ICATPP Conferences

VILLA OLMO (COMO-ITALIA)

10th ICATPP Conference on
Astroparticle, Particle, Space Physics, Detectors and Medical Physics Applications

Villa Olmo, 8-12 October, 2007
PAPER SUBMISSION FOR WORLD SCIENTIFIC PROCEEDINGS

AFTER THE ACCEPTANCE FOR ORAL PRESENTATIONS IN PLENARY OR PARALLEL SESSIONS OR FOR THE POSTER SESSION, THE CORRESPONDING AUTHOR MUST DELIVER HIS/HER PAPER AT THE REGISTRATION DESK IN VILLA OLMO

IMPORTANT:
EACH PAPER MUST BE WRITTEN FOLLOWING THE INSTRUCTIONS AND USING THE STYLES FOR WORLD SCIENTIFIC PROCEEDINGS AVAILABLE AT:
http://villaolmo.mib.infn.it/Proceedings_2007.html
EACH PAPER MUST BE DELIVERD ON A CD-ROM WITH THIS FORM

Paper Title: Experience with Atlas Distributed Analysis Tools
Session: Poster session

PRESENTING (AND REGISTERED) AUTHOR'S NAME AND EMAIL: Santiago González de la Hoz (Santiago.Gonzalez@ific.uv.es)
Note:

EXPERIENCE WITH ATLAS DISTRIBUTED ANALYSIS TOOLS*

SANTIAGO GONZALEZ DE LA HOZ, LUIS MARCH RUIZ†

IFIC – Instituto de Física Corpuscular, Centro Mixto Universitat de València – CSIC, Valencia, Apartado de Correos 22085, E-46071, Spain

DIETRICH LIKO

CERN – European Organization for Nuclear Research, CH-1211 Geneva, CH-1211, Switzerland

The ATLAS production system has been successfully used to run production of simulation data at an unprecedented scale. Up to 10000 jobs were processed in one day. The experiences obtained operating the system on several grid flavours was essential to perform a user analysis using grid resources. First tests of the distributed analysis system were then performed. In the preparation phase data was registered in the LHC File Catalog (LFC) and replicated in external sites. For the main test, few resources were used. All these tests are only a first step towards the validation of the computing model. The ATLAS management computing board decided to integrate the collaboration efforts in distributed analysis in only one project, GANGA. The goal is to test the reconstruction and analysis software in a large scale Data production using Grid flavors in several sites. GANGA allows trivial switching between running test jobs on a local batch system and running large-scale analyses on the Grid; it provides job splitting and merging, and includes automated job monitoring and output retrieval.

1. Introduction

The primary goal of the distributed analysis is to bring computation power to individual ATLAS physicists. This is achieved by providing easy access to the computing resources of the various Grids, in a way that hides most of the complexities of Grid environments.

The distributed analysis model is included on the ATLAS computing model [1] and stipulates that data is distributed in various computing facilities. User jobs are in turn routed depending on the availability of relevant data. A typical analysis job consists of a Python [2] script that configures and executes a user defined algorithm in Athena (ATLAS software framework). The script specifies

* This work is supported by the Spanish National Research Council (CSIC)

† Work partially supported by Marie Curie grant MERG-CT-2006-44258 of the European Union.

the input data and produces one or more files containing plots and histograms.
The expected volume of data recorded for offline reconstruction and analysis will be of the order of 1 PB (10^{15} bytes) per year. Due to the size of this expected data volume it is necessary to use distributed resources all over the world to perform reconstruction and analysis of the data.
The ATLAS Computing Model covers all aspect of this operation. It includes organized production of simulated data, and also user analysis. In this paper we describes our experience running ATLAS distributed analysis tools.
The ATLAS production system has been developed to perform the production of the experiment. It provides a robust framework to execute a large number of jobs in the grid infrastructures.
This experience will allow us to compare the execution of analysis task using such a system versus using direct submission to the infrastructure. This activity was part of the Data Challenges (DC's) that were organized to validate the Computing Model to ensure the correctness of the technical choices.
The collaboration decided to perform these DC’s in the context of the LHC Computing Grid project, LCG [3], which contains the majority of ATLAS resources, but also to use both the middleware and the resources of two other Grid projects, OSG [4] and NorduGrid [5]. The aim is to prepare the computing infrastructure for the simulation, processing and analysis of the LHC data.

1.1. *Atlas Production System*

In order to handle the task of DC's an automated system was designed figure 1.

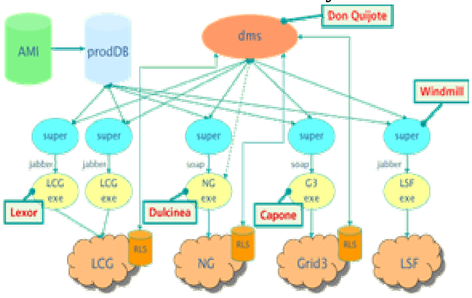

Figure1. Atlas Production System

The production system [6] is implemented in a modular way to enable ATLAS to use resources out of these three infrastructures. All jobs are defined in a specific schema and stored in a central database. A supervisor agent picks them up, and sends their definition as an XML message to the various executors. Executors are specialized agents, able to convert the ATLAS specific XML job description into a Grid specific language. Three executors were developed, for

LCG (Lexor and CondorG), NorduGrid (Dulcinea) and OSG (Capone). All the data management operations are performed using a central service, Don Quijote (DQ) [7]. DQ moves files from their temporary output locations to their final destination on some Storage Element and registers this location in the Replica Location Service of the corresponding Grid flavor. Thus all the copy and registration operations are performed through an abstraction layer provided by DQ. This allows operating the different replica catalogues of the three Grid flavors in a similar way.

The ATLAS production system has been successfully used to run production jobs at an unprecedented scale for a system deployed on about 100 sites around the world. On successful days there were more then 10000 jobs processed.

2. Distributed Analysis Strategy

ATLAS has adopted a multi-pronged approach to distributed analysis by exploiting its existing Grid infrastructure (see figure 2).

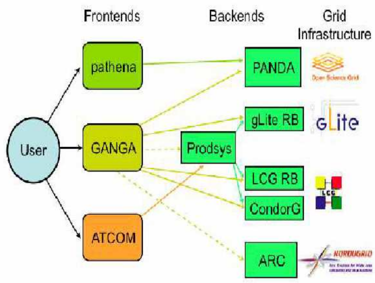

Figure2. Various front-end clients are intended to perform distributed analysis on the existing Grid infrastructure. These front-end are PanDA/Pathena, ATCOM and GANGA.

PanDA [8] is a job management system associated with OSG designed specifically for both distributed production and analysis. PanDA has native support for the ATLAS Distributed Data Management (DDM) system allowing accepting datasets as input and producing datasets as output. PanDA offers users a comprehensive system view presenting heterogeneous distributed resources as a single uniform resource, accessible via a standard interface. It also has extensive web-based job monitoring and browsing capabilities. Panda looks to GANGA [9] in order to provide a graphical job definition and submission interface. Pathena is a python script designed to submit analysis jobs to the PanDA system involving an optional build step (user code can be compiled) followed by an execution step.

ATCOM [10] was the dedicated graphical user interface front-end to the production system, designed to be used by a few expert users involved in large-scale organized production of data. It had the potential to be used for distributed analysis purposes as well. The ATLAS management computing board decided to integrate the collaboration efforts in distributed analysis in only one project, GANGA. It is a powerful user friendly front-end tool for job definition and management. GANGA provides distributed analysis users easy access to the whole Grid infrastructure. It currently provides two user interface clients: a Command Line interface (CLI) and a Graphical User Interface (GUI). GANGA allows switching between testing on a local batch system and large-scale processing on the Grid, and helps to keep track or results.
A job in GANGA is constructed from a set of building blocks (figure 3). All jobs have to specify the software to be run (application) and the processing system (back-end) to be used.

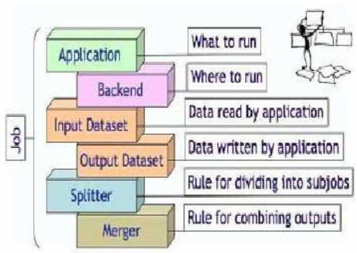

Figure3. Building blocks for constructing a GANGA job

2.1. *Distributed analysis using the production system and using GANGA*

Distribution of data on several sites and local access to the data is a very important issue to minimize failures. In total 155 GB of merged datasets were used for distributed analysis. The dataset were registered at CERN in Logical File Catalog (LFC) and were replicated in the sites shown in Table 1.

Table 1. Sites where datasets were replicated

Site	Storage Element	Computing Element
IFIC	castorgrid.ific.uv.es	lcg2ce.ific.uv.es
Sinica	Lcg00123.grid.sinica.edu.tw	lcg00126.grid.sinica.edu.tw
Cnaf	grid007g.cnaf.infn.it	gridit-ce-001.cnaf.infn.it
PIC	castorgrid.pic.es	ce01.pic.es
MI	t2-se-01.mi.infn.it	t2-ce-01.mi.infn.it
Cern	castorgrid.cern.ch	lxgate13.cern.ch
RO	T2-se-01.roma1.infn.it	t2-ce-01.roma1.infn.it

.

The algorithm of choice has been a $Z_H \rightarrow$ttbar, a heavy Z decaying into tops in the Little Higgs model. A total of 400 Analysis Object Data (AOD´s) were produced using the Athena full simulation chain, each one containing 50 events. The analysis has been performed using the production system and GANGA.

Despite the possibility to run analysis jobs via the production system, not all functionalities to support distributed analysis were currently available. In the following, the technical issues that had to be addressed are discussed in turn.

A dedicated database was setup for analysis jobs to separate private work from the ongoing production. A generic analysis transformation was created, that compiles user code or the user package on the worker node, processes AOD input files and produces histograms as output. ATCOM was used to define jobs. It was also used to monitor the status.

To perform an analysis the user has to define a task and associated jobs according to the conventions of the ATLAS production system. The task contains summary information about the jobs to be run (input/output datasets, transformation parameters, resource and environmental requirements), while individual jobs are defined by their specific parameters needed for execution.

We defined the same jobs using GANGA. It provides a set of ATLAS-specific features such as application configuration based on the Athena framework and input data location based on DDM. It can be run either on the command line, with Python scripts or through a GUI. Users need to enter just a few commands to set application properties and submit jobs to run the application on selected back-ends (Grid flavours or local batch system).

In both cases each job produced three output files (ntuple, histogram and log) stored on Castor based Storage Elements (SE). The jobs were running at several sites and were instructed to save output at one single Storage Elements close to the user. The ROOT package was used to merge the histogram output files and to analyze the results. Finally, after merging, figure 4 shows the analysis result.

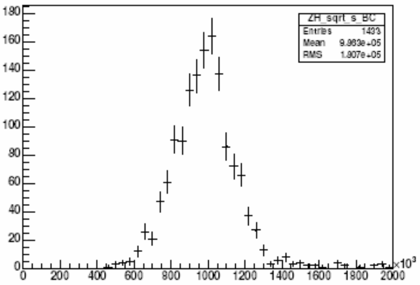

Figure 4: $Z_H \rightarrow tt$ invariant mass distribution after merging the output files (GeVunits in the y axis).

All these tests are only a first step towards the validation of the computing model and distributed analysis. More realistic tests have to involve many physicists working in concurrent mode. This requires not only progress in the application, but also progress in the Grid middleware and the site configurations. An example is the ongoing discussion on job priorities in LCG which should allow the coexistence of production and user analysis activities.

3. Conclusions

The ATLAS production system and GANGA have been used to submit physics analysis jobs to LCG Grid flavor. Using the production framework for analysis has the advantage to profit from the experience of the large scale production. Only limited additional resources were necessary to perform the required modifications to support analysis.

For the main test few resources were used in nine sites. The system was able to process 10k events jobs in about 10 minutes. It is fair to say that it was difficult to achieve this performance due to the instability of the major components of the software that were still in a development phase. Nevertheless we consider this first test as encouraging and promising. With the startup of the LHC we expect much more data and resources for analysis will be only available using the grid.

Comparing the production system with GANGA, we could observe a more robust execution and we were also profiting from the advanced monitoring capabilities of the production system. A drawback is that such a system represents an additional infrastructure element, which has to be operated by the experiment.

Distributed Analysis is still work in progress. With the startup of LHC in the next year we expect a dramatic increase of the data volume. This will requires the general ATLAS user to use resources on the grid to perform his analysis.

References

1. ATLAS Computing Technical Design Report, CERN-LHCC-2005-022.
2. G. Van Rossum and F.L. Drake, Jr (eds.) Python Reference Manual 2.4.3
3. M. Lamana,et al,“The LHC computing grid project”, NIM **A 534** (2004) 1-6
4. R. Gardner, “Grid3,” in Proc. CHEP04, p 1318 (2 vol)
5 M.Ellert et al.,“The NorduGrid. Project”, NIM **A (502**) (2003) 407-410.
6. L. Goossens, “Production System in ATLAS DC2,” in Proc. CHEP04, p501.
7. M. Branco, “Don Quijote” CERN Yellow Report 2005-002, p 661.
8. PanDA (http://twiki.cern.ch/twiki/bin/view/Atlas/Panda).
9. GANGA (http://ganga.web.cern.ch/ganga)
10. ATCOM (http://uimon.cern.ch/twiki.cern.ch/twiki/bin/view/Atlas/AtCom).