Frascati Physics Series Vol. XL (2006), pp. 287– 296
Frontier Science 2005, New Frontiers in Subnuclear Physics
Milano, 12-16 September, 2005

PROTOTYPES FOR DISTRIBUTED ANALYSIS ON THE GRID - THE ARDA PROJECT

D. Liko, J. Andreeva, J. Herrala, M. Lamanna,
A. Mayer, J. Mosckicki, B. Koblitz, A. Peters, N. Santos
CERN, Geneva, Switzerland
C. Munro
Brunel University, London, U.K.
D. Feichtinger
Paul Scherrer Institute, Villigen, Switzerland

Forschungszentrum Karlsruhe, Karlsruhe, Germany
F. Orellana
University of Geneva, Geneva, Switzerland
T.S. Chen, S. C. Chiu, H.C. Lee, W.L. Ueng
Academica Sinica Grid Computer Center, Taipei, Taiwan
A. Demichev, A. Berejnoy
Moscow State University, Moscow, Russia
V. Galaktionov, V. Pose
Joint Institute for Nuclear Research, Dubna, Russia

Abstract

In collaboration with the four LHC experiments the ARDA [1)] project has developed prototypes for distributed analysis. These prototypes were developed in close interaction with the development of the EGEE [2)] middleware, gLite. In this context, ARDA contributed to the evaluation of new middleware components. A further ARDA contribution is the development of a new metadata catalog, which is now part of next gLite release. Currently the prototypes are still in evolution with the goal to become part of the LCG environment for the LHC users.

1 Introduction

The ARDA project (A Realisation of Distributed Analysis for LHC) was created within the LCG project in 2004 to develop a prototype grid analysis

system for the experiments at the Large Hadron Collider (LHC). The guiding idea behind ARDA is that exploring the opportunities and the problems encountered in using the grid for LHC analysis would provide key inputs for the evolution of gLite the EGEE middleware. Enabling a large, distributed community of individual users and small groups to use the Grid without central control stresses the infrastructure in a radically different way compared to large scale, continuous 'production' activities like generation of simulated data or event reconstruction.

Fostering innovation is a key element: the understanding of analysis activities in the LHC era is still evolving very fast, and so we need to retain flexibility. The patterns observed in the past suggest that the new Grid infrastructure will enable and stimulate new approaches to data handling and analysis with the ultimate goal to enable a large scientific community to maximize the scientific output of the LHC programme. A sound approach to such an evolving infrastructure is to prototype the future systems together with the users, exposing them early to the Grid environment, and discussing the evolution on the basis of their experience.

Testing the Grid under real conditions gives effective feedback to the developers of Grid middleware. During the first phase of EGEE, ARDA played a key role in the testing of the middleware, with its access to 'previews' of gLite components. Progressively this activity moved towards detailed studies of performance issues, but always using the analysis scenario as a guideline. As an example, the experience and requirements of the LHC experiments led ARDA to propose a general interface for metadata access services. Eventually an ARDA prototype called AMGA (ARDA Metadata Grid Application) made its way into the gLite middleware and is now also used by non-HEP applications. The ARDA project is developing prototype for analysis on the grid for the four LHC experiments. These prototypes follow the experiment strategy and are complementary in their approach.

2 Distributed Analysis Prototypes

From the beginning, it was decided to agree with each LHC experiment an a-priori independent prototype activity. It was considered unrealistic to force at an early stage commonality in the use of tools, since each experiment has different physics goals and data organization models. On the other hand, all

different activities hosted in the ARDA team benefit for common experience and cross fertilisation. In this section, a series of examples of the prototype activity with the LHC experiments are given.

2.1 ALICE

The ALICE prototype is an evolution of the early distributed-analysis prototypes made by the experiment using their AliEn system, incorporating features from the PROOF [3] system (Parallel ROOt Facility) and providing the user with access via the ROOT/ALIROOT prompt. Close integration with the standard (local) analysis environment can be obtained only by carefully designing the gateway into the distributed system. An important component, developed within ARDA, is the C Access Library, optimizing the connection to the Grid infrastructure via an intermediate layer that caches the status of the clients (in particular their authorization) and therefore helps to optimize the performance as required for interactive use. Operations like browsing the file catalogue or inspecting a running job can be provided via mechanisms already known to the ALICE users (shell commands and ROOT system).

The ALICE framework provides user analysis in batch and interactive mode. It uses the AliEn grid middleware as a high-level service interface for access to the AliEn file catalogue and distributed computing resources via internal interfaces (LCG, native AliEn). The user interface consists of a grid shell, a QT based GUI and an AliEn grid plug-in to the ROOT framework. The GUI allows the selection of data, the execution of analysis jobs in interactive and batch mode, the retrieval and the storing of results. The PROOF system is used for interactive analysis, which on its own provides a GUI for interactive analysis. ALICE uses a multi-tier PROOF setup to allow analysis of data distributed over several mass storage systems in the same PROOF session. Depending on the data volume to be analyzed the response time for interactive analysis can be few seconds, while for batch analysis it is in the order of several minutes or hours.

2.2 ATLAS

The ATLAS strategy follows a service oriented approach to provide Distributed Analysis capabilities to its users. Based on initial experiences with a dedicated analysis service, the ATLAS production system has been evolved to support

analysis jobs. The ATLAS ARDA group has contributed to the initial system and is now coordinating the overall ATLAS effort.

As the ATLAS production system is based on several grid flavours (LCG, OSG and Nordugrid), analysis jobs are supported by specific executors on the different infrastructures (Lexor, CondorG, Panda and Dulcinea). The implementations of some of these executors in the new schema are currently under test, in particular also in analysis mode [4)].

While submitting jobs to the overall system will provide seamless access to all ATLAS resources, ATLAS support user analysis by submitting directly to the separate grid infrastructures (Panda at OSG, direct submission to LCG and Nordugrid.) A common job definition system is currently under development that will be supported by all systems.

Finally a common user interface project, GANGA, will provide support for the various submission options and will provide a consistent user experience. This project is developed in collaboration with the LHCb experiment and is discussed in more detail in section 2.4.

A point of specific interest is the coexistence of Distributed Analysis by individual ATLAS users with the overall ATLAS production. At this point the effective coexistence of different jobs (long production jobs and relatively short analysis tasks) is still under study and the final model is being worked out. Another fundamental point (and an important contribution of ARDA) is the study and the tuning of the Workload Management System, which would have a major impact for analysis jobs, when the system will have to deal with a increasingly larger number of jobs without degrading the system response time [5)]. These aspects are currently addressed in the context of the ATLAS LCG/EGEE taskforce.

2.3 CMS

The ARDA-CMS activity started with a comprehensive evaluation of gLite and the existing CMS software components. Eventually ARDA focused on providing a full end-to-end prototype called ASAP [6)], prototyping some advanced services such as the Task Monitor and the Task Control. The Task Monitor gathers information from different sources: MonaLisa (mainly providing run time information via the CMS C++ framework COBRA); the CMS production system; Grid-specific information (initially the gLite/LCG logging and

bookkeeping and recently the R-GMA system). The Task Control implements CMS specific strategies, making essential use of the Task Monitor information. The Task Control understands the user tasks (normally a set of jobs) and organizes them in a way which enables the user to delegate several tasks, e.g. the actual submission (the user registers a set of task and then disconnects) and error recovery. Some key components of this very successful prototype, which incorporated a lot of feedback from users, are now being migrated within the official CMS system CRAB (CMS Remote Analysis Builder) in the framework of the CMS-LCG taskforce.

CMS has developed a job generating tool - CMS Remote Analysis Builder (CRAB) - to support user analysis on the Grid. In the CMS workload management model, jobs are sent to the sites where the input data are available.

Though there is an overlap between CRAB and ASAP in the job generation phase, ASAP provides additional functionality by managing tasks for the user on behalf of him. Users only need to call the ASAP job generation command in order to generate a set of jobs, which are combined into a task. The newly created task can be registered in the ASAP Task Manager.

The Task Manager follows the progress of task processing, resubmits failed jobs and generates web pages, which provides physicists the status of all their tasks. Monitoring of the user jobs is based on the Monalisa monitoring system. The functionality of the Task Manager and the Job Monitor was appreciated by the pilot CMS users. These components should be integrated into the final CMS system. Currently ARDA is working on the integration of the Task Manager and the Job Monitor with the CRAB job generation tool.

Another important aspect of Grid usage, both for analysis and production, is to provide a global view of all jobs belonging to a given VO, presenting information about usage, sharing of resources, performance and data distribution issues, failure rates of the Grid and the physics applications, and load balancing between different sites. ARDA is participating in the development of the CMS Dashboard. The CMS dashboard is a tool to show the CMS computing activities, allowing to focus on specific interval of times, sites, group of jobs, sets of data etc..

2.4 LHCb

The ARDA-LHCb prototype activity is focusing on the GANGA [7] system (a joint ATLAS-LHCb project). The main idea behind GANGA is that the physicists should have a simple interface to their analysis programs. GANGA allows preparing the application, to organize the submission and gather the results. The details needed to submit a job on the Grid (like special configuration files) are factorised out and applied transparently by the system. In other words, it is possible to set up an application on a portable PC, then run some higher-statistics tests on a local facility (like LSF at CERN) and finally analyse all the available statistics on the Grid just changing the parameter which identifies the execution back-end.

The complete functionality of GANGA Core is defined in the GANGA Public Interface (GPI). GPI is a python-based, user-centric API that is a key component of the system.

The GPI provides a convenient abstraction layer for job submission and monitoring. Presently GANGA supports several back-ends, namely different version of the LCG/EGEE middleware, the DIRAC system (LHCb production system), the ATLAS production system plus local batch systems and local execution. Any application can be submitted, but the system is optimized to help in the customization of the major applications ATLAS and LHCb users are interested in, namely ATHENA and GAUDI applications. As a result, GANGA provides the same interface for local and grid environment. From the end-user perspective it is very easy to switch between various environments depending on the computing and data access needs (local debugging fully distributed execution of heavy tasks) with the reduced learning overhead.

A graphical user interface, based on the Qt framework, is a part of the new GANGA releases. The GUI integrates scripting and graphical capabilities into a single environment. It also provides an easy and intuitive way of work, especially important for the beginners. The GUI is a GPI overlay and it is a perfect example of how the GANGA Core may be easily embedded in a separate framework. The present architecture envisages the creation of different specialized user interfaces to best cope with specific activities. Although there is a clear value in graphical portals (especially for novice users on one hand and to simplify large repetitive tasks as in large productions), the availability of the 'grid scripting language' provided by the GPI is a real plus provided by

this system (in addition, the usage of the GU produces editable files which can be modified and embedded in other applications).

GANGA is an open-development framework which fully exploits the plug-in architecture. This makes the integration of new applications and backends very easy. Applications, such as Geant4 simulations in medical physics, or the BLAST protein alignment algorithm in biotechnology have been successfully run with GANGA.

3 Contributions to the EGEE middleware

3.1 Evaluation of the Workload Management

In the next generation EGEE workload management system, gLite WMS, new features such as bulk submission and input sandbox sharing are introduced for improving the job processing performance. In order to evaluate the bulk submission feature, all test jobs are submitted in bulk. The job processing performance is represented by the following three inter-periods of job lifecycle: Job submission time: the time interval from issuing job submission command on User Interface (UI) to the end of job submission (i.e. the prompt back on UI). Job dispatching time: the time interval from the job acceptance by Network Server to the completion of job transferring from Resource Broker (RB) to Computing Element (CE). Job finishing time: the time interval from job launching on CE to the end of job lifecycle. The evaluation is performed in the context of the ATLAS LCG/EGEE taskforce on the test environment at INFN-Milan with the installation of gLite 1.4 [5)].

3.2 Evaluations of the Data Management

File Catalogues represent an important component of any Grid system. ARDA focused on the evaluation of the LCG File Catalogue (LFC) and the File and Replica Management Catalogue (FiReMan) [8)] . LFC is a stateful, connection-orientated catalogue written in C while FiReMan uses a service-orientated approach using SOAP for communication and Tomcat as a server.

A series of tests were conducted using a multi-threaded client to imitate many simultaneous connections. Each of the elementary catalogue operations: insert, delete and query were tested with different test parameters such as the number of threads, bulk message size for FiReMan and use of sessions for LFC.

The Oracle and MySQL backends were used and the client and server were connected by a Grid Security enabled SSL connection over a LAN and a WAN.

The LFC catalogue shows very similar performance with both the Oracle and MySQL back-ends. A query rate of 24 with 1 client up to 227 queries/s for 20 clients. The MySQL implementation of the FiReMan catalogue with a bulk size of 1000 can perform 241 queries/s with 1 client up to a maximum of 407 with 50 clients. The Oracle implementation performs much better mainly due to the fact that the application logic is implemented as stored procedures in Oracle rather than in Tomcat. For a single client with a bulk size of 1000 763 queries/s can be obtained up to a limit of 1800 with multiple clients. This relies on the fact that the database will cache the result so that we can effectively measure the overhead the catalogue server imposes on the operation.

3.3 The Grid metadata system AMGA

AMGA [9] (ARDA Metadata Grid Application) is the metadata service within gLite. It can be used to associate key-value pairs as metadata to files stored on the grid as well as to provide simple relational database services on the grid.

AMGA was initially developed by ARDA to be a prototype system for metadata access based on the knowledge obtained from extensive testing of the experiment prototypes of metadata catalogues. Key features of AMGA include a SOAP WebService front-end as well as a TCP streaming interface, bulk operations via TCP streaming or SOAP iterators and several different database back-ends including MySQL, Oracle and PostgreSQL. AMGA can be used as an add-on to the LFC file-catalogue.

Many functionality tests similar to the tests of the experiment metadata solutions were performed with AMGA in order to validate the basic design and stability of the implementation, in particular the behaviour of bulk operations which are done via iterators in the SOAP interface or via TCP streaming. Tests on a wide area network showed very encouraging performance of the streamed operations which allow minimize the impact of the long roundtrip time. We also validated the ACL-based security of AMGA.

4 Collaboration with other sciences

ARDA is a very active contributor to the EGEE project. EGEE benefits from the exchange of ideas with and experience of groups that actively support

scientific communities, notably the LHC experiments and biomedicine. Being capable to map concepts and strategies developed in one particular application domain to other sciences is a powerful indicator of an improved theoretical understanding of Grid techniques. Each application domain can contribute with its specific requirements and knowledge and thus improve the system as a whole. At the beginning, the main field where the ARDA group collaborated with other sciences are in participating to a common testing effort of the gLite middleware. In this phase, there is a major collaboration in the field of grid databases. The AMGA system is being evaluated with encouraging results by several users' group in EGEE, notably the Gilda team (EGEE Generic application support), the biomedical community (Medical Data Management working group) and by Earth Observation groups (ESR and UNOSAT).

5 Conclusions

The different prototype activities in ARDA, together with other activities within the experiments, are converging on a first version of the distributed-analysis systems, which will be used in the first phase of LHC operation. In the second phase of EGEE we expect to streamline this activity to further support the experiments' systems for both production and analysis. This means also continuing to influence the evolution of the middleware and the Grid infrastructure, using larger-scale experience and fostering the contacts with non-HEP scientists established during the first phase.

6 Acknowledgements

We would like thank EGEE middleware developers and the Distributed Analysis teams within the LHC experiments for their excellent collaboration.
This work received support by the Federal Ministry of Education and Research (Bundesministerium fuer Bildung und Forschung), Berlin, Germany.

References

1. A Realisation of Distributed Analysis for LHC, `http://cern.ch/arda`
2. Enabeling Grids for ESciencE, `http://cern.ch/egee`

3. PROOF, The parallel ROOT facility, `http://root.cern.ch/root/PROOF.html`

4. S. Gonzalez de la Hoz et al., *Distributed Analysis Jobs using the ATLAS production system*, this proceedings.

5. ARDA/gLite Middleware Activity, `http://cern.ch/arda`

6. ARDA Support for CMS Analysis Processing, `http://cern.ch/arda`

7. Karl Harrison et al., *GANGA*, this proceedings.

8. C. Munro, *Performance Comparison of the LCG2 and gLite File Catalogues*, Proc. of Advanced Computing and Analysis Techniques (ACAT 05), DESY, Zeuthen, Germany, May 2006.
   C. Munro, B. Koblitz, N. Santos and A. Khan, *Performance Comparison of the LCG2 and gLite File Catalogues*, IEEE Nuclear Science Symposium, October 23 - 29, 2005, Puerto Rico.

9. N. Santos and B. Koblitz, *Metadata Services on the Grid*, Proc. of Advanced Computing and Analysis Techniques (ACAT 05), DESY, Zeuthen, Germany, May 2006.