

Distributed Analysis in the ATLAS Experiment

K. Harrison^a, R.W.L. Jones^b, D. Liko^c, C. L. Tan^d

^aCavendish Laboratory, University of Cambridge, CB3 0HE, UK

^bDepartment of Physics, University of Lancaster, LA1 4YB, UK

^cCERN, CH-1211 Geneva 23, Switzerland

^dSchool of Physics and Astronomy, University of Birmingham, B15 2TT, UK

Abstract

The ATLAS experiment, based at the Large Hadron Collider at CERN, Geneva, is currently developing a grid-based distributed system capable of supporting its data analysis activities which require the processing of data volumes of the order of petabytes per year. The distributed analysis system aims to bring the power of computation on a global scale to thousands of physicists by enabling them to easily tap into the vast computing resource of various grids such as LCG, gLite, OSG and Nordugrid, for their analysis activities whilst shielding them from the complexities of the grid environment. This paper outlines the ATLAS distributed analysis model, the ATLAS data management system, and the multi-pronged ATLAS strategy for distributed analysis in a heterogeneous grid environment. Various frontend clients and backend submission systems will be discussed before concluding with a status update of the system.

1. Introduction

Based in the European Laboratory for Particle Physics (CERN) [1], Geneva, the ATLAS experiment [2] is set to commence its investigations into proton-proton interactions at the most powerful particle accelerator in the world, the Large Hadron Collider (LHC) [3], in the summer of 2007. Each high energy proton-proton collision will produce hundreds of particles in the detector. Highly efficient software and electronics filter and record interactions of potential physics interest for subsequent analysis. The estimated annual yield is 10^9 , corresponding to around 10 petabytes of data.

ATLAS involves a collaboration of more than 2000 physicists and computer scientists from over 150 universities and laboratories in 35 countries across 6 continents. In anticipation of the unprecedented volume of data that will be generated by the ATLAS detector when data taking commences, and large-scale simulation, reconstruction and analysis activities, particle physicists have adopted Grid technology to provide the hardware and software infrastructure required to facilitate the distribution of data and the pooling of computing and storage resources between world-wide collaborating institutions. The ATLAS grid infrastructure currently consists of three grids: LCG (LHC Computing Grid) [4], OSG (US-based Open Science Grid) [5] and Nordugrid (grid project based in the Nordic countries) [6].

The primary aim of the distributed analysis project is to bring the power of computation on a global scale to individual ATLAS physicists by enabling them to easily tap into the vast computing resource provided by the various grids for their analysis activities whilst shielding them from the complexities of the grid environment.

This paper begins by outlining the ATLAS distributed analysis model [7]. It goes on to describe the data management system adopted by ATLAS, then details the ATLAS strategy for distributed analysis [8] in a heterogeneous grid environment. Various frontend clients will be introduced followed by descriptions of different submission systems and their associated grid infrastructure. The paper concludes by providing a status update of the ATLAS distributed analysis system.

2. Distributed Analysis Model

The distributed analysis model is based on the ATLAS computing model [7] which stipulates that data is distributed in various computing facilities and user jobs are in turn routed based on the availability of relevant data.

A typical analysis job consists of a Python [9] script that configures and executes a user-defined algorithm in Athena (the ATLAS software framework) [10] with input data from a file containing a collection of potentially interesting particle interaction information or events and producing one or more files containing plots and histograms of the results.

ID	name	current	priority	supe...	max...	lasta...	exeID	atte...	jobst...	jobn...	trans...	error
159...	santi...	ABO...	1000	sgon...	3	3	71663	3	Failed	COM...	5.12	19
159...	santi...	DONE	1000	sgon...	3	1	71754	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71755	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71756	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71757	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71758	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71759	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71760	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71761	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71762	1	FINIS...	COM...	0	0
159...	santi...	DONE	1000	sgon...	3	1	71763	1	FINIS...	COM...	0	0
159...	santi...	PEND...	1000	sgon...	3	1	71766	1	PEND...	IDLE		
159...	santi...	PEND...	1000	sgon...	3	1	71767	1	PEND...	IDLE		
159...	santi...	PEND...	1000	sgon...	3	1	71768	1	PEND...	IDLE		

Monitor

Create Jobs Reload Hidden Jobs Only last Attempt Close Task

shown:59 hidden:19 defined:59 done:41 tobedone:0 aborted:8 waitinginput:0 blocked:0

Figure 1: ATCOM – ATLAS Production System GUI

As with many large collaborations, different ATLAS physics groups have different work models and the distributed analysis system needs to be flexible enough to support all current and newly emerging work models whilst remaining robust.

3. Distributed Data Management

With data distributed at computing facilities around the world, an efficient system to manage access to this data is crucially important for effective distributed analysis.

Users performing data analysis need a *random access* mechanism to allow rapid pre-filtering of data based on certain selection criteria so as to identify data of specific interest. This data then needs to be readily accessible by the processing system.

Analysis jobs produce large amounts of data. Users need to be able to store and gain access to their data in the grid environment. In the grid environment where data is not centrally known, an automated management system that has the concept of file ownership and user quota management is essential.

To meet these requirements, the Distributed Data Management (DDM) system [11] has been developed. It provides a set of services to move data between grid-enabled computing facilities whilst maintaining a series of databases to track these data movements. The vast amount of data is also grouped into *datasets* based on various criteria (e.g. physics characteristics, production batch run, etc.) for more efficient query and retrieval. DDM consists of three components: a *central dataset catalogue*, a *subscription service* and a set of *client tools* for dataset lookup and replication.

The central dataset catalogue is in effect a collection of independent internal services and catalogues that collectively function as a single *dataset bookkeeping system*.

Subscription services enable data to be automatically *pulled* to a site. A user can ensure

that he/she is working on the latest version of a dataset by simply subscribing to it. Any subsequent changes to this dataset (i.e. additional files, version changes, etc.) will trigger a fresh download of the updated version automatically.

Client tools provide users with the means to interact with the central dataset catalogue. Typical actions include listing, retrieving and inserting of datasets.

4. Distributed Analysis Strategy

ATLAS takes a multi-pronged approach to distributed analysis by exploiting its existing grid infrastructure directly via the various supported grid flavours and indirectly via the ATLAS Production System [12].

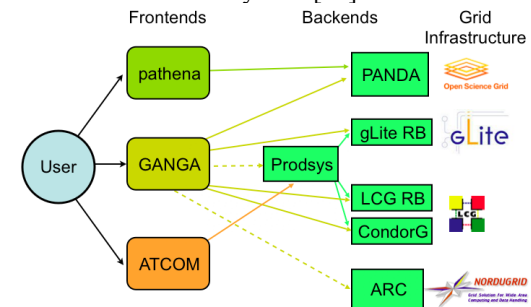


Figure 2: Distributed analysis strategy

4.1 Frontend Clients

Figure 2 shows various frontend clients enabling distributed analysis on existing grid infrastructure.

Pathena [13] is a Python script designed to enable access to OSG resources via the Panda job management system [14]. It is just short of becoming a drop-in replacement for the executable used in the ATLAS software framework. Users are able to exploit distributed resources for their analysis activities with the very minimal inconvenience.

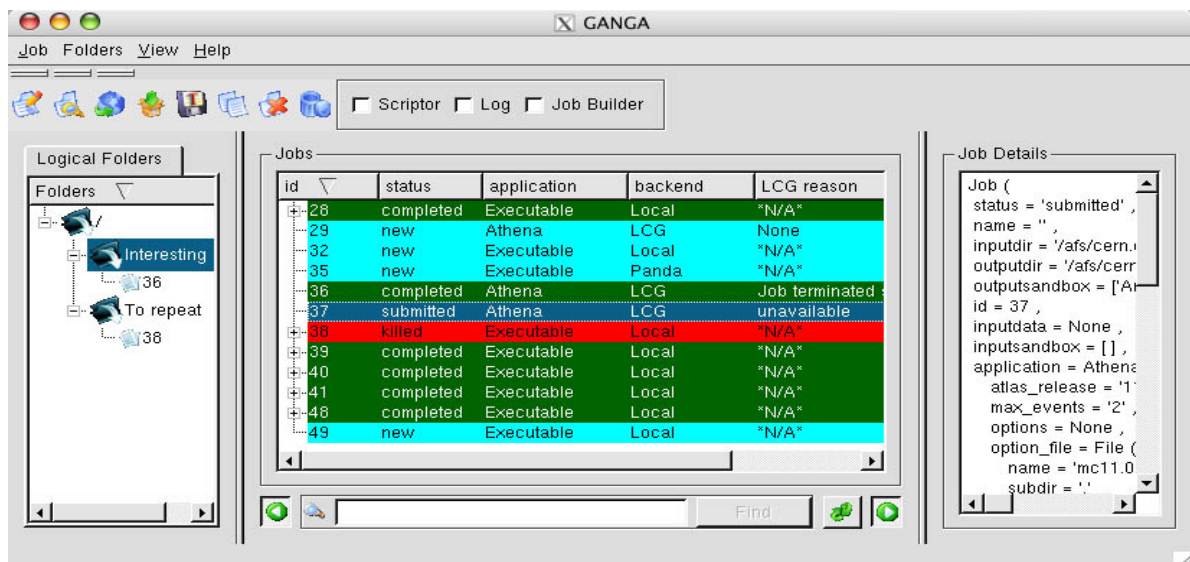


Figure 3: GANGA - Job definition and management tool

Pathena makes the submission of analysis jobs to the Panda system a painless two stage process involving an optional *build* step (where user code can be compiled) followed by an *execution* step (with built-in job splitting capabilities). A further *merge* step is in development which will allow the resulting output datasets from split jobs to be consolidated.

ATCOM [15] is the dedicated graphical user interface frontend (See Figure 1) to the ATLAS production system designed to be used by a handful of expert users involved in large-scale organised production of ATLAS data. It has potential to be used for end-user distributed analysis purposes.

GANGA [16] is a powerful yet user friendly frontend tool for job definition and management, jointly developed by the ATLAS and LHCb [17] experiments. GANGA provides distributed analysis users with access to all grid infrastructure supported by ATLAS. It does so by interfacing to an increasing array of submission backend mechanisms. Submission to the production system and ARC are planned for the not so distant future.

GANGA currently provides two user interface clients: a Command Line Interface (CLI) and a Graphical User Interface (GUI) (See Figure 3). In addition, it can also be embedded in scripts for non-interactive/repetitive use. GANGA, due to its need to satisfy ATLAS and LHCb experiment requirements (unlike Pathena and ATCOM which are specialised tools designed for specific ATLAS tasks), has been designed from the onset to be a highly extensible generic tool with a component plug-in architecture. This pluggable framework makes the addition of new applications and backends an easy task.

A synergy of GANGA and DIANE [18] (a job-distribution framework) has been adopted in several instances. In each instance, GANGA

was used to submit various types of jobs to the Grid including the search for drugs to combat Avian flu, regression testing of Geant 4 [19] to detect simulation result deviations and the optimisation of the evolving plan for radio frequency sharing between 120 countries.

A few physics experiments (e.g. BaBar [20], NA48 [21]) have also used GANGA in varying degrees while there are others (e.g. PhenoGrid [22], Compass [23]) in the preliminary stages of looking to exploit GANGA for their applications.

GANGA is currently in active development with frequent software releases and it has an increasing pool of active developers.

4.2 Production System

The ATLAS production system provides an interface layer on top of the various grid middleware used in ATLAS. There is increased robustness as the distributed analysis system benefits from the production system's experience with the grid and its *retry and fallback* mechanism for both data and workload management.

A rapidly maturing product, the production system provides various facilities that are useful for distributed analysis e.g. user configurable jobs, X509 certificate-based access control and a native graphical user interface, ATCOM.

4.3 LCG and gLite

LCG is the computing grid designed to cater to the needs of all the LHC experiments and is by default the main ATLAS distributed analysis target system. Access to the LHC grid resources is via the LCG Resource Broker (RB) or CondorG [24]. The LCG RB is a robust submission mechanism that is scalable, reliable and has a high job throughput. CondorG, although conceptually similar to the LCG RB,

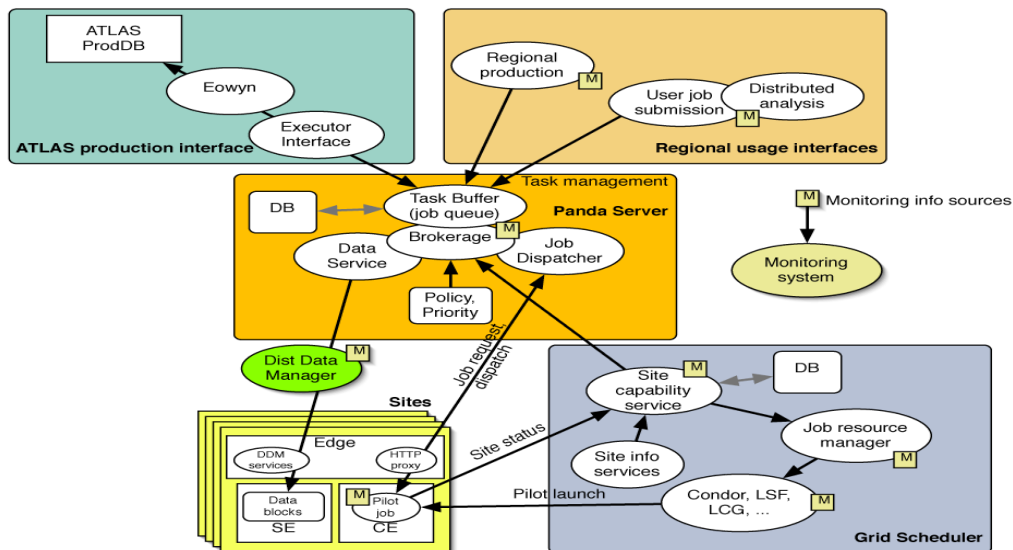


Figure 4: Panda architecture

has a different architecture. Nevertheless, both submission mechanisms have been successfully exploited in recent large-scale ATLAS production exercises.

gLite [25] is the next generation grid middleware infrastructure project by EGEE (Enabling Grids for E-Science) [26]. Recent code base convergence between gLite and LCG has resulted in gLite 3.0. The gLite RB has performance enhancements that are of particular benefit to distributed analysis: efficient bulk job submissions with improved support for output result retrieval.

GANGA supports direct analysis job submission to the CondorG, LCG RB and the gLite RB.

4.4 Panda

Panda is a job management system associated with OSG designed specifically for both distributed production and distributed analysis. See Figure 4.

Panda has native support for the ATLAS DDM system allowing it to accept DDM datasets as input (pre-staging it where required) and producing DDM datasets as output (retrievable using DDM tools).

Panda offers users a comprehensive system view which presents heterogeneous distributed resources as a single uniform resource accessible through a standard interface. It also has extensive web-based job monitoring and browsing capabilities.

Panda does not have a GUI but it looks to GANGA to provide it with a graphical job definition and submission interface. Panda has made this possible by exposing a useful set of client API.

4.5 ARC

ARC (Advanced Resource Connector) [27] is developed by the Nordugrid collaboration and is based on the Globus Toolkit [28].

Processes: - Grid - Local

Country	Site	CPUs	Load (processes: Grid+local)	Queueing
Denmark	Benedict - Aalborg pr>	48	0+0	17+0
	Louis XIV (DCGC/AAU)	52	0+1	0+0
	LSCF (NBI)	32	0+0	0+0
Estonia	Tartu Observatory	5	0+0	0+0
	UT CS Antarctica Clus>	17	0+0	0+0
	UT IMCB Anakonda clus>	13	0+0	0+0
	UT Physics Cluster	3	0+0	0+0
Finland	Akaatti (M-grid)	30	4+2	6+0
	Ametisti (M-grid)	132	0+112	0+41
	Hirmu Cluster (HIP)	4	0+0	0+0
	Jaspis (M-Grid, HIP)	8	0+0	0+0
	Kivi (M-grid)	10	0+0	0+0
	Kvartsi (M-grid)	96	0+93	0+37
	OpaaII (M-grid)	24	0+23	0+0
	SepaII (M-grid)	768	0+46	0+212
Lithuania	Spektrolitii (M-grid)	26	0+23	0+0
	Topaasi (M-grid)	24	0+22	0+0
	grid.ktu.lt	4	0+0	0+0

Figure 5: Nordugrid Grid Monitor

The ARC client is a light-weight, self-contained job submission tool with built-in highly customisable resource brokering functionality and input / output file staging facilities. It has an impressive job delivery rate of approximately 30-50 job deliveries/min making it potentially useful for interactive (i.e. responsive) distributed analysis.

As with all other grid flavours, the ARC client has a comprehensive set of command line tools for job submission and management. The web-based Nordugrid Grid Monitor [29] complements the ARC client by providing detailed system-wide job monitoring information for *all* jobs running on Nordugrid resources.

Although not specifically designed for distributed analysis, ARC has immense potential due to its stability and performance. Certain issues with data management still need to be finalised. GANGA is looking to interface with Nordugrid in the not too distant future.

5. Conclusion

Distributed analysis in ATLAS is still in its infancy but is evolving rapidly. Many key components like the DDM system have only just come online. The multi-pronged approach to distributed analysis will encourage one submission system to learn from another and ultimately produce a more robust and feature-rich distributed analysis system. The distributed analysis system will be comprehensively tested and benchmarked as part of Service Challenge 4 [30] in the summer of 2006.

Acknowledgements

We are pleased to acknowledge support for the work on the ATLAS distributed analysis system from GridPP in the UK and from the ARDA group at CERN. GridPP is funded by the UK Particle Physics and Astronomy Research Council (PPARC). ARDA is part of the EGEE project, funded by the European Union under contract number INFISO-RI-508833.

References

- [1] <http://cern.ch>
- [2] ATLAS Collaboration, Atlas - Technical Proposal, CERN/LHCC94-43 (1994); <http://atlas.web.cern.ch/Atlas/>
- [3] LHC Study Group, The LHC conceptual design report, CERN/AC/95-05 (1995); <http://lhc.web.cern.ch/lhc/>
- [4] <http://lcg.web.cern.ch/LCG/>
- [5] <http://www.opensciencegrid.org/>
- [6] <http://www.nordugrid.org/>
- [7] ATLAS Computing Group, ATLAS Computing Technical Design Report, CERN-LHCC-2005-022; <http://atlas-proj-computing-tdr.web.cern.ch/atlas-proj-computing-tdr/PDF/Computing-TDR-final-July04.pdf>
- [8] D. Liko et al., The ATLAS strategy for Distributed Analysis in several Grid infrastructures, in: Proc. 2006 Conference for Computing in High Energy and Nuclear Physics, (Mumbai, India, 2006); <http://indico.cern.ch/contributionDisplay.py?contribId=263&sessionId=9&confId=048>
- [9] G.van Rossum and F.L. Drake, Jr. (eds.), Python Reference Manual, Release~2.4.3 (Python Software Foundation, 2006); <http://www.python.org/>
- [10] <http://cern.ch/atlas-proj-computing-tdr/Html/Computing-TDR-21.htm#pgfId-1019542>
- [11] ATLAS Database and Data Management Project; <http://atlas.web.cern.ch/Atlas/GROUPS/DATABASE/project/ddm/>
- [12] ATLAS Production System; <http://uimon.cern.ch/twiki/bin/view/Atlas/ProdSys>
- [13] T. Maeno, Distributed Analysis on Panda; <http://uimon.cern.ch/twiki/bin/view/Atlas/DAonPanda>
- [14] T. Wenaus, Kaushik De et al, Panda - Production and Distributed Analysis; <http://twiki.cern.ch/twiki//bin/view/Atlas/Panda>
- [15] <http://uimon.cern.ch/twiki/bin/view/Atlas/AtCom>
- [16] <http://ganga.web.cern.ch/ganga/>
- [17] LHCb Collaboration, LHCb - Technical Proposal, CERN/LHCC98-4 (1998); <http://lhcb.web.cern.ch/lhcb/>
- [18] <http://it-proj-diane.web.cern.ch/it-proj-diane/>
- [19] <http://geant4.web.cern.ch/geant4/>
- [20] <http://www-public.slac.stanford.edu/babar/>
- [21] <http://na48.web.cern.ch/NA48/>
- [22] <http://www.phenogrid.dur.ac.uk/>
- [23] <http://www.compass.cern.ch/>
- [24] <http://www.cs.wisc.edu/condor/condorg/>
- [25] <http://glite.web.cern.ch/glite/>
- [26] <http://egee-intranet.web.cern.ch/egee-intranet/gateway.html>
- [27] <http://www.nordugrid.org/middleware/>
- [28] <http://www.globus.org/>
- [29] <http://www.nordugrid.org/monitor/>
- [30] LHC Computing Grid Deployment Schedule 2006-08, CERN-LCG-PEB-2005-05; <http://lcg.web.cern.ch/LCG/PEB/Planning/deployment/Grid%20Deployment%20Schedule.htm>