Integration of PanDA workload management system with Titan supercomputer at OLCF

K. De[1], A. Klimentov[2], D. Oleynik[1], S. Panitkin[2], A. Petrosyan[1], J. Schovancova[2], A. Vaniachine[3] and T. Wenaus[2] on behalf of the ATLAS Collaboration

[1] Department of Physics University of Texas at Arlington, Arlington, TX 76019, USA
[2] Brookhaven National Lab, Upton, NY 10573, USA
[3] Argonne National Lab, 9700 S. Cass Avenue, Lemont, IL 60439, USA

ATL-SOFT-PROC-2015-040
15 May 2015

Abstract.
The PanDA (Production and Distributed Analysis) workload management system (WMS) was developed to meet the scale and complexity of LHC distributed computing for the ATLAS experiment. While PanDA currently distributes jobs to more than 100,000 cores at well over 100 Grid sites, next LHC data taking run will require more resources than Grid computing can possibly provide. To alleviate these challenges, ATLAS is engaged in an ambitious program to expand the current computing model to include additional resources such as the opportunistic use of supercomputers.

We will describe a project aimed at integration of PanDA WMS 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 light-weight MPI wrappers to run single threaded workloads in parallel on Titan's multi-core worker nodes. It also gives PanDA new capability to collect, in real time, information about unused worker nodes on Titan, which allows precisely define the size and duration of jobs submitted to Titan according to available free resources. This capability significantly reduces PanDA job wait time while improving Titan's utilization efficiency. This implementation was tested with a variety of Monte-Carlo workloads on Titan and is being tested on several other supercomputing platforms.

1. Introduction

The ATLAS experiment [1] at the Large Hadron Collider (LHC) is designed to explore fundamental properties of matter at the highest energy ever achieved in a laboratory. Since the LHC became operational six years ago, the experiment has produced and distributed more than 150 petabytes of data worldwide. Thousands of ATLAS physicists are engaged in daily analysis of this data.

Modern research in high-energy physics is practically impossible without massive computing infrastructure. ATLAS currently uses more than 100,000 CPU cores deployed in a hierarchical Grid [2, 3], spanning well over 100 computing centers. But the next LHC runs will require more resources than this Grid can possibly provide to sustain the pace of the proposed research. The Grid infrastructure will be sufficient for the planned analysis and data processing, but it will fall short of the requirements for the large-scale Monte Carlo simulations as well as any extra activities. Additional computing and storage resources are therefore required. To meet these challenges, ATLAS is engaged in an ambitious 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 PanDA Workload Management System (WMS) [4] with Titan supercomputer at Oak Ridge Leadership Computing Facility (OLCF).

2. PanDA workload management system

ATLAS utilizes the PanDA WMS for distributed data processing and analysis. Acronym PanDA stands for Production and Distributed Analysis, the system has been developed to meet growing ATLAS production and analysis requirements for a data-driven workload management system capable of operating at LHC data processing scale.

PanDA has a highly scalable architecture. Scalability has been demonstrated in ATLAS through the rapid increase in usage over the past several years of operations, and is expected to meet the continuously growing number of jobs over the next decade. Currently, as of 2015, PanDA WMS manages processing of over one million jobs per day on ATLAS Grid. PanDA was designed to have the flexibility to adapt to emerging computing technologies in processing, storage, networking as well as the underlying software stack (middle-ware). This flexibility has also been successfully demonstrated through the past six years of evolving technologies adapted by computing centers in ATLAS which span many continents and yet are seamlessly integrated into PanDA.

PanDA is a pilot [5] based WMS. In the PanDA job lifecycle, pilot jobs (Python scripts that organize workload processing on a worker node) are submitted to compute sites. When these pilot jobs start on a worker node they contact central server to retrieve a real payload (i.e., an end-user job) and execute it. Using these pilot-based workflows helps to improve job reliability, optimize resource utilization, allows for opportunistic resources usage, and mitigates many of the problems associated with the inhomogeneities found on the Grid.

3. HPC platforms and PanDA

The High Performance Computing (HPC) platforms encompass a broad spectrum of computing facilities, ranging from small-scale interconnected clusters to the largest supercomputers in the world. They are rich sources of CPUs some claiming more cores than the entire ATLAS Grid. The HPC machines are built to execute large scale parallel, computationally intensive, workloads with high efficiency. They provide high-speed interconnects between worker nodes and facilities for low latency inter-node communications. In this environment, significant limitations can exist: typically the worker node setup is fixed, with no outbound network connections, amount of RAM per core can be quite limited, and in many cases highly customized operating system is used along with specialized software stack. All of this factors pose significant challenge for integration with PanDA WMS that was initially designed to be used on Grid resources.

4. Titan at OLCF

The Titan supercomputer [6], current number two (number one until June 2013) on the Top 500 list is located at the Oak Ridge Leadership Computing Facility. in Oak Ridge National Laboratory, USA. It has theoretical peak performance of 29 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 shared Lustre filesystem that has 32 PB of disk storage and by the 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. PanDA integration with Titan

The project aims to integrate Titan with the PanDA system using standard PanDA pilot code and logic, enhanced with tools and methods relevant for HPC. PanDA pilot was designed to be modular and extensible. It's architecture allows for transparent addition of the new functionality needed for integration with HPC resources to the code base developed for the Grid. The pilot runs on Titan's front-end nodes which allows it to communicate with PanDA server, since front end nodes have connectivity to the Internet. The interactive front-end machines and the worker nodes 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 the produced output files at the end of the job. The ATLAS Tier 1 computing center at Brookhaven National Lab 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 (PBS) via the SAGA (Simple API for Grid Applications) interface [7]. It also uses SAGA facilities for monitoring and management of PanDA jobs running on Titan's worker nodes. Figure 1 shows the schematic diagram of PanDA components on Titan. We will describe relevant parts of the system in more details in subsequent sections.

5.1. MPI wrappers

The majority of experimental high energy physics workloads do not use Message Passing Interface (MPI). They are designed around event level parallelism and thus are executed on the Grid independently. Typically, detector simulation workloads can run on a single compute node using multiprocessing. A notable exception is ATLAS Event Service (ES) [8] workload that was specifically designed for opportunistic resources like HPC and Cloud Computing. In particular ES implementation for HPC [9] utilizes MPI natively for event distribution to worker nodes. For running "ordinary" - non ES compatible - workloads on Titan we developed MPI wrapper that allows us to launch multiple instances of single node workers simultaneously. MPI wrappers are typically workload specific since they are responsible for set up of workload specific environment, organization of per-rank worker directories, rank specific data management, input parameters modification when necessary and cleanup on exit. The wrapper scripts is what pilot actually submits to a batch queue to run on Titan. The pilot reserves necessary number of worker nodes at submission time and 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 necessary preparations, will start actual payload as a subprocess and waits until its completion. In other words MPI wrapper serves as a "container" for non-MPI workloads

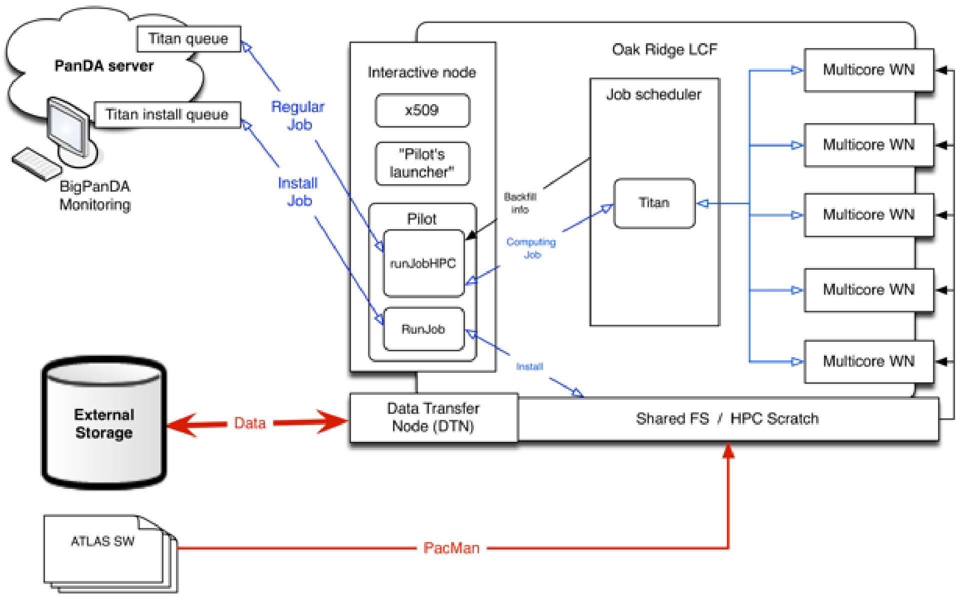

Figure 1. Schematic view of PanDA interface with Titan

and allows us to efficiently run unmodified Grid-centric workloads on a parallel computational platforms, like Titan.

5.2. Backfill: Capturing free resources on Titan

Leadership Computing Faclilities (LCF), like Titan, are geared towards large scale jobs by design. Time allocation on an LCF machines is very competitive and large scale projects are often preferred. This is especially true for Titan at OLCF which was designed to be the most powerful machine in the world capable to run extreme scale computational projects. As a consequence, on average, about 10% of capacity on Titan is unused due to mismatches between job sizes and available resources. The worker nodes sit idle because there are not enough of them to handle a large scale computing job. On Titan, these 10% correspond to estimated 300M core hours per year. Hence, a system that can occupy those temporarily free nodes would be very valuable. It would allow to deliver more CPU cycles for scientific research while simultaneously improving resource utilization efficiency on Titan. This offers a great possibility for PanDA to harvest these opportunistic resources on Titan. Functionality has been added to the PanDA pilot to interact with Titan's scheduler and collect information about available unused worker nodes on Titan. This allows PanDA pilot to define precisely size and duration of jobs submitted to Titan according to available free resources. One additional benefit of this implementation is very short wait times before the jobs execution on Titan. Since PanDA submitted jobs match currently available free resources exactly or at least very closely, they present, in majority of cases, the best solution for Titan's job scheduler to achieve maximum resource utilization, hence short wait times for these jobs. First tests of the system were quite successful and show great promise of increasing the resource utilization on Titan. We demonstrated increased system utilization levels by ≈3% of total capacity (≈14.3% of free cycles) during the performance period of this test, as well as provided short wait times (≈70 seconds) to jobs submitted to Titan via PanDA; all with no negative impact on OLCFs ability to schedule large scale jobs. We should comment here that these results were obtained with single stream of pilots. Multiple pilot streams will improve backfill utilization.

5.3. Workloads and ATLAS software deployment model for Titan

Typically, installation on Titan of stand alone HEP physics software packages, like event generators, Root, etc, consists of a straightforward compilation and build from source code. In order to run ATLAS production workloads, we need to install and maintain official software releases and corresponding databases on Titan. Usually, on the Grid, ATLAS software releases are distributed via CVMFS [10] . We tried to install CVMFS on on Titan's login nodes and then copy software releases to a shared file system so that they are available on worker nodes. Normally, on the Grid, CVMFS uses FUSE [11] kernel module. On Titan FUSE kernel module can not be installed because of security and other concerns. So, instead we tried CVMFS with Parrot [12] and this combination was not very stable for us. As a result, we decided to utilize package manager called pacman [13], that was used by ATLAS for software distribution before the advent of CVMFS. The release installation procedure on Titan is automated and performed by running a single script that performs the complete installation of the necessary components. The same script can be triggered via a special software installation queue created for Titan in PanDA.

During the course of the project, we ported to Titan a number of HEP workloads ranging from Monte-Carlo event generators to full Geant4 [14] detector simulation chains. We installed ATLAS software releases on Titan and ran several types of calculations for ATLAS physics analyses, concentrating on those which can not be effectively simulated on the Grid.

For example, we worked with a group of ATLAS physicists from NYU and Manhattan College to calculate the Vector Boson Fusion channel for Higgs productions and delivered more than 15 million simulated events. Figure 2 shows distributions in invariant mass of Z-bosons in the process pp→H→ZZ→4l . This distribution was obtained from calculations with PowhegBox/HJJ [15] model performed on Titan.

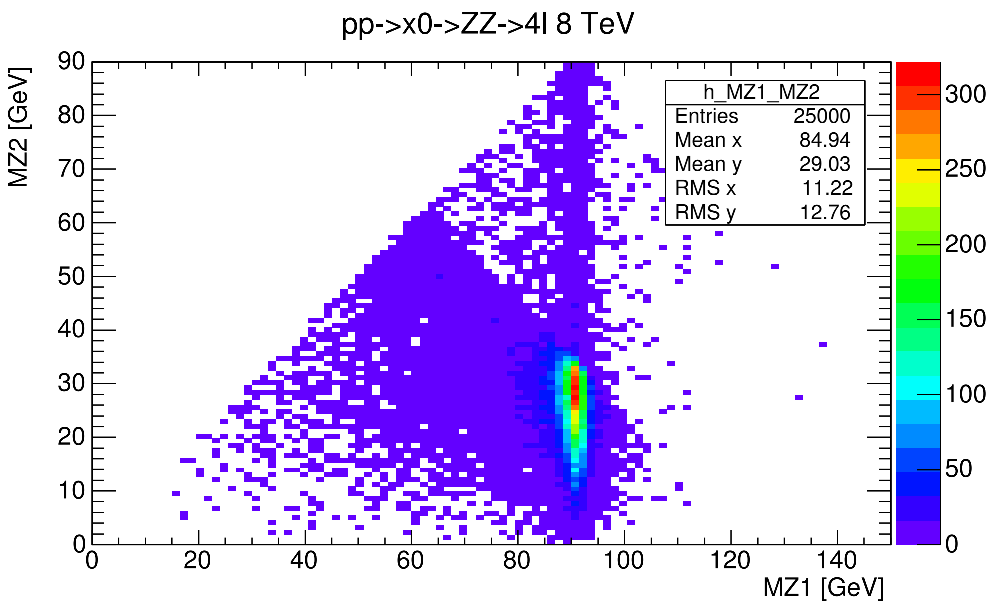

Figure 2. Invariant mass distribution of Z-bosons in the pp→H→ZZ→4l process.

We also implemented integration with the new ATLAS production system (ProdSys 2) developed for management of ATLAS Monte-Carlo production during LHC run 2. The production task can be defined for Titan by ATLAS production manager and PanDA jobs will be directed to Titan along with coordinated movement of input and output data from and to ATLAS Grid. At the time of writing a validation sample of ATLAS production Geant simulation is being generated on Titan.

6. Extentions beyond ATLAS and Titan

It was important to demonstrate that new PanDA capabilities developed for Titan can be used by scientific communities outside of ATLAS as well as at other HPC facilities.
In collaboration with physicists from the Electron Ion Collider (EIC) project at BNL, we ported FairRoot [16] and EICRoot software frameworks to Titan and performed initial validation of the code.
With colleagues from the ALICE experiment [17] at the LHC we prepared and ran Geant-based Monte Carlo simulations of the ALICE detector, which included runs with multi-threaded version of Geant4. For these tests ALICE specific MPI wrapper were developed and jobs were submitted to Titan, by ALICE scientists, via PanDA server deployed on Amazon EC2 cloud.
We also demonstrated that technology developed for Titan can be used at other supercomputing platforms. The setup was deployed on supercomputers at NERSC at LBNL and Anselm at IT4I in Ostrava, Czech Republic, both sites are used by the ATLAS Collaboration.

7. Conclusion

The LHC run 2 will pose massive computing challenges for ATLAS. With a doubling of the beam energy and luminosity as well as an increased need for simulated data, the data volume is expected to increase with a factor 5-6 or more. Storing and processing this amount of data is a challenge that cannot be resolved with the currently existing computing resources in ATLAS. To resolve this challenge, ATLAS is exploring use of supercomputers and HPC clusters via the PanDA system. In this paper we described a project aimed at integration of PanDA WMS with Titan supercomputer at Oak Ridge Leadership Computing Facility. Current approach utilizes modified PanDA pilot framework for job submission to Titan's batch queues and local data management, with light-weight MPI wrappers to run single threaded workloads in parallel on Titan's multi-core worker nodes. It also gives PanDA 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 PanDA job wait time while improving Titan's utilization efficiency. Also 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 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.*, *The ATLAS Experiment at the CERN Large Hadron Collider*, J. Inst. 3 (2008) S08003.
[2] Adams D, Barberis D, Bee C, Hawkings R, Jarp S, Jones1 R, Malon D, Poggioli L. Poulard, G, Quarrie D and Wenaus T *The ATLAS computing model* Tech. rep. CERN CERN-LHCC-2004-037/G-085, 2005

[3] Foster I. and Kesselman C. (eds) *The Grid: Blueprint for a New Computing Infrastructure* (Morgan-Kaufmann), 1998
[4] Maeno T. *et al.*, (2011) *Overview of ATLAS PanDA workload management*, J. Phys.: Conf. Ser. 331 072024
[5] Nilsson P., *The ATLAS PanDA Pilot in Operation*, in Proc. of the 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)
[6] Titan at OLCF web page. https://www.olcf.ornl.gov/titan/
[7] The SAGA Framework web site, http://saga-project.github.io
[8] Calafiura P. *et al.* for the ATLAS Collaboration (2015) The ATLAS Event Service: A New Approach to Event Processing. Proceedings of the CHEP 2015 Conference. *J. Phys.: Conf. Ser.*
[9] Calafiura P. *et al.* for the ATLAS Collaboration (2015). Fine grained event processing on HPCs with the ATLAS Yoda system. Proceedings of the CHEP 2015 Conference. *J. Phys.: Conf. Ser.*
[10] Blomer J. *et al.* 2011 *Distributing LHC application software and conditions databases using the CernVM file system*, J. Phys.: Conf. Ser. 331 042003
[11] FUSE project web site: http://fuse.sourceforge.net/
[12] Parrot project web site: http://www.nd.edu/ ccl/software/parrot/
[13] Pacman in ATLAS web site: http://heppc1.uta.edu/atlas/software/pacman.htm
[14] GEANT4 Collaboration, S. Agostinelli et al., 2003 Nucl. Instrum. Meth. A 506 250
[15] Campbell J. M. *et al.*, (2012) *NLO Higgs boson production plus one and two jets using the POWHEG BOX, MadGraph4 and MCFM* JHEP 1207 092, arXiv:1202.5475
[16] FAIR Root Project web site: http://fairroot.gsi.de
[17] Aamodt K. *et al.* (2008) *The ALICE experiment at the CERN LHC* J. Instrum. 3, S08002