JEDI status

Tadashi Maeno (BNL)

Current Status

- JEDI-α is ready for full production
  - Functions only for production
  - AKTR → DEFT table → JEDI → Panda was successfully tested with real evgen, reco, reprocessing tasks
  - Just need to check the AMI change with real tasks
    - AMI has changed to real metadata.xml from PandaDB instead of prodDB
- Missing in JEDI-α
  - Functions for analysis
  - Merging function
  - Event Server

Functions of JEDI-α 1/2

- Full replacement for bamboo
  - Job definition and submission
  - Scout → avalanche
  - Automatic retry
  - Task brokerage
- Finer grained fairshare management
  - E.g., shares of AP/AP
- Improvements in the job brokerage
  - maxwdir, minmemory, maxmemory, mintime
  - Considering the number/size of missing files when assigning jobs with large number of input files
    - E.g., pre-placed min-bias datasets can be considered

Functions of JEDI-α 2/2

- Lost file recovery
  - Built-in function to reproduce files
  - Can reproduce files which were produced through JEDI
    - Doesn't work for files produced in old prodsys
  - Reverse cascade of parent files
    - Reproduce lost parent files + task chain
- Log merging
  - Merge small log files to move them to TAPE
- Metadata retrieval from AMI
  - File metadata like the number of events can be retrieved from AMI
  - E.g. can specify the exact number of events per job, even if each input file has a different number of events

Plans for Production

- Check the AMI change with a few real tasks
- Full production with a single task type (e.g., validation) via JEDI-α
  - TBD by GDP
  - Not very easy to use JEDI only for one cloud
    - Task brokerage by Panda + Task execution by JEDI
- All tasks via JEDI-α
- Retirement of Bamboo

JEDI for Analysis

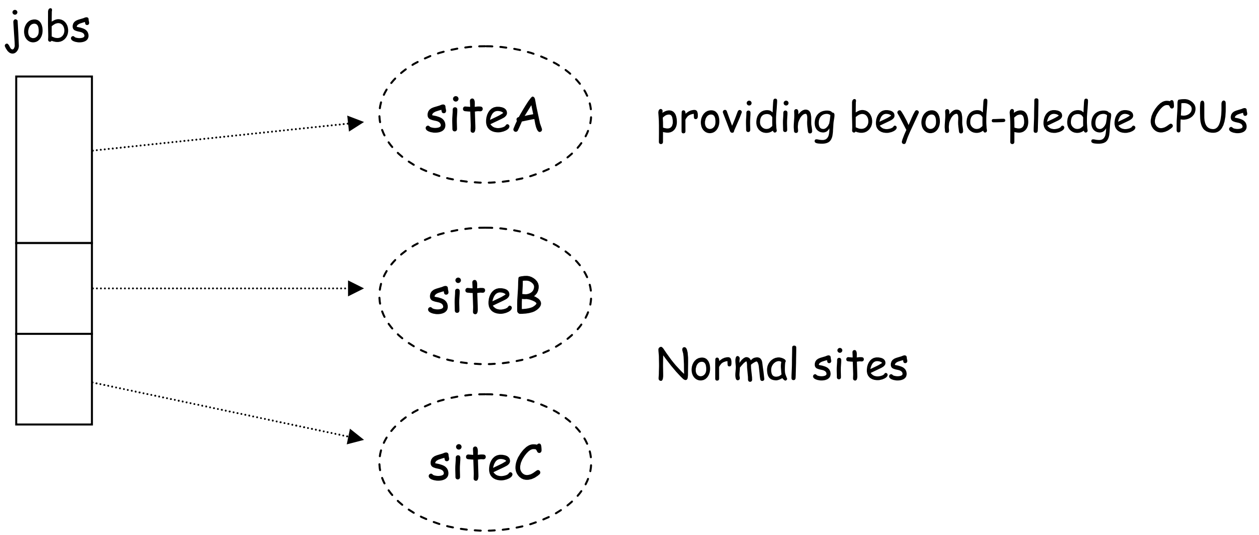
- Implemented
  - The scout → avalanche chain
  - Analysis brokerage considering data locality
    - Local
    - Remote (will take network info into account)
  - buildJob and runJob generation
  - User dataset registration
  - Finer-grained workload splitting (next page)
- For free
  - Rebrokerage
    - Just a reattempt from JEDI's point of view
  - Proper bookkeeping for jobs with event-level splitting
  - Job parameters can change when jobs are retried

Finer-grained Workload Splitting 1/2

- Old model : Jobs running on a dataset went to a single site
  - Job distribution tended to be unbalanced
  - Jobs could not go to another site unless all jobs are reassigned by the rebrokerage
  - If jobs run on a dataset container and constituent datasets are available at many sites, jobs can be distributed to multiple sites
- JEDI : Jobs can go to multiple sites (N sites) where the dataset is available
  - N is 5 at most
  - Job distribution is less affected by data distribution
    - Could have more uniform job distribution

Finer-grained Workload Splitting 2/2

- Internal complication of bookkeeping should be transparent to users
- Flexibility
  - E.g., can send a fraction of jobs to sites which provide beyond-pledge CPUs (such as T3s)



Status and Plans for Analysis

- Functions to execute prun jobs (ROOT-based or general jobs) are ready on the dev JEDI
  - Will be tested in this week
- To introduce 'preprocess job' for Athena, Event-Picking, GRL, etc
  - Time consuming procedures to define task parameters
    - Config extraction for Athena (running Athena with dummy application mgr)
    - Event lookup
    - Conversion of GRL
  - Do these kinds of procedures on WNs instead of JEDI
    - preprocessingJob → buildJob → runJob
  - Task Parameters are finalized when preprocess jobs are completed
  - ~1 week to implement
- To change client tools
  - Just simplification
  - ~ a few days
- Full functions for Athena or ROOT jobs by the end of Sep

Event Server & HPC

- Developments will start once JEDI implements analysis functions
- Event table is already available in the JEDI (ATLAS_PANDA) schema
  - Performance test was done on INTR
    - No scalability issue
- TBD : communication between the panda server and the pilot
  - Extension of job dispatch and heartbeat
    - E.g., GUID + event range to dispatch workload, retry, completion of processing on an event range
- For HPC, one job could be split internally by HPC
  - Could use the event table to keep track of processing

Development Schedule

- Implementation of Analysis functions
  - In Sep
- Long term issues (Oct~Dec)
  - Support for variable number of outputs for AthenaMP-2
  - Event server
  - HPC
  - Merging
  - etc..