Integration of Titan supercomputer at OLCF with ATLAS Production System

F Barreiro Megino^1, K De^1, S Jha^2, A Klimentov^3, P Nilsson^3, D Oleynik^1, S Padolski^3, S Panitkin^3, J Wells^4 and T Wenaus^3 on behalf of the ATLAS Collaboration

^{1} University of Texas at Arlington, Arlington, TX 76019, USA
^{2} Rutgers University, New Brunswick, NJ 08901, USA
^{3} Brookhaven National Lab, Upton, NY 10573, USA
^{4} Oak Ridge National Lab, Oak Ridge, TN 37631 USA

Abstract. The PanDA (Production and Distributed Analysis) workload management system was developed to meet the scale and complexity of distributed computing for the ATLAS experiment. PanDA managed resources are distributed worldwide, on hundreds of computing sites, with thousands of physicists accessing hundreds of Petabytes of data and the rate of data processing already exceeds Exabyte per year. While PanDA currently uses more than 200,000 cores at well over 100 Grid sites, future LHC data taking runs will require more resources than Grid computing can possibly provide. Additional computing and storage resources are required. Therefore ATLAS is engaged in an ambitious program to expand the current computing model to include additional resources such as the opportunistic use of supercomputers. In this paper we will describe a project aimed at integration of ATLAS Production System with Titan supercomputer at Oak Ridge Leadership Computing Facility (OLCF). Current approach utilizes modified PanDA Pilot framework for job submission to Titan's batch queues and local data management, with lightweight MPI wrappers to run single node workloads in parallel on Titan's multi-core worker nodes. It provides for running of standard ATLAS production jobs on unused resources (backfill) on Titan. The system already allowed ATLAS to collect on Titan millions of core-hours per month, execute hundreds of thousands jobs, while simultaneously improving Titans utilization efficiency. We will discuss the details of the implementation, current experience with running the system, as well as future plans aimed at improvements in scalability and efficiency.

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.

1. Introduction

The ATLAS experiment [1] is one of the four major experiments at the Large Hadron Collider (LHC). It is designed to test predictions of Standard Model and explore fundamental building blocks of matter and their interactions as well as novel physics at the highest energy available in the laboratory. In order to achieve its scientific goals ATLAS employs massive computing infrastructure. It currently uses more than 250,000 CPU cores deployed in a global Grid [2, 3],

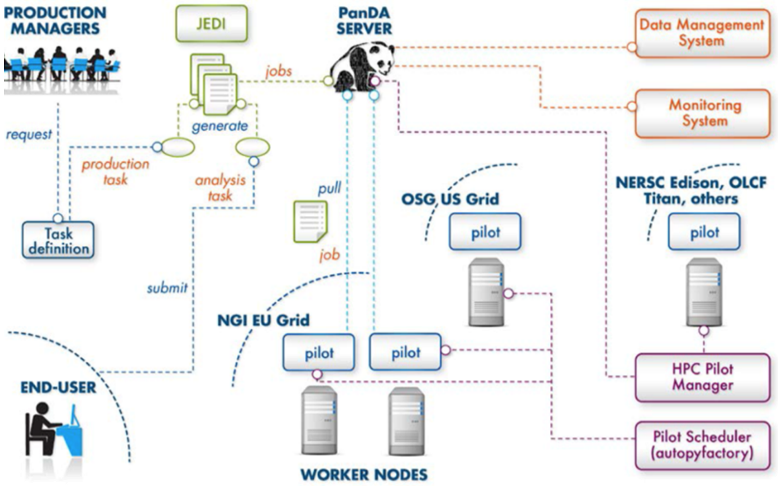

Figure 1. Schematic view of the PanDA WMS and ATLAS production system

spanning well over 100 computing centers. The Grid infrastructure is sufficient for the current analysis and data processing, but it will fall short of the requirements for the future high luminosity runs at LHC. In order to meet the challenges of ever growing demand for computational power and storage, ATLAS initiated a program to expand the current computing model to incorporate additional resources, including supercomputers and high-performance computing clusters.

In this paper we will describe a project aimed at integration of the ATLAS production system with the Titan supercomputer at the Oak Ridge Leadership Computing Facility (OLCF).

2. PanDA workload management system

ATLAS utilizes the PanDA Workload Management System [4] (WMS) for job scheduling on the distributed computational infrastructure. The acronym PanDA stands for Production and Distributed Analysis. The 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.

Currently, as of 2016, PanDA WMS manages processing of over one million jobs per day, serving thousands of ATLAS users worldwide. It is capable of executing and monitoring jobs on heterogeneous distributed resources which include WLCG, supercomputers, and public and private clouds. Figure 1 shows a schematic view of the PanDA WMS and ATLAS production system. More details about PanDA can be found in [4], here we will only describe some features of PanDA relevant for this paper.

PanDA is a pilot [5] based WMS. On the Grid pilot jobs (Python scripts that organize workload processing on a worker node) are submitted to batch queues on compute sites and wait for the resource to become available. When a pilot job starts on a worker node it contacts the PanDA server to retrieve an actual payload and then, after necessary preparations, executes the payload as a subprocess. This facilitates late binding of user jobs to computing resources. The late binding helps to optimize global resource utilization, minimize user jobs wait times and mitigate many of the problems associated with the inhomogeneities found on the Grid. The PanDA pilot is also responsible for a job's data management on a worker node and can perform data stage-in and stage-out operations.

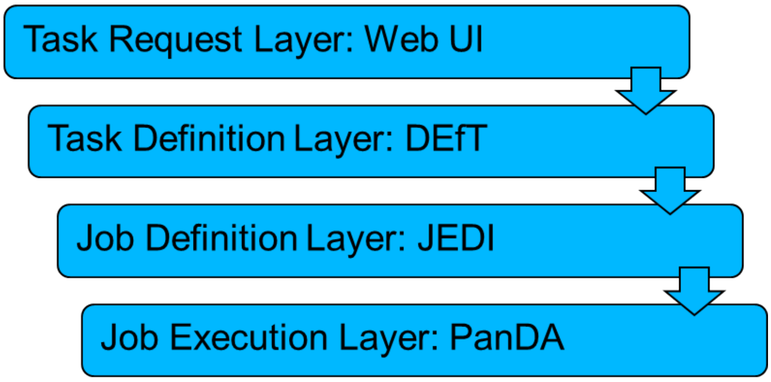
Task Request Layer: Web UI
Task Definition Layer: DEfT
Job Definition Layer: JEDI
Job Execution Layer: PanDA

Figure 2. Schematic view of the ATLAS production system

3. ATLAS production system

The ATLAS Production system (ProdSys) is a layer that connects distributed computing and physicists in a user friendly way. It is intended to simplify and automate the definition and execution of multi-step workflows, like ATLAS detector simulation and reconstruction. The Production system consists of several components or layers and is tightly connected to the PanDA WMS. Figure 2 shows the structural layout of the ATLAS Production system. The Task Request layer presents a web based user interface that allows for high level task definition. The Database Engine for Tasks (DEfT) is responsible for definition of tasks, chains of tasks and also task groups (making up a production request), complete with all necessary parameters. It also keeps track of the state of production requests, chains of tasks and their constituent tasks. The Job Execution and Definition Interface (JEDI) is an intelligent component in the PanDA server that performs task-level workload management. JEDI can dynamically split workloads and automatically merge outputs. Dynamic job definition, which includes job resizing in terms of number of events, helps to optimize usage of heterogeneous resources, like multi-core nodes on the Grid, HPC clusters and supercomputers, as well as commercial and private clouds. In this system the PanDA WMS can be viewed as a job execution layer responsible for resource brokerage and submission and resubmission of jobs. More details about the ATLAS production system can be found in the Ref [6].

4. Titan at OLCF

The Titan supercomputer [7], currently number three (number one until June 2013) on the Top 500 list [8] is located at the Oak Ridge Leadership Computing Facility in the Oak Ridge National Laboratory, USA. It has theoretical peak performance of 27 PetaFLOPS. Titan was the first large-scale system to use a hybrid architecture that utilizes worker nodes with both AMD 16-core Opteron 6274 CPUs and NVIDIA Tesla K20 GPU accelerators. It has 18,688 worker nodes with a total of 299,008 CPU cores. Each node has 32 GB of RAM and no local disk storage, though a RAM disk can be set up if needed, with a maximum capacity of 16 GB. Worker nodes use Cray's Gemini interconnect for inter-node MPI messaging but have no network connection to the outside world. Titan is served by a shared Lustre filesystem that has 32 PB of disk storage (about 100 TB of which is allocated for the ATLAS project) and by HPSS tape storage that has 29 PB capacity. Titan's worker nodes run Compute Node Linux which is a run time environment based on the Linux kernel derived from SUSE Linux Enterprise Server.

5. Integration with Titan

The project aims to integrate Titan with the ATLAS Production system and PanDA. Details of integration of the PanDA WMS with Titan were described in Ref [9]. Here we will briefly describe some of the salient features of the project implementation.

Taking advantage of its modular and extensible design, the PanDA pilot code and logic has been enhanced with tools and methods relevant for work on HPCs. The pilot runs on Titan's data transfer nodes (DTNs) which allows it to communicate with the PanDA server, since DTNs have good (10 GB/s) connectivity to the Internet. The DTNs and the worker nodes on Titan use a shared file system which makes it possible for the pilot to stage-in input files that are required by the payload and stage-out produced output files at the end of the job. In other words the pilot acts as a site edge service for Titan. Pilots are launched by a daemon-like script which runs in user space. The ATLAS Tier 1 computing center at Brookhaven National Laboratory is currently used for data transfer to and from Titan, but in principle that can be any Grid site. The pilot submits ATLAS payloads to the worker nodes using the local batch system (Moab) via the SAGA (Simple API for Grid Applications) interface [10]. It also uses SAGA for monitoring and management of PanDA jobs running on Titan's worker nodes.

One of the main features of the project is the ability to collect and use information about free worker nodes (backfill) on Titan in real time. Estimated backfill capacity on Titan is about 400M core hours per year. In order to utilize backfill opportunities, new functionality has been added to the PanDA pilot. The pilot can query the Moab scheduler about currently unused nodes on Titan and check if the free resource availability time and size are suitable for PanDA jobs, and conforms with Titan's batch queue policies. The pilot transmits this information to the PanDA server, and in response gets a list of jobs intended for submission on Titan. Then based on the job information it transfers the necessary input data from the ATLAS Grid, and once all the necessary data is transferred the pilot submits jobs to Titan using an MPI wrapper [9].

The MPI wrappers are Python scripts that are typically workload specific since they are responsible for setup of the workload environment, organization of per-rank worker directories, rank-specific data management, optional input parameters modification, and cleanup on exit. The wrapper script is what the pilot actually submits to a batch queue to run on Titan. Using backfill availability information the pilot reserves the necessary number of worker nodes at submission time, and then at run time a corresponding number of copies of the wrapper script will be activated on Titan. Each copy will know its MPI rank (an index that runs from zero to a maximum number of nodes or script copies) as well as the total number of ranks in the current submission. When activated on worker nodes each copy of the wrapper script after completing the necessary preparations will start the actual payload as a subprocess and will wait until its completion. In other words the MPI wrapper serves as a "container" for non-MPI workloads and allows efficient execution of unmodified Grid-centric workloads on parallel computational platforms such as Titan.

It is worth mentioning that the combination of the free resources discovery capability with flexible payload shaping via MPI wrappers allowed us to achieve very low waiting times for ATLAS jobs on Titan, on the order of 70 seconds on average. That is much lower than the typical waiting times on Titan, which can be hours or even days depending on the size of submitted jobs. Figure 3 shows the schematic diagram of PanDA components on Titan.

6. Running ATLAS production on Titan

The first ATLAS simulation jobs sent via the ATLAS Production System started to run on Titan in May 2015. After successful initial tests the scale-up of continuous production on Titan took place by increasing the number of pilots and introducing improvements in various system components. Data delivery to and from Titan was fully automated and synchronized with job

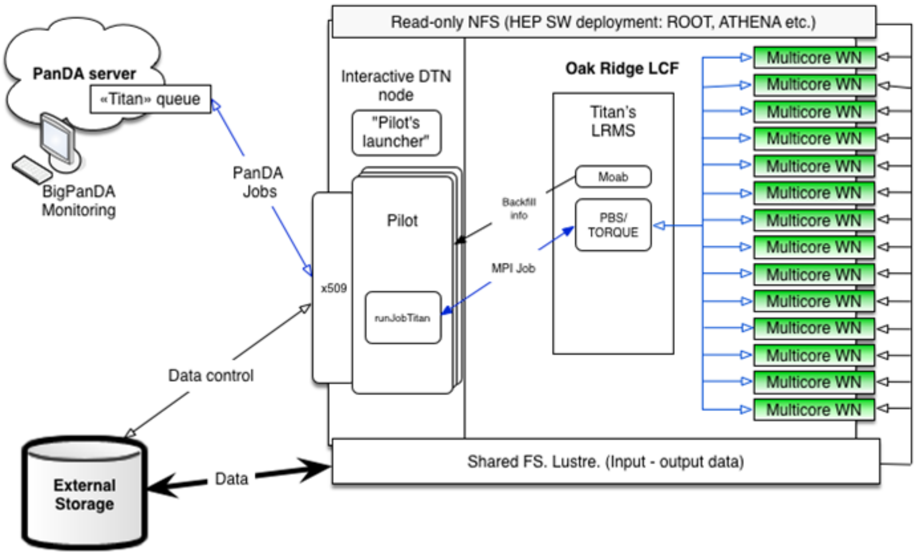

Figure 3. Schematic view of PanDA interface to Titan

delivery and execution.
Currently up to 20 pilots are deployed at a time, distributed evenly over 4 DTNs. Each pilot controls from 15 to 300 ATLAS simulation jobs per submission. This configuration is able to utilize up to 96,000 cores on Titan. We expect that these numbers will grow in the near future. Since November 2015 operation has proceeded in pure backfill mode, without a defined time allocation, running at the lowest priority on Titan. Figure 4 shows Titan core hours used by ATLAS from September 2015 to October 2016. During that period ATLAS consumed about 61 million Titan core hours. About 1.8 million detector simulation jobs were completed, and more than 132 million events processed. In September 2016 we exceeded 1 million core hours per month and by October 2016 we ran at more than 9 million core hours per month, reaching 9.7 million in August 2016. Figure 5 shows backfill utilization efficiency, defined as a fraction of Titan's total free resources that were utilized by ATLAS, for the period from September 2015 to October 2016. Due to continuos improvements in the system the efficiency grew from less than 5% to more than 20%, reaching 27% in August 2016. At the peak that corresponds to more than 2% improvement in Titan's overall utilization.

7. Conclusion

In this paper we described a project aimed at integration of the ATLAS Production system with the Titan supercomputer at Oak Ridge Leadership Computing Facility. The current approach utilizes a modified PanDA pilot framework for job submission to Titan's batch queues and for local data management, with lightweight MPI wrappers to run single node workloads in parallel on Titan's multi-core worker nodes. It also gives PanDA a new capability to collect, in real time, information about unused worker nodes on Titan, which allows us to precisely define the size and duration of jobs submitted to Titan according to available free resources. This capability significantly reduces the PanDA job wait time while improving Titan's overall

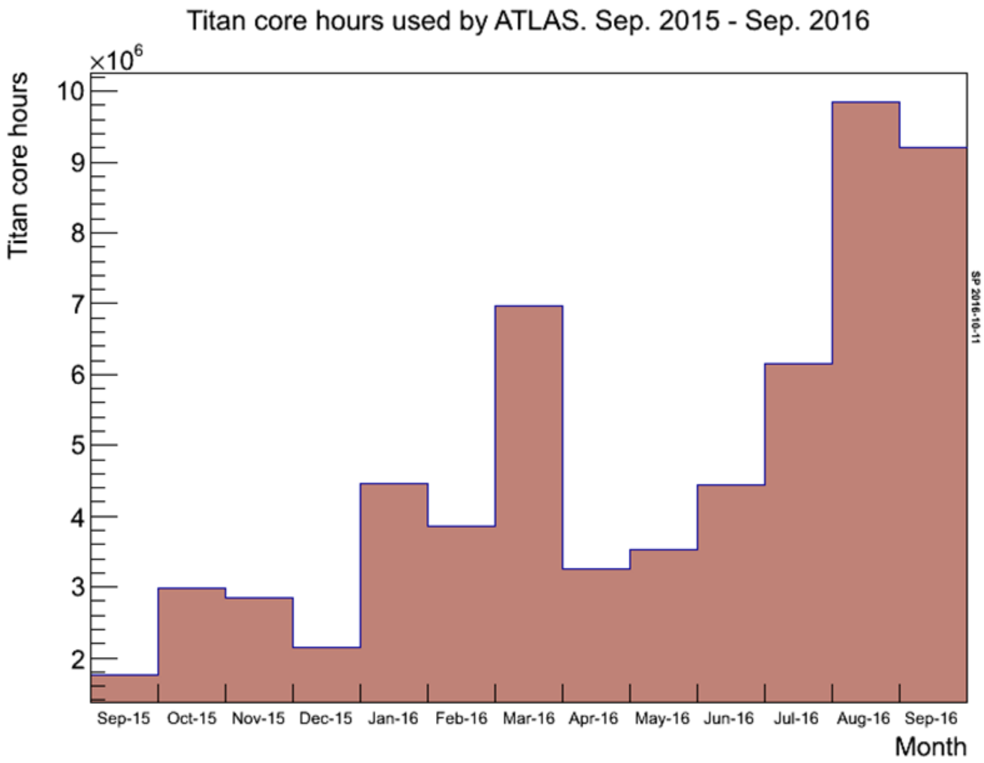

Figure 4. Titan core hours used by ATLAS from September 2015 to October 2016

utilization efficiency.
ATLAS production jobs have been continuously submitted to Titan since September 2015. As a result, during the period from September 2015 to October 2016, ATLAS consumed about 61 million Titan core hours, about 1.8 million detector simulation jobs were completed, and more than 132 million events processed. The work underway is enabling the use of PanDA by new scientific collaborations and communities as a means of leveraging extreme scale computing resources with a low barrier of entry. The technology base provided by the PanDA system will enhance the usage of a variety of high-performance computing resources available to basic and applied research.

8. Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award No. DE-SC-ER26106. We would like to acknowledge that this research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] ATLAS Collaboration, Aad J *et al.* 2008 The ATLAS Experiment at the CERN Large Hadron Collider *J. Inst.* **3** S08003

[2] Adams D, Barberis D, Bee C, Hawkings R, Jarp S, Jones R, Malon D, Poggioli L, Poulard G, Quarrie D and Wenaus T, 2005 *The ATLAS computing model* Tech. rep. CERN CERN-LHCC-2004-037/G-085

[3] Foster I and Kesselman C 1998 *The Grid: Blueprint for a New Computing Infrastructure* (San Francisco: Morgan Kaufmann Publishers)

[4] Maeno T *et al.* 2011 Overview of ATLAS PanDA workload management *J. Phys.: Conf. Ser.* **331** 072024

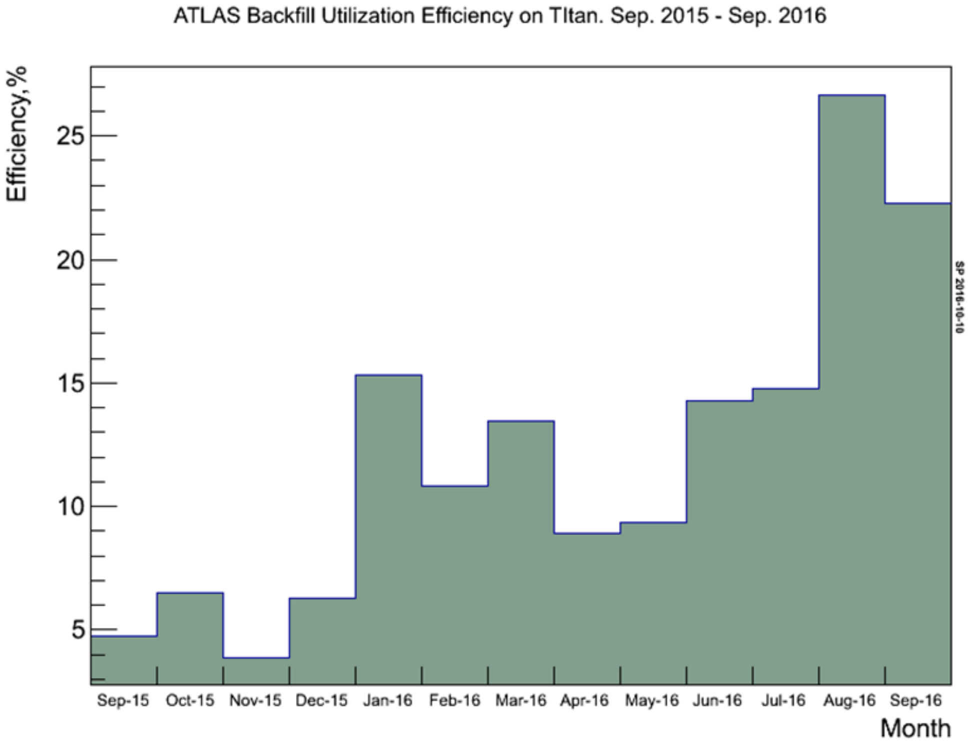

Figure 5. Backfill utilization efficiency from September 2015 to October 2016. See description in the text.

[5] Nilsson P *et al.* 2011 The ATLAS PanDA Pilot in Operation *J. Phys.: Conf. Ser.* **331** 062040

[6] 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

[7] Titan at OLCF web page. https://www.olcf.ornl.gov/titan/

[8] Top500 List. http://www.top500.org/

[9] De K *et al.* 2015 Integration of PanDA workload management system with Titan supercomputer at OLCF *J. Phys.: Conf. Ser.* **664** 092020

[10] The SAGA Framework web site. http://saga-project.github.io