Grid Engine

Simple Workflow Intro

chris@bioteam.net

Created by The BioTeam, http://blog.bioteam.net

Topics

- Talk about using SGE more effectively
- Enabling workflows & pipelines via:
  - Job Dependencies
  - Array Jobs
- Some live examples
- SGE Troubleshooting (time permitting)

Being more effective

chris@bioteam.net

Created by The BioTeam, http://blog.bioteam.net

Job Dependencies

- SGE scheduler does not promise to dispatch jobs in the order in which one submits them.
- *What if I have jobs that need to run in a certain order?*
- Imagine this scenario:
  - Step 1 - Data staging script
  - Step 2 - Data analysis script
  - Step 3 - Result QC & staging script
  - Step 4 - Cleanup script

Created by The BioTeam, http://blog.bioteam.net

Job Dependencies

- SGE Job Dependency Syntax allows for ordered job execution
- Hinges upon a simple SGE feature:
  - Job Names or jobID
- Huh?
  - We need job names or some other identifier because we can’t be sure what SGE jobID the scheduler will assign our task
  - If you use the -terse flag in qsub, you can capture the jobID
- With assignable names we can reference jobs that are already pending, holding or running

Job Dependency Example

```
qsub -N “worker1” -terse my-job-script.sh!
qsub -N “worker2” -terse my-job-script2.sh!
qsub -hold_jid worker1,worker2 cleanupJob.sh!
qsub -hold_jid jobID$1,jobID$2 cleanupJob.sh!
```

- See what we did up there?
- Our worker scripts will run when resources are available
- The cleanup script won’t run until the workers are done
- It all hinges on this:
  - By “naming” our jobs we can now reference them when using the “-hold_jid” argument.

Array Jobs

chris@bioteam.net

Array Jobs

- Extremely common use case in life science clustering:
    - *“I need to run my program 100,000 times against 100,000 different input files”*
- Most people would …
    - Use ‘qsub’ to submit 100,000 separate jobs
- This will work but is not ideal
    - Each job consumes filehandles and other system resources on the SGE qmaster host. This can slow down or even crash SGE at large enough scales
    - For users it can be a pain to monitor 100,000 jobs via ‘qstat’
- There is a better way!

Array Jobs

- Array jobs let you submit many individual “tasks” within one job submission
- Benefits:
    - Only one qsub required
    - Only one jobID or name to monitor in qstat
    - Significantly reduces load on SGE qmaster

Array Jobs: Qsub syntax

- This is a 10 element task submission

```
qsub -t 1-10:1 -N arrayJob \ !
./my-arrayJobScript.sh
```

chris@bioteam.net

Array Jobs: Qsub syntax

- The “-t” switch:
  - -t [FirstTask] - [LastTask]:StepSize
- The “-tc” switch:
  - -tc [Concurrent Task]
- Examples
  - qsub -t 1-100:1 ./my-array-job.sh
  - qsub -t 1-100:2 -tc 10 ./my-array-job.sh

chris@bioteam.net

Array Jobs: How they work

- The secret is simple
- For each task in the array, SGE will populate a special environment variable
  - $SGE_TASK_ID
- Running tasks can query this variable to learn what position they are
- Often use to build paths to input or output files

chris@bioteam.net

Created by The BioTeam, http://blog.bioteam.net

Array Jobs: Final

- For advance cases:
- Recent SGE enhancement allows for job dependency conditions among individual array job task elements

Troubleshooting

chris@bioteam.net

Job Level Troubleshooting

- Job dies instantly
  - Check the .o and .e files in the job directory
  - Check .po and .pe files for parallel jobs
  - Best resource, usually clear error messages found:
    - Permission problem, no license available, path problem, syntax error in app, etc.

chris@bioteam.net

Created by The BioTeam, http://blog.bioteam.net

Job Level Troubleshooting

- Job pending forever
    - First Pass:
    - `qstat -j <job_id>`
    - This will tell you why the job is pending and if there are any reasons why queues cannot accept the job
    - Possible root causes
        - Impossible resource requested, license not available
        - Scheduling oddness

Job Level Troubleshooting

- Job runs from command line on front end node, but not under Grid Engine
- Most common root cause:
  - Difference in environment variables
  - Difference in shell execution environment

General Troubleshooting

- Many times the problems are not SGE related
    - Permission, path or ENV problems
- Best thing to do is watch STDERR and STDOUT
    - Use the qsub ‘-e’ and ‘-o’ switches to send output to a file that you can read

General Troubleshooting (cont.)

- Checking exit status and seeing if jobs ran to completion without error
  - Use: ‘qacct -j <job_id>’ to query the accounting data
  - Will also tell you if the job had to be requeued

chris@bioteam.net

Created by The BioTeam, http://blog.bioteam.net