FIFE Workshop
Jobsub Best Practices

SUBMITTING YOUR JOBS TO THE OSG USING JOBSUB – UPDATES AND BEST PRACTICES

DENNIS BOX
PARAG MHASHILKAR

Jobsub Server

- A re-design of 'old' jobsub_tools to address short comings
- Multiple servers behind single IP address makes service more scalable, highly available
  - 25K running jobs on 2 physical servers as of 5/28/15
  - approximately 3x capacity of old jobsub_tools submission node
- Server accepts almost all of the jobsub_tools input flags and arguments.
  - Your old submission scripts should not need to change very much to run on the new system
  - The new server runs jobsub_tools to generate a condor JDF and submit the users jobs

Best Practices

- Much is the same as jobsub_tools Best Practices
- Protect shared resources
  - Use the –f and –d flags to automate ifdh transfer of data to and from worker nodes
  - Use ifdh anyway even if you don't use the –f and –d flags to avoid direct reads/ writes to/from Bluearc
  - Copy your remote file to local disc on worker node and access it from there
- Size your jobs for maximum efficiency
  - 'Short' duration jobs are inefficient – relatively high amount of time spent authenticating, setting up environment, transferring files compared to actual work
    - See if you can combine these into longer jobs .
    - I have been told 8 hrs run time is a good target
  - Submitting > 1000 processes/cluster strains condor daemons
    - Can these be combined into longer jobs with fewer processes?

Best Practices: jobsub_q, jobsub_history

- jobsub_q and jobsub_history with the wrong options are, like the condor_q and condor_history commands they run on the server, very expensive
  - Avoid jobsub_q –l queries of multiple jobids at the same time
  - Avoid jobsub_history if you can – use jobsub_fetchlog –list instead.
    - jobsub_fetchlog –list combined with jobsub_q is a workaround, we acknowledge that jobsub_history needs improving.

Using the client

- Majority of users set up client from ups on bluearc or cvmfs
  - source /cvmfs/fermilab.opensciencegrid.org/products/common/etc/setups
  - setup jobsub_client
  - jobsub_submit (client options) file:///path/to/script.sh (user_script_options)
  - This will
    - Copy script.sh to the server
    - Create a condor submission file based on (client_options)
    - Submit a condor job to the OSG, which will copy script.sh to the OSG worker node and run it with (user_script_arguments)

Steering jobs to OSG sites

- Recall: Jobsub_submit (client opts) file://path/to/script.sh (script_opts)
  - Client opts vary by supported experiment (aka group)
  - jobsub_submit –G nova –-help to see all available options for NoVA experiment
- Ex: Steering jobs to fermigrid OSG site (where FIFE groups have dedicated quotas and priorities):
  - Jobsub_submit –G lbne –resource-provides usage_model=DEDICATED file://lbne_script.sh
- Ex: Steering jobs to fermigrid OSG site, taking any available open slot:
  - Jobsub_submit –G lbne –resource-provides usage_model=OPPORTUNISTIC file://lbne_script.sh

Steering pt II

- Ex: Steering jobs to 'Omaha' OSG site:
  - Jobsub_submit –G lbne –resource-provides usage_model=OFFSITE –site Omaha file://lbne_script.sh
- I want to know what 'usage_model's are configured.
  - Ask your Fife experiment liason. Or you can query directly:
  - jobsub_submit -G lbne --resource-provides usage_model=I_HAVENT_A_CLUE file://lbne_script.sh
  - illegal --resource-provides value: I_HAVENT_A_CLUE for option: usage_model is not supported on fifebatch2.fnal.gov according to your config file /opt/jobsub/server/conf/jobsub.ini. Legal values are:FERMICLOUD_PRIV1,FERMICLOUD_PRIV,FERMICLOUD_PP_PRIV1, FERMICLOUD_PP_PRIV,FERMICLOUD_PP,FERMICLOUD,OFFSITE,PAID _CLOUD,DEDICATED,OPPORTUNISTIC,SLOTTEST,PAID_CLOUD_TEST
  - Ask before using any besides DEDICATED, OPPORTUNISTIC, OFFSITE. Your jobs probably wont start without extra configuration

Steering pt III

- Where can I send jobs with –resource-provides=OFFSITE –site depends on the group. Jobsub_status will tell me the configuration for a particular group:
  - jobsub_status -G lbne --sites
    - WT2
    - Nebraska
    - TTU
    - SU-OG
    - Caltech
    - Wisconsin
    - (a bunch more sites)
  - Note that your job will try to land there, but it is not guaranteed to run to completion
    - The site may be busy and giving you a low priority
    - It may be mis-configured for your job and either never start or start and fail right away.
    - Ask your experiment liaison what sites are currently working.
      - If a site should be working but is misconfigured they can open an OSG ticket to have it fixed.

The Wrapper Script and Data Transfer

- Jobsub generates a 'wrapper script' that is transferred to the worker node along with your user_script. The 'wrapper script' does the following:
  - Sets up ifdh via cvmfs or bluearc
  - Transfers in any files specified via –f with ifdh
  - Transfers in any tarball specified via –tar_file_name and unwinds it
  - Calls the user_script and saves its exit code
  - Transfers any output specified by –d flag back via ifdh
  - Exits with the saved exit code

Input Transfer Options

- Some options for getting an application and data on to the worker node
  - Access from cvmfs
  - Transfer in a tarball
  - Pulled in with –f option
  - Pulled to WN using ifdh directly
  - Whatever you choose, DO NOT access them directly from Bluearc. It is temptingly convenient but Its. Just. Bad.

Input File Tradeoffs

- Cvmfs:
  - Cached, efficient
  - Not meant for rapidly changing files. May take a while for caching to make your latest libraries available.
- IFDH:
  - Decides on 'best' method for data transfer
  - Protects shared resources (gridftp servers) via locking/queuing
  - Queuing can make it less than ideal for sending same file to 100s of worker nodes – they may sit idle a long time waiting for same small file
- Dropbox/Tarball
  - Good choice for sending same file to lots of worker nodes
  - Avoids ifdh lockfiles – uses default condor transfer mechanisms
  - Tarball convenient if you are changing libraries faster than they can be picked up by cvmfs

Output File Transfer

- Use ifdh to copy back from worker nodes
  - dcache scratch preferred over bluearc
- Method 1:
  - Ifdh cp to dcache
  - Pin files or write to the new non volatile area
- Method 2:
  - Ifdh cp to dcache
  - Use fife_utils to create SAM dataset out of a dcache location
    - sam_add_dataset does this with correct arguments
    - sam_clone_dataset to add files
  - See Andrew Normans talk for more details
- Method 3: (if you *must* write to bluearc)
  - Use IFDH_STAGE_VIA=srm://fndca1.fnal.gov.....
  - See Marc for further details

Jobsub Documentation

- Wiki Pages
  - https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Introduction_to_FIFE_and_Component_Services#Jobsub
  - https://cdcvs.fnal.gov/redmine/projects/jobsub/wiki#Client-User-Guide
  - https://cdcvs.fnal.gov/redmine/projects/jobsub/wiki/Using_the_Client