ARDA EXPERIENCE IN COLLABORATING WITH THE LHC EXPERIMENTS

M. Lamanna, CERN, Geneva, Switzerland
on behalf of the LCG ARDA project

Abstract

The ARDA project focuses in delivering analysis prototypes together with the LHC experiments. Each experiment prototype is in principle an independent activity but commonalities have been observed. The first level of commonality is represented by mature projects which can be shared effectively across different users. The best example is Ganga, providing a toolkit to organize users' activity, shielding users from execution back end details (like JDL preparation) and optionally actively supporting the execution of user application built with the experiment computing framework. The second level derives from the observation of commonality among different usage of the Grid: efficient access to resources from individual users, interactivity, robustness and transparent error recovery. High-level services built on top of a baseline layer are frequently needed to fully support specific activities like production and users' analysis. These high-level services can be regarded as prototypes of future general-use services. The observed commonality and concrete examples of convergence in the HEP community and outside are shown and discussed.

INTRODUCTION

The ARDA project (A Realisation of Distributed Analysis for LHC) [1] was created within the LCG project [2] in 2004 to develop prototypes of grid analysis system for the experiments at the Large Hadron Collider (LHC). The guiding idea behind ARDA was that by exploring the opportunities and the problems encountered in using the grid for LHC analysis, it would provide key inputs for the evolution of gLite the EGEE [3] middleware.

Enabling a large, distributed community of individual users 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.

Production can be regarded as a single-user activity with the emphasis is to maximize the CPU utilisation over given periods of time (typically several weeks or more). In this case, all activity should be carefully logged.

In the analysis environment, multiple users compete for resources and the (initial) latency between issuing a task and the availability of the first (partial) results becomes important and determines the number of iteration a user can achieve in a working day. Often, the analysis work will be limited by I/O capabilities more than CPU power, like in scanning analysis events or performing further event selection.

ARDA started in April 2004 at the same time with the EGEE project, which aims to provide a dependable Grid infrastructure for a variety of users from different scientific domains. In EGEE, the LHC experiments as well as a wide biomedical community have the key role of driving the evolution of the infrastructure. At the same time they are the major users of the computing infrastructure, for example with the LHC experiments' data challenges. The LHC experiments contribute to the Grid evolution through initiatives like ARDA (jointly funded by LCG and EGEE), which provides the opportunity to explore new ideas and to prototype advanced services influencing and frequently leading the development of innovative applications and services.

The understanding of analysis activities in the LHC era is still evolving very fast, and so ARDA needed to retain full flexibility. A sound approach to such an evolving situation is to prototype the future systems together with the users, exposing them to the grid environment and discussing the evolution on the basis of their experience. 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

Testing the Grid under real conditions gives effective feedback to the developers of Grid middleware. Within EGEE, ARDA played a key role in testing the middleware and having access to "previews" of gLite components. Progressively this activity has 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 [4] (ARDA Metadata Grid Application) made its way into the gLite middleware and is now also used by non-HEP applications. Another example is the detailed studies of the performance of the gLite workload management system performed together with the ATLAS experiment.

DISTRIBUTED-ANALYSIS PROTOTYPES

From the beginning, it was decided to agree with each LHC experiment a-priori independent prototype activities: it was considered unrealistic to force at an early stage commonality in the use of tools, since each experiment

has different physics goals, data organization and analysis 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.

LHCb

The ARDA-LHCb prototype activity is focusing on the Ganga [5] system (a joint ATLAS-LHCb project). The main idea behind Ganga is that the physicists should have a uniform and effective interface to their analysis programs. Ganga allows to prepare applications, 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. Making the transition across these environments is a key element to improve the productivity during the development, debugging and analysis phases.

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 combines the consistency and flexibility of the programming interface with intuitive and concise usage. The GPI is used directly by the end-users in the interactive command line shell. It may be used for writing complex user-specific scripts steering mid-sized productions, controlling user-defined workflows and adding user-specific components like data merging, error recovery and retry etc... It allows the embedding of the Ganga Core as a library in a third-party framework.

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 workload management system), the ATLAS production system plus several local-batch systems (LSF, PBS, etc…) as well as local execution.

Ganga is an open-development framework which fully exploits the plug-in architecture. This makes the integration of new application support and execution back ends very easy. Application support is available for ATLAS and LHCb (ATHENA and GAUDI applications). Other applications, such as Geant4 simulations in medical physics, or the BLAST protein alignment algorithm in biotechnology have been successfully run with Ganga.

As an example of the Ganga functionality, each user has full access to the jobs submitted in the past including their configuration and input/output. These data may be stored in a local repository on disk for disconnected operations and in a remote server providing roaming functionality. The current implementation of the remote repository is based on the AMGA [4] database interface. Ganga also offers automatic job status monitoring and output retrieval. Automatic job splitting allows a very efficient handling of large numbers of similar jobs using different datasets. Job templates provide a convenient mechanism to support repetitive tasks.

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 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 GUI produces editable files which can be modified and embedded in other applications).

CMS

The ARDA-CMS activity started with a comprehensive evaluation of gLite and the existing CMS software components. Eventually ARDA developed a full end-to-end prototype called ASAP [6] prototyping some advanced services such as the Task Monitor and the Task Manager.

The Task Monitor gathers information from different sources: MonALISA (MONitoring Agents using a Large Integrated Services Architecture) [7] (mainly collecting run time information via the CMS C++ framework COBRA); the CMS production system; Grid-specific information (initially the gLite/LCG logging and bookkeeping or the gLite/R-GMA system).

The Task Manager implements CMS specific strategies, using of the Task Monitor information. The Task Manager understands the user tasks (normally one user-defined executable reading a set of data files) and organizes them to enable the user to delegate several tasks, e.g. the actual submission (the user registers a set of tasks and then disconnects: the jobs are submitted “in background” without further user intervention) and (automatic) error recovery. Some key components of this very successful prototype, which incorporated a lot of user feedback, are now being migrated within the official CMS system CRAB (CMS Remote Analysis Builder) in the framework of the CMS taskforce.

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, selecting specific interval of times, sites, group of jobs, sets of data etc…

The system provides the necessary level of interactivity so that the users would not be just warned in case of problems but could interact with the Dashboard to find the origin of the problem. Information collected in the dashboard data base is retrieved from two main sources: R-GMA for Grid logging and bookkeeping data and Monalisa for more CMS specific data, like application behavior and data distribution.

In addition, the dashboard should be a powerful tool to monitor the experiment activity and collect information to improve the grid infrastructure as well as to pin down the most frequent (and critical from a user point view) errors.

The Dashboard project is now also part of the ARDA/ATLAS activity and first prototypes will appear soon. Using the know-how and the experience of the CMS prototypes, the ATLAS Dashboard will provide similar services within the ATLAS environment.

ATLAS

The ARDA activity within ATLAS started as a contribution to the DIAL system [8], an ATLAS analysis service which has been used in connection with the early prototypes of the EGEE middleware. Eventually, the activity moved to investigate the possibility of submitting analysis jobs via the ATLAS production system [9] and contributing to the Ganga project.

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

Ganga provides the user interface for development and job execution across different back end. The ARDA ATLAS effort concentrated on a better integration in the ATLAS environment. As in LHCb, the model is that Ganga supports the user during the application development and testing, from jobs being executed locally through batch submission to grid submission. The analysis steps requiring full logging and effective usage of very large resources can be also steered by Ganga using the production system as back end.

The ATLAS ARDA team is a main contributor (within different EGEE working groups) to activities related to the gLite Workload Management System (WMS). On one side, ARDA has contributed significant effort in testing, measuring and fine-tuning the new WMS. Detailed studies of the behaviour of this system under high load have contributed a number of observations to improve this system and eventually incorporated in the latest gLite version (available via the ARDA web site). In addition, ARDA is leading the working group providing recommendations to ensure effective coexistence of different type of jobs (long production jobs and relatively short analysis tasks). The final model is being worked out and it will be tested in the near future.

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 (Parallel ROOt Facility) [10] system 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. This component optimises 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). This component is now in production in the ALICE 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.

The ARDA ALICE group has also developed a high-performance authorization scheme integrating an ACL system (kept in the AliEn catalogue) and the performing data serving stage based on xrootd. The authorization framework has been presented at the Grid 2005 conference and an IEEE paper will be published soon [11] and it is currently in use in the ALICE system.

CONTRIBUTIONS TO THE EGEE MIDDLEWARE

In the ATLAS section we have already mentioned the contributions to the evaluation of the gLite WMS.

Similar activities on different middleware components have been performed, for example to evaluate the data management components of the LCG and gLite middleware. A series of tests were conducted using multi-threaded clients to imitate many simultaneous connections and current activity. Results have been discussed in several occasions; see for example [12].

R-GMA, the gLite information system, was used inside the CMS Dashboard. This activity provided useful feedback to the developer of the system.

A special case of the ARDA activities related to the middleware is the AMGA (ARDA Metadata Grid Application) catalogue.

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.

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. The usage of a sophisticated session management allows to minimize the impact of the long roundtrip time. We also validated the ACL-based security of AMGA.

COLLABORATION WITH OTHER SCIENCES

ARDA is a very important actor within the EGEE project. It has a leading role in the exchange of ideas and experiences with groups supporting other 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.

Initially, the main field of collaboration of ARDA with other sciences was in participating to a common testing effort of the gLite middleware. Now, 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 (e.g. the UNOSAT project).

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 continue to collaborate with the experiments to streamline these activities 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.

ACKNOWLEDGEMENTS

We would like to thank the LCG and EGEE projects for support and useful discussion (in particular the CERN IT/GD and IT/PSS groups and the gLite team). We acknowledge a very fruitful collaboration with the MonAlisa team. A special thank you is due to the entire IT/PSS/ED section for fruitful collaboration and stimulating discussion.

This work was performed within the EGEE project, which is funded in the European Commission under contract INSO-RI-508833. It also received support from Federal Ministry of Education and Research (Bundesministerium fur Bildung und Forschung), Berlin, Germany.

REFERENCES

[1] ARDA (A Realisation of Distributed Analysis for LHC), http://arda.cern.ch

[2] LCG (LHC Computing Grid), http://cern.ch/lcg

[3] EGEE (Enabling Grids for ESciencE), http://cern.ch/egee

[4] N. Santos and B. Koblitz, “Metadata Services on the Grid”, Nucl. Instr. and Methods A559 (2006) 53; B. Koblitz et al., ”The AMGA Metadata Service in gLite”, these proceedings

[5] Karl Harrison et al., “Ganga: a Grid User Interface”, these proceedings

[6] ASAP Support for CMS Analysis Processing, http://www-asap.cern.ch; J. Andreeva et al., “CMS/ARDA activity within the CMS distributed computing system”, these proceedings

[7] MonALISA (MONitoring Agents using a Large Integrated Services Architecture), http://monalisa.caltech.edu/monalisa.htm

[8] D. Adams, “DIAL: Distributed Interactive Analysis of Large Datasets”, these proceedings

[9] S. Gonzalez de la Hoz et al., “Distributed Analysis Jobs using the ATLAS production system”, included in these proceedings; D. Liko, “The ATLAS Strategy for Distributed Analysis on several Grid Infrastructures”; these proceedings

[10] PROOF, (Parallel ROOt Facility, http://root.cern.ch/root/PROOF.html

[11] D. Feichtinger and A. Peters, “Authorization of Data Access in Distributed Storage Systems”, SC05 Grid 2005 - 6th IEEE/ACM International Workshop on Grid Computing, November 13-14, 2005, Seattle

[12] C. Munro and B. Koblitz, “Performance Comparison of the LCG2 and gLite File Catalogues”, IEEE Nuclear Science Symposium, October 23 - 29, 2005, Puerto Rico