PanDA Pilot Submission using Condor-G: Experience and Improvements

2011 J. Phys.: Conf. Ser. 331 072069

(http://iopscience.iop.org/1742-6596/331/7/072069)

Journal of Physics: Conference Series **331** (2011) 072069 doi:10.1088/1742-6596/331/7/072069

PanDA Pilot Submission using Condor-G: Experience and Improvements

Xin Zhao, John Hover, Tomasz Wlodek, Torre Wenaus

Brookhaven National Laboratory

Jaime Frey, Todd Tannenbaum, Miron Livny

University of Wisconsin-Madison

(For ATLAS Collaboration)

xzhao@bnl.gov, jhover@bnl.gov

Abstract. PanDA (Production and Distributed Analysis) is the workload management system of the ATLAS experiment, used to run managed production and user analysis jobs on the grid. As a late-binding, pilot-based system, the maintenance of a smooth and steady stream of pilot jobs to all grid sites is critical for PanDA operation. The ATLAS Computing Facility (ACF) at BNL, as the ATLAS Tier1 center in the US, operates the pilot submission systems for the US. This is done using the PanDA "AutoPilot" scheduler component which submits pilot jobs via Condor-G, a grid job scheduling system developed at the University of Wisconsin-Madison. In this paper, we discuss the operation and performance of the Condor-G pilot submission at BNL, with emphasis on the challenges and issues encountered in the real grid production environment. With the close collaboration of Condor and PanDA teams, the scalability and stability of the overall system has been greatly improved over the last year. We review improvements made to Condor-G resulting from this collaboration, including isolation of site-based issues by running a separate Gridmanager for each remote site, introduction of the 'Nonessential' job attribute to allow Condor to optimize its behavior for the specific character of pilot jobs, better understanding and handling of the Gridmonitor process, as well as better scheduling in the PanDA pilot scheduler component. We will also cover the monitoring of the health of the system.

1. PanDA and Condor-G pilot submission

1.1 PanDA and pilot jobs

PanDA(Production and Distributed Analysis)[1] is the workload management system of the ATLAS experiment [2], used to run production and user analysis jobs on the grid.

Figure 1 shows how the job payload is assigned and processed by pilots in PanDA. Users submit their real jobs to Production DB, the PanDA server dispatches these real jobs to the next available pilots from a selected grid site. A separate Panda pilot scheduler submits pilots to all grid sites within a certain cloud constantly, harvesting the available computing resources and providing a stable pilot pool for PanDA to run user jobs. As a

late-binding, pilot-based system, to maintain a stable and scalable pilot submission system is critical for PanDA operation.

1.2 ACF and Condor-G pilot submission system

The ATLAS Computing Facility (ACF) at Brookhaven National Laboratory (BNL) is the ATLAS Tier1 center in the US. One of the critical services performed at ACF is the pilot submission system for the US cloud. In PanDA, the pilot submission system is called "AutoPilot", which submits pilots to grid sites via Condor-G.

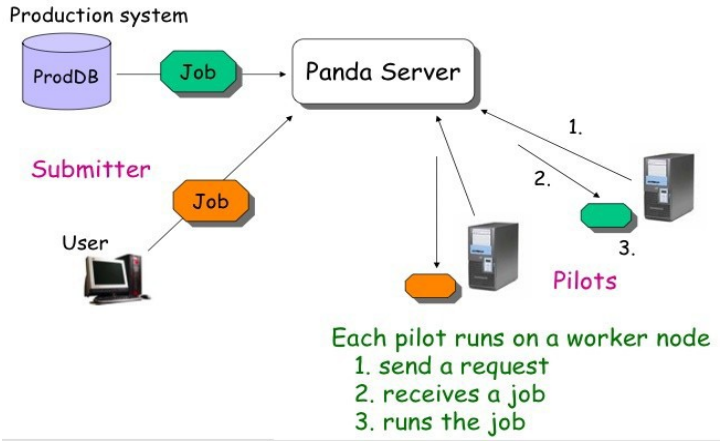

Figure 1: PanDA pilot submission and job processing

Condor-G is a generic grid job scheduling system, developed by the Condor team at the University of Wisconsin-Madison. Figure 2 shows the workflow of Condor-G job submission. Condor-G combines the resource management protocols of the Globus Toolkit and the resource and job management methods of the Condor batch system, to allow users to easily harness computing resources on the grid. For more information, refer to reference [3,4].

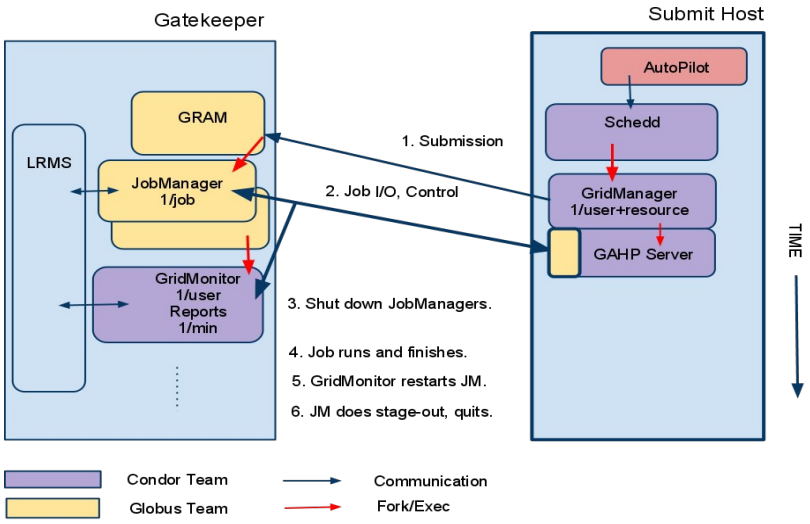

Figure 2: workflow of Condor-G job submission

2. Improving scalability and stability of Condor-G and AutoPilot

During the early days of operating the pilot submission at ACF, we quite often ran into problems with scalability and stability. It was hard to sustain the pilot submission rate at a sufficiently high level, causing starvation of pilots on the grid. In close collaboration with PanDA developers and Condor-G team, a lot of improvements and enhancements to Condor-G and AutoPilot have been implemented. Below we will go through the major improvements and enhancements:

2.1. Isolation of site specific issues from affecting submission to other sites

Condor-G can be configured to start a separate GridManager instance per remote Computing Element (CE) per user. This behavior is specified in the condor configuration file by using the "GRIDMANAGER_SELECTION_EXPR" setting. Compared to the previous version that starts a GridManager instance per user account, this feature provides better isolation of sites issues from the overall submission.

2.2. Introduction of the "Nonessential" job attribute

Generally condor treats every job as unique and important. It will hold and retry a job in case of failure. The "Nonessential" attribute allows condor to unconditionally remove and clean up problematic jobs, instead of keeping them on hold forever, and move on to new job submission. This applies perfectly to pilot-based environments, where a steady stream of identical pilots is more important than the pilots themselves, which are just job sandboxes, not as valuable as real job payloads.

2.3. Role-based throttle on limiting jobmanager processes on remote CE

Condor-G has one throttle for the total number of jobmanager processes invoked on the remote CE, to avoid overloading the remote gatekeeper. The jobmanagers are responsible for staging job inputs and outputs. A surge in job stage-out will stall job stage-in, and vice versa. Now this throttle limit is split in half based on their functionality, half for job submission (stage-in), and half for job completion and cleanup (stage-out).

2.4. Adjustable Grid monitor restart behavior

By making the grid monitor restart time limit configurable (via GRID_MONITOR_DISABLE_TIME setting), and refining gridmanager error handling, it is now possible to fine tune the grid monitor restart behavior. This reduces the possibility of slow job status update, improving the overall job throughput.

2.5. Better scheduling in AutoPilot to avoid overloading remote CE

Autopilot will adjust the pilot submission rate based on the number of real jobs destined for processing on each remote CE. Combined with multi-job pilots, these measures greatly help to reduce the load on the remote gatekeeper, so the real job processing rate can be sustained at a high level. When few jobs are destined for a site, the pilot submission rate is reduced.

2.6. Improving grid job status monitoring

The GridManager now publishes resource classads to a condor collector, from which users may query for grid job submission status.

3. PanDA AutoPilot Operation at BNL ACF

BNL ACF is responsible for running the PanDA AutoPilot submission, to feed pilots to all USATLAS grid sites. Below we will cover several aspects of this operation.

3.1. Pilot submission in the US cloud

The US cloud has 92 PanDA queues defined, served by 43 grid gatekeepers, with ~110K HEPSPEC-06 CPU capacity. The full scale production has >10K real jobs running on all the USATLAS resources, with daily peak reached at about 480K pilots (see figure 3).

The US cloud are served by 5 Condor-G submission hosts at the BNL ATLAS Tier1 center (ACF), with 3 of them being primary and 2 being warm backups.

While maintaining the production AutoPilot submit hosts, we also run stress tests with the Condor team, to continuously explore the system limits and improve the performance. So far we have successfully tested the system with 50K jobs managed by one submit host, and 30K jobs submitted to one remote CE (GT2 gatekeeper) at one time. These values suggest remaining capacity in terms of scalability, allowing us to extend the service in the future if needed.

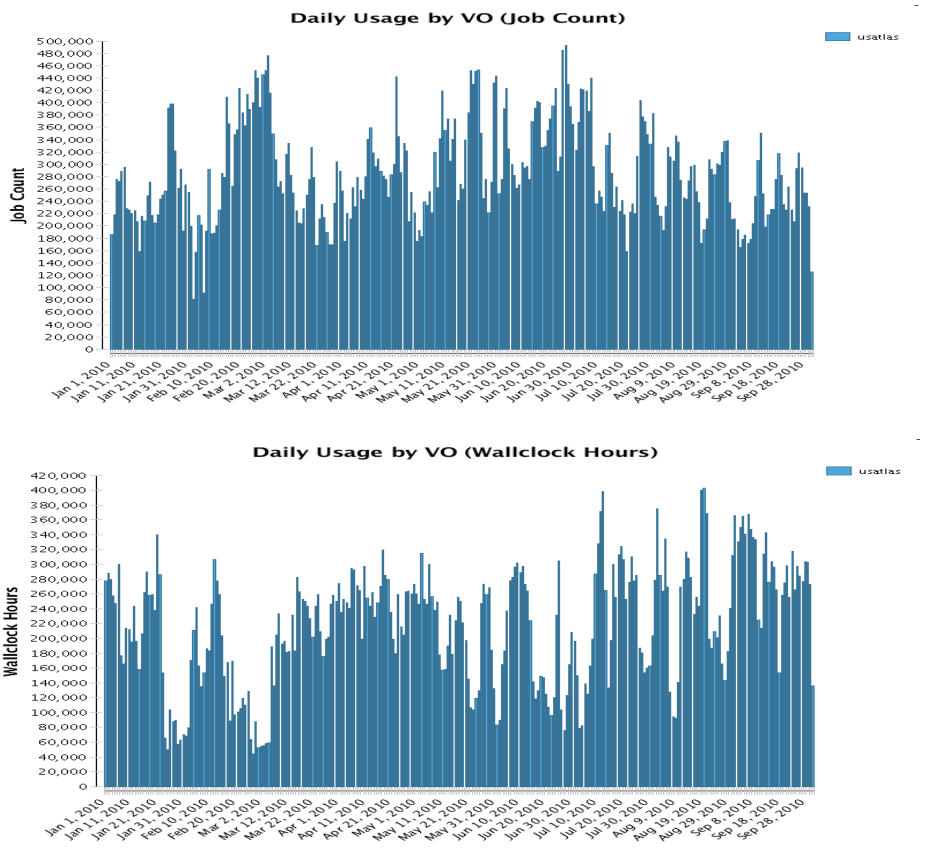

Figure 3 USATLAS Job accounting data from OpenScienceGrid (OSG)[5]

3.2. Some best practices in Condor-G submission

Operationally there are some best practices that we use to improve the stability and scalability of the service:

- Optimize PanDA queues setup among the multiple submit hosts.
  Condor-G doesn't distinguish between pilots serving different PanDA queues, which are known only to PanDA internally. By rearranging the queues among different submit hosts, we can avoid starvation of pilots for any queues.
- Multiple AutoPilot instances to increase injection rate for large-capacity PanDA queues.
- Caching Condor job status information, to reduce load on Condor Schedd on the submit hosts.
- Avoid hard-killing jobs from the submit hosts, to reduce the debris left on the remote CEs.

3.3. Monitoring of pilot submission

The pilot submission service is maintained on a 24x7 basis. We have implemented our monitoring infrastructure using a custom Dashboard and Nagios. Figure 4 shows the status of one PanDA queue (BNL_ATLAS_1) from

the Dashboard[6]. Nagios alerts will be triggered whenever a problem is detected by a probe, allowing us to proactively monitor and recover the service in case of failure.

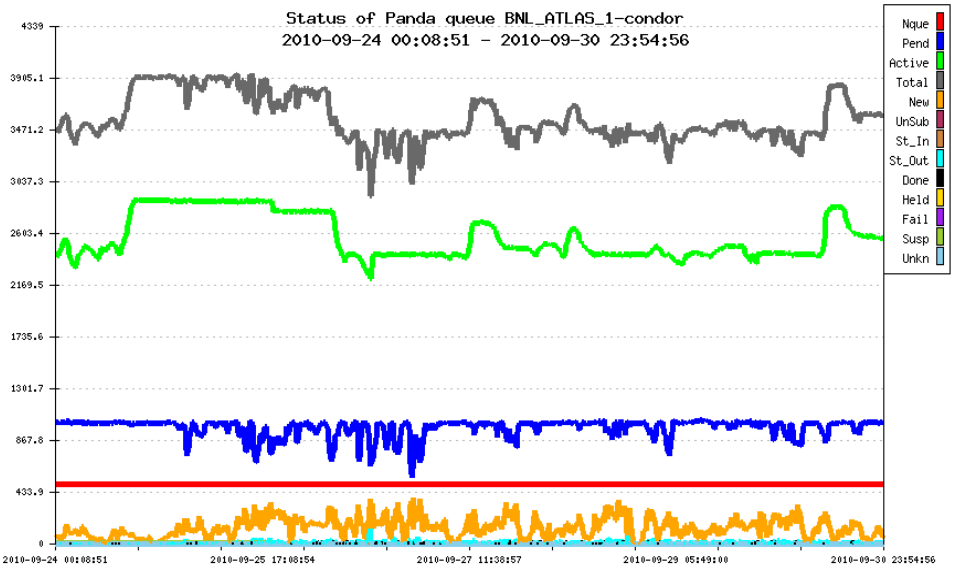

Figure 4: BNL_ATLAS_1 PanDA queue status from the monitoring Dashboard. It shows how the numbers of jobs in different states (Y-axis), change over time (X-axis). This is a weekly plot.

References

[1] PanDA twiki: https://twiki.cern.ch/twiki/bin/viewauth/Atlas/PanDA
[2] G. Aad, et al., ATLAS Collaboration, JINST 3 (2008) S08003
[3] Jaime Frey, Todd Tannenbaum, Ian Foster, Miron Livny, and Steven Tuecke, "Condor-G: A Computation Management Agent for Multi-Institutional Grids", Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC10), San Francisco, California, August 7-9, 2001.
[4] Condor Homepage: http://www.cs.wisc.edu/condor/
[5] OpenScienceGrid homepage: http://www.opensciencegrid.org/
[6] Dashboard URL: https://nagios.racf.bnl.gov/nagios/cgi-bin/prod/dashboard.php