# BigPanDA Workflow Management on Titan for High Energy and Nuclear Physics and for Future Extreme Scale Scientific Applications

Progress report - May 2017

# Table of content


**Introduction** **2**

**Personnel and Responsibilities** **3**

**Progress with Research Tracks** **4**

Track 1: Adaptation of BigPanDA WMS for OLCF 4

HEP scientific applications on OLCF 5

BigPanDA and the BlueBrain Project 8

Track 2: Data Transfer and Access 8

Workload run-time optimization 8

Pilot launcher optimization 8

Move of pilot operations to Data Transfer nodes 9

Track 3: Adaptive Execution of Next-generation Workloads 10

ATLAS workloads with NGE 11

Molecular Dynamics workloads with NGE 13

Future Work 14

Lessons Learned 16

Track 4: Operations and Operational Model 17

Examining the I/O impact of PanDA@OLCF 17

PanDA instance installation at ORNL 18

**Plans for Year 2 (PY2)** **19**

Track 1: Adaptation of BigPanDA WMS for OLCF 19

Track 2: Data Transfer and Access 19

Track 3: Adaptive Execution of Next-generation Workloads 20

Track 4: Operations and Operational Model 20

**Summary** **21**

**Publications, Conference Talks, and Workshops 2016/2017** **21**

**Bibliography** **22**

# Introduction

The goal of this project is to translate the R&D artifacts and accomplishments from the BigPanDA and AIMES projects into LCF operational advances and enhancement. This project will provide an important model for future exascale computing, increasing the coherence between the technology base used for high-performance computing and that used for data-analytic computing, and thereby directly advancing the first of the five objectives of the National Strategic Computing Initiative announced by the White House in June 2015. This proposal also directly advances the first of five strategic themes articulated by the White House Office of Science and Technology Policy (OSTP) to "create systems that can apply exaflops of computing power to exabytes of data."

Achieving operational goals are important steps towards enabling the convergence of HPC and HTC “paradigms” for a range of application types and communities. In meeting the operational goals of this demonstrator project, we have both developed practical experience as well conceptual insight into some of the challenges which will ultimately help ASCR devise a working plan of engagement and support with science applications from other DOE offices. These include but are not limited to supporting diverse use-cases and requirements, sharing workloads across HTC and HPC resources, and investigating the operational implications of running traditional HTC workloads on HPC resources. The latter has required working with and hardening implementations of standards such as OpenMPI as well as prototyping possible architectures for “edge” and other software services on leadership class facilities. Throughout this project, the operational model has prioritized enabling high impact scientific advances for a wide range of communities through the merging of HPC computing resources as an integral part of distributed resources traditionally used by HTC communities.

During the first year of this project we achieved progress on all four work areas described in the proposal. First, we developed detailed work plans, defined development goals and chose operational metrics. Next, we focused on early prototyping and implementation of production services. We significantly improved the operational model for LHC experiments on LCF resources, demonstrating the power of the PanDA system on Titan. Working with the LSST Dark Energy Science Collaboration (DESC), we implemented and tested a prototype system to run DESC simulations on Titan using a PanDA server installed at Amazon EC2. We have installed PanDA servers at Oak Ridge National Laboratory (ORNL) and at Blue Brain Project (BBP), to engage a wide range of new communities beyond HEP and NP.

On the operational side we addressed critical issues, identified in the proposal, with I/O performance of ATLAS simulation workloads on Titan's file system. We increased the rate of ATLAS CPU consumption on Titan to ~14M core hours/per month and achieved backfill utilization efficiency of ~30%, without negative impact on Titan's shared file system. We have integrated File Transfer service (aka FTS3) with PanDA and we shipped data to/from OLCF/BNL. We have addressed MPI issues and implemented improved PanDA-Titan monitoring.

# Personnel and Responsibilities

There are several project participants in BigPanDA from Brookhaven National Laboratory (BNL). Alexei Klimentov is the overall project leader, he has not drawn any salary support from this project. As the co-leader of the ATLAS Workflow Management Software project and ATLAS HPC projects, Klimentov is one of primary architects of the ATLAS Production system. Sergey Panitkin at BNL is supported 100% from this project. He worked in the BigPanDA team since the beginning of the project in 2012. Panitkin is PI of Track 2 : Data Transfer and Access and he also works as a liaison with nEDM experiment and LQCD community. Pavlo Svirin is a new hire at BNL, since February 2017, funded 100% through this proposal to bring NP community (in particular ALICE) into this project. Svirin is also working together with the LSST DESC community to port their applications to Titan PanDA.

There are three active participants in BigPanDA from the University of Texas at Arlington (UTA). Kaushik De is the coPI from UTA. He has not drawn any salary support from this project so far, since he was funded locally by UTA in FY17. De supervises Research Track 1: Workload Management System. De started the highly successful PanDA project for ATLAS in 2005 and has played a supervisory role in the project ever since. Danila Oleynik is supported 50% from this project. He has worked on BigpanDA since 2013, and is the developer of the HPC/Titan pilot. He has participated in all phases of the BigPanDA project. He is supported by DOE-HEP for the remaining 50% of his time, as a key developer of PanDA, to work on ATLAS computing. Ruslan Mashinistov is a new hire at UTA, since January 2017, funded 100% through this proposal, to bring other communities into the BigPanDA ecosystem. Ruslan has installed multiple PanDA servers for a wide range of non-ATLAS communities, and is providing support for these communities to port their applications to Titan PanDA.

At Rutgers, Alessio Angius (AA) serves as the Post-doctoral Researcher on this project, and Matteo Turilli (MT) as the Research Scientist. AA has been primarily responsible for experiments that have characterized the use of NGE for various workloads on TITAN; MT has been primarily responsible for the design and integration of NGE with PanDA. AA and MT have both been supported at 50% by this project since it was established. The project also provides training opportunities for two graduate students -- Ming Ta Hai and Manuel Maldonado. Jha serves as the PI at Rutgers providing overall technical leadership and outreach to biomolecular sciences community.

At ORNL, all support has been in-kind effort from the OLCF project within the National Center for Computational Sciences (NCCS). Jack Wells, Director of Science for NCCS, serves at the coP and supervises Research Track 4. Sarp Oral from the NCCCS Technology Integration Group has worked on the monitoring and performance issues associated with the Lustre filesystem (Spider). Jason Kincl from the NCCS HPC Operations Group has developed the OpenShift container platform within which the PanDA instance at OLCF is deployed. Jason Kincl and Daniel Pelfrey, again from HPC Operations, developed and deployed the DTN upgrade project and consulted with BigPanDA collaborators on the strategy and deployment of the PanDA Pilot on the DTN cluster. Adam Simpson from the NCCS User Assistance Group has

made critical contributions consulting on the deployment of the ATLAS/Athena software stack at OLCF. Participation from these NCCS staff is critical to the operational success of the BigPanDA project. While ORNL is not directly funded through the ASCR research grant, active involvement of ORNL staff advances the success of this project.

# Progress with Research Tracks

The overarching aim of this project is to integrate several strands of active research and translate them into a scalable and operational model for the next generation of data-intensive workloads on Titan, and subsequent DOE leadership-class machines. The research strands that will be integrated span applications and workload characterization, hardening of operational services to system level optimization, with significant algorithmic and scheduling advances in between. We organized the research into four tracks, which have well-defined and independent objectives. The progress for each track is described in more details below.

## Track 1: Adaptation of BigPanDA WMS for OLCF

The goal of this track is to continue the development of workload management systems of the sophistication and scale required to meet the science objectives of the future. We are undertaking significant architectural modifications so as to optimize multi-job capabilities of the PanDA pilot, to improve software packaging and installation of PanDA server, and to optimize and improve application execution characteristics on Titan. Track one will also provide support for heterogeneous workloads and additional scientific communities through the generalization of workload definitions and the enhanced and flexible support for MPI-based workloads. This track provides important improvements to the interface layer between workload management systems including application characteristics and OLCF as a facility.

### HEP scientific applications on OLCF

Starting in 2015, the previously ASCR funded BigPanda project demonstrated the usage of OLCF as a valuable computing resource for the ATLAS experiment. Following up on that success, during the first year of this current project, which started in the summer of 2016, we hardened our use of Titan for ATLAS simulations, and integrated OLCF fully in the data processing chain of the experiment. Specifically:

- we methodically increased the usage of Titan, with continuous improvements of the software systems and production operations
- we learned valuable lessons from real operational experience that informed our design of the architecture of the next generation of software

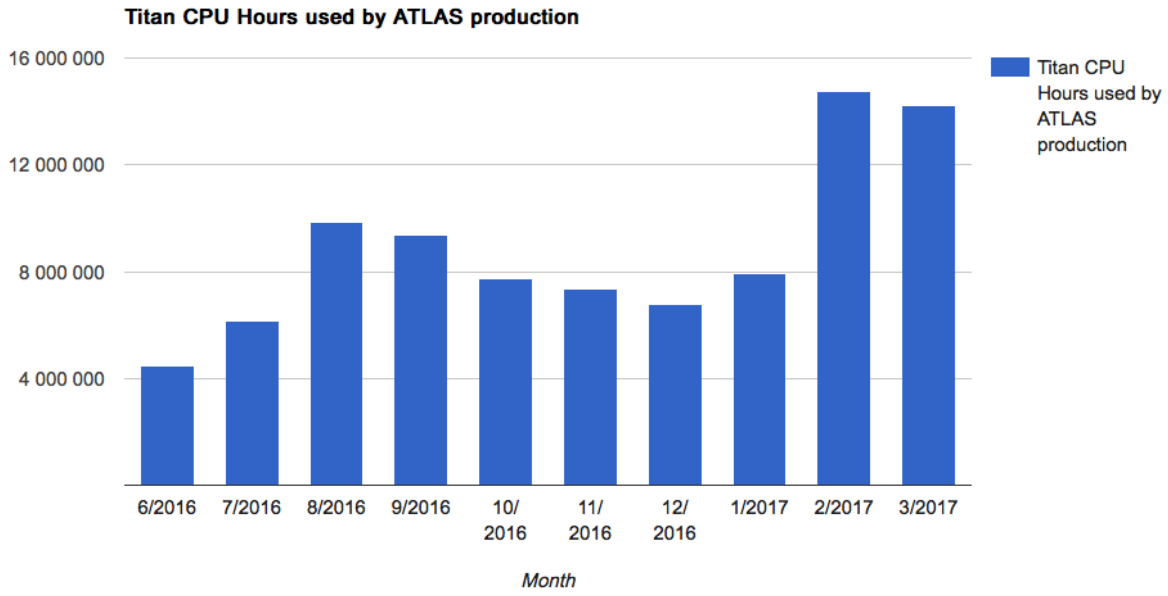



**Figure 1**. Increase in Titan usage by ATLAS PanDA during the first year of this project.

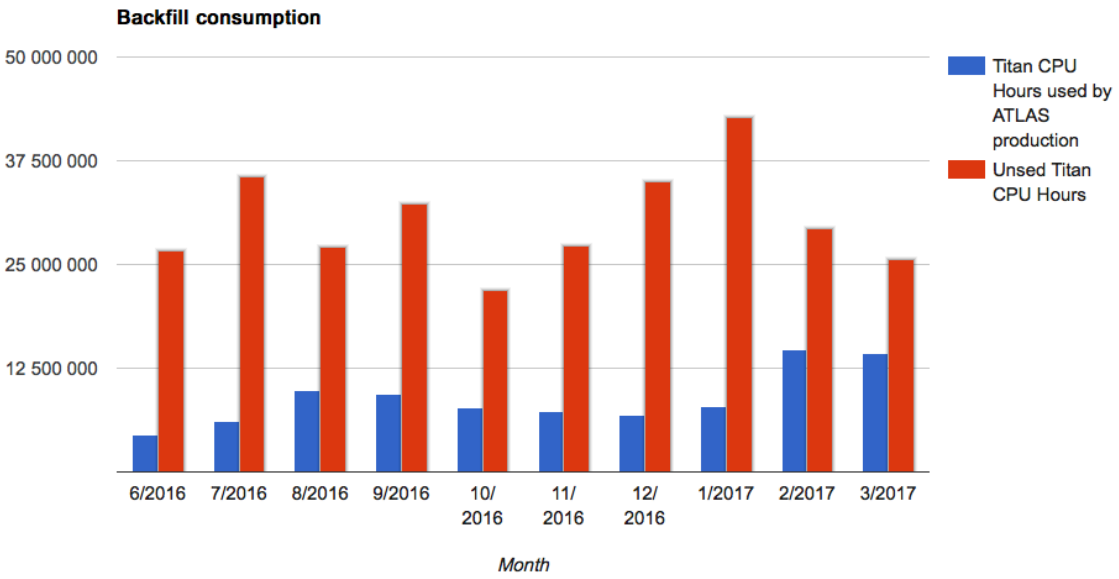



**Figure 2**. Increasing use of backfill opportunities by PanDA during the first year of this project.

In the followings sections we describe the progresses made in this project during several phases in the first year.

*June 2016 - July 2016*

Increased backfill utilization by increasing the number of Pilots, through changes made both to the pilot launcher and Titan policy. Gradually increased from 2 slots to 10 slots used simultaneously in Titan scheduler's queue. Initially, data movement was serial with workload execution. New pilots were not launched even if slots were available while data transfers were in progress. To improve efficiency, we implemented new service for launching Pilots on Titan. This service allows to launch new Pilot as soon as slot in local queue becomes available, properly taking into account those pilots which are performing stage-out of produced data. We also extended monitoring information to measure efficiency of usage (number of used and available nodes / number of available slots). Reached efficient utilization of ~15% of available backfill.

*August 2016 - October 2016*

New service for launching upto 20 pilots evenly distributed between 4 DTN (Data Transfer Nodes). Increased efficiency to ~26% of available backfill. Started discussions about improvements of operation mode of ATLAS at OLCF. Pilots were modified to provide more accurate data about walltime and cputime consumption for PanDA monitoring. Initial architecture of next generation of software named 'Harvester' was presented.

*November - December 2016, January 2017*

Slightly changed policy for assigning computational tasks (set of jobs for processing particular physics process) on ATLAS. First attempt at simultaneous execution of several tasks simultaneously on Titan. Efficiency of usage of backfill decreased to 21% in November, about 16% in December and January.

*February 2017 - March 2017*

New DTN cluster deployed at OLCF. More powerful hardware, new OS (RHEL7), new middleware for data transfers. Pilot's state model was updated, to propagate more accurate information about used resources to PanDA monitoring. Decreased the number of failed jobs. Significantly increased efficiency of backfill consumption: 33.5% in February, 35.7% in March. Active development of Harvester.

Overall, for the period from June 2016 to March 2017 about 2 millions of ATLAS Geant4 simulation jobs were successfully executed at OLCF, which corresponds to ~200M simulated physics events. Titan became one of the top facilities for simulation of physics processes for ATLAS. A grand total of 88,604,734 Titan CPU hours were consumed.

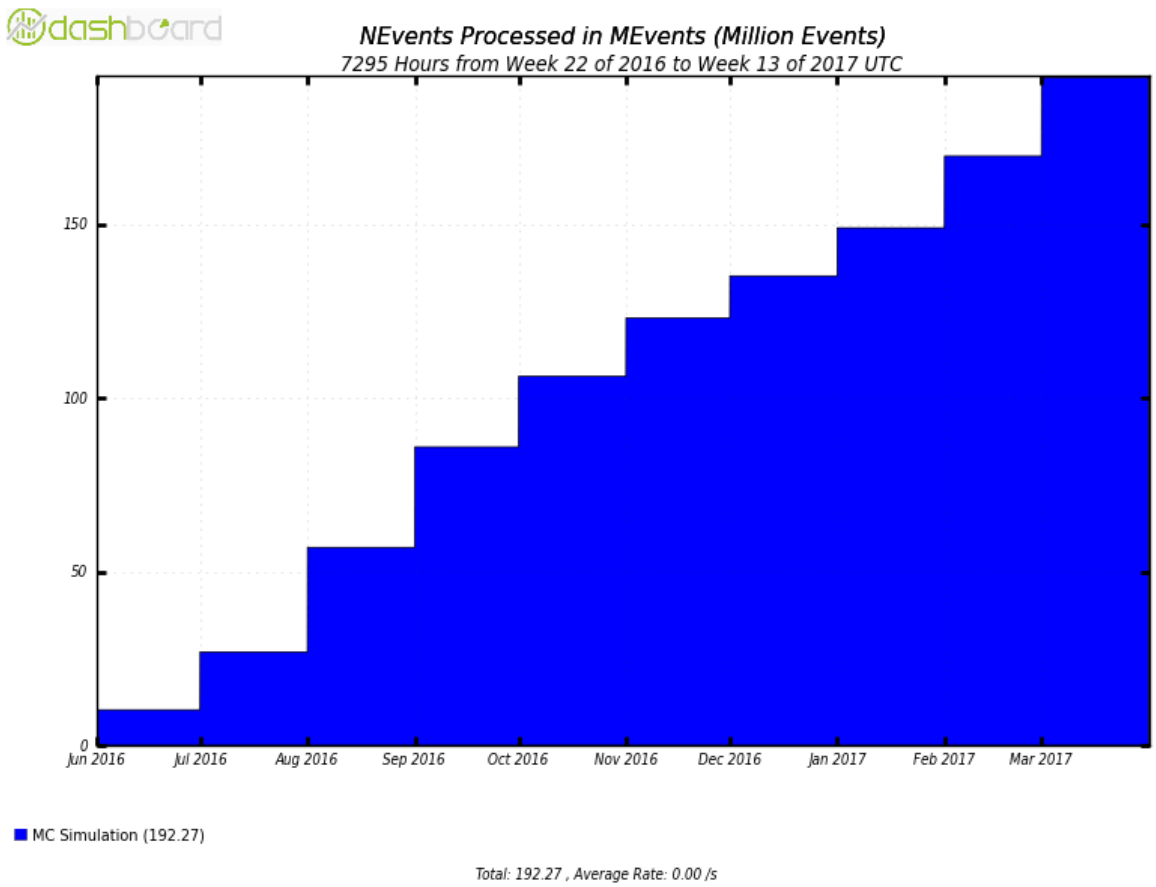



**Figure 3**.

## BigPanDA and the BlueBrain Project

The success of the projects to use PanDA beyond HEP and traditional HTC resources has drawn attention from other compute intensive sciences such as biology. The EU funded BlueBrain Project (BBP) received a sufficient allocation on LCF machines in US. BBP management contacted the BigPanDA team at the Supercomputing Conference in November 2016 and suggested to launch a common pilot project to demonstrate how BigPanDA WMS can be used for BBP scientific applications. BigPanDA Management team had follow-up meetings and videocons with BBP management, and in February 2017 the pilot project between BigPanDA and Blue Brain Project (BBP) (Swiss Federal Institute of Technology in Lausanne) teams was started. The duration of the pilot project is 6 months. As a proof of concept, the pilot project aims to demonstrate efficient application of the PanDA software for the supercomputer-based reconstructions and simulations, offering a radically new approach for understanding the multilevel structure and function of the brain.

The goal of this project is to run BBP SW on a distributed computing environment powered by PanDA. The targeted systems for demonstration are: x86 BBP clusters located in Geneva and Lugano (Switzerland), the Titan Supercomputer operated by OLCF and Cloud based

resources (Amazon). We have installed a PanDA server instance in Geneva. MPI test jobs have been successfully submitted to the x86 BBP clusters in Geneva (14 nodes) and Lugano (36 nodes) and IBM BlueGene/Q supercomputer with 65,536 cores hosted by the Swiss National Supercomputing Center (CSCS) in Lugano.

# Track 2: Data Transfer and Access

## Workload run-time optimization

Several issues identified in the proposal for workload run-time performance were addressed early on. Relocation of ATLAS software from Luster file system to NSF improved startup time of the Athena framework to the values normally observed on Grid. Software delivery from NFS was tested in runs up to 6000 nodes per job and showed good scalability. A study of the IO patterns of ATLAS jobs during steady state operations confirmed (see track 4 report) that switch to NFS brought IO load on Luster from ATLAS simulations well within operational parameters.

## Pilot launcher optimization

Analysis of data transfers of PanDA pilots to and from Titan identified a bottleneck that prevented launch of the new pilots before currently running pilots complete data stage out operations. New logic was implemented into pilot launcher to remove this bottleneck. This allowed us to increase backfill utilization by a factor of two to about 7 million hours per month and backfill utilization efficiency to ~19%.

## Move of pilot operations to Data Transfer nodes

When PanDA edge services operations on Titan were moved from interactive login nodes to the (old) Data Transfer Nodes (DTN) it was observed that at peak number of pilots the CPU load during data transfer operations was close to maximum. That posed a limit on the number of simultaneous running pilots on DTNs and number of running jobs. Also data transfer rates after the move to DTNs did not change compared to login nodes. Consequently no improvement in backfill utilization was observed after the move. When a new DTN farm became available at OLCF in February 2017, PanDA edge services were moved there. The new hardware and updated data transfer software (GFAL) provided noticeable improvement in data transfer rates and allowed to increase backfill utilization to about 14M core hours. Backfill utilization efficiency reached ~30%.

A study of CPU loads on the new DTN was performed shortly after the move to the new DTN farm. The study was based on the information from system level monitoring logs (SAR). Average CPU load on the DTNs used for PanDA operations varied from 1% to 6% over the period of observation. Maximum observed peak load was about 30%. The measurements were done over several days during regular ATLAS operations on Titan. During that period of time on average ~9k Titan cores per hour were used by PanDA, with maximum peak of ~70k cores and several peaks at ~30k cores. It was concluded that the new DTN farm provides enough CPU

headroom for the current scale of PanDA operations on Titan and will allow to expand these operations in the future.

We also performed analysis of the data transfer times on two generations of DTNs. In the current implementation of PanDA edge services on Titan, the time of input data transfer contributes to PanDA‘s “reaction time” from detection of available free resources to job submission to Titan’s queue and thus directly affects backfill utilization efficiency. The new DTNs provided about 30% improvement in input data transfers (average 8 sec. per file on the new DTN farm vs 12 sec. on the old DTN's) and more than factor of two improvement for output data transfers (on average 9 sec. on the new DTN nodes vs 25 sec on the old DTNs).

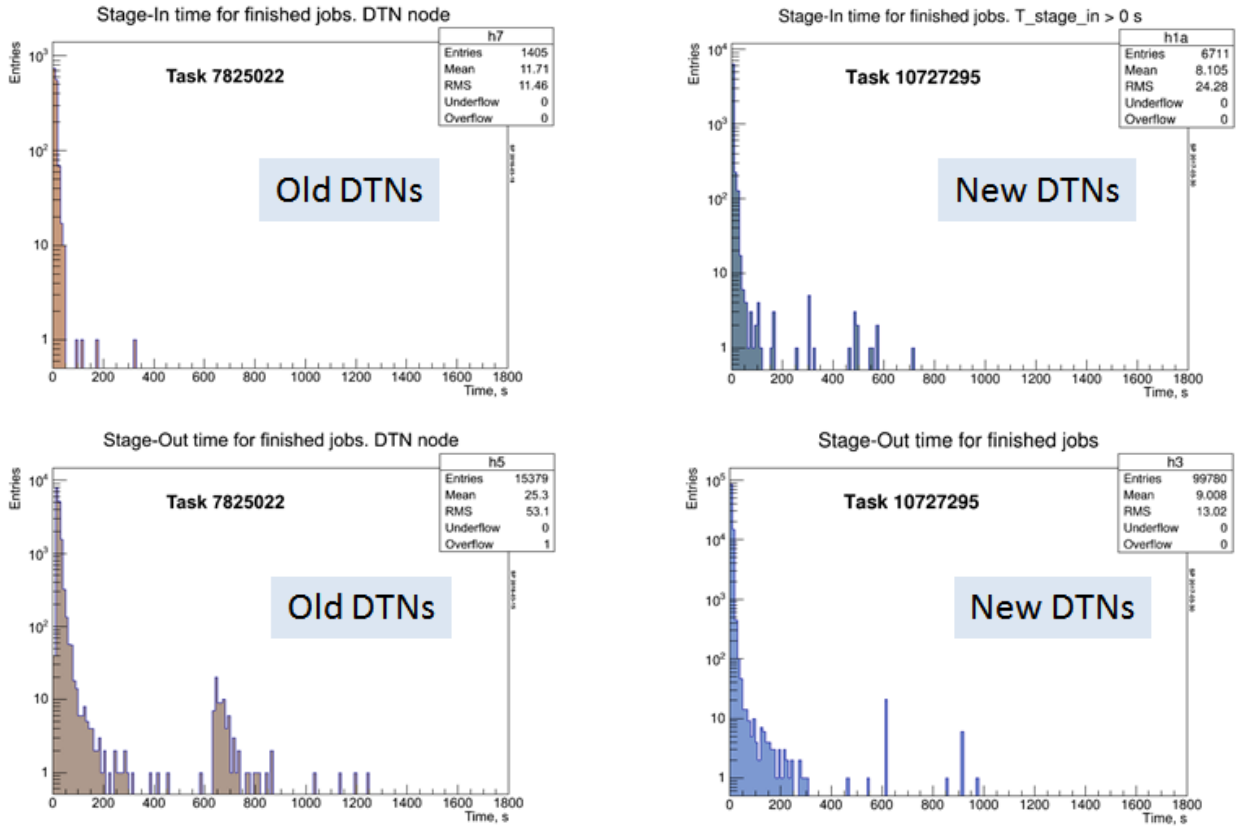



**Figure 4**.

## Track 3: Adaptive Execution of Next-generation Workloads

Like most infrastructure projects, PanDA had to grapple with the tension of “functional and reliable but internally developed” capabilities with that of “externally developed but with a functional impedance” gap. Due to the global state of the science of workflows and infrastructure, there have been no simple answers, and PanDA has historically taken a pragmatic approach to the problem. The aim of Research Track 3 is to investigate whether the use of abstractions and models can help improve PanDA’s utilization of infrastructure without waiting for global solutions and science of workflows and cyberinfrastructure to emerge. The

primary measures of improvement include, (i) the ability to support a wider range of execution modes for existing workload (AthenaMP/Geant4), (ii) support new (different) workloads, and (iii) provide models and a common conceptual basis to evaluate the effectiveness. These must be achieved while respecting an important constraint that only surgical and minimally invasive changes to production grade PanDA are practical.

In response to the objectives outlined above, we developed a prototype of a pilot system capable of executing on Titan to study experimentally the quantitative and qualitative benefits that it could bring to PanDA. We called this prototype Next Generation Executor (NGE). NGE is a runtime system to execute heterogeneous and dynamically determined tasks that constitute workloads. The figure below illustrates its current architecture as deployed on Titan: the two management modules (Pilot and Unit Managers) represent a simplified version of the PanDA Pilot while the agent module is the pilot submitted to Titan and executed on its worker nodes. The communication between PanDA Pilot and Server is abstracted away as it is not immediately useful to evaluate the performance and capabilities of a pilot on Titan.

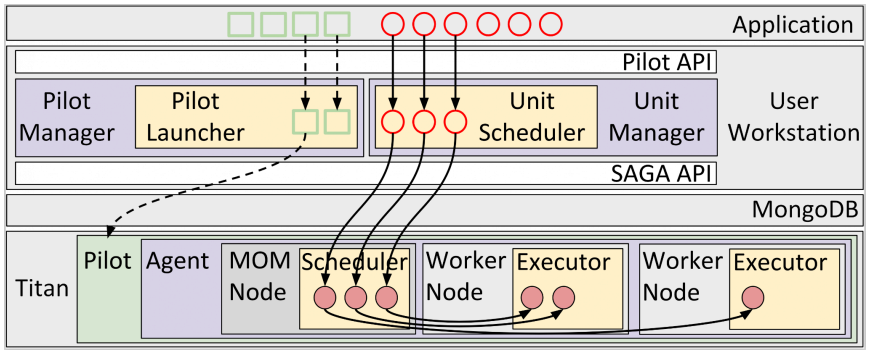



**Figure 5**.

In order to reach the desired scales while providing a level of interoperability, the NGE Agent relies on the Open Run-Time Environment (ORTE) for communication and coordination of the execution of units. For a more detailed description of NGE and ORTE we refer the reader to [1][2] respectively.

## ATLAS workloads with NGE

On Titan, PanDA Pilot is deployed on Data Transfer Nodes because Titan's worker nodes lack connectivity to the wide area network. This makes PanDA Pilot act as a resource broker, impacting both the efficiency and the flexibility of PanDA's execution process. Pilot capabilities could improve efficiency by increasing throughput and enabling greater backfill utilization. Further, pilot capabilities would make it easier to support heterogeneous workloads.

The absence of pilot capabilities imposes the static coupling between MPI scripts submitted to Titan's PBS batch system and detector simulations. This makes the scheduling of multiple generations of workload on the same PBS job impossible: once a statically defined number of detector simulations are packaged into a PBS job and this job is queued on Titan, no further

simulations can be added to that job. New simulations have to be packaged into a new PBS job that needs to be submitted to Titan based upon backfill availability.

The support of multiple generations of workload would enable more efficient use of Titan's backfill availability of walltime. Currently, when a set of simulations ends, the PBS job also ends, independent of whether more wall-time would still be available. With pilot capabilities, additional simulations could be executed to utilize all the available wall-time, while avoiding further job packaging and submission overheads.

Multiple generations would also relax two assumptions of the current execution model of the Titan's deployment of PanDA Pilot: (i) knowing the number of simulations before submitting the MPI script; and (ii) having a fixed number of events per simulation (100 at the moment). Pilot capabilities would enable the scheduling of simulations independently from whether they were available at the moment of submitting the pilot job. Further, simulations with a varying number of events could be scheduled on a pilot job, depending on the amount of remaining walltime and the distribution of execution time per event. These capabilities would increase the efficiency of the Titan's deployment of PanDA Pilot, especially when there is a large difference between the number of cores and walltime.

Pilot capabilities can offer a payload-independent scheduling interface while hiding the mechanics of coordination and communication among multiple worker nodes. This could eliminate the need for packaging payload into MPI scripts within each PanDA Pilot, greatly simplifying the submission process. This simplification would also enable the submission of different types of payload, without having to develop a specific PBS script for each payload. The submission process would also be MPI-independent, as MPI is used for coordination among multiple worker nodes, not by the payload.

### *Experimental Results with ATLAS Workload and NGE*

NGE provides functional enhancements to execution; but what is the price of these enhancement? In order to measure the overheads as a function of scale, we designed experiments to characterize the performance of the NGE on Titan. We perform three groups of experiments in which we investigate the weak scalability, weak scalability with multiple generation, and strong scalability of the NGE.

Each experiment entails executing multiple instances of AthenaMP to simulate a pre-determined number of events. All the experiments have been performed by configuring AthenaMP to use all the 16 cores of Titan's worker nodes.

In this experiment we run as many AthenaMP instances (hereafter referred to as tasks) as the number of nodes controlled by the pilot. Each AthenaMP simulates 100 events, requiring ~4200 seconds on average.

Tasks do not wait within the NGE Agent's queue since one node is available to each AthenaMP instance. Overheads in task execution are consequence primarily of the three other factors: (i) the initial bootstrapping of the pilot on the nodes; (ii) the UnitManager's dispatching of units (tasks) to the agent; and (iii) time for the agent to bootstrap all the tasks on the nodes.

We tested pilots with 250, 500, 1000 and 2000 worker nodes and 2 hours walltime. The time duration is determined by the Titan's walltime policy. The Figure on l.h.s. at the end of the

section depicts the average pilot duration, the average execution time of AthenaMP, and the NGE pilot overhead as function of the pilot size.

We observe that, despite some fluctuations due to external factors (e.g., Titan's shared filesystem and the shared database used by the NGE), the average execution time of AthenaMP ranges between 4200 and 4800 seconds. We also observe that in all the cases the gap between AthenaMP execution times and the pilot durations is minimal, although it slightly increases with the pilot size. We notice that NGE's overhead does grow linearly with the number of units.

The NGE provides an important new capability of submitting multiple generations of AthenaMP to the same pilot. In order to investigate the cost of doing so, we performed a variant of the weak scalability experiments. This stresses the pilot's components, as new tasks are scheduled for execution on the Agent while other tasks are still running.

In these experiments, we run five AthenaMP instances per node. As these experiments are designed to investigate the overhead generated by the scheduling and bootstrap of AthenaMP instances, we reduced the number of events simulated by each AthenaMP task to sixteen in such a way that the running time of each AthenaMP is, on average, ~1200 seconds. This experiment design choice does not affect the objectives or accuracy of the experiments, but allows us to scale experiments to large node counts by being conservative with allocation.

We ran pilots with 256, 512, 1024 and 2048 worker nodes and 3 hours walltime. The central figure at the end of the section depicts the average pilot duration, the average execution time of five sequential generations of AthenaMP, and the corresponding overhead. We observe that the difference between the two durations is more marked than in the previous experiments. Despite this, we notice that the growth of the overhead is consistent with the increment of the number of tasks per node for pilots with 256, 512 and 1024 worker nodes, and less than linear for the pilot with 2048 worker nodes.

The last experiments study strong scalability by running the same number of tasks for different pilot sizes. We used 2048 AthenaMP instances and pilots with 256, 512, 1024 and 2048 nodes. Thus, the number of AthenaMP generations is equal to eight times the size of the smallest pilot and corresponds to the size of the largest pilot. As a consequence, the number of consecutive generations of AthenaMP decreases with the pilot size by generating different dynamics within the pilots. These experiments are designed to investigate whether pilot overhead is affected by the degree of concurrency within the pilot and/or the number of tasks. Each AthenaMP instance simulates sixteen events as in the previous experiment.

The figure on r.h.s. at the end of the section shows the average pilot duration and the average execution time of possibly sequential AthenaMP instances. We notice that the difference between the pilot duration and the AthenaMP execution times is almost constant for all the pilot sizes, although the overall duration of the pilot decreases linearly with the pilot size.

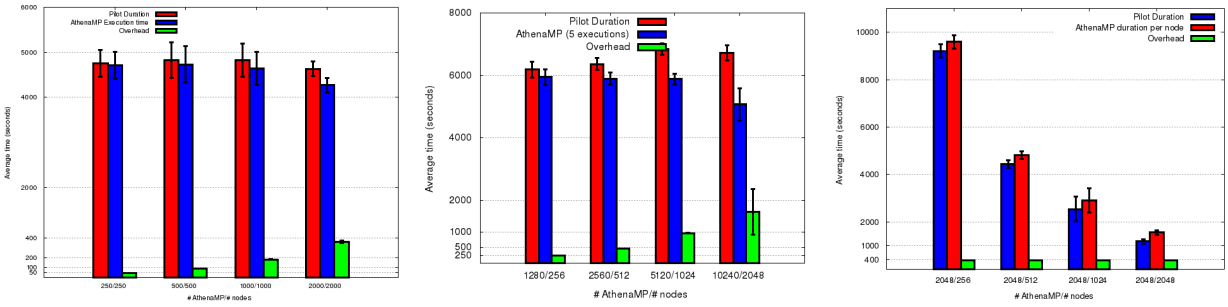



**Figure 6**.

## Molecular Dynamics workloads with NGE

Arguably no where is the importance of running ensembles of high-performance simulations greater than in biomolecular sciences. A range of novel algorithms and methods have emerged in the past few years which depend upon the ability to execute a large number of concurrent classical molecular dynamics (MD) simulations with varying levels of coupling between them. Further, the number of CPU cycles devoted to atomistic simulations is a significant fraction of the overall utilization of the leadership (and NSF) machines. Also, most (if not all) MD applications do not use workload management systems and resort to direct submission. Thus, we believe applications comprised of MD workloads are a promising representation of future workload systems that could benefit from enhancements to PanDA arising from NGE.

We performed experiments to determine the weak scalability and strong scalability by using the same procedure that has been used for the ATLAS workload, i.e., each experiment entails executing multiple instances of Gromacs -- a widely used MD simulation engine. We set Gromacs in such a way that it uses only one core of Titan's worker nodes. For the given physical system and physical duration for which it is simulated, the average execution time is ~1020 seconds on a single x86 core.

Weak scalability has been tested by running pilots on 8, 16 and 32 working nodes. Each pilot runs one Gromacs instance per core; thus, each run executes a number of tasks equal to 16 times the number of nodes.

Strong scalability has been tested by running 512 tasks on 8, 16 and 32 working nodes. As a consequence, the number of generations within the pilot decrease with the size of the pilot. Figures below show the result ( l.h.s. plot depicts weak scalability and r.h.s. plot depicts strong scalability).

In reference to weak scalability, It is possible to observe that the overhead grows linearly with the number of tasks whereas strong scalability shows that the overhead is constant for all the pilot sizes whereas the execution time and the pilot duration decrease.

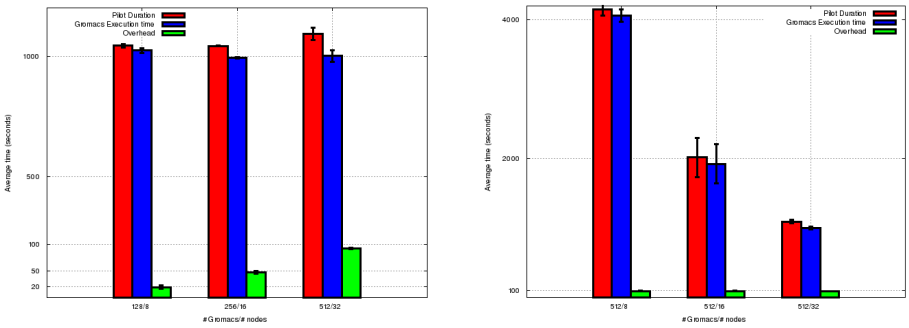



**Figure 7**.

## Future Work

The integration of PanDA WMS with NGE offers the opportunity for a principled understanding of how to integrate middleware developed within different projects. This is a problem that besets the field of scientific distributed computing, often addressed by endless duplication of development effort. We plan to expand upon this “building block” approach [3], by designing an interface between PanDA and NGE based on a well-defined set of abstractions.

Currently, we are evaluating alternative designs, all based on the abstractions of resource capabilities, availability and capacity. The idea is for the NGE to expose information about type, size and duration of resources it can provide, abstracting away the details and processes of resource selection, acquisition, and management. In this way, PanDA would see NGE as yet another resource, not dissimilar to a grid resource or Titan itself. From the NGE point of view, PanDA would be yet another workload provider, i.e., a system that submits either a bulk or a stream of tasks for execution.

### *Introspection through profiling*

In the next year, we plan to further increase the introspection capabilities of NGE to enable performance profiling and optimization. Specifically, we will focus on enabling performance measurements for both the NGE middleware and the task executed by it. This will require the evaluation of existing performance profilers and their suitability to be used with standalone kernel applications and with a distributed system as NGE. The following two requirements will be particularly important: (i) the profiler has to work with both compiler and interpreted languages; (ii) the profiler does not have to require recompilation and/or tailoring coding inside the application.

The use of profilers such as Tau or SCORE-p will enable the understanding of NGE overhead as well as the interaction between middleware, application executable, and infrastructure. This is particularly relevant when considering the high level of performance customization supported by Titan’s and then Summit’s operations. For example, enabling the profiling of file system overheads, mpi overheads, or communication and coordination gives ORNL the opportunity to customize their support of specific classes of applications. Modern HPC systems have not been explicitly designed to support and optimize the execution of large amount of short-lived, I/O-intensive, single-core tasks. However such workloads are increasingly important. The insight

offered by our profiling will be of paramount importance to isolate relevant bottlenecks and opportunities for tailored configuration at resource level.

### *NGE integration with Fireworks*

To establish the broad impact of using NGE to improve execution capabilities of a range of tools, Fireworks is a widely used workflow system that has been developed to support the materials genome project. Fireworks currently support a large number of electronic structure calculations (such as VASP) but it has limits in terms of bootstrapping, scalability and resource utilizations.

For this reason, we are investigating its integration with NGE together with the Fireworks team. The main challenge is to perform the integration in such a way that it is transparent for the final users.

After considering the pros and cons of several possible integration strategies, we decided that the use of the NGE as execution layer for Fireworks fits the requirements in terms of scalability, sustainability and reuse of the software. The scheme below provides a sketch of the current and the future execution mode of Fireworks.

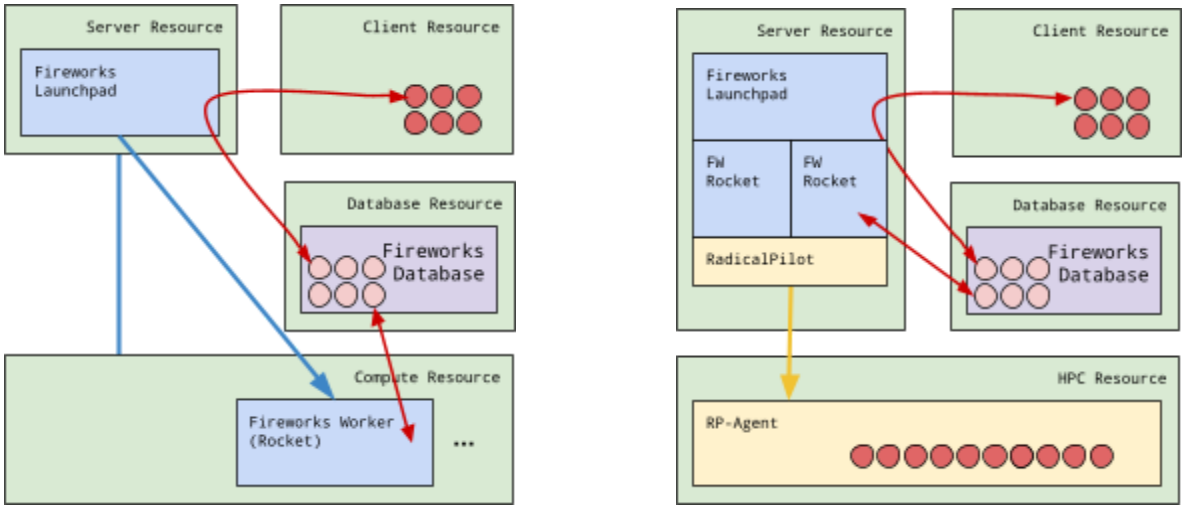



**Figure 8**.

## Lessons Learned

In principle, the time required to compute the same task should be independent of which working node of Titan is used, except for marginal noise that is always present due to the shared file-system, network and measurement errors. However, our experiments have put on the spotlight on fluctuations that in some cases reached 100% of the average execution time. Figures below provide a quick glimpse of this phenomenon by reporting all the execution times that we collected during the experimental campaign.

The three plots on the top depict the execution times of ATLAS workload (from left to right weak scalability, weak scalability with multiple generations and strong scalability) whereas the two plots on the bottom correspond to the execution times of Molecular dynamics workload (weak and strong scalability).

Each dot on the plots below corresponds to the execution time of a run. Runs are aligned temporally according to the starting time of each run. Colors are used to discriminate between different pilot sizes (inter pilot independence).

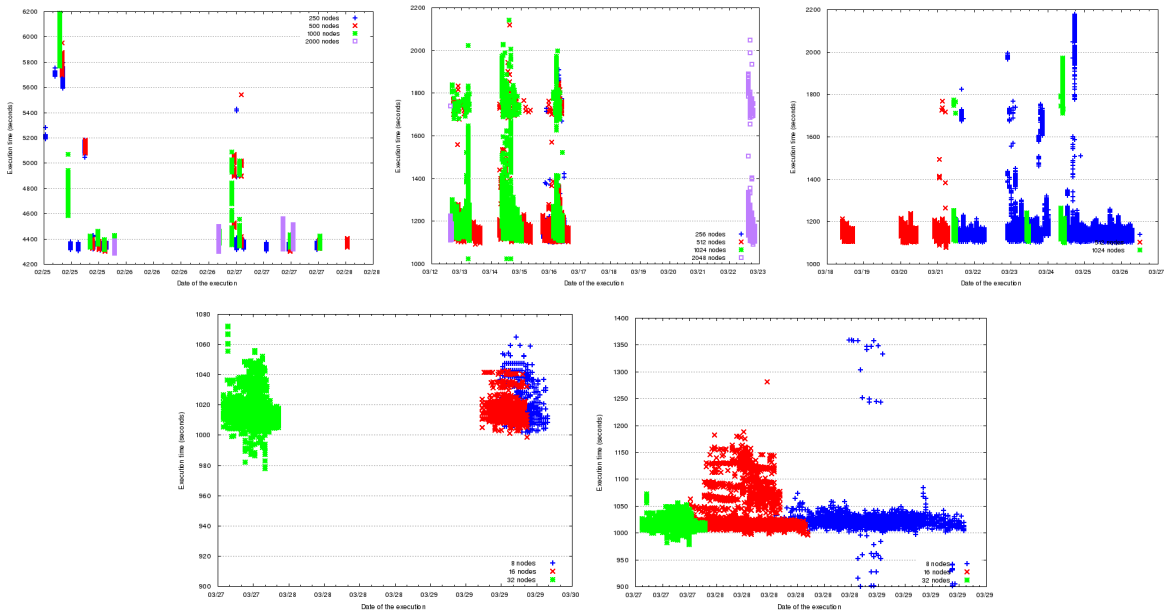



**Figure 9**.

We observe that fluctuations of the same magnitude can be found on different pilot sizes and different workload. This suggest that fluctuations are not caused by the size of the pilot or by the specificity of the workload. Observing the plots, we notice that tasks that start together tend to have similar execution times. This suggest that the status of the system at a given time might play a role in the fluctuations but also that the working nodes on which the pilot runs might affect the execution of tasks (intra pilot dependence).

Insights gained from research track 3 will ultimately aid the design of middleware that supports the scalable execution of thousands/millions of concurrent tasks and insulates tasks from each other.

# Track 4: Operations and Operational Model

## Examining the I/O impact of PanDA@OLCF

Initially, all the shared libraries of AthenaMP and the python scripts for the configuration stage were stored on OLCF's Spider 2 file system, using the Lustre technology. However, the I/O patterns of the initialization and configuration stages degraded the performance of the

filesystem, affecting the overall performance for all OLCF users. This was addressed by moving the AthenaMP distribution to a read-only NFS directory, shared among data transfer nodes and worker nodes. This NFS implementation eliminated the problem of metadata contention, improving metadata read performance from ~6,300 seconds on Lustre to ~1,500 seconds on NFS.

We also analyzed the file and metadata I/O impact of ATLAS on OLCF systems. ATLAS used between 1 and 300 worker nodes, and 35 on average. 75% of the jobs run by ATLAS consumed less than 25 nodes and 92% less than 100. During the 174 hours of data collection, 6.75 ATLAS jobs were executed on average per hour, each job running for an average of 1.74 hours.

Every job read less than 250 GB and wrote less than 75 GB of data and, on average, each job read 20 GB and wrote 6 GB of data. ATLAS jobs are read intensive: on average, the amount of data read per worker node is less than 400 MB, while the amount of data written is less than 170 MB. Distributions of read and written data are different: The read operation distribution per job shows a long tail, ranging from 12.5 GB to 250 GB, while the written amount of data has a very narrow distribution.

The metadata I/O breakdown shows that ATLAS jobs yield 23 file open() operations per second (not including file stat() operations) and 5 file close() operations per second, with similar distributions. On average, the maximum number of file $open()$ operations per job is ~170/s and the maximum number of file close() operations is ~39/s. For the 1,175 ATLAS jobs observed, the total number of file open() operations is 172,089,760 and the total number of file close() operations is 40,132,992. The difference between these two values is still under investigation: One possible explanation is that ATLAS jobs don't call a file $close()$ operation per every file open() issued.

Overall, the average time taken to read events from input files stored on Lustre is 1,320, comparable to the time taken to read the file required by assembling AthenaMP from NFS. Preliminary investigation shows that this time could be reduced to 40 seconds by loading the event descriptions into the RAM disk available on each worker node. Event descriptions could be transferred from Lustre to the RAM disk while configuring and initializing AthenaMP, almost halving the time currently required by initiating a Geant4 simulation.

Based on our experiments with ATLAS jobs, it can be safely concluded that the file and metadata I/O load of ATLAS PanDA project on the OLCF Titan supercomputing environment and the Spider 2 file system is not detrimental to the center operations. At the current scale of the project, the overall impact of ATLAS operations on Titan is minimal.

## PanDA instance installation at ORNL

In March 2017, we implemented a new PanDA server instance within ORNL in order to serve various experiments at OLCF. To streamline the deployment procedure and provide possibility to manage service on high level without experience of PanDA development, a powerful container cluster management and orchestration system, Red Hat OpenShift Origin [1], was proposed and implemented. Origin is a complete container application platform that natively integrates technologies like Docker [2] and Kubernetes [3]. And its implementation within the OLCF

operations infrastructure inspired the accelerated installation of the PanDA instance in PY1 of this project.

The PanDA Docker container was built and pushed into the OpenShift Origin system deployed at ORNL. Container includes PanDA server, client and server DB backend using MariaDB DBMS. The PanDA operation within a container was verified in standalone mode by successful submission of the test job and it's registration on server. Currently we are working on network configuration within Origin system and the next step will be complete PanDA service testing including the job submission, obtaining the job by the pilot and payload execution on Titan supercomputer.

# Plans for Year 2 (PY2)

By the end of the second year of the project we will have a full scale PanDA workflow and data management demonstrator for compute-intensive scientific applications in DOE energy and intensity frontier programs, bioinformatics and MD, as well as an early Summit operational model demonstrator. The evaluation of the Adaptive Execution strategies will be sufficiently mature to enable OLCF to evaluate the requirements of various user program models.

By the end of the second year, a full scale scientific demonstrator from ATLAS and ALICE will run on OLCF at the scale of production needs of both experiments integrated with Titan monitoring and jobs scheduling. This will pave the way for future HENP computing model and routine LCFs usage by the LHC and HENP experiments for the LHC Run3 (2021-2023) and HL-LHC era.

We will also demonstrate that this model works for user communities outside of HENP, the demonstration will be done for a real use-case from biophysics and bioinformatics, to astro-particle physics, and the Intensive Frontier.

The project results will be presented at International Scientific Conferences and published in peer-reviewed journal.

## Track 1: Adaptation of BigPanDA WMS for OLCF

- ATLAS is going to migrate to a new generation of PanDA software, which will allow us to optimize the workflow of execution of PanDA jobs on HPC, providing better management of data transfers, and increasing the usage of ''backfill" resources.
- PanDA server installation at ORNL will be completed and the validation of the deployment procedure within the OpenShift Origin system has to be done. Finally PanDA instance has to be moved to production for payloads launching on Titan Supercomputer.
- The goal of the joint project between BigPanDA and BlueBrainProject working groups is to support the execution of BBP software on a variety of distributed computing systems powered by PanDA during PY2. The targeted systems for demonstration include: Intel x86-NVIDIA GPU based BBP clusters located in Geneva and Lugano, BBP IBM BlueGene/Q supercomputer located in Lugano, the Titan Supercomputer and Cloud based resources such as Amazon Cloud.

## Track 2: Data Transfer and Access

- Further improvements in data transfer operations are expected from a new edge service for PanDA called Harvester which is under development by the PanDA team. Harvester should allow asynchronous parallel data transfers that are decoupled from job submission and that should lead to further increase in the backfill utilization on Titan. Increased number of threads performing concurrent data transfers will need careful monitoring of loads on DTNs in order to determine optimal operational conditions.
- Deployment of the ATLAS Event Service on Titan and corresponding shift in I/O operations to a significantly larger number of smaller output files will require a study of an impact on the Luster shared filesystem.
- Integration of Harvester with NGE will also require additional studies of I/O operations on the shared file system.
- We plan to explore use of RAM disk on Titan's worker nodes in order to reduce I/O loads on Luster during ATLAS operations.

## Track 3: Adaptive Execution of Next-generation Workloads

As discussed under Research Track 3, specific future work items include:

1. Deeper integration between the NGE and PanDA WMS.
2. Horizontal integration with other high-performance applications (biomolecular simulations) and tools (Fireworks) used by other communities.
3. Enabling, in conjunction with "execution strategies," empirical answers to "optimal" utilization of backfill versus regular queues. The technological capability and modeling (e.g., distribution of job sizes) will allow adaptive execution strategies.

The cumulative effect of these future work items along with ongoing efforts such understand performance fluctuations will be to provide a formal basis upon which to determine the relative merits of different execution strategies.

## Track 4: Operations and Operational Model

- In PY2, we will demonstrate PanDA@OLCF operations for three non-HENP applications, using the PanDA instance installed at OLCF.
- In PY2, we will extend our monitoring and analysis of the I/O impact of PanDA workload, initially with the ATLAS Athena/MP application currently in production use. We will also perform a second detailed analysis of IO impacts with one of our new non-HENP applications, including the use of Darshan, a scalable HPC I/O characterization tool. Darshan is designed to capture an accurate picture of application I/O behavior, including properties such as patterns of access within files, with minimum overhead. But it does require recompilation of the application.

- In the second half of PY 2, we will perform initial porting of the PanDA WLMS on Summitdev[1] at OLCF. The Summitdev system is an early access system that is one generation removed from OLCF's next big supercomputer, Summit. Summitdev is operational now for Summit application development projects for OLCF-4 and the Exascale Computing Project (ECP). As these projects transition to Summit in early CY2018, there will be opportunity to develop PanDA for the new OpenPOWER architecture beginning in early CY 2018.

# Summary

Building upon the demonstrated capability to utilize compute cycles that would be otherwise unusable and the ability to increase the overall utilization on Titan, in the first year of this project we have begun to harden, extend and generalize the ability of using HPC for HTC for ATLAS and other communities. This, in and of itself, show something of the potential science impact of these project.

The first year was used for early prototyping and implementation of production services:

- OLCF facilities are fully integrated with the ATLAS Production System.
- OLCF utilization was increased to 94% of usable cycles for ATLAS workload: this was done in steps to detect potential issues at early stage.
- The first prototyping for energy and intensity frontier experiments (ALICE, LSST, nEDM) and LQCD applications was conducted. More experiments (nEDM, IceCube) and projects (BlueBrain) have expressed interest to run their payloads on OLCF via PanDA. For ATLAS and ALICE compute-intensive tasks have been identified and executed, it allowed to increase precision of simulations needed by the experiments.
- File Transfer Service (FTS3) was integrated with PanDA and data were shipped from OLCF to BNL.
- Next Generation Workload Management System development, coding and prototyping is progressing well, the first prototype has been used to submit ATLAS payloads.
- Improved Titan monitoring has allowed to account precisely OLCF contribution to the LHC Computing program.
- PanDA instance installation is in progress at ORNL, with the goal to finish it before September 2017 and to use for scientific payloads submission on OLCF.
- Four talks were presented at International Conferences, three papers were published and two more are submitted. The first BigPanDA Technical Interchange Meeting was conducted at ORNL in March 2017, 5 experiments and projects have presented progress and plans with OLCF usage via PanDA.

[1] https://www.olcf.ornl.gov/kb_articles/summitdev-quickstart/

# Publications, Conference Talks, and Workshops 2016/2017

- K. De, S. Jha, A. Klimentov, T. Maeno, R. Mashinistov, P. Nilsson, A. Novikov, D. Oleynik, S. Panitkin, A. Poyda, K. F. Read, E. Ryabinkin, A. Teslyuk,V. Velikhov, J. C. Wells, and T. Wenaus. Integration of Panda Workload Management System with Supercomputers, ISSN 1547-4771, Physics of Particles and Nuclei Letters, 2016, Vol. 13, No. 5, pp. 647–653.
- K. De, A. Klimentov, T. Maeno, R. Mashinistov, P. Nilsson, D. Oleynik, S. Panitkin, T. Wenaus. Accelerating Science Impact through Big Data Workflow Management and Supercomputing, EPJ Web Conf. (108) (2016) 01003.
- K. De, S. Jha, A. Klimentov, T. Maeno, P. Nilsson, D. Oleynik, S. Panitkin, J. Wells and T. Wenaus. Integration Of PanDA Workload Management System With Supercomputers for ATLAS and Data Intensive Science. Journal of Physics: Conference Series, Volume 762, Number 1.
- BigPanDA project Technical Interchange Meeting. Oak-Ridge, Mar 2017. https://indico.cern.ch/event/618513/
- Supercomputing conference. SC16, Salt-Lake City, USA:
    - Presentations at UTA and DOE booths
    - Round table and technical discussion (BNL, ORNL, UTA, Rutgers, ANL, UCSD, Princeton, Google, OSG) . SC future for HENP applications
- The 22nd International Conference on Computing in High Energy and Nuclear Physics, CHEP 2016 October 2016, San Francisco, USA:
    - F. Barreiro Megino, K. De, S. Jha, A. Klimentov, T. Maeno, P. Nilsson, D. Oleynik, S. Padolski, S. Panitkin, J. Wells, T. Wenaus on behalf of the ATLAS Collaboration. Integration of Titan supercomputer at OLCF with ATLAS production system.
    - F. H. Barreiro Megino, K. De, A. Klimentov, T. Maeno, P. Nilsson, D. Oleynik, S. Padolski, S. Panitkin, T. Wenaus on behalf of ATLAS collaboration. PanDA for ATLAS Distributed Computing in the Next Decade.
- 38th International Conference on High Energy Physics (ICHEP), Chicago, USA, Aug 2016:
    - K. De, S. Jha, A.Klimentov, T. Maeno, P. Nilsson, D. Oleynik, S. Panitkin, J.C.Wells and T. Wenaus. PanDA Workload Management System on Supercomputers for ATLAS and Data Intensive Science.
- GRID 2016 , Dubna Russia, July 2016:
    - D. Oleynik, A. Klimentov, K. De, T. Wenaus, T. Maeno, P. Nilsson, F. Barreiro, W. Guan, S. Panitkin, S. Jha, J. Wells. Integration of PANDA Workload Management System with Supercomputers for ATLAS.

# Bibliography


[1] Angius A., Oleynik D., Panitkin S., Turilli M., De K., Klimentov A.,Oral S. H., Wells J. C., Jha S., Converging High-Throughput and High-Performance Computing: A Case Study, Submitted to SC'17.

[2] M. Santcroos, R. Castain, A. Merzky, I. Bethune, and S. Jha, "Executing dynamic heterogeneous workloads on blue waters with radical-pilot," in Cray User Group 2016. online: cug.org, 2016, pp. 1–14. [Online]. Available: https://cug.org/proceedings/cug2016 proceedings/includes/files/pap130.pdf

[3] Shantenu Jha and Matteo Turilli, "A Building Blocks Approach towards Domain Specific Workflow Systems?", https://arxiv.org/abs/1609.03484