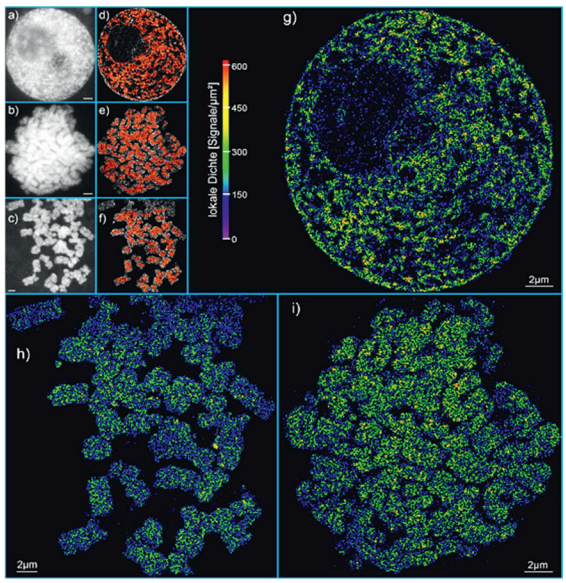

Figure 4 *Histone H2B distribution in HeLa cell nuclei and (pro-) metaphase chromosomes* [9].

Many scientists are aware of the necessity to sustain their data over a long period of time like decades and beyond. Unfortunately, the related efforts are often underestimated leading to short-

term solutions and long-term problems, e.g. missing support for technology and format changes, unclear ownership of data and tremendous amounts of dark data. Changing the view from simple file-based to object-based scientific data represents a paradigm shift solving many problems. Managing digital data objects in a repository allows long-term archiving, data access and sharing in an easy and sustainable way. Apart from sustainability and extensibility by design a repository system must be easy-to-use in order to raise its acceptance. For this purpose, human and machine readable interfaces must be provided by the system to allow immediate access as well as the integration into existing scientific workflows. Finally, especially for LSDMA, the support for high data rates, large-scale data and interfaces to data analysis are desirable to advance data-intensive science. Apart from these features, typical properties of an archive must be supported for long-term preservation:

- Support for data citation
- Access policies
- Bit and content preservation
- Curation

Some institutions, e.g. libraries, are already perfectly equipped for long-term archiving, but their repository systems are customized for specific digital data objects. They are hard to adapt for other communities and are not applicable for experimental data. During the last years, we have supported many different communities solving their data management problems, often by providing custom solutions. Over time, the lessons learned from the various community projects merged into the development of a customizable architecture allowing to build-up repositories for scientific data, the KIT Data Manager.

The idea behind KIT Data Manager is to provide a generic repository architecture that can be fully customized. The goal is to allow almost arbitrary communities to build up repositories for experimental data. If required, different of these repository systems can be combined to enable cross-disciplinary research. For this purpose the set of basic services shown in Figure 5 has been defined.

The architecture integrates seamlessly a collection of basic services and resources which are the building blocks for high-level services. The access to these basic services and resources is abstracted by generic interfaces. This approach has two advantages: On the one hand, these interfaces define a basic set of functionalities and are normally hidden from the user. High-level services can benefit from this interface definition as they can rely on the availability of a particular functionality. On the other hand, standard technologies and software products are used, updated or replaced easily in the background without affecting high-level services, user- or community-applications. If products or technologies on this lowest layer are changed an interface implementation for the new product or technology and a data migration will be necessary in the worst case. However, this can be done without affecting the user's work. This fosters the sustainability of implementations of this architecture.

Public access to the KIT Data Manager is provided by a collection of high-level services. These services offer repository functionalities like storing, accessing and sharing data and metadata as well as enhanced services for lifecycle management, policy enforcement and data processing. As far as possible the high-level services depend only on the basic services. To compose specific community applications a subset of these high-level services can be used.

Access to high-level services is offered by the top layer of the architecture. Various methods are provided depending on the user's needs, e.g. Web

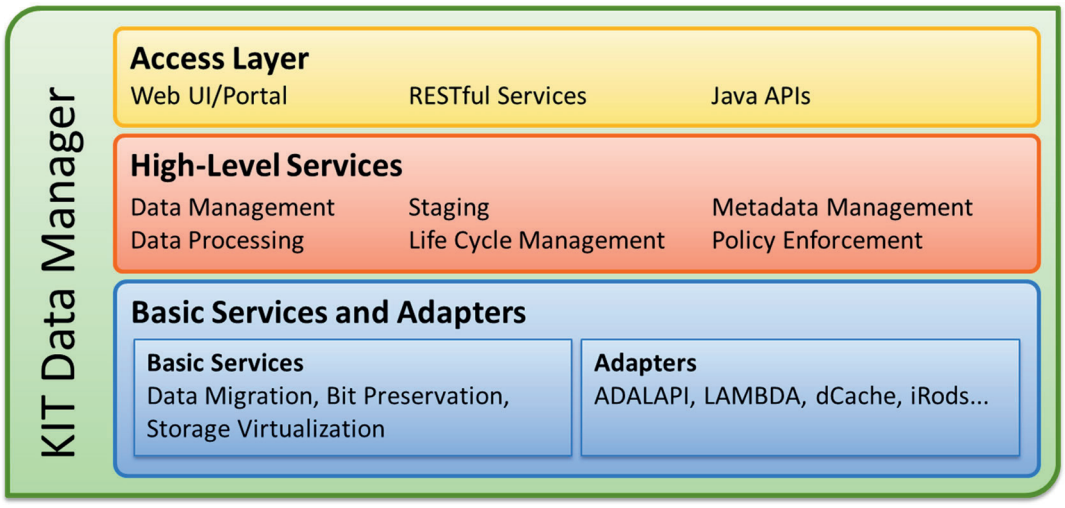

Figure 5 *The architecture of KIT Data Manager consists of several layers providing various levels of abstraction for long-term sustainability, extensibility and flexibility.*

UIs, RESTful service interfaces or plain Java APIs. As the representation of a digital data object, its contained data and metadata is highly community dependent, implementing appropriate access is carried out in close cooperation with community experts. This provides an optimal user experience. Currently, the KIT Data Manager offers reference implementations of a comprehensible set of high-level services:

- Data Management and Staging
- Metadata Management
- Authorization and Sharing
- Metadata Search

This enables the composition of several community repositories by implementing the generic workflow presented in Figure 6 and a few community-specific extensions, e.g. graphical user interfaces on the access layer. Figure 6 shows the generic ingest process for transferring data into a repository provided by a KIT Data Manager instance. Most of the steps are identical for all communities or just have to be slightly customized, e.g. the metadata acquisition (step 1), the choice of the data transfer protocol (step 4) or defining default permissions for accessing the digital data objects (step 6). Other parts like the metadata extraction (step 5) or building up graphical user interfaces (not in Figure 6) are highly domain-specific. In most cases this effort is negligible due to the layered architecture of the KIT Data Manager.

Uniform interfaces, standardized workflows and the extensive enrichment of digital data objects with various kinds of metadata allow setting up relations between digital data objects originating from various scientific disciplines. Links between digital data objects can be determined by distributed searches over metadata directories of different communities in a semi-automated fashion. In the near future, identifying and classifying relationships between digital data objects will be performed fully automatically allowing cross-discipline exploitation of scientific repositories.

The first public release of KIT Data Manager, targeted for the end of 2014, will be available as open source providing all tools and documentation necessary to build up cross-disciplinary repositories. Maintenance, support and extension of basic services, high-level services, community-applications and integration of new data technologies will be continued supported by the Helmholtz programme “Supercomputing and Big Data”. The core development team of four computer scientists is supported by several PhD scientists building up domain-specific repositories and extending the system in close cooperation with the Data Life Cycle Labs “Key Technologies” and “Energy”.

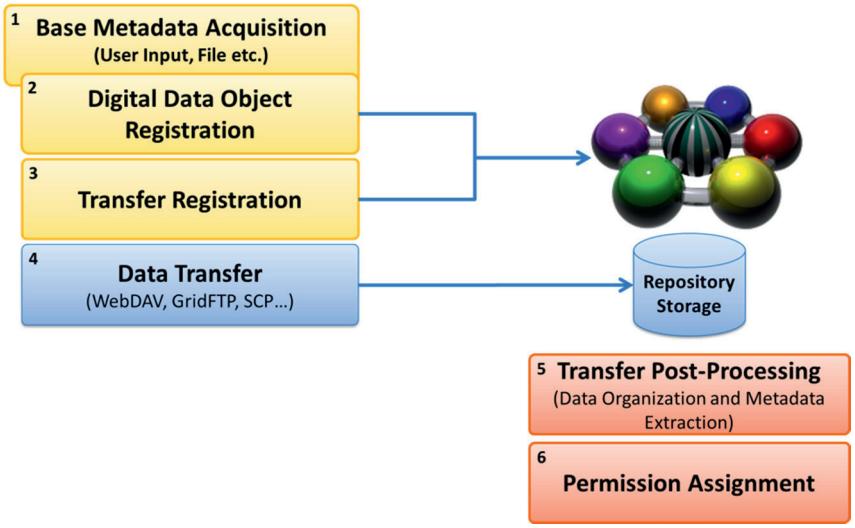

Figure 6 *Generic workflow for initial data ingest implemented for different communities. The different steps are carried out in sequential order. Steps colored yellow and blue are performed on the user side, steps in red are executed on the server side.*

A Geospatial Data Life Cycle Services Framework

Carsten Ehbrecht - DKRZ
Jörg Meyer - KIT

Motivation

ClimDaPs (Climate Data Processing) is the name of a geospatial data services framework to enable the stepwise development and integration of data life cycle management services. The idea is to support end users in data life cycle management activities involving distributed data centers.

Typical data management activities are composed of a set of basic operations. In the framework these operations are exposed as services by the data centres. Services can be composed to build up complex data management workflows.

These services are discoverable and they expose standardised interface descriptions.

Approach

ClimDaPs uses Web Processing Services (WPS) to provide climate data processes via a standard interface. Web Processing Services are standardised by the Open Geospatial Consortium. WPS processes can be chained by a workflow engine.

WPS processes are self-describing. A WPS server can be asked which processes are available (getCapabilities), which input and output parameters a process has (describeProcess) and finally a process can be submitted both synchronously and asynchronously (execute). In case of an asynchronous process the status of the process can be checked. Figure 1 shows these WPS operations.

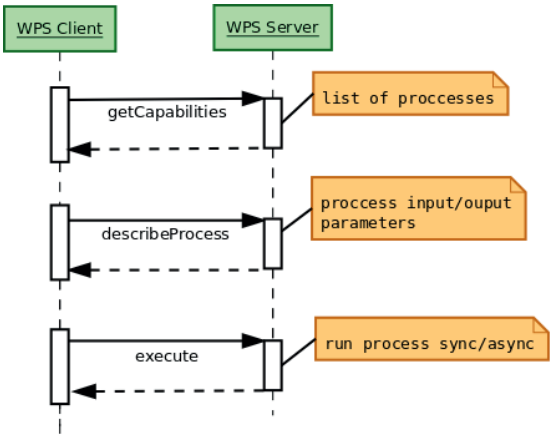

Figure 1 *Web Processing Service Operations.*

WPS processes can be executed by simple HTTP requests or by WPS client libraries like OWSLib for Python.

ClimDaPs comes with a graphical user interface for end users to access services conveniently via the web. The user can compose, invoke, and execute processes with individual parameters. Figure 2 shows the basic interaction between a WPS client (Web UI or terminal) and a WPS server.

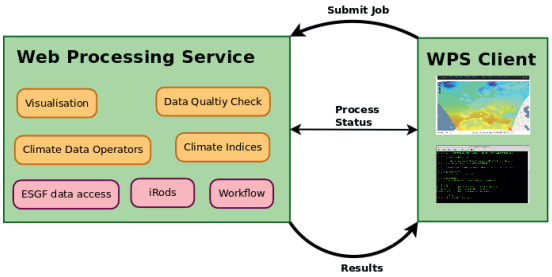

Figure 2 *WPS client submitting a job to a WPS server.*

Use Cases

The following use case describes the quality check workflow of climate model data run by a climate researcher on KIT and DKRZ resources.

A climate researcher wants to copy initial data from KIT to DKRZ. He or she then runs a climate model on compute resources at DKRZ and collects the output at DKRZ storage resources. A further step in the workflow involves running a data post-processing in order to store the data in a standardised format or layout. Data then is checked by data quality check software. In case of successful tests a persistent identifier (PID) is assigned to the data set and it can be published to a worldwide data federation portal at DKRZ. This way the data is visible and accessible via any of the worldwide portals of the Earth System Grid Federation (ESGF). A final workflow step then could be the archival of important parts of the published data.

This use case demonstrates the necessity for a close collaboration between researchers and data scientists coming from different institutions. Also it shows the large variety of services and tools involved in this collaboration.

Figure 3 shows the steps of the described data quality check workflow. The workflow steps are available as WPS processes in ClimDaPs.

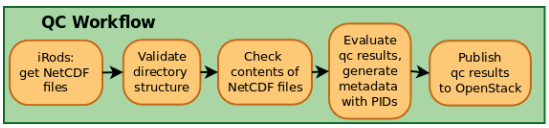

Figure 3 *Steps of the Data Quality Check Workflow.*

Components
The main components of ClimDaPs are shown on Figure 4.

Phoenix user interface
Phoenix is a web-based user interface to provide convenient access to Web Processing Services. It is built on the Python Pyramid web framework. One can choose a process from a WPS server, enter process parameters and execute this process.
Complex workflows which consist of several chained single WPS processes (like the QC workflow) can be submitted by a wizard component. Phoenix has a map to visualize climate data which is based on OpenLayers.
The climate data map is generated by a Web Mapping Service (WMS). WMS supports a time attribute which can be used to step through the map by time. Phoenix uses OpenID for authentication.

Malleefowl WPS Server
Malleefowl is the WPS server part of the ClimDaPs project. It uses PyWPS as WPS service engine and provides a simplified interface to add new WPS processes. Malleefowl comes with some basic WPS processes, for example to access ESGF data and to publish results to a cloud service like OpenStack. Malleefowl has also a workflow-engine (Restflow) to chain WPS processes. The workflow-engine again is accessible by a WPS process.
Malleefowl uses the ncWMS Web Mapping Service to generate maps of NetCDF files.

Available Services
The following set of climate WPS processes is already available by the ClimDaPs project.

Low level data/metadata operations
Low level processes are metadata generation supporting ISO 19139 and ESGF solr metadata schemata, iRods based data transfer and publication of results on an OpenStack storage cloud. There is also a Handle system based persistent identifier (PID) process for assignment and retrieval of identifiers for single data products and data aggregations.

Higher level operations
Higher level operations implemented so far are data quality checking as well as CDO based climate data processing and calculation of climate indices.

Complex Workflows
A complex workflow involving key parts of the described use case is implemented in the Cordex data quality control workflow including quality result publication on an openstack data cloud.

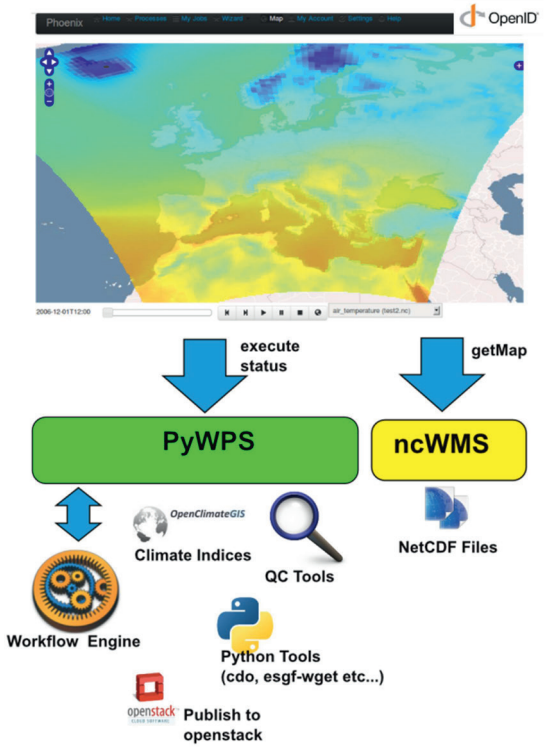

Figure 4 *Phoenix interacting with Web Processing Service and Web Mapping Service.*

Outlook
Further work has to be done on the security infrastructure of WPS servers. The WPS standard does not provide a solution besides using HTTPS and securing the WPS server with username/password. Currently we are using security tokens which are passed as simple WPS parameters.

In addition to this the next steps include the collaboration with European partners to make WPS services interoperable and usable in international collaborations.

FRESCO: A Framework to Estimate the Energy Consumption of Computers

Pavel Efros, Erik Buchmann, Klemens Böhm - KIT

Many application areas, e.g., energy accounting or energy-aware scheduling, require estimates of the energy consumption of computer systems. However, existing estimation approaches often make restrictive assumptions regarding the effort at setup time or run time that is acceptable, they are tailored for specific hardware or software, or they cannot provide accuracy guarantees for the estimates. To tackle these issues, we introduce *FRESCO*, a FRamework for the Energy eStimation of COmputers. FRESCO is a flexible framework for the estimation of the energy consumption of a wide range of computer systems. Based on accuracy requirements and information available, FRESCO deploys and executes appropriate estimators.
In the following we first describe three application scenarios, from which we derive requirements for FRESCO. We subsequently present the classes of effort it must take into account. We then explain the workflow of FRESCO and describe the estimators it integrates. Finally, we present our evaluation of FRESCO.

Application Scenarios

Energy-Aware Management of Data Centers

Increasing the performance per watt is a key performance optimization for data centers. For this purpose, it is important to obtain the energy consumption of a complex IT system as early as the design time of the data center or the allocation time of the various computing workloads. This is important to design the power distribution infrastructure, to decide about computing hardware acquisitions or to find out if a scheduled workload exceeds the cooling capacity. Thus, FRESCO must be able to provide estimates for the typical case that are sufficiently accurate to make educated decisions for hardware acquisitions, and to provide bounds for the energy consumption in extreme cases.

Demand-Response

Demand Response (DR) contains measures that influence energy-consumption patterns. For example, DR might be used to shift energy-intensive computing tasks to times of an energy surplus. Since a data center is a large, adjustable energy sink, it is particularly well suited to perform demand response measures. To realize DR in a data center, an estimator must deliver continuous estimates of the energy consumption of the various IT components at run time.

Computer Energy Accounting and Billing

Energy accounting and billing of the IT infrastructure becomes more and more important. For example, an enterprise might wish to assign each benefactor (a good or a service) the energy costs required for its production. Typically, computer energy accounting requires estimates of the consumption with a frequency of 15 minutes to one hour.

Classes of Effort

We identified two classes of effort an operator can invest to obtain energy consumption estimates:
The *Setup Effort* is necessary to set the estimator up and running. This includes collecting technical specifications of the energy consumption of certain hardware components. Furthermore, it contains the effort of installing a monitoring application to measure run-time parameters of the hardware usage. Finally, the setup effort includes the calibration of an energy consumption profile for a given hardware.
The *Run-Time Effort* includes the network and computational overhead of the estimation process, and the overhead of a monitoring application collecting hardware parameters like CPU frequency, if required by the estimator.

The FRESCO Workflow

With “Target System” we refer to the computer system whose energy consumption FRESCO must estimate. “The Operator” is responsible for installing and maintaining the estimator on the target system. FRESCO consists of three stages “Setup”, “Configuration” and “Estimation”, as shown in Figure 1.
At the *“Setup”* stage, the operator quantifies the trade-off between effort and estimation accuracy for the target system. In particular, he specifies the categories of information obtainable from the target system. At the end of the setup stage, FRESCO either indicates the operator that, given his input, estimation is impossible or lets the operator choose one or a combination of estimators. At the *“Configuration”* stage, FRESCO helps the operator to configure the estimators selected. Finally, at the *“Estimation”* phase, FRESCO runs instances of the chosen estimators with the configuration parameters just fixed on the target system and estimates its energy consumption.

FRESCO Estimators

FRESCO can use static, dynamic or calibration-based estimators or a combination of them. In the following we present a succinct description of each type of estimator.

Static Estimator

Our static estimator uses solely technical information on the target system. Thus, it might be sufficient for any application that does not need time series of estimates. It requires a small effort at setup time for obtaining the hardware specifications, and no effort at run time. The accuracy of this estimator depends on the detail level of its input values and the availability of tolerance bounds. In particular, the static estimator can provide bounds on the energy consumption.

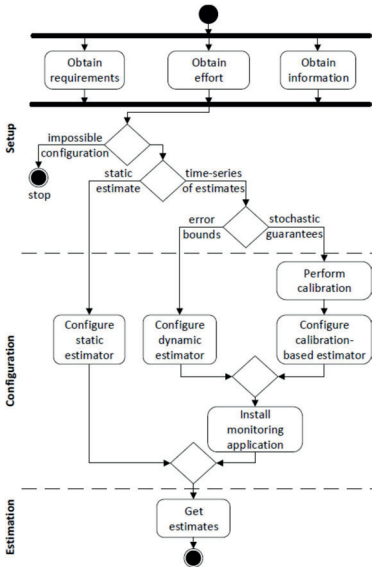

***Figure 1** Workflow of FRESCO*

Dynamic Estimator

Our dynamic estimator models the energy consumption similarly to the static estimator, but installs a monitoring application on the target system to periodically measure detailed load information in real-time, e.g., CPU. Thus, our dynamic estimator generates time series of estimates. It is also possible to integrate specific models for multi-core systems and to model virtual machines as components of the target system.

Calibration-Based Estimator: Our calibration-based estimator executes a detailed benchmark at setup time, which gradually stresses each system component in isolation. At the same time, a power meter records the actual energy consumption, and our monitoring application measures load information such as e.g., CPU frequency. FRESCO then builds a regression model which it uses to estimate energy consumption.

Evaluation

FRESCO operates as intended if its estimates are appropriate for a wide range of applications. In the following we evaluate FRESCO by means of one use case. More details about our evaluation can be found in [10].

The use case “Demand-Response” requires time series of estimates to identify periods of time with high energy consumptions (peaks), together with upper and lower bounds. As the operator is willing to invest only a small effort, FRESCO suggests our dynamic estimator model.

To evaluate this scenario, we let FRESCO estimate the consumption based on the CPU load and on information on the maximal and minimal energy consumptions of our target systems with a frequency of one second. We use these estimates to identify points in time when the energy consumption is above a given threshold. In particular, we evaluate two thresholds (80 and 95%) that consider the difference between the largest and smallest values of a time series T:

$$\theta_1 = 0.8 \cdot (max_{i=1}^{|T|}(T_i) - min_{i=1}^{|T|}(T_i))$$

$$\theta_2 = 0.95 \cdot (max_{i=1}^{|T|}(T_i) - min_{i=1}^{|T|}(T_i))$$

We compute time series of peak consumption from our measured values as well as for the time series FRESCO has estimated, by filtering out all values that are smaller than $\theta_{1,2}$. If our estimates are accurate, FRESCO can identify periods with high energy consumption and can thus enable operators to perform Demand Response.

Figure 2 illustrates the cumulative distribution function (CDF) of the real energy consumption during specific intervals for one of our datasets. The first set of intervals is when FRESCO estimated the consumption to be greater than θ_1 (continuous line). The second set is when FRESCO estimated the consumption to be greater than θ_2 (dashed line).

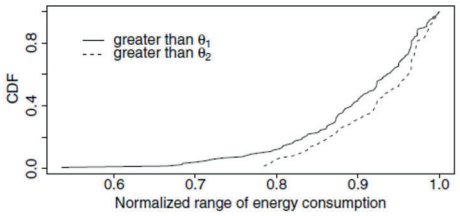

***Figure 2** CDF, Desktop Computer Energy Consumption Dataset*

We observe that, if the estimator predicts a value greater than θ_1, then the real energy consumption is greater or close to θ_1. Thus, in around 88% of all cases, a value predicted to be greater than θ_1, is greater than θ_1. Similar results are obtained for θ_2.

Conclusions

FRESCO is a general and flexible FRamework for the Energy eStimation of COmputers. Depending on the effort the operator is willing to invest and on the requirements of the application, FRESCO can propose and run appropriate estimators with good parameter settings. It gives quality guarantees on the estimates and considers heterogeneous hardware components and loads. Experimental results show that our framework is useful in many business cases.

Towards Smart Archives for Scientific Data

Marco Strutz, Martin Gasthuber - DESY

Purpose of Archives

An archive is a place to store data for a longer period of time where the data is currently not longer needed to be actively accessed in a searchable way.

For any given point in time the archive must be able to provide information about the health and validity of any single managed object. If necessary, data will be automatically migrated to different storage media or even converted to more proper file formats. Also it provides tools and interfaces for information retrieval like browsing and searching methods.

While ingesting new data, policies needed to be assigned to it. Policies help to define constraints bound to data like after how many years data will be purged or under which license the content is allowed to be published. To proper scale up with the number of managed objects, most operations are being executed fully automatically.

Metadata Are Crucial

To be able as user to retrieve back files which have been stored in the past, a common practise is to assign additional information (so called metadata; data about data) along with the data, such as key-value attributes.

A currently wide spread way for users to define metadata is to encode them directly as part of filenames. Also, they save attributes and associated filenames inside separate text-files. Both methods will not scale in terms of how to search large data sets and to assure low latency query times. Furthermore, as the metadata is not part of the archive, the data will not be searchable.

To be able to store vast amount of data but still keep it searchable with low latency responses the archive needs to handle the data itself and the associated metadata differently, in terms of such as storage media and aggregations. For example, the data itself might be stored on media optimized for low-energy cost where access time has a minor subordinate role.

Whereas metadata needs to be placed on low-latency media to speed up discovery operations on it.

Also managing metadata within relational databases would not be efficient enough as they are mostly bound to a fixed schema. Whereas user-metadata can be highly schema-less over time as nobody can predict the structure or types of metadata for future data ingests.

New workflows are needed for upcoming requirements

In near future new guidelines for good scientific practises such as open access and the preservation of scientific data for at least 10 years become more important and even mandatory. Both instruments aiming to improve quality and traceability of scientific publications.

As established practices cannot fully cope with future demands, new workflows need to be established. We think of a possible use case like described next.

During an experiment a scientist will create various types of data such as raw data, derived data, personal logbooks, plots and vary kinds of paper. For a publication all relevant data need to be aggregated and bundled in a container-like manner (Figure 1).

The more metadata the scientist will add to the data and to the container the easier it gets to index the content and to make it discoverable for other scientists or the creator.

The container then will be ingested into an archive combined with a policy attached to it (Figure 2 - 1a). As reference for a publication, one or many universally unique identifiers (UUIDs) will be created (Figure 2 – 1b).

These identifiers will be used to reference the work in journals, papers or other systems. Persons interested in the data would therefore be able to present UUIDs to the archive which stages all

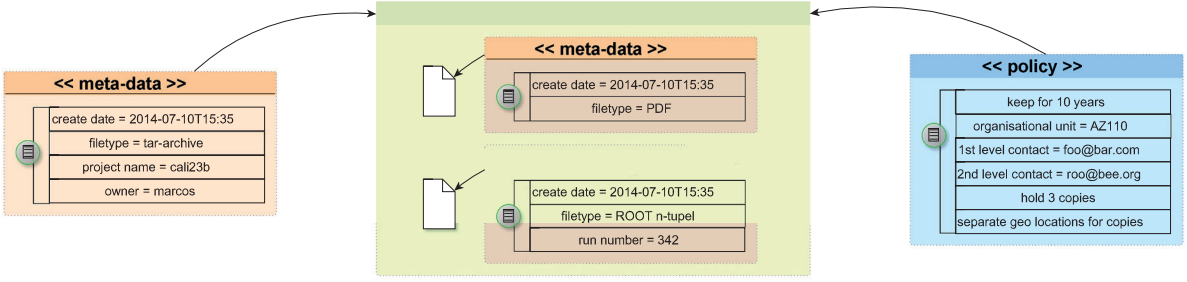

***Figure 1** A Container enhanced by Metadata.*

relevant data of this specific publication where they have access to.

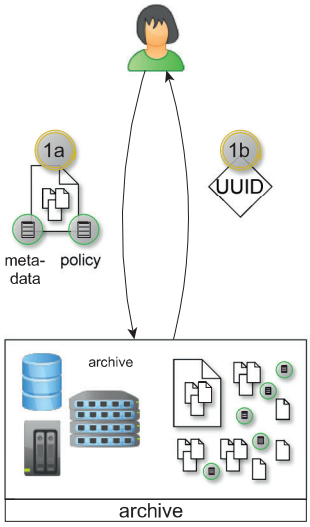

Figure 2 *Ingest Data.*

Big Data Challenges

Imagine after feeding an archive for a couple of years with petabytes of data you need to retrieve specific information out of it. Unfortunately you hardly remember any details about it. Therefore, as an integral part, an archive should be able to provide proper discovering tools to let you search and browse for wanted data.

Furthermore, somebody else might want to retrieve a subset of your data. In case access to data needs to be restricted or follows a specific open access policy the archive needs to have a mechanism in place to proper handle access rights and ownership information.

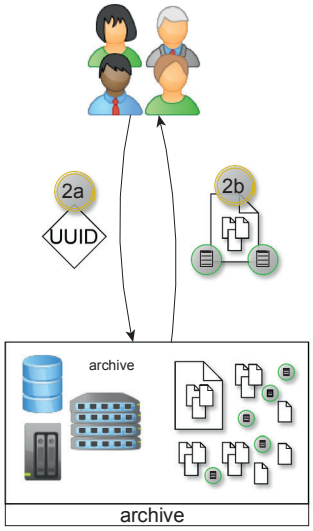

Figure 3 *Retrieving Data by UUID.*

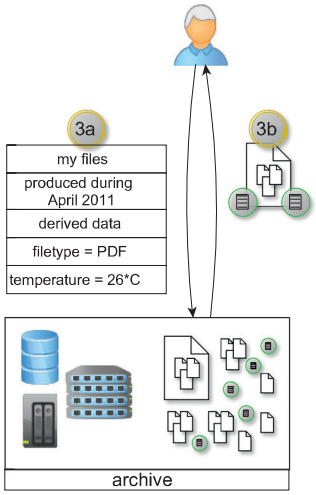

Figure 4 *Retrieving Data by Metadata.*

How to Access an Archive

From a user's perspective an archive must offer suitable interfaces to ingest data and to get it back.

An interface for ingesting must enable the user to define a set of files, to define metadata and to assign policies to it. The data doesn't necessarily needs to be local but can also be stored in a remote place where the archive has access to. For retrieving the targeted data-set the user presents one or many UUIDs to the archive which results in a direct download (Figure 3). When the data-set exceeds a critical size where a direct download would take too long or would just consume too much space on the users local file-system a third party transfer needed to be activated. The user can tell the system to which supported storage location the data-set will be staged to and also will be informed as soon as the transfer has been completed.

Furthermore, should UUIDs be unknown for the desired data-set, the archive must offers alternatives to search and browser for data instead. So the user will be able to execute interactive queries based on the underlying metadata (Figure 4).

Conclusion

Most of the described aspects results in workflows and requirements which cannot be met by today's software solutions and hardware products. Therefore new concepts needed to be designed and practically tested to gain more experience, such as on how to handle metadata efficiently and how scientists can be easily handle the vast amount of data without remembering any little detail about every experiment.

Petra III - Data Taking and Analysis

Marco Strutz, Steve Aplin - DESY

Petra III

With a circumference of 2.3 km, PETRA III at DESY (see Figure 1) is the biggest and most brilliant synchrotron light source in the world. Since the end of 2012 all 14 beam lines are available for users. New beam lines will be built and go into production for users data taking. This article sheds light on various kinds of problem domains regarding:

- data taking for Photon Science,
- next generation detectors and
- why well established workflows needs to be adapted or rethought to handle upcoming data rates.

Figure 1 *Aerial view of the almost 300 m long PETRA III experimental hall "Max von Laue" 2012.*

A Changing Landscape

After data taking, data was put on local commodity media by users, recently on USB3 hard-drives. As upcoming data exceeds by far the 1-disk-capacity it is not possible to use single hard disks anymore. Also, it it not possible to have a proper data management, access control or archiving mechanismn in place for such external media. Furthermore, data rates become higher outperforming specifications for transportable media devices. As a result traditional workflows will not work for future detectors. This effects the whole chain of data handling.

Data Taking since 2009

Up to now most of the data pipelining happens inside the Computer Center. Experiment PCs and offices desktop PCs are connected to the Computer Center by 1GE to 10GE. Data permissions and delegation for data being produced at the beam lines are handled by a dedicated *Data Portal*.

The data-processing chain starts with a detector for each beam line, producing many files per seconds with a specific file size.

Typical data-rates from the detector to the data storage were up to 175 MByte/sec (25 Images per second, each 7 MByte).

Next Generation Detectors

Eiger, PCO Edge and LAMBDA (see Figure 2) are next generation detectors differing in frame rate, data rate and the operating system there are managed by. As example, data rates of these detectors will be orders of magnitudes higher, so instead of having 175MByte per seconds per beam line one can expect like 10 GBytes.

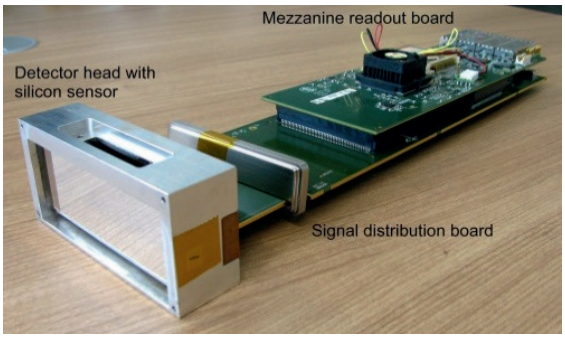

Figure 2 *LAMBDA, a next generation detector developed at DESY.*

As beam lines at Petra III are generally not bound to a specific detector the underlying data management need to be able to cope with the heterogeneity.

The Problem

The development of detectors at 3rd generation light sources currently outpacing experimental method and data acquisition. Single clients will produce 0.5 GBytes/sec and the next generation is already at frame rates of 2 kHz for 4MB files. For 30 beam lines they provide possible aggregated peak rates of up to an average of 50 GBytes/sec. Also, measurements last from a few hours to a few days resulting in many single data sets up to tens of TBs each. From next generation detectors we also expect multi GBytes/sec spread over many 10GE connections.

Furthermore there is a very dynamic experimental setup with inherent burst nature and a very heterogeneous environment regarding technology, social context and requirements.

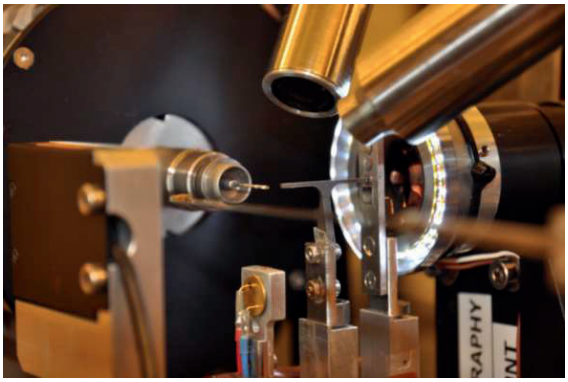

Figure 3 *Detailed view of P11 crystallography experiment with goniometer.*

Phase Change

On the one hand, we need to rethink previous common practices where storage systems were used as FIFOs, where faster and faster disks could have solved many speed constraints or where file systems were used as central data entry point.

On the other hand there are more and more facilities for light sources with same detectors in place targeting the same user communities challenging same or similar issues regarding upcoming data taking.

Data Handling for Experiments

Four major steps are involved in a typical data processing chain for Photon Science experiments.

1) Before: "Planning the next experiment." Processing of older datasets combined with simulations which will be run at the users institution.
2) Immediate: "Is the measurement setup and the data acquisition producing useful data?"
   During experiments "real time" data processing takes place, also analysis and visualisation to make experimental decisions.
3) Short term: "Does the data I am taking help to answer my scientific question? Before the user goes home data reduction and processing is performed. Users go home with clean data free of instrument artefacts. This step is preliminary for the data analysis which might helpful, but may require significant processing power and know-how.
4) Long term: "Does the data I am taking help to answer my scientific question? At last, users do a detailed analysis from their institution, turning data to information. This incorporated results from other techniques.

Target Data Flow

To cover the described aspects new approaches are being developed. One of the main focuses is to persist all data produced by a camera by all means within a realistic time frame.

Before data will arrive in central storage it will be buffered in a dedicated, persistent cache. This cache also smoothing peak rates and burst pattern of the detectors data streams and enables the central storage to receive at a steady input rate (which is below the camera effective data rate).

Additionally, visualization and near real-time analysis must not block the data stream to the central storage.

Further effort is also put into developing a data distribution solution based on messaging systems instead of writing detectors data directly into a file system.

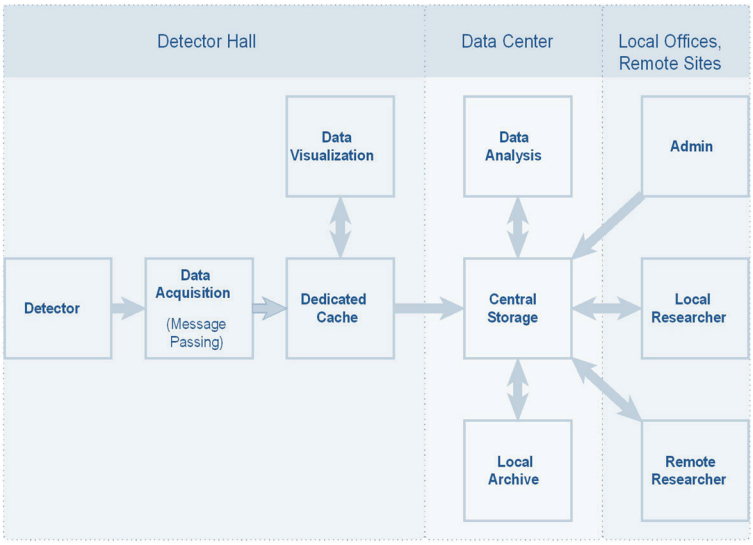

Figure 4 *Projected Data Flow.*

Fast Analysis of Image Stacks in Optical Nanoscopy

Michael Hausmann, Jürgen Hesser, Nick Kepper - Heidelberg University
Ajinkya Prabhune - Heidelberg University, KIT

Scientific Motivation

Light microscopy is a routine imaging technique in biological and medical research and diagnosis. Although nowadays instrumentation has made substantial progress concerning imaging quality and speed, there is still a gap in resolution between light microscopy (~200 nm) and electron micro-scopy (~10 nm). This so far missing scale range would however open new insights into the nano-cosmos of a cell and its sub-cellular structures [9].

Localization nanoscopy, being a candidate to fill this gap, is a novel technique overcoming resolution limits due to diffraction. During the last decade several setups have been developed and used to answer interesting and challenging questions in the field of cellular biology and molecular biomedicine [11].

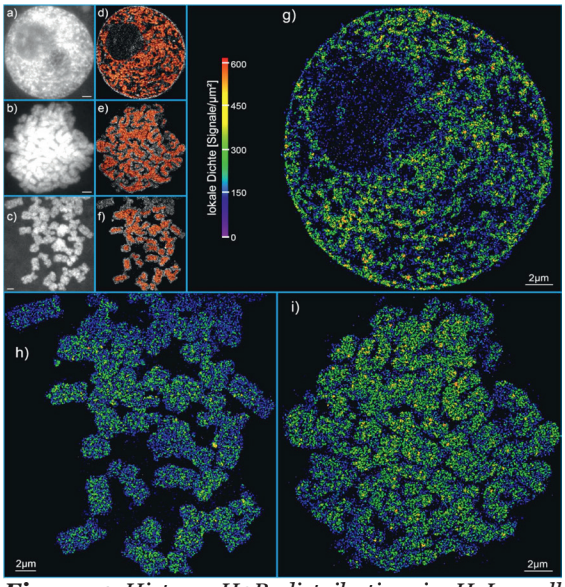

Figure 1 *Histone H2B distribution in HeLa cell nuclei and (pro-)metaphase chromosomes* [9].

Instrumentation

For localization nanoscopy standard microscopic optics and fast imaging systems are required. The principle of the so far developed techniques depends on optical isolation and separation of individual dye molecules by their spectral signature. The embodiment (SPDM = Spectral Position Determination Microscopy) used in our collaboration makes use of dye molecules for specific labeling of cellular sub-structures that are able to undergo so called reversible photo-bleaching which results in stochastic molecular blinking. Taking a huge time stack of images (~1000 frames) the switch off/on of each molecule can be detected and the molecular coordinates can be determined precisely (in the range of nm). Hence, distances between dye molecules can be calculated in the ten nm range and thus sub-cellular structures can be visualized and measured also in 3D conserved cells or even under vital conditions.

Examples

In the following two typical examples will be explained: In Figure 1 an example of a cell nucleus and (pro-) metaphase chromosomes are shown. a) – c) show the wide field microscopic images; d) – f) present the merged images from the time stack of SPDM displaying thousands of individual molecules by a color dot. In g) – i) these images are enlarged and coded according to the numbers of next neighbors so that structural information can be elucidated [12]. Such chromatin 2D/3D nano-structures are of importance to understand chromatin rearrangements during repair processes of DNA after exposure to ionizing radiation. This information is used to create and validate a consistent architectural model in the field of radiobiology.

On the left of the Figure 2 an overlay of a standard wide-field image (green) is shown. The right image in Figure 2 shows the result of localization imaging (red) of a membrane section of a breast cancer cell. This is where the Human Epidermal growth factor Receptor 2 (Her2/neu, a typical breast cancer marker) is specifically labeled. The right image shows the result of localization imaging which is obtained from a time series of 1000 image frames (979 x 816 pixels, 150 ms per image). Here, each point represents a single fluorochrome respectively antibody attached to a receptor molecule. The wide-field image does not allow the identification of any detailed nano-structural information about the spatial arrangement of the antibodies/receptors. This shortcoming of wide-field image is overcome by the localization image, which reveals details of the formation of receptor clusters or linear arrangements of receptors (inserts) which can be correlated to dimerization induced functional activity [13]. Such analyses help to elucidate mechanisms of breast cancer therapy using antibody treatment (e.g. Herceptin®).

These examples indicate the huge progress going along with localization nanoscopy. However the volume of the data is drastically increasing by orders of magnitudes requiring novel approaches of managing, archiving and analyzing.

Technologies

From the examples shown above we assume the digital volume of one cell nucleus of about 20 µm diameter with a resolution of approximately 10 nm is about 32 GB per channel of color. In larger screening experiments the limit of one PB data volume is thus reached easily. For the highly sensitive analyses and structure elucidation, very complex and highly variable algorithms have to be used to avoid artifacts and to find out structural re-arrangements. This includes iterative variation based denoising and deblurring techniques. Still the data is saved and worked on in an ad hoc manner, which with serial computation systems leads to extremely long processing times and a limitation of the selectable volume size due to limitations in the computer memory. The data rate created by a nanoscope is in the range of up to GB/s depending on the size of the detected region of interest and the dimensionality (2D/3D) required for scientific investigations.

Actual algorithms and techniques have been developed for a PC basis without usage of techniques for parallelization. This strongly limits the handling of large data sets as being necessary in biological research and medical diagnosis especially if a serious significance of statistics is required (i.e. if a large series of cells have to be evaluated). Here, we develop a pipeline for parallel data analysis.

Variation based methods need, with parallel analysis of the data, a synchronous update of all analyzed regions, which will be realized with message passing. The access to the data has to be self-explaining for the user and has to fulfill the rules for storage of the DFG for several years.

Nanoscopy Reference Data Archive

Light optical nanoscopy produces datasets of up to 200 TB. As nanoscopy is a novel methodology the archiving, analysis, access and handling of these extreme large datasets is a new challenge. For that it is necessary to dynamically annotate the datasets for experts to share and to compare their finding. Hence there is a need to build a Reference Data Archive which will enable the scientific research community to store and access extreme large datasets in a repository, annotate the datasets (reference data is important for disseminating knowledge, thus needs to be maintained if new insights about the data appear), share the annotated datasets, and analyze the datasets interactively.

In the Data Life Cycle Lab “Key Technologies” data and microscopy experts jointly developed a repository for registering and storing extreme large datasets, a client for automatic ingest and access, web based tool for monitoring and accessing the datasets, and a basic representation of ingested data.

Future components will include a content metadata schema to enable data discovery and reuse, automated metadata extraction, an annotation framework, and access to high performance computing infrastructure for large datasets analysis.

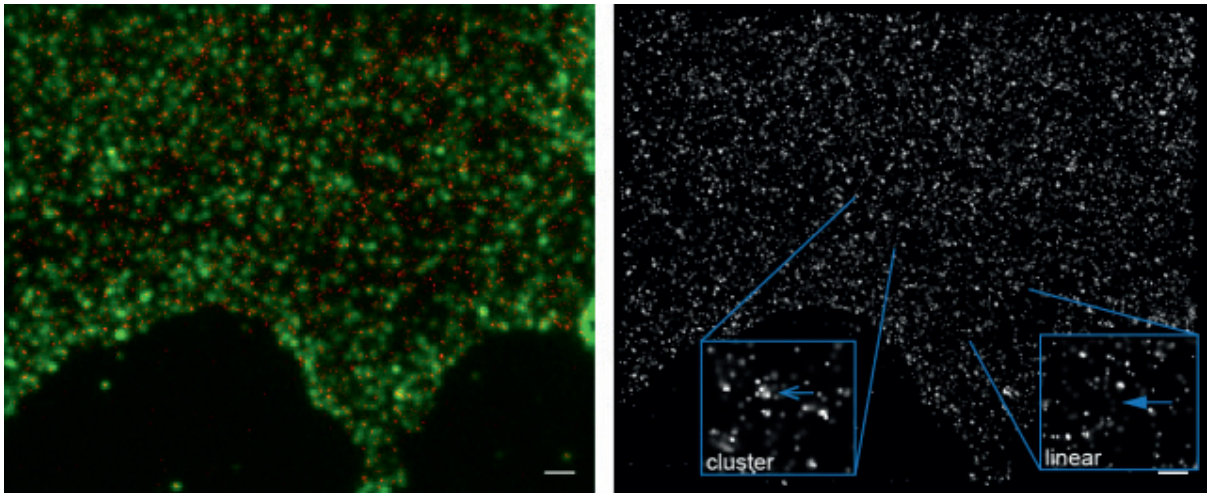

Figure 2 *Image section of the membrane of a breast cancer cell after specific labeling of the Her2-receptors by means of fluorescence labeled antibodies. (courtesy J. Neumann, Kirchhoff-Institute for Physics, University of Heidelberg).*

Dynamic Storage Federations with Standard Protocols

Paul Millar, Patrick Fuhrmann, Karsten Schwank - DESY
Fabrizio Furano - CERN

In many current scientific communities the requirements for storing data are very strong and sophisticated. Both Data Security and Data Safety are crucial and have been addressed by several software-packages from the scientific community. Data Security is granted through powerful authentication and authorization methods and Data Safety has been addressed on different levels by means of redundant storage. This means, many of the data files are available from different storage endpoints at the same time. The problem that persists for the user is to know how to access them.

To address this issue, different experiments around the *Large Hadron Collider (LHC)* have developed their own data federations, allowing users to access data using central entry points, while being transparently redirected to the actual location of the data. However those federations are based on limited sets of products often entangled with their own sets of proprietary data transfer protocols.

To improve the situation of large numbers of incompatible systems and protocols, middleware providers from all over Europe have cooperated in the *European Middleware Initiative (EMI)* and have put additional effort into the development of endpoints supporting standard file transfer protocols, like *HTTP/WebDAV* and *NFS4.1* in their systems, thus allowing the use of widely available clients (e.g. web browsers) to access data.

Federating Storage

The goal of the *Dynamic Storage Federations*-project [14] was to make use of this advancement towards standards and to create a federation engine that can act as a smart central entry point to federations of storage endpoints. It allows file listing and metadata operations through a real-time consolidated global namespace, while the clients read data directly from the endpoints. It can integrate the most widespread data storage elements of the *Grid* software stack, like *DPM* and *dCache* [15; 16], as well as whole clusters of storage elements and even commercial cloud storage providers. To optimize data access it makes use of existing replicas, by providing smart redirection, i.e., it automatically picks the most suitable location of the data to redirect the clients to, based on, for example, their geo-location. Since every file's location is checked upon request it is also suited for loosely coupled federations with endpoints dynamically joining and leaving.

In large international scientific communities, data is usually automatically distributed by sophisticated frameworks that take care of keeping replicas and of optimizing storage space and data access. The users usually don't know where the data is stored and they shouldn't have to. Instead they should be able to access all data through a single entry point: The *Generic Redirector*.

Let us assume the following simple scenario (depicted in Figure 1): File 1 is available on a site with Endpoint A, File 3 is available on a site with Endpoint B and File 2 is available on both sites and through both endpoints.

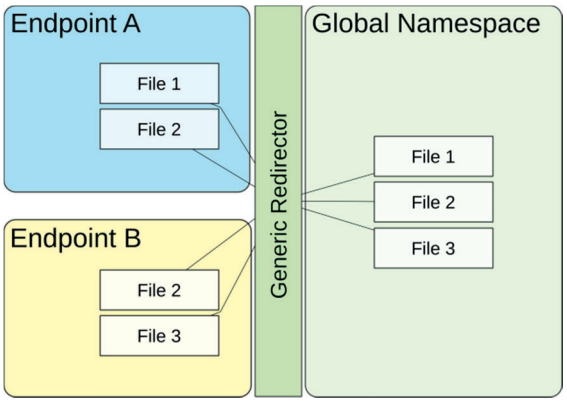

Figure 1 *The Generic Redirector presents the consolidated namespace to the user.*

If a user requests File 1, the *Generic Redirector* will redirect the request to Endpoint A, if she requests File 3, the request will get redirected to Endpoint B and if File 2 is requested, the *Generic Redirector* will redirect the request to the endpoint that can serve the file fastest.

The Generic Redirector

The *Generic Redirector* is the core component of the federation engine. It exposes an API for namespace operations including file metadata and directory listing information. Typically it will be loaded by some front end system, for example as a plug-in for the *DMLite* file catalogue system that can run inside an Apache web-server.

The Redirector has a plug-in interface enabling it to integrate with different types of endpoints. Currently available plug-ins support *HTTP/WebDAV* and *DMLite*. The latter allows native connections to *LFC* databases and *Hadoop Distributed File System (HDFS)* storage clusters.

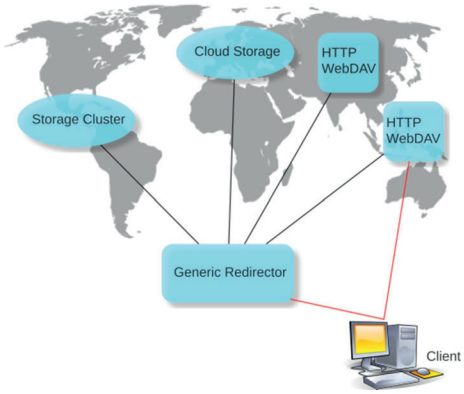

Figure 2 *The Generic Redirector redirects a file request of the Client to an HTTP/WebDAV-endpoint.*

Internally the Redirector acts as a sophisticated handler of parallel requests for metadata information. Upon a request the Redirector will trigger all activated plugins with the query and wait for their replies. As soon as sufficient information has been returned by the plugins the client is notified. For even faster response times, an effective in-memory cache is being used.

Use cases

The following three generic use cases for the *Storage Federation Engine* are supposed to give clear examples of the features that the system provides and that have already been proven to work in different deployments. However, neither do these use cases exclude each other, nor is the system limited to these use cases.

Use case #1: run application on clouds

Canada's Advanced Resource and Innovation Network (*canarie*) want to be able to run data-intensive applications, like batch services, software distribution and storage federation, on distributed clouds using standard protocols. They use the *Federated Storage Engine* to federate multiple Storage Elements located on several sites in North America.

Use case #2: add third-party storage farms

A company wants to offer their users a simple way to access files. Some files are hosted by the company, but additional space may need to be bought from commercial cloud storage providers. The company integrates the *Generic Redirector* into their web portal and allow the users to access their files using a standard web-browser. The requests to the files are transparently redirected to the different storage endpoints. This allows the company to change storage providers without interrupting the service.

Use case #3: federate storage of several sites

A company has several branches all over the world. User data is synchronized regularly between those branches, but it may take new files a couple of days to be distributed to all branches. The users often access their data from different locations (e.g., airports, hotels) and need to be able to have fast access to their data independent of their location. The company sets up a central site hosting the *Generic Redirector*. This allows the users to access their data in a normal fashion, while transparently being redirected to the optimal replica of some file, dependent on their current location.

Summary

The presented *Federated Storage Engine* is an efficient, persistency-free, scalable and easily manageable approach to federate remote storage and metadata endpoints. It is a big step forward towards open standards, simplification of data access in storage federations and to make powerful mechanisms and tools from the *High-Energy-Physics*-Community, like *DPM* and *dCache*, available to users in other contexts.

The plug-in interface and the possibility to integrate the *Generic Redirector* into web services already existing at sites, makes it highly adoptable to a multitude of use cases.

The *Federated Storage Engine* is ready for production. It can currently be installed from our website at *CERN*. In the near future we will provide downloads and documentation of the *Federated Storage Engine* through *EPEL* and a dedicated website.

A demonstration of the *Federated Storage Engine* can be found at [17].

Federated AAI: Enabling Collaboration

Paul Millar, Patrick Fuhrmann - DESY
Dennis Klein - GSI
Arsen Hayrapetyan, Marcus Hardt - KIT

Introduction

Progress in experimental research has often gone hand-in-hand with technological advances. With more advanced equipment, more detailed investigations are possible: either through improvements in automation or by providing a higher level of detail.

One effect of this progress is that an ever increasing amount of data is available for analysis. Where previously researchers could have their personal copy of the data (e.g., stored on their workstation), now sufficient data is collected that it becomes more economical that it be stored on dedicated equipment, from which authorised users can access. Also, with sufficiently large amounts of data, dedicated computing resources are required to process it; for example, analysis may use a High-Throughput Computing (HTC) or High-Performance Computing (HPC) cluster.

While this rich source of data is a boon for researchers, it places new burdens in how the data and the analysis is handled. One example is when two or more institutes wish to collaborate. To do this, they must allow members of the collaboration access to the shared resources: access to the data and access to any shared analysis facilities.

There are many challenges in providing access to members of a collaboration spanning many institutes. In this article we describe just one part: that of how users identify themselves.

Often an institute will have a common authentication framework, so that a user can authenticate with the same name and password when accessing any service at that institute.

One solution to this is for all users to have an account on all shared resources. While functional, this approach has several disadvantages:
The users face having to remember their account name (either because institutes have different naming policies or due to name clashes), best practice says that passwords should be different for each account. As the number of members in the collaboration increases, people joining or leaving quickly becomes a burden for all administrators in the collaboration.

A better approach is for the user to have a single username and password to remember and accounts are created automatically. When a user wants to use a service for the first time, she sends her username and password to her home institute. If the information is correct, a token is returned that is automatically sent to the service. Provided the service and home institute trust each other, the user is logged in. This is the basis of federated identity.

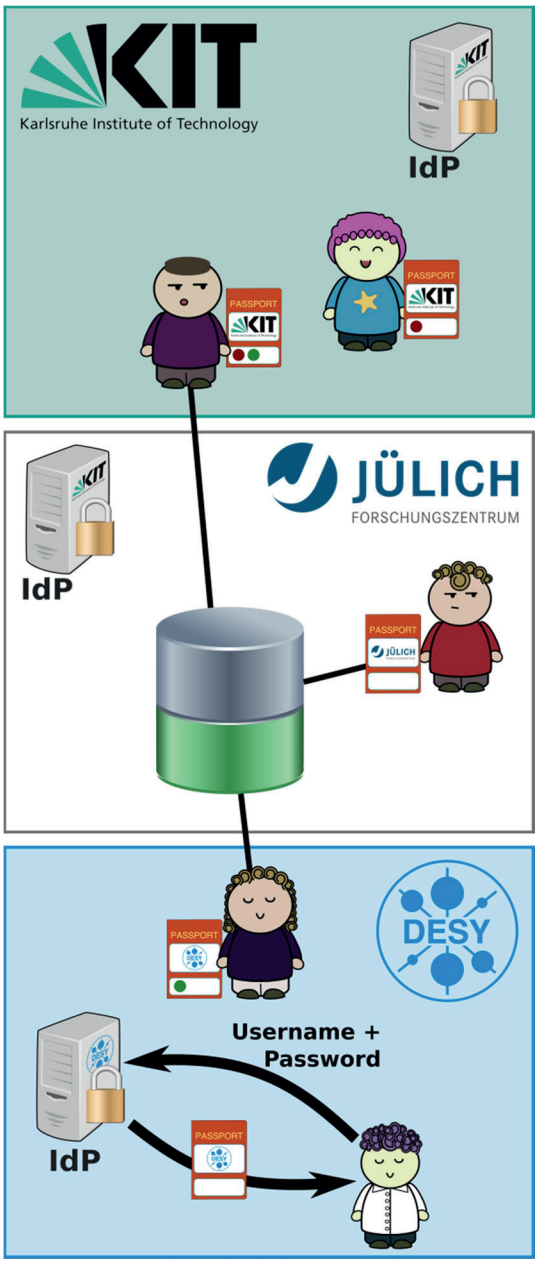

Figure 1 *Using home-institute credentials for accessing distributed resources.*

While federated identity helps in forming collaborations, another key problem is knowing who should be allowed to use a service. The decision can be made simpler if the service is told (by a

trustworthy source!) of which groups is the user a member. If she is a member of the collaboration group, then the service should grant access. While this service could be run at any site, providing a centralised group-membership service facilitates creation of new collaborations: just create a new group. Another benefit of a central group server is that members can join or leave the collaboration at the discretion of the manager.

There are several federated identity systems currently in place; for example, X.509, OpenID OAuth and SAML all allow users to “log in” to different services with the same credential. Each system has strengths and weaknesses: X.509 provides excellent integration with data access but poor adoption; OpenID and OAuth have good adoption but (to a large extent) are limited to web-based activity.

We chose SAML as a technology basis for the work, as it is widely deployed and already covers many use cases; however, there are aspects that make SAML challenging:

- Using SAML without a web-based isn't widely available.
- No commonly deployed storage service accepts SAML authentication: X.509 is the current standard.
- Group-membership assertion services are not commonly used in SAML.

High Level Objectives

The high level objective of LSDMA in the context of AAI is to provide software, services and support when a research community wishes to share resources between different facilities: to collaborate. To achieve this, users authenticate with their home institute (only one username and password) and membership of groups comes from a centrally run group server. Services allow access, either based on individual identity or from the user's membership of groups, shown as green and red dots in Figure 2.

There are several “missing pieces” that need to be filled to achieve this.

Integration with web portals

We assume that the services that should be shared are not created for this project, but will already exist and be in active use. This might be web-portals that allow users to create analysis workflows or to query a database of images. To allow sites to adopt LSDMA solutions, we evaluate the problems associated and give practical advice on how they may be solved. We are developing expertise in this process, using the LSDMA wiki as a proving ground.

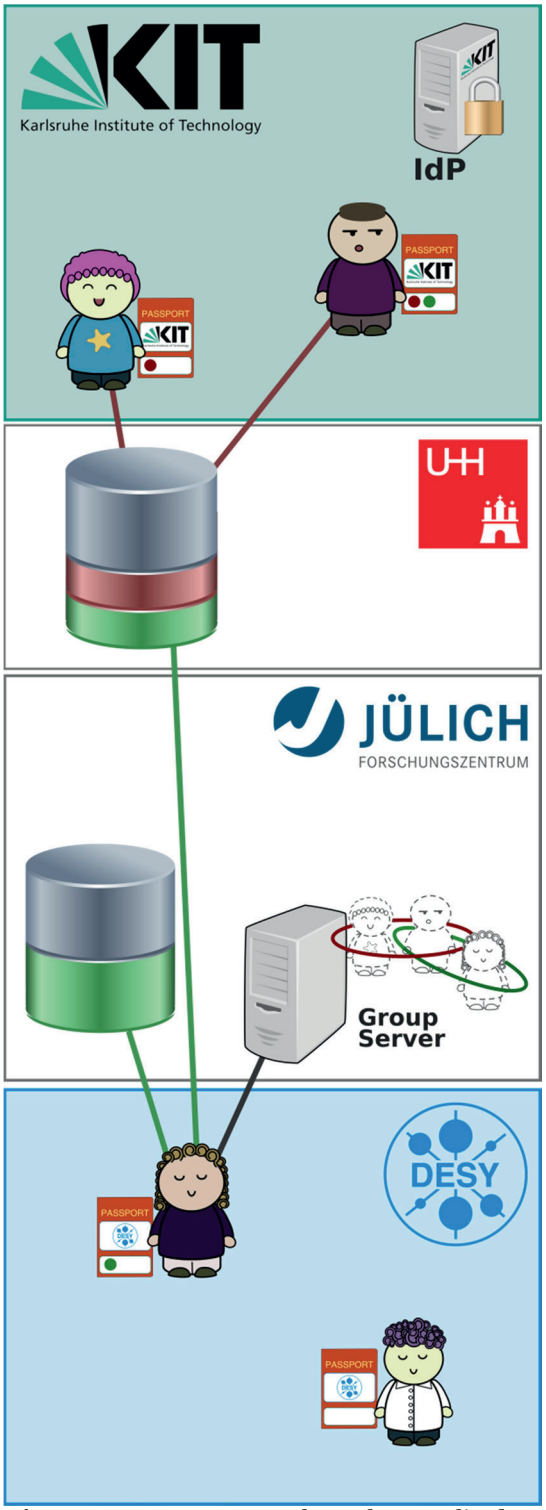

Figure 2 *Resource access through centralized group management.*

Big Data Access

In contrast to web-based access, large-scale data access is mostly achieved using X.509 credentials. Since almost all data facilities cannot make use of SAML-based credentials, the user must obtain an X.509 certificate before they can store or retrieve data.

LSDMA is investigating how to allow users to authenticate using SAML and automatically obtain a short-lived (typically 24 hours) X.509 credential. This

also involves querying a service that provides group-membership information. This information is provided as an attribute certificate, which is embedded within the short-lived X.509 certificate. The resulting X.509 certificate allows the user to upload, download and manage their stored data on all existing storage systems that support X.509 authentication. As obtaining the X.509 certificate happens automatically, the user is oblivious that this is happening.

LSMDA Data Management and Data Transfer
Moving data between different sites is a common requirement: analysis often requires data to be transferred. Control of such transfers requires access to both the source and destination storage systems, which the client authenticates via X.509.

Transferring large amounts of data between sites can involve handling timeouts, retries and optimal network tuning. Specialist websites, such as GlobusOnline, exist to allow non-experts to transfer large amounts of data with ease. However, this presents an extra challenge: how to upload a freshly created X.509 credential to some external web portal so that portal can manage the transfers.
To achieve this, LSDMA is running a demo service that allows a web-portal to receive the X.509 credential it needs to transfer the data, which will be expanded to support all users in the DFN.

Group Management
Group membership was mentioned as a key element in making the authorisation problem tractable: a centrally run group-server is needed so collaboration can manage its user list.

LSDMA has been evaluating different solutions. A very promising candidate is the HPC UNITY group management service that remembers which groups a particular user is a member. A web interface makes managing a group easy. The user's membership is then either provided directly (either as SAML assertion or X.509 certificate) or may be queried directly by the service.

Use case #1: work-flow engine
Researchers of a particular genomic field investigate active areas mostly through a web portal. Different levels of access are given to different users, based on who they are. The general public can use pre-defined queries against limited data-sets; ordinary members of the collaboration can use parameterised queries against different data-sets; power users can make arbitrary queries against any dataset. The researchers want to expand the set of people to include the institutes with which they collaborate.

This use case makes use of web-based SAML authentication and the group-membership service.

Use case #2: access to data
Data taken from a telescope is embargoed for a fixed period before becoming publically available; this allows collaboration members access to the data before non-members. Therefore, to access embargoed data, a user must prove they are a member of the collaboration. As the collaboration involves many different institutes, the list of members is changing often. A web portal allows the users to browse the fresh data and direct access is available via X.509-authenticated FTP.

This use case makes use of web-based and non-web-based SAML authentication, the group-membership service and X.509 credential-translation.

Use case #3: moving data
An HPC centre and human genomic project want to team up to look for a possible cause for an illness that is believed to have a strong genetic component. Researchers must transfer large amounts of genomic data to the HTC centre so that it can be processed on the supercomputer.

This use case makes use of web-based and non-web-based SAML authentication, the group-membership service, the X.509 credential-translation service and the data transfer service.

Summary
LSDMA is developing technologies, gaining experience and deploying services to allow institutes to share access to data and computing resources becomes simple, enabling them to focus on their research.

Imaging in Human Brain Project Using UNICORE Based Workflows

André Giesler - FZ Jülich

Understanding the anatomical structure of the human brain on the level of single nerve fibers is one of the most challenging tasks in neuroscience nowadays. In order to understand the connectivity of brain regions (affecting the brain function) on the one hand and to study neurodegenerative diseases on the other hand, a detailed three-dimensional map of nerve fibers has to be created.

The Human Brain Project (HBP) uses recent imaging techniques in post-mortem studies to derive patterns of connectivity between brain regions and to identify fibre tracts connecting layers and cells within brain regions. This data is essential for modelling the large-scale structural architecture of the brain and to verify data from *in vivo* experiments. One state of the art technique applied to histological sections of post-mortem brains is Polarized Light Imaging (PLI) which allows the study of brain regions with a resolution at sub-millimetre scale (Figure 1). It is based on an optical property referred to as birefringence of myelin which surrounds the axons of nerve fibers. Therefore about 1500 slices, each 70 micron thick, of the post-mortem brain are imaged with a microscopic device using polarized light.

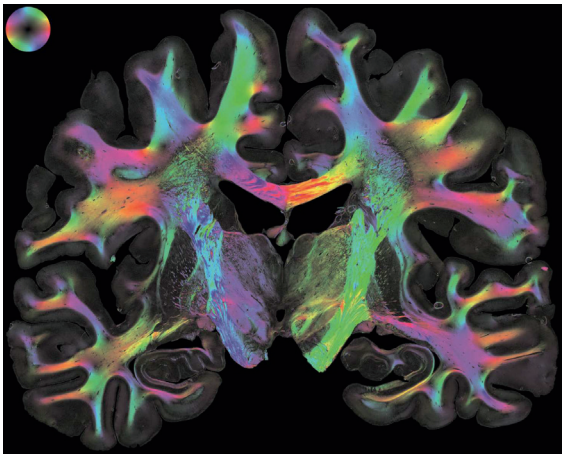

Figure 1 *Cross-section image of a human brain after PLI processing.*

The images of brain slices are processed with a chain of tools for cleaning, alignment, segmentation, and recognition. These tools have been integrated in a UNICORE workflow (Figure 2), exploiting many of the workflow system features, such as control structures and human interaction. Prior to the introduction of the UNICORE workflow system, the tools involved were run manually by their respective developers. Thus, once one step in the process was finished, the developer of the next tool in the chain would retrieve the image data and run his tools on the output of the former. Another difficulty is that the tools are located on distributed resources. Thus, the intermediate results must be transferred by the users between different storages and file systems. This manual approach led to delays in the entire process.

The introduction of the UNICORE workflow system for this particular use case resulted in several benefits. First of all, the results are easier to reproduce now, as fewer manual steps are involved. Secondly, the processing time of the entire workflow could be reduced to hours rather than weeks, because of the almost fully automated data workflow. The amount of data for a single brain slice is on the order of 1TB, with intermediate results at the same scale. The performant UFTP file transfer protocol is used in the workflow system to move large files effectively between distributed storages. Lastly, only the automated approach will allow for the timely analysis of a large number of brain slices that are expected to be available in the near future.

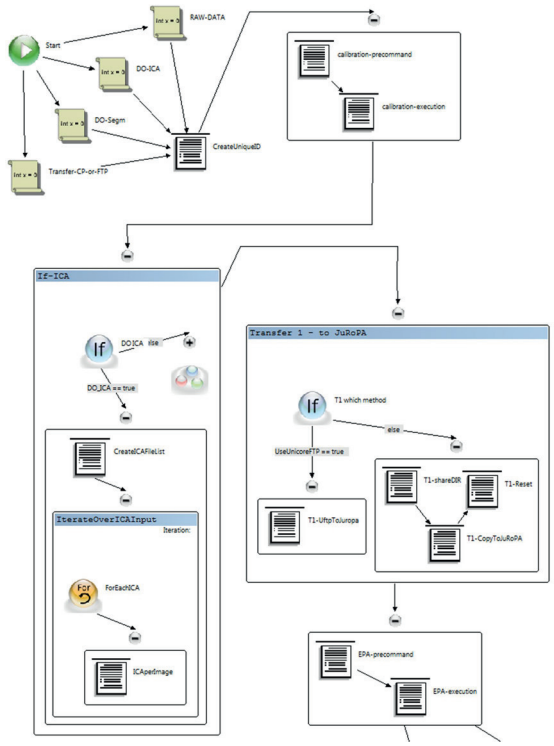

Figure 2 *Excerpt of the UNICORE Workflow integrating PLI tools and data flows.*

Real-time Response Framework Using MongoDB and 3D Visualisation

Parinaz Ameri, Marek Szuba - KIT

Introduction

Observation of Earth from space has become an important part of climate studies, with dedicated satellites and instruments providing measurements. As a consequence of both high number of observables monitored by satellite instruments and wide geospatial distribution of observation points, the volume of data in modern climatology has become considerable. For instance, the ESA Envisat satellite alone – which featured 9 Earth-observing instruments and orbited the Earth once every 100.16 minutes – during its 10 years of operation acquired over 1 PB of data. Furthermore, several Earth-based stations now produce huge amounts of data as well. Additionally, on top of dealing with large amounts data from one satellite at a time it is often useful to match results from two or more satellites with each other or with stations on Earth.

The goal of the project at hand is to develop a state-of-the-art framework facilitating management of satellite climate data. The framework is to gracefully handle high-volume data reads, and scale well as data from more satellites and instruments is added to the database. Last but not least, we aim to provide visualisation and basic analysis capabilities which could be used by scientists with minimal to no knowledge of the underlying infrastructure.

Storage

For storage purposes of this project we chose MongoDB – a document-based NoSQL database which stores data as structured key-value pairs. Input and output of documents is based on the JSON format, considerably simplifying the use of the database. This is particularly true in the context of Web applications, among which JSON is the de facto standard for data exchange, as well as object-oriented programming, in which case documents can naturally be represented and treated as objects.

For the handling of storage in a big-data project, it is very important to consider scalability of the system. MongoDB offers a horizontally scalable database solution with its *sharding* concept: each shard is a partition of the data that can be stored on different physical machines than any others. Information about the physical location of the data in each shard is kept in some redundant Config Servers and the routing process is done through *mongos*, a lightweight service which can be run on any system – even one that runs other cluster components. *mongos* is also responsible for balancing the load coming from clients (Figure 1).

In order to increase the performance of the application, queries to the database are parallelised. Unlike traditional databases, MongoDB fully supports parallel query handling. In addition, in order to provide fault-tolerant architecture each shard is set up as the primary node in a *replica set* of three nodes, where the other two are secondary nodes that can be used only for reading information from database.

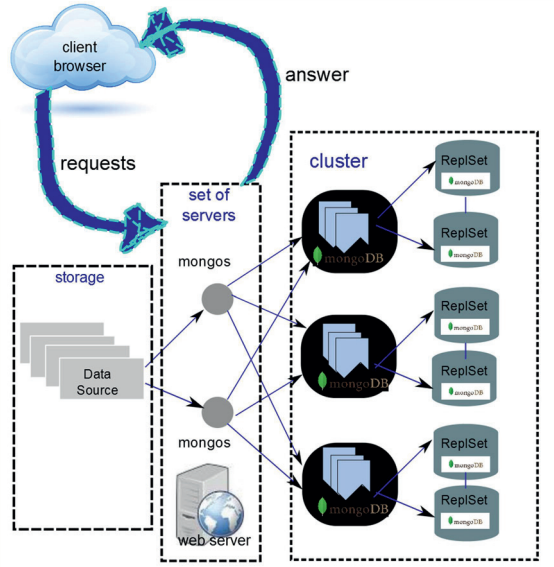

Figure 1 *Illustration of different components of the project.*

Finally, MongoDB offers a built-in index called *2dsphere* which is useful for indexing geospatial locations of satellite data information around the globe.

Input and Output

As a direct consequence of the nature of data at hand, our system is largely read-only. Writing is only necessary when data for new instruments is added to the system. Therefore, our input and output follow different designs and the framework has been optimised primarily for the latter.

The fact that there is no universal data structure for all of the satellites was a motive to take advantage of MongoDB schema-less design for storing data. Although there are sets of variables that climatologists might be interested in, the exact combination of variables chosen to be measured differs from device to device.

There are cases when as time passes one specific device might have different versions of an instrument (due to upgrades, changes of mode of operation, effects of aging or hardware failures and so on), or an instrument can measure different

observables (concentration of different gasses, for instance) at different times. The user might want to target different gases in different ways and using different instrument versions in each of their analysis.

For few satellites, there are some variables that are considered mandatory. The user cannot use the data of such devices without specifying a special value for these variables. On the other hand, there are also optional variables provided for some of the satellites that the user might freely choose to require or not. For on the ground stations, the user might want to filter the data taken by one specific station, or just simply use the data coming from all of the stations in one project.

As a result, the import of data into the database is handled by IT experts who carefully analyse all of these conditions, then plan and implement necessary extensions of document structure and import scripts. Import operations themselves are typically handled by Python scripts executed on machines with direct access to the database.

A different approach was employed in case of reading the data from the database as unlike input, it is primarily done by users who are not necessarily IT experts, have not got shell access to the database cluster and may or may not have MongoDB drivers installed on their local machines. In light of this, the primary read interface is a Web service which wraps database queries in a simple Representational State Transfer (REST) API. This API provides functions returning data as required by specific applications such as the visualisation interface described below, as well as a low-level MongoDB query interface which is useful e.g. for debugging purposes. In both cases the data is returned in JSON format over HTTP.

Our tool of choice for the REST Web service is Node.js, a server-side JavaScript platform whose event-driven, non-blocking I/O model allows it to gracefully handle even highly data-intensive applications. Its high performance aside, Node.js also features an extensive library of modules usable in one's application – including both a MongoDB driver and sophisticated Web-application frameworks such as Express.

Visualisation

In order to make our visualisation interface portable it was designed as a JavaScript Web application, which should automatically make it compatible with any platform capable of running a recent version of a standard-compliant Web browser such as Mozilla Firefox or Google Chrome. The application submits REST commands to the server, fetches requested data and renders it in 3D in the user's browser superimposed on an image of Earth. Its current version can display orbital paths of the satellite as well as apply user-provided criteria to cloud-index measurements from the database in order to calculate and present the altitude of clouds on given days (Figure 2).

We display acquired data using the WebGL Globe – an open platform for visualisation of geographic data which was developed by the Google Data Arts Team. This in turn, as the name suggests, employs the WebGL 3D-graphics JavaScript API, which allows it to take advantage of locally available GPU power. Preliminary performance tests have shown the current version of the visualisation interface to perform well even on relatively weak hardware such as integrated graphics chipsets found in modern laptop computers.

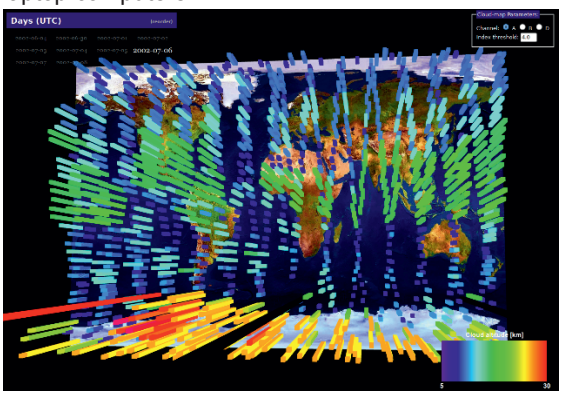

Figure 2 *Visualisation front-end showing measured cloud altitude.*

As WebGL uses the HTML5 canvas element, we can take advantage of other features provided by this version of HTML. In particular, other canvas elements are used to display dynamically generated two-dimensional graphics such as heat-map legends, and the use of Web Storage is being considered for the caching of fetched and/or pre-computed data sets.

Summary and Outlook

A framework has been developed for storage and visualisation of satellite climate data. The framework uses a MongoDB cluster as its storage back-end, several Python scripts for insertion of data, a RESTful Web service for queries and a WebGL-based JavaScript application for visualisation of orbital paths and cloud altitude.

In the near future we expect to extend the database and the input system to include more data. This requires a careful design for structure of data and generating and structuring metadata to store in the database. We are also working on a Node.js Web service to provide the first production version of the REST API. Last but not least, the visualisation interface is being modified to support further data types as well as to make it more flexible and user-friendly.

Reducing Energy Consumption of Large-Scale Storage Systems

Michael Kuhn, Konstantinos Chasapis, Manuel F. Dolz – University of Hamburg

Due to the increasing electricity footprints, energy used for storage represents an important portion of the total cost of ownership (TCO).

Processor speed and disk capacity have roughly increased by factors of 500 and 100 every 10 years, respectively. The speed of disks, however, grows more slowly: We have observed a 400-fold increase of throughput over the last 25 years for hard disk drives (HDDs). Even newer technologies such as solid-state drives (SSDs) only boost the speedup to 1,200. In comparison, over the same period of time, the computational power increased by a factor of 1,000,000 for supercomputers due to increasing investments.

Moreover, the growth of disk capacity has recently also started to slow down. While the same is true for processor clock rate, this particular problem is being compensated for by growing numbers of increasingly cheap processor cores. Additional investment is required to keep up with the advancing processing power.
While this problem cannot be solved without major breakthroughs in hardware technology, it is necessary to use the storage hardware as efficiently as possible to alleviate its effects. The outcome of this is that it is not possible to increase storage speed and capacity by the same factor as processing power when keeping investment constant.

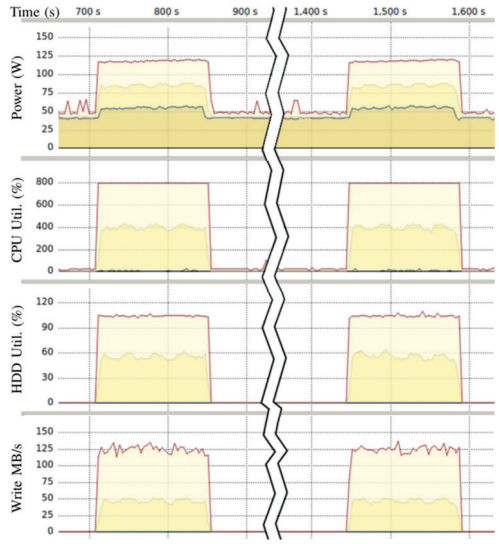

Figure 1 *Power-performance traces make it possible to correlate the storage servers' utilization with their power consumption.*

As current-generation CPUs provide ample performance for data processing, we provide a case for turning on compression by default to reduce the number of required storage devices and thus minimize the storage system's power consumption.

To analyze power and performance metrics of the storage servers, we employ an integrated framework that works in combination with VampirTrace and Vampir, which are profiling/tracing and visualization tools, respectively. In addition to the power measurements, we can also account for the storage servers' resource utilization values, such as CPU load, memory usage and storage device utilization. Finally, using the Vampir visualization tool, the power-performance traces can be easily analyzed through a series of plots and statistics (Figure 1).

Initial evaluations show that data compression in HPC storage servers can be used to save energy and improve I/O performance. On the one hand, less HDDs are required to store the same amount of data due to the compression. On the other hand, it is also possible to achieve a higher throughput by storing more data in the same amount of time; this is especially relevant in I/O-intensive cases. These two advantages lead to lower procurement and operational costs (Figure 2).

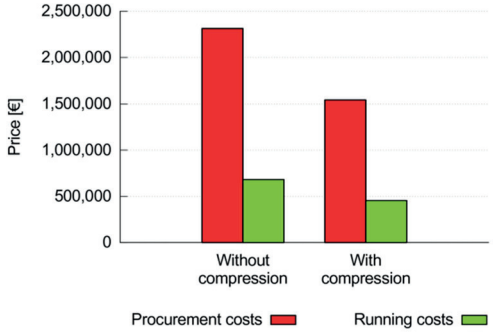

Figure 2 *Necessary investments for a storage system of 40 petabytes with and without compression.*

However, it is important to carefully choose compression algorithms due to their inherent CPU overhead as expensive algorithms will increase power consumption. We have identified lz4 as a suitable compression algorithm for scientific data and will use it for further analysis in the future.
Real world data-sets can achieve compression ratios of more than 1.5 without any significant increase in CPU utilization. We have observed a reduction of 7% in energy consumption for write-intensive applications.

STXXL 1.4.0 and Beyond

Timo Bingmann, Peter Sanders - KIT

In the age of Big Data, we are challenged with designing and developing applications that process large amounts of data efficiently and gain knowledge from the data or transform it into other representations. As large amounts of RAM are expensive, most data remains stored on hard disks. If the working set of an application exceeds the amount of available RAM, then algorithms that process data efficiently in external memory are required.

Efficient algorithms for external memory are characterized by the number of block I/O operations they require to process an input. While simple file access and databases queries are readily available, more sophisticated calculations, algorithms and advanced methods of computation with external memory are much tougher to implement. For example, highly efficient sorting in external memory, an efficiently priority queue, asynchronous I/O and overlapping of I/O and computation are data structures and acceleration techniques not easily accessible.

The STXXL (Standard Template Library of Extra Large Datasets) is a multi-platform C++ template library which provides many efficient external memory algorithms and data structures with a well-known interface. It was started in 2003 with the PhD thesis of Roman Dementiev, in which STXXL's layered design (see Figure 1) was developed, and many authors have since contributed and extended it with new functionality. The library is fully open-source and available under the liberal Boost Software License.

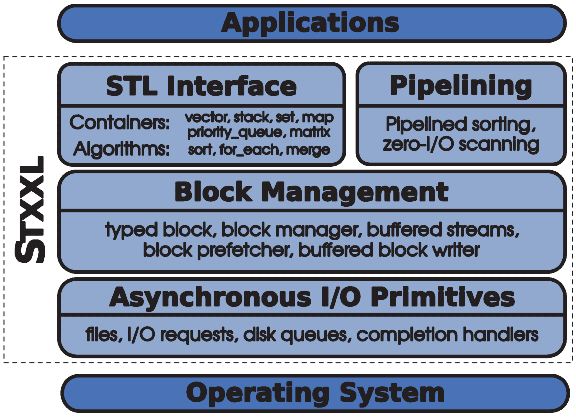

Figure 1 *Layer Diagram of STXXL.*

For the basic Standard Template Library (STL) data structures vector, stack, queue, priority_queue and map, the STXXL library provides a drop-in replacement which keeps its data in external memory, but has an interface that remains identical to the well-known STL data structure, as far as this is theoretically possible or desired. While vector, stack and queue are simple data structures, the priority_queue and map are sophisticated implementations with good theoretical performance guarantees.

Besides the easy to use STL-like interface, the STXXL provides highly engineered sorting implementations and support for pipelining algorithms to reduced constant factors in time and I/O volume. It is the only library transparently supporting multiple parallel disks and also optimizing parallel disk access during sorting.

In LSDMA the development of STXXL has continued and aims to create a reliable foundation for algorithms and applications to efficiently process large amounts of data. The release 1.4.0 of STXXL, published December 2013, was pivotal to bringing STXXL onto a modern software development stage.

In release 1.4.0 the whole source code hierarchy was reorganized according to modern standards, and the old build system was replaced with CMake for easy cross-platform compilation on Linux, Windows with Visual C++, and Mac OS X. However, the most important improvement was to greatly extend the documentation of STXXL, now providing extensive design documentation, tutorials and examples for most data structures. Furthermore, the version 1.4.0 incorporates the efficient external matrix operations developed by Raoul Steffen, and the skew3 suffix sorter as a complex real-world pipelining application.

In the next release 1.4.1, we plan to support the native Linux asynchronous I/O interface, which can take advantage of native command queueing (NCQ) on the hard disks, and to integrate asynchronous pipelined sorting with the aim of improving parallel sorting speed. Beyond these concrete improvements, we are currently doing research on a bulk-parallel priority queue which aims to exploit multi-core parallelism during bulk operations. With the availability of SSDs, which provide much higher I/O bandwidths than rational disks, more work will also be needed to further improve the throughput of external memory sorting and other algorithms.

Best Practices for Metadata Management in LSDMA

Richard Grunzke - TU Dresden
Volker Hartmann, Thomas Jejkal - KIT
Bernd Schuller - FZ Jülich

Big Data applications in science are producing huge amounts of data. The management of this data is a challenge as it is no longer feasible to access the data directly due to limited local storage capacities, limited transfer rates, and too many files. As an alternative action like searching for specific data will be based on metadata which describes the content of the data. Metadata is specific to communities and some communities have already defined their own standard (e.g.: OME, TEI) while other communities even lack metadata completely. No global standard exists which is usable for all communities. Dublin Core is a kind of global standard but it covers only very basic properties. In the following metadata capabilities of the KIT Data Manager, UNICORE and the MoSGrid Science Gateway will be presented.

KIT Data Manager

KIT Data Manager is an architecture to build up experiment data repository systems suitable for huge amounts of primary data. It provides a set of services for managing data on terabyte scale supporting the whole data life cycle of scientific data. Therefore, different types of metadata are supported. For the different aspects of data life cycle management specific metadata exists (e.g.: base metadata, content metadata, data organization metadata, authorization metadata, workflow metadata, curation metadata). The metadata allows getting all necessary information about data without having direct access to it (base metadata, data organization metadata, content metadata). Other metadata describes rules on how the data should be treated (curation metadata) or who is allowed to access to which extend (authorization metadata). All this metadata sets are linked to the original data via the Object Identifier (OID) which provides a unique identifier for each dataset. The base metadata the KIT Data Manager uses is based on the Core Scientific Metadata Model (CSMD). Its hierarchical structure is shown in Figure 1. The 'Digital Data Object' is linked to the data and contains some base properties like experimenter, start of the data acquisition, end of data acquisition and upload date. The 'Investigation' holds one or more 'Digital Data Objects' which are in at least one aspect similar to each other. As they are organized as a collection there is the possibility to define unitary actions/rules for such a collection. The 'Study' itself can be regarded as a natural collection of 'Investigations'.
These base metadata entities are generated during the data ingest in XML format and are stored next to the data. As most tools support at least Dublin Core (DC) as metadata standard also an XML file holding the DC metadata is generated. If there is community specific metadata available it will also be extracted and stored in XML format. Based on these metadata additional services are available.
To enable search the extracted metadata is registered to an elastic search cluster. If the metadata has also to be distributed an OAI-PMH server can be established which provides a standardized interface for harvesting metadata.
As all metadata is available in XML it is possible to transform them to another format using XSLT transformations. Therefore the metadata concept of the KIT Data Manager is easily adaptable to new demands.

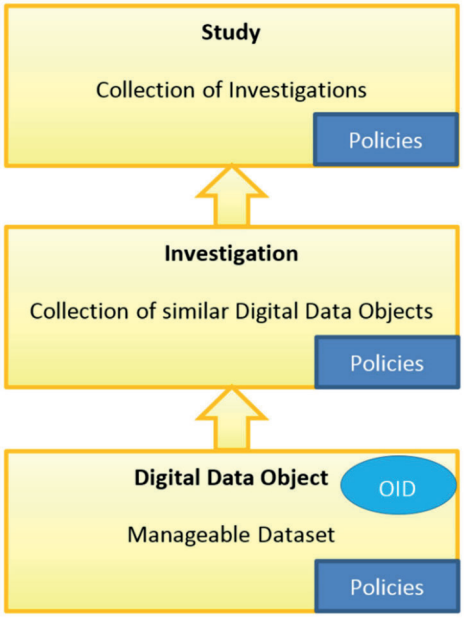

Figure 1 *The hierarchical base metadata structure of the KIT Data Manager is based on the Core Scientific Metadata Model (CSMD).*

UNICORE Metadata Management

The UNICORE middleware includes data and metadata management functionality as well. The metadata features are intended to complement the data management and file access functions, and are designed to be fully compatible with the UNICORE security and access control layers. The metadata interface offers the typical functions for creating, updating, deleting, indexing and searching metadata. Furthermore, there is a framework for extracting metadata from data automatically, using configurable parsers. The metadata is stored in a schema-free fashion as JSON key-value pairs. One distinguishing feature of UNICORE's metadata system is that the metadata is stored next to the

data, i.e. on the same physical storage, and is subject to the same access control and shares user and group management with the actual data.
Indexing and searching uses the Apache Lucene engine, while the metadata extraction system is built on Apache Tika. The search indexes are updated automatically each time the metadata is updated. The search function can be executed on a single data store, but can also be run as a federated search on all the data stores that are available to the user.
The metadata system can be accessed using UNICORE's command line, GUI clients and via its API in any Java application. Figure 2 shows a graphical metadata user interface built into the UNICORE Rich Client (URC).

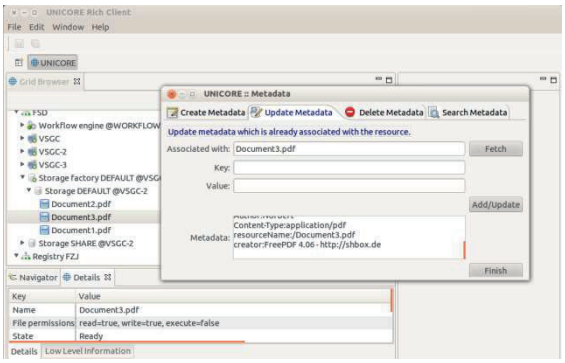

Figure 2 *The metadata management within the UNICORE Rich Client includes capabilities for creating, updating, deleting and searching for metadata.*

Generic Metadata Management based on the MoSGrid Science Gateway

Science gateway architectures (Figure 3) provide a single point of entry to make complex infrastructures such as HPC and data resources more efficiently and readily available by integration with workflow, metadata, HPC, and data management systems to handle complex computing tasks and big data requirements. The user is enabled to utilize these resources in an easy to use and efficient way, despite of the complexity of the underlying infrastructure.
The MoSGrid science gateway is such a solution for molecular simulations and docking tools. It is based on the Liferay portal, the WS-PGRADE/gUSE science gateway middleware, the distributed file system XtreemFS, and the grid middleware UNICORE. The integration of metadata management capabilities in MoSGrid necessitated two steps. First, MSML (Molecular Simulation Markup Language) has been designed to represent information about small and large molecular structures, workflows, and results. Besides the reasonable representation of data, one of the main aspects of metadata is the possibility to quickly search data on a large scale. Thus, the second step was to integrate the UNICORE metadata service which uses Apache Lucene as the most widely used library in efficiently searching data based on meta information.
Building on these design and implementation experiences a novel and generic metadata management approach is currently being designed to significantly enhance big data on HPC systems.
First, underlying metadata systems shall be transparently accessible via a programming interface. The interface is planned to support important metadata systems such as UNICORE and access standards such as CDMI and OAI-PMH. The goal is to enable users to utilize metadata systems without noticing which one is deployed. An important aspect is the efficient integration with HPC systems to enable the seamless execution of computing tasks based on metadata. Also, automatic extraction, annotation, and indexing of metadata is essential to enable management of millions of files.
Then, advanced user interfaces are planned to be offered as generic components for integration in specific use cases. Developers shall be enabled to avoid re-creating user interfaces. An example is a search interface where results can be seamlessly used as input during job submission. Another example is a filterable metadata browser to flexibly discover large data sets. Also, data views are planned to display files depending on a given context. For example, in a monitoring display only relevant job results are shown. Automation will enable quick adaptions of these interfaces to different types of use cases.

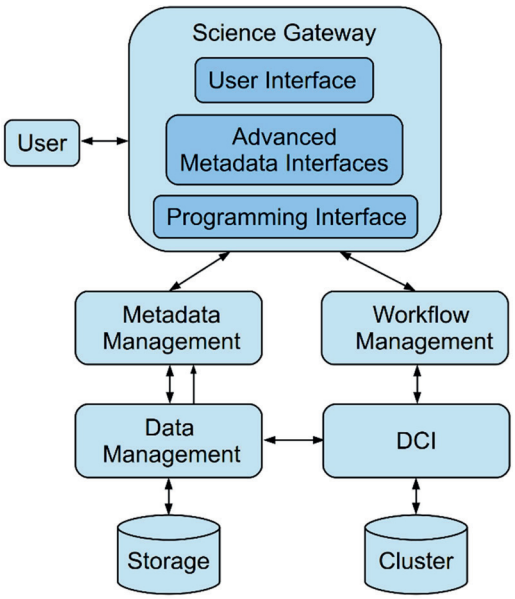

Figure 3 *A generic metadata management design in a complex Science Gateway infrastructure is depicted that can enable the seamless handling of millions of files.*

The Electrical Data Recorder

Fabian Rigoll, Heiko Maaß - KIT

Our society heavily relies on the ubiquitous availability of energy. However, in order to slow down the possibly irreversible climate change a paradigm shift is needed: electrical power grids that are currently characterized by a demand-driven production will make a transition to a production-dependent control of the electrical energy consumption.
Even though today's energy production is still largely based on fossil fuels, the future power generation will largely depend on volatile and decentralized energy sources, such as photovoltaics, wind turbines, and biogas plants. The increase in decentralized feed-in as well as the growing number of electric vehicles can cause imbalances in the electrical power grid.

An electrical power grid's quality is defined by two main properties: frequency and amplitude. In Europe, a frequency of 50 Hz is targeted as the balanced case between production and consumption. A decreasing energy demand causes the frequency to rise in the entire grid, whereas an increasing demand of energy causes the frequency to fall.
In contrast to the frequency control, voltage is not a global property of an electrical power grid. The consumption has a retroactive effect on the local voltage in the same part of the grid. If they are not levelled out on a local level, deviations from the targeted 230 V can be observed.
The rising number of decentralized feed-ins and the incorporation of electric vehicle charging affect the grid supply quality. Thus, it is important to monitor the grid comprehensively in order to maintain reliability, stability, and quality.

The Electrical Data Recorder (EDR) is a KIT developed device which is capable to measure voltages and currents of three phases in an electrical power grid at a rate up to 25 kHz. This allows for a detailed estimation of frequency and voltage as well as for the detection of harmonics and irregularities in the electrical network. If equipped with Rogowski coils, the EDR is able to measure power flow parameters on all three phases. Distributed measurements using different EDRs are synchronized by GPS for enabling wide area monitoring and comparison. All data are transferred to a large database for permanent storage.
As both temporal and voltage resolution are comparably high, large amounts of data are produced. Recording voltage channels only, one single EDR creates roughly 9 GiB of data per day or about 3 TiB of data per year at a typical acquisition rate of 12.8 kHz. Currently, three EDRs are installed and operate continuously. One of them is placed at KIT Campus south at the Energy Smart Home Lab, whereas the other two are used at different locations at KIT Campus North. More will be installed in the near future.

A web service has been employed to receive the EDR data and store them in the Large-Scale Data Facility. A Hadoop cluster is used to provide data search, semantic search, and smart browsing on the data. Our big data methods provide access rates greater than 350 MiB/s even for evaluation intervals of 3 weeks. This allows for efficient data analysis and interactive visualisations as shown in Figure 1.

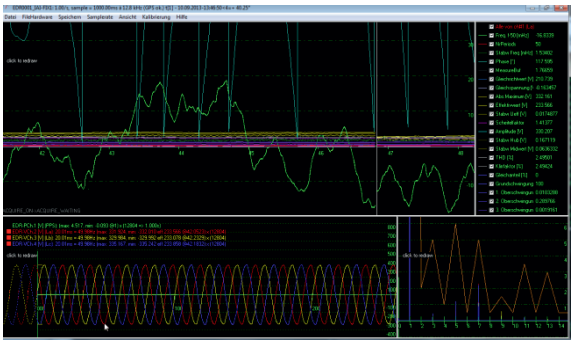

Figure 1 *The EDR enables detailed analysis of voltages, frequencies, and harmonics with subsequent interactive visualisation.*

The EDR offers sophisticated wide-area and large timescale comparison of measurements at different energy grid locations. New big data tools and pattern recognition techniques are currently being developed to contribute for advanced energy grid monitoring and reliable control in the future.

Complexity of Electro-Chemical Systems

Josef Anton, Timo Jacob – University of Ulm

Motivation

The relevant processes in electro-chemical energy storage occur on vastly varying time and length scales (see Figure 1). However, modeling of the crucial processes on the various scales also requires different methods and algorithms. By a combination of these methods, properties of electro-chemical cells can be predicted on a first principles basis, i.e. without involving empirical parameters. For example, in order to understand the macroscopic charge and mass transport, various factors such as the barriers hindering the elementary diffusion steps have to be known. This information can be determined on the atomistic scale, where specific structural and energetic information is calculated using first principles methods.

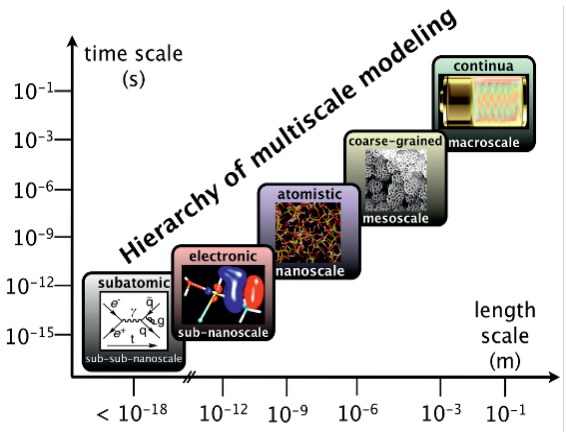

Figure 1 *Schematic presentation of the hierarchy of multiscale modeling. Data generated on smaller time and length scales will be used as parameters for simulation on a larger time and length scales.*

The main aim of our activities is to bridge the gap between the different time and length scales involved in the electro-chemical energy storage establishing the appropriate methods for a reliable multiscale approach and by applying these methods to relevant systems such a Li-ion or metal-air batteries. Thus, the information gained on macroscopic level will help to understand the crucial processes on a mesoscopic and macroscopic level. Besides structural and stability aspects of the different battery materials (i.e. electrodes, liquid or solid electrolytes, and their interfaces), their influence on the electronic and macroscopic properties is addressed.

Data Life Cycle

In order to improve the Data Life Cycle for the community, a server-client solution (see Figure 2) is implemented. Clients establish an encrypted connection to the main server in order to access its services. This dedicated server provides all necessary software tools for multiscale modelling. It runs automatically plausibility checks of the input data, submits the calculations to one of the available computational clusters and monitors them. After the completion of a computation, the system checks the output data and will copy them back to the server. In addition, it will evaluate these data, will annotate them with metadata and it will take care of the long term archiving

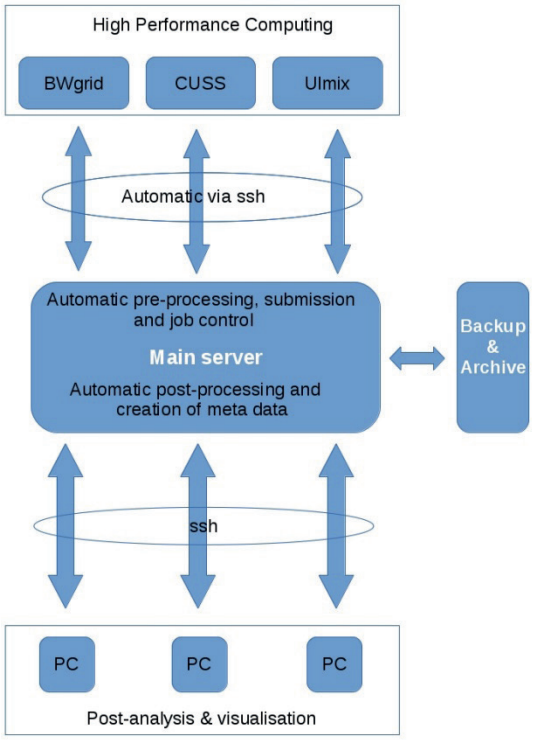

Figure 2 *Schematic presentation of the server-client solution. Besides the automatic pre- and post-processing of the input and output as well as job submission and control, the main server annotates the generated data with the metadata and handles their archival. Post-analysis and visualisation of generated data will be performed on the clients.*

FAIR Tier0: Building Large-Scale Cross Site Connections

Dennis Klein, Thorsten Kollegger, Walter Schön, Kilian Schwarz, Thomas Stibor - GSI

Introduction

At GSI an accelerator facility of the next generation, FAIR (Facility for Antiproton and Ion Research, see Figure 3) is being built. The computing resources required will be dictated mainly by the two large experiments, CBM (Compressed Baryonic Matter) and PAnDa (Proton Anti proton Darmstadt). Current estimates for the first year of data taking are 200,000 CPU cores, 30 PB of disk space and the same amount of tape archive. The data rate from the experiments will be in the order of magnitude of 1 TB/s. Due to the signature of the events the FAIR experiments cannot rely exclusively on hardware triggers, though. Therefore the complete reconstruction up to particle identification will have to be done in quasi real time in order to be able to distinguish between signal and background events and to reduce the amount of data to be stored in the end to manageable sizes.

The FAIR computing model foresees a distributed tier0/tier1 centre consisting of GSI and the surrounding universities and partner institutions. The combined FAIR tier0 centre will be embedded in an international Grid/Cloud infrastructure. Computing clusters will be loosely and densely coupled and large data sets need to be efficiently processed and analysed in a distributed and parallel manner. A suited file system for the requirements presented above and in addition scales effectively and allows seamlessly accessing large data in wide area networks (WAN) environments is the Lustre file system.

In the following sections we present a realization of Lustre high-speed connections for large-scale data transfers in WAN and address security related access control mechanisms required for the FAIR project. In addition we present how this approach can be embedded into international Grid and Cloud infrastructures.

Tera-Link Connections with Lustre Routers

For addressing the first issue a 120 GBit/s high-speed connection between GSI (Hera cluster) and LoeweCSC in Frankfurt based on LNET routers is realized and seamless Lustre mounts are implemented (see Figure 1).

Additionally, several experiments are performed to verify that full network bandwidth saturation can be achieved. Results on network performance are visualized in Figure 2.

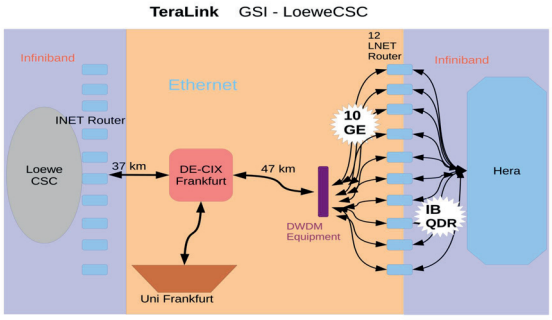

Figure 1 *High-speed connection is realized by bundling 12 machines equipped with 10 GBit/s ethernet cards acting as LNET routers. Both computing sites LoeweCSC and Hera cluster at GSI seamlessly are connected via Infiniband over IP over a distance of around 84 km.*

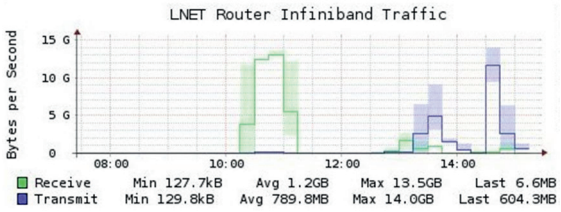

Figure 2 *Bandwidth saturation experiments between LoeweCSC in Frankfurt and the Hera cluster located at GSI. One can observe, that closely the optimal bandwidth saturation of 15 Gbyte/sec is achieved.*

A Lightweight Access Control Mechanism for Lustre in WAN Domains

For controlling access to Lustre of clients outside the GSI domain, that is partner institutions and universities, a Linux kernel module based on Linux user and group identification (short UID/GID) and Lustre network identifier is developed. It allows controlling read and writing access for arbitrary specified UID's /GID's and Lustre network identifier ranges.

In the context of WAN Lustre deployment the proposed mechanism enables a straightforward and lightweight access control of Lustre clients located in different WAN domains. The access control mechanism is implemented as a separate Linux kernel module and exports an access-granting function which is hooked into Lustre's core metadata system for granting or denying data access. Further details can be found at [18].

Embedding in International Grid and Cloud Infrastructures.

On the level of data management and repositories many software systems are available. Interoperability is rarely given due to missing implementation of common standards. In the long run reacting on changing technologies therefore is difficult.

Under participation of GSI, a prototype of a globally distributed file system has been set up based on the xrootd protocol which is commonly used within the High Energy Physics community. An important idea is the transition from separated and localized storage elements to a global “file system”. A major building block is a working interface between xrootd and Lustre. This way it can be guaranteed that synergy effects are created and that aspects of both storage system which are important for FAIR are being taken into account.

Via modular plugins which are available for xrootd version 4 also proxy solutions can be set up so that Grid jobs will be able to run in firewall protected HPC clusters. Moreover via plugins to various cluster file systems xrootd will be able to hand over to clients URLs pointing directectly to the cluster file systems used at the participating centres.

Summary

A crucial step towards the tera-scale FAIR computing model was presented. That encompassed processing and analysing large-scale data in WAN environments and taking scalability and security related issues into account. In addition, the concept of embedding the combined FAIR tier0/tier1 centre into an international Grid and Cloud infrastructure was discussed.

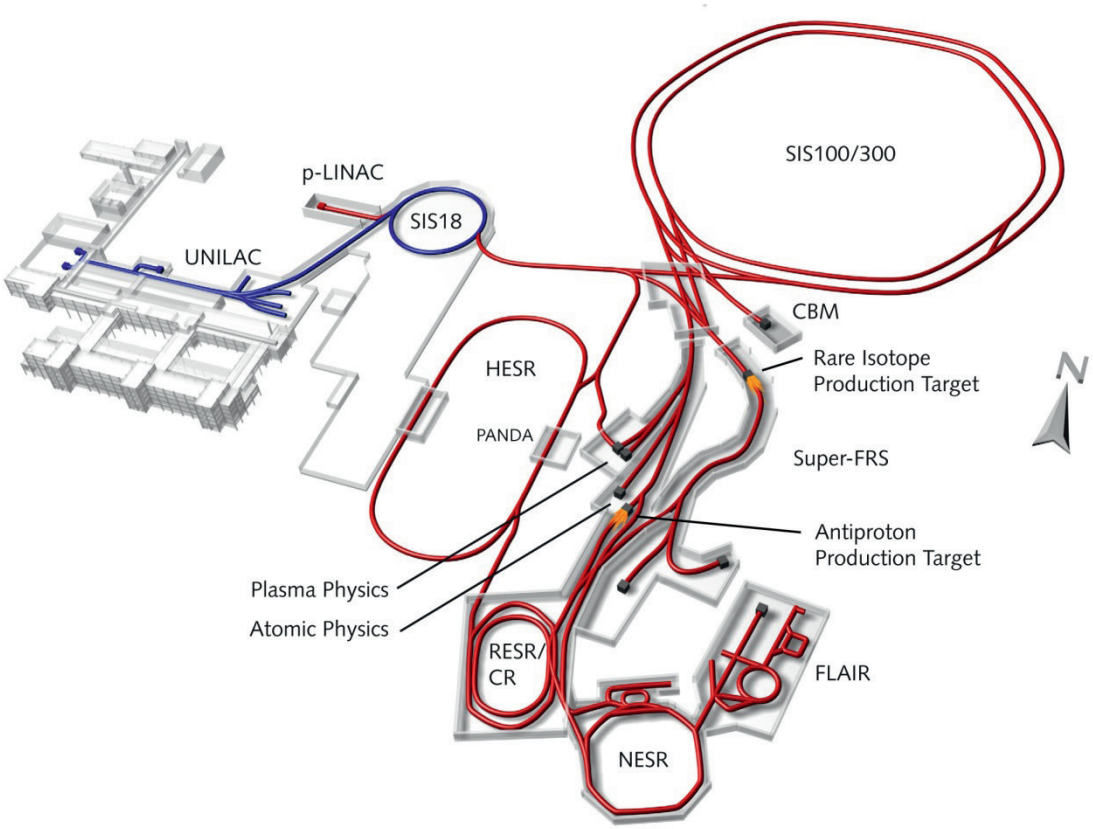

Figure 3 *FAIR is an international accelerator facility currently under construction. In the final setup FAIR consists of eight ring colliders with up to 1,100 meters in circumference, two linear accelerators and about 3.5 kilometers beam control tubes (see red coloured areas). The existing GSI accelerators serve as pre-accelerators (blue coloured areas). FAIR will use antiprotons and ions to perform research in the fields of: nuclear, hadron and particle physics, atomic and anti-matter physics, high density plasma physics, and applications in condensed matter physics, biology and the bio-medical sciences.*

Bibliography

1. **Jung, Christopher, et al.** Optimization of data life cycles. *Journal of Physics: Conference Series*. 2013.

2. **Chapman, H. N., et al.** Femtosecond x-ray protein nanocrystallography. *Nature*. 2011.

3. **Barty, A., et al.** Cheetah: software for high-throughput reduction and analysis of serial femtosecond x-ray diffraction data. *Journal of Applied Crystallography*. 2014.

4. **Boutet, S., et al.** High-Resolution Protein Structure Determination by Serial Femtosecond Crystallography. *Science*. 2012.

5. **Szegedy, Christian, et al.** Intriguing properties of neural networks. 2013. arXiv:1312.6199.

6. **Becker, Daniel.** A neural network-based pre-selection of Big Data in photon science. 2014. in preparation.

7. **Schock, B.** Cross-application communication on NUMA systems, Master Thesis. *HTW Berlin*. 2014.

8. **Stotzka, R. (ed.).** Data Life Cycle Lab. Key Technologies. Status 2013. Big Data in Science. *KIT publications*. 2013.

9. **Müller, P., et al.** Analysis of fluorescent nanostructures in biological systems by means of Spectral Position Determination Microscopy (SPDM). *Current microscopy contributions to advances in science and technology*. 2012.

10. **Efros, P., Buchmann, E. and Böhm, K.** FRESCO: A Framework to Estimate the Energy Consumption of Computers. *IEEE Conference on Business Informatics*. 2014.

11. **Cremer, C., et al.** Superresolution imaging of biological nanostructures by spectral precision distance microscopy. *Biotechnology Journal*. 2011.

12. **Bohn, M., et al.** Localization microscopy reveals expression-dependent parameters of chromatin nanostructure. *Biophysical Journal*. 2010.

13. **Kaufmann, R., et al.** Analysis of Her2/neu membrane protein cluster in different types of breast cancer cells using localization microscopy. *Journal of Microscopy*. 2011.

14. **Furano, Fabrizio, et al.** Dynamic federations: storage aggregation using open tools and protocols. *J. Phys.: Conf. Ser*. 2012.

15. **Millar, A. P., et al.** dCache, agile adoption of storage technology . *J. Phys.: Conf. Ser*. 2012.

16. **Millar, P., et al.** dCache: Big Data storage for HEP communities and beyond. *J. Phys.: Conf. Ser*. . 2014.

17. *The Dynamic Federations demo*. [Online] http://federation.desy.de.

18. European Lustre Conference 2013 website. [Online] http://www.eofs.eu/?id=lad13.

LSDMA contacts

URL: http://www.helmholtz-lsdma.de/

E-mail: lsdma@scc.kit.edu

Project Coordination: Achim Streit

Project Management: Christopher Jung

Heads of subprojects:

- DLCL Climatology: Jörg Meyer
- DLCL Energy: Fabian Rigoll
- DLCL Key Technologies: Rainer Stotzka
- DLCL Matter: Martin Gasthuber, Kilian Schwarz
- DLCL Neuroscience: André Giesler
- DSIT: Marcus Hardt

LSDMA events:

- Symposium 'The Challenge of Big Data in Science' (each autumn)
- Community Forum (each spring)
- PhD Workshop (each summer)
- LSDMA All Hands Meeting (each spring and each autumn)

Big Data in Science – 1st Edition (September 2014)

Editors: Christopher Jung, Achim Streit