BROOKHAVEN NATIONAL LABORATORY

The Future of PanDA in ATLAS Distributed Computing

K. De, A. Klimentov, T. Maeno, P. Nilsson,
D. Oleynik, S. Panitkin, A. Petrosyan,
J. Schovancova, A. Vaniachine, T. Wenaus
on behalf of ATLAS collaboration

University of Texas at Arlington, USA

Brookhaven National Laboratory, USA

Joint Inst. for Nuclear Research, RU

Argonne National Laboratory, USA

CHEP2015, Okinawa, Japan, April 13-17 2015

Introduction

BROOKHAVEN NATIONAL LABORATORY

- PanDA = Production and Distributed Analysis System
  - Designed to meet ATLAS production/analysis requirements for a data-driven workload management system capable of operating at LHC data processing scale
- PanDA has performed well for ATLAS including the LHC Run1 data taking period
  - Producing high volume Monte-Carlo samples and making huge computing resources available for individual analysis
    - Running ~150K jobs concurrently
    - Processing ~0.7 million (~1.5 million at peak) jobs per day
  - Being actively evolved to meet the rapidly changing requirements for analysis use cases
    - No significant service disruptions
- New developments for Run 2 and beyond

Motivation for New Developments

- More efficient usage of pledged resources
- Partitioning of workload suitable for opportunistic resources based on their dynamic characteristics
- Handling of the workflow and bookkeeping both with coarse and fine granularities
- Integration of network awareness
- Improvement of visualization

Major System Evolution for Run 2 and Beyond

Dynamic Job Definition 1/3

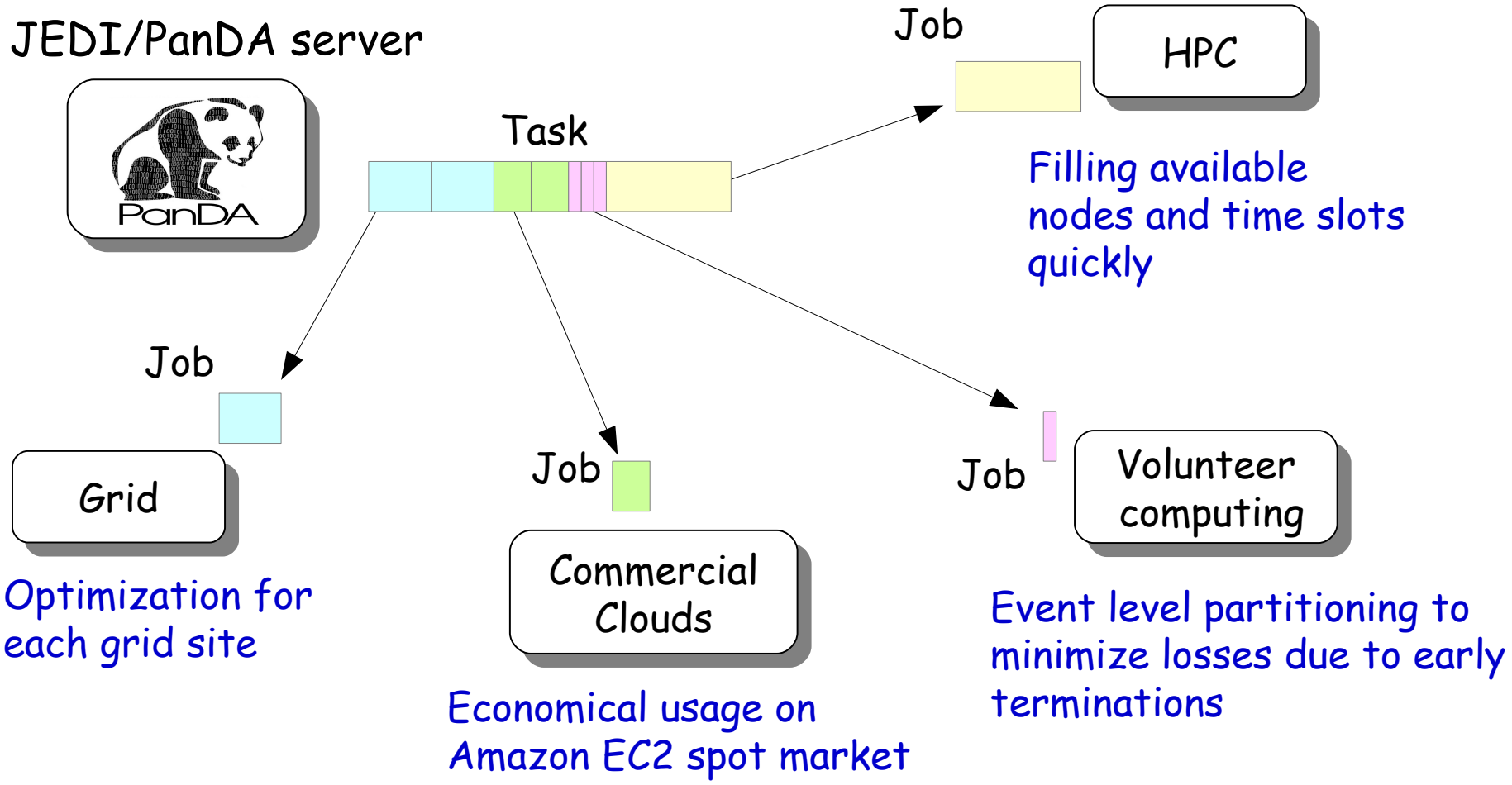
BROOKHAVEN NATIONAL LABORATORY

- Two new components
  - DEFT (Database Engine for Task)
    - Handles production requests and tasks
  - JEDI (Job Execution and Definition Interface)
    - Dynamically splits workload for optimal usage of resources
    - Manages workload at task, job, file, and event level
    - Automatically merges outputs
- Tasks are accepted to be partitioned to jobs based on the dynamic state of available resources
  - Jobs are an implementation detail of getting tasks done

Dynamic Job Definition 2/3

BROOKHAVEN NATIONAL LABORATORY

- Workload partitioning for traditional and opportunistic resources



Dynamic Job Definition 3/3

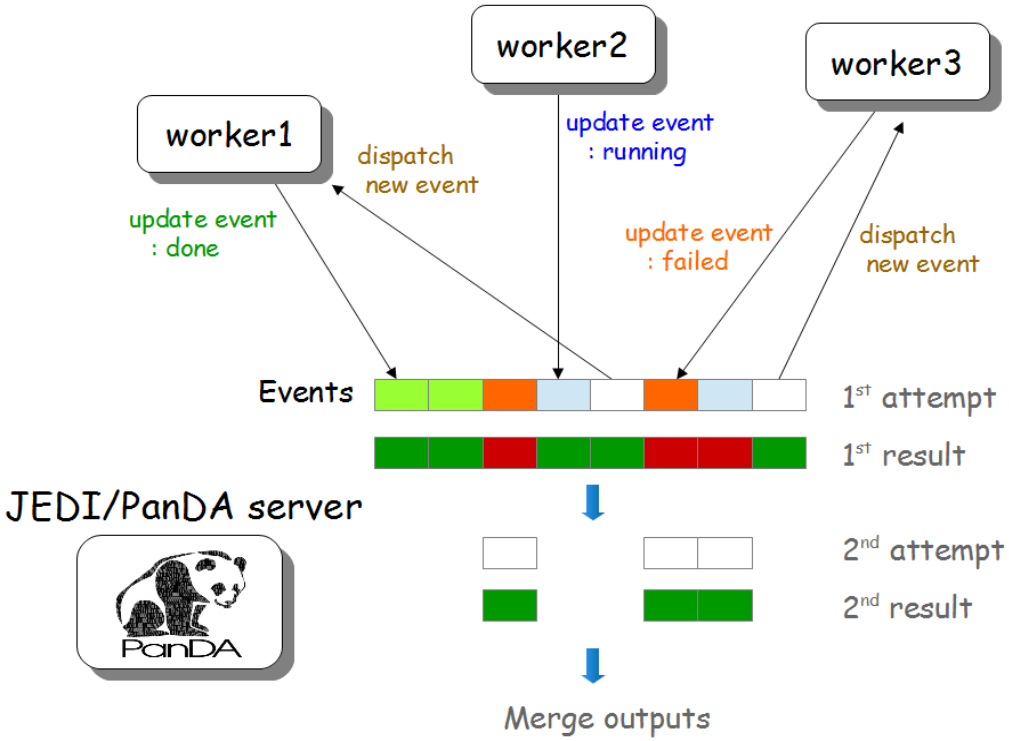
BROOKHAVEN NATIONAL LABORATORY

- Benefits
  - Excluding requirements from users of detailed knowledge on computing resources
    - Especially for heterogeneous resources, e.g., many CPU cores, very short walltime limit, etc
  - Self-optimization of job parameters
    - Real job metrics are collected using scout jobs
      - A small number (~10) of jobs (= scout jobs) are generated for each task with minimum input chunks
    - Job parameters are optimized using job metrics for the rest of input
  - Simplification of client tools and centralization of user functions

Integration of Network Awareness

- Usage of WAN data access for user jobs
  - Job brokerage taking costs for WAN data access into account
  - Slightly relaxing the ATLAS computing model
    - Sending a fraction of user jobs to sites which don't locally have data but have good network connection to remote data
  - Throttle mechanism to protect SE
- Dynamic coupling of Tier1 and Tier2 sites based on network performance and data locality
  - Files are transferred to Tier2 site from Tier1 site via good network without multi-hop

Event Service



- The fine grained partitioning of processing
  - Allowing workloads to be tailored dynamically to resources currently available
  - Minimizing losses when opportunistic processing slots are abruptly revoked

- HPC : validated on NERSC Edison, being ported to other platforms
- Commercial cloud : validating on Amazon EC2 spot market
- Volunteer computing : being ported to ATLAS@Home
- More details in CHEP15 talks

#140, V. Tsulaia : Fine grained event processing on HPCs with the ATLAS Yoda system

#183, T. Wenaus : The ATLAS Event Service: A new approach to event processing

Evolving PanDA Pilot 1/2

BROOKHAVEN NATIONAL LABORATORY

- Refactorization to core modules and experiment-specific plugins
- Supporting new workflows for Event Service and HPCs
  - HPC plug-ins have been developed for Titan (OLCF, US), Edison/Hopper (NERSC, US), Mira (ALCF, US) and Anselm (Ostrava, CZ)
  - Event Service on HPCs using newly developed Yoda software suite
    - Yoda acts as an intermediary layer between the PanDA Server and the PanDA Pilot which does not have access to outside connections on HPCs
    - Successfully validated on NERSC sites and is currently being extended to Titan
    - In development for Volunteer Computing ATLAS@Home project

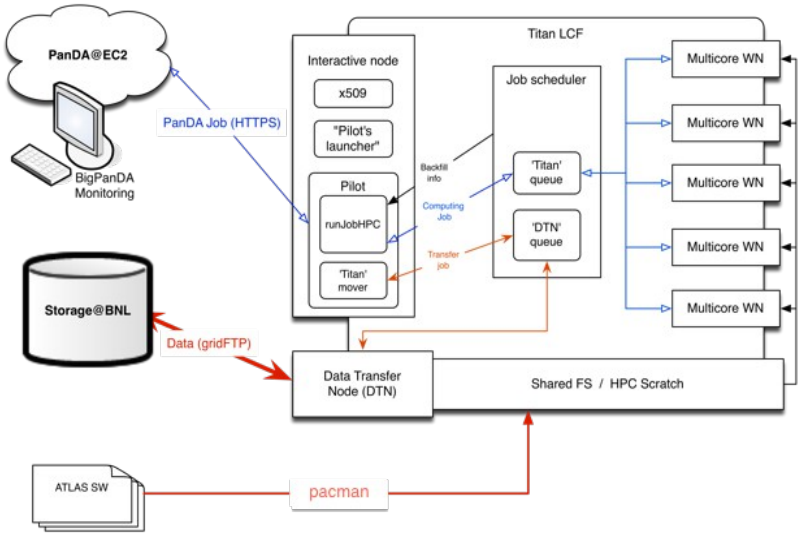
Evolving PanDA Pilot 2/2

BROOKHAVEN NATIONAL LABORATORY

- Using object stores as temporary storage
  - Highly useful for event service jobs that produce many small output files that are merged after the event service job has finished
  - Also in testing for log files
  - Ceph based object stores are available at BNL and CERN, and soon as RAL (UK)
- Support for gLExec
  - Pilot was also refactored to enable gLExec integration, the ability to dynamically switch the identity from the pilot to the user when executing the payload
  - More details in CHEP15 poster

    #155, E. Karavakis : gLExec Integration with ATLAS PanDA Workload Management System

PanDA on Titan at OLCF



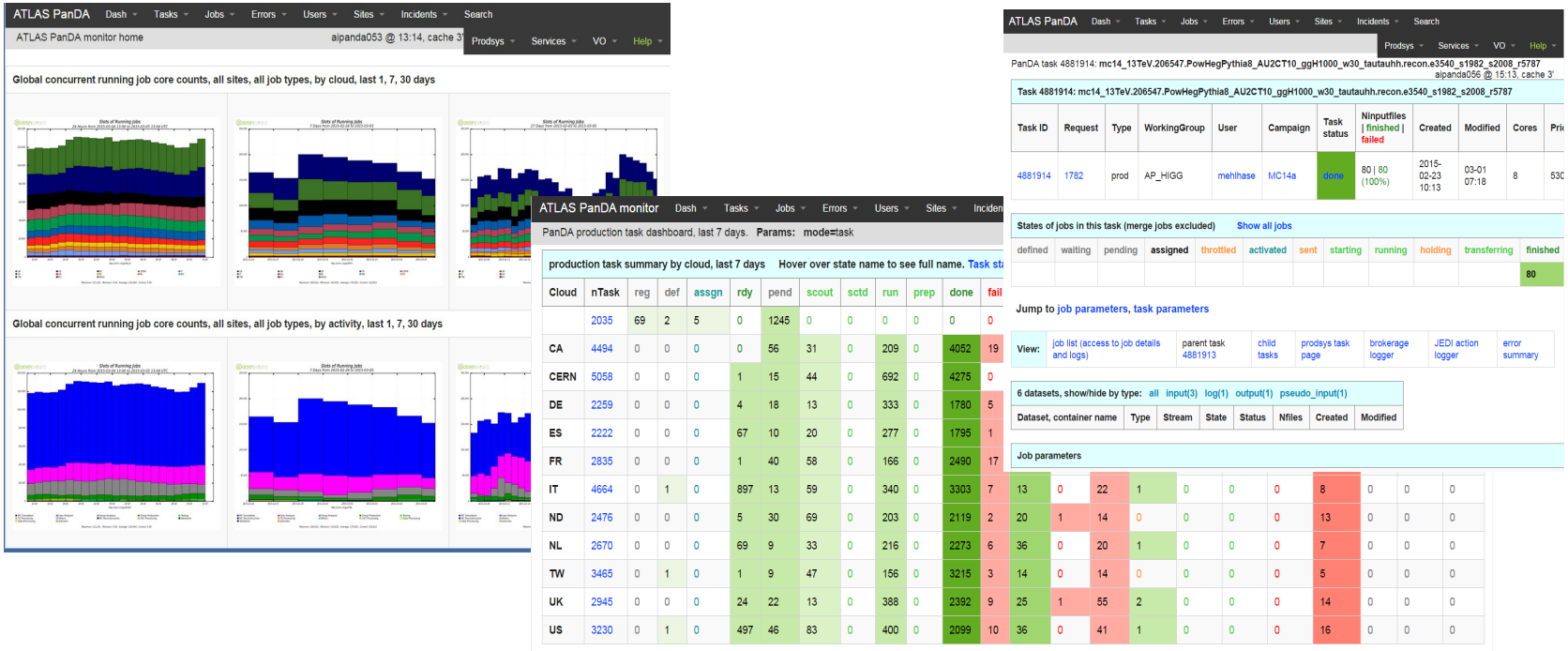
- Work on integration of Titan machines with PanDA
- Modified PanDA pilot to run on Titan's front-end nodes with backfill mode
  - Collecting information about free resources in quasi-realtime
  - Submits jobs to Titan's scheduler based on the info
- Successfully demonstrated steady operations for continuous PanDA job submission in backfill mode

- More details in CHEP15 talk

  #152, S. Panitkin : Integration of PanDA workload management system with Titan supercomputer at OLCF

New Monitoring

- Based on Django framework
- Clear separation between data access and visualization
- REST APIs to access object information
- Provide task-oriented view



Future Plans

- More intelligence to workload partitioning and brokerage
- Proactive control of the network to optimize workflows and dataflows
- New computing resources in production more efficiently and economically
- Lightweight tools for users, who are not fully integrated to the grid, to leverage PanDA for utilization of local beyond-pledge resources

Conclusions

- PanDA has performed well for ATLAS including the LHC Run 1 data taking period
- New components and features have been delivered to ATLAS before LHC Run 2
- Many developments and challenges to come while steadily running for LHC Run 2