Workload Management Portal for High Energy Physics applications and compute intensive science

V. Aulov¹, K. De², D. Drizhuk¹, A. Klimentov¹,³, D. Krasnopevtsev¹,⁵, R. Mashinistov¹, A. Novikov¹, D. Oleynik²,⁴, A. Poyda¹, E. Ryabinkin¹, I. Tertychnyy¹, A. Teslyuk¹

¹ National Research Center “Kurchatov Institute”, Moscow, Russia
² University Texas at Arlington, Upton, NY, USA
³ Brookhaven National Laboratory, Upton, NY, USA
⁴ Joint Institute for Nuclear Research, Dubna, Russia
⁵ National Research Nuclear University MEPhI

VasiliyAulov@gmail.com, kaushik@uta.edu, d.drizhuk@gmail.com, alexei.klimentov@cern.ch, Dimitriy.Krasnopevtsev@cern.ch, Ruslan.Mashinistov@cern.ch, novikov@wdcb.ru, Danila.Oleynik@cern.ch, Poyda_AA@nrcki.ru, rea@grid.kiae.ru, ivantertychnyi@gmail.com, anthony.teslyuk@grid.kiae.ru

Abstract

Creation of global e-Infrastructure involves an integration of isolated local resources into common heterogeneous computing environment. In 2014 a pioneering work to develop a large scale data- and task- management system for federated heterogeneous resources has been started at the National Research Centre "Kurchatov Institute" (NRC KI, Moscow). As a part of this work, we have designed, developed and deployed a portal to submit payloads to heterogeneous computing infrastructure. It combines Tier-1, Cloud-infrastructure, and a supercomputer at the Kurchatov institute. This portal is aimed to provide a common interface to submit tasks to Grid sites, commercial and academic clouds and supercomputers. Integration of Tier-1 and the supercomputer has allowed to notably increase total CPU capacity available for Large Hadron Collider (LHC) experiments. The portal can be used not only for High Energy Physics (HEP) applications, but also for other compute-intensive sciences such as bioinformatics with genome and sequence analysis; astrophysics with cosmic rays analysis, antimatter and dark matter search, etc.

This article describes developed portal as a top layer for computing facilities infrastructure for High Energy Physics and other compute-intensive science applications. The article presents the results of using PanDA at NRC KI supercomputer/Cloud as underlying technology for the portal.

Keywords: Workload Management Systems, High Energy Physics, compute intensive science, PanDA, supercomputer
1 Introduction

The National Research Centre "Kurchatov Institute" (NRC KI, Moscow) possesses a wide range of different computing resources, and our aim was to integrate them into a unified computing system intended for data handling. One of the steps in this work was to develop a portal which allows scientists to submit jobs to KI's computing infrastructure. As underlying technology the PanDA (Production and Distributed Analysis) Workload Management System (WMS) (Maeno, 2008) developed and used by the ATLAS (The ATLAS Collaboration, 2008) experiment at LHC has been chosen. PanDA project was started in 2005, when a variety of workload management systems, such as LEXOR (De Salvo, Negri, Rebatto, & Vaccarossa, 2004), CAPONE (Mambelli & al., 2004) or DULCINEA (Sturrock & al., 2004), were deployed in ATLAS to support different applications. During the following years PanDA's superiority in terms of performance lead to its adoption as the default and single system for ATLAS before LHC started operating in 2009. Today PanDA has grown to support all distributed workflows in ATLAS and is backed by a huge user and support base worldwide.

Within this project we have installed and configured PanDA instance at NRC KI, adopted it for Grid/Cloud/supercomputer facilities with shared data storage, extended the system to support multicore parallel jobs, developed user-friendly web interface for jobs submission and to control-local user authentication.

Now we are using the portal to run HEP applications and bioinformatics applications (like de-novo assembly, SNP calling, genome annotation and so on).

2 PanDA Overview

As a basis technology PanDA workload management system has been chosen.

PanDA delivers transparency of data and its processing in a distributed computing environment to LHC scientists. It provides execution environments for a wide range of experimental applications, automates centralized data processing, enables data analytics for dozens of physics groups, supports custom workflow of individual scientists, provides a unified view of distributed worldwide resources, presents status and history of workflow through an integrated monitoring system, archives and curates all workflow, manages distribution of data as needed for processing or scientists access, and provides other features mentioned later in this proposal. The rich menu of features provided, coupled with support for heterogeneous computing environments, makes PanDA ideally suited for scientific data processing.

PanDA has a highly scalable and flexible architecture. Scalability has been demonstrated in ATLAS through the rapid increase in usage over the past three years. 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 and file management). This flexibility has also been successfully demonstrated through the past five years of evolving technologies adapted by computing centers in ATLAS, which span many continents and yet are seamlessly integrated into PanDA. This proven scalability and flexibility makes PanDA ideally suited for adoption by future megascience projects.

PanDA's scalability was demonstrated in ATLAS environment, during the first LHC datataking period (2010-2013). In 2015, the number of used slots per day for ATLAS jobs being processed on WLCG resources exceeded 200000.

The generalized architecture of PanDA system deployed at Kurchatov institute can be schematically illustrated as shown on figure 1. The system includes the following main components:
Server. The server is the heart of the PanDA system and will be factorized as a general WMS service. The main components of the server are:

- Database. A system-wide job database that records comprehensive static and dynamic information on all jobs in the system. Relational databases implement the job queue and all metadata and monitoring repositories.
- Brokerage. An intelligent module operates to prioritize and assign work on the basis of job type, priority, software availability, input data and its locality, real-time job statistics, and available CPU and storage resources.
- Dispatcher. A component in the PanDA server which receives requests for jobs from pilots and dispatches job payloads by taking priorities, resource allocation policy, and retry strategies into account.

- Pilots. Pilot jobs are used for acquisition of processing resources. Workload jobs are assigned to successfully activated and validated pilots by the PanDA server based on brokerage criteria. This 'late binding' of workload jobs to processing slots prevents latencies and failure modes in slot acquisition from impacting the jobs, and maximizes the flexibility of job allocation to resources based on the dynamic status of processing facilities and job priorities. The pilot is also a principal 'insulation layer' for PanDA, encapsulating the complex heterogeneous environments and interfaces of the grids and facilities on which PanDA operates. Pilots generally carry a generic 'production' grid proxy, with an additional attribute 'pilot' indicating a pilot job.

- Autopilot Factory (APF). An independent subsystem manages the delivery of pilot jobs to worker nodes via a number of schedulers (‘pilot factories’) serving the sites at which PanDA operates. A pilot once launched on a worker node contacts the dispatcher and receives an available job appropriate to the site. An important attribute of this scheme for pseudo-interactive analysis, where minimal latency from job submission to launch is important, is that the pilot dispatch mechanism bypasses any latencies in the scheduling system for submitting and launching the pilot itself. The pilot job mechanism isolates workload jobs from grid and batch system failure modes (a workload job is assigned only when the pilot successfully launches on a worker node).

- End-user client module responsible for user jobs definition and submitting to the server. In general client can make a description of a job in a text editor and send it to the server via HTTPS protocol using third party utilities, for example, cURL. However, forming the description by hand is inconvenient, so the process is partially automatized with client libraries, usually including a graphic interface.
- Data Management System (DMS) intended for data control. Its main components are file catalog responsible for storing the metadata of files and their replicas, and data transfer system which moves replicas between different storage systems.
- Data Storage. Usually distributed data storage is used.
- Logging System intended to accept and store logging information from different system modules.

Module approach of PanDA allows the independent realization of different components adapted for different requirements and jobs. The most known version of PanDA is the PanDA realization for the needs of ATLAS experiment. In this version a number of components are Grid specific (for example, data storage) or ATLAS specific (data management system). Also, this version is intended for multicomputer clusters, thus some components (for example, APF and pilots) have to be modified for usage with other architectures.

Therefore, expansion or additional implementation of several system modules, including pilots system, data management system, data storage, and also integration of the local computing resources, while taking in consideration their architectural features were required to create a portal for processing not only ATLAS data, but also data for other compute intensive science areas.

3 PanDA Pilot Scheme Extending

Basic PanDA pilot realization is demonstrated at figure 2.

![Figure 2. Basic pilot scheme for the ATLAS experiment](image)

Server stores the information about all resources and jobs in the system. Server distributes the jobs among various queues associated with computing sites. Pilot runs on the same computing node as the requested job. Main 3 modules of the pilot are:

- Monitor, which controls the payload and pilot execution and updates PanDA server.
- RunJob, which launches the payload at the working node and processes its outputs.
- Mover, which gets the input data and transfers the output data and logs and registers them into the DMS.

All pilot modules run on the same node and use local access to data as shown at figure 2.
While this scheme works well for ATLAS jobs on grid sites, it cannot be used for infrastructures with limited access to working (computing) nodes, like HPC. To solve this issue we extended the pilot scheme to work in “remote job launch” mode. The extended scheme is shown at figure 3.

In this scheme the pilot and jobs run on different nodes. To implement this scheme we extended pilot modules to support remote control mode through SSH tunnels.

- The RunJob module launches payload by remote call of batch system. In this scheme exact amount of resources could be requested from batch system accordingly to job definition.
- Mover plugin, which allows transfer data through requests to special data-transfer system, was realized.

In small modification of the scheme Mover module transfers data through local file system of the edge node. It allows to run the pilot when worker nodes have no direct connection to the data management system.

4 Portal Deployment at NRC “Kurchatov institute”

The following resources were integrated for the portal at NRC KI:

- The supercomputing facilities at Kurchatov Centre for Collective Usage which include high performance cluster with peak performance of 122.9 TFLOPS. It operates since September 2011. It consists of 1280 nodes each with 2 CPUs of 8 cores (10240 cores total) interconnected with high throughput InfiniBand DDR network. Its total RAM is 20.5 Tb and local data storage has 144 Tb in Lustre 2.0 parallel file system. Nodes are operated by SLURM batch system in CentOS Linux.
- One of the largest Russian WLCG Tier-1 centers at NRC KI. The Tier-1 facility will process, simulate and store up to 10% of total data obtained from ALICE, ATLAS and LHCb experiments.
OpenStack based Cloud platform having performance 1.5 TFLOPS and providing 16 nodes, 256 cores, 512 Gb RAM, 60 Tb at storage system and InfiniBand connectivity.

4.1 Using Portal for HEP jobs

ATLAS is the HEP experiment supported by our portal. ATLAS jobs are single core and use unique ATLAS software. This allows us to reuse a lot of already existing components, however, we still had to perform a considerable work to integrate different computing resources, such as supercomputer, grid and Cloud facilities.

Portal working scheme which supports ATLAS jobs at grid Tier-1 center and supercomputer at NRC KI is given at figure 4. In this scheme the portal interacts with the central PanDA server at CERN, where queues for NRC KI are defined and intended for the grid and the supercomputer resources at NRC KI. Special applications on client side are used by physicists to submit ATLAS jobs (Prun, Pathena or panda-client). Grid storage element and ATLAS Distributed Data Management system (CERN RUCIO, 2015) are used.

![Figure 4. Portal working scheme supporting HEP jobs at Grid and supercomputer resources at NRC KI](image)

Default ATLAS APF supports HTCondor batch system. Additional component to translate jobs to SLURM for the supercomputer was developed. In classical approach each pilot submitting to the batch system slot usually associated with one core. In this case pilots are running locally at working nodes of the Grid cluster or supercomputer.

4.2 Using Portal for Compute-intensive jobs

**Portal and Resources integration scheme**

In case of compute-intensive jobs it is required to exclude ATLAS specifics from the system. Main demand is to support multi core jobs on par with single core.

Portal scheme for compute-intensive jobs at the resources of supercomputer and Cloud-infrastructure at NRC-KI is given at figure 5.
Local PanDA server was installed at NRC-KI. An innovation of this instance is that the single queue for both supercomputer and Cloud-infrastructure was defined to provide additional load balancing at site side. Graphical web user interface and API were developed for job description and submission (see below).

Both supercomputer and Cloud-infrastructure resources at NRC KI are managed by SLURM batch systems. We developed a new module which can launch pilots through batch system to run single and multicore jobs and enable load balancing by redistribution of pilots between resources. Cloud infrastructure as an additional resource is involved when supercomputer is busy or it’s worker nodes don’t correspond jobs requirements. In details this module described below.

Pilot is running on the supercomputer interactive node and communicating with local batch scheduler to manage multicore jobs. We use own common data storage and data management system which are described below. Data management system provides a service which is used by Mover to submit separate jobs to organize input and output data transfer.

**Jobs Definition Interface**

We designed a user friendly web interface to provide more convenient way for job definition. Interface supports user authentication and a graphical menu which allows user to select a distributive, upload input files, specify command to run, and initiate job submission to PanDA workflow. Main features of the interface:

1. Web-interface and API were implemented to interact with scientists and external integrated workflow composers. A new authorization endpoint allows us to remove restrictions of CERN certificates and Virtual Organizations.

![Figure 5. Portal scheme for compute-intensive jobs at the resources of supercomputer and Cloud-infrastructure at NRC KI](image-url)
2. Modular structure supports connection with various external storage systems including ones unsupported by original PanDA. All files passing through job’s lifecycle are described in lightweight file catalog in generalized style (see below).

3. Interface provides monitoring of submitted jobs during their lifecycle, fetches output files and produces report based on log files analysis.

**Pilot Launcher Module with Workload Balancing System**

In case of multicore jobs running one pilot per core is not an option, therefore, a new pilot launcher system was developed. This system is capable of optimizing resources usage by balancing workload between them. In the conventional PanDA scheme shown at figure 1 the information about site loads is only available to the APF module and not the server, which means the latter distributes jobs using information about queues and jobs statistics only. In case of multicore jobs this scheme is inappropriate. Thus, a new functionality to distribute workloads among the computing resources was implemented in the developed pilot launcher.

Queue can be defined to include several computing resources. In this case pilot launcher decides where to launch a new pilot by taking into account the following basic parameters:

- Number of running pilots which have not received job yet. High number means that pilot launching rate should be reduced.
- Number of running jobs and resources allocations per sites. If one of these exceeds site limits then launching pilots may be continued only on the specified resource or will be stopped.
- Priority in resource preference. The resources with higher priorities will be occupied first. For example the Cloud platform will be used only when the supercomputer is overloaded or down if Cloud priority is low.

**Common Data Storage and Data Management System**

Data storage is the central part of distributed computing systems. It’s responsible for reliable storage of user’s input/output files, portal backend system data and logs. Native to WLCG Storage Element is well supported by PanDA, but its architectural complexity makes integration of other external storage elements rather difficult. Our solution has no such drawback.

File transfer systems along with file catalogs are able to move data between lots of distributed nodes with high data rate and to keep consistency of replicas number (For ex.: Rucio (CERN RUCIO, 2015) and FTS3 (Ayllon, Salichos, Simon, & Keeble, 2014)). But it’s hard to adopt such systems for other cases due to their complexity and close association with collaborations. For this reason the new lightweight file catalog and file transfer system were implemented for PanDA to support file transfer by http, ftp, srm and lots of commercial protocols.

5 Approbation

5.1 ATLAS Tasks

Since November 2014 about 200 ATLAS analysis jobs have been successfully completed on daily basis at portal deployed in NRC “Kurchatov Institute”. One of the most important studies conducted on supercomputer is reconstruction of proton-proton events with high pile-up in Transition Radiation Tracker (TRT). Proper understanding of TRT performance at high occupancy is important for different ongoing physics analyses and future work of TRT detector with higher energies during Run 2. This study was successfully done as a qualification task in ATLAS TRT SW group.

Figures 6 displays the statistics of processing ATLAS jobs on NRC KI resources (system jobs are shown in brighter color, user jobs – in darker).
5.2 Genome Analysis Tasks

One of the aims of this project was to run user applications for scientific areas other than HEP where we can use advantages of the architecture of our workload management system. As a pioneer application we chose genome sequence de-novo assembly and short sequence reads to long reference alignment. We prepared software packages Bowtie2 (Johns Hopkins University, 2015) for sequence alignment, ABYSS 1.9.0 (Simpson & al., 2009), SPAdes 3.5 (Bankevich & al., 2012) - for de-novo assembly and QUAST (Gurevich & al., 2013) package for providing detailed and user-friendly reports of de-novo assembly results. SPAdes and Bowtie2 support OpenMP while ABYSS can take advantage of MPI for parallelization.

User-friendly Web-interface was developed to simplify jobs submission and results gathering procedures. Our service was first probed on synthetic data and then we used our de novo assembly service for assembly of microbial reads that were sequenced in Kurchatov institute’s genomics lab. Our web service is actively used by bioinformaticians since it combines computing power of supercomputer facilities at NRC KI with user-friendly application specific web interface.

6 Conclusion

We have designed, developed an integrated portal to submit jobs at Tier-1 grid site, cloud platform and supercomputer. We deployed this portal at NRC “Kurchatov institute” and provided it to run HEP jobs (ATLAS experiment at LHC) and compute-intensive bioinformatics jobs including multicore ones of genome sequencing analysis.

Developed portal supports all types of computing resources. We have adapted workflow management system PanDA for grid/Cloud/supercomputer facilities with shared data storage and extended job lifecycle management schemes to support multicore parallel jobs, developed web interface to simplify job definition and submission by users. We extended PanDA users community beyond LHC by implementation of local authentication mechanism.

Our experience and successful approbation results show that PanDA architecture suits well for integration of heterogeneous resources with various architectures as well as for different compute-intensive sciences.
As one of the next steps in development of the portal, we are planning to increase the volume of computing resources intended for ATLAS tasks. Furthermore, in the near future the portal will start to handle ATLAS Production tasks (currently only ATLAS user analysis tasks are handled).

In order to support more non-HEP tasks the pool of bioinformatic applications will be increased. In particular, new software will be installed (e.g., for ancient DNA sequence data analysis) and the portal will be integrated with existing frameworks for bioinformatics (e.g., GALAXY portal). Also we are going to expand the application area of the portal into other scientific disciplines. For this purpose we are integrating the portal with CLAVIRE system (Knyazkov & al., 2012) to create and manage composite tasks consisting of PanDA jobs sequences.

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. PanDA project 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 by centre for collective usage in NBICS-center within “Kurchatov Institute” (project RFMEFI62114X0006, funded by Ministry of Science and Education of Russia).

References