PanDA setup at ORNL

Sergey Panitkin, Alexei Klimentov
BNL

Kaushik De, Paul Nilsson, Danila Oleynik, Artem Petrosyan
UTA

for the PanDA Team

Outline

- Introduction
  - PanDA
  - Panda pilot
- ORNL setup
- Summary

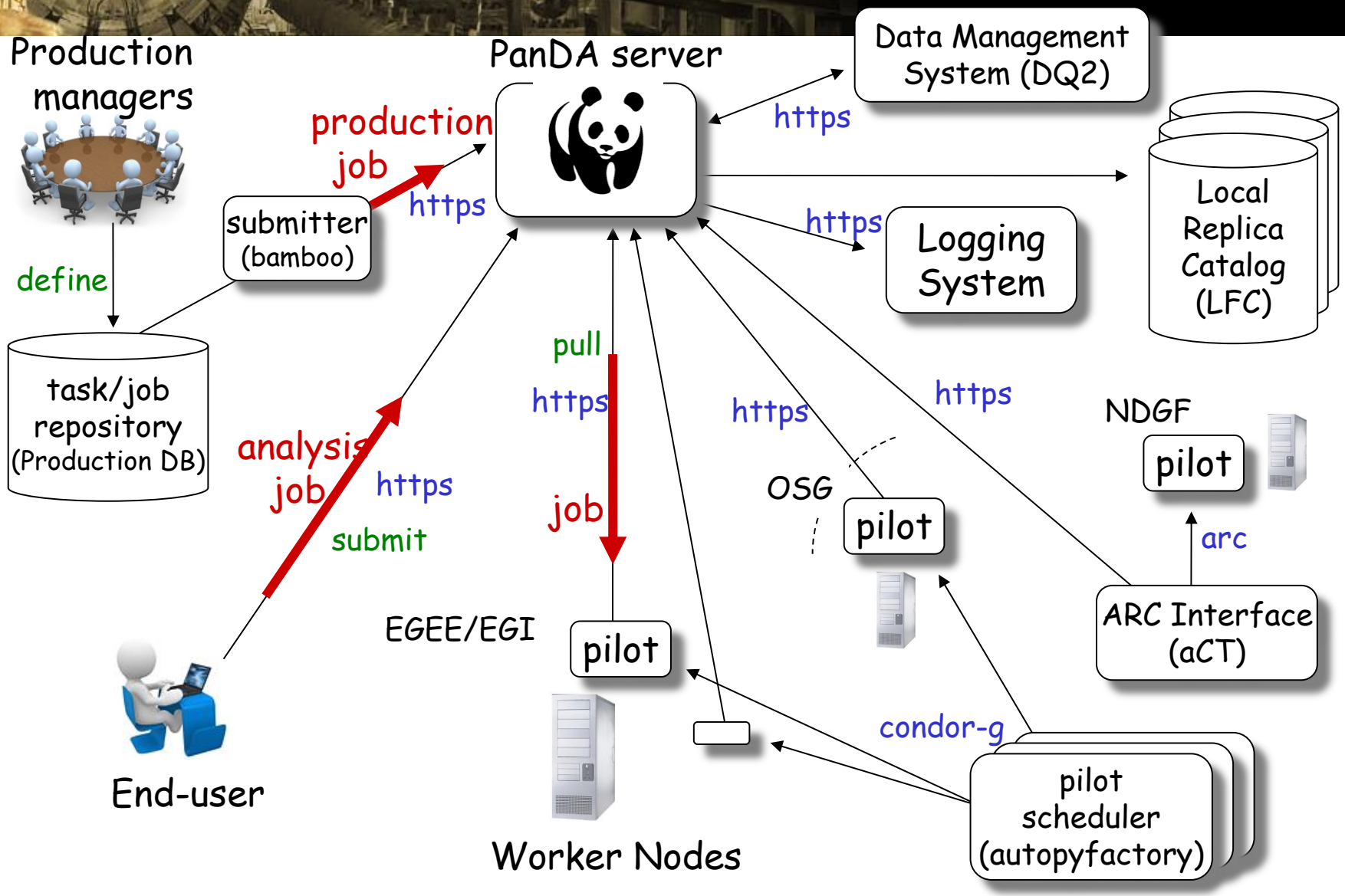
PanDA in ATLAS

- 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

Key Features of PanDA

- Pilot based job execution system
  - Condor based pilot factory
  - Payload is sent only after execution begins on CE
  - Minimize latency, reduce error rates
- Central job queue
  - Unified treatment of distributed resources
  - SQL DB keeps state - critical component
- Automatic error handling and recovery
- Extensive monitoring
- Modular design
- HTTP/S RESTful communications
- GSI authentication
- Workflow is maximally asynchronous
- Use of Open Source components

PanDA. ATLAS Workload Management System



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)

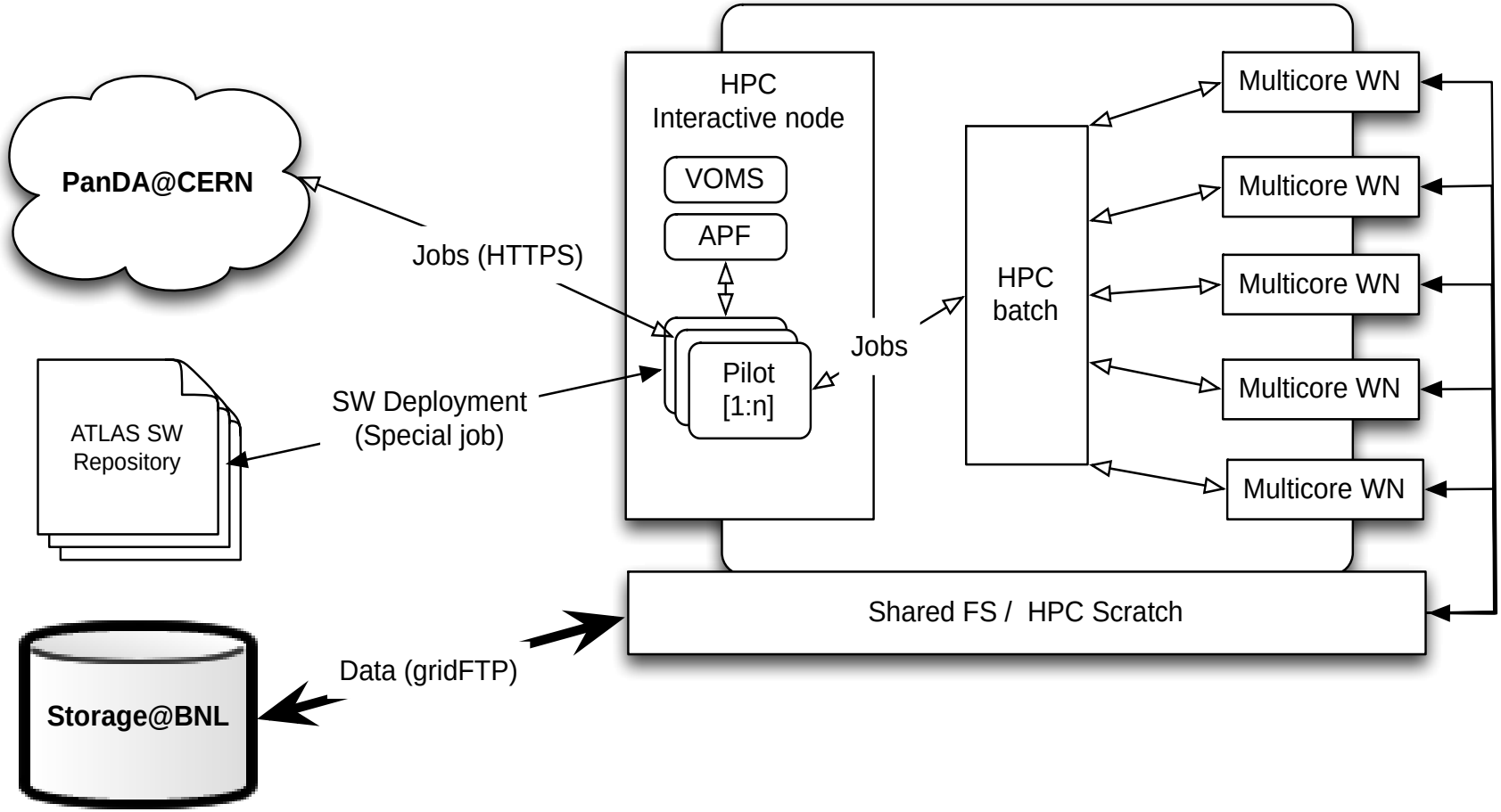
PanDA pilot

- PanDA is pilot based system. Pilot is what is submitted to batch queues
- PanDA pilot is an execution environment used to prepare computing element
  - Request actual payload from PanDA
  - Transfers input data from SE
  - Executes payload and monitors it during execution
  - Clean up after the payload is finished
    - Transfer output
    - Clean up, transmit logs and monitoring information
- Pilots allow for low latency job scheduling which is especially important in data analysis

Panda set up on ORNL machines

- 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 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
- Please note that Titan FE is running SLES 11 not RH5 !

Schematic PanDA setup at ORNL



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 ask PanDA for workload
- Pilot gets workload description
- Pilot gets input data, if any
- Pilot sets up output directories for current workload
- Pilots generates and submits JDL description to 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

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
- Output data moved by pilot to ATLAS storage element after job completion.
  - Currently to BNL SE. End point is configurable.

Current status

- Sergey has access to Titan, still waiting for a fob for Kraken
- Danila has access to Kraken, waiting for (approval?) on Titan
- ATLAS pilot is running on Titan FE
  - Connections to PanDA server verified
- AutoPilotFactory (APF) is installed and tested on Titan FE
  - 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

Next Steps

- Pilot job submission module (runJob) development
  - SAGA based interface to PBS
  - Better understanding of job submission to worker nodes
    - Multicore, GPU, etc
- DDM details at ORNL.
  - Use of data transfer nodes
- Realistic Workloads
  - ATLAS codes
  - Root
  - GEANT4 on GPU
  - etc

Summary

- Work on integration of ORNL machines and PanDA has started
- Key Panda system components ported to Titan
- Component integration is the next step
  - Pilot modifications for HPC
  - Pilot-SAGA runJob module
  - Pilot DDM details on Titan
- Realistic workloads are desirable

Acknowledgements

- K Read (UTK)
- J. Hover and J. Caballero Bejar(BNL)
- S. Jha and O. Weidner (Rutgers)
- M. Livny and HTCondor team (UW)
- A. DiGirolamo (CERN)