PanDA
A New Paradigm for Computing in HEP
Through the Lens of ATLAS and Other Experiments

Kaushik De
Univ. of Texas at Arlington
On behalf of the ATLAS Collaboration

ICHEP 2014, Valencia
July 4, 2014
Computing Challenges at the LHC

- The scale and complexity of LHC computing
  - Hundreds of petabytes of data per year, thousands of users worldwide, many dozens of complex applications…

- Required a new approach to distributed computing
  - A huge hierarchy of computing centers had to work together
  - Main challenge – how to provide efficient automated performance
  - Auxiliary challenge – make resources easily accessible to all users

- Goals of this talk
  - Present a new model of computing developed for the ATLAS experiment, to make optimum use of widely distributed resources
  - How this system is being used beyond the LHC
  - Future evolution
Early History

- **Lessons learned from grid computing before 2005**
  - Errors were difficult to debug in distributed environment
  - A few percent systemic error rate could lead to user frustration
  - Waits in small percentage of queues = huge task completion delays
  - Difficult for users to manually schedule among long list of sites

- **New project started in 2005**
  - Based on previous 5 years of experience
  - Primary goal – improve user experience with distributed computing, making it as easy as local computing
  - Users should get quick results by leveraging distributed resources
  - Isolate users from heterogeneity in infrastructure and middleware
  - Fair sharing of resources among thousands of users
Enter PanDA

- **PanDA – Production and Distributed Analysis System**
  - Designed for the ATLAS experiment during LHC Run 1
  - See talk by D. Barberis yesterday on ATLAS Run 2 computing
  - Deployed on WLCG infrastructure
  - Standards based implementation
    - REST framework – HTTP/S
    - Oracle or MySQL backends
    - About a dozen Python packages available from SVN and GitHub
    - Command-line and GUI/Web interfaces

- **References**
  - [https://twiki.cern.ch/twiki/bin/view/PanDA/PanDA](https://twiki.cern.ch/twiki/bin/view/PanDA/PanDA)

July 4, 2014
Core Ideas in PanDA

- Make hundreds of distributed sites appear as local
  - Provide a central queue for users – similar to local batch systems

- Reduce site related errors and reduce latency
  - Build a pilot job system – late transfer of user payloads
  - Crucial for distributed infrastructure maintained by local experts

- Hide middleware while supporting diversity and evolution
  - PanDA interacts with middleware – users see high level workflow

- Hide variations in infrastructure
  - PanDA presents uniform ‘job’ slots to user (with minimal sub-types)
  - Easy to integrate grid sites, clouds, HPC sites …

- Production and Analysis users see same PanDA system
  - Same set of distributed resources available to all users
  - Highly flexible system, giving full control of priorities to experiment

Kaushik De
July 4, 2014
Additional PanDA Ideas

- Excellent fault tolerance across distributed resources
  - Independent components with asynchronous workflow
  - Internal re-scheduling at every step of execution

- Trust but verify all sources of information
  - PanDA uses own information system and internal metrics

- Multi-level scheduling for optimal use of resources
  - Task brokerage (T1), job brokerage (sites), job dispatch (to pilots)

- Multi-step data placement for maximum flexibility
  - Algorithmic pre-placement, asynchronous transfers with callback, pilot data movers, special optimizations for federated storage

- Integration with independent data management systems
  - DQ2 and Rucio for ATLAS (see talk by C. Serfon in this session)
PanDA Workload Management

Production managers define production job

submitter (bamboo/JEDI)

End-user

Data Management System

Logging System

Local Replica Catalog

NDGF

ARC Interface (aCT)

pilot scheduler (autoppyfactory)

Worker Nodes

PanDA server

EGI

OSG

condor-g

https

Kaushik De
Panda jobs go through a succession of steps tracked in central DB:

- Defined
- Assigned
- Activated
- Starting
- Running
- Holding
- Transferring
- Finished/failed
Recent Evolutions

- **Dynamic caching of input files – PD2P**
  - For user analysis, jobs go to data, which are initially placed by policy
  - PanDA dynamically re-distributes based on usage and demand

- **Evolution to mesh model**
  - Rigid hierarchy of sites relaxed based on network performance
  - Work and data can flow dynamically between sites

- **Dynamic subdivision of tasks into jobs - JEDI**
  - New component to dynamically slice work to match resources

- **Event service**
  - Processing small chunks of events on demand
  - Integrated with Event Index (see talk by A. Fernandez this session)

- **Network as a managed resource**
  - On par with CPU’s and storage
Current scale – 25M jobs completed every month at >hundred sites
First exascale system in HEP – 1.2 Exabytes processed in 2013
About 150,000 job slots used continuously 24x7x365
Paradigm Shift in HEP Computing

- **New Ideas from PanDA**
  - Distributed resources are seamlessly integrated
  - All users have access to resources worldwide through a single submission system
  - Uniform fair share, priorities and policies allow efficient management of resources
  - Automation, error handling, and other features in PanDA improve user experience
  - All users have access to same resources

- **Old HEP paradigm**
  - Distributed resources are independent entities
  - Groups of users utilize specific resources (whether locally or remotely)
  - Fair shares, priorities and policies are managed locally, for each resource
  - Uneven user experience at different sites, based on local support and experience
  - Privileged users have access to special resources

Kaushik De

July 4, 2014
The Growing PanDA EcoSystem

- **ATLAS PanDA**
  - US ATLAS, CERN, UK, DE, ND, CA, Dubna, Protvino, OSG …

- **ASCR BigPanDA**
  - DoE funded project at BNL, UTA – PanDA beyond HEP, at LCF

- **ANSE PanDA**
  - NSF funded network project - CalTech, Michigan, Vanderbilt, UTA

- **HPC and Cloud PanDA**

- **Taiwan PanDA** – AMS and other communities

- **CMS PanDA** – Common Analysis Framework

- **AliEn PanDA, LSST PanDA, other experiments**

- **MegaPanDA** …
Conclusion

- PanDA – a scalable and universal computing system
- Processing million jobs a day at hundreds of sites
- Thousands of active users in ATLAS
- Computing as a valuable resource for all users, irrespective of location or affiliation, enabling fast physics results
- Being evaluated/tested by other experiments, communities
- Continuing to evolve