# Big PanDA on HPC/LCF Activities

Sergey Panitkin
BNL

# Introduction

- This talk overlaps with several talks given today
  - Alexei – BigPanDA project overview and HPC intro
  - Danila – PanDA integration work at OLCF
  - Ken – OLCF and Titan details
- I will try to not to duplicate material too much but to highlight essential features of the project and report on its current status

# PanDA in ATLAS

- The ATLAS experiment at the LHC - Big Data Experiment
  - ATLAS Detector generates about 1PB of raw data per second – most filtered out
  - As of 2013 ATLAS DDM manages ~140 PB of data, distributed world-wide to 130 of WLCG computing centers
  - Expected rate of data influx into ATLAS Grid ~40 PB of data per year
  - Thousands of physicists from ~40 countries analyze the data
- PanDA project was started in Fall 2005. **P**roduction **an**d **D**ata **A**nalysis system
  - Goal: An automated yet flexible workload management system (WMS) which can optimally make distributed resources accessible to all users
  - Originally developed in US for US physicists
- Adopted as the ATLAS wide WMS in 2008 (first LHC data in 2009) for all computing applications
- Now successfully manages O(10E2) sites, O(10E5) cores, O(10E8) jobs per year, O(10E3) users

# Next Generation "Big PanDA"

- ASCR and HEP funded project "Next Generation Workload Management and Analysis System for Big Data". Started in September 2012.
- Generalization of PanDA as meta application, providing location transparency of processing and data management, for HEP and other data-intensive sciences, and a wider exascale community.
- Project participants from **ANL**, **BNL**, **UT Arlington**
- **Alexei Klimentov –** Lead PI, **Kaushik De** Co-PI
- **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)

# HEP and HPC

- Historically HEP community was not using LCF extensively
  - Early experience was not very encouraging, hardware and programming environment was not very convenient for HEP.
- Current pace of research and discovery at LHC is limited by ability of LHC computing Grid to generate Monte-Carlo events - ”Grid luminosity limit”
  - Not enough CPU power !
  - Many physics simulation requests have to wait for many month
  - Currently O(100k) CPU available to ATLAS worldwide, ¾ dedicated to MC production
- LCF are rich source of CPUs
  - Typically CPUs are weaker than on servers on the Grid, but there are many of them!
- LCF typically have good storage infrastructure
  - O(1-10PB) per installation

# Some features of the HPC platforms

- Often non x86 CPUs (Blue Gene)
  - Cross compilation required
- Typically two component architectures: front end – worker nodes
  - Front end – user direct login nodes
    - Full Linux OSes – though non-Red Hat based
  - Worker nodes – only accessible via HPC batch systems
    - Typically Linux kernels cut down for efficiency, with limited functionality, no outside connectivity
      - Very limited OS functionality on Blue Gene
      - More feature rich OS on Cray
- If front end and worker nodes environments are sufficiently different cross compilation is required
- Often monolithic binary is required – no shared libraries
  - Cray (Titan, Hopper) allows for shared libraries (with performance hit) that simplifies application software port significantely

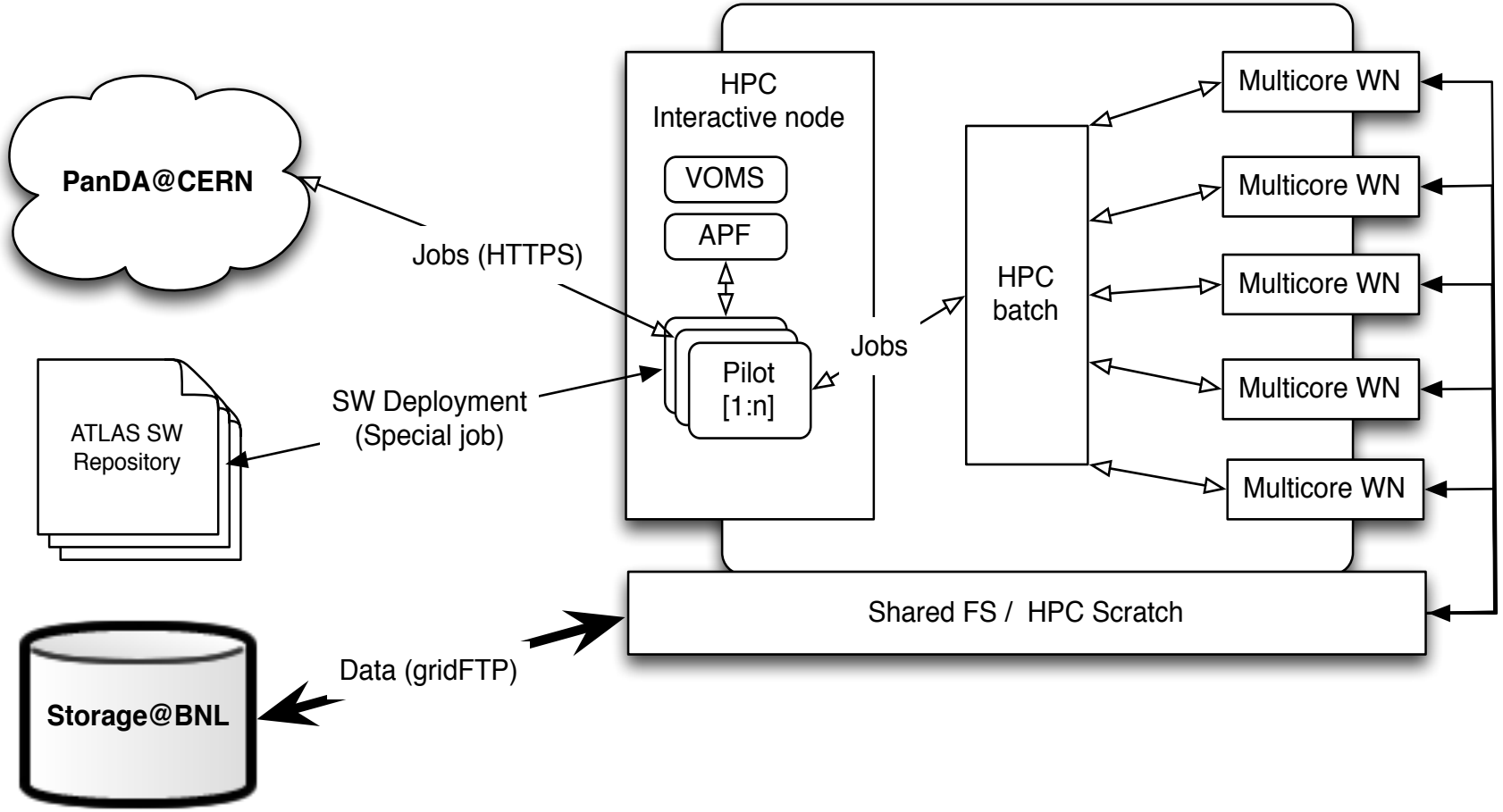
# Panda set up on HPC platforms

- Main idea - try to reuse existing PanDA components and workflow logic as much as possible
  - PanDA pilot, APF, etc
- PanDA  connection layer runs on front end machines, in user space
- All connections to PanDA server at CERN are initiated from the front end machines
- “Pull” architecture over HTTPS connections to predefined ports on PanDA server
- For local HPC batch interface use SAGA (Simple API for Grid Applications) framework
  - http://saga-project.github.io/saga-python/
  - http://www.ogf.org/documents/GFD.90.pdf

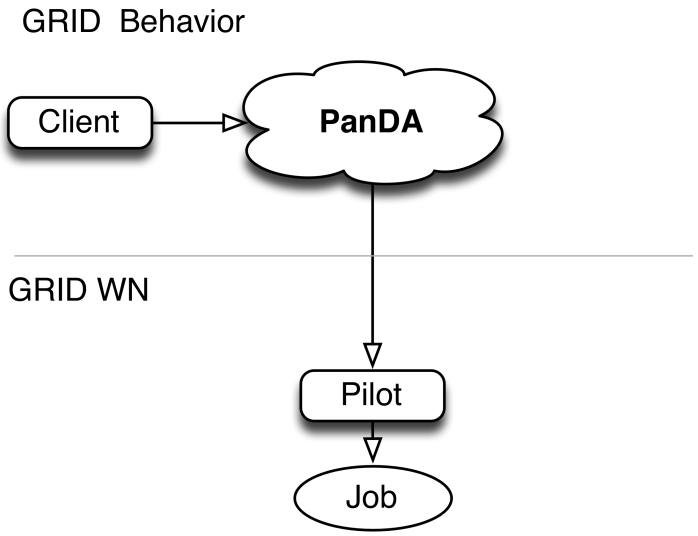
# Workflow on HPC machines

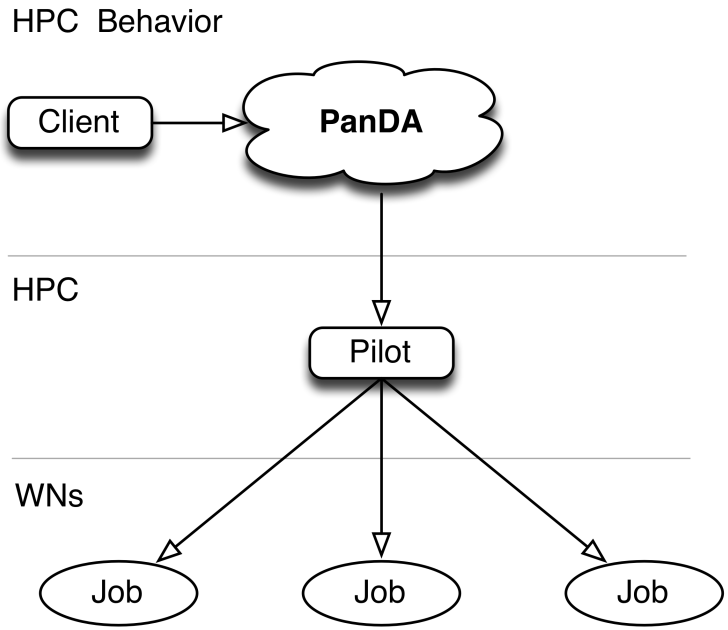
- Software is installed on HPC machine via CernvmFS or direct pull from repositories (for example non-ATLAS workload)
- Pilot is instantiated by APF or other entity
- Pilot asks PanDA for a workload
- Pilot gets workload description
- Pilot gets input data, if any
- Pilot sets up output directories for current workload on shared a file system
- Pilots generates and submits JDL description to a local batch system
- Pilot monitors workload execution (qstat, SAGA calls)
- When workload is finished pilot moves data to destination SE
- Pilot cleans up output directories
- Pilot exits

# Schematic PanDA setup on HPC



# Pilot on HPC




“One to One”

“One to Many”

# Current HPC resources for Big PanDA

- Currently have accounts at:
  - **Oak Ridge Leadership Class Facility (OLCF)** more details in Ken's talk
    - Titan (our own Big PanDA project (CSC108) allocation)
    - Kraken (part of NSF Xsede infrastructure, through UTK allocation)
  - **National Energy Research Scientific Computing Center (NERSC)**
    - Hopper, Carver (through OSG allocation)
  - **New York Blue at BNL**
    - Blue Gene /P (our own project allocation)
- We concentrate on ORNL development right now
  - Synergy with Geant 4 proposal for Titan (use of GPUs on Titan)
  - Great support and interest from OLCF management in BigPanda
  - Large CPU time allocation
- Parallel port to NERSC machine
  - Similar platform to ORNL - Cray

# Current status on ORNL

- Sergey Panitkin has access to Titan, still waiting for a fob for Kraken
- Danila Oleynik has access to Kraken and Titan
- ATLAS pilot is running on Titan and Kraken FEs
  - Connections to PanDA server verified
- AutoPilotFactory (APF) is installed and tested on Titan FE (J. Hover)
  - Local HTCondor queue for APF installed
- APF's pilot wrapper is tested with the latest version of ATLAS pilot on Titan FE
- SAGA-Python is installed on Titan FE and Kraken FE. In contact with SAGA authors from Rutgers (S. Jha, O. Weidner)
- A queue for Titan is defined in PanDA
- Connection from Titan FE to Federated ATLAS Xrootd is tested
- **More details in Danila's talk**

# Data management on ORNL machines

- Input and output data on /tmp/work/$USER or /tmp/proj/$PROJID
  - Accessible from both front end and worker nodes
  - High Performance, high capacity Lustre file system
- Output data moved by pilot to ATLAS storage element after job completion.
  - Currently to BNL SE. End point is configurable.

# Situation with Workloads

- Root is ported to Titan and Hopper @NERSC
  - Many thanks to Ken Read for advise on Titan port!
- ATLAS t-tbar analysis code ported to Titan and Hopper
  - ATLAS data (D3PD for ttbar analysis) transferred to Titan and Hopper
  - Proof-Lite mode tested on interactive batch nodes
- Started event generator ports
  - SHERPA (v. 2.0.b2 and v. 1.4.3) was ported to Titan and Hopper
  - MadGraph 5 (v. 1.5.12) was ported to Titan and Hopper
  - Simple examples and tutorials do run.
  - Need expert help for more realistic workload. Alexei discussed this with ATLAS management. Vakho Tsulaia from LBNL was contacted.
- **Will have to go through workloads validation steps!**

# Current resource allocation at OLCF

PanDA/OLCF meeting in Knoxville. Aug 9

- PanDA deployment at OLCF was discussed and agreed, including AIMS project component
- Cyber-Security issues were discussed both for the near and longer term.
- Discussion with OLCF Operations
- OLCF management is very interested in prospects of increased efficiency of machine utilization
- After the meeting PanDA project (CSC108) allocation was increased from 10k to 500k hours on Titan
  - To compare:
    - ATLAS allocation at NERSC (m1092) - 450k hours
    - OSG allocation at NERSC (m670) - 300k hours

# Next Steps

- Pilot job submission module  (runJob) development (see Danila's talk)
  - SAGA based interface to PBS tests
  - Better understanding of job submission to worker nodes
    - Multicore, GPU usage, etc
- DDM details at ORNL.
  - Possible use of Cray data transfer nodes
  - Integration with ATLAS storage infrastructure
  - Non ATLAS DDM solutions (Globus OL, SAGA, iRODS)
- Workloads
  - ATLAS Simulation workloads
  - Athena and Athena MP
  - Geant 4 (in particular on Titan's GPUs)

# Summary

- Work on integration of OLCF, NERSC machines and PanDA has started
- Key PanDA system components ported to Titan@OLCF
- Component integration is in progress (more in Danila's talk)
- Realistic workloads ports are in progress
- Significant increase in time allocation at OLCF

# Acknowledgements


- J. Caballero Bejar(BNL)
- Kaushik De (UTA)
- A. DiGirolamo (CERN)
- J. Hover (BNL)
- S. Jha and O. Weidner (Rutgers)
- A. Klimentov (BNL)
- M. Livny and HTCondor team (UW)
- D. Oleynik (UTK)
- K Read (UTK)
- P. Nilsson (UTA)