

## INTEGRATION OF PANDA WORKLOAD MANAGEMENT SYSTEM WITH SUPERCOMPUTERS

*K. De<sup>a</sup>, S. Jha<sup>b</sup>, A. A. Klimentov<sup>c,d</sup>, T. Maeno<sup>c</sup>,  
R. Yu. Mashinistov<sup>d,1</sup>, P. Nilsson<sup>c</sup>, A. M. Novikov<sup>d</sup>,  
D. A. Oleynik<sup>a,e</sup>, S. Yu. Panitkin<sup>c</sup>, A. A. Poyda<sup>d</sup>, K. F. Read<sup>f</sup>,  
E. A. Ryabinkin<sup>d</sup>, A. B. Teslyuk<sup>d</sup>, V. E. Velikhov<sup>d</sup>, J. C. Wells<sup>f</sup>, T. Wenaus<sup>c</sup>*

<sup>a</sup> University of Texas at Arlington, Arlington, TX, USA

<sup>b</sup> Rutgers University, Piscataway, NJ, USA

<sup>c</sup> Brookhaven National Laboratory, Upton, NY, USA

<sup>d</sup> National Research Center “Kurchatov Institute”, Moscow

<sup>e</sup> Joint Institute for Nuclear Research, Dubna

<sup>f</sup> Oak Ridge National Laboratory, Oak Ridge, TN, USA

The Large Hadron Collider (LHC), operating at the international CERN Laboratory in Geneva, Switzerland, is leading Big Data driven scientific explorations. Experiments at the LHC explore the fundamental nature of matter and the basic forces that shape our universe and were recently credited for the discovery of the Higgs boson. ATLAS, one of the largest collaborations ever assembled in the sciences, is at the forefront of research at the LHC. To address an unprecedented multipetabyte data processing challenge, the ATLAS experiment relies on a heterogeneous distributed computational infrastructure. The ATLAS experiment uses PanDA (Production and Data Analysis) Workload Management System for managing the workflow for all data processing on over 140 data centers. Through PanDA, ATLAS physicists see a single computing facility that enables rapid scientific breakthroughs for the experiment, even though the data centers are physically scattered all over the world.

While PanDA currently uses more than 250,000 cores with a peak performance of 0.3+ petaFLOPS, next LHC data taking runs will require more resources than grid computing can possibly provide. To alleviate these challenges, LHC experiments are 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 supercomputers in the United States, Europe, and Russia (in particular, with Titan supercomputer at Oak Ridge Leadership Computing Facility (OLCF), Supercomputer at the National Research Center “Kurchatov Institute”, IT4 in Ostrava, and others).

The current approach utilizes a modified PanDA pilot framework for job submission to the supercomputers batch queues and local data management, with light-weight MPI wrappers to run single-threaded workloads in parallel on Titan’s multicore worker nodes.

This implementation was tested with a variety of Monte Carlo workloads on several supercomputing platforms. We will present our current accomplishments in running PanDA WMS at supercomputers and demonstrate our ability to use PanDA as a portal independent of the computing facility’s infrastructure for high energy and nuclear physics, as well as other data-intensive science applications, such as bioinformatics and astroparticle physics.

PACS: 29.85.-c

---

<sup>1</sup>E-mail: Ruslan.Mashinistov@cern.ch

## INTRODUCTION

The ATLAS experiment [1] at the Large Hadron Collider (LHC) is designed to explore the fundamental properties of matter for the next decade at the highest energy ever achieved at a laboratory. Since LHC became operational in 2009, the experiment has produced and distributed hundreds of petabytes of data worldwide among the O(100) heterogeneous computing centres of the Worldwide LHC Computing Grid (WLCG) [2]. Thousands of physicists are engaged in analyzing these data.

The Large Hadron Collider returns to operations after a two-year offline period, Long Shutdown 1, which allowed thousands of physicists worldwide to undertake crucial upgrades to the already cutting-edge particle accelerator. The LHC now begins its second multiyear operating period, Run 2, which will take the collider through 2018 with collision energies nearly double than those of Run 1. In other words, Run 2 will nearly double the energies that allowed researchers to detect the long-sought Higgs Boson in 2012. The WLCG computing sites are usually dedicated clusters specifically set up to meet the needs of the LHC experiments. More than one million grid jobs run on the distributed computing sites all over the world per day on more than 200,000 CPU cores.

The 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. Additional computing and storage resources are therefore required. To alleviate these challenges, ATLAS is engaged in an ambitious program to expand the current computing model, include additional resources, i.e., the opportunistic use of supercomputers and high-performance computing clusters (HPCs).

## PANDA WORKLOAD MANAGEMENT SYSTEM

A sophisticated Workload Management System (WMS) is needed to manage the distribution and processing of huge amount of data. PanDA (Production and Distributed Analysis) WMS [3] 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.

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 the 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 (middleware). 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-based [4, 5] 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 a central server to retrieve a real payload (i.e., an end-user job) and execute it. Using these pilot-based workflows help to improve job reliability, optimize resource utilization, allow for opportunistic resources usage, and mitigate many of the problems associated with the inhomogeneities found on the grid. Extending

PanDA beyond the grid will further expand the potential user community and the resources available to them.

The JEDI (Job Execution and Definition Interface) extension to PanDA adds a new functionality to the PanDA server to dynamically break down the tasks based on optimal usage of available processing resources. With this new capability, the tasks can now be broken down at the level of either individual events or event clusters or ensembles, as opposed to the traditional file-based task granularity. This allows the recently developed ATLAS Event Service to dynamically deliver to a compute node only that portion of the input data which will be actually processed there by the payload application (simulation, reconstruction, and/or data analysis), thus avoiding costly pre-staging operations for entire data files. The Event Service leverages modern networks for efficient remote data access and highly scalable object store technologies for data storage. It is agile and efficient in exploring diverse, distributed, and potentially short-lived (opportunistic) resources: “conventional resources” (grid), supercomputers, commercial clouds, and volunteer computing.

### EXTENDING PANDA TO SUPERCOMPUTERS

The modern 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 (compute) nodes and provide facilities for low latency internode communications.

For the ATLAS experiment (or we should say WLCG) to make effective and impactful use of HPCs, 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 to ATLAS is the total number of cycles that can be offloaded from the traditional grid resources. The standard ATLAS workflow is not well-adapted for HPCs due to several complications: typically the worker node setup is fixed, with no direct wide-area network connections, the amount of RAM per core can be quite limited, and in many cases a customized operating system is used along with a specialized software stack.

The following is a nonexhaustive list of some of the known problems with suggested solutions. Communication with the 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 nodes where wide-area 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 is also true concerning the usage of local copy of the database release file instead of connecting to the database service. The network throughput to and from the HPC is limited which can make jobs with large input/output challenging. 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.

Supercomputing centers in the USA, Europe, and Asia, in particular, the Titan supercomputer [6] at Oak Ridge Leadership Computing Facility (OLCF), the National Energy Research

Supercomputing Center (NERSC) in the USA, Ostrava supercomputing center in Czech Republic, and “Kurchatov Institute” in Russia (NRC KI) are now integrated within the ATLAS workflow via the PanDA WMS. This will make Leadership Computing facilities of much greater utility and impact for HEP computing in the future. Developments in these directions are presently underway in the LHC experiments.

### **PANDA ON TITAN AT OAK RIDGE LEADERSHIP COMPUTING FACILITY**

The Titan supercomputer, current number two (number one until June 2013) on the Top 500 list [7] is located at the Oak Ridge Leadership Computing Facility within Oak Ridge National Laboratory, USA. It has theoretical peak performance of 29 petaFLOPS. Titan was the first large-scale system to use a hybrid CPU–GPU 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 connection to the wide-area network. Titan is served by the 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.

Taking advantage of its modular and extensible design, the PanDA pilot code and logic have been enhanced with tools and methods relevant for HPC. The pilot runs on Titan’s front-end nodes, which allows it to communicate with the 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 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 Laboratory is currently used for data transfer to and from Titan, but in principle that can be any grid site. The pilot submits ATLAS payloads to the worker nodes using the local batch system (PBS) via the SAGA (Simple API for grid Applications) interface [8]. Figure 1 shows the schematic diagram of PanDA components on Titan.

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.

For running such workloads on Titan, we developed an MPI wrapper to launch multiple instances of single-node workloads simultaneously. MPI wrappers are typically workload specific since they are responsible for setup 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 are what the pilot actually submits to a batch queue to run on Titan. The pilot reserves the 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 the necessary preparations, will start the actual payload as a subprocess and

will wait until its completion. In other words, the MPI wrapper serves as a “container” for non-MPI workloads and allows us to efficiently run unmodified grid-centric workloads on parallel computational platforms, like Titan.

Leadership Computing Facilities (LCF), like Titan, are geared towards large-scale jobs by design. Time allocation on 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 of running extreme-scale computational projects. As a consequence, on average, about ten percent 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 ten percent correspond to estimate of 300M core hours per year. Hence, a system that can occupy those temporarily free nodes would be very valuable. It would allow the LCF to deliver more compute 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 the 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, resulting in short wait times for these jobs. We note that care must be taken to manage potential contention for shared system resources, e.g., internal communication bandwidth, IO bandwidth, and access to front-end and data-transfer nodes.

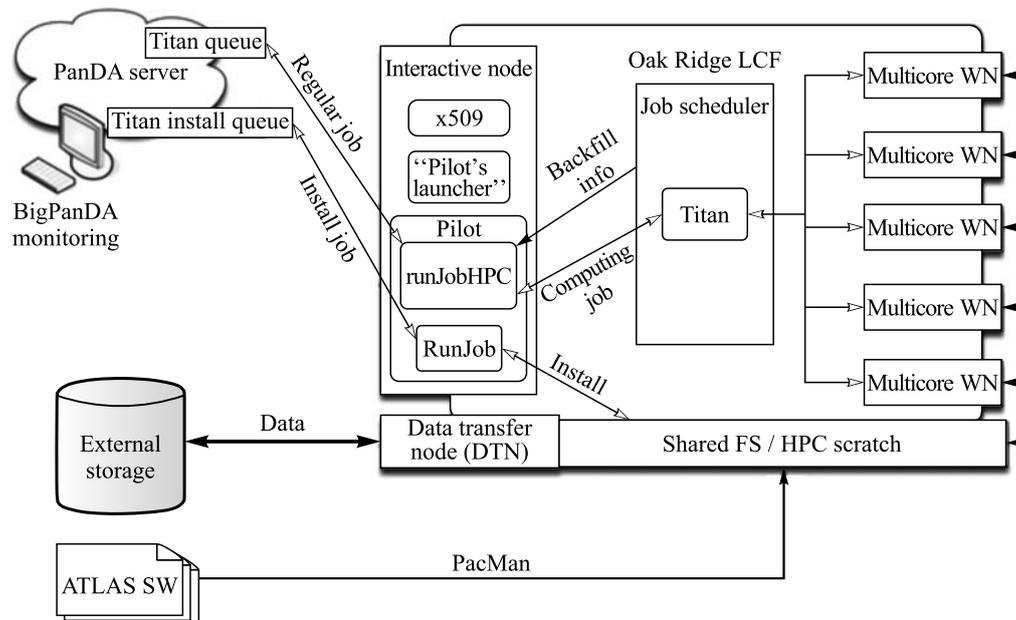


Fig. 1. Schematic view of PanDA interface with Titan

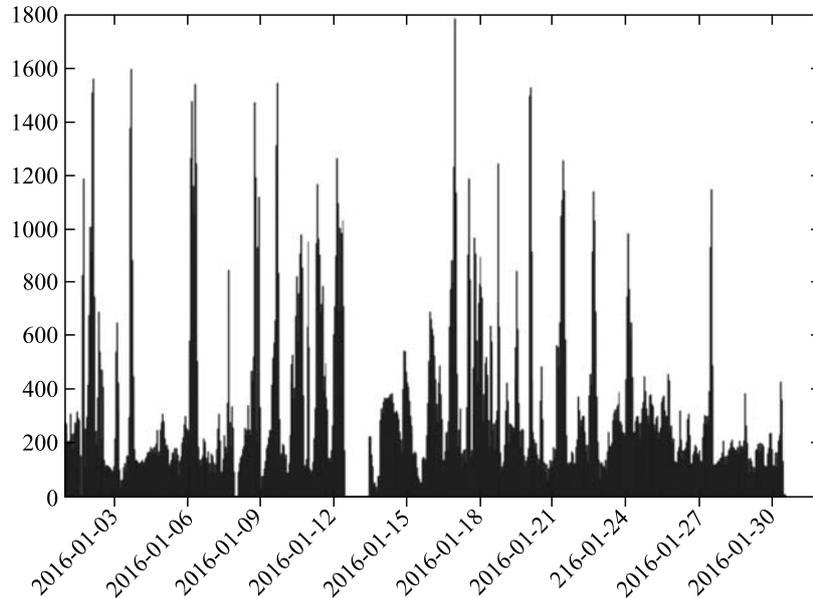


Fig. 2. Number of ATLAS production jobs on Titan

Titan was fully integrated with the ATLAS PanDA based Production and Analysis system, and now the ATLAS experiment routinely runs Monte Carlo simulation tasks there. All operations, including data transfers to and from Titan, are transparent to the ATLAS Computing Operations team and physicists. Figure 2 shows an example of the ATLAS monitoring dashboard plot of running ATLAS production jobs on Titan.

#### **INTERGRATION OF TIER-1 GRID CENTER WITH HIGH-PERFORMANCE COMPUTER AT NRC KI**

Pioneer project of combining Tier-1, supercomputer, and the cloud platform into a single portal at the Kurchatov Institute began in 2014 and continues to the present day. Now the portal is aimed to provide an interface to run jobs at the Tier-1 grid and supercomputer using common storage. The portal is used for ATLAS production and user analysis tasks and also for biology studies for genome sequencing analysis.

The Tier-1 facility at NRC KI is the part of WLCG and it will process and store up to 10% of total data obtained from ALICE, ATLAS, and LHCb experiments. The second-generation supercomputer HPC2 [9] based on Intel(R) Xeon(R) E5450@3.00GHz with peak performance 122.9 TFLOPS. Currently, 32 worker nodes (256 cores) are provided for ATLAS Production tasks, two worker nodes (16 cores) are provided for ATLAS user analysis jobs. OpenStack based cloud platform having performance 1.5 TFLOPS and providing 16 nodes, 256 cores, 512 GB RAM, 60 TB at the storage system and InfiniBand connectivity.

The integration schema of PanDA WMS with Kurchatov's supercomputer is shown in Fig.3. Local APF (Auto Pilot Factory) — an independent subsystem that manages the delivery of “pilots” to HPC's worker nodes via a number of schedulers serving the sites at which PanDA operates — was installed to launch ATLAS jobs.

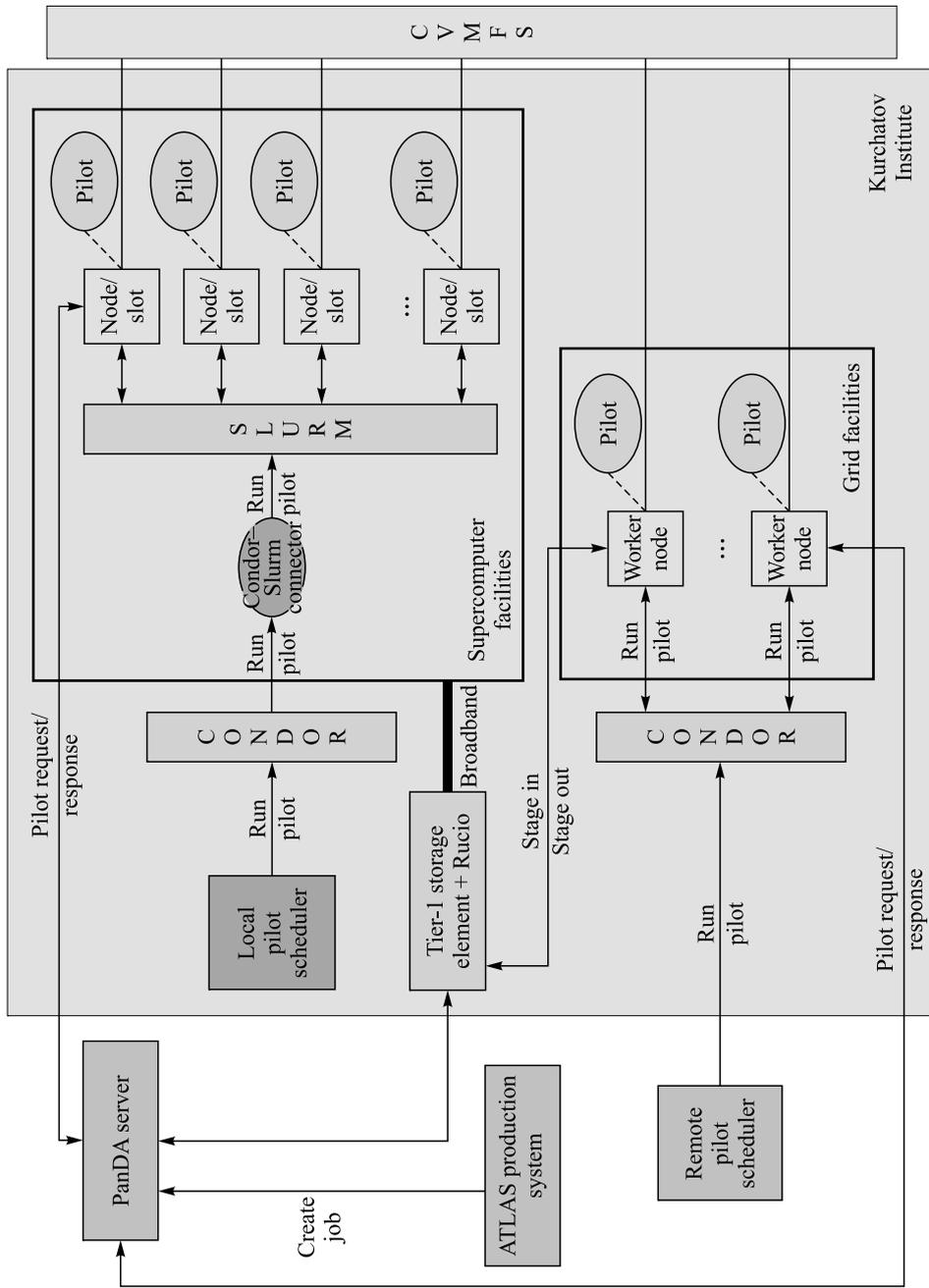


Fig. 3. Integration of Tier-1 grid and supercomputer using PanDA

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 the Internet. They have direct access to the data, SW, and to 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.

Also, the local PanDA instance was installed in NRC KI for biology studies. The local instance consists of the following main components: server (MySQL), Auto Pilot Factory, monitor, and database server. Auto Pilot Factory is configured in a way that it works with standard pilots to run ATLAS jobs derived from production server at CERN and also operates with the pilot adopted to HPC to run non-ATLAS jobs derived from the local PanDA server at NRC KI. The PanDA monitor performs detailed monitoring of the jobs for its status diagnostics.

The HPC-pilot project was initiated for Titan supercomputer and successfully adopted for Kurchatov Institute's supercomputer HPC2. The HPC-pilot provides the ability to run MPI parallel jobs and to move data to/from the supercomputer. The HPC-pilot runs on the HPC interactive node and communicates with the local batch scheduler to manage jobs over the available CPUs.

The implementation of the HPC-pilot at the NRC KI is used to run biology jobs that analyze the data obtained in genome sequencing in collaboration with the Genomics laboratory of the Kurchatov Institute. This analysis consists in studying the ancient DNA samples using the Paleomix [10] pipeline application. This pipeline includes a number of open-source software for quick data processing of the Next Generation Sequencing (NGS). The common shared queue of supercomputer is provided to run biology jobs allocating up to 1000 available CPU cores.

## **PANDA EVENT SERVICE AND SUPERCOMPUTERS**

The Event Service is a complex distributed system in which different components communicate with each other over the network using HTTP. For event processing, it uses AthenaMP, a process-parallel version of the ATLAS simulation, reconstruction and data analysis framework Athena. A PanDA pilot starts an AthenaMP application on the compute node and waits until it goes through the initialization phase and forks worker processes. After that, the pilot requests an event-based workload from the PanDA JEDI, which is dynamically delivered to the pilot in the form of event ranges. The event range is a string that, together with other information, contains positional numbers of events within the file and a unique file identifier (GUID). The pilot streams event ranges to the running AthenaMP application, which takes care of the event data retrieval, event processing and output file producing (a new output file for each range). The pilot monitors the directory in which the output files are produced and as they appear sends them to an external aggregation facility (Object Store) for final merging.

Supercomputers are one of the important deployment platforms for Event Service applications. However, on most HPC machines there is no Internet connection from compute nodes to the wide-area network. This limitation makes it impossible to run the conventional Event Service on such systems because the payload component needs to communicate with central services (e.g., job brokerage, data aggregation facilities) over the network.

In summer 2014, we started to work on an HPC-specific implementation of the Event Service that would leverage MPI for running on multiple compute nodes simultaneously. To

speed up the development process and also to preserve all functionality already available in the conventional Event Service, we reused the existing code and implemented light-weight versions of the PanDA JEDI (Yoda, a diminutive JEDI) and the PanDA Pilot (Droid), which communicate with each other over MPI.

Figure 4 shows a schematic of a Yoda application, which implements the master–slave architecture and runs one MPI-rank per compute node. The responsibility of Rank 0 (Yoda, the master) is to send event ranges to other ranks (Droid, the slave) and to collect from them the information about the completed ranges and the produced outputs. Yoda also continuously updates event range statuses in a special table within an SQLite database file on the HPC shared file system. The responsibility of a Droid is to start an AthenaMP payload application on the compute node, receive event ranges from Yoda, deliver the ranges to the running payload, collect information about the completed ranges (e.g., status, output file name, and location), and to pass this information back to Yoda.

Yoda distributes event ranges between Droids on a first-come, first-served basis. When some Droid reports completion of an event range, Yoda immediately responds with a new range for this Droid. In this way, Droids are kept busy until all ranges assigned to the given job have been processed or until the job exceeds its time allocation and gets terminated by the batch scheduler. In the latter case, the data losses caused by such termination are minimal, because the output for each processed event range gets saved immediately in a separate file on the shared file system.

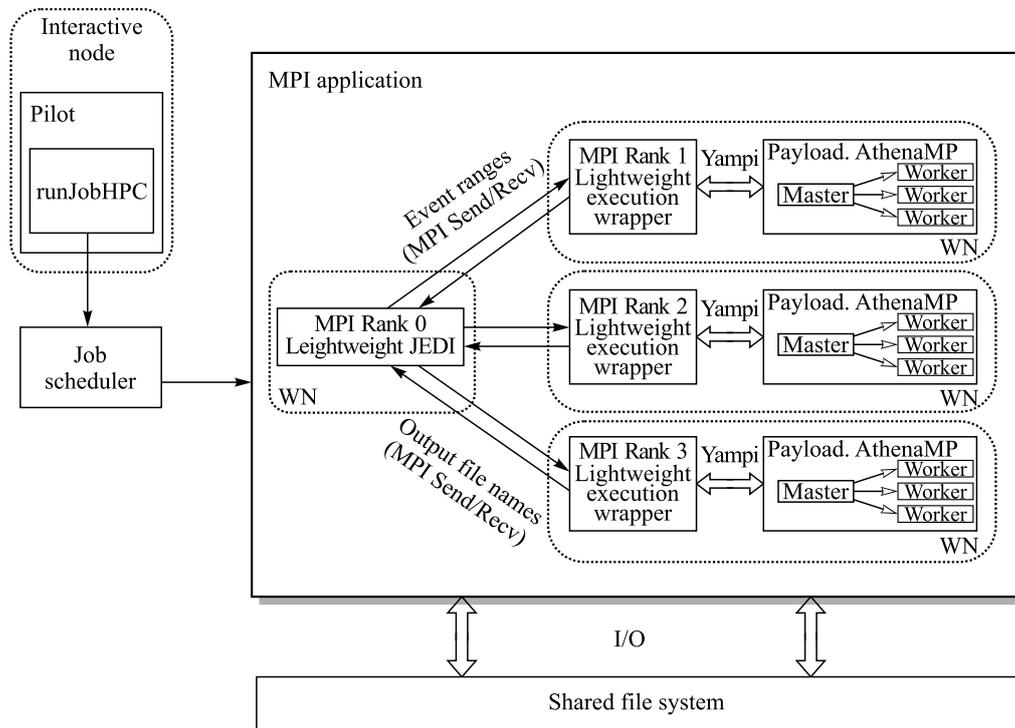


Fig. 4. Yoda application which implements the master–slave architecture and runs one MPI-rank per compute node

## 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. The layered structure of PanDA, which enables it to support a variety of middleware, heterogeneous computing systems, and diverse applications, makes PanDA also ideally suited for a common big-data processing system for many data-intensive sciences. The PanDA lowers the barrier for scientists to easily carry out their research using a variety of distributed computing systems.

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 by a factor of 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 different supercomputers. The detailed information is given for Titan supercomputer at Oak Ridge Leadership Computing Facility and supercomputer HPC2 at NRC KI. The current approach utilizes modified PanDA-pilot frameworks for job submission to supercomputer's batch queues and local-data management. Also, the work underway enables the use of PanDA by new scientific collaborations and communities beyond LHC and even HEP.

**Acknowledgements.** This work was funded in part by the US Department of Energy, Office of Science, High Energy Physics and Advanced Scientific Computing Research under Contracts Nos. DE-SC0012704, DE-AC02-98CH10886, and DE-AC02-06CH11357. The NRC KI team work was funded by the Russian Ministry of Science and Education under Contract No. 14.Z50.31.0024. Supercomputing resources at the NRC KI are supported as a part of the center for collective usage (project RFMEFI62114X0006, funded by the Russian Ministry of Science and Education). 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 US Department of Energy under Contract No. DE-AC05-00OR22725.

## REFERENCES

1. *Aad J. et al. (ATLAS Collab.). The ATLAS Experiment at the CERN Large Hadron Collider // J. Instr.* 2008. V. 3. P. S08003.
2. The Worldwide LHC Computing Grid (WLCG). <http://wlcg.web.cern.ch/LCG>.
3. *Maeno T.* Overview of ATLAS PanDA Workload Management // *J. Phys.: Conf. Ser.* 2011. V. 331. P. 072024.
4. *Nilsson P.* The ATLAS PanDA Pilot in Operation // *Proc. of the 18th Intern. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)*.
5. *Turilli M., Santcroos M., Jha S.* A Comprehensive Perspective on the Pilot-Job Systems. <http://arxiv.org/abs/1508.04180>.
6. Titan at OLCF Web Page. <https://www.olcf.ornl.gov/titan/>.
7. Top500 List. <http://www.top500.org/>.
8. The SAGA Framework Web Site. <http://saga-project.github.io>.
9. Kurchatov Institute HPC Cluster. <http://computing.kiae.ru/resources/hpc2/>.
10. *Schubert M. et al.* Characterization of Ancient and Modern Genomes by SNP Detection and Phylogenomic and Metagenomic Analysis Using PALEOMIX // *Nat. Protoc.* 2014. V. 9, No. 5. P. 1056–1082.