

Data-intensive analysis for scientific experiments at the Large Scale Data Facility

A.O. García*, S. Bourov, A. Hammad, J. van Wezel, B. Neumair, A. Streit
Karlsruhe Institute of Technology
Steinbuch Centre for Computing
Karlsruhe, Germany

V. Hartmann, T. Jejkal, P. Neuberger, R. Stotzka
Karlsruhe Institute of Technology
Institute for Data Processing and Electronics
Karlsruhe, Germany

J.C. Otte
Karlsruhe Institute of Technology
Institute of Toxicology and Genetics
Karlsruhe, Germany

S. Pfeiffer, T. Schenker, C. Schmidt
Karlsruhe Institute of Technology
Institute for Applied Computer Science
Karlsruhe, Germany

ABSTRACT

To address the growing need for value-added storage services for data intensive experiments at the Karlsruhe Institute of Technology (KIT), the Large Scale Data Facility (LSDF) was conceived and launched end of 2009. The LSDF main focus is to support scientific experiments producing large data sets reaching into the petabyte range in their data requirements. Working closely together with the involved scientific communities, the LSDF provides them not only with adequate storage space but also with an analysis facility and –more importantly– with support and value added services for data management, processing and preservation. In this poster we describe the approach taken to perform data analysis in LSDF, as well as for data management of the scientific datasets.

1 INTRODUCTION

In the last few decades information technology did see exponential gains in most areas –including computing, storage and networking capacities, driven by Moore’s [1] and similar laws. This trend allowed a new scientific paradigm to emerge: the generation and exploration of massive data sets became a scientific discovery method on equal footing with the traditional experimental paradigm, theoretical analysis, and more recently computer simulation[2]. Even considering the capacity and throughput growth of IT hardware, the tasks of managing and analyzing exponentially growing datasets remain. These tasks need to be tackled independently and keep becoming more and more critical for many scientific communities, at least for all non-technically oriented disciplines, but very often also beyond.

2 THE LSDF PROJECT

The LSDF project aims at supporting the different scientific communities at KIT in dealing with those problems, while providing the necessary resources. The LSDF infrastructure consists of two storage systems with a total capacity of currently roughly 2 PB, while the data processing is guaranteed by a compute cluster directly attached to the storage of 60 nodes respectively 480 physical cores. Two alternatives are offered for analysis tasks: the Hadoop [3] framework for data intensive distributed computing, and a Cloud solution based on the OpenNebula [4] middleware. Cloud instances started in the cluster can then access the scientific data in the LSDF

*e-mail: garcia@kit.edu

storage for processing, allowing users to deploy data analysis tools requiring a dedicated or highly customized environment, or running solely on a specific operating system.

On the software side, LSDF gives major emphasis to metadata handling, a critical aspect of assuring long-term value of the scientific data. The project developed several community specific metadata schemes, a metadata repository, a software library, as well as a graphical tool for accessing the storage resources, all of these aiming at supporting the data management and processing workflows of the partner scientific communities.

2.1 Data and metadata management

The LSDF provides access to data via several storage protocols. Depending on the users’ location and requirements different access methods may be used. Furthermore, new protocols and services for access to data might be added in the future, possibly conditioned by technological changes. For the scientific user, accessing the LSDF data must not be significantly more complicated than accessing local data. For software developers writing scientific applications for LSDF, it would be unacceptable to have to constantly adapt to new data access methods. Therefore, the Abstract Data Access Layer API[5] (ADALAPI) was developed to address this challenge: it hides access technologies behind one single interface. The ADALAPI also handles user authentication tasks, supporting command line, dialog and file based user authentication. Hence, any access protocol which supports these functionalities can be integrated into the ADALAPI. Currently the implementation can handle POSIX like data access, the GridFTP protocol, SSH Secure Copy, WebDAV, and access to Hadoop filesystems.

Of course, a basic data access API is not sufficient for accessing data in a large scale. Storing data for several years means that a sustainable data management and organization is essential for the LSDF. Another layer of abstraction was developed to address the meta data aspect: the DataBrowser API. This API is responsible for accessing and administrating meta data programmatically. Due to the fact that the LSDF supports a wide variety of scientific communities, this API must allow generic access to meta data of any kind. Meta data schemes are defined in close cooperation with each community. Thereby the main goal is to support existing, community specific meta data standards, and to extend them to fit the functionality of the LSDF.

By combining the DataBrowser API and the ADALAPI, LSDF is able to build generic data workflows whose sustainability is achieved by a full abstraction from underlying technologies. Access and storage protocols can be replaced in the background while the front-end is not affected. Future developments will be focused on integrating existing meta data standards and emerging technologies

to establish the ADALAPI and the DataBrowser API as a flexible groundwork for large scale storage solutions.

2.2 Workflow execution for image processing

One central use case in the daily operation of LSDF is, of course, the storage of raw experimental data, as well as tagging it with adequate metadata. But another equally critical functionality is processing the stored data for scientific analysis. Therefore, Jejkal *et. al.* [6] implemented workflow execution management as another part of the LSDF software stack, aiming at keeping it as generic as possible. The user clears newly stored data for processing by tagging it on the metadata by means of the DataBrowser, while a cronjob takes care of polling the corresponding database and initiating the corresponding processing tasks. The execution of a scientific application is described by a process description file in XML-format, which contains information about the runtime environment, the algorithm and its parameters as well as version information, and of course the information of which data set should be processed by which data processing framework. In the background the database is queried for new execution tasks automatically, as shown in Fig. 1. If a new task was found the execution is prepared and scheduled using the corresponding processing framework. An associated monitoring system takes care of running executions and notifies responsible persons on success or failure. This structure allows the definition of workflows to execute almost arbitrary applications. This decoupling of process description from execution framework allows to keep the flexibility for future changes in the software landscape.

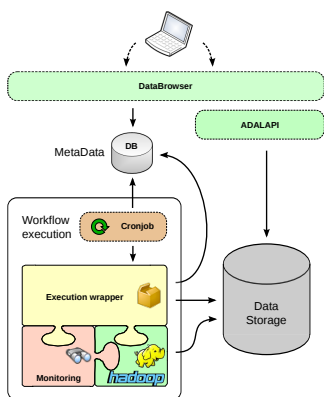


Figure 1: Workflow execution

2.3 Analysis for data intensive scientific experiments

In this work we describe two data analysis use cases in LSDF.

2.3.1 High throughput microscopy

Modern microscopy techniques enable a very detailed insight into the development of different biological organisms. One LSDF use case is given by the toxicological analysis of different chemical substances on the early vertebrate development. State of the art microscopes are able to generate in a completely automated way three dimensional image data *over time* of fluorescent-labeled fish [7]. Quantitative *in vivo* imaging results therefore in many 5D-data sets, each of several GB up to the TB range. These data sets represent a challenge not only for the storage hardware, but also for efficient image processing.

Based on the constraints defined by the LSDF setup, an automated and flexible processing pipeline was developed by S. Pfeiffer, J. Otte *et. al.* [8, 9] to analyze this huge multi-dimensional

biological data. Single operators are connected with each other resulting in a pipeline as operators of many muxels¹ [8]. Operators from several libraries like Matlab, ITK, VTK, Boost, etc., are in this way turned compatible within the tool, while additional operators can be implemented as plug-ins. Visualization of 2-4D image data is integrated in order to assist the user in parametrising and in identifying an applicable analysis chain of image data, which is then saved in the sequence-XML file and stored as part of the workflow metadata. The scientist can tag any data set to be processed according to his chosen pipeline by means of the DataBrowser. The subsequent processing of the images is performed as a service on LSDF's Hadoop cluster, as explained earlier. Within the automatically running workflow, a distributor takes care of combining any data set tagged for processing with the corresponding operators (sequence-XML) to a concrete workflow, and finally executes the required Hadoop job in the cluster. Hadoop runs the project processor task in multiple mapping instances by the MapReduce principle.

2.3.2 3D visualization of biomedical data

The High throughput Screening Platform mentioned above creates stacks of many varying-depth images of zebrafish embryos, and can produce very high resolution 3D volume imaging sets. Sizes of several terabytes can be achieved if desired. These data sets need to be visualized for the scientific analysis. However, processing terabytes of data can still not be done in a timely manner on a single computer. For visualizing large 3D volume data sets in LSDF, Neuberger *et. al.* [10] did use the Axis parallel Maximum Intensity Projection (MIP) to project them into 2D images for different angular viewpoints. The data processing is performed in the Hadoop MapReduce framework, by dividing a volume dataset into smaller sections, where the small volumes are projected in parallel in corresponding map tasks. The whole reconstructed projection is obtained by compositing the partial 2D images in the reduce step of the Hadoop job. To reduce the I/O overhead several projections from different angles are performed at once while keeping the input data cached. Using this procedure, datasets of 1 terabyte size can be projected in approximately 3 minutes per viewpoint by 330 mapping tasks on the 60 nodes cluster, which represents a significant gain with respect to a single-node analysis.

REFERENCES

- [1] <http://www.intel.com/about/companyinfo/museum/exhibits/moore.htm>
- [2] T. Hey, S. Tansley, and K. Tolle (Editors), *The Fourth Paradigm: Data-Intensive Scientific Discovery*, Microsoft Research, 2009.
- [3] <http://hadoop.apache.org/>
- [4] <http://www.opennebula.org/>
- [5] R. Stotzka *et. al.*, *Perspective of the Large Scale Data Facility (LSDF) supporting nuclear fusion applications*, Proceedings of The 19th Euro-micro International Conference on Parallel, Distributed and Network-Based Computing (PDP2011), 2011, pp. 373-379.
- [6] T. Jejkal *et. al.*, *LAMBDA – The LSDF Execution Framework for Data Intensive Applications*, to be published in Proceedings of The 20th Euro-micro International Conference on Parallel, Distributed and Network-Based Computing (PDP 2012), February 2012.
- [7] J. Gehrig *et. al.*, *Automated high-throughput mapping of promoter-enhancer interactions in zebrafish embryos*, Nature Methods **6**, 2009, pp. 911-916.
- [8] S. Pfeiffer, M. Mai, and J. Calliess, *On the Computational Benefit of Tensor Separation for High-Dimensional Discrete Convolutions*, Multidimensional Systems and Signal Processing **31**, 2010, pp. 1-25.
- [9] J.C. Otte, S. Pfeiffer, T. Schenker, and C. Schmidt, *Concept of multi-dimensional image analysis with flexible operators and cluster-computing*, poster presentation at the 6th European Molecular Imaging Meeting, Leiden, The Netherlands, June 2011.

¹Multi-dimensional ELEMENTS

- [10] P. Neuberger *et. al.*, *An approach for distributed visualization of 3D data at terascale*, to be published in Proceedings of The 20th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP 2012), February 2012.