How to keep the Grid full and working with ATLAS production and physics jobs

A Pacheco Pagés, F H Barreiro Megino, D Cameron, F Fassi, A Filipcic, A Di Girolamo, S González de la Hoz, I Glushkov, T Maeno, R Walker and W Yang on behalf of the ATLAS Collaboration

Abstract. The ATLAS production system provides the infrastructure to process millions of events collected during the LHC Run 1 and the first two years of Run 2 using grid, clouds and high performance computing. We address in this contribution the strategies and improvements that have been implemented to the production system for optimal performance and to achieve the highest efficiency of available resources from operational perspective. We focus on the recent developments.

1. Introduction

The production system of the ATLAS experiment [1] provides the infrastructure to process millions of events collected during the LHC Run 1 and the first two years of Run 2 using grid, clouds and high performance computing. We address in this contribution the strategies and improvements that have been implemented to the production system for optimal performance and to achieve the highest efficiency of available resources from operational perspective. We focus on the recent developments.

1 Also at Port d’Informació Científica (PIC), Campus UAB - Edifici D. E-08193 Bellaterra (Barcelona), Spain
At the beginning of LHC data taking, access to data and the networking infrastructure was still under development. To operate the worldwide distributed computing system, the MONARC model [2] was chosen by the LHC Computing Grid project [3]. This model is based on a hierarchy of Tiers, where Tier-1s are associated with Tier-2 computing centres. The sites are distributed worldwide but had to provide an overall homogeneous infrastructure, therefore they followed some common requirements requested by the ATLAS experiment as well as the WLCG computing grid project.

Over time the network infrastructure has greatly improved and the diversity of the resources has increased. ATLAS relies on large contributions from non-grid resources like High Performance Computers (HPCs), commercial and private clouds and other opportunistic resources that require additional operational and development effort [4]. The main aspects of processing huge amounts of data on distributed resources that require special handling from the operational point of view are: memory, worldwide distribution of jobs, fast processing of critical tasks and loss of data.

The computing power used by the ATLAS Production system can be seen in figure 1, where the time frame is the LHC Run2 period 2015-2016, and figure 2, where the time frame has been expanded to the start of LHC operation in 2011. From the figures it can be seen that ATLAS was able to fill the resources over and above the pledged values assigned by the LCG Resources Review Board shown as a grey line on the plots, to extract physics results from experimental data taken by the ATLAS experiment.
Figure 1: Wallclock consumption in HEPSPEC06 units of all ATLAS jobs per activity type and including the allocated pledge as a flat grey line. The plot is from Run 2 period that started in 2015.
2. Jobs with large physical memory requirements
In day-to-day operations the problem of the memory consumption of ATLAS jobs is a common issue. Software gets more complicated, there is more complexity in the detector layout, the energy of the collision beams increases as well as the number of simultaneous collisions in a bunch crossing.

Since the start of LHC operations in 2009, the ATLAS experiment has requested that sites contribute computing resources with 2GB of physical RAM per core. This value cannot be easily changed scaling any increase on this memory value to the more than 300,000 cores used simultaneously in ATLAS would represent a huge increase in the computation operations budget.

According to job accounting data collected in September 2016, ATLAS jobs consume on average 1.7GB per core, with some jobs requiring just 1-1.5GB/core and others up to 4GB/core. There is a clearly the case of high memory usage with Heavy Ion collisions, but it can also appear in proton-proton collisions. On average 12% of the finished jobs used more than 2GB per core and they were mostly reconstruction jobs.

To cope with this situation ATLAS requested the largest Tier-1 and Tier-2 sites to offer
dedicated resources to run high memory jobs. Some of the resources provided by the sites support running jobs using up to 64 GB per core, usually at a cost of running fewer jobs per physical server.

In addition to this, ATLAS requested in 2015 the Tier-1 and the Tier-2 sites to offer resources allowing multi-core processing, and that ATLAS software supports running in a multi-core mode. Multi-core support is available for all workflows with high memory demands: simulation, data processing, and trigger. In reference [5] the different memory limits one needs to deal with running jobs like Resident Set Size (RSS) memory, Proportional set size (PSS) memory and virtual set size (VSS) memory are detailed.

The strategy today in ATLAS is to use multi-core processing as much as possible. This enables sharing memory pages between multiple threads which results in less memory per core.

In order to optimize the usage of the computing infrastructure, the production system developers were requested in early 2015 to introduce a mechanism to estimate the memory and CPU requirements of future jobs from real values in order to manage properly those jobs with tasks demanding large memory or processing time. For every new task request in ATLAS, only the first 10 jobs are submitted, these jobs are also called scouts. From the accounting of the scout jobs and adding some safety margins, the rest of the payload is submitted with accurate computing resources requirements. In this way limits are calculated automatically from measured values.

However, there are and there will be jobs that exceed the safety margins calculated from real-time accounting of scout jobs. To cope with these jobs a new mechanism was developed called the “retry module” which detects the reason for which a job has failed and follows a decision tree mechanism. If the job failed due to the lack of memory or excessive CPU consumption, then a new job is resubmitted with higher memory limits.

3. Worldwide distribution of jobs
The next operational challenge in ATLAS operations is the optimal distribution of the computing workload. The typical complaint from sites is that they are not filled with jobs while other sites are full. There can be many reasons why a site has no load, but in some of them the cause was connected to the initial grid computing model, which was very hierarchical.

In ATLAS Run 1 task requests were assigned to a given Tier-1 site which had statically appointed Tier-2 sites to distribute the workload and collect the output. In order to cover cases where the sites had their assigned Tier-1 offline or were just not getting payload through lack of workload in the region, the ATLAS system allowed some sites to get jobs from other tasks assigned to other Tier-1 regions. This mechanism, called Multiple Cloud Processing (MCP), mitigated the problem of empty sites and in fact at the end all sites in ATLAS had this capability, at the expense of an increase in chaotic transfers from one part of the world to another. In order to add some order within the mesh some limits were introduced so that only the most loaded Tier-1 regions were providing jobs to the less loaded. This was needed because some extreme cases were observed, where sites from some regions were running jobs from other regions and the assigned payload to their Tier-1 sites was taken by external sites.
Now, there is only one region in ATLAS called WORLD which replaces regions based on Tier-1s with statically associated Tier-2 sites. All sites belong to the WORLD region.

Sites can be assigned two roles for processing a task request [6][7]:

1. Nucleus - Expected to be the main processing site for some of the payload, certainly the merging of the outputs and the location of the primary copy of the output.
2. Satellite - Expected to be a processing site for the payload.

This means that the role of sites processing a task request is decided dynamically at the very last moment. Any site with sufficient storage capacity can potentially be a nucleus, and all sites can be satellites. For each task defined in the production system a nucleus is selected with some criteria including the availability of input files needed, availability of free disk space to hold primary copy of the output and availability of processing slots.

Then for each task request, 10 other sites are selected with a role of “satellites” where the jobs are distributed based on availability of CPU resources and connectivity to the nucleus associated to the task. The result is very satisfactory because real-time network bandwidth and availability of resources is taken into account at the decision level. This new nucleus/satellite model has made the previous Tier-1/Tier-2 distinction largely irrelevant in ATLAS.

4. Critical processing
Some of the workflows in ATLAS require a critical time of completion, this means that it is crucial to complete them within some time limit.

A worldwide distributed production system comes with uncertainties on when an individual job can finish and when a task can be completed. Sometimes there are issues with the network, the sites, with the batch system running the job or the local or remote storage where the input data is located. The system already detects some of these issues, but actions are effective only after 12-24 hours. From the perspective of instantaneous processing power, if we are running in ATLAS 300,000 cores simultaneously, then we should be able to complete few thousand critical jobs within hours. To run critical tasks as soon as possible, the chance that a job can start running as soon as possible must be maximised.

In ATLAS, as in all other LHC experiments, a paradigm of late binding of payload to jobs is used. This means that jobs are assigned to the available free resources, vanilla pilot jobs are submitted to the resources and after the job is started the payload is assigned. This is very effective as the probability of success when job starts is very high. To mitigate the dependency of critical jobs on the resources the concept of submitting in parallel multiple instances of the same job was developed.

In Run 1 the strategy for critical task requests was based on assigning manually the tasks with the highest priority to reliable Tier-1 sites with locally available disk copies of the input files and requesting the jobs to run only at those sites. In Run 2 the strategy has been dramatically improved by submitting jobs from critical tasks more than once to the production system. When the first of the identical jobs start the other queued jobs are killed. In this way the probability to finish critical tasks in less than 12 hours has increased dramatically.
5. Lost files
The last operational issue that all big distributed systems face during day-to-day operations is the loss of data. This presents the worst risk in a workflow.

In principle, critical data is protected against loss via multiple copies of the files. But replication of files is usually the last step in the workflow. Intermediate files or important files not yet replicated are at risk of being lost.

The ATLAS Production system relies on the availability and reliability of the order of 150 storage systems distributed worldwide. The storage systems have checksum mechanisms to ensure that files are transferred safely and the sites usually have redundant storage. One common scenario is the case where some files are successfully created in one site then copied successfully to the Tier-1 site (or nucleus) with a successful checksum verification. Then, the storage at the Tier-1 has an operational issue (software, hardware) and the file is corrupted or unavailable. At some point later after the original copy of the file has been deleted the ATLAS production system tries to use the Tier-1 replica. This fails and the file is automatically declared bad. If there are no more replicas, the file is lost and ATLAS needs to declare the file lost and possibly recreate it.

If a file is lost and there are no replicas the first step is to contact the submitter of the affected task request. Sometimes he resubmits again the whole task and obsoletes the task with the files lost. In some cases the data has a new sample with a new version of the software, the lost files were part of an intermediate step or not data to be used for analysis, just for validation. In these cases, there is no need to replace the lost file, but the databases in ATLAS need to be updated to correct the number of events, luminosity and other bookkeeping.

If the lost file is needed, then a new replica of the job that created the lost files is resubmitted, and the new created files are added to the output datasets of the task request, making data complete again. But some cases are very difficult as the lost file can be the input to other tasks which merge data or produce other outputs which need to be recreated. Extra care must always be taken that the operation will not produce duplicate outputs.

6. Conclusions
We have presented several examples of operational issues which affect the ability of keeping the Grid full, the management of ATLAS production and physics jobs, and the current strategies from the ATLAS system to cope with them.

- Memory limits are profiled from real executed jobs, if jobs still fail due to memory they are resubmitted individually with higher limits.
- The classical hierarchical grid topology based on the MONARC model is replaced by one flat region called WORLD to which all sites belong. Each task is processed by one site acting as nucleus and 10 sites acting as satellites.
- For critical tasks multiple copies of identical jobs are submitted to be sure they start running as soon as possible, in order to mitigate possible last minute site issues. Once the first instance starts running, the duplicate jobs are cancelled.
Lost files are replaced if necessary by resubmitting the job or jobs that created them taking into account the dependencies on tasks using the lost data.

Acknowledgements
This work was partially supported by MINECO in Spain under grants FPA2013-47424 and FPA2016-80994, which include FEDER/ERDF funds from the European Union. PIC is maintained through a collaboration of IFAE and CIEMAT. The support of the FEDER/ERDF fund 2007-13 is gratefully acknowledged.

References
[1] ATLAS Collaboration 2008 The ATLAS Detector, JINST 3 (2008) S08003
[2] CERN/LHCC 96-43,19 1996 ATLAS Computing Technical Proposal
[3] CERN-LHCC-2005-024 2005 LHC Computing Grid: Technical Design Report
[4] Filipcic A et al. 2017 Benefits and performance of ATLAS approaches to utilizing opportunistic resources Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016.
[5] Forti A et al. 2017 Memory handling in the ATLAS submission system from job definition to sites limits Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016.
[6] Barreiro Megino FH et al. 2017 ATLAS WORLD Cloud and networking in PanDA. Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016.
[7] Maeno T et al. 2017 PanDA for ATLAS distributed computing in the next decade Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016.