PRACTICAL EXPERIENCE IN ARDA WITH ATLAS SOFTWARE AND THE PROTOTYPE FOR THE EGEE* MIDDLEWARE - gLite

21. September 2004

The ARDA team,

Julia Andreeva[a], Tao-Sheng Chen[b], Andrey Demichev[c], Derek Feichtinger[d], Juha Herrala[e], Birger Koblitza†, Massimo Lamanna[a], Dietrich Liko[a], Andrew Maier[a], Kuba Moscicki[a], Frederik Orellana[f‡] , Andreas Peters[a], Victor Pose[g], Wei-Long Ueng[b]

and

David Adams[h]

ATL-SOFT-2004-002
18 October 2004

Abstract

The ARDA project aims to provide end-to-end systems for physics analysis for the LHC experiments. In collaboration with the experiments, prototype systems are being developed that are based on the experimental software and on the middleware under development within the EGEE project – gLite. The strategy of the ATLAS experiment is to develop high level services that supports physics analysis and simplifies grid usage for the physicist. A first implementation of such a service, DIAL, has already been developed and interfaces to conventional batch systems such as LSF or Condor. In this report first experiences using ATLAS software with the gLite prototype are described. The integration of ATLAS software components with the prototype is discussed. Some aspects of job execution, data management, metadata and Software installation are considered. A first implementation of a DIAL service based on the middleware prototype is demonstrated.

[a] CERN, Geneva, Switzerland
[b] Academica Sinica Computer Center, Nanking, Taiwan
[c] Moskow State University, Moskow, Russia
[d] Paul Scherrer Institute, Villigen, Switzerland
[e] Forschungszentrum Karksruhe, Karlsruhe, Germany
[f] University of Geneva, Geneva, Switzerland
[g] Joint Institute for Nuclear Reasarch, Dubna, Russia
[h] Brookhaven National Laboratory, Brookhaven, USA

* EGEE is a project funded by the European Union under contract IST-2003-508833
† Funded by Bundesministerium für Bildung und Forschung, Berlin, Germany
‡ Supported by EU COST Action 283

1. INTRODUCTION

1.1. The ARDA project

The mandate of ARDA [1] is to deliver four grid based prototype analysis systems, one for each experiment. These prototypes will integrate the next generation middleware with common HEP tools and the analysis software of the experiments. An essential goal of the project is the validation of these systems including users from the experiments: end-to-end prototyping going beyond mere demonstrations. The prototyping phase is to evolve within a limited time into distributed analysis services with enhanced middleware and applications software suitable for sustainable distributed analysis services for the four experiments deployed at the Regional Centers.

For ATLAS the prototype is based on DIAL [2], a PPDG/ATLAS project to develop software and demonstrate the feasibility of distributed interactive analysis of large event data samples.

1.2. The reengineered middleware

The ARDA prototypes will be based on the next generation GRID middleware under development in the JRA1 group of the EGEE project – gLite [3,4].

The design of this middleware is based on following principles

- Lightweight approach, based on existing services
- Interoperability (multiple implementations possible)
- Resilience and Fault Tolerance
- Co-existence with deployed infrastructure (e.g. LCG-2, Grid3, NorduGrid)
- Service oriented approach (Web Services)

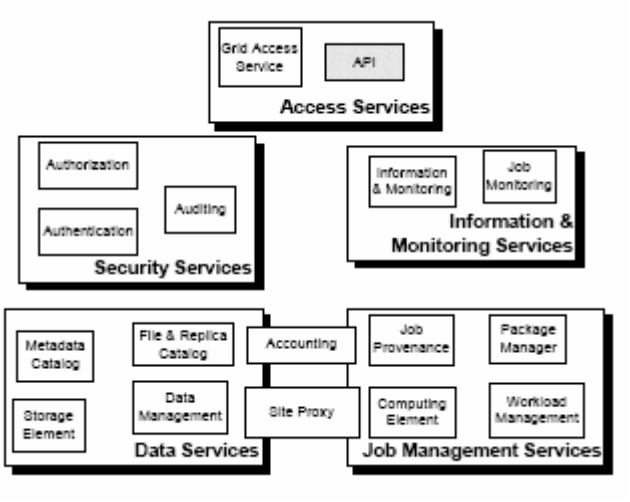

Figure 1: gLite Services thematically group min 5 service groups

First versions of an architecture [5] and a design [6] document have been presented and comments are welcome. The gLite Grid services follow a Service Oriented Architecture which will facilitate interoperability among Grid services and allow easier compliance with upcoming standards, such as WSRF, that are also based on these principles. The architecture constituted by this set of services is not bound to specific implementations of the services. Although the services are expected to work together in a concerted way in order to achieve the goals of the end-user, they can be deployed and used independently, allowing their exploitation in different contexts. Figure 1 depicts the high level services, which can thematically be grouped into 5 service groups.

To structure the interaction with the experiments it is channeled through ARDA and the experiment contact person[§].

1.3. The gLite prototype

On May 18th 2004 a middleware prototype installation was delivered by EGEE and the ARDA group started to evaluate its status. This middleware prototype has been delivered at a very early stage of the EGEE project. While the system is already a fully functional grid system, not all components have their final form and we expect a number of iterations. A number of services that have been outlined in the design and architecture documents are not yet delivered. On the other hand it demonstrates the wish of the EGEE development team to have strong interactions with the future users of the system.

In the first month of operation the system was still plagued by instabilities and complications. These problems were addressed in several iterations and the system is right now in a stable state. In several aspects this prototype still resemble the original implementation of the Alien grid middleware, but work is ongoing to improve components following the architecture and the design plans. The status of the prototype was reviewed in collaboration with the experiments in the 2nd ARDA workshop in June [7].

It has also to be pointed out that in the current stage the prototype is targeted more at development than at production. The prototype is actually shared between developers and users. The available resources in terms of CPUs and disk space are relatively moderate. Nevertheless, the installation

[§] In case of ATLAS the contact person is David Adams (dladmas@bnl.gov).

includes two sites, CERN and University of Wisconsin - Madison. To fulfill the needs of ARDA much more resources are required. It is foreseen that the prototype will move into an operational phase on the LCG preproduction system in the coming months.

2. THE ATLAS DISTRIBUTED ANALYSIS STRATEGY

ATLAS plans to provide high level services to support physics analysis. The DIAL project has delivered a first implementation of such a service based on conventional batch systems. The ATLAS physicist can require the transformation of a dataset by specification of an application and a task.

Applications have to be provided to DIAL to allow processing of the datasets. Typically such an application could be the experiment reconstruction program, Athena or other analysis frameworks (ROOT, PAW etc). The application has to be embedded into a DIAL specific environment.

A *Task* allows the user to extend the application in an application specific way. In case of PAW this could be a FORTRAN program to analyze Ntuple data, in case of Athena this could be a user algorithm.

In Figure 2 major DIAL components and their interaction are shown. In praxis the physicist uses root or the command line to assemble a job by specification of an Application, a Dataset and a Task. The job is submitted to a DIAL server that supervises the processing. This processing includes splitting into many sub jobs on the underlying system (computer cluster or grid) and merging of the results. Results are returned to the user in an interactive way. In all stages the physicist uses DIAL to interact with the system and is shielded from the complexity.

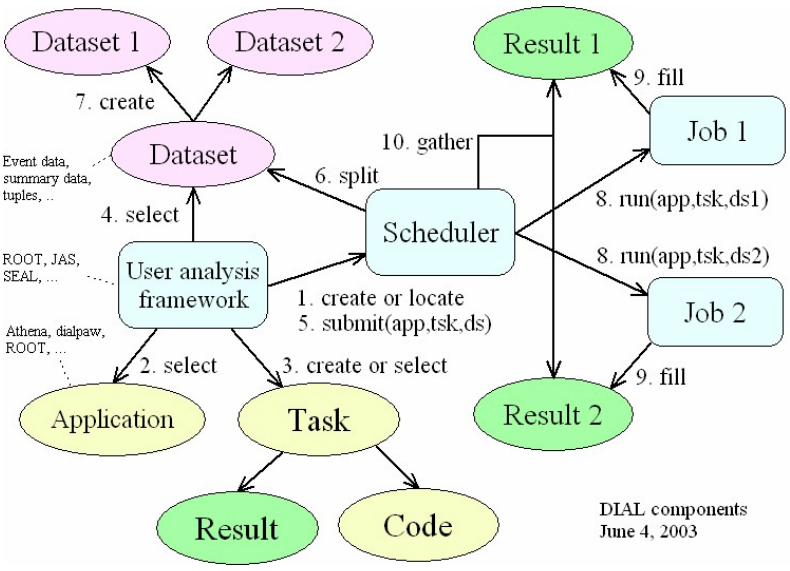

Figure 2: Major components of DIAL and their interactions

In the current status the DIAL service interfaces to batch environments like LSF or CONDOR. The servers are operated at BNL, but are available to all ATLAS physicists. For more details see the DIAL web pages [2].

3. PRACTICAL EXPERIENCE

After the initial instabilities of the system have been addressed the gLite prototype provides a stable environment. Actually ARDA is periodically monitoring the status of system (see Figure 3). The system has been used to study various aspects of integration with the experimental software. These aspects are discussed in turn.

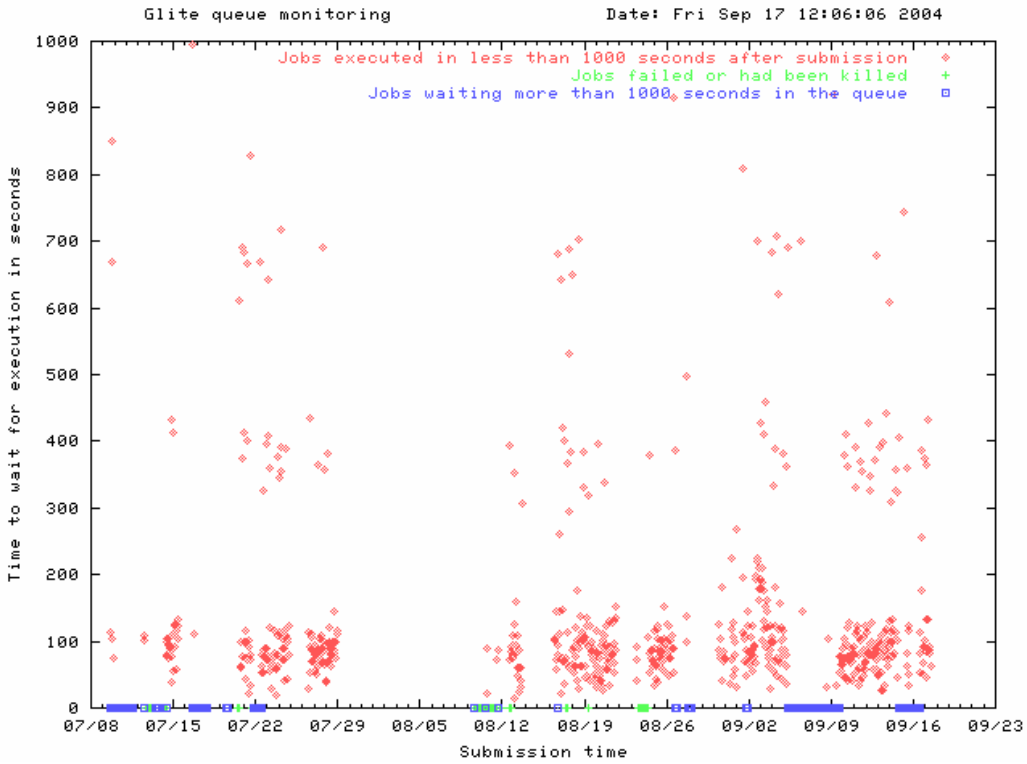

Figure 3: gLite queue monitoring by ARDA [8].

3.1. Running Athena jobs

The prototype user interface presents the grid system as a virtual computer. When the user logs into the system he/she is presented with a familiar environment with a virtual file system. Most aspects of the system can be accessed by familiar UNIX commands.

It is straight forward to register a script to run standard applications such as a standard reconstruction program (e.g. RecExCommon).

To support more sophisticated analysis we have developed a script that wraps up the user development area and ships it to worker nodes. Only small modifications of standard scripts are necessary to integrate Athena with the system. This provides the means to perform arbitrary analysis tasks on the system. The task can be customized by modification of the job submission parameters in the JDL file (see Example 1).

```
Executable="Athena-on-gLite.sh";
Arguments="RecExCommon_topOptions.py";
InputFile={"/egee/user/d/dliko/data/testsample.dat"};
InputData={"PF:file://pcarda01.cern.chafs/cern.ch/user/l/liko/gLite/RecExCommon-001.tar.gz"};
OutputData={ "ntuple.root", "histo.hbook" };
Email="Dietrich.Liko@cern.ch";
```

Example 1: Typical JDL file to submit an Athena job

3.2. Data management

Data management in the gLite prototype is presented to the user by means of the virtual file system. All details of file catalogs, replica location services and the actual data transfer, SRM, is hidden from the user. Advanced features, such as POSIX file access are not yet delivered.

In the actual prototype, data at CERN can be accessed from CASTOR. It is possible to register files already present in CASTOR and to add new files in a dedicate area. ATLAS LFNs can be mapped into the file system in a straight forward way. Therefore, in a first iteration, data in CASTOR can be made accessible.

In the longer term it is essential to integrate gLite with the ATLAS file management based on Don Quijote [9]. This would allow the transfer of data between gLite and the other grids, a feature that is considered essential for ATLAS. First attempts to interface Don Quijote to gLite are encouraging.

3.3 Metadata

In the gLite architecture metadata is considered to be application specific[**]. The responsibility for a metadata catalog lies with the experiments. For an end-to-end analysis system the integration with the experimental metadata is fundamental.

In the context the Web Service interface of AMI, the ATLAS Metadata Interface, was of special interest. ARDA performed several test of its performance under a load that might be expected in an analysis environment (see Figure 4). These studies were performed in direct contact with the AMI development team. Several aspects of the observed performance could be understood [10].

[**] gLite has the possibility to store file metadata.

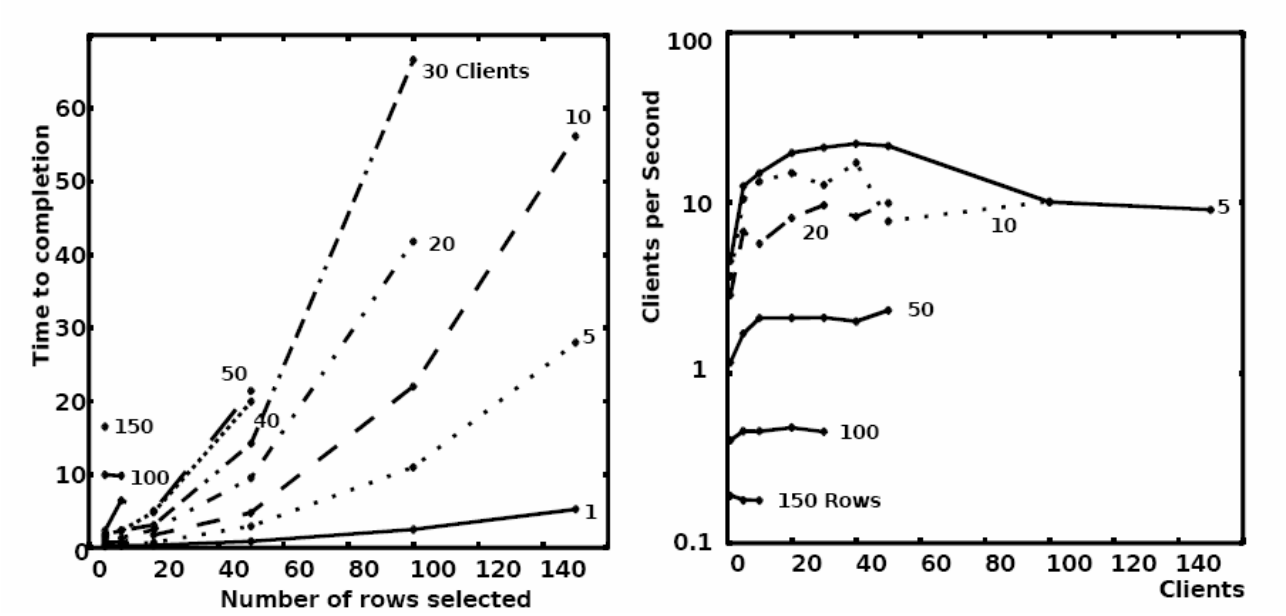

Figure 4: In the left figure the time to complete a request of several clients in parallel is shown as a function of the number of rows selected from the database and the number of clients. In the right figure the number of requests of several clients in parallel per second is shown as a function of the number of rows and the number of clients.

As a general question, the suitability for Web services to retrieve large datasets was discussed. This is an aspect relevant to all experiments and is still under discussion.

3.4. Package Management

The prototype provides a simple system, called PackMan that provides the means to install software at remote sites. An important aspect is the service character of the feature, e.g. it is the role of the system to decide if it better to run the job at a site that has the software already installed or to install the software at an additional site.

On a first view the system seems similar in functionality to Pacman [11], the package manager of the ATLAS distribution kits. A fundamental difference is the installation strategy: Pacman is installing the software in a common installation area, while in PackMan each package has a separate area. In the first approach binaries and libraries can be installed in a common directory, in the second the binaries and libraries of each package is in separate places. The first system has a clear advantage with respect to the program environment (PATH and LD_LIBRARY_PATH environment variables), the second makes management of a package a much simpler operation (consistency, removal etc).

A possible solution for the current prototype is to use Pacman under the control of PackMan. A virtual PackMan package depends in turn on Pacman to install a full ATLAS release. First experiments are encouraging. It is even imaginable to combine such a step with a validation of the installation, as performed by ATLAS. This would provide a first possibility to install required ATLAS software in the granularity of full releases.

This approach has also its clear limitations, as many Pacman features are not available. It has to be pointed out that architecture and design document are very limited with respect to package management. The experiments are invited to address that point to define an improved solution for the future.

3.5. Interfacing DIAL with gLite

The DIAL server has been adapted to the CERN environment and an experimental service has been installed at CERN.

A first implementation of a gLite scheduler has been developed. As the future C++ API of gLite has not yet been delivered, the implementation is using Perl scripts provided by gLite. The script sends a job request from the dial service to the prototype and monitors the job execution. At this point AFS is used internally for communication between the worker node and the client.

The interface has been successfully used to perform the dial demos, as described on the web pages. A number of issues have been identified and work is in progress to study these aspects (e.g. internal communication, data access, authentication with gLite, etc.).

It is planned to evolve the implementation following closely the development of the prototype. The architectural aim is to implement a scheduler that interacts directly with the underlying middleware services using Web Services.

Apart from that is has to be stated that the current size of the test bed cannot compete with the BNL installation available to DIAL. Please feel free to contact us if you want to use our experimental server (http://lxb0712.cern.ch:6194). For larger request you will have to use the servers at BNL.

4. SUMMARY AND CONCLUSIONS

First practical experiences with ATLAS software on the gLite test bed have been presented. At this early stage of the development not all promised features have been provided and the test bed has not yet reached a sufficient size to allow ARDA to fulfill its mandate.

Nevertheless the experience has been very positive: the system evolved fast to a stable platform and we found it very easy to use.

Using the available functionality we have successfully interfaced DIAL to the prototype. No major showstoppers have been identified. Following the evolution of the test bed, this activity will continue and the interface will be iterated to deliver the ATLAS prototype.

The ARDA team is composed of participants from all experiments. While each experiment has its own prototype we have found much communality. This has led to a stimulating exchange of ideas between the participants from different experiments.

We would like to use this opportunity to remind you of the next ARDA workshop, October 20 to 22 at CERN. At this occasion the full community is invited to interact with the development team.

We also encourage physicists from the ATLAS community, who would like to move their analysis to the GRID, to join us. Your input is essential to deliver a successful system for ATLAS physics analysis.

ACKNOWLEDGMENTS

The authors would like to thank the gLite development team for their excellent work and congratulate them to the progress we have observed. We appreciate their patience to discuss with us all aspects of the system.

REFERENCES

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

[2] DIstributed Analysis of Large datasets, DIAL,
http://www.usatalas.bnl.gov/~dladams/dial

[3] Enabling Grids for E-science in Europe, EGEE,
http://public.eu-egee.org

[4] Lightweight Middleware for Grid Computing, gLite,
http://cern.ch/glite

[5] DJRA1.1 – Architecture and Planning,
https://edms.cern.ch/document/476451/

[6] DRJA1.2 - Middleware Design,
https://edms.cern.ch/document/487871/

[7] ARDA workshop “The first 30 days (of EGEE middleware)”, 21-23 June 2004 at CERN,
http://cern.ch/lcg/PEB/arda/workshops/june04.htm

[8] gLite queue monitoring by ARDA,
http://cern.ch/test-glite/Plot.html

[9] Don Quijote – ATLAS Production Data Management System,
http://cern.ch/Miguel.Branco/cern/donquijote

[10]The ARDA Group (ed. B. Koblitz), Report on the Atlas Metadata Catalogue AMI,
http://cern.ch/lcg/PEB/arda/public_docs/CaseStudies/ami_new.pdf

[11] Pacman – a package manager,
http://physics.bu.edu/~youssef/pacman/index.html