

Maximizing efficiency of parameter parallel workloads on a condo-style cluster

Hyak Overview

Strong condo model

- > Node owners' jobs run on nodes they purchased
- > Users can utilize any node via a preemptable queue
- > Preemptable jobs
 - Requeued immediately by node owner's job
 - Priority determined by node count and usage
 - Requeued every four hours
 - Must utilize all cores in assigned node(s)

Hyak Overview

Technologies employed

> Focus of this talk

- GNU parallel
- parallel-sql

> Others

- Moab allprocs flag
- BLCR: Berkeley Lab Checkpoint/Restart

GNU parallel

Overview

- > Takes a list of command lines and keeps N tasks running until all the tasks in the list have run
- > Multi-node capability starts tasks on remote computers
- > Many other capabilities for parameter parallel workloads we have not utilized

GNU parallel

Shortcoming for our site

- > Multi-node capability ties unrelated tasks together
 - The more nodes a preemptee uses, the more likely it is to be preempted
 - More lost task progress when a job is preempted
- > Naive users
 - Too few tasks for nodes requested
 - Tasks with widely varying runtimes

parallel-sql

Overview

- > Users load tasks into a PostgreSQL database using provided tools
- > parallel-sql retrieves tasks from database until all tasks in the database are marked complete
- > Multiple parallel-sql instances retrieve unique tasks
- > Optionally resets tasks' status to available when parallel-sql is killed with -TERM

Questions / Talk to Me

Contact info

> Stephen Fralich

- sjf4@uw.edu
- <https://github.com/stephen-fralich/>