ATL-SOFT-PROC-2011-011
13 January 2011

Reinforcing user data analysis with Ganga in the LHC era: scalability, monitoring and user-support.

Johannes Elmsheuser[1], **Frederic Brochu**[2], **Ivan Dzhunov**[3], **Johannes Ebke**[1], **Ulrik Egede**[4], **Manoj Kumar Jha**[5], **Lukasz Kokoszkiewicz**[3], **Hurng-Chun Lee**[6], **Andrew Maier**[3], **Jakub Mościcki**[3], **Tim München**[7], **Will Reece**[4], **Bjorn Samset**[8], **Mark Slater**[9], **David Tuckett**[3], **Daniel Vanderster**[3] **and Michael Williams**[4]

[1] Ludwig-Maximilians-Universität München, Germany
[2] University of Cambridge, UK
[3] CERN, Switzerland
[4] Imperial College London, UK
[5] INFN Bologna, Italy
[6] NIKHEF, Netherlands
[7] University of Wuppertal, Germany
[8] University of Oslo, Norway
[9] University of Birmingham, UK

E-mail: `johannes.elmsheuser@physik.uni-muenchen.de`

Abstract. Ganga is a grid job submission and management system widely used in the ATLAS and LHCb experiments and several other communities in the context of the EGEE project. The particle physics communities have entered the LHC operation era which brings new challenges for user data analysis: a strong growth in the number of users and jobs is already noticeable. Current work in the Ganga project is focusing on dealing with these challenges. In recent Ganga releases the support for the pilot job based grid systems Panda and Dirac of the ATLAS and LHCb experiment respectively have been strengthened. A more scalable job repository architecture, which allows efficient storage of many thousands of jobs in XML or several database formats, was recently introduced. A better integration with monitoring systems, including the Dashboard and job execution monitor systems is underway. These will provide comprehensive and easy job monitoring. A simple to use error reporting tool integrated at the Ganga command-line will help to improve user support and debugging user problems. Ganga is a mature, stable and widely-used tool with long-term support from the HEP community. We report on how it is being constantly improved following the user needs for faster and easier distributed data analysis on the grid.

1. Introduction

The Ganga job management system [1, 2] has been developed as a common project between the ATLAS and LHCb experiments. Ganga provides a simple and consistent way of preparing, organizing and executing analysis tasks within the experiment analysis frameworks or general computing tasks implemented through a plug-in system. It allows trivial switching between running test jobs on a local batch system and running large-scale analyses on the Grid, hiding Grid technicalities from the users.

2. LHC analysis

The challenges in a LHC data analysis are high: there are large data volumes from the detectors and simulation which require a large number of CPUs for processing. There is a complex experiment software structure and a high connectivity to the data that needs to be provided. LHC experiments produce and store several PetaBytes/year, the ATLAS experiment e.g. has recorded almost 1 billion events from proton-proton collision at a centre-of-mass of $\sqrt{s}$ =7 TeV in the year 2010. The collision event complexity due to the large number of detector channels and number of analysis users demand at least 100k fast CPUs based on computing model of the experiments. Therefore the distributed analysis tools used to analyze the data should be easy to configure, reliable, fast to work with, and analysis jobs should have a high success rate possibly with at the 1st attempt.

As an example the ATLAS experiment analysis model is explained in more detail in the following. The distributed analysis model in the ATLAS experiment is based on the ATLAS computing model [3]. An ATLAS analysis in a nutshell looks like the following:

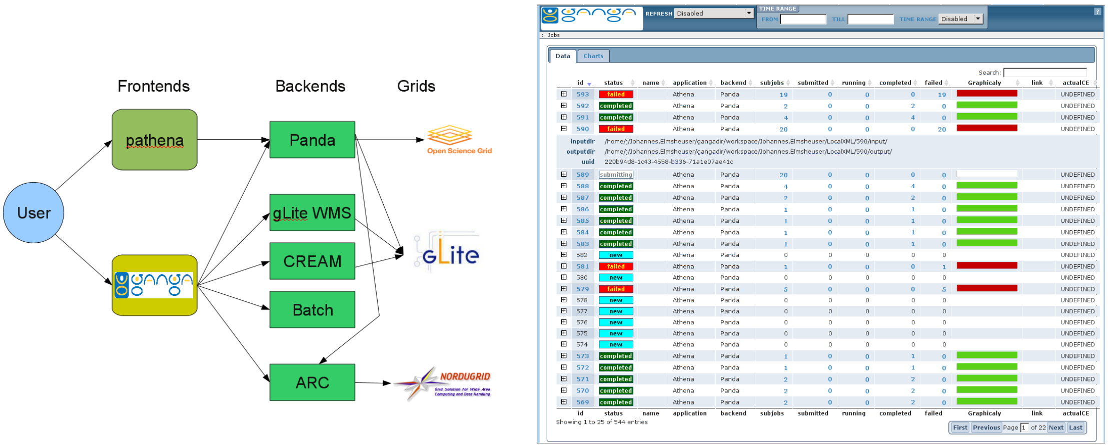
- The detector and simulation data is centrally distributed worldwide by the data management system DQ2 [4] among the different computing centers which are organized in a hierarchical tier structure.
- The different releases of the experiment software Athena [3] are installed at the different sites worldwide by a central installation system.
- The user analysis code and tasks are sent to the sites where the data to be analyzed is located. The model is called: ,,Job to Data".
- The distributed analysis tools for handling all the user job management are Ganga and the Panda clients tools [5].
- The output and results of the user jobs are stored on the scratch disk at the site the job was executed or is transfered on demand to a remote site.
- The results can be retrieved back to the user's local desktop computer using the DQ2 command line tools.

ATLAS foresees different analysis work-flows for the data formats produced by the central production system:

- Athena user code sequentially processes large Monte Carlo (MC) or data stream samples on the Grid in parallelized jobs. These jobs are producing ROOT tuple output which is further processed locally or on the Grid.
- TAG files which contain a condensed event summary can be used to process only events of interest and offer the possibility to seek through large data samples.
- Small MC samples can be produced using the production system transformations for special or official usage.
- Generic ROOT applications eventually with DQ2 access can be executed.

3. Ganga functionality

For user job management Ganga allows users to submit and monitor their jobs and also manipulate previously submitted job objects for repeated analysis. Ganga is one of the main distributed analysis tools in the LHCb and ATLAS collaborations. It offers modules for the experiment specific software and provides access to the resources available to the collaborations, namely the Grids on the LCG/EGEE/EGI [6], NDGF/Nordu-Grid [7] and OSG [5] infrastructure. The access to these resources is mainly managed by the Panda [5] (for ATLAS) and DIRAC [8] (for LHCb) workload management systems. But also gLite WMS [6], CREAM CE [6], and ARC [7] submissions are supported. These different execution plug-ins are called back-ends. The left plot in Figure 1 gives an overview about the different ways for an ATLAS experiment user job into the Grid. In addition local resources can be used through the support of different batch system flavors like PBS, LSF, SGE and Condor. The user can switch between these back-ends by changing a single configuration option and can add special options for every back-end.



	id	status	name	application	backend	subjobs	submitted	running	completed	failed	Graphicaly	link	actualCE
⊞	593	failed		Athena	Panda	19	0	0	0	19	🟥		UNDEFINED
⊞	592	completed		Athena	Panda	2	0	0	2	0	🟩		UNDEFINED
⊞	591	completed		Athena	Panda	4	0	0	4	0	🟩		UNDEFINED
⊟	590	failed		Athena	Panda	20	0	0	0	20	🟥		UNDEFINED
inputdir	/home/j/Johannes.Elmsheuser/gangadir/workspace/Johannes.Elmsheuser/LocalXML/590/input/												
outputdir	/home/j/Johannes.Elmsheuser/gangadir/workspace/Johannes.Elmsheuser/LocalXML/590/output/												
uuid	220b94d8-1c43-4558-b336-71a1e07ae41c												
⊞	589	submitting		Athena	Panda	20	0	0	0	0	□		UNDEFINED
⊞	588	completed		Athena	Panda	4	0	0	4	0	🟩		UNDEFINED
⊞	587	completed		Athena	Panda	2	0	0	2	0	🟩		UNDEFINED
⊞	586	completed		Athena	Panda	1	0	0	1	0	🟩		UNDEFINED
⊞	585	completed		Athena	Panda	1	0	0	1	0	🟩		UNDEFINED
⊞	584	completed		Athena	Panda	1	0	0	1	0	🟩		UNDEFINED
⊞	583	completed		Athena	Panda	1	0	0	1	0	🟩		UNDEFINED
⊞	582	new		Athena	Panda	0	0	0	0	0			UNDEFINED
⊞	581	failed		Athena	Panda	1	0	0	0	1	🟥		UNDEFINED
⊞	580	new		Athena	Panda	0	0	0	0	0			UNDEFINED
⊞	579	failed		Athena	Panda	5	0	0	0	5	🟥		UNDEFINED
⊞	578	new		Athena	Panda	0	0	0	0	0			UNDEFINED
⊞	577	new		Athena	Panda	0	0	0	0	0			UNDEFINED
⊞	576	new		Athena	Panda	0	0	0	0	0			UNDEFINED
⊞	575	new		Athena	Panda	0	0	0	0	0			UNDEFINED
⊞	574	new		Athena	Panda	0	0	0	0	0			UNDEFINED
⊞	573	completed		Athena	Panda	1	0	0	1	0	🟩		UNDEFINED
⊞	572	completed		Athena	Panda	1	0	0	1	0	🟩		UNDEFINED
⊞	571	completed		Athena	Panda	2	0	0	2	0	🟩		UNDEFINED
⊞	570	completed		Athena	Panda	2	0	0	2	0	🟩		UNDEFINED
⊞	569	completed		Athena	Panda	2	0	0	2	0	🟩		UNDEFINED
	id	status	name	application	backend	subjobs	submitted	running	completed	failed	Graphicaly	link	actualCE

Showing 1 to 25 of 544 entries

Figure 1. Left: Overview of the different Grids and Job submission mechanisms offered to the users in the ATLAS experiment. Right: The Ganga WebGUI display with an overview of a user job repository.

An user provides an application and its configuration before job execution. The dataset to be processed is queried for its location and contents in the ATLAS experiment DQ2/DDM data management database. Depending on the size of the dataset the Grid job is divided into several sub-jobs that are executed in parallel and are only processing a subset of the input dataset. During execution jobs are monitored. During job completion the results and output files are stored as an output dataset with a reference in the DQ2/DDM data management system database. Afterwards the user can retrieve the output by simple matters. The work-flow previously described is shown as an example Ganga job configuration in Figure 2. The given example shows a configuration of an Athena job for the command line submission to the Grid. The user specifies the input dataset, the number of sub-jobs for parallel processing, the back-end sent to and the job executable configuration file. The output files and configuration are automatically detected. The job follows the ATLAS computing model and is automatically sent to the location of the data and the shorted estimate job queuing time.

```
ganga athena
  --inDS data10_7TeV.periodF.physics_Muons.PhysCont.AOD.repro05_v02/
  --split 100
  --panda AODToPhysicsD3PD.py
```

Figure 2. Example Athena job configuration of a shell command line job submission with Ganga.

During the last months Ganga has matured its stable environment for distributed analysis. The development model is combining two paradigms: with the growing user community user requests are integrated into the system. At the same time changes of the underlying systems need to be followed and the integration of external components need to be improved. Ganga 5.4.0 has been released on 27 October 2009 followed by 5 minor bugfix and feature releases. Ganga 5.5.0 has been released on 13 February 2010 followed by 17 minor bugfix and feature releases.

Some highlights of new Ganga features to the core part are given as follows:

- A new job repository has been implemented. The job repository is the core Ganga component for job bookkeeping and monitoring. The job repository provides job persistence in a simple database, so that any subsequent Ganga session has access to all previously defined jobs. A new outline has been implemented based on a XML file architecture to store the job meta data. Using the new repository makes Ganga much more reliable with respect to eventual underlying system instabilities. Also the overall application speed and the Ganga start-up time has significantly improved. Using a so called "lazy loading" mechanism which only loads relevant information from disk to memory instead of the full job records increases the performance. LHCb users experience a Ganga start-up time of only 1-2s. The repository easily holds several 10k jobs.
- A new user friendly error reporting tool has been added. This module eases the experiment user support. The user uploads with a single command his/her Ganga environment and job configuration to a central web server. The experiment support teams can then easily inspect the configuration and provide necessary support.
- A new plug-ins for job monitoring has been added to several back-ends. The job wrapper scripts have been instrumented to send out messages about the job configuration and job status to the Task Monitoring Dashboard. The dashboard provides a centralized web page view of all jobs. Job statistics are displayed and error diagnosis can be performed.
- A new plug-in for direct CREAM computing element (CE) submission has been added. The LCG back-end code for gLite WMS submission has been modularized and allows now also CREAM CE submission. The experiment specific application code that is running on the Local, Batch and LCG back-ends, can also be used on the CREAM back-end without any modifications.
- Ganga provides a QT3 based graphical user interface (GUI). A new web browser based WebGUI has been added. Simultaneously to the text based interactive IPython Ganga command shell, a web based GUI can be used to monitor the job status and display different job statistics. The right plot in Figure 1 shows an example display of an user job repository.
- The GangaSAGA plug-in provides an interface to the SAGA-API [11]. The SAGA back-end allows Ganga to exploit Grid infrastructures that are based on middle-ware, currently not supported by Ganga directly, like for example Globus GRAM or OMII GridSAM and to give it potential access to large computing resources, like the US TeraGrid.
- The GangaJEM [12] provides a job level monitoring plug-in. Job performance information and user script monitoring are provided and allow detailed monitoring and debugging of currently running jobs.

Some highlights of new Ganga features specific to the ATLAS experiment part are given as follows:

- The GangaTasks package provides an automatic job configuration, steering, throttling and resubmission system with ATLAS experiment specific applications and data management plug-ins. Large analysis tasks like job chains with interdependent jobs can be configured. Failed jobs will be automatically resubmitted under the user defined conditions which allows an easier job bookkeeping especially when processing large data volumes like a full year of ATLAS data.

- A plug-in for the ATLAS Metadata Interface (AMI) for event based job splitting and luminosity information has been added. AMI provides more detailed information about ATLAS datasets which can be exploited for an user job configuration.
- A plug-in to the Event-Level database (ELSSI) for single event picking has been added. Instead of processing a large data volume at once, ELSSI provides an interface to retrieve physics and DQ2 informations about single detector events. Jobs can be configured using these informations to skim out single events for further processing.
- Many new features have been added to the Panda back-end. Different ROOT based workflows with improved user code handling are now supported. Improved job brokering is utilized to allow cross site job brokering to minimize the job waiting times.
- The job statistics reporting for the Athena application has largely been improved. The number of files, events, processing times and speed are reported for each individual job.

New Ganga features specific to the LHCb plug-ins are described in detail in [10].

4. Ganga usage

In the year 2010 Ganga has been used by over 1300 individual users in total. Currently there are approximately 150 users per week in each of the ATLAS and LHCb experiments (see Figure 3). This is twice as much compared to one year ago. Since the start of the LHC proton-proton collisions at $\sqrt{s}$ =7 TeV in April 2010 there has been a large increase of the distributed analysis activity compared to the previous times. In total there have been more than 370k Ganga sessions and Ganga has been used at more than 130 different sites worldwide.

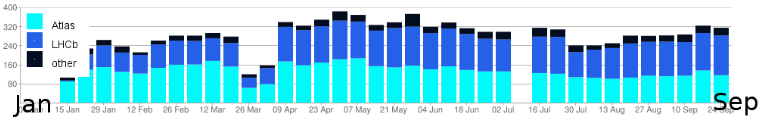


Figure 3. Number of unique Ganga users since January 2010 until October 2010. The Light blue bars show the users of the ATLAS experiment and the dark blue bars show the LHCb experiment users.

Ganga can be used in different user interaction modes: in the IPython, batch, GUI or scripting mode. The left plot in Figure 4 shows that the IPython mode is the most popular followed by the batch interaction mode. The largest user communities of Ganga are from the ATLAS and LHCb experiments as shown in the right plot of Figure 4. About 50% of users are from the ATLAS experiment, followed by about 33% from the LHCb experiment.

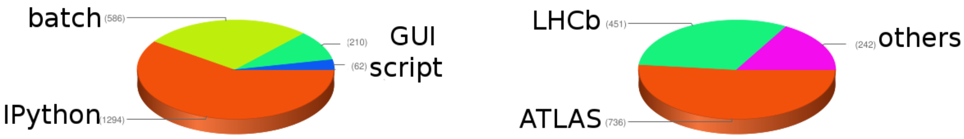


Figure 4. Left: Popularity of the different user interaction modes: IPython, batch, GUI and scripting mode. Right: Distribution of the different user communities. The ATLAS and LHCb experiment users are the largest user communities. Both plots show usage numbers from January 2010 until October 2010.

With the grown user community the ATLAS and LHCb experiments have introduced a formal user support setup. All distributed analysis related questions are sent to an experiment specific mailing list. Shifters in the EU and US time zone are answering and solving user questions and directing users to the documentation of the experiment grid systems. User to user support has increased over time. The most frequently asked questions or problems of users presently are:

- There was a problem with a special experiment software configuration.
- The retrieval of the job output was not fully functional due to a storage element glitch at a remote site.
- A job had problems with accessing the input data files.

The first issue can be addressed either with improved documentation or experience of the users with the tools. The latter two problems are presently addressed by the HammerCloud system [13]: Large storms of analysis jobs or single functional test jobs allow to study the detailed behavior of a test and are used for the site commissioning and site stability improvement. Ganga is core part of this testing system and has helped to improve the overall site reliability [14].

5. Future plans and conclusions

Ganga has entered a consolidation phase after the full start of LHC proton-proton collisions in the year 2010. A large increase in distributed analysis activity can be observed. To further improve the usability, the individual job monitoring will be improved using the task monitoring dashboard plug-in. The WebGUI plug-in will be improved and the ATLAS experiment specific plug-in GangaTasks should be extended for all supported applications in Ganga.

The distributed analysis system Ganga has been described in the context of the ATLAS and LHCb experiments. The functionality offered enables users to fully exploit all Grid functionalities and follow to work-flows foreseen by the experiments' computing model. Many experimenters have successfully used the tools and have successfully analyzed the first LHC data.

References

[1] J.T. Moscicki *et al.*, "Ganga: a tool for computational-task management and easy access to Grid resources", Computer Physics Communications, Volume 180, Issue 11, (2009), doi:10.1016/j.cpc.2009.06.016

[2] Ganga Project Webpage: http://cern.ch/ganga

[3] G. Duckeck *et al.* [ATLAS Collaboration], "ATLAS computing: Technical design report," ATLAS-TDR-017 ; CERN-LHCC-2005-022.

[4] M. Branco *et al.* [ATLAS Collaboration], J. Phys. Conf. Ser. **119** (2008) 062017.

[5] T. Maeno [ATLAS Collaboration], J. Phys. Conf. Ser. **119** (2008) 062036.

[6] P. Andreetto *et al.*, J. Phys. Conf. Ser. **119** (2008) 062007.

[7] M. Ellert *et al.*, Future Generation Computer Systems **23** (2007) 219.

[8] A. Casajus, R. Graciani, S. Paterson and A. Tsaregorodtsev, J. Phys. Conf. Ser. **219** (2010) 062049.

[9] GangaRobot overview page: http://gangarobot.cern.ch

[10] M. Williams *et al.*, "Distributed Analysis in LHCb", these proceedings.

[11] The GangaSAGA webpage: http://faust.cct.lsu.edu/trac/saga/wiki/Applications/GangaSAGA

[12] The JEM webpage: https://svn.grid.uni-wuppertal.de/trac/JEM/

[13] D. Vanderster *et al.*, "HammerCloud: A Stress Testing System for Distributed Analysis", these proceedings.

[14] F. Legger *et al.*, "Distributed analysis functional testing using GangaRobot in the ATLAS experiment", these proceedings.