

Accelerating Science Impact through Big Data Workflow Management and Supercomputing

K. De¹, A. Klimentov^{2,3}, T. Maeno², R. Mashinistov^{3,a}, P. Nilsson², D. Oleynik^{1,4}, S. Panitkin², E. Ryabinkin³, and T. Wenaus²

¹Physics Department, University of Texas at Arlington, 502 Yates Street, Arlington, TX 76019-0059, USA

²Physics Department, Brookhaven National Laboratory, Upton, Long Island, New York 11973, USA

³Kurchatov complex of NBIC-technologies, National Research Centre Kurchatov Institute, 1 Akademika Kurchatova pl., Moscow 123182, Russia

⁴Laboratory of Information Technologies, Joint Institute for Nuclear Research, Dubna, Moscow region, 141980, Russia

Abstract. The Large Hadron Collider (LHC), operating at the international CERN Laboratory in Geneva, Switzerland, is leading Big Data driven scientific explorations. ATLAS, one of the largest collaborations ever assembled in the the history of science, is at the forefront of research at the LHC. To address an unprecedented multi-petabyte data processing challenge, the ATLAS experiment is relying on a heterogeneous distributed computational infrastructure. To manage the workflow for all data processing on hundreds of data centers the PanDA (Production and Distributed Analysis) Workload Management System is used. An ambitious program to expand PanDA to all available computing resources, including opportunistic use of commercial and academic clouds and Leadership Computing Facilities (LCF), is realizing within BigPanDA and megaPanDA projects. These projects are now exploring how PanDA might be used for managing computing jobs that run on supercomputers including OLCF's Titan and NRC-KI HPC2. The main idea is to reuse, as much as possible, existing components of the PanDA system that are already deployed on the LHC Grid for analysis of physics data. The next generation of PanDA will allow many data-intensive sciences employing a variety of computing platforms to benefit from ATLAS experience and proven tools in highly scalable processing.

1 Introduction

The computing model of the ATLAS Experiment [1, 2] at the Large Hadron Collider (LHC) was originally based on the grid paradigm [3] with hierarchically distributed computing and storage resources. During Run-1 (2009–2013) at the LHC, ATLAS produced and processed over 130 PB of data. The computing model was proven to be very successful. However, a naive extrapolation of the Run 1 experience indicates that the computing resources needs in Run-2 (2015–2018) might increase by a factor 5–6, which will be impossible to manage with the current computing and storage resources within the anticipated flat computing budgets in the near future. Furthermore, while ATLAS currently uses more than 100,000 cores at well over 100 grid sites with a peak performance of 0.3 petaFLOPS,

^ae-mail: Ruslan.Mashinistov@cern.ch

Run-2 data processing will require more resources than grid computing can possibly provide. The Worldwide LHC Computing Grid (WLCG) infrastructure will be sufficient for the planned analysis and data processing, but it will be insufficient for Monte-Carlo (MC) production and any extra activities. Therefore additional computing and storage resources are required. To alleviate this challenge, ATLAS is engaged in an ambitious program to expand the current computing model to include additional resources such as the opportunistic use of supercomputers at Leadership Computing Facilities (LCF) as well as commercial and volunteer clouds.

The Production and Distributed Analysis (PanDA) system [4] was designed to meet ATLAS requirements for a data-driven workload management system for production and distributed analysis processing capable of operating at the LHC data processing scale. It has been used in the ATLAS experiment since 2005 and is now expanding into a BigPanDA and megaPanDA projects to extend PanDA as a meta application, providing location transparency of processing and data management for High Energy Physics (HEP) and other data-intensive sciences, and a wider exascale community. Initially PanDA has been used only on the facilities managed by WLCG. Support for Leadership Computing Facilities will expand the potential user community for PanDA and will also, even in the near term, benefit LHC experiments by running jobs on Leadership Computing Facilities with direct access to the data hosted by Grid centres, dynamically acquiring supplementary CPU resources when needed. Extending PanDA beyond the Grid and integrating distributed computing with cloud computing (or grids of clouds) will further expand the potential user community and the resources available to them, for example making it possible for non-LHC and non-HEP experiments to use PanDA [5].

2 Extending PanDA to Clouds

The ATLAS Cloud Computing project was set up a few years ago to exploit virtualization and clouds with a goal to utilize public and private clouds as extra computing resources and as an additional mechanism to cope with peak loads on the grid [6]. During the project many studies were performed, several cloud services were incorporated into the PanDA system and excellent progress was shown. Both commercial and private clouds have been studied.

In 2012–2013 PanDA was extended to support the Amazon Elastic Computing Cloud (EC2) [7] and other cloud resources, with prototype cloud-resident PanDA sites operating for Monte-Carlo production, data reconstruction and user analysis. PanDA jobs have run successfully in the Amazon cloud, Magellan cloud, the CERN Ixcloud and Google cloud. Also the academic cloud in Canada was used in ATLAS production and supported by PanDA [8, 9]. During February–May of 2013 the common project with Google was successfully completed. PanDA was successfully used to submit ATLAS simulation jobs and produce physics results using Google Compute Engine resources. The results were presented at the Google IO 2013 conference [10].

2.1 Workflow in Clouds

The PanDA system is using pilot jobs to execute the ATLAS payloads on the worker nodes. The pilot itself is launched on the worker node by a wrapper script that on the normal grid site is sent to the worker node from the pilot factory via the local batch system. On the cloud system an additional layer is needed since the pilot and the payload will be executed on the virtual machine (VM), which has to be started first. The chosen approach is to use Cloud Scheduler [11] for managing the VMs and HTCondor (High Throughput Computing) [12] for scheduling the jobs is shown in figure 1.

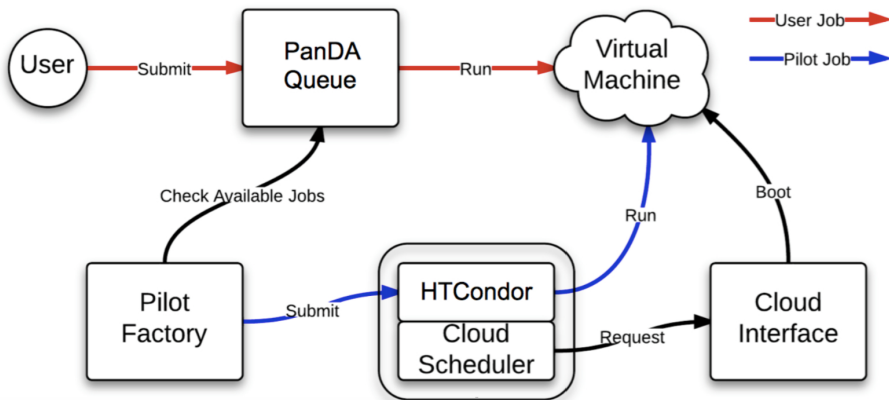


Figure 1. Job workflow in the cloud

2.2 Google Compute Engine Project

ATLAS was invited to participate in the closed preview of the Google Compute Engine (GCE) Project in August 2012. After several months of initial preparations, Google allocated 5M core hours on 4,000 cores to ATLAS for two months in March-April of 2013. The goal of the project was to test long term stability while running on the cloud cluster similar in size to a Tier-2 grid site in ATLAS. The resources were organized as the HTCondor based PanDA queue and transparently included in the ATLAS computational grid.

The CPU intensive workloads, such as physics event generators, fast detector simulations and full detector simulations were run in the GCE during eight weeks. Totally 458,000 jobs were completed and about 214M events were generated and processed as shown in figure 2. All output data was transferred automatically to the ATLAS grid storage. The project shows very stable running on the GCE side. The overall failure rate was about 6% where most of the problems were on the ATLAS side and not related to the cloud.

2.3 Amazon Elastic Compute Cloud

The Amazon Elastic Compute Cloud spot market offers an interesting alternative to otherwise expensive commercial clouds. The EC2 spot pricing, which trades a willingness to accept abrupt VM terminations for a significantly reduced cost, has turned IaaS (Infrastructure as a Service) into an economical alternative to site-based dedicated resources. The RHIC (Relativistic Heavy Ion Collider) and ATLAS Computing Facility (RACF) group at Brookhaven National Laboratory (BNL) received a grant allocation from Amazon EC2 in 2013. A hybrid cloud was set up and included resources from the BNL Tier-1 and the “elastic” part of the cloud on Amazon EC2, spanning geographically distributed EC2 sites (US-East, US-West1 and US-West2 zones).

Standard ATLAS production simulation jobs, with high CPU and low I/O usage, were run on 5,000 VMs in the large virtual cluster for nearly three weeks. Operations on the EC2 platform were found to be very reliable but the job efficiency was poor due to long running jobs.

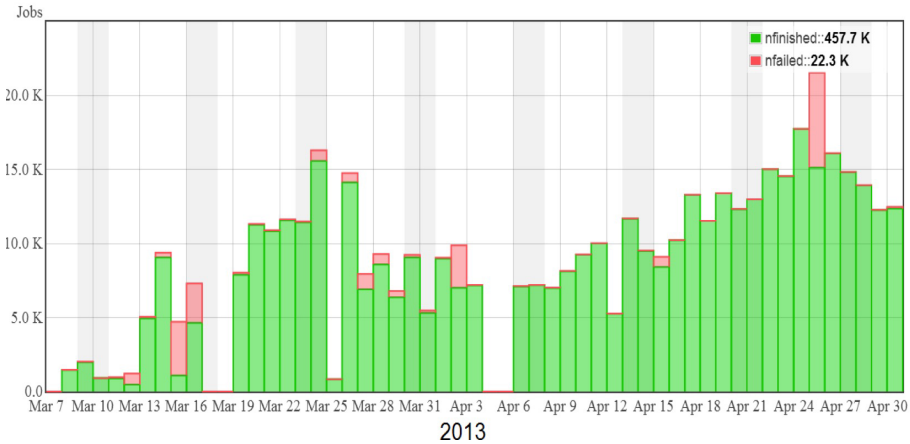


Figure 2. ATLAS jobs running in GCE

3 Extending PanDA to Supercomputers

The modern High Performance Computing (HPC) involves a very large number of cores connected through a high-speed network. On the HPCs, a typical job is highly parallel and each core calculates a small part of the problem and coordinates the activity via the processor interconnects. While this is different from the Grid paradigm it still shares common features such as the use of parallelization. It is not a requirement that HPCs are able to run any possible task, nor is it relevant how many kinds of job types can be run. What matters is the total number of cycles that can be offloaded from the grid.

Standard ATLAS workflow is not well adapted for HPCs due to several complications such as fixed worker node setups, no outbound network connections, no local disks per worker node, limited memory, specialized operating systems and non-Scientific Linux (SL6) environments, binary incompatibilities with ATLAS software releases, etc. A reorganization of the standard workflow is therefore needed.

The following is a non-exhaustive list of some of the known problems with suggested solutions. Communication with PanDA server typically is not possible on the worker node level. Hence the payload must be fully defined in advance, and all communications with the PanDA server will be done from the HPC front-end node where network connections are allowed. The central software repository is not always accessible from within the HPC so it should be synchronized to a shared file system instead. This also concerns the usage of local copy of database release file instead of connecting to the database service. The network throughput to and from the HPC is limited which makes jobs with large input/output unsuitable. The usage of the CPU intensive event generation and Monte-Carlo simulations is the ideal case for the HPCs. Finally the usage of the Storage Element (SE) from a close Tier-1/2 for stage-in and stage-out could be a perfect solution as HPCs typically do not provide a SE.

3.1 PanDA on Oak Ridge Leadership Computing Facility

The current number two (number one from November 2012 until June 2013) on the Top 500 list is located at the Oak Ridge Leadership Computing Facility (OLCF) in Oak Ridge National Laboratory,

US. The supercomputer, called Titan, was the first large-scale system to use a hybrid architecture that utilizes both AMD 16-core Opteron 6274 CPUs and NVIDIA Tesla K20 GPU accelerators.

Together with ORNL LCF PanDA was adapted for OLCF (Titan) [13], trying to reuse existing PanDA components and workflow as much as possible. The PanDA connection layer runs on front-end nodes under user account. There is a predefined host to communicate with CERN from OLCF, connections are initiated from the front-end nodes. The SAGA (a Simple API for Grid Applications) framework was used as a local batch interface. The PanDA architecture on Titan is shown on figure 3. The pilot (payload submission) is running on the HPC interactive node and communicating with the local batch scheduler to manage jobs on Titan. Outputs are transferred to Brookhaven National Laboratory Tier-1 Grid centre. The initial demonstration was done for the Monte-Carlo event generators. The event generators are mostly computational and stand-alone code with small amount of stage-in/out data. In the LHC experiments 10–15% of all CPU resources are utilized to run event generators. It is important to define the mode of payloads execution. One of the possible scenarios would be to run HEP applications in backfill mode. The HPCs nodes are geared toward large scale jobs, about 10% of capacity of a typical HPC machine is unused due to a mismatches between job sizes and available resources. The rough estimation of a potential resource is 300M CPU hours per year, and it could be used by short jobs. PanDA can be a tool to harvest Titan resources opportunistically. Taking the above considerations into account we have adopted a PanDA pilot algorithm to use backfill information and to submit jobs in backfill mode.

- Pilot periodically queries Titan's job scheduler about available resources;
- Scheduler returns information about available (unscheduled) nodes and interval of availability;
- Pilot chooses the largest available block of nodes and submits jobs taking into account Titan's scheduling policy limitations.

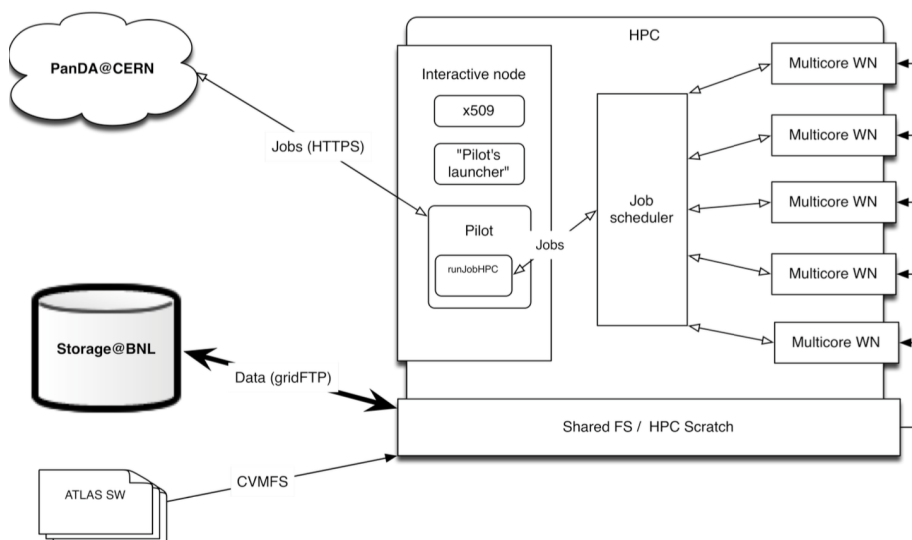


Figure 3. PanDA Architecture on Titan OLCF

First tests of the system were quite successful and show great promise of increasing the resource utilization on Titan. We demonstrated increased utilization by 3% of total capacity. Currently Titan is presented as ATLAS PanDA site.

3.2 Integration of Tier-1 Grid Center with the High Performance Computer at NRC-KI

In 2014 the pioneer work was initiated to develop a portal combining the Tier-1 and the supercomputer in Kurchatov Institute. The Tier-1 facility at Kurchatov Institute (NRC-KI) in Moscow is part of WLCG and it will process and store up to 10% of the total data obtained from ALICE, ATLAS and LHCb experiments. The second generation supercomputer HPC2 at the Kurchatov Institute based on Intel(R) Xeon(R) E5450@3.00GHz with peak performance 122,9 TFLOPS currently provides two worker nodes (16 cores) for ATLAS. This portal is aimed to provide an interfaces to run jobs at the Tier-1 Grid and the supercomputer, using common storage. The portal will be used not only for HEP, but also for other computing and data-intensive projects like the genome sequencing analysis in biology.

The integration scheme of PanDA WMS with Kurchatov's supercomputer is shown in figure 4. To launch ATLAS jobs the local pilot sheduler APF (Auto Pilot Factory) was installed. The APF is an independent subsystem which manages the delivery of "pilots" to worker nodes via a number of schedulers serving the sites at which PanDA operates.

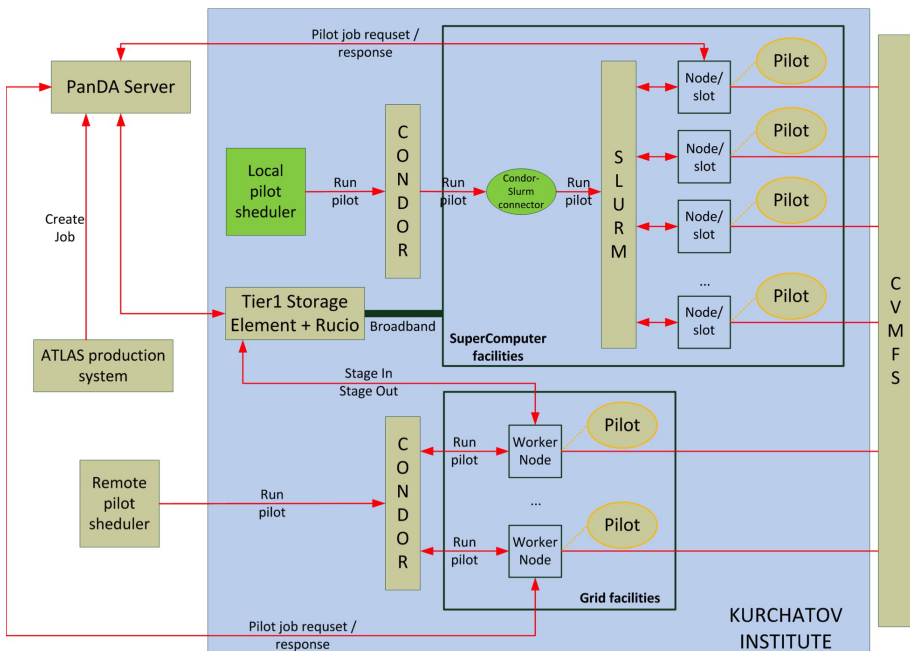


Figure 4. Integration of Tier-1 Grid and supercomputer using PanDA

The integration was done following the basic WLCG approach where one pilot is running on one core. The worker nodes of the supercomputer have access to Internet. This advantage avoids additional actions for data and SW moving and for connectivity with the PanDA server. To support the ATLAS workflow the CVMFS (CERN Virtual Machine File System) was installed on the worker

nodes. The CVMFS provides access to the full set of ATLAS SW releases. In addition the required libraries and compilers were installed.

Also a local PanDA instance was installed in NRC-KI to address studies in biology as well. It consists of four main specialized components: a server, an Auto Pilot Factory, a monitor, and a database server. The MySQL was chosen as the database backend technology. The Auto Pilot Factory is configured such that it works with standard pilots to run ATLAS-jobs derived from the production server at CERN and also it operates with the HPC pilot to run non-ATLAS jobs derived from the local PanDA server at NRC-KI. The PanDA monitor enables detailed monitoring of the job status diagnostics.

The HPC-pilot project was initiated for the Titan supercomputer and was successfully adopted for the HPC2 Kurchatov institute supercomputer. The HPC-pilot provides the ability to run MPI parallel jobs and to move input/output data to the worker nodes and out of the supercomputer. The HPC-pilot is running on an HPC interactive node and is communicating with the local batch scheduler to manage jobs over the available CPUs.

The implemented HPC-pilot working under PanDA at the NRC-KI proved its effectiveness in running two biological applications: the short read alignment using bowtie2 tool and the genome de-novo assembly using ABySS software. These two tools are frequently used for next generation sequencing data analysis. Two approaches to data processing parallelization were implemented: input data partitioning and OpenMP thread parallelization for bowtie2 and MPI+OpenMP combination for ABySS. Both approaches showed to be effective when processing genomic datasets varying from 1 to 100 Gbytes runs.

4 ATLAS Event Service

JEDI is an extension for PanDA [14] that adds to the PanDA server the capability to intelligently and dynamically break down tasks to jobs and even events based on optimal use of the processing resources currently available. The initial application of this capability was for event-level job splitting: jobs are defined in terms of the event ranges within files that they should process, and their dispatch and execution otherwise proceeds as normal. However this capability also presents the opportunity to manage processing and dispatch work at the event or event cluster level. The PanDA task decomposition is shown in figure 5.

The fine grained event service [15] based processing can enable agile, efficient utilization of opportunistic resources that appear and disappear with unpredictable timing and quantity. The event consumers can be injected into such a resource when it becomes available and participate in the ongoing task processing, concurrently delivering outputs to an aggregation point until the resource disappears, with negligible losses if the resource disappears suddenly.

The Event Service (ES) is designed to open opportunistic resources for efficient utilization and to improve the overall efficiency in the utilization of the processing and storage. The ES achieves this by decoupling processing from the chunkiness of files, streaming events into a worker and streaming the outputs away in a quasi-continuous manner, with a tunable granularity.

On the HPCs, the MPI-based master/client adaptation “Yoda” [16] of the Event Service allows tailoring workloads automatically to whatever scheduling opportunities the resource presents.

5 Summary and Conclusions

The PanDA’s capability for large-scale data-intensive distributed processing has been thoroughly demonstrated in one of the most demanding big data computing environments. PanDA was designed

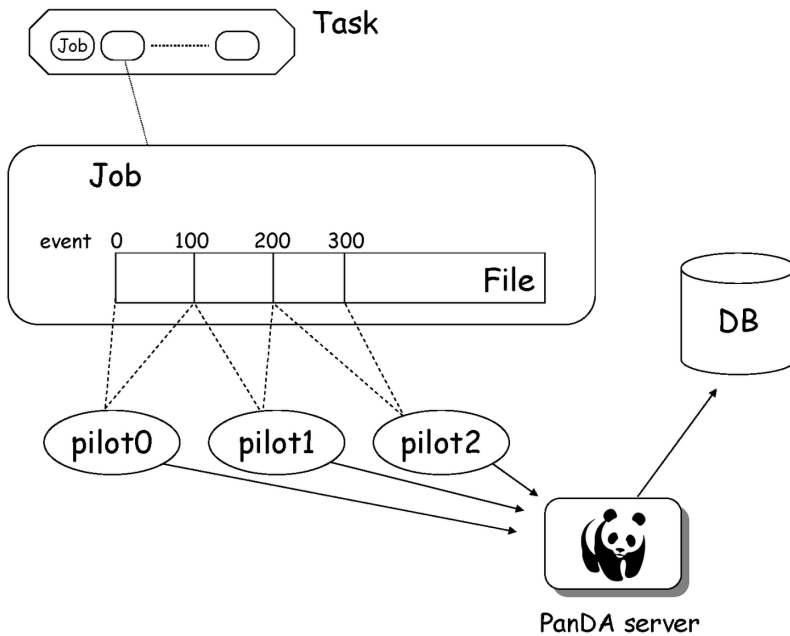


Figure 5. Integration of Tier-1 Grid and supercomputer using PanDA

to have the flexibility to adapt to emerging computing technologies in processing, storage, networking and the underlying software stack (middleware and file transfer). This flexibility has also been successfully demonstrated through the past five years of evolving technologies adapted by computing centres in ATLAS, which span many continents and yet are seamlessly integrated into PanDA.

The ATLAS Experiment is currently preparing for the massive computing challenges posed by the Run-2 at the LHC. With the doubling of the beam energy and luminosity together with the increased need for simulated data, a large increase in the computing resources is also needed. The storing and processing Run-2 data is a challenge that cannot be resolved with the currently existing ATLAS computing resources. To resolve this challenge, ATLAS is turning to commercial as well as academic Cloud services, Volunteer Computing and HPCs via the PanDA system. Tests have been performed on several Cloud services including Amazon EC2, Google Compute Element, and many others. Some of these resources are already routinely being used in ATLAS production. The developments in the HPCs integration with the ATLAS Computing Model is progressing rapidly. Several HPCs have been already added to the PanDA workflow.

The diversity of computing platforms and fabrics available for scientific computing, which are significantly increasing with the new additions of cloud resources and supercomputers, presents challenges that PanDA has been designed to address. PanDA provides a coherent, homogeneous processing system layered over diverse and heterogeneous processing resources. This helps insulate production operators and analysis users from the complexity of the underlying processing infrastructure. It also maximizes the amount of code in the system that is independent of the underlying middleware and facilitates the actual resource use in any given environment. The layered structure of PanDA, which enables it to support a variety of middleware, heterogeneous computing systems, and diverse appli-

cations, also makes PanDA ideally suited for a common big data processing system for many data intensive tasks. PanDA lowers the barriers and enable the scientists to easily carry out their research using a variety of distributed computing systems.

Acknowledgements

We wish to thank all our colleagues who contributed to ATLAS data processing and Monte-Carlo simulation activities and to PanDA software development and installation. We wish to thank OLCF computing team for helping to port PanDA on Titan. This work was funded in part by the U.S. Department of Energy, Office of Science, High Energy Physics and Advanced Scientific Computing Research under Contracts No. DE-SC0012704, No. DE-AC02-98CH10886 and Contract No. DE- AC02-06CH11357. NRC-KI team work was funded by the Russian Ministry of Science and Education under Contract No 14.Z50.31.0024. Supercomputing resources at NRC-KI are supported as a part of the center for collective usage (project RFMEFI62114X0006, funded by Russian Ministry of Science and Education).

References

- [1] G. Aad et al. [ATLAS Collaboration], *Journal of Instrumentation* **3**, S08003 (2008)
- [2] D. Adams, D. Barberis, C. Bee, R. Hawkings, S. Jarp, R. Jones, D. Malon, L. Poggioli, G. Poulard, D. Quarrie, and T. Wenaus, Tech. rep., CERN CERN-LHCC-2004-037/G-085 (2005)
- [3] I. Foster and C. Kesselman (eds), *The Grid: Blueprint for a New Computing Infrastructure* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998)
- [4] T. Maeno on behalf of the PanDA team and ATLAS collaboration, *J. Phys.: Conf. Ser.* **119**, 062036 (2008)
- [5] T. Maeno on behalf of the PanDA team and ATLAS collaboration, *J. Phys.: Conf. Ser.* **513**, 032062 (2014)
- [6] S. Panitkin et al. [ATLAS Collaboration], *J. Phys.: Conf. Ser.* **513**, 062037 (2014)
- [7] EC2: Amazon Elastic Computing Cloud, <http://aws.amazon.com/ec2>
- [8] A. Klimentov, P. Nevski, M. Potekhin, and T. Wenaus, *J. Phys.: Conf. Ser.* **331**, 072058 (2011)
- [9] F. Barreiro, D. Benjamin, K. De, I. Gable, V. Hendrix, S. Panitkin, M. Paterson, D. van der Ster, R. Taylor, R. Vitillo, and R. Walker, *J. Phys.: Conf. Ser.* **396**, 032011 (2012)
- [10] G. Evans, S. Panitkin, and A. Hanushevsky, *Cloud Computing and High-Energy Particle Physics: How ATLAS Experiment at CERN Uses Google Compute Engine in the Search for New Physics at LHC*, Google IO 2013, <https://developers.google.com/events/io/sessions/333315382>
- [11] P. Armstrong et al., [arXiv:1007.0050v1 [cs.DC]]
- [12] HTCondor: High Throughput Computing, <http://research.cs.wisc.edu/htcondor/>
- [13] The Titan platform, <http://www.olcf.ornl.gov/titan/>
- [14] K. De, D. Golubkov, A. Klimentov, M. Potekhin, and A. Vaniachine on behalf of the ATLAS collaboration, *J. Phys.: Conf. Ser.* **513**, 032078 (2014)
- [15] P. Calafiura et al. [ATLAS Collaboration], *The ATLAS Event Service: A New Approach to Event Processing*, *J. Phys.: Conf. Ser.* to be published in CHEP 2015 Proceedings
- [16] P. Calafiura et al. [ATLAS Collaboration], *Fine grained event processing on HPCs with the ATLAS Yoda system*, *J. Phys.: Conf. Ser.* to be published in CHEP 2015 Proceedings

