Experience with the gLite Workload Management System in ATLAS Monte Carlo production on LCG

Simone Campana (CERN), David Rebatto (INFN Milano), Andrea Sciaba' (CERN)

Simone.Campana@cern.ch

Abstract. The ATLAS experiment has been running continuous simulated events production since more than two years. A considerable fraction of the jobs is daily submitted and handled via the gLite Workload Management System, which overcomes several limitationsof the previous LCG Resource Broker. The gLite WMS has been tested very intensively for the LHC experiments use cases for more than six months, both in terms of performance and reliability. The tests were carried out by the LCG Experiment Integration Support team (in close contact with the experiments) together with the EGEE integration and certification team and the gLite middleware developers. A pragmatic iterative and interactive approach allowed a very quick rollout of fixes and their rapid deployment, together with new functionalities, for the ATLAS production activities. The same approach is being adopted for other middleware components like the gLite and CREAM Computing Elements. In this contribution we will summarize the learning from the gLite WMS testing activity, pointing out the most important achievements and the open issues. In addition, we will present the current situation of the ATLAS simulated event production activity on the EGEE infrastructure based on the gLite WMS, showing the main improvements and benefits from the new middleware. Finally, some preliminary results on the new flavors of Computing Elements usage will be shown, trying to identify possible advantages not only in terms of robustness and performance, but also functionality for the experiment activities.

1. Introduction.

The ATLAS collaboration [1] is preparing for LHC [2] data acquisition in 2007 and is therefore validating its computing model. Because of the required amount of computing resources (more than 9 MSI2000[1] of CPU capacity and more that 7 PB of storage space for the first year of data taking), ATLAS embraces the Grid paradigm i.e. a high level of decentralization and sharing of computing resources. More in particular, the ATLAS computing model [3] organizes the different computing facilities in a hierarchical structure, with distinct roles at different levels. The Tier0, located at CERN, will hold the master copy of the raw data (coming directly from the ATLAS pit) and will be

[1] A Intel Pentium IV processor with a 2.8 GHz CPU corresponds to about one kSI2000; the current ATLAS share of CPU resources in the CERN batch facility correponds to about 600 kSI2000.

responsible for the first pass reconstruction of data to produce a data format suitable for analysis. The ATLAS Tier1s (10 sites in total) will be custodial of a second replica of the raw data and take care of data reprocessing. The Tier2 sites will be responsible to run Monte Carlo production and accommodate user analysis.

ATLAS is running continuous distributed Monte Carlo production since at least two years, not only to provide simulated data for physics studies, but also to contribute to the commissioning of the ATLAS computing system as well as the complete software suite. ATLAS benefits from resources in three different Grid infrastructures: EGEE [4], OSG [5] and NorduGrid [6]. Moreover, despite the description of tier responsibilities in the computing model, at the moment distributed production is run at every ATLAS site and even some opportunistic resources.

In this paper we will describe recent results of ATLAS production on the EGEE infrastructure. More in particular we will focus on the job submission and handling via the gLite [7] Workload Management System (WMS), describing the limitations which have been faced, the effort spent to overcome them and the latest results.

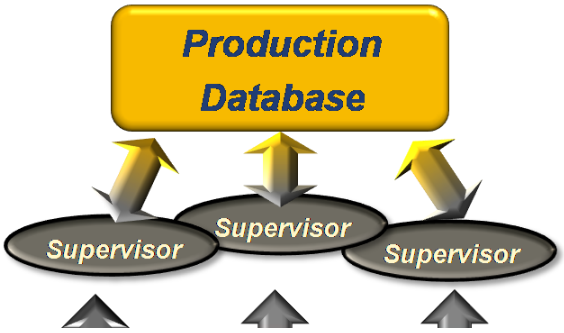

Figure 1: the ATLAS Production System

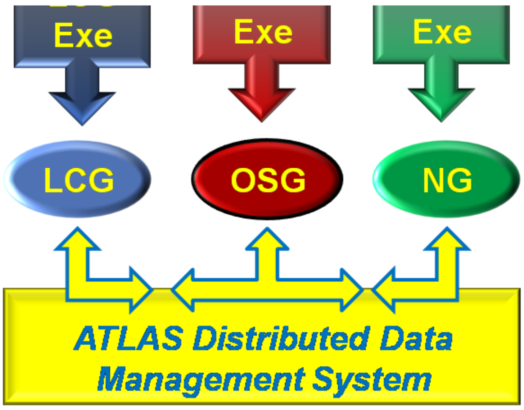

Figure 1: the ATLAS Production System

2. The ATLAS production system

As already mentioned, ATLAS resources are spread over different Grid infrastructures, deploying different middleware services and clients and therefore presenting a very heterogeneous environment.

The ATLAS Production System [8], must be able therefore to implement jobs submission and handling over the three Grid. A description of the ATLAS Production System is given in Figure 1. The job definitions, organized in production tasks are defined in a central database in a grid-neutral format. Job definitions are injected into the database via a web-based task creation tool and handled by a supervisor agent, also Grid-unaware. The Supervisor fetches job definition from the production database, submits jobs to the computer centers, polls from time to time the job status and validates the job in case of successful completion or triggers resubmission in case of failure. Being the Supervisor a grid-neutral object, a series of plug-ins (Executors) handles the interaction with the underlying grid middleware. For the EGEE grid, there is a variety of Executors, differing for the job submission and handling method; in this contribution we will focus on Lexor, which relies on the gLite WMS service. To conclude, data movement across Grids is guarantee via the ATLAS Distributed Data Management system [3].

2.1. *Lexor* is a translator of Production System -to- WMS requests. It converts the python objects passed by the Supervisor into the User Interface API specific python objects, and vice versa. The main ideas leading Lexor implementation were not to duplicate existing middleware functionalities, and to have a thin, stateless layer (states are already stored in the production database and in the grid middleware itself). Some manipulation is anyway required, as the mapping between middleware and Production System objects is not always that trivial. For example, Lexor needs to aggregate jobs in order to take profit of the ”bulk submission” feature of the WMS, thus introducing a jobs’ collection concept which is extraneous to the production system. A similar bulk operation for retrieving the status of the jobs is available in the middleware, and will soon integrated in Lexor. In its original implementation, Lexor also included the runtime wrapper (i.e. the script around the actual transformation, responsible in particular of the whole data transfer from and to the grid). This is now part of the Common Executor - the code shared among the three LCG executors - and evolved a lot since its first implementation. It was rewritten in Python and better integrated with both the transformation itself (which is now in Python too) and the Data Management layer.

2.2. *The gLite WMS* is a set of services and clients which allow job submission and handling to the EGEE Computing Elements. Lexor interacts with the WMS via the WMProxy web service interface, which translates the job definition from the client (Lexor) into internal format for the various WMS internal services. Moreover, the WMProxy records the job entry into the Logging&Bookkeeping service, which provides the state storage of active jobs. Jobs inside the WMS are processed in a pipeline. The requirements specified in the job definition are considered and various matching resources are ordered based on a user-defined ranking expression. Jobs are then delivered to the best matching resource at the moment via a Condor-G client. The gLite WMS offers many improvements in respect of its ancestor, the LCG Resource Broker: bulk submission allows to submit sets of independent jobs in a much more reliable and compound operation, up to a rate of 200MHz for job submission and 0.5Hz for job dispatching to the Computing Elements; also the matchmaking of jobs to most suitable resources is done in bulk, where job classes, with common requirements and ranking expressions, are matched in a single operation. The sandboxing of the gLite WMS (for job inputs and outputs), does not need to transit from the WMS anymore, but can use external services via well established protocols like gridFTP and http. The reliability of the service is also improved via a “limiter” mechanism which prevents job submission in case the WMS presents already too heavy load. Last, but not least, the gLite WMS is actively developed, deployed and maintained, while the LCG Resource Broker is “frozen”.

3. WMS Issues And Experimental Services

The gLite WMS has been one of the sources of inefficiencies in the past for the ATLAS production. Lexor is using the gLite WMS since summer 2006 and the feeling was that the WMS was not really ready for production at that point. The problems were identified more in the reliability of the service rather than in its performance. The bulk submission mechanism, at the time based on Condor DagMan presented several limitations: jobs were remaining in stale non-final states forever (“zombies”), requiring frequent human intervention and therefore a lot of attention from ATLAS production managers. In addition, the memory consumption of the server was not under control, presenting a linear growth under continuous job submission, and requesting many service restarts over limited periods of time. The period from September to November 2006 has been particularly problematic.

Starting from January 2007, an intense effort of testing and debugging of the gLite WMS started. Many parties have been involved: the ATLAS and CMS experiments, the EGEE SA1, SA3 and JRA1 teams and the WLCG Experiment Integration Support Team. The approach has been extremely pragmatic: starting from experiment requirements about performance and reliability of the system, a test plan and toolkit has been setup. Results of the various tests have been reported promptly to software developers. Patches were produced starting from the outcome of the tests and rolled out in a very restricted environment bypassing the normal EGEE certification activity. This Experimental Services approach, which resulted in a very fast turnaround between testing and development, gave the possibility to fulfil the experiments baseline requirements.

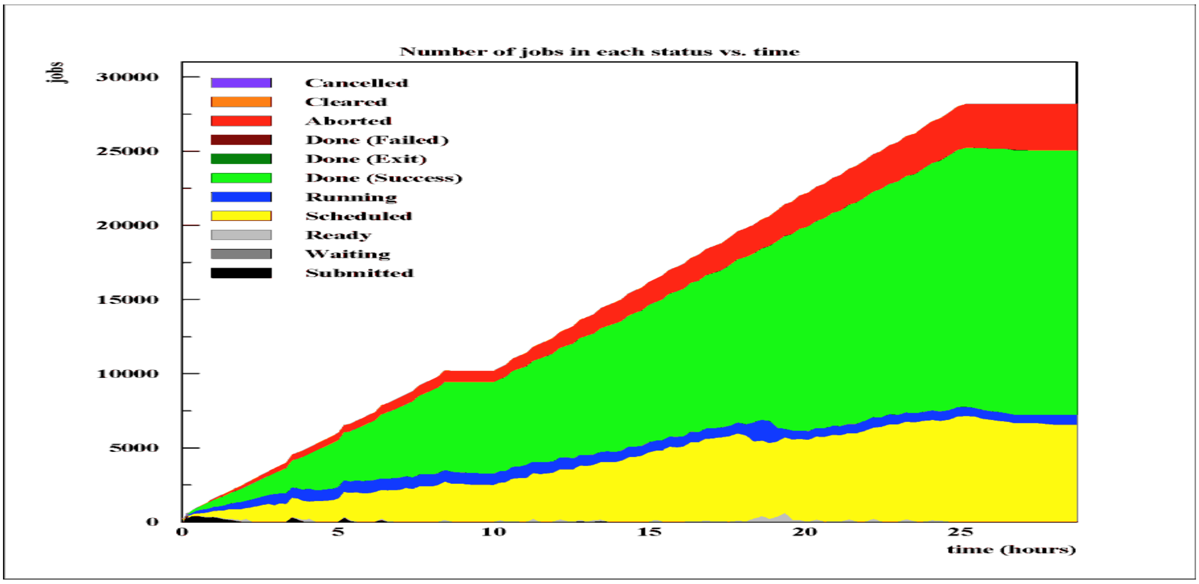

Figure 2: number of jobs in any given state during the acceptance test

3.1. The Experiment Requirements for a reliable WMS service (acceptance criteria) can be summarized as follows: the WMS must be capable to handle the submission and dispatching of 10,000 jobs per day to EGEE Computing Elements for at least 5 consecutive days.

No service should be restarted during this time and no intervention on the WMS should be carried on. The performance of the WMS at the end of this period of time should be the same as the one observed at the beginning of the test. The number of stale jobs at the end of the test should be less that 1% of the total number of submitted jobs. This baseline should accommodate largely the requirements from the ATLAS and CMS experiments for years 2007 and 2008 which imply a submission rate between 50,000 and 100,000 jobs per day with less than 10 WMS nodes.

3.2. The test setup. The tests have been carried on submitting jobs to a single gLite WMS installed on a dedicate hardware at CNAF (Dual Opteron, 2.4 GHz per core, 4 GB of memory). The WMS has been stressed via continuous jobs submission over a period of 1 week. Jobs consisted of negligible payload and negligible input and output sandbox and were submitted in bulks of 100 jobs each with ATLAS credentials to all available ATLAS sites using the default ranking expression of the WMS. The behaviour of the WMS has been monitored under many aspects: a job monitoring agent was inspecting the status of each job during its lifetime to address eventual problems of stale jobs; several agents running on the WMS export information about the internal status of the service (length of the various internal queues in particular) on a web page; other relevant quantities like load of the WMS machine, memory consumption and I/O have been gathered from the fabric monitoring.

3.3. The test results. The problem with the memory consumption growing linearly under continuous job submission has been identified in a very short time and corrected. No problem of this kind ever appeared afterward. The issue of stale jobs has been identified in how the collection handling was performed by the Condor DagMan component. In this case, a radical change has been made and a native collection handling component has been re-written by the WMS developers. The final results of the tests for the acceptance criteria are summarized in Figures 2, showing the evolution with time of the number of jobs in any given state. In summary, after 5 days of continuous job submission, with no

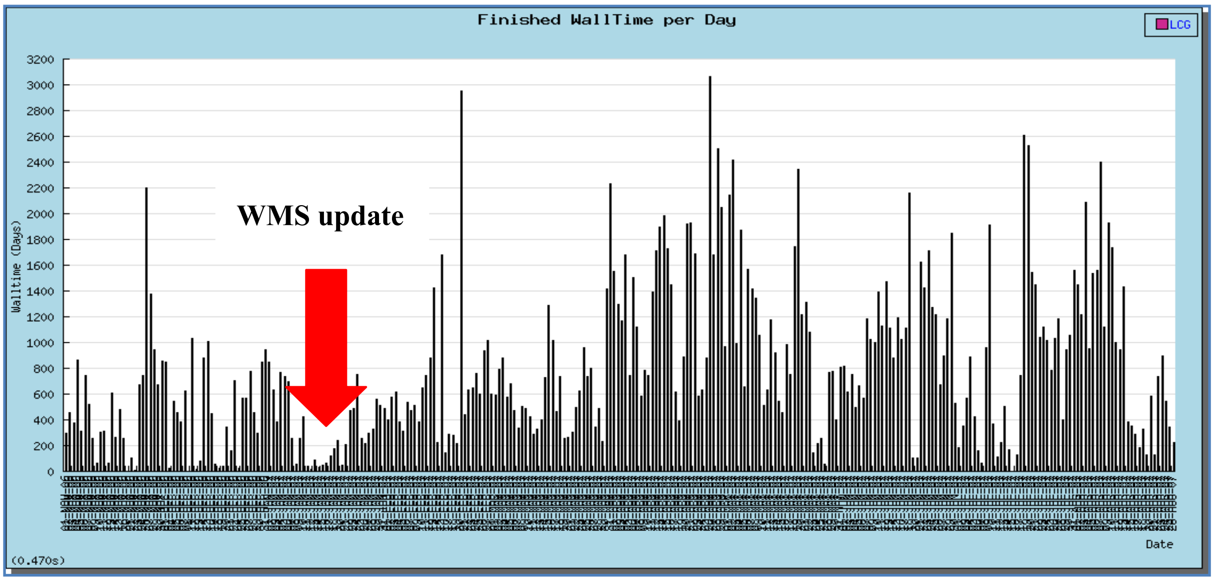

Figure 3: amount of WallClockTime spent successful by ATLAS jobs in the last 9 months

interruptions and no interventions of the service, 115,000 jobs have been successfully dispatched to EGEE Computing Elements (16,000 jobs per day, which is well above the acceptance criteria). In two occasion job submission was prevented by the WMS “limiter” mechanism since the load was considered to high (roughly 12, while the threshold was set to 10). All jobs were processed normally except for 320 jobs (roughly 0.5% of the total, below the maximum limit of 1% of the acceptance criteria) which remained in submitted status. Those jobs could be restarted by the user at the end of the test using a standard User Interface command line tool without any administrator privilege. In addition, the WMS dispatched the jobs to the Computing Elements without noticeable delay. Given the results of the test, it has been declared that the gLite WMS had met the acceptance criteria of the ATLAS and CMS experiments.

4. Experiment activities

Experiments production activities received a considerable benefit from the WMS testing activity with experimental services. In Figure 3 we show the amount of wall time spent by ATLAS jobs in the last 9 months. The arrow indicates the upgrade to the latest version of the gLite WMS fixing the problems spotted during the acceptance test. Considering that the average length of ATLAS jobs does not significantly change over time, it can be seen that the improvement is considerable. The average number of successful ATLAS jobs run per day increased by at least a factor of four, reaching a peak of 3,000 KSI2000 in a single day, which is still lower that the total ATLAS capacity on the EGEE infrastructure (about 5,000 KSI2000) but it should not be forgotten that other submission system compete with Lexor for EGEE resources in the official production and that several resources are taken for user data analysis. In terms of job failures, in the first semester of year 2007, the ATLAS production presented a 55% job efficiency (to be compared with 65% efficiency for the overall ATLAS production activity) and about 25% of the errors could be associated to WMS inefficiencies. In the last three months the efficiency raised to 62% and with WMS related problems counting for 13% of the total number of failures. In addition, Lexor is rather efficient in terms of WallClocTime spent for successful versus failed jobs. This efficiency for Lexor in the last three months sits around

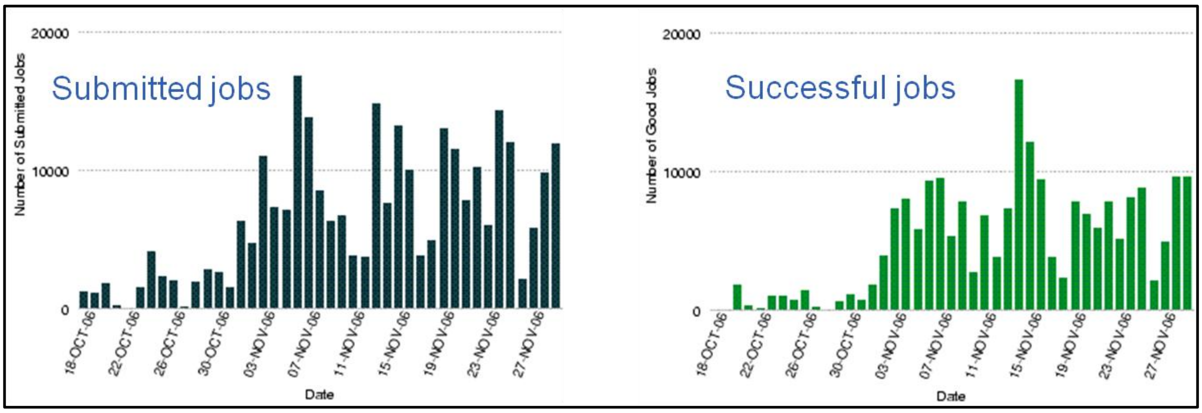

Figure 4: submitted and successful CMS “fake” jobs during the Computing Challenge in 2006

87%, in line with the overall WallClockTime Efficiency of the ATLAS production system (base also on different submission mechanism considered generally very reliable).

Considerable improvements can also be shown for the CMS experiment. CMS supports submission of job collections via the WMS for the analysis activity. The gLite WMS has been tested by CMS during the Computing/Software/Analysis challenge in October 2006 for a period of about one month. The submission rate of “fake” analysis jobs reached about 16,000 jobs per day using two WMS instances. In Figure 4 we show the number of submitted and successful jobs via WMS over a period of 10 days during the challenge mentioned above.

5. Conclusions

The critical problems which affected the gLite WMS in the past months have been understood. There are obviously still some issues to be addressed, in particular related to the algorithm matching job definition to available resources and a possible evolution toward a more stochastic algorithm. There are still several features offered by the gLite WMS which have not been explored and could be useful in the future (inspection of job standard error and standard output streams at runtime, dynamic exclusion of resources based on failure rate of previous job and many other). In general, the improvements in respect of the old LCG Resource Broker are significant and those improvements had large impact over several experiment activities. In this paper we presented only the ATLAS and CMS cases as main users of the gLite WMS at the present moment, however all LHC Virtual Organizations are ready to use it and its usage extends beyond High Energy Physics applications. The “Experimental Services” approach has been proven to be very efficient, therefore it has been adopted also for other kind of testing activities like the Condor Computing Element and the CREAM Computing Element

6. AKNOWLEDGEMENTS

EGEE is a project funded by the European Union under contract INFSO-RI-508833

References

[1] ATLAS Collaboration, "ATLAS Technical Proposal", CERN/LHCC/94-43, 1994

[2] T.S. Virdee, “Detectors at LHC”, Phys. Rep. 403-404:401-434, 2004 and F. Gianotti, “Physics

at the LHC", Phys. Rep. 403:379-399, 2004.

[3] The ATLAS Computing Group, "ATLAS Computing Technical Design Report", ATLAS-TDR-017, CERN-LHCC-2005-022, June 2005

[4] B. Jones, "An overview of the EGEE project", Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures, Lecture Notes in Computer Science, Volume 3664/2005, Springer, 2005

[5] L. Field, "Grid Deployment Experiences: The interoperations activity between OSG and LCG", Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India).

[6] M. Gronager, "LCG and ARC middleware interoperability", Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India).

[7] E. Laure et al., "Programming the Grid with gLite", Computational Methods in Science and Technology, 12(1), 2006

[8] L. Goossens, "Production System in ATLAS DC2", Conference on Computing in High Energy and Nuclear Physics (CHEP04), September 2004, Interlaken (Switzerland).

[9] M. Branco, D. Cameron, T. Wenaus, "A Scalable Distributed Data Management System for ATLAS", Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)