



Interfacing PanDA to OLCF

Sergey Panitkin
Brookhaven National Lab



PanDA WMS

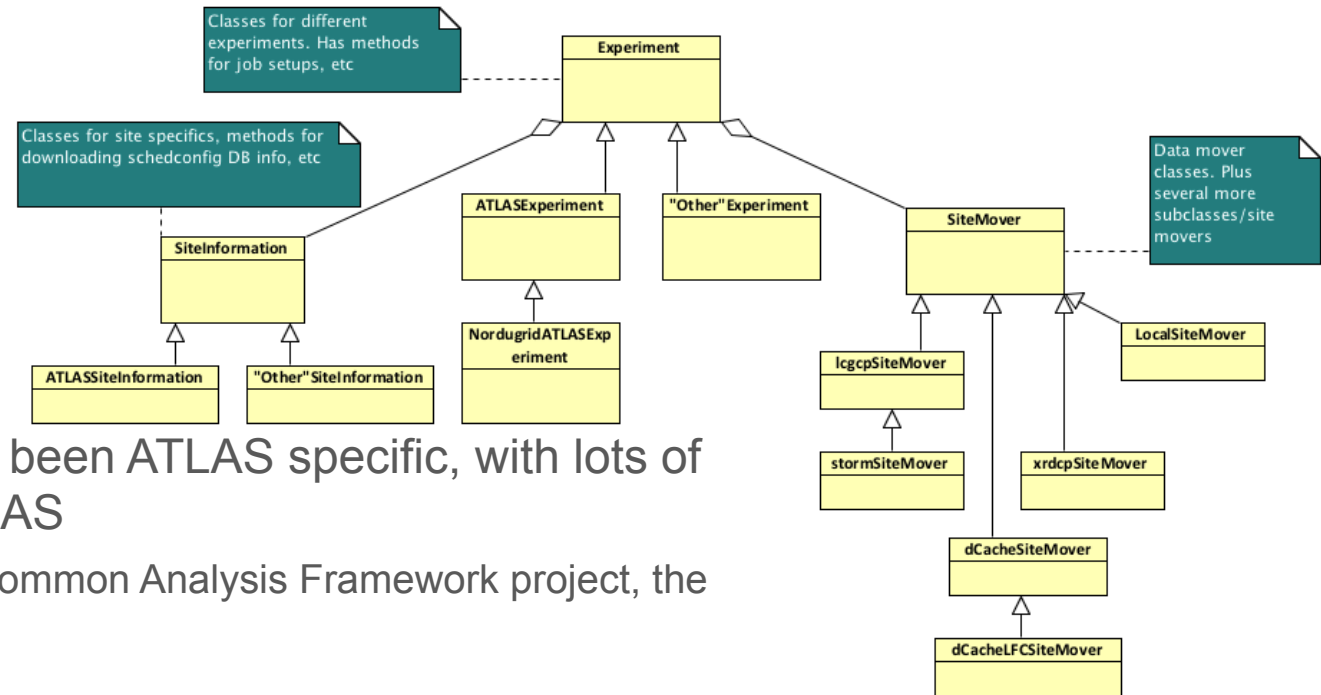
- ◆ **Production and Data Analysis (PanDA)** workload management system was developed by the ATLAS Experiment at LHC
- ◆ PanDA aggregates and manages distributed collections of batch queues and storage elements that comprise LHC Grid
- ◆ It successfully manages $O(10E2)$ sites, $O(10E5)$ cores, $O(10E8)$ jobs per year, $O(10E3)$ users
- ◆ 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



Next Generation “Big PanDA”

- ◆ ASCR and HEP funded project “Next Generation Workload Management and Analysis System for Big Data”
- ◆ 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
- ◆ One of the work packages is extension of PanDA to LCF and computational clouds.
- ◆ We have extensive experience with cloud computing
 - ◆ Public clouds: EC2, GCE, Helix-Nebula, RackSpace, FutureGrid, ...
 - ◆ Private clouds, BNL T1, CERN T0, Canada, Australia,...
- ◆ We just started work with SC
 - ◆ BG/P at BNL, ATLAS activity at ALCF
 - ◆ NERSC
 - ◆ ORNL now

Evolving Panda Pilot



- Until recently the pilot has been ATLAS specific, with lots of code only relevant for ATLAS
 - To meet the needs of the Common Analysis Framework project, the pilot is being refactored
- Experiments as plug-ins
 - Introducing new experiment specific classes, enabling better organization of the code
 - E.g. containing methods for how a job should be setup, metadata and site information handling etc, that is unique to each experiment
 - CMS experiment classes are currently being implemented
- Changes are being introduced gradually, to avoid affecting current production



PanDA Pilot for LCF

- ◆ Expanding PanDA from Grid to Leadership Class Facilities will require changes
- ◆ Each LCF is unique
 - ◆ Unique architecture and hardware
 - ◆ Specialized OS, “weak” worker nodes, limited memory per WN
 - ◆ Code cross-compilation is typically required
 - ◆ Unique job submission systems
 - ◆ Unique security environment
- ◆ Pilot submission to a worker node is typically not feasible
- ◆ Pilot/agent per supercomputer or queue model



Geant4 project on OLCF

- ◆ Geant4 at OLCF proposal will allow us to get access to Titan
- ◆ Get experience with all relevant aspects of the platform and workload
 - ◆ job build phase – compilation, linking, etc
 - ◆ job submission mechanism
 - ◆ job output handling
 - ◆ local storage system details
 - ◆ outside transfers details
 - ◆ security environment
 - ◆ adjust monitoring model
- ◆ Develop appropriate pilot/agent model for Titan
- ◆ Work can be done asynchronously to the main project
- ◆ Early access to Titan is very desirable



Outlook

- ◆ Incorporation of LCF into PanDA workload management system will require significant changes in many parts of the PanDA , in particular, workload delivery on worker nodes, scope and role of pilot framework on SC, it's interaction with native SC job submission mechanisms, data management and job monitoring to name a few.
- ◆ Participation in the Geant4 project on OLFC's Titan will allow us to get first hand experience on this unique platform and will give us an opportunity to develop new pilot framework for Titan as a prototype of the pilot for LCF in general
- ◆ This will be an important step in the evolution of PanDA and in development of the next generation workload management system that includes LCF