PanDA for ATLAS distributed computing in the next decade

F H Barreiro Megino [1], K De [1], A Klimentov [2], T Maeno [2], P Nilsson [2], D Oleynik [1], S Padolski [2], S Panitkin [2], T Wenaus [2], on behalf of the ATLAS Collaboration

[1] University of Texas at Arlington, TX, USA
[2] Brookhaven National Laboratory, NY, USA

tmaeno@bnl.gov

Abstract. 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 the Large Hadron Collider (LHC) data processing scale. Heterogeneous resources used by the ATLAS experiment are distributed worldwide at hundreds of sites, thousands of physicists analyse the data remotely, the volume of processed data is beyond the exabyte scale, dozens of scientific applications are supported, while data processing requires more than a few billion hours of computing usage per year. PanDA performed very well over the last decade including the LHC Run 1 data taking period. However, it was decided to upgrade the whole system concurrently with the LHC's first long shutdown in order to cope with rapidly changing computing infrastructure. After two years of reengineering efforts, PanDA has embedded capabilities for fully dynamic and flexible workload management. The static batch job paradigm was discarded in favor of a more automated and scalable model. Workloads are dynamically tailored for optimal usage of resources, with the brokerage taking network traffic and forecasts into account. Computing resources are partitioned based on dynamic knowledge of their status and characteristics. The pilot has been re-factored around a plugin structure for easier development and deployment. Bookkeeping is handled with both coarse and fine granularities for efficient utilization of pledged or opportunistic resources. An in-house security mechanism authenticates the pilot and data management services in off-grid environments such as volunteer computing and private local clusters. The PanDA monitor has been extensively optimized for performance and extended with analytics to provide aggregated summaries of the system as well as drill-down to operational details. There are as well many other challenges planned or recently implemented, and adoption by non-LHC experiments such as bioinformatics groups successfully running Paleomix (microbial genome and metagenomes) payload on supercomputers. In this paper we will focus on the new and planned features that are most important to the next decade of distributed computing workload management.

1. Introduction

The Production and Distributed Analysis (PanDA) system [1] has been developed to meet ATLAS [2] production and analysis requirements for a data-driven workload management system capable of operating at LHC [3] data processing scale. PanDA scalability has been demonstrated in ATLAS through the rapid increase in usage over the last decade. PanDA was designed to have the flexibility to adapt to emerging computing technologies in processing, storage, networking and distributed computing middleware. The flexibility has been successfully demonstrated through the past years of evolving technologies adopted by computing centers in ATLAS which span many continents. PanDA performed very well including the LHC data taking period. The system had been producing high volume of Monte Carlo samples and making large-scale diverse computing resources available for individual analysis. However, to cope with rapidly changing computing infrastructure, new components and features were delivered to ATLAS in the LHC's first long shutdown, such as the Database Engine for Tasks (DEFT) [4], the Job Execution and Definition Interface (JEDI) [1], dynamic workload partitioning, Event Service [5,6], and new monitoring [7]. The major system upgrade was successful and the system has revealed great improvements in LHC Run 2. There are typically 250,000 jobs concurrently running in the system and more than 5 million jobs are processed in total per week.

In spite of great successes there are strong motivations for new developments. First, there is inefficiency in the PanDA system due to old resource partitioning based on geographical and/or national grouping of computing centers. Second, usage of non-traditional resources is suboptimal due to job-based workload management. Third, various High Performance Computing (HPC) workflows have been incoherently implemented. Fourth, the architecture of the pilot has been overextended to support non-traditional resources. Fifth, it is beneficial to leverage prediction capabilities for resource availability actively developed with recent computing technologies like machine learning. Finally, there are operational difficulties with new workflows due to job-centric monitoring. We will present in this paper a brief overview of the major aspects of PanDA systems evolution, as well as plans for the future.

2. Overview of system evolution

2.1. Resource consolidation

In the ATLAS computing model, computing resources were partitioned based on the old, hierarchical MONARC model, where each partition was composed of one Tier 1 computing center and multiple Tier 2 computing centers. Combinations between Tier 1 and Tier 2 centers were statically defined based on national and/or geographical groupings. Data traffic for production was limited to a partition, i.e., input data was transferred from the Tier 1 center to its Tier 2 centers while output data was aggregated to the Tier 1 center from its Tier 2 centers. The workload brokerage assigned a set of jobs (task) to one of the partitions to process within the partition. There were a couple of problems in this model. First, the brokerage algorithm was complicated since each task varied in size due to various physics needs and each partition had a different computing capability and workload occupancy. For example, generally the brokerage avoided to assign large high priority tasks to small partitions since those tasks could not finish quickly. However, the brokerage made different decisions when large partitions were busy and it was expected for small partitions to process tasks quickly. Second, network usage was suboptimal since the model didn't take network information into account. For example, some foreign Tier 2 centers had better network connections to a Tier 1 center than some domestic Tier 2 centres. Finally, Tier 2 storage was only used for secondary data replicas and was therefore not optimally exploited, while Tier 1 storage was quite full.

To address those issues a new flat model has been introduced, where all computing centers belong to a single partition without any hierarchical structuring. There are two new concepts; 'nucleus' is the destination of output data while a 'satellite' processes jobs to produce output data. A sub-partition is dynamically formed for each task with one nucleus and multiple satellites based on static configuration and dynamic information on network quality between nuclei and satellites. Reliable Tier 2 centers can be nuclei in addition to Tier 1 centers, so that usage of Tier 2 storage is improved to host input and output data for production. The details of resource consolidation are described in Ref [8].

2.2. Intelligent brokerage

The brokerage has been improved to have more intelligence based on retry history of jobs, forecast of network performance, and cache hit rate of input data. If previous job attempts have permanently failed at some computing resources, the brokerage avoids them for the next attempt. Network forecast is provided by the Network Weather Service [9] so that the brokerage takes into account the expected transfer time for input and output data. The brokerage assigns more jobs to computing resources if cache replicas of input data are available at those resources in order to reduce data trafics for input data.

A new mechanism is being added for workload provisioning. Currently workload is passively assigned, i.e., jobs are assigned to computing resources once those resources become active, which is good for traditional grid resources since a steady number of CPUs is available except in unusual situations such as site downtime. However, latency is too high for non-traditional resources like the ATLAS HLT farm since the number of available CPUs tends to ramp up and down immediately. The new mechanism allows workloads to be proactively assigned to computing resources, i.e., jobs are assigned just before the resources become available and they are removed just after the resources become unavailable, based on (quasi-) real-time resource information. The mechanism should be smart enough to minimize redundant data traffic, since workload assignment could trigger input data transfers.

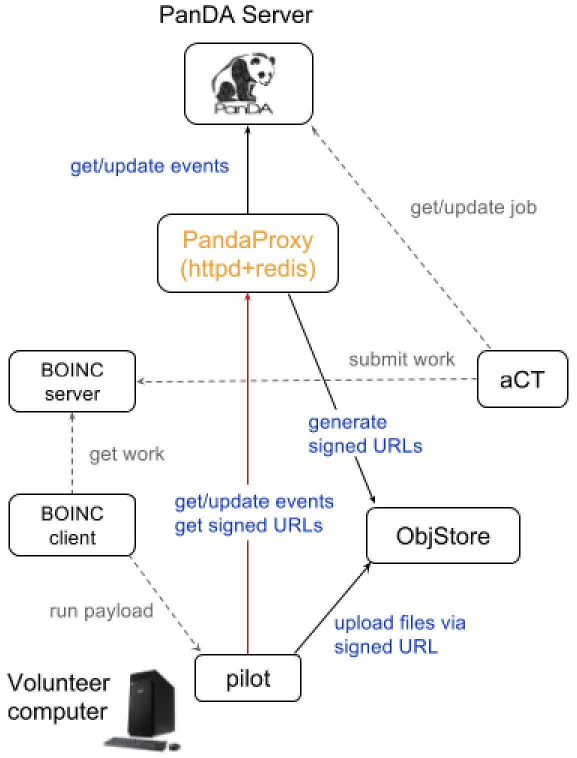

Figure 1. Work diagram of in-house security mechanism.

2.3. In-house security

A new security mechanism “PandaProxy” has been implemented to authenticate requests from the pilot running in special environments, where standard X509-based authentication is unavailable and/or suboptimal. Figure 1 shows how the mechanism works. An internal secret token is generated by the PanDA server for each job or pilot scheduler. The secret tokens are stored in a Redis data store and are propagated to the pilot via job specification or VM contextualization. The pilot accesses the PanDA server or ObjectStore data stores through PandaProxy with secret tokens. PandaProxy checks secret tokens and forward pilot requests once they are verified. There are a couple of promising use-cases, such as volunteer computing where it is not desirable to distribute grid credentials and access keys to outsiders, off-grid computing clusters like Tier 3 centers which are not fully integrated to the grid, and commercial cloud services.

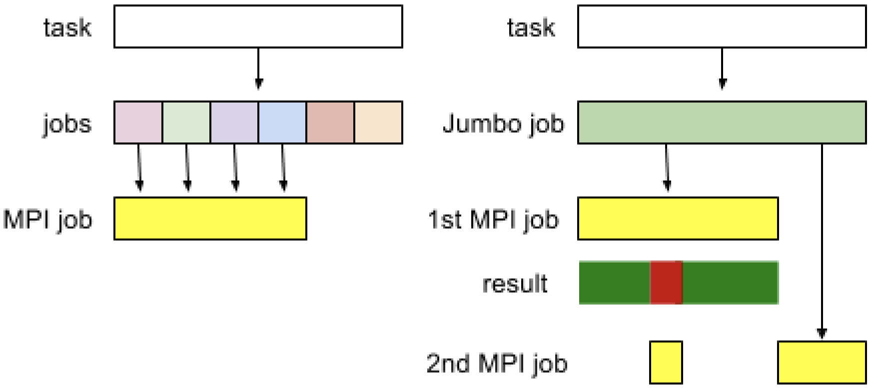

Figure 2. The left figure shows event processing with combined jobs and the right figure shows event processing with jumbo jobs.

2.4. Enhancement of Event Service

The Event Service has been developed to perform processing at fine granularity, down to the event level. It allows jobs to be revoked in the middle of processing with minimized losses. The old implementation of the Event Service assumed a modest number of events per job, typically ~1k events per job, which was good for preemptable resources since jobs could dynamically be fragmented. However, it was not good for large HPC resources since they prefer a large number of events in one go for better scheduling in HPC batch systems. One solution was to combine jobs to a single MPI job, which worked to some extent but required complicated workload management and bookkeeping.

The new “jumbo jobs” feature has been implemented to address the issue. Figure 2 shows how workload is processed with jumbo jobs. One or more jumbo jobs are generated from a single task. They allow workloads to be tailored to any size of MPI jobs. The details on current status and future plans of the Event Service are described in Ref.[10].

2.5. PanDA at HPC centers

PanDA has been well integrated with various HPC resources. Steady operation for continuous PanDA job submission in backfill mode has been successfully demonstrated with Titan machines at the Oak Ridge Leadership Computing Facility [11], as described in Ref.[12]. 10k jobs ran per day on average while 18k running jobs at peak. Multiple jobs are combined to a single MPI job to be submitted to the HPC batch scheduler. Work on integration of Edison and Cori with PanDA is ongoing at the National Energy Research Scientific Computing Center [13]. Jobs are processed with the Event Service to allow dynamic fragmentation and preemption. This working model is actively pursued since there are commonalities for coming Theta and Aurora at the Argonne Leadership Computing Facility [14]. Event Service jobs are sent through the ARC computing element to SuperMUC at the Leibniz Supercomputing Center [15] in order to cope with problems with frequent preemption and short execution time. PanDA has been adopted by non-LHC experiments such as Next Generation Genome Sequencing successfully running Paleomix (microbial genome and metagenomes) payload on HPC supercomputers at National Research Center - Kurchatov Institute [16].

2.6. Monitoring evolution

High quality user interfaces are essential for supporting effective use and comprehensive optimization and diagnostics of the system. Data visualization has evolved in various forms such as plots, histograms, and task chain diagrams. Predictive analytics have been added for the expected task completion time and comparison with actual progress of task processing. Redis cache and page preloading are intensively used to improve user experiences. The monitoring system has been interconnected with external systems like Kibana, AGIS, Rucio and CERN's Dashboard service. Currently, there are two major development activities. The first development is to exploit data aggregation on the Oracle backend. Data aggregation with an advanced data layout strategy has been evaluated, and successfully demonstrated capability for more flexible search queries with a significant reduction of the query processing time. Page build time was reduced from ~1 minute (with a 30k limit on job records in the last 12 hours only) to ~10 seconds (for ~800k job records without the 12 hour limit). The second development introduces an information access control layer with an authentication mechanism supporting Single Sign-On, Virtual Organization Membership Service, and Interoperable Global Trust Federation. Authentication enables command execution directly from the monitor and allows user customized contents. The details of monitoring evolution are described in Ref.[17].

2.7. Pilot 2.0

The PanDA pilot [18] is one of the major components in the PanDA system and has actively evolved throughout PanDA's history. The pilot continues to perform, but the architecture has been overextended to support new workflows and resources, which is leading to maintenance difficulties. A new long-term project was launched in April 2016 to almost completely rewrite all pilot components and incorporate some recent developments. The project involves developers within the core PanDA team as well as from external teams. Currently the project is in the design phase, but development activities are also ramping up. A mini-pilot system has been developed. It is a fully working pilot script for developers to test new components and could be evolved in the future into a simple pilot to provide new PanDA users with a rapid introduction. A git-based testing framework has been set up. Pull requests into the git repository trigger a verification sequence including unit tests. Various implementations with the component model are being evaluated with all workflows.

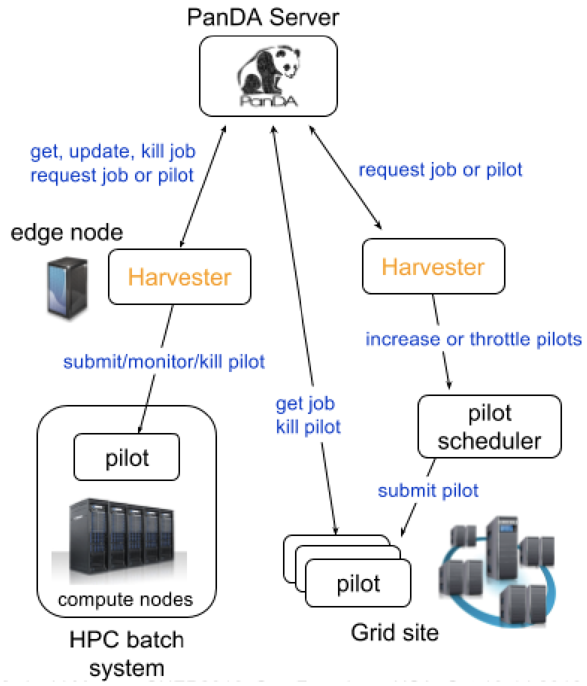

Figure 3. Interactions of harvester with the PanDA server and resource manages.

2.8. Harvester

Harvester is a new resource-facing service to propagate information and requests between the PanDA server and resource managers such as batch systems and pilot schedulers. Figure 3 shows interactions of Harvester with the PanDA server and resource managers. The main objectives are to add a capability for timely optimization of CPU allocation among various resource types, to provide a commonality layer bringing coherence to HPC implementations, and to have better integration between PanDA and resources for new workflows. Development is actively underway to deliver a first prototype in late 2016, with a wider collaboration engaged in writing plugins for various resource types.

3. **Future plans**

New developments and challenges are still coming. PanDA should be further automated using resource availability prediction and the expected completion time for each task. PanDA should proactively control the network to optimize workflows and dataflows. New computing resources will be brought into production efficiently and economically through the new PanDA components and features.

4. **Conclusions**

The PanDA system has performed very well for ATLAS in the last decade including the LHC Run 1 and Run 2 data-taking periods, steady state high volume Monte Carlo production, and individual analysis for the full ATLAS community, all making use of large-scale heterogeneous computing resources. New components and features were delivered to ATLAS in the LHC's first long shutdown and the system demonstrated great improvements in LHC Run 2. Nonetheless many developments and challenges are still ongoing and to come while the system steadily delivers the distributed production and analysis capability required by ATLAS during Run 2.

Notice:

This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a nonexclusive, 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] De K et al. 2015 The future of PanDA in ATLAS distributed computing *J. Phys. Conf. Ser.* **664** 062035

[2] ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider *J. Inst.* **3** S08003

[3] Evans L and Bryant P (editors) 2008 LHC Machine, *J. Inst.* **3** S08001

[4] Borodin M et al. 2015 Scaling up ATLAS production system for the LHC Run 2 and beyond: project ProdSys2 *J. Phys. Conf. Ser.* **664** 062005

[5] Wenaus T et al. 2015 The ATLAS Event Service: A new approach to event processing *J. Phys. Conf. Ser.* **664** 062065

[6] Tsulaia V et al. 2015 Fine grained event processing on HPCs with the ATLAS Yoda system *J. Phys. Conf. Ser.* **664** 092025

[7] PanDA monitoring http://bigpanda.cern.ch/

[8] Barreiro Megino F H et al. 2017 ATLAS WORLD-cloud and networking in PanDA. Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016 *J. Phys. Conf. Ser.*

[9] Lassnig M et al. 2017 Using machine learning algorithms to forecast network and system load metrics for ATLAS distributed computing. Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016 *J. Phys. Conf. Ser.*

[10] Tsulaia V et al. 2017 Production experience with the ATLAS Event Service. Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016 *J. Phys. Conf. Ser.*

[11] OLCF Titan https://www.olcf.ornl.gov/titan/

[12] Panitkin S et al. 2017 Integration of Titan supercomputer at OLCF with ATLAS production system. Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016 *J. Phys. Conf. Ser.*

[13] NERSC Edison and Cori https://www.nersc.gov/

[14] ALCF Theta and Aurora http://www.alcf.anl.gov/

[15] LRZ SuperMUC https://www.lrz.de/services/compute/supermuc/

[16] Mashinistov R et al. 2017 The ATLAS Event Service: Using HEP Computing Tools, Grid and Supercomputers for Genome Sequencing Studies. Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016 *J. Phys. Conf. Ser.*

[17] Borodin M et al. 2017 The ATLAS Production System Evolution. New Data Processing and Analysis Paradigm for the LHC Run2 and High-Luminosity. Proc. Int. Conf. on Computing in High Energy and Nuclear Physics 2016 *J. Phys. Conf. Ser.*

[18] Nilsson P 2014 Next Generation PanDA Pilot for ATLAS and Other Experiments *J. Phys. Conf. Ser.* **513** 032071