

PanDA Server developments and plans

Tadashi Maeno (BNL)

Output Merging for Analysis Jobs (1/3)

- Analysis jobs tend to produce many small files
 - Not very good for SE and data transfer
- Previously a general merging scheme using zip/unzip by the pilot was proposed but was not very well appreciated
 - Small files are zipped to a temporary file, DDM transfers the file, and then the file is unzipped to the original files at the destination → There will be small files at destination eventually
- Trying more rigid scheme

Output Merging for Analysis Jobs (2/3)

➤ New Workflow

1. The user submits a job with the `--mergeOutput` option
2. The job could be split to multiple sites. One dataset is created per site. All datasets are added to a container
3. Normal analysis jobs run as usual
4. Once all jobs contributing to an output dataset have finished/failed, the panda server generates merge jobs
 - Files are merged at each site
 - One merge job per 5GB and/or 200 files
 - Merge jobs cannot be predefined when the job is submitted, since the size of output file is unknown at that time
5. New datasets for merged files are created and added to the container
6. Each merge job executes a general merge `trf` or user-defined script (see Hurng-Chun's talk in DA session)
7. When each merge job is successfully finished, unmerged files are deleted from old dataset and merged files are added to new dataset
 - When all merge jobs for an old dataset successfully finished, the dataset becomes empty and gets deleted
8. Email notification is sent out when all merge jobs for all datasets are finished/failed

Output Merging for Analysis Jobs (3/3)

- Merge trf takes file-type as an input parameter, so that it uses correct tool, e.g., Athena for POOL files, ROOT for NTuple files, zip for log files, etc
 - The panda server detects types of unmerged files by checking job specification and gives them to the trf
 - Each runAthena job knows what types of files are produced since job option is parsed before job submission. For extra output files specified in the --extOutFile, file extension is checked, e.g., XYZ.pool.root → POOL
 - For runGen jobs, file extension is checked
 - Users can explicitly specify file-type
- Transfer request to DaTRI is made for merged datasets when the first merge job has finished
 - Unmerged files are not transferred
- Reattempt
 - Client tools will be updated to allow reattempt of merge jobs

Event Picking in PD2P

- To transfer only files which contain interesting events, rather than transferring the whole dataset
 - Proposed in the context of life without ESD.
e.g. to transfer only selected RAWs
- New functionality added to PD2P
 - Takes a list of event and run numbers, converts them to a list of files using EventLookup service and DQ2, makes a temporary dataset
 - The dataset could be used for data transfer
- Integration with DaTRI or client tools?

Multi Cloud

- A site can be associated to multiple clouds
 - The site can run jobs in associated clouds
- `schdconfig.ddm` takes a list of clouds
e.g., `ddm=DE,CA,FR`
 - The first in the list is defined as 'home cloud'
 - Input/output files are transferred from/to T1 of task's cloud instead of home cloud
- T1 can also be associated to foreign cloud to be used as T2
 - `T1_PRODDISK` is used for input/output files
- The brokerage calculates the weight using the number of running/queued jobs at the site per cloud
 - Low prio jobs at siteX in cloudA could be stuck due to high prio jobs at the same site in cloudB → More aggressive reassignment has been added for activated `evgen/simul` jobs at multi-cloud sites
 - 6 hours instead of 48 hours

DDM-related

- Setting replica lifetime for dis and sub datasets
 - dis : 7days, sub:14 days
- Setting the 'hidden' attribute to dis and sub
 - dq2 enduser tools don't show them
- Automatically freezing output datasets in user's container when jobs finished/failed/canceled
 - DaTRI can transfer user datasets quickly
- Coming changes
 - File-level callbacks via ActiveMQ
 - E.g., Reprocessing jobs could be activated more quickly by using file-staged callbacks
 - Panda server is ready for testing
 - Change DQ2 SS config to send callbacks to ActiveMQ
 - How DB load is increased?
 - LFC registration by panda server
 - Core functionality of dq2-register will be extracted to API
 - Panda server will use API instead of CUI to avoid spawning redundant child processes

Near Term Plans

- File-level callbacks via *ActiveMQ*
- LFC registration by panda server
- PD2P for T1-T1
- Optimization of rebrokerage
- Using output file merging by default
- Integration of EventPicking-PD2P with DaTRI or client tools