PanDA server with MySQL backend on Amazon EC2

Jaroslava Schovancová (BNL)

PanDA Workshop @ UTA

September 3-4, 2013

BROOKHAVEN
NATIONAL LABORATORY

EC2 instance of PanDA server

- EC2 instance: m1.large, us-east-1a, ebs
  - domain pandawms.org
- Scientific Linux release 6.4 (Carbon)
  - Python 2.6, MySQL 5.1
  - MySQLdb 1.2.3, cx_Oracle 5.1.2
  - PanDA server: fork of current/rev16473
- Sanity test
  - Request: `isAlive ?`
  - Response: `alive=yes !`
- Installation documented in SVN/INSTALL notes

PanDA server running with MySQL

- VO independent
- Configuration of desired DB backend through the same config file **`panda_server.cfg`**
  - Oracle, MySQL
- Data access layer: DB proxies and their pools
- Introduced **`WrappedCursor`**, **`WrappedConnection`**
  - Address DB backend-specific features/differences
  - **Maintainable**
  - Generic: easy to extend to yet another DB backend
- Forked current branch Rev:16473

WrappedCursor, WrappedConnection

- **`WrappedCursor`** implements

execute, prepare, executemany, fetch[one|many|all],

rowcount, description, next, var, close

- **`WrappedConnection`** implements

begin, close, commit, rollback, connection, cursor, ping, wakeUp, disconnect

- Used in server and monitoring

Future challenges

- **`pandaserver`** package
  - Test against all supported DB backends on code change
  - Merge into trunk

    → deploy to CERN production instance
- EC2 instance
  - Performance benchmarking and optimization
  - Start rolling!

Instance plans

- Provide PanDA as a service, close collaboration with the experiment experts
- The first experiment:

Large Synoptic Survey Telescope (LSST)

- Areas to address
  - Job submission
  - Data management
  - Activity monitoring
  - Resources and their topology
  - Pilots and pilot factories

Job submission

- Transparent to the user
  - User declares payload to experiment-specific analysis framework
  - Experiment framework communicates with PanDA
- PanDA as a service

  → comes with full support from the BigPanDA team

Data management

- Experiment specific
  - LSST assumes SW and data is available at the site
  - Storages, catalogues, ...
- Possible ideas: Globus Online, Amazon S3

Activity monitoring

- No “cloud” notion, no task, but multiple VOs
- Job progress monitoring
  - Production summary: all VOs vs. single VO
  - User activity summary
  - Job details and history
- Background activity monitoring
  - Status of resources
  - SW availability
  - Data management

Resources

- Leverage opportunistic resources
  - LSST at BNL, HPC; later US T2s
- Resources topology
  - Use OSG and WLCG resources
  - *Own schedconfig instance*
- Pilots
  - *Own pilots*
  - *Own pilot factories (APF2.*)*

Summary

- **VO-independent PanDA instance running on EC2**
  - Capable to run on top of MySQL or Oracle DB
  - First draft of monitoring
    - clasical/new platform on top of MySQL DB
  - Getting ready for the **first** tests with the **real community: LSST**
- Challenges
  - Integration
  - Performance benchmarking
  - And much more...
- Many thanks to BigPanDA team members for fruitful discussions!