Hands-on exercises on

Playing with JOB scheduler

Introduction

Computer clusters

This activity requires successful installation of a 2 node HPC cluster based on the ROCKS - Open Source Toolkit for Real and Virtual Clusters (http://www.rocksclusters.org/) covered in a prior hand-out.

NOTES:

- The most recent ROCKS DVD may be downloaded from http://www.rocksclusters.org/
- After installation, you can use a web-browser to view the documentation on the local server using the URL http://localhost/ or http://localhost/roll-documentation/
- URL for monitoring
  - http://localhost/ganglia/
- On-line documentations and references may be found at

  http://www.rocksclusters.org/roll-documentation/

Activity

Adding normal users

Requirements

- Working ROCKS cluster (master and compute-0-0)

Steps

Login as a root user

Open a terminal Window

Run the following commands, replacing *{username}* with expected login-name

useradd *{username}*

Set a password on the new account, replace *{username}* with expected login name

passwd *{username}*

Sync the account information across the ROCKs cluster

rocks sync users

Activity 2

Using SGE for job submission

Requirements

- Working ROCKS cluster (master and compute-0-0)

Steps

- Login as a normal user
- Create a job script with a suitable text editor (pico or nano) containing the following lines

```
#!/bin/bash
#
#$ -cwd
#$ -j y
#$ -S /bin/bash
#
date
/usr/bin/openssl speed
date
```

Option	**Explanations**
-cwd	Run in the current working directory
-j y	Stdout and stderr in the same output file
-S /bin/bash	Use the bash shell for running the job
-M *{email-address}*	Send notifications about job to email-address
-o {*filename}*	Send output into file *{filename}*
-e *{filename}*	Send stderr into file *{filename}*

- Save the file as test.sh

- Submit the job using the qsub command

  qsub test.sh

  Note: you can submit the test.sh job multiple times, repeat the qsub command about 5 times.

- Check the status of your job using the command qstat

  qstat

  qstat -f

- Delete one of the jobs using the qdel command

  qdel

- Other commands are qconf for checking which queues are available

  qconf –sql

  qmod –d to disable a queue and qmod –e to enable a queue

The SGE roll documentation roll-sge-usersguide.pdf contains an example for a parallel job.

					28-Apr-2013
User Commands	**PBS/Torque**	**Slurm**	**LSF**	**SGE**	**LoadLeveler**
Job submission	qsub [script_file]	sbatch [script_file]	bsub [script_file]	qsub [script_file]	llsubmit [script_file]
Job deletion	qdel [job_id]	scancel [job_id]	bkill [job_id]	qdel [job_id]	llcancel [job_id]
Job status (by job)	qstat [job_id]	squeue [job_id]	bjobs [job_id]	qstat -u * [-j job_id]	llq -u [username]
Job status (by user)	qstat -u [user_name]	squeue -u [user_name]	bjobs -u [user_name]	qstat [-u user_name]	llq -u [user_name]
Job hold	qhold [job_id]	scontrol hold [job_id]	bstop [job_id]	qhold [job_id]	llhold -r [job_id]
Job release	qrls [job_id]	scontrol release [job_id]	bresume [job_id]	qrls [job_id]	llhold -r [job_id]
Queue list	qstat -Q	squeue	bqueues	qconf -sql	llclass
Node list	pbsnodes -l	sinfo -N OR scontrol show nodes	bhosts	qhost	llstatus -L machine
Cluster status	qstat -a	sinfo	bqueues	qhost -q	llstatus -L cluster
GUI	xpbsmon	sview	xlsf OR xlsbatch	qmon	xload
Environment	**PBS/Torque**	**Slurm**	**LSF**	**SGE**	**LoadLeveler**
Job ID	$PBS_JOBID	$SLURM_JOBID	$LSB_JOBID	$JOB_ID	$LOAD_STEP_ID
Submit Directory	$PBS_O_WORKDIR	$SLURM_SUBMIT_DIR	$LSB_SUBCWD	$SGE_O_WORKDIR	$LOADL_STEP_INITDIR
Submit Host	$PBS_O_HOST	$SLURM_SUBMIT_HOST	$LSB_SUB_HOST	$SGE_O_HOST	
Node List	$PBS_NODEFILE	$SLURM_JOB_NODELIST	$LSB_HOSTS/LSB_MCPU_HOST	$PE_HOSTFILE	$LOADL_PROCESSOR_LIST
Job Array Index	$PBS_ARRAYID	$SLURM_ARRAY_TASK_ID	$LSB_JOBINDEX	$SGE_TASK_ID	
Job Specification	**PBS/Torque**	**Slurm**	**LSF**	**SGE**	**LoadLeveler**
Script directive	#PBS	#SBATCH	#BSUB	#$	#@
Queue	-q [queue]	-p [queue]	-q [queue]	-q [queue]	class=[queue]
Node Count	-l nodes=[count]	-N [min[-max]]	-n [count]	N/A	node=[count]
CPU Count	-l ppn=[count] OR -l mppwidth=[PE_count]	-n [count]	-n [count]	-pe [PE] [count]	
Wall Clock Limit	-l walltime=[hh:mm:ss]	-t [min] OR -t [days-hh:mm:ss]	-W [hh:mm:ss]	-l h_rt=[seconds]	wall_clock_limit=[hh:mm:ss]
Standard Output File	-o [file_name]	-o [file_name]	-o [file_name]	-o [file_name]	output=[file_name]
Standard Error File	-e [file_name]	e [file_name]	-e [file_name]	-e [file_name]	error=[File_name]
Combine stdout/err	-j oe (both to stdout) OR -j eo (both to stderr)	(use -o without -e)	(use -o without -e)	-j yes	
Copy Environment	-V	--export=[ALL | NONE | variables]		-V	environment=COPY_ALL
Event Notification	-m abe	--mail-type=[events]	-B or -N	-m abe	notification=start|error|complete|never|always
Email Address	-M [address]	--mail-user=[address]	-u [address]	-M [address]	notify_user=[address]
Job Name	-N [name]	--job-name=[name]	-J [name]	-N [name]	job_name=[name]
Job Restart	-r [y|n]	--requeue OR --no-requeue (NOTE: configurable default)	-r	-r [yes|no]	restart=[yes|no]
Working Directory	N/A	--workdir=[dir_name]	(submission directory)	-wd [directory]	initialdir=[directory]
Resource Sharing	-l naccesspolicy=singlejob	--exclusive OR--shared	-x	-l exclusive	node_usage=not_shared
Memory Size	-l mem=[MB]	--mem=[mem][M|G|T] OR --mem-per-cpu=[mem][M|G|T]	-M [MB]	-l mem_free=[memory][K|M|G]	requirements=(Memory >= [MB])
Account to charge	-W group_list=[account]	--account=[account]	-P [account]	-A [account]	
Tasks Per Node	-l mppnppn [PEs_per_node]	--tasks-per-node=[count]		(Fixed allocation_rule in PE)	tasks_per_node=[count]
CPUs Per Task		--cpus-per-task=[count]			
Job Dependency	-d [job_id]	--depend=[state:job_id]	-w [done | exit | finish]	-hold_jid [job_id | job_name]	
Job Project		--wckey=[name]	-P [name]	-P [name]	
Job host preference		--nodelist=[nodes] AND/OR --exclude=[nodes]	-m [nodes]	-q [queue]@[node] OR -q [queue]@@[hostgroup]	
Quality Of Service	-l qos=[name]	--qos=[name]			
Job Arrays	-t [array_spec]	--array=[array_spec] (Slurm version 2.6+)	J "name[array_spec]"	-t [array_spec]	
Generic Resources	-l other=[resource_spec]	--gres=[resource_spec]		-l [resource]=[value]	
Licenses		--licenses=[license_spec]	-R "rusage[license_spec]"	-l [license]=[count]	
Begin Time	-A "YYYY-MM-DD HH:MM:SS"	--begin=YYYY-MM-DD[THH:MM[:SS]]	-b[[year:][month:]daty:]hour:minute	-a [YYMMDDhhmm]	