PanDA Pilot for HPC

Danila Oleynik, Wen Guan

Introduction

"HPC PanDA Pilot": mode of working of PanDA pilot when payload executes on separated node(s) from pilot and job management provided through interactions with job scheduler (early bindings)

This type of mode required when we have restricted access to working node, or when we need to perform special payload (for example MPI based)

HPC Pilot speciality

Each HPC/Supercomputer different and not only by architecture but also by policies of usage, but does it mean that each time we need special solution?

Proper (flexible, plug-in based) architecture of Pilot should allow us to configure application with only needful extra development. Also this will allow us to have incremental development process without overlapping of functionality between components.

Where we are now.

NERSC	Event service through Yoda	ES jobs
OLCF	Multijob Pilot	Regular jobs
ALCF	Integration with ARGO	Special payloads and processing
NorduGRID	ARC Tower	Regular jobs, ES jobs

PanDA pilot developers collect a lot of expertise about different (sometimes exotic) architectures of computing systems and workflows.

Current issues

Even with different workflows, we face sometimes similar problems:

- Flexibility: same workflows on different architectures for example: adaptation of AES for Titan and MultiJob for NERSC;
- Efficiency: stagein/stageout operations managing of multiply operations for MultiJob, managing of small files for ES, bulk operations with PanDA server, adaptation of algorithms for particular architectures.
- Monitoring/accounting lacks (it's not means, that tools bad, but our needs increases with new solutions)
    - We would like to have extended set of job statuses;
    - Information about pilot statuses and performance;
    - Clear understanding of CPU/Wallclock time calculations;
    - Publishing, representation of special information: for example backfill reports

Over issues (infrastructural)

- Managing of Pilots on systems with restricted access (systems without APF);
  - Cron jobs at NERSC, Pilot launcher at OLCF
- Managing of transient storage at HPC/LCF;
  - Authorization issues

Yoda-ES (Details)

Wen Guan

ES monitor

- ES accounting/monitoring is needed to measure performance in pilot
  - Setup time is long in some cases
  - Not efficient if only process few events after setup
- Pilot reports this info to panda for monitoring
- Based on accounting more smart scheduling on pilot like Yoda or panda

ES stagein/stageout

- Too many small files
    - Not efficient for stagein/stageout
- Data transfer node supports
- More than one event in one range
- Zip file supports

ARC-CE integration

- Pilot runs in rank-0
  - ARC-CT prepares payload
  - ARC CE stagein/stageout
  - Pilot at rank-0 doesn’t have much delay to start MPI
- Zip output integration
- Multiple job integration
  - Currently ARC-CT only sends one panda job per ARC submission
  - Yoda supports multiple panda jobs per submission
- BOINC
  - Yoda supports to run without MPI in single node

ES job scheduling

Automatically schedule some jobs between normal jobs and ES

- Not create special ES jobs
- Panda automatically schedule some normal jobs to ES queue
- Panda automatically adapts the jobOptions