# PanDA: Distributed Production and Distributed Analysis System for ATLAS

**Tadashi Maeno on behalf of PANDA team and ATLAS collaboration**

Brookhaven National Laboratory, Upton, NY, 11973, USA

tmaeno@bnl.gov

**Abstract**. A new distributed software system was developed in the fall of 2005 for the ATLAS experiment at the LHC. This system, called PANDA, provides an integrated service architecture with late binding of jobs, maximal automation through layered services, tight binding with ATLAS Distributed Data Management system [1], advanced error discovery and recovery procedures, and other features. In this talk, we will describe the PANDA software system. Special emphasis will be placed on the evolution of PANDA based on one and half year of real experience in carrying out Computer System Commissioning data production [2] for ATLAS. The architecture of PANDA is well suited for the computing needs of the ATLAS experiment, which is expected to be one of the first HEP experiments to operate at the peta-byte scale.

## 1. Introduction

The PANDA production and distributed analysis system has been developed by US ATLAS to meet ATLAS requirements for petabyte scale production and distributed analysis processing. ATLAS processing and analysis places challenging requirements on throughput, scalability, robustness, minimal operations manpower, and efficient integrated data/processing management. Current estimates are for 100-200k jobs/day within US ATLAS during full scale running, and ~4 times that number ATLAS-wide. PANDA development began in August 2005 and it took over US ATLAS production responsibilities in December 2005. The key features are as follows:

- Designed from the beginning to support both managed production and individual analysis via flexible job specification/injection
- Tight integration with the ATLAS distributed data management (DDM) system working with datasets
- Use of grid and/or farm batch queues to pre-stage job wrappers (pilots) to worker nodes
- An integrated monitoring system for production and analysis operations, user analysis interfaces, data access, and site status
- Easy-to-use client interface makes integration with diverse analysis front ends simple
- Works both with OSG and EGEE middleware
- Highly automated system requiring low operation manpower

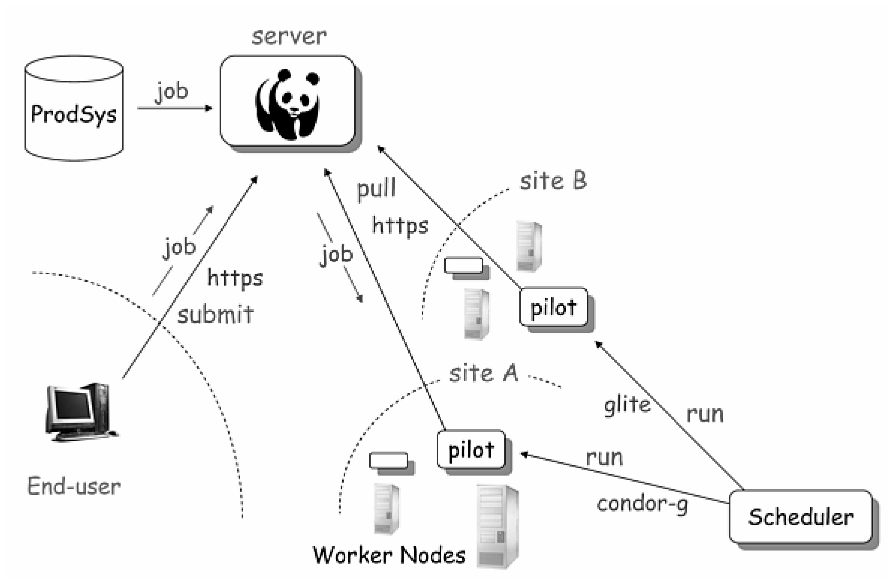


**Figure 1.** Schematic view of the PANDA System

PANDA was originally developed for ATLAS and is being extended as a generic high level workload manager usable by anyone in the OSG or wider grid community. CHARMM [3] is the first OSG user and has achieved great success with PANDA-based production.

This paper reports on the PANDA architecture, the current status of the project, and future plans.

## 2. The PANDA System

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 jobs. 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. Allocation of job blocks to sites is followed by the dispatch of input data to those sites, handled by a data service module interacting with ATLAS DDM. The implementation of the PANDA server is a LAMP stack (Red-Hat Enterprise Linux 3/4 or Scientific Linux CERN 4, Apache, MySQL and Python) with a multi-processing/multi-threading architecture running 50 child-processes concurrently. The Apache module mod_gridsite [4] is loaded to provide access control with X.509 certificates. The server-client interface is based on curl/urllib and Python pickle without any dependence on grid middleware. Clients are insulated from the grid by the HTTP layer and can submit jobs for any grid flavor with minimal knowledge of the grid, which is important especially for end-users like physicists.

The scheduler sends pilots to batch systems and grid sites. There are three kinds of scheduler: CondorG scheduler, Local scheduler, and Generic scheduler. The CondorG scheduler sends pilots to most US-ATLAS OSG sites with the CondorG infrastructure. The Local scheduler uses local-batch commands like qsub for PBS and condor_submit for Condor, which results in very high efficiency and robustness. The Generic scheduler has been developed to support also non-ATLAS OSG VOs and EGEE, and is being extended through OSG extensions project to support Condor-based pilot factory. The idea is to move pilot submission from a global submission point to a site-local pilot factory, which itself is globally managed as a Condor glide-in [5].

Pilots are pre-scheduled to batch system and grid sites. They retrieve jobs from the PANDA server to run the jobs as soon as CPU's becomes 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 process, reports job status to the PANDA server, and recovers failed jobs. Ref.[6] describes the details on pilots.

The PANDA system has a pipeline structure running data-transfer and job-execution in parallel. DDM takes care of actual data-transfers: the PANDA server sends requests to DDM to dispatch input files to Tier2's or to aggregate output files to Tier1's. DDM transfers files and then sends notifications to the PANDA server. Jobs wait in the task queue after they are submitted to PANDA, and get activated when all input files are transferred. Pilots pick activated jobs and can run immediately since DDM has already pre-staged the input files. Similarly, pilots copy output files to local storages and exit immediately to release CPU's. Then the PANDA server sends requests to DDM to move files over the grid. That is, pilots occupy CPU's only during execution of the ATLAS jobs and access to the local storage.

As mentioned, PANDA is designed for production and analysis. This means that both production and user analysis use the same software, monitoring system and facilities, and thus no duplicated manpower is required for maintenance. For instance, when the system has a problem, production will see the same problem as well as analysis. Once the operations team fixes the problem for the official production, analysis is automatically fixed. However, analysis and production run on separate CPU clusters to avoid competition with each other, and have different policies for data-transfers. DDM transfers files over the grid for production, while analysis does not trigger file-transfers since analysis jobs are sent to sites where the input files already exist. Therefore, the total amount of data traffic is under control.

## 3. Current Status and Future Plan

Computer System Commissioning (CSC) for ATLAS is ongoing. Massive Monte-Carlo samples have been produced for software validation, physics studies, calibration and commissioning. There are many hundreds of different physics processes fully simulated with Geant 4. More than 10k CPU's are available in this exercise. Thousands of bug reports have been filed, which is an important feedback to software developers and physics users. PANDA has significantly contributed to CSC; taking care of all managed US-ATLAS production with ~2500 CPU's and have produced ~35% of total CSC samples [7]. Note that PANDA is not only for US/OSG. Canadian EGEE sites have used PANDA for their CSC production since the middle of 2007: TRIUMF, Toronto, WestGrid, Victoria and so on. 25% of PANDA data is now produced in Canada. Figure 2 shows walltime of finished jobs per day since 2006 in PANDA. We can see a steady increase in this figure.

For distributed analysis, PANDA has been in general use since June 2006. It is popular with about hundred users and has been requested by ATLAS facilities outside the US which we're working to satisfy.

Panda development is continuing actively. But the changes are mostly incremental to provide steady operation for the coming data-taking period at the LHC. There are still many improvements to come. First, the Generic scheduler should be deployed on OSG and EGEE to support ATLAS production and analysis across these grids. The key things are Glide-ins, pilot factory and further Condor integration through OSG extensions project, collaborating with Condor and CMS. Second, a self-scheduled data-transfer service has been implemented in the PANDA system. The pilot transfers files as an agent and the PANDA server schedules transfer jobs. Transfer jobs are treated similarly as other jobs like production and analysis so that most of PANDA code is reusable with small modifications. The advantage is that PANDA can smoothly schedule transfer jobs and other normal jobs without external dependences. Finally, improvements to scale-up. CPU and disk resources available to ATLAS are rising steadily and production system efficiencies are steadily increasing. But much more will be required for data taking. Additional resources are coming online soon.

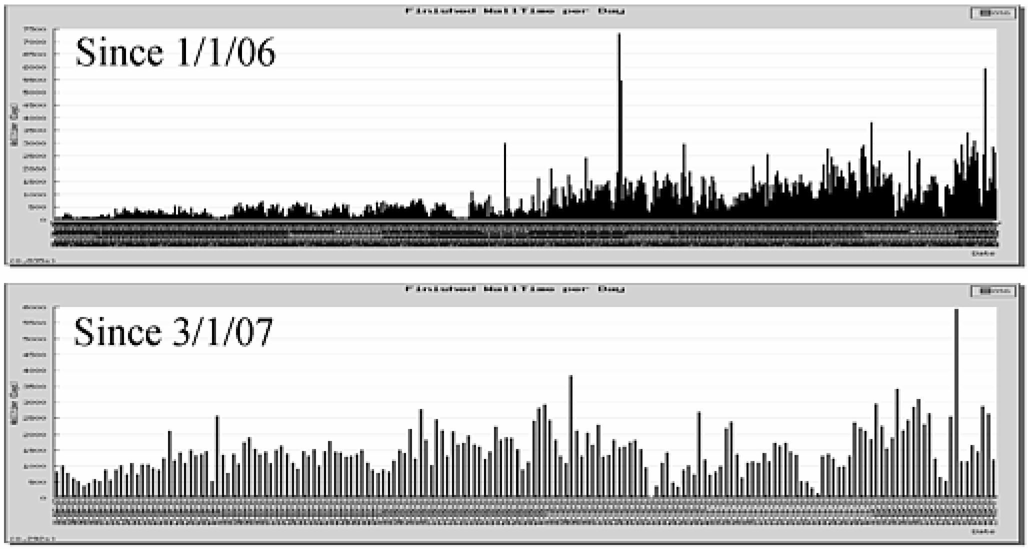


**Figure 2.** Walltime of finished jobs per day since the beginning of 2006 in PANDA

## 4. 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, e.g., broadening deployment and supporting EGEE sites, OSG Tier3's and so on. But no big-bang migration is expected and the changes should be incremental and transparent to users. We believe PANDA is ready to provide stable and robust service for coming data-taking of ATLAS experiment.

**References**


[1] Lassnig M., *Managing ATLAS data on a petabyte-scale with DQ2*, Journal of Physics: Conference Series

[2] Klimentov A., *ATLAS Distributed Data Management Operations. Experience and Projection*, Journal of Physics: Conference Series

[3] Maksimovic, P., *CHARMM on OSG*, OSG User's Meeting, Fermilab, Batavia, 26-27 July 2007, http://indico.fnal.gov/contributionDisplay.py?contribId=3&sessionId=1&confId=870

[4] McNab A., *The GridSite security architecture*, Journal of Physics: Conference Series

[5] Sfiligoi I., *glidinWMS – A generic pilot-based Workload Management Sytem*, Journal of Physics: Conference Series. http://www.cs.wisc.edu/condor/manual/v6.9/5_4Glidein.html

[6] Nilsson P., *Experience from a Pilot based system for ATLAS*, Journal of Physics: Conference Series

[7] Smirnov Y., *ATLAS Production Experience on OSG Infrastructure*, Journal of Physics: Conference Series



**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.