

MicroBooNE Data Management Feedback

FIFE Workshop
June 21, 2017

H. Greenlee
M. Kirby
J. Mousseau

Data Management Tools

- Data management tools used by MicroBooNE
 - SAM & SAM4USERS.
 - IFDH.
 - dCache & enstore.
 - Jobsub.

SAM

- Overall impression: good.
 - Mostly trouble free, good support.
- MicroBooNE hasn't tried to link the sam database to an external database yet, but that is something we want to do.
 - Documentation?
- We have sometimes had issues with our sam station crashing or refusing to start projects because of running out of memory or having too many projects running.
 - More capacity would be a plus.
 - We think we have reduced the problem by fixing our production scripts.
 - Incorrect behavior of jobsub dags (see later).

SAM4USERS

- Good for users who want to archive data after the fact and didn't generate sam metadata, or who aren't smooth using regular sam.
- MicroBooNE has begun advertising and encouraging analyzers to archive samples using sam4users.
- Few weeks ago tried to use sam4users for a large-scale user-assisted production & exposed some corner cases (bugs).
 - 200 character file limit.
 - Already declared files.

IFDH

- IFDH tools.
 - Posix-like file access interfaces for grid tools.
 - Samweb client.
 - Python and command line interface.
- From time to time, MicroBooNE has had requests for features or bug fixes, but these have addressed.
 - Support is pretty good.
- Some posix-like features are missing due to limitations of underlying grid tools. Exmples:
 - Make or read a symbolic link.
 - Delete a directory tree (“rm -r”).

dCache

- Performance & reliability of dCache has improved over the years.
 - Still not perfect. Higher latency and error rate than a regular nfs filesystem.
- Posix vs. grid interface.
 - Posix interface is definitely more performant, due to not having to authenticate each operation.
 - For this reason, we normally prefer posix access (good practice?).
 - Every posix access to dCache is a hang risk.
 - MicroBooNE has been forced to equip our production scripts with a very high level of error handling when dCache is involved.

dCache Scratch vs. Persistent vs. Tape-Backed

- MicroBooNE production scripts are written to store temporary results in dCache scratch (/pnfs/uboone/scratch).
 - Don't have to worry about cleaning up.
 - Don't have to worry about persistent dCache space filling up.
 - There is potentially an issue due to the fact the dCache scratch doesn't have a minimum guaranteed lifetime.
 - There is some evidence that files sometimes disappear long before the advertised dCache average lifetime.
- Tape-backed dCache. It is hard to tell whether files are staged or not.
 - It would be nice if there was a way to do a sam query for staged files (feature request).

Jobsub / Fifebatch

- Availability of fifebatch servers is not very good.
 - Often overloaded/unavailable, need more capacity.
 - MicroBooNE would be happy to have our own fifebatch server.
 - Job submission is subjectively slow.
- The DAG issue.
 - DAGs used by jobsub_submit_dag have the unfortunate feature that if any job returns an error status, dagman aborts the DAG.
 - Dagman kills entire batch cluster if any process returns error.
 - Sam stop project job won't run if any prior job in project ends with error status.
 - MicroBooNE has a long-standing feature request to teach DAGs to ignore error status of individual jobs.