ATLAS Grid Data Processing: system evolution and scalability

D Golubkov\textsuperscript{1,2}, B Kersevan\textsuperscript{3}, A Klimentov\textsuperscript{4}, A Minaenko\textsuperscript{1}, P Nevski\textsuperscript{4}, A Vaniachine\textsuperscript{5} and R Walker\textsuperscript{6} for the ATLAS Collaboration

\textsuperscript{1} Experimental Physics Department, Institute for High Energy Physics, Protvino, 142281, Russia
\textsuperscript{2} Physics Department, CERN, CH-1211 Genève 23, Switzerland
\textsuperscript{3} Experimental Particle Physics Department, Jozef Stefan Institute, Jamova 39, 1001 Ljubljana, Slovenia
\textsuperscript{4} Physics Department, Brookhaven National Laboratory, Bldg. 510A, Upton, NY 11973, United States of America
\textsuperscript{5} High Energy Physics Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, United States of America
\textsuperscript{6} Fakultat fur Physik, Ludwig-Maximilian-Universität München, Am Coulombwall 1, DE-85748, Garching, Germany

E-mail: vanachine@anl.gov

Abstract. The production system for Grid Data Processing handles petascale ATLAS data reprocessing and Monte Carlo activities. The production system empowered further data processing steps on the Grid performed by dozens of ATLAS physics groups with coordinated access to computing resources worldwide, including additional resources sponsored by regional facilities. The system provides knowledge management of configuration parameters for massive data processing tasks, reproducibility of results, scalable database access, orchestrated workflow and performance monitoring, dynamic workload sharing, automated fault tolerance and petascale data integrity control. The system evolves to accommodate a growing number of users and new requirements from our contacts in ATLAS main areas: Trigger, Physics, Data Preparation and Software & Computing. To assure scalability, the next generation production system architecture development is in progress. We report on scaling up the production system for a growing number of users providing data for physics analysis and other ATLAS main activities.

1. Introduction

Facing a petascale challenge of processing data flowing from the detector [1], the ATLAS collaboration is relying on the infrastructure deployed in the framework of the Worldwide LHC Computing Grid (WLCG). Following the massive data processing on the grid, thousands of scientists analyze ATLAS data in search of new discoveries.

In 2011 the LHC achieved instantaneous and integrated luminosity values that far exceeded expectations. 2011 was also an excellent year for ATLAS. Enabled by continuous software performance improvements and flexibility of the ATLAS distributed computing, the collaboration managed unexpected levels of pileup and larger physics datasets, coped with higher trigger rates, and
generated more simulated datasets to support the broad range of physics analyses, thereby making possible a productive and timely physics program. ATLAS distributed computing manages over fifty petabytes of data on the grid [2]. ATLAS leads the WLCG usage in the number of jobs, processed data volume and in core-hours, as shown in figure 1.

2. Grid Data Processing
The ATLAS experiment uses PanDA [3] for managing the workflow for all data processing jobs on the WLCG. In the WLCG distributed computing environment, PanDA provides transparency of data and processing. As a result, ATLAS Grid Data Processing (GDP) sees a single computing facility that is used to run all data processing for the experiment, even though the sites are physically located all over the world. The GDP production system supports a diverse range of workflows handling centrally ATLAS petascale data reprocessing (reconstruction of LHC data) and Monte Carlo production (full and fast simulations, digitization and reconstruction of simulated data).

2.1. Managing Complexity
In addition to challenges in petascale data processing, computing places a significant burden on physicists to configure and manage the large number of parameters and options provided in the ATLAS software. The laborious process of steering the data processing application by providing physics parameters is manual. The error-prone manual process does not scale to the GDP challenges.

To reduce human errors and automate process of defining millions of jobs for execution by PanDA, we developed a scalable meta-application for ATLAS knowledge management ("Knowledgement") of Task Requests (AKTR). The AKTR manages configuration parameters used for massive grid data processing tasks (sets of similar jobs). The meta-application assures scalable management of ATLAS-wide knowledge of GDP production complexity and guaranties reproducibility of results. The AKTR management of the institutional knowledge of the process of tuning and setting up data processing tasks resulted in major gains in efficiency and productivity of the GDP production infrastructure.

1 http://www4.egee.cesga.es/accounting/egee_view.html
Thanks to AKTR, the task became a main unit of computation (instead of a job) in ATLAS petascale data processing. In ATLAS data management [4], physicists deal not with individual files but with large datasets - collections of similar files. Similarly, a task – not a job – is a main unit in the GDP production system. Most of the production system functions are fully automated: throttled job submission, e-mail notification in case of errors in user’s task request, resubmission of jobs with transient failures, no resubmission of jobs that fail repeatedly, etc. Status of the production tasks is monitored via web pages integrated with PanDA monitoring. Demonstrating scalability, the production system sustained double exponential growth of the number of tasks requests (figure 2).

2.2. Data Reprocessing

It takes about three million core-hours for the reconstruction of one petabyte of ATLAS data with 1B collision events from the LHC. It took four weeks to complete the full MapReduce-like workflow reprocessing 0.9B events. The combined ATLAS workflow used more than 100k CPU-cores, consuming at peak about 0.2 petaflops — a performance on the Top100 list of supercomputers.

Splitting of a large data processing task into jobs (small data processing tasks) is similar to the splitting of a large file into smaller TCP/IP packets during the FTP data transfer. Generally, during a file transfer, application users are not concerned how many TCP/IP packets were dropped. The network performance is an area of concern of the network researchers and engineers. Similarly, physicists do not care about transient job failures, when GDP production system delivers “six sigma” quality performance¹ for the petascale data reprocessing campaigns with hundreds of tasks [5].

Automatic job resubmission recovers transient event losses at the expense of CPU time used by the failed jobs. Distribution of tasks ordered by CPU time used to recover transient failures is not uniform: most of CPU time required for recovery was used in a small fraction of tasks (figure 3). In 2010

¹ Corresponding to event losses below the $10^{-8}$ level.
In reprocessing, the CPU time used to recover transient failures was 6% of the CPU time used for reconstruction. In 2011 reprocessing, the CPU time used to recover transient failures was reduced to 4% of the CPU time used for the reconstruction.

2.3. Group Production
The GDP production system empowered further data processing steps on the Grid performed by dozens of ATLAS physics groups with coordinated access to computing resources worldwide, including additional resources sponsored by regional facilities. Unlike major reprocessing campaigns that are conducted only a few times per year or less, the GDP production for physics groups process the whole available dataset once every few weeks, providing further improvements in the data used for ATLAS physics analysis shortly after the reprocessing or data taking.

2.4. Simulations
Excellent LHC performance in 2011 resulted in the ability to address a much wider range of physics analyses, with a higher level of precision, surpassing the most optimistic expectations. In addition, detailed physics studies established that the simulation is of unprecedented quality compared to previous generations of experiments, describing the data quite well in most analyses. These two facts together significantly enhanced ATLAS physics output in 2011, and they motivated production of higher than foreseen simulation statistics. Not counting single particle events, 3B full and 0.7B fast simulated events were produced, with seven digitization and reconstruction campaigns processed 4.9B events in 2011. Monte Carlo production activities dominate ATLAS CPU consumption (figure 4).

2.5. Trigger Validation
The production system was also adopted for the trigger reprocessing, which is performed to validate new trigger menus and/or software releases during data taking. In 2012 ATLAS is again confronted with much higher event pileup than previously planned. The increase in LHC centre-of-mass energy to 8 TeV and squeezing the beams as tightly as possible increases the luminosity. In preparation for the
LHC 2012 operations, more than seventy trigger reprocessings validated new versions of trigger software and menus assuring smooth start of ATLAS data taking under new LHC conditions.

3. System Evolution
The system evolves to accommodate a growing number of use cases and new requirements from our contacts in ATLAS main areas: Trigger, Physics, Data Preparation and Software & Computing (table 1). The system evolution is driven by their requirements, including “good in current and should be kept.”

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Frequency</th>
<th>Workflow Length</th>
<th>Number of Tasks</th>
<th>Tasks Duration</th>
<th>Same Tag for Many Tasks</th>
<th>Data Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulations</td>
<td>Several campaigns per year</td>
<td>Long</td>
<td>Thousands</td>
<td>Months</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Reprocessing</td>
<td>Yearly</td>
<td>Medium</td>
<td>Hundreds</td>
<td>Weeks</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Group Production</td>
<td>Weekly</td>
<td>Short</td>
<td>Thousands</td>
<td>Days</td>
<td>yes</td>
<td>yes/no</td>
</tr>
<tr>
<td>Trigger Validation</td>
<td>Daily</td>
<td>Medium</td>
<td>Dozens</td>
<td>Hours</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

The next generation production system architecture development is in progress, accommodating new requirements:

- **Forking**: Capability to process a part of the input dataset with one setup and the other part with a different setup.
- **Extensions**: Capability to identify tasks, which do not have sufficient success rate to extend them automatically. Care should be taken to avoid physics bias introduced by automatic task extensions.
• **Speed up**: Automatic speed up and finishing tasks that are close to the end (missing a few jobs) in absence of physics bias in the missing jobs.
• **Force-finish**: When both transient or repeated job failures delay completion of urgent task, the user should be able to 'force-finish' the task in absence of physics bias in the failures.
• **Scout probing**: Update the user-provided CPU and memory estimates with actual values used by scout jobs.
• **Recovery**: Capability to re-run jobs if output files are lost, re-generating automatically the upstream chain.
• **Transient & Intermediate**: Flexibility not to save the requested task outputs or request saving of the intermediate data processed by the task jobs.

### 3.1. Design Principles

The design of the next generation production system architecture is driven by the following principles:

- **Flexibility.** As the requirements update process works well, the list of requirements and use cases continue to grow. Accommodating that growth, the next generation production system architecture must be flexible by design.
- **Isolation.** Avoiding inherent fragility of the monolithic systems, we adopted another core design principle – isolation. Separating core concerns, the production system logic layer will be separated from the presentation layer, so that users will have a familiar but improved interface for task requests.
- **Redundancy.** A core concern is scalability, required to assure eventual consistency of the distributed petascale data processing. Since in a scalable system, the top layers should not trust the layers below, we retain the redundancy.

### 3.2. DEfT: Dynamic Evolution for Tasks

The GDP production system architecture grew organically. Now obsolete, legacy requirements constrained the design and implementation. The next LHC shutdown provides an opportunity to rethink the architecture (figure 5). The GDP production system will be enhanced with a new major component — the Dynamic Evolution for Tasks (DEfT). Behind the familiar AKTR interface the new DEfT architecture will emerge from the 2013 shutdown. Figure 6 shows DEfT refactoring that makes implementations maintainable, consistent and scalable while retaining core capabilities, which valued

---

**Figure 5.** High-level representation of the GDP production system architecture.  
**Figure 6.** Modular DEfT architecture.
most: knowledge management of configuration parameters, reproducibility of results, orchestrated workflow and performance monitoring, dynamic workload sharing, automated fault tolerance, petascale data integrity control and scalable database access on the grid [6].

4. Conclusions

The GDP production system fully satisfies the requirements of ATLAS data reprocessing, simulations, and production by physics groups. The evolution of the production system is driven by the updated requirements, including “good in current and should be kept.” The next LHC shutdown provides an opportunity for refactoring the production system, making implementations more maintainable, consistent and scalable, whilst retaining those core capabilities valued most by the users. We collected the requirements and identified core design principles for the next generation production system. Prototyping is in progress, to scale up the production system for a growing number of tasks processing data for physics analysis and other ATLAS main activities.

Acknowledgments

We thank all our colleagues who contributed to ATLAS Grid Data Processing activities.

References