Wiki

Clone wiki Create page

Sin-Bad / Rocks Cluster

View History Edit Delete

Overview

To login, go to *rocks-login.is.cs.cmu.edu*; rocks.is.cs.cmu.edu is not used for login. If things seem to be going wrong, please at this point send e-mail to Florian and help. To access files in AFS, the users will need to issue a kinit and an aklog command to get the required credentials.

Most filesystems (AFS, NFS) should be visible on Rocks (only those hosted on non-public nodes on the existing cluster are not). Please note that your rocks home-directory is not on AFS, but on a separate partition on the cluster for stability reasons. You can however link directories, files etc. to your AFS home directory, if you so desire. Recent versions of some tools have been installed in /opt, so you can check there, before compiling your own autoconf, etc. If you require certain software, facilities may be able to install it for everybody (check http://rocklinux.net/ and send e-mail to help).

To see the machines, visit Ganglia from within the CS network.

Here is a recipe/ readme from Roger on how to use the PBS scheduler (Maui/ Torque flavor):

```
submit.sh contains description about several commmonly used options for qsub.

simple.sh represents the job to be submitted to the scheduler.

some useful commands

qsub : see submit.sh
qdel <job id> : stop and remove a job from the queue
qstat : look at the queue
qhold : hold a job. That job will not be executed.
qrerun : try to rerun a job

diagnose -n : see all machines status
diagnose -j <job id> : check your job
checkjob <job id> : similar to diagnose -j, with more details
diagnose -f : check fair share status
diagnose -u : check user status
diagnose -p : list the idle jobs in the queue, you can see the priority.
```

The attached two files (submit.sh and simple.sh) show how to use the cluster (please don't delete them). You can get these files from this Sin-Bad project's Repository link. You can also use qsub -I to get an interactive shell through the scheduler.

File Systems

Space in the home directories is limited, therefore, there is a quota. Also, your home directory is readable only by you, and this is ideally kept this way. There are a number of other storage areas, e.g.

- **/people/store*N*** (with *N*=0...5) - this is a "data graveyard" of slow RAIDs in Craig St. Use it to store data that you do not need often, or maybe not anymore, but which you still don't want to delete. **DATA**, **RECORD**, and **PROCEEDINGS** live here, too.
- **/data/ASR*N*** (with *N*=1...5) - main directories for running ASR experiments, this is a NAS RAID
- **/data/MM*N*** (with *N*=1...3, 2?, 3?) - main directories for running MM experiments and storing data (in /data/MM3, I believe), a NAS RAID
- /scratch - this is a **local** file-system, much like /tmp. compute-0-29:/scratch can be seen as /compute/compute-0-29 on other cluster machines, so you can use it to exchange data across machines. This is to be treated strictly like temporary storage, e.g. data can disappear at any time and make sure to clean up after you.

None of these file-systems are under back-up, not even /home. You are on your own. **Make sure you don't fill up file-systems.**

Scheduler

The scheduler currently has four queues:

- *standard* is the "normal" queue which enforces a 36h maximum wall-clock time of your job
- *long* is for jobs that need to run for more than 36h (unlimited time), these can not take up more than 25% of the cluster's slots at any given time

given time

- *testing* is for testing, the maximum wall-clock time per job is 2h, and some low-powered nodes are exclusive to this queue, so there should always be "availability"
- *gpu* for computing on K20 GPUs **(see below)**

Please let us know if the system doesn't seem to behave according to this policy, or if you think the policy should be changed.

Please make sure to either re-direct stdout/ stderr of your jobs to NFS, or don't write a large amount (several Gb) of log output. It will be buffered on the compute node, and might fill up the local filesystem, breaking the submission of new jobs for other users.

Makefile (also attached) can be used to submit a whole "directed acyclic graph" of jobs to PBS, similar to what Condor does. You specify a target ("test") and the dependencies ("train", ...), define what pattern you want to use for each step ("mapreduce", "simple", "iterative", ...), and the Makefile produces a shell script which you can submit (it can also do that for you), plus the shell scripts for the individual steps. It is a first shot at the problem, so feel free to improve...

Almost all the machines have ~2TB available locally under /scratch for temporary data. Please do not store persistent data on these drives. All of the compute nodes are exporting their /scratch filesystem(s) cluster wide. We've also created an autofs group for the compute nodes. Any given compute node's scratch space is now available cluster wide under /compute/<hostname>. We may introduce a policy to automatically delete the oldest data, if these drives fill up with junk. We do have the /people/storeN/ RAIDs (N=[0,5]) for data you want to keep around, in addition to the normal RAIDs for experiments.

Here's a few useful macros:

```
# User specific aliases and functions
export   JANUS_LIBRARY="${HOME}/janus/library"
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:/opt/tcltk-8.5.10/lib:${HOME}/tools/portaudio/lib/.libs:${HOME}/tools/ffmpeg-0.7.4/libs"
export           PATH="${PATH}:${HOME}/bin"

# Convenience aliases
alias pbsq='qstat -n -u `whoami`'
function pbsl() { ssh `qstat -n $@|awk -F/ ' /compute/ { print \$1 } '` cat "/opt/torque/spool/$@.*"|less ;}
function pbsf() { ssh `qstat -n $@|awk -F/ ' /compute/ { print \$1 } '` tail -f "/opt/torque/spool/$@.*" ;}
function pbst() { ssh `qstat -n $@|awk -F/ ' /compute/ { print \$1 } '` top -b -n 1|head -23 ;}
```

Kaldi experiments on ROCKS

Also attached find example modified Kaldi run scripts that let you run experiments in the "Kaldi standard" way on our ROCKS cluster, rather than Sun GridEngine (the default). Replace both cmd.sh and utils/queue.pl in the Kaldi experiment working directory. Also for any steps that run mkgraph, replace them with a reference to $mkgraph_cmd with the appropriate amount of memory. This ensure that they get scheduled to run on a node that has sufficient memory (NOT rocks-login!)

GPUs

To use the GPUs (and Theano/ Detl):

- The GPU nodes are (currently) compute-0-25, compute-0-36 and compute-1-36, each has 4 K20 GPUs (12 GPUs in total on the three nodes)
- To submit to the GPU queue, do something like

```
# Submit to the GPU queue
qsub -q gpu script.sh
```

- You need to specify explicitly on which of the 4 GPUs to run, the script to submit to the scheduler should look like

```
# Determine the GPU to use from the PBS JobId
export gpu=`qstat -n $PBS_JOBID|awk ' END { split ($NF, a, "/"); printf ("gpu%s\n", a[2]) } '`
```

Some more code examples (do not use without understanding)

```
#!/bin/sh
# Submit something to the GPU and set up Theano/ Detl correctly
# ~/.theanorc contains:
# [global]
# base_compiledir = /var/tmp/<username>

source /data/ASR1/tools/theanoenv/setup
export PATH=/data/ASR1/tools/python27/bin:$PATH
export PYTHONPATH=/opt/python27/lib/python2.7/site-packages:/data/ASR1/tools/theanoenv/lib/python2.7/site-packages:$PYTHONPATH

# Get GPU slot we will be using from PBS_JOBID
```

```
gpu=`qstat -n $PBS_JOBID|awk ' END { split ($NF, a, "/"); printf ("gpu%s\n", a[2]) } '`
export THEANO_FLAGS="cuda.root=/usr/local/cuda,device=$gpu,floatX=float32,config.nvcc.fastmath=True,allow_gc=False"
```

- Another option (rather than using the above) would be to use "nvidia-smi -c 1", to switch the GPUs to "thread exclusive" mode - but the system is not currently configured to do this
- If you need to work interactively, please also use the above procedure to make sure the scheduler knows that you're using a certain GPU
- Local storage is available in /scratch, you may want to copy data locally to make sure GPU training runs efficiently

Other Resources

Other useful web-pages related to PBS variants (feel free to add):

- http://rcsg.rice.edu/ada/FAQ/scheduling.html
- https://wiki.hpcc.msu.edu/display/hpccdocs/Submitting+a+Job

Updated 2014-11-23