Evolution of the ATLAS PanDA Production and Distributed Analysis System

T. Maeno [1], K. De [2], T. Wenaus [1], P. Nilsson [2], R. Walker [3], A. Stradling [2], V. Fine [1], M. Potekhin [1], S. Panitkin [1], G. Compostella [4], for the ATLAS Collaboration

[1] Brookhaven National Laboratory, NY, USA
[2] University of Texas at Arlington, TX, USA
[3] Ludwig-Maximilians-Universitat Munchen, Munchen, Germany
[4] Max-Planck-Institut für Physik, Munchen, Germany

tmaeno@bnl.gov

Abstract. The PanDA (Production and Distributed Analysis) 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. PanDA has performed well with high reliability and robustness during the two years of LHC data-taking, while being actively evolved to meet the rapidly changing requirements for analysis use cases. We will present an overview of system evolution including automatic rebrokerage and reattempt for analysis jobs, adaptation for the CernVM File System, support for the multi-cloud model through which Tier-2 sites act as members of multiple clouds, pledged resource management and preferential brokerage, and monitoring improvements. We will also describe results from the analysis of two years of PanDA usage statistics, current issues, and plans for the future.

1. Introduction

The PanDA (Production and Distributed Analysis) system plays a key role in the ATLAS distributed computing infrastructure. PanDA is the ATLAS workload management system for processing all Monte-Carlo (MC) simulation and data reprocessing jobs in addition to user and group analysis jobs. ATLAS production and analysis place challenging requirements on throughput, scalability, robustness, minimal operational manpower, and efficient integrated data/processing management. The system processes more than 5 million jobs in total per week, and more than 1400 users have submitted analysis jobs in 2011 through PanDA. PanDA has performed well with high reliability and robustness during the two years of LHC data-taking, while being actively evolved to meet the rapidly changing requirements for analysis use cases. We will present a brief overview of the PanDA system, an overview of system evolution, results from the analysis of two years of PanDA usage statistics, current issues, and plans for the future.

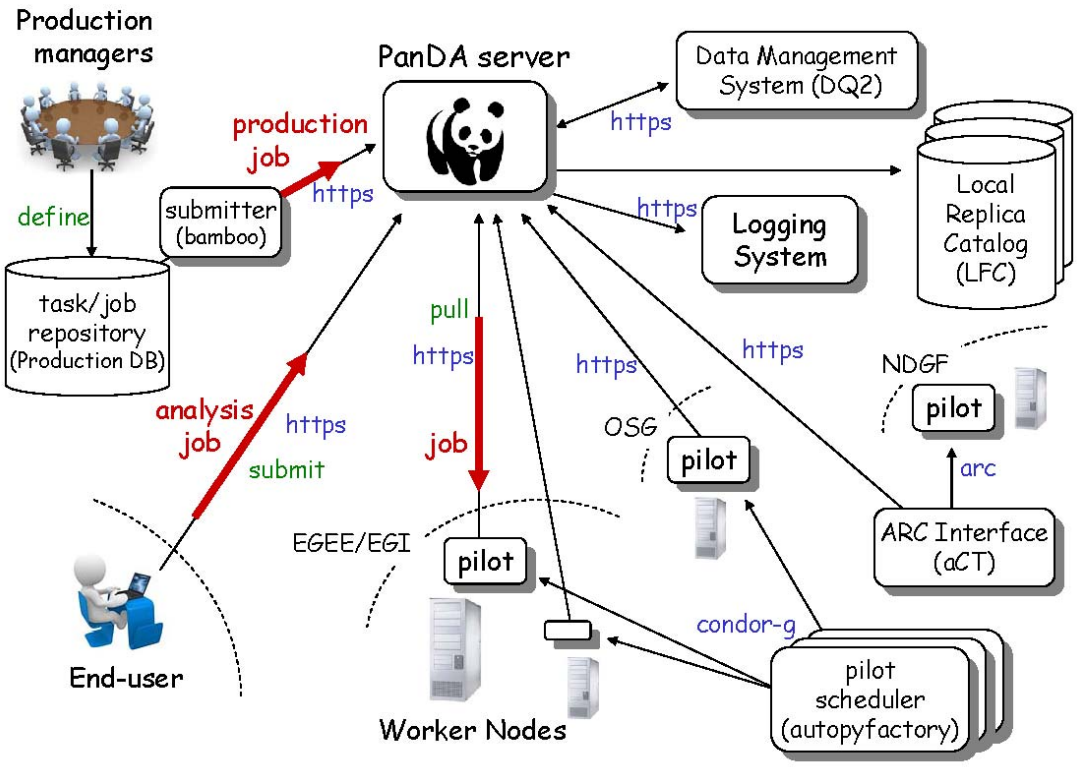

Figure 1. Schematic view of the PanDA System

2. The PanDA System

Figure 1 shows a schematic view of the PanDA system [1]. Jobs are submitted to the PanDA server. The PanDA server is the main component which provides a task queue managing all job information centrally. The PanDA server receives jobs into the task queue, upon which a brokerage module operates to prioritize and assign work on the basis of job type, priority, software availability, input data and its locality, and available CPU resources. The autopyfactory pre-schedules pilots to OSG and EGEE/EGI grid sites using Condor-G [2]. 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 on a worker node (WN), detects zombie processes, reports job status to the PanDA server, and recovers failed jobs. Ref.[3] describes the details on pilots. For NDGF, the ARC control tower [4] retrieves jobs from the PanDA server and sends the jobs together with pilot wrappers to NDGF sites using ARC middle-ware.

Workflow is different between production and analysis although PanDA is designed for production and analysis to use the same software, monitoring system [5], and facilities. For production, a limited number of people centrally define tasks on the basis of physics needs and resource allocation in ATLAS. ATLAS has one Tier-0 site, ten Tier-1 sites and about 70 Tier-2s. 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, 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. 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. Assignment of jobs to sites is followed by the dispatch of

input files to those sites. Atlas Distributed Data Management (DDM) system [6] takes care of actual data-transfer. Once jobs successfully finished on WNs, output files are aggregated back to the Tier-1 site from Tier-2 sites.

On the other hand, for analysis, each end-user submits a user task (job set) that is split to multiple job subsets according to localities of input datasets, workload distribution, 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 many jobs. One of the most significant differences between production and analysis 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.

3. An Overview of System Evolution

3.1. Rebrokerage for analysis jobs

Once a job set is submitted to PanDA, the brokerage assigns all jobs in each job subset to a single site and jobs wait in the queue until pilots pick them up to run on WNs. The brokerage selects the best site based on data locality, workload distribution, site downtime and status, and available CPUs at sites, exactly when the job set is submitted. However, the situation may change during the time jobs are waiting in the queue. E.g., the site may have unexpected downtime, new jobs with very high priorities may be submitted to the site so that waiting jobs have to wait longer than expected, or more replicas of input data may become available at free sites. In order to solve this issue, the rebrokerage mechanism has been implemented so that waiting jobs are periodically reassigned to other sites. In addition, the mechanism is triggered as soon as the site is blacklisted by HammerCloud [7] or Site Status Board [8], so that end-users are not very affected by incidents on the grid.

3.2. Automatic reattempts for analysis jobs

Analysis jobs fail due to various problems. When jobs fail, the pilot investigates causes of failures and reports corresponding error codes to the panda server. Each failed job is automatically retried at most three times at the same site if it fails due a recoverable problem, such as temporary glitches of network or storage. Furthermore, when jobs keep failing due to site-specific problems such as corrupted input files or wrong records in the local file catalogue, they are sent to another site by using the rebrokerage mechanism.

3.3. Adaptation for the CernVM File System (CVMFS)

Many ATLAS sites have installed CVMFS [9] and have been configured to read ATLAS software from CVMFS. The brokerage takes software availability into account when assigning jobs to sites. Although all software installed in the central CVMFS repository should be in principle available at those CVMFS sites, each site publishes software availability only when the software is validated at the site. The idea is to avoid failures due to missing system libraries or site-specific configuration problems such as a strict firewall setting and so on. How the pilot uses CVMFS is described in Ref.[10].

3.4. Support for the multi-cloud model

Historically ATLAS had used a strictly hierarchical cloud model for production. Each cloud was composed of one Tier-1 site and regional Tier-2 sites. The combination between Tier-1 and Tier-2 sites was rather static. Tier-2 sites were not moved to another cloud very frequently and each Tier-2 belonged to only one cloud. Data for production were transferred in each cloud. The 'monolithic' model simplified flow of data transfers, which was a great advantage from an operational point of

view. However, the entire system is increasingly matured and robust and thus the constraint has been relaxed, so that one Tier-2 site can be associated to multiple clouds. When a Tier-2 site X belongs to cloud A and cloud B the site executes jobs in both clouds. For example, when more jobs are assigned to cloud A than cloud B, the site X would execute more jobs for cloud A. The idea is to use CPU resources at Tier-2 sites efficiently even if job distribution is unbalanced or even if a Tier-1 site is down, so that entire system throughput is improved. Also, a Tier-1 site can be associated to another foreign cloud. In this case, the Tier-1 site acts as a Tier-2 member of the foreign cloud, i.e., input files are transferred from the foreign Tier-1 site and output files are transferred back to the foreign Tier-1 site.

3.5. Beyond-pledge resource management and preferential brokerage for analysis jobs

Some sites have regional CPU/storage resources by using budgets beyond ATLAS MoU share in addition to official resources. Each site can be configured to use additional resources only for the users who belong to a particular country group when their jobs are waiting in the queue, otherwise, use those resources for other general users. There are two special site configuration parameters; *pledgedCPU* and *availableCPU* stand for the amount of official and total CPU resources, respectively. Every time the pilot tries to get a job, the PanDA server calculates the ratio of the number of running jobs in the country group at the site to the total number of running jobs at the site, and selects a job so that the ratio is nearly equal to 1–(*pledgedCPU* / *availableCPU*). Also, when the users who belong to a country group submit jobs, the brokerage preferentially assigns them to sites that provide additional resources for the country users.

3.6. Fairshare for production activities

ATLAS production jobs are roughly categorized to three groups of activities; MC simulation, data reprocessing, and physics working group production. Generally jobs for data reprocessing and physics working group production have higher priorities than jobs for MC simulation because of urgent needs for the former activities. Sometimes MC simulation jobs had to wait until other higher priority jobs were completed when they were assigned to the same site. MC simulation jobs are still important and thus their production requests must be completed on time. The machinery of beyond-pledge resource management has been extended to solve this issue. ATLAS can set a fairshare policy at each site to define how CPU resources are allocated to production activities and/or physics working groups. Every time the pilot tries to get a job, the PanDA server calculates the ratio of numbers of running jobs for various production activities and/or physics working groups, and selects a job so that the ratio follows the policy at the site.

3.7. Outbound network connection monitoring on WN

Based on the ATLAS policy end-users can submit any kind of jobs to PanDA. Sometimes naïve users submitted jobs which invoked network-related commands, such as curl and lcg-cp, and unintentionally caused DOS attacks since a single job set is automatically split to multiple jobs to be executed on WNs in parallel. A connection monitoring mechanism has been implemented in order to detect and/or kill problematic jobs which establish redundant outbound network connections. Before jobs get started, a shared library is compiled and pre-loaded on the WN to trap *connect* and *execve* system calls. When a job tries to connect to a remote host, the hostname, port number and application name are written to a log file, and the job is killed if the connection is not appropriate. The log file is uploaded to the PanDA monitoring system. Currently this mechanism is used only for monitoring. The kill function is disabled for now and will be enabled once enough expertise is accumulated.

4. Current status and future plans

Figure 3 shows the number of production jobs running concurrently in 2011 and 2012. We can see the maximum number of running jobs is 100k which has doubled since 2010. The system has

continuously been running about 70k jobs. The single job failure rate of production jobs was about 8% in 2012 similarly as 2010. The breakdown is 2% from ATLAS software problems and 6% 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 2011 and 2012. We can see that analysis activities on PanDA have increased steadily, which clearly shows the usefulness of PanDA for physics analysis. The single job failure rate of analysis jobs has been improved to 13% in 2012, down from 21% in 2010, due to various improvements reported in this paper. The major causes of failures are software problems since end users often submit problematic jobs.

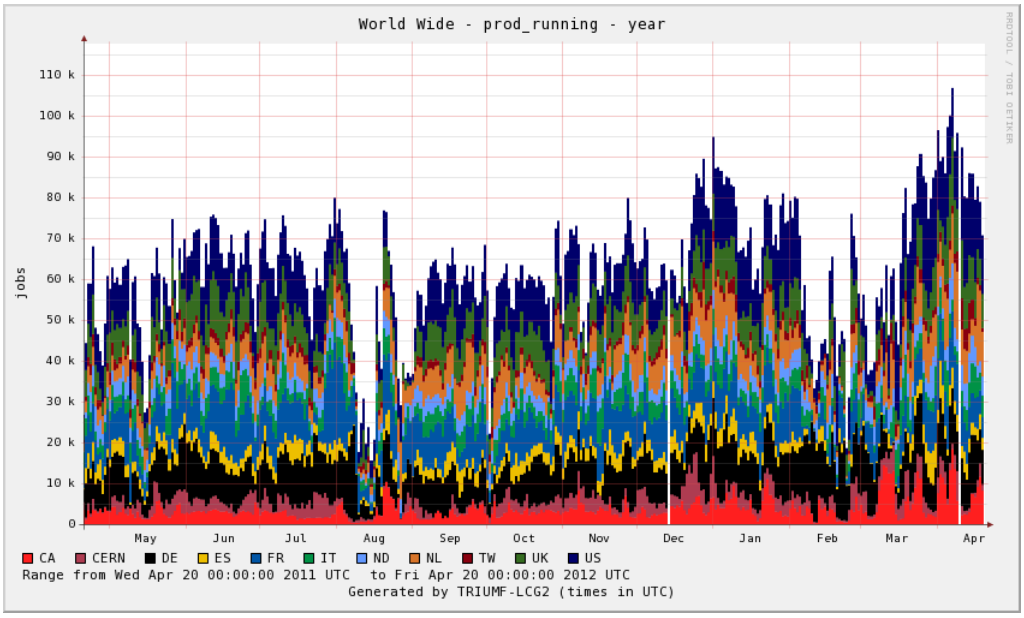

Figure 3. The number of production jobs concurrently running in 2011 – 2012

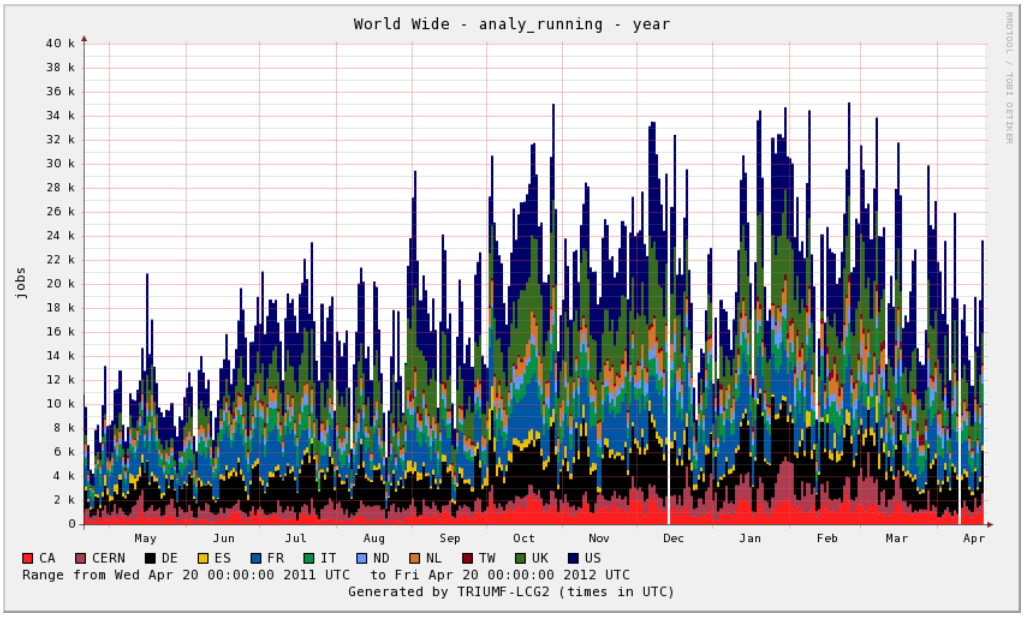

Figure 4. The number of analysis jobs concurrently running in 2011 - 2012

As described above, the PanDA system is in good shape, but development is not complete and there are still many improvements to come. First, new technologies should be adopted to make the PanDA system more robust and capable, such as highly scalable NoSQL databases, message queue services, and commercial/private cloud services. Second, the Job Execution and Definition Interface (JEDI), which dynamically generates jobs based on real-time requirements, should be developed to improve automation and efficiency. Third, multi-core queues and athenaMP [11] should be utilized. Fourth, functionalities for end-user job submission should be moved to the server side to simplify client tools. Finally, the PanDA system should be extended for the wider grid community.

5. Conclusions

The PanDA system has performed very well during the LHC data-taking year, producing high volume Monte-Carlo samples and making huge computing resources available for individual analysis, while being actively evolved to meet the rapidly changing requirements for analysis use cases. There are still many challenges to come in the ATLAS data processing, and in addition PanDA is being extended as a generic high level workload manager usable by the wider high throughput distributed processing community.

References

[1] Maeno T., *Overview of ATLAS PanDA Workload Management*, J. Phys. Conf. Ser. **331** (2011)
[2] Caballero J., *AutoPyFactory: A Scalable Flexible Pilot Factory Implementation*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[3] Nilsson P., *The ATLAS PanDA Pilot in Operation*, J. Phys. Conf. Ser. **331** (2011)
[4] Filipcic A., *arcControlTower, the sytem for Atlas production and analysis on ARC*, J. Phys. Conf. Ser. **331** (2011)
[5] Potekhin M., *The ATLAS PanDA Monitoring System and its Evolution*, J. Phys. Conf. Ser. **331** (2011)
[6] Garonne V., *The ATLAS Distributed Data Management project: Past and Future*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[7] Legger F., *Improving ATLAS grid site reliability with functional tests using HammerCloud*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[8] Iglesias C. B., *Automating ATLAS Computing Operations using the Site Status Board*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[9] Predrag B., *Status and Future Perspectives of CernVM-FS*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[10] Nilsson P., *Recent Improvements in the ATLAS PanDA Pilot*, in Proc. of the 19th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2012)
[11] Leggett C., *Parallelizing ATLAS Reconstruction and Simulation: Issues and Optimization Solutions for Scaling on Multi- and Many-CPU Platforms*, J. Phys. Conf. Ser. **331** (2011)