

Running D0 batch jobs on Fermigrid

K. Herner

June 6, 2014

Abstract

As dedicated D0 computing resources slowly diminish over the next few years the experiment needs to be able to run its batch jobs elsewhere. Fermilab's onsite batch computing resources (collectively "Fermigrid") provide a natural home for future computing needs. This document describes the development of the job submission and file delivery system within the Run II Data Preservation project and describes the changes to the existing systems.

1 Introduction

Support for the existing D0 computing infrastructure will continue through the end of FY 2016, or five years after the Tevatron shutdown. CAB will continue to operate to this date but we will need an alternative for any jobs that may run after 2016.

The Fermilab general purpose farm (part of Fermigrid) has adequate resources to meet D0's computing needs. In order for D0 to use Fermigrid, the experiment needs to modify its job submission tools and develop a way to deliver files from SAM to worker nodes. First, D0 needs to shift to using the `jobsub_client` package in place of `qsub` commands, and files must be delivered via `dCache` and `gridftp`. Both of these changes are largely complete and we describe them in this document.

2 The `jobsub_client` package

When one submits jobs to Fermigrid, commands from the `jobsub_client` package [1] will replace the PBS commands traditionally used for sending

jobs to the clued0 or CAB batch systems. The package uses condor behind the scenes but the end user only needs to know the base jobsub commands. Numerous options are available for customization.

3 Submitting jobs to Fermigrid

NOTE: As of May 2014 the following will only work from the d0mino04, d0mino05, and d0mino06 machines.

3.1 Submitting a test job

Every user should first try submitting a simple test job to Fermigrid. This will exercise the authentication systems and give a basic feel for the new system. To send a test job, do the following:

- Log into one of the d0mino0X machines.
- Get to a Bash shell by running `/bin/bash` (Note that simply running “bash” will run the bash version in `/usr/local/bin` by default, which is very old.)
- `source /d0/app/users/kherner/simpletest/simpletest_setup.sh`
- The setup script will create a test output directory for you and will print two commands for you to do. One is the `get-cert` command and the other one is the actual job submission command.

The `simpletest_setup.sh` and `simpletest.sh` scripts (both in `/d0/app/users/kherner/simpletest`) are heavily commented to ease understanding of what they do. The reader is encouraged to study both scripts to gain additional insight.

3.2 Additional submission options

Some of the important submission flags that `jobsub_submit.py` accepts include the `-f`, `-d`, and `-e` flags. The `-f` flag is used for copying in specific files, but is unlikely to be needed at D0. The `-e` flag passes environment variables to the job so that they have the same value within the job as they do in the shell that submitted the job. The `-e` option can be used as many times as needed. The `-d` option is for copying output back. It consists of

two parts, a tag and an output directory. There are examples for both the `-e` and `-d` options in `simpletest_setup.sh` and `simpletest.sh`, both available in `/d0/app/users/kherner` on any of the `d0mino0X` machines. Additional documentation is available at:

https://cdcvns.fnal.gov/redmine/projects/jobsub/wiki/Using_the_Client

3.3 Getting output to the right place

If you use the `-d` flag your output will go to the directory that you specify. Note that the legacy disk areas available on `clued0` such as the `/work`, `/rooms`, or `/prj_root` areas will not be available. For the initial copyback, one should generally choose a directory in the `/d0/data` area on `Bluearc` such as `/d0/data/condor-tmp/<username>/<path of your choice>`. **Important note:** The main output directory needs to have group write permission because the Fermigrid jobs run under the `dzeroana` account rather than under your normal user account. The `simpletest` procedure above gives an example of using the `-d` flag. In the future we hope to automate this step within `d0tools` and `caf.tools`. Once the jobs are done, you can use `scp`, `rsync`, `rcp`, `bbcp`, or any program you choose to copy the output to the final storage location of your choice. For example from the `d0mino0X` machines you can do something like `rsync -r /d0/data/condor-tmp/<username>/<path of your choice> somenode-clued0:/full/path/on/clued0`.

3.4 Tracking/removing jobs and getting log files

The equivalent command to `qstat` in the `jobsub_client` setup is `jobsub_q.py`. It does require that the user set up the `jobsub_client` package first. A usage example is below:

```
jobsub_q.py -G dzero --jobsub-server=https://fifebatch1.fnal.gov:8443
```

That will list all jobs that the user has in the queue. To stop or remove a submitted job, do

```
jobsub_rm.py -G dzero --jobsub-server=https://fifebatch1.fnal.gov:8443  
--jobid <id number>
```

By default, the `stdout` and `stderr` log files do not get copied back along with the rest of the job (these are equivalent to the `<name>.o<job number>.e*` files that come from `D0` jobs to `CAB` or `clued0`) but are stored on the server. To get these files, run the `jobsub_fetchlog.py` command, e.g.:

```
jobsub_fetchlog.py -G dzero --jobsub-server https://fifebatch1.fnal.gov:8443
--job N
where N is the job ID number.
```

4 File Delivery

The SAM system has served D0 well throughout Run II. While SAM will remain available well past 2020, some D0 infrastructure and interface protocols will need to change in order to keep up with planned changes to SAM and to minimize the support effort required for the D0 instance. One of these changes, the use of samweb, was already implemented for D0 in 2013. The other major change concerns delivering files to farm jobs, and involves a dCache instance for D0 and making use of the ifdhc software package developed at Fermilab.

4.1 D0 dCache instance

To date D0 has used a large array of SAM cache disks to store files requested from tape and to quickly transfer them between the worker nodes. The SAM cache totaled about 1 PB at its peak, but there will be no additional purchases and the existing disks will slowly die off or be retired. We have created a dCache instance for D0 to act as an eventual replacement for SAM cache [2]. At present dCache is widely used through HEP by experiments in all three frontiers. It has a very large user base at Fermilab and we can be confident that it will be easy to get support for the D0's dCache instance throughout the Data Preservation period.

D0's dCache is called `d0dca` and there are extensive information and monitoring pages available at <http://d0dca.fnal.gov>. Figure 1 shows a snapshot of the home page.

Operationally the dCache instance is not very different from SAM cache: as a job requests files from SAM, the files will be copied out of dCache to the worker node as described in the next section. If the file is not already in dCache, it will be copied in from Enstore. Interactive copying of files is possible through GridFTP. dCache also requires a new SAM station (version 9 or greater) in order to use the SAM+dCache features. The new station is currently called `d0_dcacheltest` and is fully operational, although the name may change in the future.

Browser address bar: d0dca.fnal.gov

Navigation menu: Most Visited, Latest Headlines, D0 Pages, STT Pages, News, DataPreservation, SCC

D0 dCache System Status

Detailed System Status	D0 dCache internal status
Recent FTP Transfers	History of recent FTP transfers
Active Transfers	Current and pending transfers
Plots Billing	Data movement plots and daily billing
File Lifetime Plots	Plots of file lifetime, last access time
Pool Directory Listings	Daily snapshot of files in cache
Detailed Statistics	Internal statistics for pools, file families
Queue Plots Sum	Plots of pool queue occupancies
Login List Restore List	Lists of dCache logins and restores
Alarms	Enstore alarms
Meta-Data Checks	PNFS internal consistency monitoring
MSS Servers Transfers	D0EN Enstore summary, servers, encps

Information

dCache Project Home	dCache Project global home page
dCache User Primer	FNAL dCache Primer and User Guide
FNAL MSS Home	FNAL Mass Storage System home page
Operations E-mail	FNDCA dCache operations e-mail list
Community E-mail	FNAL dCache user discussion e-mail list
Running dCache	Admin instructions on running dCache

[Legal Notices](#)

Figure 1: Screenshot of D0 dCache home page.

Important note: when running SAM projects on Fermigrid you **must** use SAMWeb. If you are running a cafe job, add “SAM.UseSAMWeb: true” to your list of cafe arguments when running your job.

Using ifdhc to transfer input files

As mentioned, the legacy disk areas available on clued0 such as the /work, /rooms, or /prj_root areas will not be available on Fermigrid worker nodes. Thus any input tarballs or files need to be pre-positioned in a place available to the worker nodes to be copied in, such as /d0/data on the d0mino0X machines.

The ifdhc package is used by several experiments to transfer files during jobs, and includes mechanisms to prevent Bluearc overloading by a single experiment or user; in this way it is similar to the `limit_transfers` packages used by D0. It is available as a UPS product on the Fermigrid worker nodes. For running on Fermigrid, **all input files should be transferred with the ifdh cp command; do not do any copying on your own in the job script.** A simple example follows that can go in your script for copying in some input tarball:

- `setup ifdhc -z /grid/fermiapp/products/common/db`
- `cd $_CONDOR_SCRATCH_DIR`
- `ifdh cp -D /full/path/to/input/file.tar.gz ./` (the -D option tells ifdh that the last item is a directory)
- `tar mxzf file.tar.gz`

5 Running analysis jobs

More advanced setups are easily possible for sending Fermigrid jobs. As part of the testing steps the Data Preservation group has successfully sent SAM projects to Fermigrid that run cafe jobs in the vjets framework. All other types of D0 jobs are possible, though at the moment users will need to create their own job scripts. These are not difficult to make but one should keep the following keys in mind when making them:

- Initial setup of the D0 CVMFS areas. The `simpletest.sh` script provides an example that should suffice in nearly all cases.

- Setup of the D0RunII release. Again, not all legacy releases are available.
- Be sure to run any other initialization scripts that your job requires (similar to specifying an init script in d0tools or runcafe)
- Be sure that the script does a `cd` into `$_CONDOR_SCRATCH_DIR` before copying in any files or creating any output files
- Set the `PATH` and `LD_LIBRARY_PATH` variables as needed if not done in your init script
- Use `ifdh cp ...` to copy in any input files.
- At the end of the job move the files that you want to copy in the appropriate output directory. If you are using the `-d` flag as in the `simpletest.sh` script, you do not need to write any code to do the copying to the final output directory; just put the files in the output directory to be copied.

5.1 SAM Project considerations

As mentioned above it is possible to run SAM projects on Fermigrid. If you are running a D0 framework executable or a cafe executable, you just need to remember to pass, at minimum, the `SAM_PROJECT` and `SAM_STATION` environment variables to your job using the `-e` option for each one in `job-sub-submit.py`. `SAM_PROJECT` should be whatever the name of your project is (or the samweb URL that points to it) and `SAM_STATION` needs to point to the dCache-enabled station, which is `d0_dcache_test` at the moment. Also, you must use SAMWeb and add that to your cafe arguments as described in Sec. 4.1. When starting the project you must use `d0_dcache_test` as the SAM station. Users are also at this time responsible for issuing the `samweb stop-project $SAM_PROJECT` command when all jobs are done (in the future we will incorporate that into the usual job submission tools.)

5.2 CVMFS considerations

While the CVMFS repository provides D0 software release and required UPS packages, one must keep in mind that the UPS product and software release

locations do not have the same absolute path on Fermigrid; the absolute path is `/cvmfs/d0cfs.fnal.gov/<usual location>`. Therefore one has to take care to check for such hard-coded absolute paths that may be present in various pieces of software.

5.3 Future– modifications to d0tools

We have seen that a number of the replacement commands could easily be added into the usual job submission toolkits such as `d0tools` and `caf_tools`. We are now working to include this functionality in order to make the transition to Fermigrid as transparent as possible for end users.

6 Summary

The D0 experiment needs alternatives to the traditional batch computing infrastructure after 2016 in order to ensure the long-term ability to submit new analysis jobs, one of the central goals of the Data Preservation project. We now have that ability to send jobs to the Fermilab’s general purpose computing cluster (part of Fermigrid) and have the ability to deliver files from SAM to the worker nodes via a new `dCache` implementation. Taken together, and following the conclusion of efforts to adapt the usual job submission tools, this infrastructure should meet the experiment’s computing needs through the end of 2020.

References

- [1] <https://cdcvs.fnal.gov/redmine/projects/jobsub/wiki>
- [2] <http://www.dcache.org>