

Overview of ATLAS PanDA Workload Management

T. Maeno ¹, K. De ², T. Wenaus ¹, P. Nilsson ², G. A. Stewart ³, R. Walker ⁴,
A. Stradling ², J. Caballero ¹, M. Potekhin ¹, D. Smith ⁵, for The ATLAS
Collaboration

¹ Brookhaven National Laboratory, NY, USA

² University of Texas at Arlington, TX, USA

³ Department of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ,
UK

⁴ Ludwig-Maximilians-Universitat Munchen, Munchen, Germany

⁵ SLAC National Accelerator Laboratory, CA, USA

tmaeno@bnl.gov

Abstract. The Production and Distributed Analysis System (PanDA) plays a key role in the ATLAS distributed computing infrastructure. All ATLAS Monte-Carlo simulation and data reprocessing jobs pass through the PanDA system. We will describe how PanDA manages job execution on the grid using dynamic resource estimation and data replication together with intelligent brokerage in order to meet the scaling and automation requirements of ATLAS distributed computing. PanDA is also the primary ATLAS system for processing user and group analysis jobs, bringing further requirements for quick, flexible adaptation to the rapidly evolving analysis use cases of the early datataking phase, in addition to the high reliability, robustness and usability needed to provide efficient and transparent utilization of the grid for analysis users. We will describe how PanDA meets ATLAS requirements, the evolution of the system in light of operational experience, how the system has performed during the first LHC data-taking phase and plans for the future.

1. Introduction

The Production and Distributed Analysis (PanDA) system has been developed to meet ATLAS production and analysis requirements for a data-driven workload management system capable of operating at LHC data processing scale. ATLAS production and analysis place challenging requirements on throughput, scalability, robustness, minimal operational manpower, and efficient integrated data/processing management. The key features of PanDA are as follows:

- Apache and Oracle based central task queue and management
- Use of grid and/or farm batch queues to pre-stage job wrappers (pilots) to worker nodes
- Tight integration with the ATLAS Distributed Data Management (DDM) system [1] working with datasets
- An integrated monitoring system [2] for production and analysis operations, user analysis interfaces, data access, and site status
- Support of OSG, EGEE/EGI and NDGF middleware stacks
- Highly automated system requiring low operation manpower

- Well developed security mechanisms [3]

PanDA was originally developed for ATLAS and is being extended as a generic high level workload manager usable by non-ATLAS OSG activities as well. This paper reports here the PanDA system overview, workload management for production and analysis, the current status and future plans.

2. PanDA System Overview

Figure 1 shows a schematic view of the PanDA system. Jobs are submitted to the PanDA server via a simple Python/HTTP client interface. The same interface is used for ATLAS production, distributed analysis, and other general job submission. The PanDA server is the main component which provides a task queue managing all job information centrally. The PanDA server receives jobs through the client interface into the task queue, upon which a brokerage module operates to prioritize and assign work on the basis of job type, priority, input data and its locality, and available CPU resources. The autopyfactory pre-schedules pilots to OSG and EGEE/EGI grid sites using Condor-G. Pilots retrieve jobs from the PanDA server in order to run the jobs as soon as CPU slots become available. Pilots use resources efficiently; they exit immediately if no job is available and the submission rate is regulated according to workload. Each pilot executes a job, detects zombie processes, reports job status to the PanDA server, and recovers failed jobs. Ref.[4] describes the details on pilots. For NDGF, the ARC control tower [5] retrieves jobs from the PanDA server and sends the jobs together with pilot wrappers to NDGF sites using ARC middle-ware.

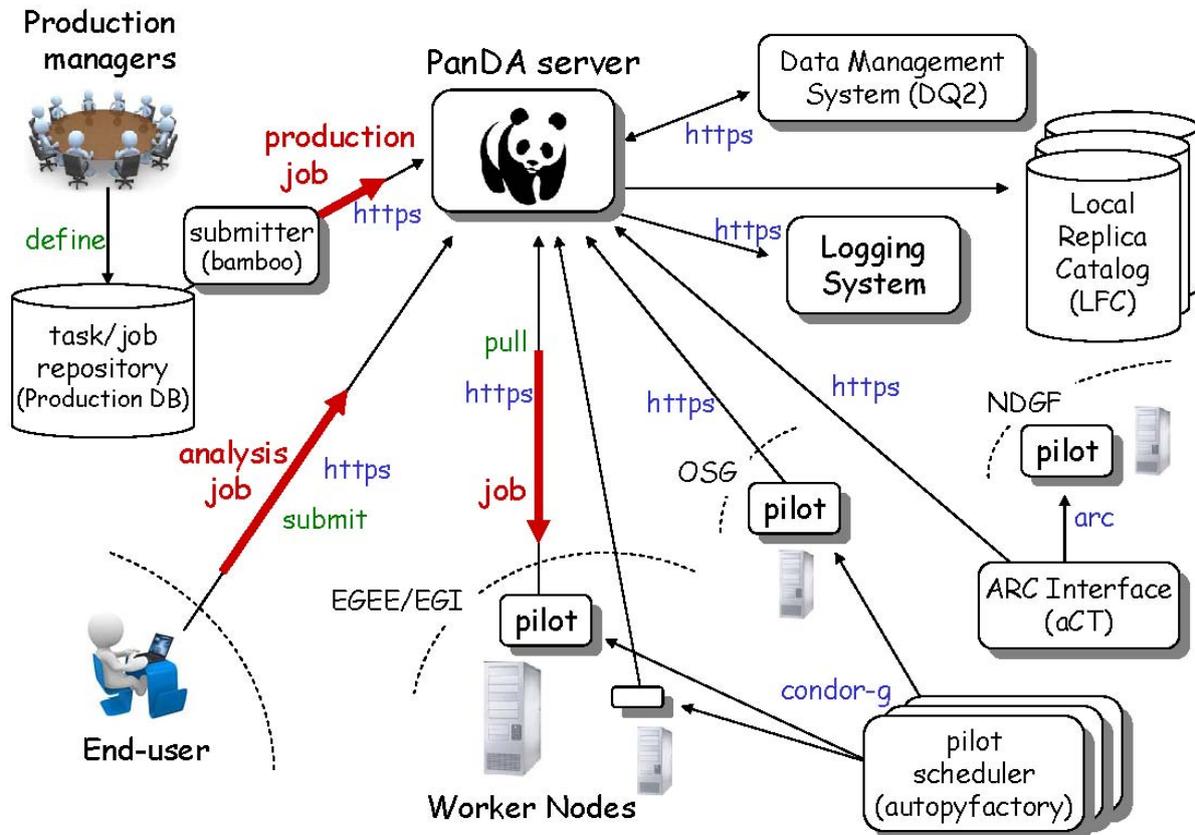


Figure 1. Schematic view of the PanDA System

2.1. Workload Management for Central and Group Production

The purpose of central and group production is to produce data for overall ATLAS collaboration and/or a particular physics working group. Production tasks are centrally defined on the basis of physics needs and resource allocation in ATLAS. ATLAS has one Tier-0 site, ten Tier-1 sites and many Tier-2 sites. A cloud is composed of one Tier-1 site and associated Tier-2 sites, or, at CERN, contains only the Tier-0 site. Tasks are assigned to clouds based on the amount of disk space available on the Tier-0/1 storage element (SE), the locality of input data, available CPU resources, the Memorandum of Understanding (MoU) share which specifies the contributions expected from each participating institute, and downtime of the Tier-0/1. Also, the amount of queued tasks with equal or higher priorities is taken into account so that high priority tasks can jump over low priority tasks even if the cloud has already been occupied with low priority tasks.

Tasks are automatically converted to many jobs for parallel execution. Once a task is assigned to a cloud, jobs are assigned to sites in the cloud based on CPU and I/O usage pattern of the jobs, software availability, the amount of disk space available on the local SE, sizes of scratch disk and memory on worker nodes, site downtime, the ratio between the number of running jobs to the number of queued jobs, the number of input files pre-existing on the local SE. Assignment of jobs to sites is followed by the dispatch of input files to those sites. The system has a pipeline structure running data-transfer and job-execution in parallel. Atlas Distributed Data Management (DDM) system [1] takes care of actual data-transfer: The PanDA server sends requests to DDM to dispatch input files to Tier2s or to pre-stage input files from a tape storage at Tier0/1. DDM transfers or pre-stages files, and then sends notifications to the PanDA server. Jobs wait in the task queue after they were submitted to PanDA, and get activated when all input files are ready. Pilots pick activated jobs and can run immediately since DDM has already staged the input files. Similarly, pilots copy output files to local storage and exit immediately to release CPUs. Then the PanDA server sends requests to DDM to aggregate output files to the Tier1 SE over the grid. That is, pilots occupy CPUs only during running jobs and accessing the local SE.

2.2. Workload Management for Analysis

PanDA is designed for production and analysis. This means that both production and analysis use the same software, monitoring system and facilities, and thus no duplicated manpower is required for maintenance. However, one of the most significant differences is policy for data-transfers. DDM transfers files over the grid for production, while analysis does not trigger file-transfers since analysis jobs are sent only to sites where the input files are already available. This is mainly because analysis jobs are typically I/O intensive and run on many files. Priorities and quotas are calculated per user, or per working group if the user submits jobs for a particular working group. Working groups can be defined in VOMS or in the PanDA database. Each site can provide regional CPU/storage resources by using budgets beyond ATLAS MoU share in addition to official resources. Each site/country/cloud can decide how those resources are used. A typical usage policy is to run jobs for users in a particular country group when their jobs are queued, otherwise, run other users' jobs.

Figure 2 shows a schematic view of workflow for analysis jobs. The user submits a user task (job set) that is split to multiple job subsets according to localities of input datasets and available CPU resources at sites. A job subset is sent to a site where input files are available, i.e., if input files are distributed over multiple sites there will be multiple job subsets and they will be sent to multiple sites. Each job subset is composed of one compile job and many execution jobs. The compile job receives source files from the user through the PanDA server to produce binary files. Execution jobs are triggered once the binary files are put into the local storage element (SE) by the compile job. The execution jobs retrieve input files and the binary files from the local SE, run jobs in parallel, produce and add output files to an output dataset. Output datasets are added to an output dataset container and

users can access the dataset container using standard DDM tools/services [6]. As mentioned, analysis jobs go to sites where input data is available. This means that job distribution depends intimately on data distribution that follows the ATLAS computing model. It is difficult to distribute popular data uniformly since the usage pattern of data is highly unpredictable. Therefore, the policy-based data distribution causes unbalanced job distribution and also makes redundant replicas that are not actually used. That is, CPU and storage resources are not used very efficiently. In order to solve those problems, the PanDA Dynamic Data Placement (PD2P) has been developed. The idea is that PanDA triggers replication of input data based on usage pattern of analysis jobs. The fundamental strategy of PD2P is to make many replicas for popular data and to send data to idle sites. Stuck jobs are periodically reassigned to other sites since PD2P may have replicated datasets to idle sites during jobs are waiting in the queue.

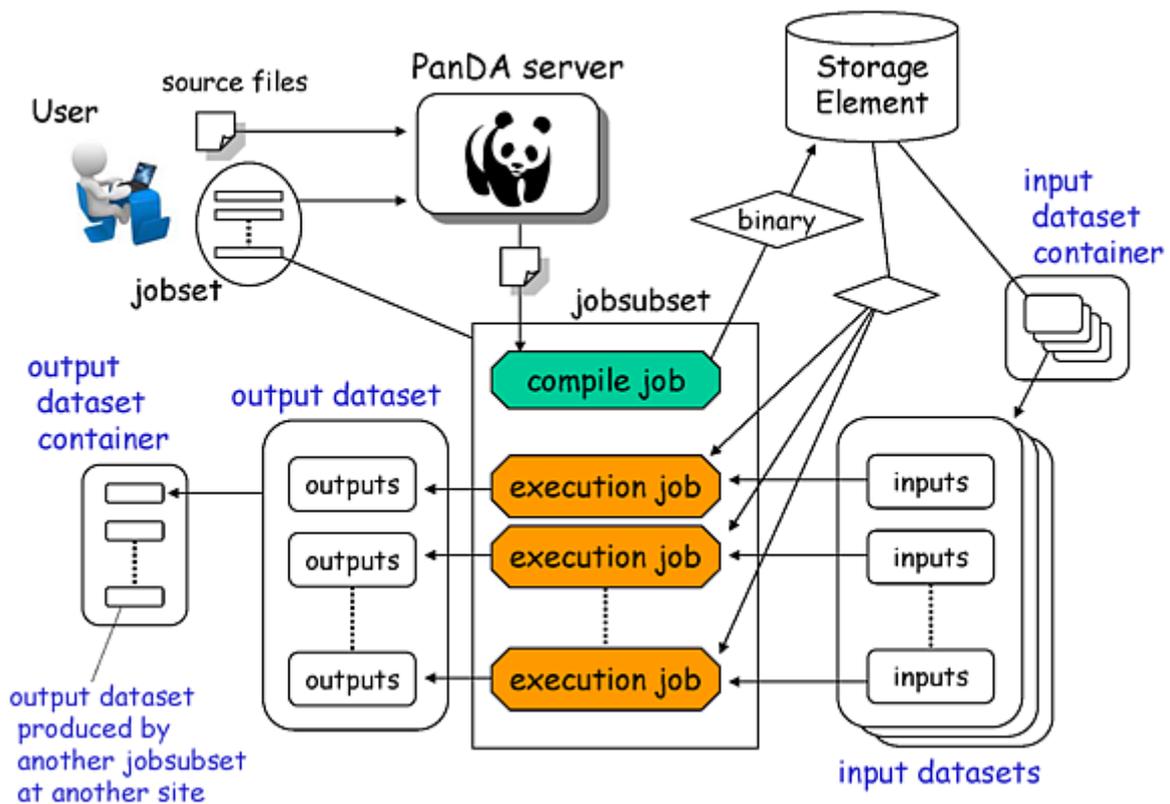


Figure 2. Workflow for Analysis jobs

3. Current Status and Future Plans

Figure 3 shows the number of production jobs running concurrently in 2009 and 2010. We can see a decrease of the maximum number of running jobs to 40k from 50k. This is because CPU resources have been shared with other VOs who have been becoming more active and also ATLAS analysis activities have increased by factor of 10. There are large fluctuations in 2010 due to a lack of defined tasks. With the start of LHC data taking, ATLAS software had been updated frequently. Once a new version of ATLAS software was released it had to be validated before bulk production, which caused a

temporary shortage of tasks. The single job failure rate of production jobs was about 9% in 2010. The breakdown is 3.6 % from ATLAS software problems and 5.4% from site problems. Although site failures are still significant, automatic retries help in many cases and thus those failures are hidden from the task requester.

Figure 4 shows the number of analysis jobs running concurrently in 2009 and 2010. We can see factor of 10 rise compared to one year ago in this figure. Analysis activities on PanDA have increased steadily, which clearly shows usefulness of PanDA for physics analysis. The single job failure rate of analysis jobs has been improved to 21% in 2010, down from 36% in 2009 thanks to good user support by the ATLAS distributed analysis support team. The major causes of failures are software problems since end users often submit problematic jobs.

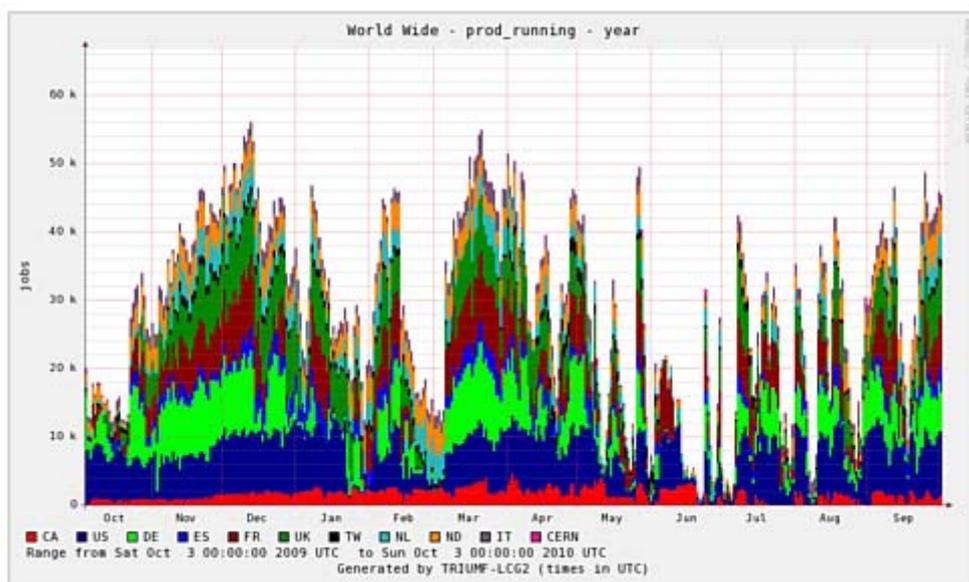


Figure 3. The number of production jobs concurrently running in 2009 - 2010

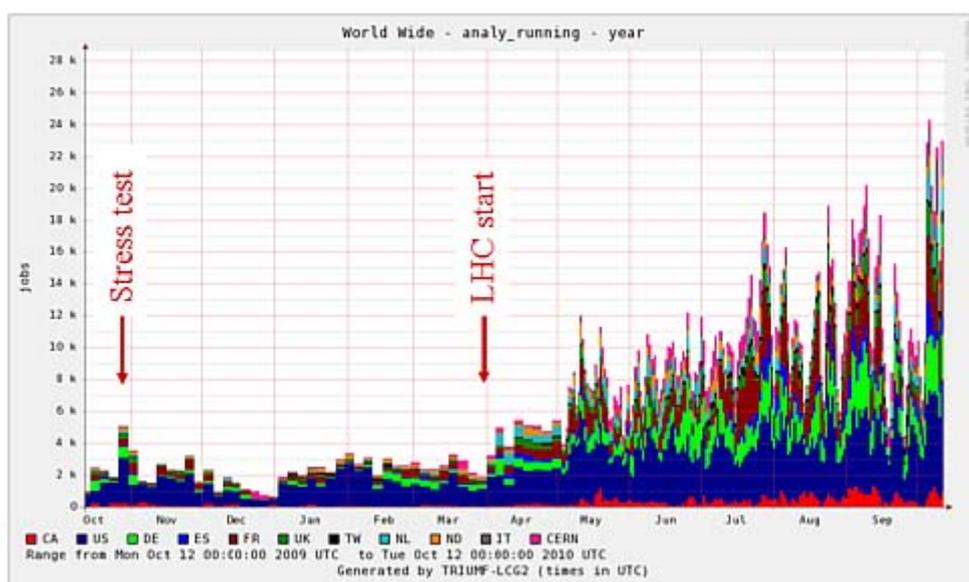


Figure 4. The number of analysis jobs concurrently running in 2009 - 2010

As described above, the PanDA system is in good shape, but development is not complete and there are still many improvements to come. However, the changes should be incremental and feedback-driven to avoid disruptions to ongoing activities. First, more flexibility should be added for group production. Second, PD2P should be enhanced for Tier1 to Tier1 data distribution. Third, a combination of remote IO and local caching mechanism should be evaluated for analysis jobs. Finally, output file merging should be implemented to solve an outstanding problem with transfers of small files produced by analysis jobs.

5. Conclusions

The PanDA system has been developed for ATLAS and is being extended as a generic high level workload manager usable by the wider grid community. PanDA is performing very well both for production and analysis for ATLAS, producing high volume Monte-Carlo samples and making huge computing resources available for individual analysis. There are still many new challenges to come but no big-bang migration is expected and the changes should be incremental and transparent to users.

Notice:

This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-AC02-98CH10886 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

- [1] Garonne V., *Status, news and update of the ATLAS Distributed Data Management software project: DQ2*, in Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)
- [2] Potekhin M., *The ATLAS PanDA Monitoring System and its Evolution*, in Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)
- [3] Caballero J., *Improving Security in the ATLAS PanDA System*, in Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)
- [4] Nilsson P., *The ATLAS PanDA Pilot in Operation*, in Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)
- [5] Filipcic A., *arcControlTower, the sytem for Atlas production and analysis on ARC*, in Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)
- [6] Klimentov A, *ATLAS Data Transfer Request Package (DaTRI)*, in Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)