Prototype Features for PBS Professional at NAS

May 23, 2017

Greg Matthews – PBS Administrator
NASA Advanced Supercomputing (NAS) Division
NASA Ames Research Center, Moffett Field, CA
gregory.matthews@nasa.gov

A day in the life of PBS at NAS

- PBS at scale
  - Single PBS complex comprising:
    - Pleiades - 11,322 node InfiniBand cluster
    - Electra – 1,152 node InfiniBand cluster
  - Each node runs pbs_mom
    - sharing = default_excl
  - Job mix
    - 300-1,200 running jobs
    - 100-1,000 queued user jobs
    - 0-500 queued 5-node “system” jobs

- Additional smaller clusters
  - Merope
    - 1,792 node InfiniBand cluster
  - Endeavour
    - Two SGI UV 2000 machines

Suspend jobs for system maintenance

- Search community.pbspro.org for PP-389
- Status: in production at NAS

- NAS has used job suspend via qsig for 3+ years
  - Suspend all user jobs in cluster
  - Eliminate job drain for certain classes of maintenance
    - Filesystem upgrade
    - Remove racks
    - Interconnect-at-risk activities
- Standard suspend is geared toward freeing resources

- qsig –s admin-suspend job_id
  - Sets node state “maintenance” for all nodes in job
- qsig –s admin-resume job_id
  - pbs_server resumes job, no need for scheduler

Node ramp down

- PP-339 and PP-647
- Status: in production at NAS

- pbs_release_nodes - shrink a job on-the-fly by releasing sister nodes
- Immediate uses for NAS
  - Efficient worker-task jobs
  - Skip queue wait for follow-on post-processing jobs

- Difficulties overcome
  - $PBS_NODEFILE
  - Released node(s) and pbs_server handshake

- Release at job stageout
- Job accounting

Reliable job startup

- Status: Altair's initial prototype tested at NAS

- Mechanism to express admin's confidence in ability of nodes to join a job
- Admin uses a queuejob hook to provide an increased select statement
  - E.g. user requests select=**10**:ncpus=8, hook adjusts to select=**12**:ncpus=8
  - Hook logic can be as complex as necessary, e.g. only provide extra nodes to jobs requesting more than 256 nodes

- Scheduler satisfies the adjusted select statement

- Head node of job is responsible for finding what the user really requested
  - Eligible nodes are those that pass all pre-job hooks/etc.
    - Excess nodes beyond the user's request are released from the job
  - Problematic nodes can be offlined before being released from the job

Reliable job startup, continued

- The window of opportunity for a job to start is a precious thing
  - Failure to start may give time for a higher priority job to be submitted
  - Our users may wait 1+ days for the window, real bummer if some transient issue on a single node causes problems

- We've done enough babysitting of large (1,000+ node) jobs to know that having a few spares available is good insurance

- System problems can be bursty
  - Suppose a new Lustre client build has a few bugs that only present themselves at scale
  - Perhaps a bad batch of DIMMs has found its way into your nodes

Reliable user workflows

- Status: NAS kicking around ideas

- We'd like to push a bit further on this notion of "window of opportunity"
  - Did I mention they're precious?
  - Expand notion of reliable job startup

- At times we assist users with partly defined sets of jobs that are best run "together"
  - E.g. scaling study – user adjusts parameters on the fly as results are obtained
  - Desire to avoid repeated draining for large jobs
  - A single large interactive job could work, but it's vulnerable to any single node in the job dropping out
    - Maybe introduce a new layer of job fault tolerance?

- Feels like a use for reservations, but…
  - Reservations are a priority cheat code
  - Maybe reservations could be more job-like?

Reliable user workflows, continued

- Maybe a job dependency-like construct?
  - Job A gets resources, dependent job B inherits those resources

- We could get really fancy…
  - Job-like reservation
  - PBS spins up a new scheduler just for the reservation
  - Define some number of policy constraints
    - Nodes get released if not used for X amount of time
    - Entire reservation is deleted if not used for Y amount of time

Front-end nodes via PBS

- Status: NAS working on initial prototype

- Like most sites we have a set of front-end/login nodes where jobs are launched and various non-job activities take place
- Resource management of front-ends is not unlike PBS management of compute nodes
  - Some jobs/users can share nodes, some can't
  - Pool of front-ends may be oversubscribed, would be helpful to dynamically grow the pool and/or schedule the load

- What if we treated some number of compute nodes like front-ends?
  - At NAS we expect jobs to run days, but front-end activity for a user could be weeks/months
  - Use reservation to set aside node(s)
    - Avoids vulnerabilities of jobs
  - Disable pbs_mom's $restrict_user for authorized users of reservation
  - Users could ssh directly to node(s), use screen/VNC/etc.
  - Users could submit jobs to the node(s)

Scheduler node buckets

- Status: Altair's initial prototype tested at NAS

- Pleiades/Electra have 12,474 nodes, but scheduler can lump them into about 50 different buckets
  - Architecture
  - Ncpus
  - Memory
  - Queue assignment
  - Site-specific resources
- Prototype scheduler using buckets is fast*
  - 10-15x faster overall – 90 second cycle is reduced to less than 10 seconds!
  - Specific speedup is roughly 130x (ignoring communication between pbs_server and pbs_sched)
- * Caveats:
  - Does not apply to node sharing, reservation jobs, or jobs requesting specific nodes
  - Prototype skips some details, primary goal was proof of concept

qsub node filtering

- PP-507
- Status: Altair working on design

- We've sped up the scheduler – time to slow it down again

- Altair is considering several possibilities for qsub to offer more flexibility in the way resources are requested
  - PP-507 introduces conditional operators for node resources
  - E.g.
    - qsub –l select=1:node_filter="mem>=1gb and mem<=3gb"
  - NAS likes the idea of selecting "this, but not that" semantic
  - Possible downside is breakage of site hooks that parse select

- May be other options to get filtering
  - Multi-dimensional place=group= ?

Questions?