ALABAMA SUPERCOMPUTER CENTER MOVES FROM SLURM TO ALTAIR® PBS PROFESSIONAL®

The Alabama Supercomputer Center (ASC), operated by the Alabama Supercomputer Authority, provides high-performance computing (HPC) services to students, faculty, and staff across the state. Unlike its predecessor, which ran Slurm, ASC's newest system, ASA-X, uses the Altair® PBS Professional® workload manager. A federal government HPC system managed by the same systems integration contractor had recently undergone the same transition — and the team at ASC knew the change was not to be undertaken lightly.

The Decision to Switch Workload Managers

The transition from SchedMD's open-source Slurm to PBS Professional started when ASC's HPC system administrators became frustrated enough with Slurm that they were ready to switch workload managers. An upcoming major funding cycle was the ideal time. Most of the HPC staff had never used any other workload manager, but HPC group leader Dr. David Young had decades of experience using NQS, LoadLeveler, PBS Professional, TORQUE/Moab, and Slurm.

ASC's ASA-X supercomputer supports students, faculty, and staff across Alabama

The ASA-X system uses PBS Professional for efficient queueing and workload management

The team's concerns with Slurm were multifaceted. As is typical with supported open-source software products, Slurm documentation was sparse and often hard to find; some that had been available online when Slurm was new had since been removed. "The expansion of SchedMD came with some growing pains," Young said, "including inadequate training of technical support personnel and inadequate testing of new versions prior to release." Due to ASC's mandate to maximize hardware utilization, they were using resource-pool scheduling with QoS, a Slurm configuration that was supported but less common than partition-based scheduling. In one instance, a Slurm version upgrade had to be rolled back after stress testing on the system, extending a maintenance shutdown. It was deemed unwise to remove nodes or partitions from Slurm except during maintenance shutdowns, which are infrequent at ASC. "To its credit," Young said, "the Slurm scheduler has some nice features, including built-in prevention of starving jobs and a good preemption system" — but that wasn't enough to satisfy the team's needs.

To avoid a hasty decision, the team documented detailed requirements, surveyed popular options, and discussed features with solutions providers at the annual Supercomputing (SC) conference. PBS Professional emerged as the front-runner, and they tested it on-site. As a mature, commercial product, PBS Professional offers a significant amount of documentation and professional technical support. In addition, Altair proactively adds new functionality to accommodate changing industry needs like new brands of GPU chips, Arm processors, artificial intelligence (AI), and cloud computing. ASC's policy is to pay for workload management support; since academic pricing from Altair and SchedMD was the same, price wasn't a factor in its decision.

Several of PBS Professional's features were important to consider for ASC when deciding to switch workload managers:

- PBS Professional is a mature product from a stable company that features established development procedures and reliable support.
- PBS Professional is a commercial product with professional support and testing.
- The PBS hooks system offers a variety of customization options.
- Altair® Access™ works with PBS Professional to optimize scheduling. The Access AI assistant predicts job requirements for metrics like runtime and memory usage, leading to better system utilization.
- Documentation is extensive, with thousands of pages to answer any question.
- Configuration changes through qmgr can be made in a way that prevents job loss and node downtime.
- Log files are straightforward to read and easily distinguishable, based on function and location.
- Vnodes allow hardware to be split into target groups.

As a mature, commercial product, PBS Professional offers a significant amount of documentation and professional technical support

Workload Manager Scheduling Strategies

Before wading into the details of workload manager implementation, there are some big-picture scheduling strategies to consider. Think of these as the design patterns of workload manager implementation.

First In, First Out (FIFO): A simple queue scheduling algorithm where jobs are put into a run state in exactly the order they were submitted. Most installations use more complex scheduling algorithms. FIFO only makes sense for very simple installations, such as a workstation only used by one person or a group cluster only used by a few people. Sometimes FIFO isn't the right choice even in these basic situations.

Partition-Based Scheduling: On some clusters, X cores are available to the small queue, Y cores are available to the medium queue, and so on. Some organizations that use this method have weekly meetings to debate whose research project is most urgent, then reassign the number of cores available to the queues belonging to each project. The downside of partition-based scheduling is that some partitions may have a big backlog of work while others have idle cores.

Resource Pool Scheduling: The installation described here considers the processor cores as one big pool of available resources — meaning any queue can utilize any core (with some necessary exceptions). This is done to maximize cluster utilization, and thus maximize the value of money spent on HPC hardware. ASC uses resource pool scheduling to maximize utilization; they also buy nodes that are somewhat heavy on memory, so jobs are more likely to be waiting for cores to become available than waiting for memory to become available.

Reservation Scheduling: Some facilities have very high-priority work and very low-priority work. They use workload manager reservations to ensure high-priority work always has the resources it needs; the rest of the work can utilize the remaining resources. There are several ways to implement this using reservations, partitions, or preemption. Weather modeling services, for example, may do this to process high-priority weather data so weather maps will always be available when needed. Some workload managers support advanced per-job reservations.

Cycle Scavenging: A specialized type of queueing used by the HTCondor system. A cycle scavenger runs work on desktop computers when their screen savers kick in at night. This can be an efficient use of hardware if jobs are small-memory and easily restarted, but it's insufficient for big jobs that aren't restartable. Folding@home is also a cycle scavenger.

Gang Scheduling: Takes advantage of hyperthreading and multitasking to oversubscribe the physical processor cores. Gang scheduling can be a good idea when running a large number of very small jobs, but it's a bad idea for running large jobs and can make usage reporting a nightmare. Some facilities also oversubscribe memory, which saves on hardware cost but may incur severe performance penalties.

PBS Professional Implementation Decisions

To fairshare or not to fairshare — that is the question. When installing PBS Professional, the ASC team needed to make a big up-front decision between two ways of setting up scheduling.

ASC elected to to use fairshare in PBS Professional to share resources among groups and users

Option 1: Use priorities, backfill, job sort formula, and limits on how many jobs, cores, memory, and GPUs a single user can have in a run state.
Option 2: Use fairshare, preemption, and priorities.

ASC has run fairshare for years, so they chose to continue using it with PBS Professional. Fairshare algorithms and the parameters to configure them are considerably different from one workload manager to another.

Pros and cons of scheduling options:

- Backfill is designed to fully utilize the system, but in practice only mostly utilizes it.
- Backfill can go wrong, creating starving jobs that never run. This can be mitigated by using a job sort formula that incorporates the amount of time a job has spent waiting for resources.
- Fairshare requires less manual adjustment once it's set up well.
- Years ago, ASC used PBS Professional with backfill, then went to TORQUE/Moab with fairshare, then to Slurm with fairshare and preemption. Each step was seen as an improvement at the time.
- PBS Professional fairshare is different from Slurm fairshare. Slurm sets everything as a priority and features less hard restrictions and requirements. PBS Professional is more algorithmic, meaning it's "fairer." It also means there are more restrictions on what features do or don't work with fairshare in PBS Professional.
- The biggest concern about the PBS Professional fairshare algorithm was how well it would maximize hardware utilization. In PBS Professional, fairshare doesn't work with backfill. After implementation, the ASC team found that PBS Professional fairshare utilized the hardware as fully as is practical.

In many large organizations, reporting is an important part of HPC system management. ASC does monthly, quarterly, and annual reports that break down utilization by dedicated hours, GPU hours, and TB hours for users, academic departments, and universities. In PBS Professional, one solution for this is to have bash scripts that extract information from qstat in json format. To do that, PBS Professional is configured to maintain qstat information for six weeks. PBS Professional also offers complete, formatted job accounting logs that record all the resources a job uses.

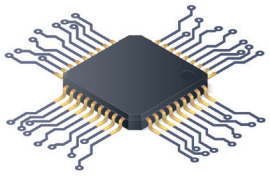
Other reporting functions don't use PBS Professional. Reporting software utilization is done by extracting information from the pacct files on all nodes. The ASC staff also instrumented the LMOD environment module system to log module checkouts. An expert system using AI technology runs monthly to notify HPC software analysts when an old software version is no longer being used. Monthly reports of disk and partition high-utilization marks are generated via bash scripts on crontabs. Altair also offers Altair® InsightPro™, a reporting solution for comprehensive visibility into HPC clusters.

PBS hooks are an excellent tool for adding site-side customization. Since each hook is connected to a specific stage of the job life cycle, users can perform specific checks for jobs, queues, nodes, and more to meet specific criteria and perform custom actions based on those criteria. ASC uses a cgroups hook to maintain resource usage across the cluster, and they create and import hooks as new needs arise.

Maximum job length can be limited based on CPU time or wall time. Over the years, ASC has switched from CPU-time limits to wall-time limits. Many student users may write MPI code with deadlocks or submit jobs with poorly chosen parallelization options. Wall time limits ensure these bad jobs don't get stuck in an endless running state.

For monitoring, a custom exporter written in Python is used to export PBS Professional metrics for tracking and visualizing in Grafana. This exporter can be updated when new metrics are needed from PBS Professional for visualization.

Configuring PBS Professional for GPU device detection and isolation required cgroups to be enabled, then vnode per NUMA node to be set within the cgroup configuration. Once the GPUs were moved into separate vnodes, the ASC team found that GPU jobs ran as expected.



The ASC team enabled cgroups so they could use GPU nodes with PBS Professional

For user convenience, a workload-manager-aware MPI launcher is recommended. In Slurm this is srun. In PBS Professional it is mpiexec. This means recompiling OpenMPI with flags for the queue system being used. Fortunately, almost all MPI software is dynamically linked, so the environment modules can be changed to load a PBS-aware version of OpenMPI without requiring the software itself to be recompiled.

Both Slurm sbatch and PBS Professional qsub have a rerunnable flag, but the two work differently. In Slurm, all jobs are submitted as rerunnable jobs to support preemption. In PBS Professional, jobs are submitted as not rerunnable to avoid an infinite restart-fail loop.

The ASC team didn't do anything special on the PBS Professional side for integration with Warewulf and InfiniBand. However, there was significant work to get InfiniBand and PBS Professional to work with stateless nodes through Warewulf.

ASC sets custom resources and non-consumable resources to help designate where jobs run, helping them get to compatible hardware. These include gpuname, gpuarch, amd/intel, arm/x86-64, avx2, avx512, milan/genoa, mem, ncpus, and node number (host is built in).

One issue that has plagued HPC workload management systems is the possibility of runaway or rogue jobs. These are jobs that still have processes running on a node even through the workload manager thinks the job has ended. The use of cgroups has almost always ensured that this is no longer an issue. Likewise, cgroups mitigates the risk that an unknowing user will submit a job that tries to use more cores than they requested from the workload manager. For example, an OpenMP parallelized job that doesn't set OMP_NUM_THREADS tries to use all processor cores on the node by default. This can still happen, but cgroups ensures the threads are all confined to the physical cores assigned to the job.

ASC needed to work out issues with jobs stuck in H and E states. The team found that the trouble was partially due to how their queues were configured. For jobs stuck in E state, they ran 'qdel -Wforce $JOBID' to ensure the job was forcefully removed from the queue.

In Slurm, exclusive jobs (only one job on a node) are set with the job flag. In PBS Professional, the ASC team set up a benchmark queue that was the only queue that could submit jobs to a designated node and was configured to run only one job at a time. This is important when doing benchmarking work and not wanting multiple jobs sharing the node. This can also be accomplished by using qsub -lplace=scattered:excl.

A job submission defaults to one chunk if not specified (set at the server level; if unset, it defaults to one and can't be zero). If a user wants to split cores among two nodes but doesn't want to specify the number of chunks, there's no way for PBS Professional to know how to place the cores, and it will default to "free" placement, meaning "try wherever possible." That can be changed.

Ease-of-Use Innovations at ASC

ASC uses a number of status scripts for user convenience. These are small, one-line scripts that allow users to type qstat2, qstat3, etc. instead of remembering qstat flags. They include:

- qstat -f
- qstat -f-p
- qstat -n -w
- qstat -s
- qstat -Q
- pbsnodes -a -S
- pbsnodes -a

One script loops over jobs to make a list, by user, of aggregate cores and memory in a run state, and aggregate cores and memory for pending jobs. If a user has pending jobs but no running jobs, it prints a warning message.

ASC uses queue wrapper scripts that generate the qsub command for users. This ensures flags recommended by the HPC staff are being utilized. It lets those flags be updated with new workload manager versions, without a massive user training effort. This has been very popular with users. Here's an example of submitting a job; items typed by the user are in bold.

Because workload management tools have complex configurations, thorough testing is an important part of installation

```
> run_script ls_test

This runs a script in the current directory via the workload manager.
The script enables the job to use multiple processors on the same node.

Choose a batch job queue:

Queue                      Wall Time      Mem   # Cores
-------------------------- ---------- -------- --------
express                      4:00:00     16gb      1-4
small                       60:00:00      4gb      1-8
medium                     150:00:00     16gb     1-16
large                      360:00:00    120gb    1-128
bigmem                     360:00:00  130-500gb   1-32
benchmark                   24:00:00    500gb    1-128
class                       40:00:00     64gb     1-60
special                   1008:00:00   2000gb     1-64
sysadmin                   168:00:00      4tb   1-1000

Your job will have a shorter wait time if your memory request is
reasonable (about 20% more than needed), and your time request is
reasonable (about 50% more than needed).
Find this out by running 'jobinfo -j JOB_NUMBER' for a correctly
completed job.

Enter Queue Name (default <cr>: small) medium

Enter number of processor cores (default <cr>: 1 ) 2

Enter Time Limit (default <cr>: 150:00:00 HH:MM:SS) <Return>

Enter memory limit (default <cr>: 2gb ) 1gb

Enter a name for your job (default: lstestSCRIPT)
test7
Run on;  icelake, milan (default: asax)
<Return>

================================================================
=====          Summary of your script job               =====
================================================================
  The script file is: ls_test
  The time limit is 150:00:00 HH:MM:SS.
  The memory limit is: 1gb
  The job will start running after: 202404120858.41
  Job Name: test7
  Virtual queue: -q medium
  Queue: -q medium
  Constraints:
```



```
WARNING: Most programs should use mpiexec in place of srun.
         Please check the https://hpcdocs.asc.edu/ documentation.
  Queue submit command:
qsub -q medium -j oe -N test7 -a 202404120858.41 -r n -M dyoung@asc.edu -m ae -l
walltime=150:00:00 -l select=ncpus=2:mpiprocs=2:mem=1000mb

47757.asax-pbs1
```

Workload management is critical for HPC system users, and for system managers to encode their utilization policies

ASC uses a locally created defaults file. If items in this file are changed from PROMPT to a value, that item is not prompted by the script above. Here's an excerpt from that file.

```
#  queue_name_pref sets a name for the queue to use
#  if PROMPT then you will be asked to type a queue name
#  Otherwise, it can be set to a valid queue name
queue_name_pref=PROMPT
#queue_name_pref=express
#queue_name_pref=small
#queue_name_pref=medium
#queue_name_pref=large

#  memory_pref sets the amount of memory the job gets.
#  Setting 8gb to a 4 processor job, will give each processor 2gb.
#  if PROMPT you will be asked to enter a memory amount
#  if DEFAULT a default for the queue an number of processors will be used
#  Otherwise a number can be given with units like 200mg or 4gb (no spaces)
memory_pref=PROMPT
#memory_pref=DEFAULT
#memory_pref=1gb
```

Another popular customization is a script called jobinfo, as illustrated below. The "elapsed time" output is used to compute time complexity. The "max memory used" field is used to compute memory complexity. Users are told to keep memory efficiency above 50% and parallel efficiency above 75%. The jobinfo script takes data from qstat in json format, then processes and displays it.

```
ssh                                                    ⌥⌘2
asndcy@asaxlogin1:tests> jobinfo -j 62

#################################################################
# Your username for this job is:            asndcy@asaxlogin2.asc.edu
# Your group for this job is:               analyst
# Your job ID is:                           62.asax-pbs1
# Your job name is:                         watercomG16
# Your job queue is:                        small
# Your job ran on nodes:                    asax002/0*2
# Your number of processors used:           2
# Your job work directory was:              /home/asndcy/calc/gaussian/tests
# Your job was submitted at:                Tue Jan 9 16:42:18 2024
# Your job started at:                      Tue Jan 9 16:42:18 2024
# Your job ended at:                        Tue Jan 9 16:44:05 2024
# Your job elapsed time is:                 00:01:47
# Your job dedicated time is:               00:03:34
# Your job cpu time is:                     00:00:02
# Your job was last modified:               Tue Jan 9 16:44:06 2024
# Your requested wall time is:              60:00:00
# Your job cpu parallel efficiency is:      0.93%
# Your requested memory is:                 2000mb
# Your max memory used:                     448620kb
# Your job memory efficiency is:            22.43%
# Your job state is:                        F
# Your job exit code is:                    0
# Your requested resource was:              ncpus=2:mpiprocs=2:mem=2000mb
# Your job commercial value is:             $ 0.00475555
#################################################################

asndcy@asaxlogin1:tests>
```

Summary

Several months after the new HPC system went into full production, ASC's HPC team reviewed the transition from Slurm to PBS Professional. A number of users had reported a smooth change, including one of the least technically savvy research groups, led by a 60-year-old veterinarian. System administrators liked the qmgr interface for PBS Professional administration and its ability to make changes while jobs are running on the system. Altair's support organization consistently delivered fast turnaround times when the team needed help. PBS Professional provides a rich set of features for limiting various types of usage, either globally or per queue, and the ASC team used that functionality more frequently than they'd originally anticipated. Overall, switching from the less mature, open-source Slurm to the more mature, heavily tested, and well-supported PBS Professional has been a win for ASC's HPC system management. The transition was a win for their HPC users with minimal disruption to their work, enabled by the queue wrapper scripts described above and by PBS Professional's reliable, feature-rich capabilities.

The switch from Slurm to PBS Professional has been a win for ASC system admins and users

Workload management is critical for users to run their jobs efficiently, and for HPC system managers to encode their organization's utilization policies. Switching workload managers is not a trivial undertaking, but the ASC team found that the long-term benefits of switching from Slurm to PBS Professional were worth the effort. PBS Professional comes with extensive documentation and expert Altair support, plus developers who are always working to add features and optimize performance for customers like ASC — ultimately delivering robust, efficient workload management.