Large-scale ATLAS production on EGEE

Xavier Espinal¹, Simone Campana² and Rod Walker³
on behalf of the ATLAS EGEE production team

¹PIC( Port d’Informació científica), Universitat Autonoma de Barcelona, Edifici D 08193 Bellaterra, Barcelona (SPAIN)
²CERN (European Laboratory for Particle Physics), Rue de Geneve 23 CH 1211 Geneva, Switzerland
³TRIUMF (Tri - University Meson Facility), 4004 Wesbrook Mall Vancouver (BC), Canada
E-mail: espinal@ifae.es

Abstract. In preparation for first data at the LHC, a series of Data Challenges, of increasing scale and complexity, have been performed. Large quantities of simulated data have been produced on three different Grids, integrated into the ATLAS production system. During 2006, the emphasis moved towards providing stable continuous production, as is required in the immediate run-up to first data, and thereafter. Here, we discuss the experience of the production done on EGEE resources, using submission based on the gLite WMS, CondorG and a system using Condor Glide-ins. The overall wall time efficiency of around 90% is largely independent of the submission method, and the dominant source of wasted cpu comes from data handling issues. The efficiency of grid job submission is significantly worse than this, and the glide-in method benefits greatly from factorising this out.

1. Introduction

The Large Hadron Collider (LHC) will begin to take data in Early 2008 at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland.

ATLAS (A Toroidal LHC ApparatuS) is one of the four big experiments being prepared for the Large Hadron Collider (LHC), a particle accelerator ring installed in a 50-150 meter underground tunnel 27 kilometers in circumference astride the border between Switzerland and France, ATLAS is designed to explore the fundamental nature of matter and the basic forces that shape our universe.

Protons accelerated by the LHC in two counter-rotating beams will be kept circulating for hours, guided by thousands of powerful super-conducting magnets operating at 300 degrees below room temperature, before colliding in the heart of the ATLAS detector at almost the speed of light (pp @ $\sqrt{s}=14$ Tev). The resulting energy of colliding protons will transform fleetingly into particle debris to be examined for signs of extremely rare events, such as the creation of the much-sought Higgs boson.

With a rate of 800 million collisions per second, the LHC will be worlds largest and most powerful particle accelerator when it commences operations in 2008, and ATLAS will be the
largest collaborative effort ever attempted in the physical sciences with 1,800 physicists (including 400 students) participating from more than 150 universities and laboratories in 34 countries, all of whom eager to see what new discoveries will be revealed.

With such a rate of collisions, even including the reduced rate due to the online trigger processing farms, the expected volume of data recorded for offline reconstruction and analysis will be few of the order of several ($10^{15}$ bytes) per year. This will be analyzed by physicists all over the world [1].

MC production is performed all over the world both at large computer centers, called Tier-1s, and at smaller sites, called Tier-2s, as well as in institute or university sites, called Tier-3s, which in the ATLAS computing model are associated to the Tier-1s. Currently on EGEE grid infrastructure ATLAS uses 8 Tier-1s (plus 2 additional Tier-1s: one in OSG and one in NDGF), more than 40 Tier-2s plus few tens of Tier-3s. Its resources are used to generate huge amount of simulated data, around 120 million events were produced within the period between October 2006 and June 2007.

2. The ATLAS production system

The design and construction of an experiment like ATLAS requires a large amount of simulated data in order to optimize the detector design, estimate physics performance, and test the software and computing infrastructure. These samples consists of a large number of simulated events, representing collisions between protons. The full simulation requires the following steps:

- **Event generation**: Hadronic final states using the proton-proton collisions are generated using programs relying on theoretical calculations, phenomenological models and experimental inputs.

- **Detector Simulation**: Interaction of the generated particles inside the ATLAS detector is simulated. Taking into account the real geometry, distribution of material, etc. (CPU time per event=800 kSI2k.seconds, event size=2 MB).

- **Digitization**: The detector response is derived from the particle interactions and it is written in a format compatible with the real output of the detector. In addition, because of the high rate of collisions in the LHC, digitized signals from several simulated events can be piled-up to create samples with a realistic experimental background. The digitized events (with or without pile-up) can now be used to test the software suite that will be used on the real LHC data. (CPU time per event=25 kSI2k.seconds, Event size=2 MB)

- **Reconstruction**: particle trajectories and energies from the detector are reconstructed. Usually final samples to be used by the physicists. (CPU time per event=15 kSI2k.seconds, Event size=1.2 MB)

This chain requires to run different programs with different characteristics in terms of memory usage and CPU consumption. Typically a simulation job run for about 24 hours, while a digitization or reconstruction jobs runs for 3 or 4 hours.

The ATLAS production system distinguishes between two levels of abstraction. On the higher level, input datasets are transformed into output datasets by applying a task transformation. The process of doing this is called a task. Datasets are usually quite large and consist of many logical files. At a lower level of abstraction,
input logical files are transformed into output logical files by applying a job transformation. This process is called a job [4].

The ATLAS production system provides a common framework in which any grid flavor may be integrated, is formed from several individual elements which when plugged together provide the required functionality for the submission, tracking, recovery and validation of jobs. The individual elements of the production system are the following (fig. 1):

- Common database for the production jobs (ProdDB).
- Data management system to data transfer and file cataloging (DDM).
- Common Supervisor (Eowyn).
- Executors developed by middle-ware experts (CondorG, Lexor and Cronus in EGEE).

![Figure 1. ATLAS Production system schema](image_url)

The core of the ATLAS production system is formed by the coupling of the Executor with the Supervisor. The Supervisor provides an interface to the job definition data and metadata associated to the computing resources, and retrieves job specific information (Needed software release, data I/O, number of events, etc.) The Supervisor-Executor system allows these jobs to be passed to one of the grid flavors, and then jobs land at the batch system of one of the ATLAS sites around the globe. Supervisor continues to monitor the state of the submitted job and finally retrieve detailed information about the job once ended. All job info is sent to an ORACLE back-end Database which is used to monitor the production system.

2.1. Executors in EGEE

The executor creates the wrapper files and submit the jobs to the Grid (taking into account the free slots in the sites, etc.) Executors are able to interpret the job related errors and grid specific problems and retrieve the
job after execution. The three executors running in EGEE are:

- **Lexor** Lexor is not much more than a translator of prodsys-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 middle-ware functionalities, and to have a thin, stateless layer (states are already stored in the production database and in the grid middle-ware).

Some manipulation is anyway required, as the mapping between middle-ware and prodsys 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 middle-ware, 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 DDM layer.

- **CondorG** CondorG is standard Grid middle-ware for remote job submission to CEs, and indeed it forms part of the LCG RB. In this case the RB chooses the destination CE, and CondorG submits to the named site. However, when given information about the resources, CondorG can also do the resource brokerage. This information is taken from the BDII and converted into the Condor ClassAd format. The fundamental difference, compared with the original Lexor executor, is that the resource brokerage and the submission are done by separate components. The Negotiator and one or more Schedulers run on different machines, and scalability is achieved by increasing the number of Schedulers only. Furthermore, the scheduler is sufficiently lightweight to run much closer to the UI, perhaps on the same machine.

The interaction with the local Scheduler is therefore much faster. Similarly the status and getOutput requests are instantaneous as the response is like that of a local batch system. There are, however, two perceived deficiencies with this approach. First, if the UI machine hosts the Scheduler then it cannot be turned off, which is inconvenient if the UI machine is, for example, a laptop.

A second concern was the lack of central logging and book-keeping (L&B) when using CondorG. We should stress ”central” because there is in fact a local record of the stages in the job’s life, and a mechanism exists to extract this to a MySQL database. The LCG central L&B has been identified as a potential cause of the poor performance, so not having this architecture is an advantage of CondorG. This does not prevent the L&B information being migrated to a central place, asynchronous to job submission. Lexor was used as the basis for the Lexor-CG executor because its modular design allowed the easy exchange of the LCG submission with the CondorG submission. Everything else, including the run scripts, stage-in, stage-out, validation, etc. remained the same and was re-used. During production operation, improvements to Lexor were also applied to Lexor-CG.

- **CRONUS** Production jobs goes to a scheduler that interacts with the Cronus-Central Manager (CCM). From then the Condor-G glide-ins are submitted to the CE’s and finally ending in to the WN’s, once activated they preserve the Master-Worker relationships, with the worker pulling the production jobs sequentially until the expiry of their lifetimes. The communication between the WN and the CCM is performed via ClassAds and if the glide-in find that the WN requirement are correct, jobs are submitted. This communication with the CCM allows to have a full control and monitoring of the jobs running across the grid.
3. Experience and scope
One of the targets of the production system is to prove the operability of high level distributed computing, since computing demands of the LHC has no precedents. Production of simulated data was planned with a continuous ramp-up in the number of simulated events.
For that reason, since November 2006 the ATLAS simulated production in EGEE is supervised by the EGEE production team, a group of people following a shift system that perform the job and associated data flow monitoring for all the production jobs running in EGEE.
In early November 2006 there was planned a ramp-up exercise for the production jobs (within the Service Challenge 4 framework) to demonstrate the scalability, robustness and power of the production system. The target was to finish 20 M events during November and December 2006, and 40M events during the next quarter (January, February and March 2007). The challenge was successfully covered and the production system showed to be able to cope with the defined targets. This ramp-up exercise ended earlier as expected because the disk resources of almost all the Tier-1 centers were quickly filled. After this saturation of resources, the production system is running in a steady state, which has been shown very constant during the last months (fig. 2). The number of simulated events produced are summarized in (tab. 1).

![Figure 2](image)

**Figure 2.** Finished jobs distribution from October 2006 to January 2007 (three grids). Two main zones are clearly seen: ramp-up period (green) and steady state production (red). Finished jobs peaked at 55000 jobs finished in a day.

An important factor to measure the operability of the system are the efficiencies of the jobs and the efficiency of the WCT, both has been almost continuously improving since the start of the operations. Nevertheless one has to keep in mind keeping in mind that simulated production is a vivid body as new releases/patches appears regularly provoking temporal periods of inefficiency (mainly during software validation periods). Mean efficiencies for the three quarters are shown in (figs. 3).

3.1. Workload management and Data management monitoring
EGEE production operations team look after the jobs running in the grid. Basically there are two main parts to control: Workload and Data management (I/O), hence this two different
| Period  | # events | Comb. job efficiency | Comb. WCT efficiency |
|---------|----------|----------------------|----------------------|
| 4Q2006  | 20M      | 42 %                 | 76 %                 |
| 1Q2007  | 63M      | 59 %                 | 82 %                 |
| 2Q2007  | 38M      | 57 %                 | 90 %                 |

Table 1. Simulated production in EGEE, quarter report. Number of events is extrapolated assuming a mean value of 50 events simulated per job. The combined efficiencies for job and Wall Clock Time (WCT) are the mean value for the three executors actually running on EGEE (Lexor, CondorG and CRONUS).

| Executor | Finished Jobs | Job efficiency | WCT efficiency | Event Production |
|----------|---------------|----------------|----------------|------------------|
| Lexor    | 816029        | 53 %           | 85 %           | 34%              |
| CondorG  | 1039646       | 51 %           | 86 %           | 43%              |
| CRONUS   | 577680        | 61 %           | 86 %           | 23%              |

Table 2. Number of finished jobs, and efficiencies for the jobs and the WCT for the three type of executors: Lexor, CondorG and CRONUS. The numbers are the average since November 2006 ("begin" of 4Q2006) until June 2007 (end of 2Q2007). Last column (event production) show the percentage of produced events by each one of the executors.

![Figure 3](a) Job and (b) WCT efficiencies

...duties are taken by two different groups during the shift period.

*Workload management* is referred to the intrinsic job related problems, and can have many different sources. Most common ones are: **Software** problems at the site: local filesystem, local grid configuration, etc. (usually these errors are not easily found and need special investigation at the sites batch system). **Grid-related** problems: proxy failures, incorrect mappings/permissions etc. Or **ATLAS** specific software errors: wrong transformation, incorrect task definition, etc.

*Data management* is related to the stage-in and stage-out failures. This usually have two different sources, one that is related to the site: SE outages, LFC time-outs, BDii errors, etc. And a different one that is a problem of the global Data Management: missing files in the grid
needed by the jobs, corrupted files that yield error after the size/checksum exploration, etc.

3.2. Job error categorization
Errors can be categorized in three dominant groups:

a) Executor errors: This errors are mainly related to site-specific problems and are particularly difficult to debug as for a large fraction of failures no output is produced. This requires to log into a site and chase the problem at the WN level. Some examples of this type of errors at WMS/CE level are: proxy expired, job cancelled by the batch system or some sophisticated failure in the executor machinery.
Usually is quickly spotted as the job consumed almost no CPU. This is clearly seen in (tab. 3) where the percentage of job error related to the group Executor is 41% but the impact in the lost WCT is very little (3%).

b) Data handling errors: are the most worrying ones, and can be separated in two main groups: Stage-in and Stage-out failures. Both consumes CPU and block the WN while is trying to get or send data.

Stage-in process consist of querying the Local File Catalogues (LFC’s) of the Tier-1 centers, then the most proximal one is chosen, and the files are copied back to the WN using underlying lcg-utils (lcg-cp) directly form the source Storage Element (SE). This may cause some problems in case the LFC are not responding or the SE is not reachable, as the command for staging-in keeps trying until times-out. The impact of stage-in related errors in WCT loss is of about 28%.

Stage-out process is the more critical one as the file cannot be copied back after the successful finishing of the jobs, wasting all the CPU consumed. Fallback solutions has been implemented since some time, at first step the job try to stores the file at the local SE, in case of failure the job try another SE from the same cloud 1. If those two failed CERN (Tier-0) was taken as a final stage and the job try to store the file there. Even with this three layer of fallback, errors in the stage-out (lcg-cr) weights around 32% of the total loss in the WCT.

c) Software errors: this are commonly due to a problem in the task definition or in the transformation, provoking the job to abort unexpectedly, the impact on the WCT has been around the 20% within the considered period, this is because the transformation is executed some time after the job start. Also site software related problems has been counted in this field: error in lfc-mkdir (7.6%) or downloading Pacman Job Transform (2.3%), software missing at the site (1.5%). The weight of this errors in the ATLAS production system is shown in (tab. 3), and the pie charts with the error breakdown are shown in (fig. 4), in both cases the period considered ranges from the 1st. of November 2006 until 30th of June 2007.

4. Operations deployment
EGEE production operations team is composed by about 20 people from all over the globe. There are three production coordinators, eight senior shifters and 6 trainee shifters (plus a team of four shifters that works within the French cloud). Each week there are two production coordinators on shift, plus two senior and two associated trainee shifters.

1 ATLAS is organized in clouds. For a cloud is understood the Tier-1 and its associated Tier-2s
| Error group | Percentage in job failures | Percentage in WCT loss | CPU loss |
|-------------|---------------------------|------------------------|---------|
| Data I/O    | 35 %                      | 65 %                   | 9%      |
| Software    | 9 %                       | 32 %                   | 4%      |
| Executor    | 41 %                      | 3 %                    | <1%     |

**Table 3.** Weight of the dominating errors in the ATLAS production system. In the second column is shown the percentage of job errors (over the total jobs failed), in the third column WCT loss is counted for each of the cases (over the total WCT loss produced by the job failures). The last column show the net impact on the CPU loss by each one of the cases, this has been measured by multiplying the percentage of WCT loss, i.e. mean inefficiency in the WCT from the three executors which is 14% (tab. 1), with the considered error weight.

![Job and WCT error pies](image)

**Figure 4.** (a) Job and (b) WCT error pies. Since 1st November 2006 until 30th June 2007

The duties of the Production coordinators is to assign the tasks defined by the physics coordinators in a proper way to maintain a balanced load on the grid and monitor the overall production activity.

Senior and trainee shifters monitor the production and take care of the assigned tasks, they chase and investigate the potential problems appearing in both, Workload Management and Data Management. After investigation the shifter take some actions depending on the error nature: notify the errors to the responsible group, i.e open a ticket in the Global Grid User Support (GGUS) when finding problems in a site, submit Savannah bugs in case of Data Management issues or found wrong tasks. After the notifications are done, tasks can be paused (if the problem can be solved within a reasonable period of time) or aborted (in this case the task has to be redefined and resubmitted). The production system has a mean number of 5000 jobs per day and it is crucial to have the appropriate monitoring tools and some expertise with the whole system, for that reason it takes a bit to get used to do shifts. However during this nine months of activity three trainee shifters has already been promoted to senior in the last months, giving to the group more flexibility and more expertise.

The internal organization of the operation is based on a weekly phone meeting and an ex-
tremely active mailing lists. Each action done by the shifters is recorded in a web logbook, hence maintaining a kind of historic where everybody can cross check the actions performed. Every three months a face to face meeting is performed in the framework of the ATLAS software weeks.

Since the starting of the joint operations in November 2006 more than 2.7M jobs has been successfully done, yielding an amount of 135M simulated events, more than 3500 mails has been sent to the list, 500 bugs has been reported and around 100 reports had been written.

Now the main objective of our group is to automatized some of the shifter work at Supervisor level and hence lower the load on the shifters, this is of capital importance as the MC production is expected to ramp-up in the next months.

5. Resources

Resources for ATLAS simulated production are spread around more than 50 sites (10 Tier-1s and approximately 40 Tier-2s), yielding a total power of 26 MSI2k.month and more than 100TB of disk. The merged wall-time days from all sites has been increasing, reaching an average of around 3500 days/day in the last months in (fig. 5) is shown the finished vs. failed wall-time and in (fig. 6) the finished wall-time is shown for the three different Grid flavors.

![Wall-time per day during 1st November 2006 until 30th June 2007, darker bars show the failed CPU time.](image)

6. Summary and Conclusions

The EGEE production system showed to be able to cope with the requirements from the ATLAS experiment, although some effort should be made toward automation in order to reach scalability for future challenges.

The CPU wall time has been continuously improving obtaining an efficiency of about 90% during the past three months, this is the result of a better understanding of the whole system and the continuous improvement in the machinery, from executors to site reliability. However, there is still room to improve in the data input/output, and the job efficiency.

The EGEE production system is expecting new advances in middle-ware, i.e. the new SRM
Figure 6. Finished Wall-time per day during 1st November 2006 until 30th June 2007, the three Grid flavors are shown together with the CRONUS instance.

(Storage Resource Manager), FTS 2.0 (File Transfer Service), etc. This would yield a more robust infrastructure and a higher performance system that will impact in a better and more efficient simulated event production.

[1] G. Poulard Experience on large scale production on the grid, CHEP 2006.
[2] J. Kennedy et al. The ATLAS production system
[3] A. De Salvo (INFN Roma), G. Negri (CNAF Bologna), D. Rebatto, L. Vaccarossa (INFN Milano) LEXOR, the LCG-2 Executor for the ATLAS DC2 Production System CHEP 2004.

[4] Computing TDR: http://atlas-proj-computing-tdr.web.cern.ch/atlas-proj-computing-tdr/Html/Computing-TDR-50.htm
[5] http://grid.desy.de/testbed/EDG/CE.html
[6] S. Padhi. Production using CRONUS, ATLAS SW March 2007, presentation &confId=5060
[7] All statistics has been taken from the ATLAS monitoring web: http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/php/DbAdmin/Ora/php-4.3.4/proddb/monitor/Home.php
[8] LCG production Wiki: https://twiki.cern.ch/twiki/bin/view/Atlas/LcgProduction