Fermilab | U.S. DEPARTMENT OF ENERGY | Office of Science

Jobsub - Best Practices

Dennis Box

FIFE Workshop

21st June 2016

FABRIC FOR FRONTIER EXPERIMENTS

Best Practices:

- Kens' running on OSG talk pretty much covered it

- Use Resources Wisely

Whats New - Since Last Workshop

- **Improved Job Query and Control**
  - **Hold, Release, Remove**
    - by user
    - by condor_constraint (ex: all ‘dbox’ jobs in held state)
    - super users per experiment
  - **Jobsub Q**
    - —better-analzye (why are my jobs not starting?)
    - —hold (show held jobs only)
    - —idle (show idle jobs only)
  - **Jobsub History**
    - much faster response time
- **Improved Resource Request Mechanism**
  - help your jobs land on machines that meet your jobs needs
  - easier to tailor resource requests land on more available machines
    - result: they start and finish more quickly

Best Practices - Submitting Jobs

- Use file transfer flags (-f , -d) to let jobsub figure out the best way to transfer your input/output
  - these use ifdh behind the scenes
  - they work with dCache and BlueArc locations
  - prefer dCache when possible
- Jobs requiring and requesting less resources will generally start and finish before ones needing more
  - the pool of available machines to run them is larger
  - they have a better chance of finishing before being preempted
  - use —memory, —disk, —cpu, —expected_lifetime to request this

Best Practices - jobsub_submit (2)

- Structure your jobs so that -N (number of copies of job) is not excessive
  - creates a directory with at least 4 files per job on server - easy to get unwieldy number of files
  - submits all N jobs at same time, overloading schedd
  - fetchlog of this directory creates and transfers a huge tearful
- max allowed value of -N currently 10000
- again - break this into a series of smaller jobs if you can!

Best Practices - Jobsub_hold, release, rm

- —user , —constraint can cut down on how much typing you need to do
- examples
  - jobsub_q -G nova --constraint '(JobStatus=?=5)&&(Owner=?="dbox")' # see dbox's held jobs
  - jobsub_rm -G nova --constraint '(JobStatus=?=5)&&(Owner=?="dbox")' #remove dbox's held jobs

Best Practices - Jobsub_q

- jobsub_q (no other spec) I/O intensive
- jobsub_q —long (no other spec) very I/O intensive
- —group, —user, —jobid, —constraint reduce I/O expense
- think twice about using these for polling in a loop!
  - if you need to poll for job completion, its a good candidate for a DAG

‘Worst Practices’

- misbehaving cron jobs
  - use flock to make sure you’re only running one copy of cron job
  - make sure cron job isn’t doing something stupid.

- jobsub_fetchlog
  - failing to use —partial if you only want a small portion of the logs

Tips and Tricks

- —timeout is a good way to retrieve jobs that you know are getting stuck
- ifdh log “I am doing this on machine $HOSTNAME” another useful debugging technique
- jobsub_q —long a good way to figure out how to build -constraints
- POSIX access to libraries and data files is deprecated
  - use cvmfs
  - custom libraries can be uploaded using dropbox://my_custom_stuff.tar
- File redmine or SNOW tickets! We can’t fix what we don’t realize is broken

Conclusion

- Jobsub makes submitting to the OSG much easier
- Save bandwidth, typing, and time using —constraint for querying and job control
- Specify —memory —disk —expected-lifetime to maximize throughput
- The girl scouts have a good philosophy about resource usage.