Project Narrative

Next Generation Workload Management and Analysis System for Big Data

1 Overview

DoE supported data intensive research programs, such as high energy and nuclear physics, astrophysics, earth science, and material science will generate exabytes of data in the near future. The challenges posed by such big data sciences are numerous and not limited to the unprecedented size of the data. Exascale data is often highly distributed and accessed by large international collaborations. A sophisticated Workload Management System (WMS) is needed to manage the distribution and processing of such data. One of the most successful WMS developed in the U.S. is PanDA (acronym for Production and Distributed Analysis System) [1], used by thousands of physicists in the ATLAS [2] experiment at the Large Hadron Collider (LHC) [3]. We propose a program of work here to adapt PanDA for wider use among big data sciences. PanDA would become an enabling technology for scientific discoveries that require access to exascale data.

PanDA delivers transparency of data and processing in a distributed computing environment to ATLAS physicists. It provides execution environments for a wide range of experimental applications, automates centralized data production and processing, enables analysis activity of physics groups, supports custom workflow of individual physicists, provides a unified view of distributed worldwide resources, presents status and history of workflow through a integrated monitoring system, archives and curates all workflow, manages distribution of data as needed for processing or physicist 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 data intensive sciences.

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, and is expected to easily meet the expected continuously growing needs over the next decade. 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 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 a wide variety of exabyte scale sciences.

Funding for the initial development and operation of PanDA was provided exclusively by the Department of Energy (DoE) and the National Science Foundation (NSF). The project began in 2005 as part of the U.S. ATAS program and managed by this proposal's co-PI's K. De and T. Wenaus. At the time, a variety of workload management systems were deployed in ATLAS, based on deployed grid systems, separately for different applications, and separate systems for physicists and central production. PanDA emerged as the best system and was adopted as the default and single WMS for ATLAS before the LHC started operating in 2009. PanDA continues to be primarily supported by DoE and NSF, and managed by the original team of Wenaus and De, while enjoying a large expansion in contributions from many countries in ATLAS. Today, PanDA has grown to support all distributed workflows in ATLAS, and enjoys a huge user and support base worldwide.

Through a work package supported by the Open Science Grid (OSG), PanDA was adapted and successfully used for molecular dynamics simulations of protein folding using the CHARMM molecular modeling software [4], leading to a publication [4a] describing the implementation as general, flexible, easily modifiable for use with other molecular dynamics programs and other grids and automated in terms of job submission, monitoring, and resubmission. In 2011, the Alpha Magnetic Spectrometer (AMS) experiment [5] began testing PanDA for possible adoption as WMS. AMS dedicated a 3 FTE effort in 2012 to install PanDA in the AMS Payload Operations and Computing co-Center. AMS Physicists successfully adopted PanDA and conducted their first Monte-Carlo simulation campaign in April 2012. Now AMS is setting up the infrastructure and reconfiguring computing facilities to have PanDA as the primary WMS of the experiment. Recently, the Compact Muon Solenoid (CMS) experiment [6] has expressed interest in using PanDA for data analysis by thousands of physicists on their experiment. The CERN IT group is conducting an evaluation of PanDA for suitability to the CMS workflow. The first results of the evaluation were presented in April, 2012, with the conclusion that there is no show stopper to use PanDA as an analysis framework for the CMS experiment. The feasibility study results will be reported in the CHEP 2012 conference. [7,8]

The recent interest in PanDA by other big data sciences provides the primary motivation for this proposal. With some additional effort, PanDA can be generalized for other user communities. It is relatively straightforward to take the core components of PanDA and re-factorize them in an experiment neutral package. The ATLAS specific parts can then be layered as customizations. This will benefit ATLAS by making the PanDA code easier to maintain and evolve. More importantly, this will allow PanDA to be adapted for a variety of use cases. Other sciences will be able to customize and deploy PanDA quickly and easily.

2 Background

The largest scientific instrument in the world, the Large Hadron Collider (LHC), is operating at the international CERN Laboratory in Geneva, Switzerland. Experiments at the LHC explore the fundamental nature of matter and the basic forces that shape our universe. ATLAS, one of the largest collaborations ever assembled in the sciences, is at the forefront of research at the LHC. To address an unprecedented multi-petabyte data processing challenge, the ATLAS experiment is relying on the computational grid infrastructure deployed by the Worldwide LHC Computing Grid (WLCG) [9]. WLCG is by far the largest academic distributed computing environment in the world consisting of hundreds of distributed computing centers. Using the massive data processing power of the Grid, more than 8000 scientists analyze LHC data in search for new measurements and discoveries in physics.

Thanks to the outstanding LHC performance in 2011-2012, ATLAS manages over fifty petabytes of data on the grid. ATLAS leads the WLCG usage in the number of jobs, processed data volume and in core-hours, as shown in Fig. 1.

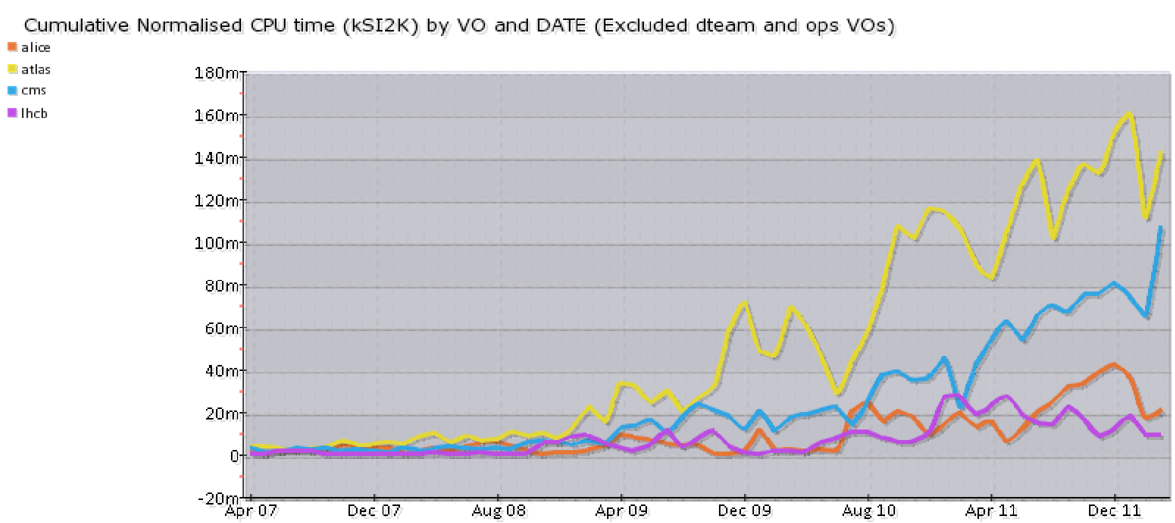

Figure 1: ATLAS dominates the WLCG normalized CPU usage over the past five years.

The ATLAS experiment uses PanDA for managing the workflow for all data processing on the WLCG. The scale of the workload is given by the following metrics: between 0.5 and 1 million jobs per day (one ATLAS "job" is a 32-bit Linux Python/C++ data processing application using up to 2 GB of memory on a single core for up to 12 hours, processing a maximum of several gigabytes of input data and producing output data of a similar volume); and continuous worldwide data transfers at the level of 100 Gbps on aggregate. It takes three million core-hours for the first step processing of one petabyte of ATLAS data with about one billion collision events from the LHC. It took four weeks to complete the full MapReduce-like task of reconstructing one billion events. The combined ATLAS workflow used more than 100k CPU-cores, consuming at peak about 0.2 petaflops — a performance on the Top100 list of supercomputers.

Through PanDA, ATLAS physicists see a single computing facility that is used to run all data processing for the experiment, even though the data centers are physically scattered all over the world. Central computing tasks (Monte Carlo simulations, processing and re-processing of LHC data, re-processing of MC simulations, mixing and merging of data, and other tasks) are automatically scheduled and executed. Group production tasks, carried out by groups of physicists of varying sizes, are also processed by PanDA. User analysis tasks, providing the majority of activities by individual physicists leading to scientific publications, are seamlessly managed.

WLCG computing facilities in ATLAS are organized into Tier's. CERN is the source of all primary data, referred to as Tier 0. There are 10 Tier 1 centers. Each Tier 1 center hierarchically supports 5-18 Tier 2 centers in a cloud. PanDA is deployed at all ATLAS Tier 1 and Tier 2 centers. In Fig. 2 we show the

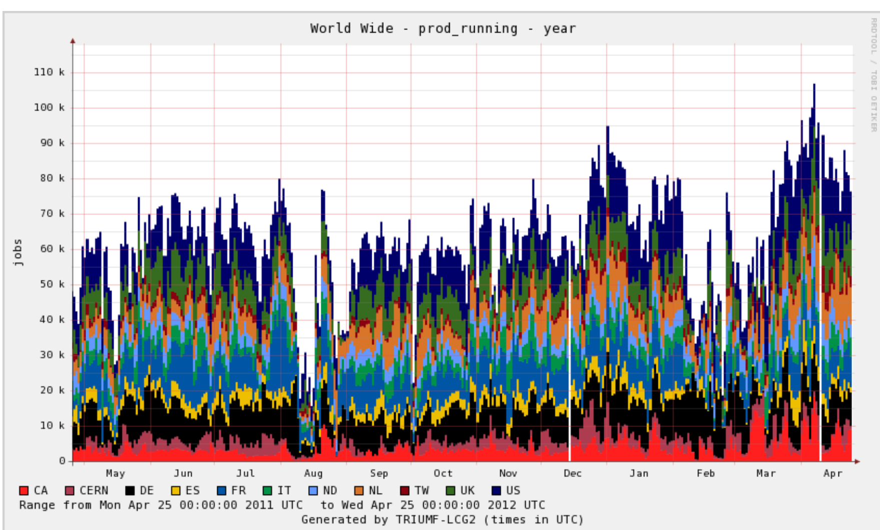

Figure 2: Snapshot of number of running production jobs in ATLAS on WLCG resources.

number of concurrently running jobs managed by PanDA over the ten regional groups of Tiered centers during the past one year.

Each PanDA site provides a grid accessible Compute Element (CE) and a Storage Element (SE). Pilot jobs are continuously and automatically scheduled at the CE of each site. When the pilot jobs start execution, they contact PanDA Apache servers, which then dispatch the execution workload. PanDA maintains a central database of all activities, and consequently a central queue of all work-flows. This architecture provides an integrated view of all resources managed by PanDA. The pilot based system also enables integration of non-grid based resources. Local resources at universities are integrated using local pilot submission factories. New cloud-based resources are also added to PanDA using the CE model.

The SE plays a central role in the PanDA work-flow. For central computing tasks, input data is asynchronously staged in and output data is staged out to the SE. The Tier 1 hierarchy is maintained for all workflow of these types of tasks. However, the user analysis workflow is different. Processing always goes to the location of the data. Both work-flows are automatically managed by PanDA. The ATLAS data management system DQ2 is used by the PanDA system for all data registration, data discovery, and data movement. A large variety of SE's are supported by PanDA across the hundreds of computing sites. Recent work has focused on supporting Federated Storage, specifically through XRootD. The PanDA workflow for central production jobs is shown in Fig. 3.

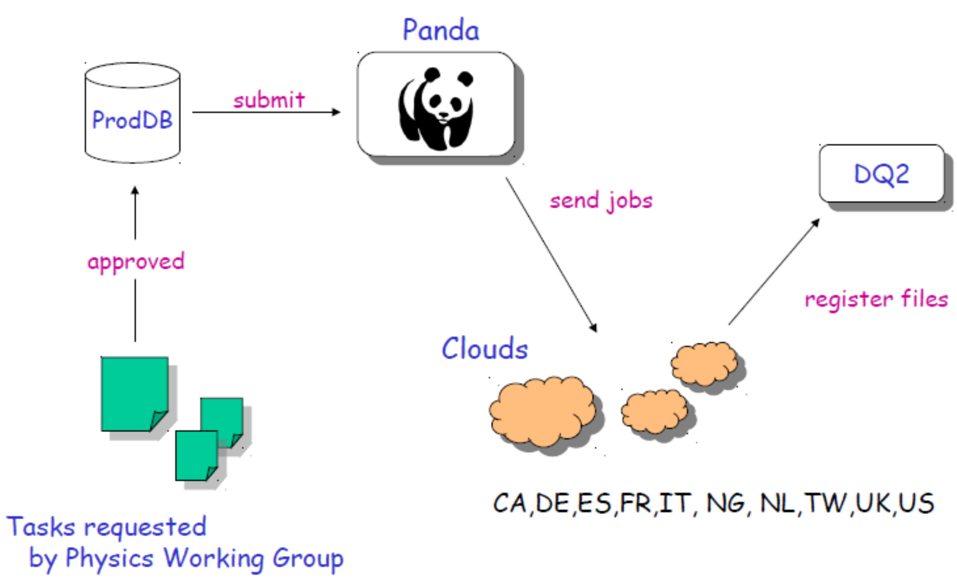

Figure 3: Example of PanDA workflow

The system supports both managed production and individual users (analysis) so as to benefit from a common WMS infrastructure and to allow analysis users to leverage production operations support, thereby minimizing overall operations workload. The common PanDA architecture is shown in the Fig. 4 below.

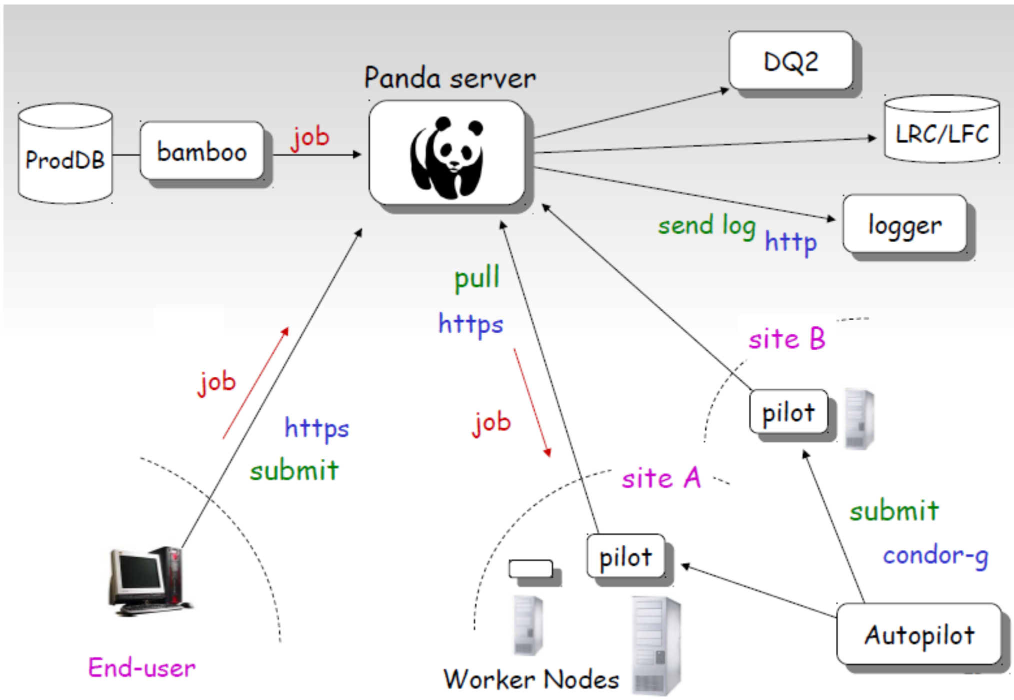

Figure 4: Schematic of PanDA architecture

Recently PanDA has been extended to support the Amazon Elastic Computing Cloud (EC2) [11] and other cloud resources, with prototype cloud-resident PanDA sites operating for both analysis and production. PanDA jobs have run successfully in the Amazon cloud, Magellan cloud, and the CERN lxcloud. Also an academic cloud in Canada was used in ATLAS production and supported by PanDA [12,13,14].

The diversity of computing platforms and fabrics available for scientific computing – a diversity which cloud resources are only increasing, at least in the near term – 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 PanDA systems code that is independent of the underlying middleware and facilities actually used for processing in any given environment. 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 WMS for many data intensive sciences. PanDA lowers the barrier for scientists to easily carry out their research using a variety of distributed computing systems.

3 Proposed Research and Methods

PanDA's capability for large scale data-intensive distributed processing has been thoroughly demonstrated in one of the most demanding computing environments in large scale science. PanDA processes a diverse range of workloads, over 200 million jobs/year, on over 100k job slots worldwide. Thousands of physicists use it for their personal processing needs. We propose to leverage this capability by generalizing PanDA from its present ATLAS- and HEP-oriented scope to an easy to use meta-application providing location transparency of processing and data for other data-intensive sciences and a wider exascale community.

3.1 PanDA components

PanDA consists of several subsystems. Each of these will be factorized into general components and customizable layers. The experiment specific layers will be configurable. Advanced features will have sensible defaults and can be turned on for demanding applications. The main components of PanDA to be factorized are:

- **The PanDA server**. The central components of the server are implemented in python and run under Apache as a web service (in the REST sense; communication is based on HTTP GET/POST with the messaging contained in the URL and optionally a message payload of various formats). The server is the heart of the PanDA system and will be factorized as a general WMS service.
- **PanDA database**. A system-wide job database that records comprehensive static and dynamic information on all jobs in the system. Oracle databases implement the job queue and all metadata and monitoring repositories (MySQL is supported as an alternative database backend). To users and to PanDA itself, the job database appears essentially as a single attribute-rich queue feeding a worldwide processing resource.
- **The PanDA pilot system**. 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. Analysis pilots are able to switch their identity on the worker node to that of the job submitter, using the glexec tool, should site security policies require it.

- **PanDA 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. This is a key component of the PanDA workflow automation.
- **The PanDA 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.
- **Auto pilot factory**. 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 once the pilot successfully launches on a worker node). A variety of pilot delivery mechanisms are supported: local cron, CondorG, and glideInWMS.
- **PanDA information system**. A system-wide site/queue information database recording static and dynamic information used throughout PanDA to configure and control system behavior from the region level down to the individual queue level. The database is an information cache, gathering data from grid information systems, the data management system and other sources. It supports information access and limited remote control functions via an http interface. It is used by pilots to configure themselves appropriately for the queue they land on; by PanDA brokerage for decisions based on cloud and site attributes and status; and by the pilot scheduler to configure pilot job submission appropriately for the target queue.

- **Job submission interface**. A simple, python based client interface allows easy integration with diverse front ends for job submission to PanDA. Pathena, Prun, Ganga, and scripting languages are the most common front ends in use. A personal job manager, Pbook, is available to users.
- **Workflow management**. Support is provided for usage regulation at user and group levels based on quota allocations, job priorities, usage history, and user-level rights and restrictions. Workflow regulations can be applied globally or locally.
- **PD2P – Dynamic data caching**. The ATLAS experiment produces a wide range of data types, from RAW LHC data to simulated data, from processed physics data to reduced analysis streams. At the startup of LHC collisions, the experiment employed a static data distribution model. Multiple copies of all data types were generated and distributed automatically using pre-defined shares to all sites. The static data distribution model proved to be highly inefficient. Storage systems filled up rapidly with seldom accessed data. The data distribution strategy was manually optimized several times, but the revisions could not keep up with changing analysis patterns. After a few months, a new system based on PanDA was proposed by the participants in this proposal, and deployed at all US sites. This system named PD2P (for PanDA Dynamic Data Placement) proved to be extremely useful. It has become the default ATLAS data distribution model for all sites. PD2P distributes data using a caching policy. More copies are made of frequently used data. Frequency of use is tracked by PanDA and checked for all new user tasks. Processing bottle-necks will also trigger additional copies. Production tasks have always moved and cached data as needed in PanDA, since the work flow is fully defined. User analysis is chaotic and benefitted enormously from the introduction of PD2P. Initially, PD2P was used only for Tier 2 sites. Since the beginning of the 2011 LHC run, PD2P is also used for all Tier 1 sites.
- **PanDA monitoring**. A comprehensive monitoring system supporting production and analysis operations; user analysis interfaces with personalized 'My PanDA' views; detailed drill-down into job, site and data management information for problem diagnostics; usage and quota accounting; and health and performance monitoring of PanDA subsystems and the computing facilities being utilized.

In addition, PanDA will be enhanced with a new component - the Dynamic Evolution for Tasks (DEfT). In Big Data processing the task became a main unit of computation (instead of a job). A computational task is a collection of similar jobs that could be executed in parallel. In petascale data management scientists deal not with individual files but with large datasets - collections of similar files.

Similarly, a task – not a job – is a major unit in Big Data processing. Splitting of a large data processing task into jobs (small data processing tasks) is similar to the splitting of a large file into smaller TCP/IP packets during the FTP data transfer. Generally, during a file transfer, application scientists are not concerned how many TCP/IP packets were dropped. The network performance is an area of concern of the network researchers and engineers. Similarly, scientists do not care about transient job failures, when PanDA data processing delivers "six sigma quality" performance for the petascale data processing campaigns with hundreds of tasks [15].

3.2 Adding network awareness to PanDA WMS

We would like PanDA to benefit from new networking capabilities that are making their way into production services on research and education (R&E) networks globally. These capabilities include:

- Extensive standardized monitoring data from network performance monitoring (perfSONAR) instances located within backbone networks, national networks, regional networks and end-sites.
- Traffic engineering capabilities that allow transparent re-routing of high impact flows onto separate infrastructures.
- Dynamic circuit capabilities allowing the temporary construction of point-to-point network circuits with dedicated bandwidth for specific durations
- Intelligent networking systems (such as VNOD (Virtual Network On Demand) [16]) that have capabilities to create on demand virtual network domains with necessary bandwidth guarantees along hops. User is oblivious to the underlying physical topology of virtual network and sees the network as multiple point-to-point virtual links.

The question for globally distributed, data-intensive, scientific collaborations is how best to take advantage of these capabilities to improve their ability to do their science? We will provide a means of doing this for the ATLAS collaboration. Evolving network services and capabilities are transforming what was previously an opaque networking infrastructure into something visible, manageable and potentially deeply integrated into distributed data and computing architectures. Many of the research efforts into dynamic network provisioning, quality of service and traffic management have made their way into production services on major research and education networks globally. Sorely lacking are the higher-level services and capabilities required to:

- Integrate these services within existing and evolving scientific infrastructures. As an example, PanDA/PD2P currently does not interface with any of the advanced network provisioning technologies available.
- Provide intelligent decision support for when such services are beneficial to the task at hand.
- Automate the discovery and use of such capabilities transparently to the scientists using the infrastructure. As an example, instead of exposing intelligent network services directly to scientists, PanDA/PD2P can be updated to directly interface with them without any involvement of scientists.

As part of this proposal, we intend to develop tools that will discover services and capabilities to support specific workflows depending on the involved components and locations. Monitoring information from the involved components, including the network, will serve as an input to a software agent that can optimize various tasks based on the input. This requires minimal interaction from the scientist invoking the task. As a specific example, scientists sometimes have prior knowledge of data transfers that are to be carried out as part of the workflow. In such situations, PanDA/PD2P can significantly benefit if it utilizes the services of intelligent QoS systems such as VNOD. PanDA can provide the data volumes and any transfer deadlines to the software agent, which in turn will forward the knowledge about future data transfers to VNOD. VNOD can then intelligently provision the network for appropriate amount of bandwidth that is required to meet the data transfer deadlines. The VNOD system has access to the time varying bandwidth availability along each component that connects two end points of data transfer. Once deployed, the VNOD system exposes an API that can be used by the software agent to interact with it. To illustrate the benefits of intelligent network services for PanDA, we now present three usage examples:

1) A physicist needs to quickly re-run a prior work-flow because of bug discovered in a reconstruction algorithm he or she used. Unfortunately the bug was discovered not very long before the physicist needs to report about his work at a conference. Site A has enough available job-slots for that physicist to run on but is missing required input datasets. Sites B, C and D have a backlog of jobs queued and the required datasets are distributed among these three sites. In addition to regular best-effort networking, all sites have access to VNOD and ESCPS (End Site Control Plane Service) [17] resources that are capable of creating dynamic network circuits on-demand. A decision agent may recommend that jobs be sent to Site A. At the same time, VNOD along with ESCPS can be used to build virtual circuits connecting sites B, C, and D to site A. VNOD

will make sure that such virtual circuits have sufficient bandwidth reservations so that Site A gets uncongested access to needed input data from other sites. If the Sites also supported network-aware storage resource management, I/O bandwidth on the storage systems involved could also be co-scheduled to ensure data would be readily accessible at the rate needed by the jobs on Site A.

2) The job-management system is showing that a number of sites are building up a backlog of tasks. Quick analysis shows a number of sites are using a common congested link in trying to transfer back output datasets to their destinations. Using available network services present at some of the sites involved will allow the construction of dynamic circuits for some of the data-flows. Such circuits can guarantee a specific minimum bandwidth and move that data onto another path. This will remove the congestion on the common link. The backlog can be quickly drained and the system restored to normal operation.

3) HEP computing is often described as an example of an embarrassingly parallel workflow. Indeed, it appears so on the scale of one worker node which does not communicate with other nodes during job execution. However, the large scale global workflow is highly interconnected, as each worker node job typically does not produce an end result in itself. Rather the data produced by a job serve as input to a subsequent job in the workflow. These workflows require a complex mesh of global interprocess communications during large scale data processing. PanDA already handles this interprocess communication and managed workflow extremely well. The new intelligent network services will make it possible to dynamically create the needed data transport channels on demand to increase workflow throughput and efficiency.

Initial agent and tool developments will focus on discovery and categorization of applicable services that may exist. The early versions of the system will present options to the physicists about capabilities that exist and how these may be glued into the task to improve its speed or resiliency. At this stage, the agent will be more advisory than automated and the physicists will need to manually alter their work-flow to take advantage of any applicable services. To better support automation of the use of network services, we will need to identify locations in the existing and evolving ATLAS distributed data and computing architectures where service call-outs need to be added to allow integrated, transparent use of network services. The next stage of development will more closely tie the agent to the underlying experimental software stack. Physicists will still need to interact with the agent but it will automate much more of the process of activating specific services deemed useful to support specific tasks. Deeper integration with the experiment's software architecture will allow the agent to gather most of its needed inputs for decision support and infrastructure provisioning without requiring physicists to provide it. This will improve transparency and provide ease-of-use while reducing misconfigurations in using PanDA. The agent will be instrumented to gather relevant data about its performance and the impact of its recommendations and service invocations. These data will be used to help steer future development of the agent, targeting the creation of an increasingly effective expert system to reduce the amount of user interaction required. The goal is a fully automated system capable of discovering service options and invoking them, as appropriate, to improve the overall physicist experience in using the available infrastructure.

Currently PanDA does not use information about network resources in workflow management. This is not an issue in static data distribution scenarios like the ATLAS planned data replication model. However, this is a disadvantage for dynamic data distribution, because the workflow does not take advantage of advanced networking capabilities to intelligently provision the network for extreme scale data transfers. The best effort data transfer adds uncertainty into the workflow of data processing, and unpredictable job starting and completion time. Large scale resources such as the DoE leadership computing facilities typically use time shares and CPU hours defined by INCITE proposals [18]. Without predictable data stage-in and out, users are at risk of using time shares inefficiently due to idling resources or incomplete jobs when the time share is up. In such cases it is not efficient for PanDA to rely on the existing best effort network to utilize the shared computing and storage resource with a prior-arranged starting and completion time.

Under a new network-aware workflow we propose to implement, PanDA will use information on how much bandwidth is available and can be reserved before requesting dedicated circuits for data movement. Users will specify as part of their task definition the volume of data they need to transfer and a deadline by which they would like data to be transferred. The calculations of (i) how much bandwidth to reserve, (ii) when to reserve, and (iii) along what path to reserve will be carried out by an intelligent network system such as VNOD.

We have past experience in developing technologies such as ESCPS (End Site Control Plane Service) and VNOD that can intelligently construct end-to-end network paths and topologies with guaranteed bandwidth reservations. Such intelligent network systems will be extremely beneficial to the next generation of workload management systems as it improves the reliability of when data transfers can be expected to finish. Intelligent network systems such as VNOD interact with wide area networks (ESnet and Internet2) as well as the local area networks of the end sites to provision true end-to-end network paths. ESnet is a wide area network that connects all the DoE national laboratories. Internet2 is a wide area network that connects educational institutions, and has peering points with ESnet.

3.3 Use of Leadership Computing Resources

PanDA has thus far not been used at Leadership Computing Facilities. An objective of the proposed work is to change this, adding such resources to those supported by PanDA. We propose to prototype PanDA workflows appropriate to such facilities (leveraging high bandwidth between nodes) on the BlueGene/P platform at ANL, taking advantage of the available RDMA interfaces between the nodes. An attractive option is presented by the open source software developed in the framework of the IBM Kittyhawk project [19] and related DoE ASCR-funded projects. While in the short term PanDA capability at Leadership Computing facilities is not likely to see significant usage for HEP computing, present trends in computing technology to which HEP must respond – in particular many-core processing with memory/core an increasingly scarce and expensive component – will drive HEP computing and software towards fine grained concurrency and tightly managed memory requirements which will make Leadership Computing facilities of much greater interest for HEP computing in the future. Developments in these directions are presently underway in the LHC experiments.

Aligned with these developments, we will prototype a PanDA workflow for Monte-Carlo simulations at the leadership-class computing facilities. For example, the Geant4 [20] simulation of LHC physics is constrained by the computational resources only. A Monte-Carlo simulation job of minimum-bias events uses only 1 GB of memory per core to produce a 20 MB file within about 2 hours. This corresponds to a dataflow of 20 kbits/s per core, which is within the throughput limits of inter-process communications via RDMA interfaces between the nodes at the leadership-class computing facilities. Mapping the PanDA Monte-Carlo simulations workflow requires transformational changes to ATLAS Computing Model. The research proposed in this area intend to hide the complexity of migration to leadership-class computing facilities for major HEP computationally intensive tasks behind the common PanDA interface.

3.4 Beyond ATLAS to AMS

The Alpha Magnetic Spectrometer, also designated AMS-02, is a particle physics experiment module that is mounted on the International Space Station. Similar to accelerator-based experiments, the data collected from the detector undergoes reconstruction and analysis utilizing sophisticated software and large scale computing facilities. Both reconstruction and analysis depend on extensive simulations which likewise utilize complex software and considerable resources. Efficient use of, and "just in time" access to such resources presents a challenge to any project, AMS being no exception.

In recent months, AMS has been investigating feasibility of adapting Panda for the needs. As a pilot project, the AMS Payload and Computing co-Center at Academia Sinica in Taiwan has been hosting a cluster comprising about 500 CPU slots, which was configured as a PanDA resource. Submission of pilot jobs was configured and monitored by ATLAS personnel at Brookhaven National Laboratory, while AMS researchers activated the payload jobs from Taiwan and CERN using standard command-line tools developed for PanDA. The first stage of the project has just been completed, with a batch of 500 simulation jobs executed successfully, while being monitored using standard PanDA web interface.

The next steps in this project are:

- automation of PanDA job definition and integration with the existing AMS web services, which will make the simulations workflow efficient and easy to manage
- providing convenient mechanism for data movement across the Grid (such as based on Globus Online facility), which will make the data easy to place and share across AMS sites regardless of the location of the computing resource where it originated

4 Scientific and technical merit of the project

The experiments at the LHC are probing the fundamental laws of nature at the highest energies available. The LHC program is one of the highest priority projects funded by the DoE. Already, the LHC is operating at 8 TeV collision energy, which is four times the energy previously available at particle accelerators. Over the next few years this will increase to 14 TeV. These energies open up the exciting potential for fundamental discoveries, while continuing with the exploration of the current Standard Model of physics. Two of the longest standing mysteries of nature which may be finally unlocked are:

- The mystery of mass. The Higgs particle was proposed over 40 years ago to account for the mass of elementary particles. Experimentally, the Higgs particle has never been confirmed. The higher energies and very high luminosities available at the LHC may finally lead to the discovery of the Higgs particle.
- The mystery of Dark Matter. Astronomical observations over 70 years ago hinted at the existence of dark matter in the universe. Subsequent observations have confirmed that the vast majority of matter in the universe is dark. ATLAS, CMS and AMS are actively searching for the fundamental particles of dark matter. Supersymmetry is a theory proposed more than 40 years ago, which may hold the clue to dark matter. Experimentally, supersymmetry has never been observed. It is a top priority for the LHC.

The mysteries of mass and dark matter are just two examples of the exciting physics potential of the LHC experiments. AMS also provides a rich menu of physics. The scientific communities which use PanDA or are planning to use PanDA will pursue many other measurements and discoveries over the next decade. The benefits to the basic physical sciences are innumerable.

Through the exciting physics goals of the LHC, PanDA already enables cutting edge scientific results in support of a large scale scientific collaboration. It has proven to be a highly successful software product funded by the DoE which is in active use by the ATLAS experiment. We can leverage this successful example for other data intensive sciences. PanDA will address the needs of a wide range of scientific communities with the work proposed here. This is an exciting synergy between basic science supported by the DoE and technology derived from DoE projects.

The work proposed here will enable the use of PanDA by new scientific collaborations and communities as a means of leveraging extreme scale computing resources with a low barrier of entry. Particular attention will be given to enabling the computing infrastructures supported by the DoE that are not yet supported by PanDA such as the Leadership Computing facilities. The work will both draw on and enhance the development and use of PanDA for ATLAS, factorizing the system and expanding its scope in ways most relevant to usage beyond ATLAS but which will also bring benefit to ATLAS by turning PanDA into a more modular and maintainable common software system. The technology base provided by the PanDA WMS will enhance the usage of a variety of computing resources available to basic research supported by the DoE.

Support for Leadership Computing facilities will expand the potential user community for PanDA and will also, even in the near term, benefit ATLAS by running ATLAS jobs on Leadership Computing facilities with direct access to the data hosted by US National Laboratories, such that ATLAS can dynamically acquire 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 such as AMS and LSST [21,22] to use PanDA.

In the longer term, exascale facilities and cloud platforms could become a real alternative to very large data centers owned and managed by the scientific community. This work would create in PanDA a WMS system enabling users to bridge the near and longer terms smoothly, with minimal effort, and preserving the same global system view throughout, by leveraging PanDA support for both present and emerging computing infrastructures. ATLAS would certainly lead the way in utilizing exascale platforms this way, and we would seek through this work to make it an easy path for others to follow as well.

5 Project work Plan.

5.1 Milestones and deliverables

The work proposed here will take place as an integral part of the PanDA project in order to maintain the overall coherence of the product and in order to maximally leverage core PanDA effort and expertise for this project, and vice versa. The co-PI's in this proposal include the principals in the management, development and operation of the PanDA project. Evolution of PanDA to meet the future challenges of ATLAS is ongoing. The work proposed here in terms of both personnel and work program will be incremental over the existing PanDA project, in order to sustain the focus of present PanDA developers on ATLAS needs, and to ensure developers for this work program are not distracted by ATLAS needs. The work program and deliverables fit harmoniously into the overall PanDA project. The current development team, who are supported by the DoE and NSF through the U.S. ATLAS project, are all affiliated with the primary institutions in this proposal. Their work will continue. In our proposal we propose the addition of a few key hires to work with the current developers in refactoring PanDA into an easily deployable, generalized workload management system for data intensive processing at extreme scales.

An attraction of this ASCR call for us – a mixed HEP/CS team focused on extreme scale computing as an active research tool – is that it is directed at practical objectives and invites building on existing practical successes. It emphasizes ease of use across disciplines, giving exascale access to the scientific computing masses. Within ATLAS and PanDA it is a perfect time and opportunity for such an endeavor, as ATLAS enters a development round directed at PanDA extensions for ATLAS beyond 2014, and possibly for wider LHC adoption as well. We believe that the LHC use case, scale and success may catch the wider scientific community's imagination and become visible enough to encourage others to use tomorrow's exascale computing and networking capabilities for extreme-scale science, particularly if we can make use of our experience and software to offer others a lower threshold of entry.

We have identified the following four work packages over the next three years period. The work packages, the primary institution which will execute each package, and the PI responsible for each work package are indicated below:

- WP1 (Factorizing the core): Factorizing the core components of PanDA to enable adoption by a wide range of exascale scientific communities (UTA, K.De)
- WP2 (Extending the scope): Evolving PanDA to support extreme scale computing clouds and Leadership Computing Facilities (BNL, S.Panitkin)
- WP3 (Leveraging intelligent networks): Integrating network services and real-time data access to the PanDA workflow (BNL, D.Yu)
- WP4 (Usability and monitoring): Real time monitoring and visualization package for PanDA (BNL, T.Wenaus)

These four packages optimally organize the required work. The participants in this proposal will work together on the software development, deployment and testing. While the developers will be physically located at two different locations, they will communicate frequently and meet regularly. This mode of collaborative software development works well within ATLAS, as demonstrated by the success of the PanDA development team over the past six years, where the developers are distributed globally among various locations. The PanDA development team has benefited from the unique expertise available at each location, while working coherently as a group. We describe each of the work packages in further detail below.

5.1.1 Year 1. Setting the collaboration, define algorithms and metrics

Year One will focus on team and development infrastructure building, factorization design, extensions to support cloud resources, planning of extensions to support Leadership Computing facilities, and monitoring design for ease of use and user customization.

WP1 (Factorizing the core)

- Establish the collaborative teams to carry out this work program integrated with overall PanDA development, making the requisite hires
- Design the factorization for each of the core PanDA components described in this proposal, to isolate generic functionality from user specifics
- Prioritize the component factorizations into three priority phases
- Define components, interfaces and APIs by which user specific components configure and utilize PanDA core services
- Implement the priority 1 factorizations and test with prototype workflows
- Implement a code development and packaging infrastructure supporting both core development and multiple independent user community developments

WP2 (Extending the scope)

- Implement in concert with the core PanDA team an extended site/facility definition and configuration capability allowing user communities to autonomously add new sites/facilities to the list of known PanDA sites
- Also in concert with the core PanDA team, identify and implement the extensions to PanDA infrastructure and services required for seamless support of general cloud-based resources as PanDA sites serving arbitrary PanDA user communities
- Implement and demonstrate ATLAS and non-ATLAS usage of cloud-based PanDA sites (commercial and/or research clouds)
- Identify any PanDA core infrastructure and service extensions required for basic utilization of Leadership Computing facilities as PanDA sites, and organize their implementation

WP3 (Leveraging intelligent networks)

- Design a set of client API and user interface to allow Panda to specify as part of their task definition the volume of data they need to transfer and a deadline by which they would like data to be transferred.

- Design internal network planning algorithm to calculate (i) how much bandwidth to reserve, (ii) when to reserve, and (iii) along what path to reserve

WP4 (Usability and monitoring)

- Design a factorization of the PanDA monitor into ATLAS-specific and generic components over a common foundation
- Design a generic PanDA monitor browser view and skeleton from which experiment (ATLAS and other) browser views are derived customizations
- Design generic components and APIs to be used by user communities to easily implement and customize their monitoring views and workflow visualizations
- Define database schema and information exchange extensions and revisions required to support modularization and factorization
- Establish MySQL as a full alternative to Oracle for PanDA active job monitoring
- Implement monitoring/visualization services for the priority 1 set of factorizations in WP1
- Produce documentation and examples for monitoring component architecture and customization
- Implement a prototype demonstrating monitoring supporting the priority 1 factorizations

5.1.2 Year 2. Prototyping and Implementation

Year Two will follow up on the design and early development work of Year One. Throughout the year extensive prototyping and code development will occur. Deliverables will be moved to pre-production, and we will start to measure and tune system performance. The first phase extension to Leadership Computing facilities will complete, and a second phase to support more complex workflows will be initiated.

WP1 (Factorizing the core)

- Implement the priority 2 and 3 factorizations, testing each with prototype ATLAS and non-ATLAS workflows
- Coordinate with ATLAS and the core PanDA team the ATLAS migration to use of the priority 1 set of factorized user components/APIs

WP2 (Extending the scope)

- With the core PanDA team, complete the implementation of core infrastructure and service extensions required for basic utilization of Leadership Computing facilities as PanDA sites
- Implement and demonstrate ATLAS and non-ATLAS usage of PanDA sites serving at least one Leadership Computing facility
- With the core PanDA team, identify and implement any extensions to PanDA infrastructure and services required for the support of Leadership Computing platform applications that utilize multi-core and/or multi-node batch slots and/or high-bandwith inter-core communication (such as MPI)

WP3 (Leveraging intelligent networks)

- Integrate PanDA prototype with RDMA access to the experiment data via WAN.
- Leverage and integrate existing intelligent network services to support time bound and reliable data movement between end hosts, and implement the internal function components to support the network API for PanDA.
- Create a fallback mechanism so that when external intelligent services are not available, simple scheduling decisions can still be made.

WP4 (Usability and monitoring)

- Implement monitoring/visualization services for the priority 2 set of factorizations in WP1, and accompanying documentation/examples
- Design, develop and document system usability enhancements through monitoring based on the component based monitor, for non-ATLAS user communities
- Design the advanced networking component of the monitor and integrate it within the framework
- Tune and optimize database schemas, transactions rates and monitoring interfaces
- Migrate ATLAS monitoring in concert with the migration to priority 1 set of factorized components

5.1.3 Year 3. Production and Operation.

Year Three will complete the PanDA factorization work, its deployment and document, and shift emphasis to supporting its adoption. Support for sophisticated workflows at Leadership Computing

Facilities will be completed, and deployed for PanDA production together with cloud resources. Overall system packaging and documentation will be completed.

WP1 (Factorizing the core)

- Coordinate with ATLAS and the core PanDA team the ATLAS migration to use of the priority 2 and 3 sets of factorized user components/APIs
- Complete and deploy a pre-production PanDA instance utilizing the fully factorized code base and operating at full ATLAS scale, shadowing the production PanDA service
- Debug, tune and harden components, APIs and the integrated factorized system
- After validation, deploy factorized PanDA into ATLAS production
- Complete the documentation, usage examples and operating generic prototypes for non-ATLAS use of the fully factorized PanDA.
- Support communities adopting/considering PanDA in evaluation and implementation

WP2 (Extending the scope)

- Implement and demonstrate ATLAS and non-ATLAS usage of PanDA sites serving a second Leadership Computing facility
- Implement and demonstrate a PanDA site at a Leadership Computing facility capable of supporting multi core/node batch slots and inter-core communication/coordination
- Deploy and operate PanDA as a production service in Leadership Computing and cloud environments

WP3 (Leveraging intelligent networks)

- Incorporate resiliency such that if bandwidths cannot be reserved, PanDA can still transfer data over available network connections.

WP4 (Usability and monitoring)

- Tune and improve system usability and monitoring coverage based on non-ATLAS feedback
- Migrate ATLAS monitoring in concert with the migration to priority 1 set of factorized components
- Support communities adopting/considering PanDA in evaluation and implementation

6 Budget request

We estimate that three full time programmer's will be needed to carry out the work proposed here. 1.5 FTE will be based at BNL, and 1.5 FTE at UTA. These hires represent the bulk of the budget. The three programmers are expected to be primarily new hires, though the proposed funding may be used to redirect experienced software developers already integrated into PanDA to this project. At UTA, 1 FTE is expected to be a person trained in Computer Science Engineering, while the 0.5 FTE will be a physicist with strong computational skills and interest. Proposed personnel at BNL are also expected to have a mix of CS and HEP training. In addition, we request funds to support a CS graduate student at UTA.

7 Collaboration and Management Plan

Lead Institution: Brookhaven National Laboratory (submitting this proposal):

PI's: Alexei Klimentov*, Sergei Panitkin, Torre Wenaus, Dantong Yu

Argonne National Laboratory (collaborating institution - not submitting proposal)

PI: Alexandre Vaniachine

The University of Texas at Arlington (collaborating institution - submitting this proposal):

PI's: Kaushik De, Gergely Zaruba

*Alexei Klimentov will be the lead PI for this proposal.

The collaborative research proposed here will be primarily executed by three groups; the Brookhaven National Laboratory Physics Application Software Group, BNL Computational Science Center and the University of Texas at Arlington (UTA). Other partners (who are not funded through this proposal) will be the US ATLAS Computing Program, OSG, WLCG and other collaborators from the ATLAS experiment.

The work proposed here will be carried out by the core PanDA development team: which is a collaboration between BNL and UTA. The PanDA project leaders are Torre Wenaus from BNL and Kaushik De from UTA, both of whom are co-PI’s in this proposal. Dantong Yu from BNL will lead the networking part of the project and he is co-PI of this proposal. The lead PI and point of contact will be

Alexei Kilmentov, who is currently leading the BNL Physics Application Software group. While the current PanDA development team is focused 100% on the support of ATLAS, we are confident that the proposed addition of new people requested in this proposal will be sufficient to develop PanDA for other data intensive sciences.

The BNL Physics Applications Software (PAS) group is playing a leading role in the ATLAS Distributed Computing and has years of experience in Grid computing development, deployment and successful operations on the multi-petabyte scale. PAS members are core developers of PanDA and other Grid applications.

BNL's Computational Science Center develops high performance data transfer tools (such as rFTP100), which enable the proposed data processing on the foundation of the DOE 100G network infrastructure. The Center has extensive experience in high-performance computer networking, data mining, and large-scale data management over the WAN. BNL established contacts with Google and researchers from other DOE Labs who expressed interest in the development of enabling technologies for data-intensive computing beyond the petascale.

UTA researchers are involved in ATLAS Software design, commissioning and deployment from the earliest days of the collaboration. They bring a strong synergy between HEP and CS expertise to this proposal. The two co-PI's from UTA represent both High Energy Physics (De) and Computer Science Engineering (Zaruba). Numerous CS students have obtained graduate degrees from UTA while working on ATLAS software development.

Kaushik De is the Director of the ATLAS South-West Tier 2 center, and is the US ATLAS Facilities Operations Coordinator. He proposed and co-leads PanDA workload management system development and related software projects.

Alexei Klimentov is the Physics Application Software Group Head at BNL and ATLAS Distributed Computing (ADC) Coordinator. The ATLAS Distributed Computing delivers a production quality tools for ATLAS Grid activities such as data placement, data processing and analysis. The system has been capable of sustaining with large contingency the needed activities in the first years of LHC data taking. Led R&D and Task Forces in several new computing areas, leading ADC effort for the coherent system design for 2013 and beyond.

Sergey Panitkin is the senior software engineer at BNL who ported PanDA into EC2 and is currently leading ATLAS Research and Development on cloud computing.

Alexandre Vaniachine is the software engineer at Argonne who develops core components of the ATLAS workload management system.

Torre Wenaus is US ATLAS Computing Coordinator and ADC Software Development co-coordinator. He proposed and continues to co-lead PanDA software development.

Dantong Yu has extensive experience in high-performance computer networking, data mining, and large-scale data management through global networks. He is BNL Co-PI of Kbase project. He led the USATLAS distributed data storage and processing facility for nine years.

8 Student Participation

In this proposal, we request funding for a CS graduate student. UTA has successfully integrated CS doctoral students into HEP experiments. At UTA, the traditional CS training is enhanced by practical applications in HEP distributed computing. Zaruba and De supervised the successful CS Ph.D. thesis of Dr. Aaron Thor. Currently they are jointly supervising the research of doctoral student Mikhail Titov, who developed the DaTri on-demand data transfer system used by ATLAS physicists. We propose to follow a similar successful model for PanDA. We will select a CS graduate student with strong interest in physics applications. The three year program of work proposed here will lead to successful doctorate degree for the selected students, while contributing to the work plan outlined here.

Literature Cited

[1] PanDA : Distributed Production and Distributed Analysis System for ATLAS. T.Maeno et al. Journal for Physics vol 119 part 6, 2008

[2] ATLAS Technical Proposal . CERN LHCC 94/43

[3] LHC – The Large Hadron Collider. http://lhc.web.cern.ch/lhc/

[4] CHARMM. Chemistry at HARvard Macromolecular Mechanics. http://www.charmm.org/

[5] The Alpha Magnetic Spectrometer (AMS) on the International Space Station. M.Aguilar et al, Physics Reports 366 (2002) 331-405

[6] CMS Technical design report. Journal of Physics, Nucl. Part Physics, 34, 995.

[7] Common Analysis Framework for ATLAS and CMS. Feasibility Study. ATLAS Distributed Computing Technical Interchange Meeting. Annecy, Apr 18-20,2012. https://indico.cern.ch/getFile.py/access?contribId=12&sessionId=5&resId=1&materialId=slides&confId=176443

[8] Conference on Computing in High Energy and Nuclear Physics. New York, May 2012. http://www.chep2012.org/

[9] WLCG – Worldwide LHC Computing Grid, http://lcg.web.cern.ch/lcg/

[10] Distributed Computing in ATLAS. D.Barberis, K.Bos, A.Klimentov, M.Lamanna, G.Negri and J.Shank, Proceedings of Science, November, 2008. XII conference on Advanced Computing and Analysis Techniques in Physics Research.

[11] Amazon Elastic Computing Cloud, http://aws.amazon.com/ec2/

[12] PanDA Production and Analysis backend. 17th International Conference on Computing in High Energy and Nuclear Physics (CHEP09), March 2009. Journal of Physics Volume 219, 210. K.De, A.Klimentov, T.Maeno, P.Nilsson and T.Wenaus

[13] The ATLAS PanDA Monitoring System and its Evolution. 2011 J. Phys.: Conf. Ser. 331 072058, A.Klimentov, P.Nevski, M.Potekhin and T.Wenaus

[14] "Exploiting Virtualization and Cloud Computing in ATLAS" to be published in CHEP2012 proceedings. F.Barreiro, D.Benjamin, K.De, I.Gable, V.Hendrix, S.Panitkin, M.Paterson, D. van der Ster, R.Taylor, R.Vitillo, R.Walker on behalf of ATLAS collaboration.

[15] Vaniachine, A.V. for the ATLAS Collaboration, "ATLAS detector data processing on the Grid," Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE, pp 104 – 107.

[16] D. Katramatos, S. Sharma, D. Yu, Virtual Network On Demand: Dedicating Network Resources to Distributed Scientific Workflows the Fifth International Workshop on Data Intensive Distributed Computing (DIDC 2012).

[17] ESCPS: The End Site Control Plane Service project, https://plone3.fnal.gov/P0/ESCPS

[18] INCITE: The Innovative and Novel Computational Impact on Theory and Experiment, https://hpc.science.doe.gov/allocations/calls/incite2013

[19] Kittyhawk: http://kittyhawk.bu.edu/kittyhawk/Kittyhawk.html

[20] Geant4: http://geant4.cern.ch

[21] LSST. Large Synoptic Survey Telescope http://www.lsst.org/lsst/

[22] Charles F. Clavier et al (2007-03-19) LSST Reference design, LSST Corporation