BASIC FAIRSHARE FOR ALTAIR® PBS PROFESSIONAL®

Joe Miller III – Technical Support Engineer, Altair / June 18, 2020

Introduction to PBS Professional

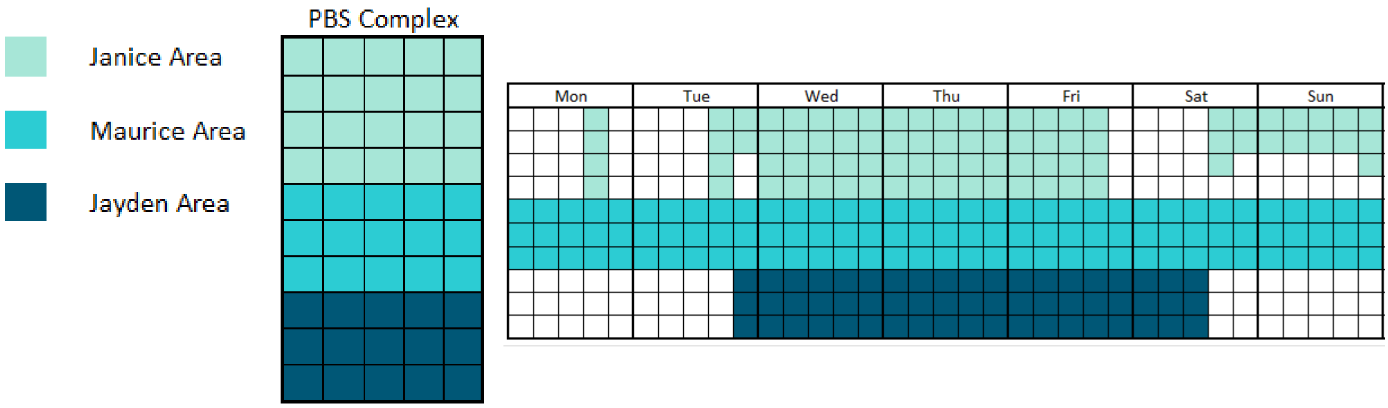
Altair PBS Professional is a fast, powerful workload manager designed to improve productivity, optimize utilization and efficiency, and simplify administration for HPC clusters, clouds, and supercomputers. It automates job scheduling, management, monitoring, and reporting, and it's the trusted solution for complex Top500 systems and smaller clusters.

Challenges

By default, PBS Professional does not place restrictions on entities' (users, groups, projects, etc.) use of cluster resources. All system users can run jobs and use the entire cluster. PBS Professional is equipped with many parameters and attributes to configure the resource usage on your cluster.

For example, your site's policy may have contracted server availability to different entities. You can set up complicated rules, hooks, and resources for your users, jobs, and queues to gain desired outcomes, but that requires configuration and tracking of many parameters and attributes.

By using parameters and attributes you can effectively create a walled-off space for each user, safe from others.



While this ensures that each user gets access to the system, it may lead to inefficient use of overall cluster resources. Sharing is needed to make efficient use of available resources; users will not likely be using their space 100% of the time. Users' jobs may also require a larger portion of resources than can be allocated through "walled-off" space or time.

Simple Case Study

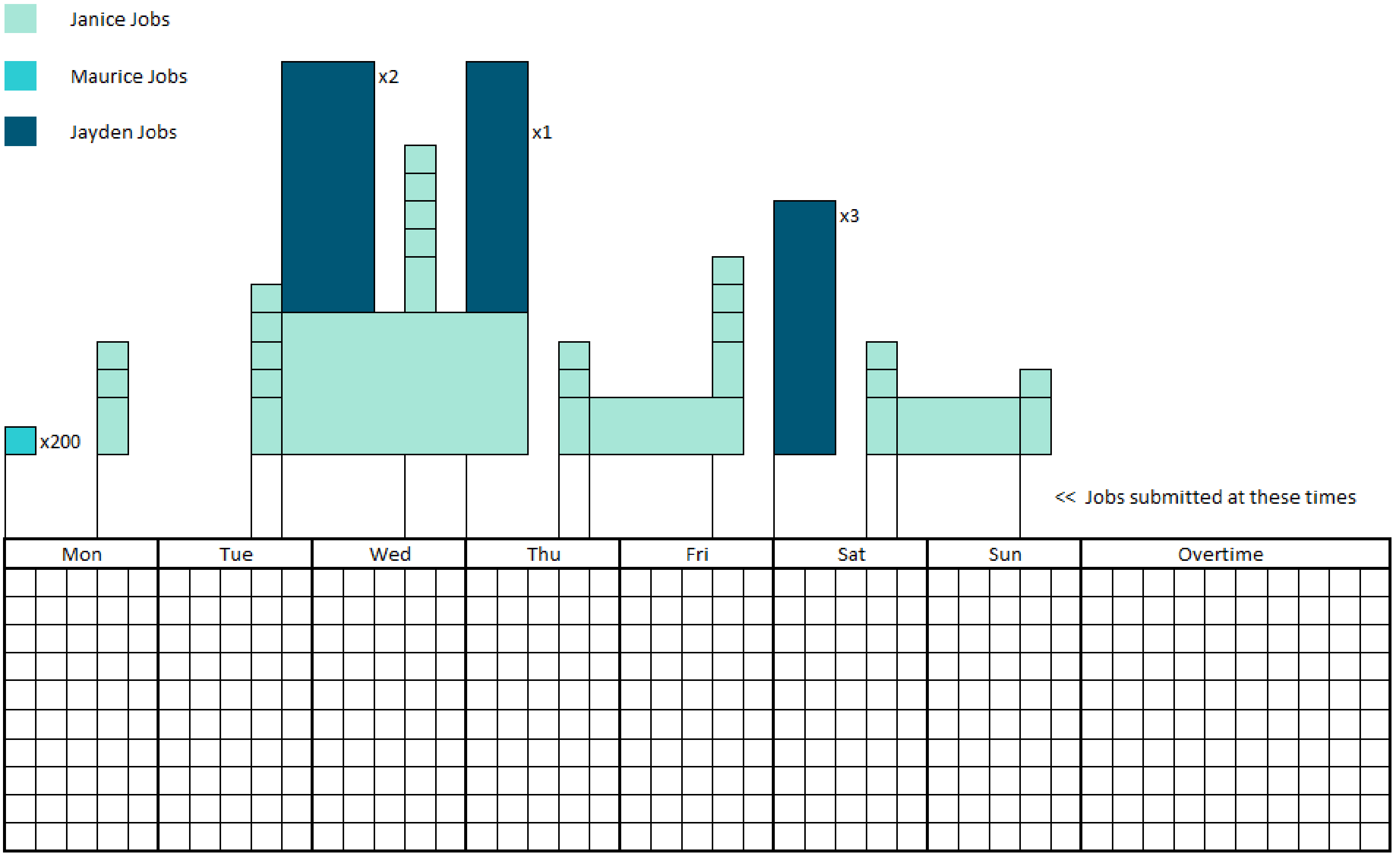
Three users are contracted to use a cluster.

Janice – 25%	Maurice – 50%	Jayden – 25%
Long-term side jobs and daily end-of-day jobs 20+% of cluster	Batch 10,000 small jobs on Monday to run through next week; highly possible over/under	Large jobs taking 90% of cluster; many submitted at once; sporadic

It would be difficult to use FIFO, LIFO, limits, priority, or time-based scheduling to satisfy all three of these users on your cluster.

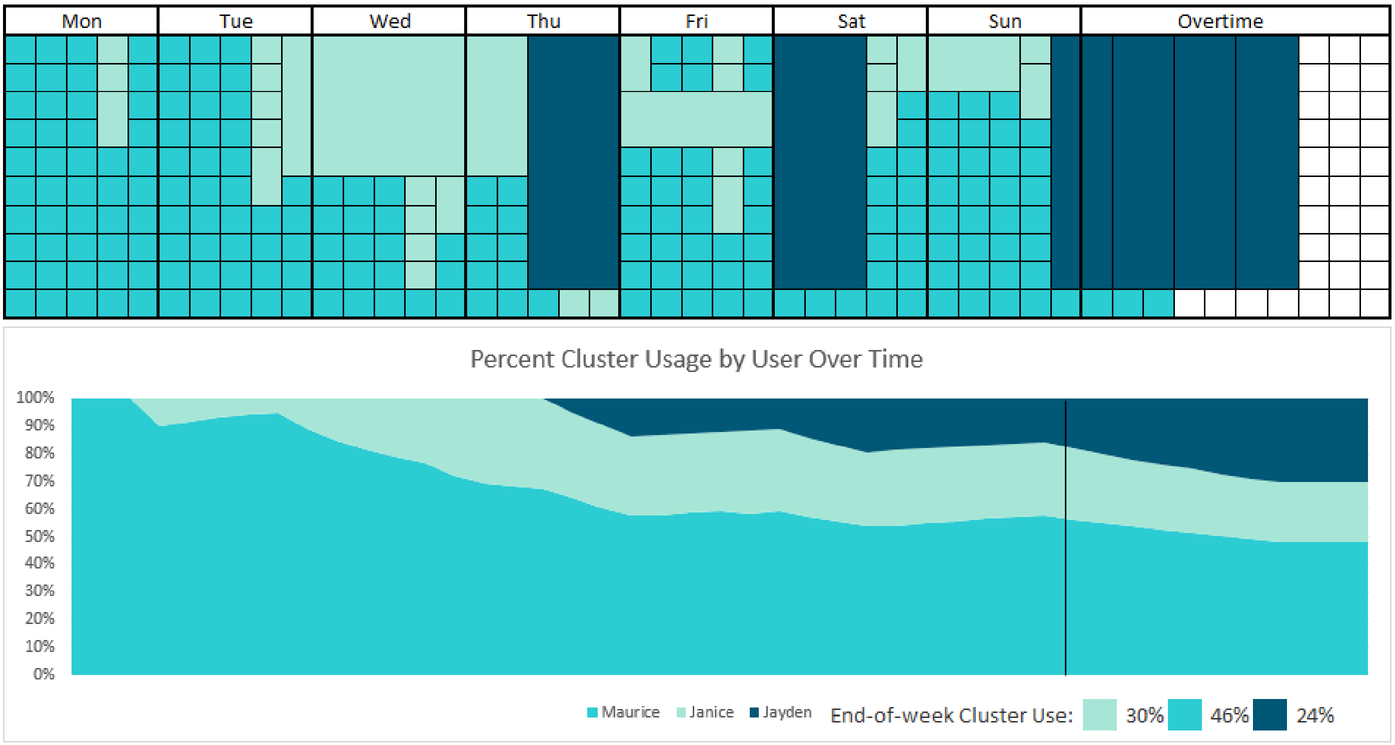
- The order in which jobs are submitted will leave Janice without daily reports.
- Limiting the number of jobs running doesn't allow running at full capacity during off-peak hours.
- Prioritizing Janice will possibly move Jayden into the far future.

- It’s hard to schedule dedicated time without disruption.



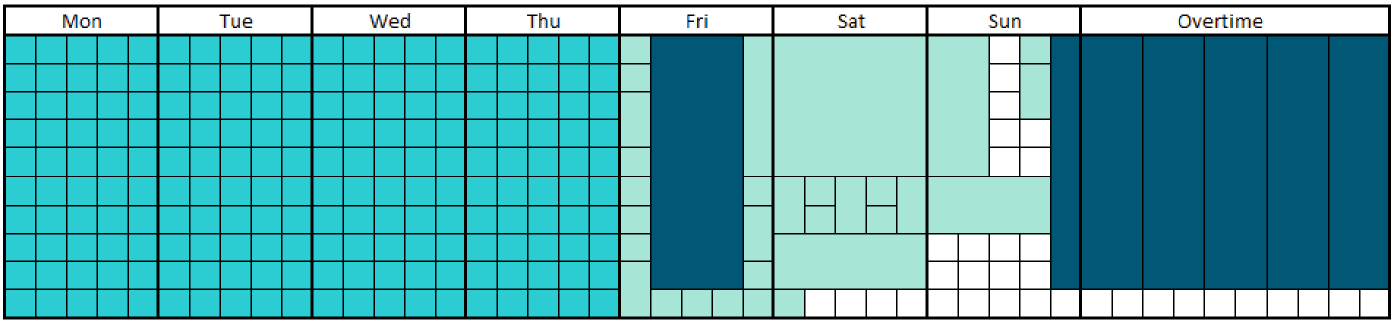
Below are two examples of scheduler tools used to attempt fairness on the cluster. These do not consider continued submission of jobs during overtime (into the next week). The end-of-week (Sunday, midnight) cluster use is labeled for comparison.

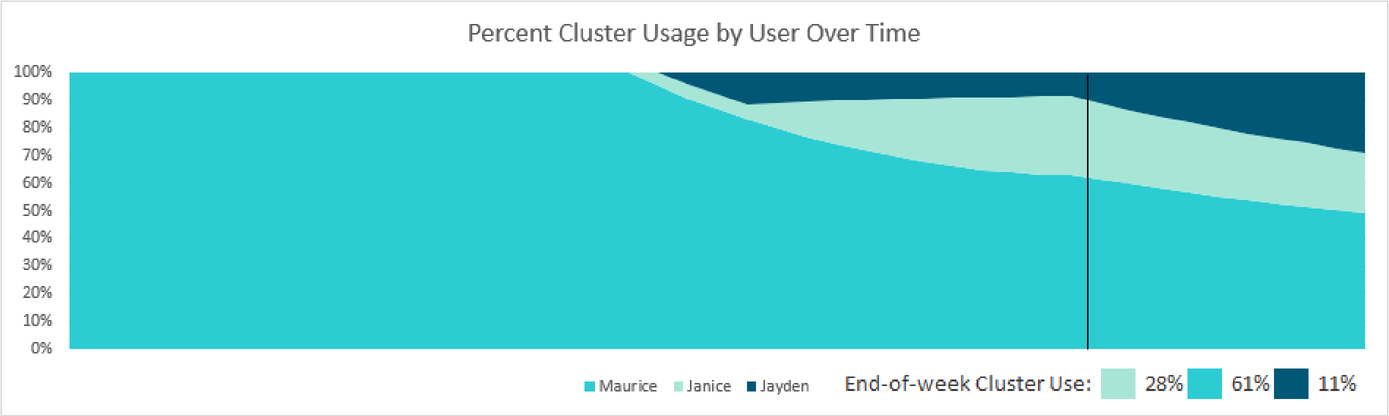
Example 1: Priority: Janice > Jayden > Maurice



Jayden’s jobs are moved around and waiting in queue until later in the week. Maurice’s and Jayden’s shares of the cluster are entirely based on how much Janice doesn’t use.

Example 2: Job Submission Order + Availability

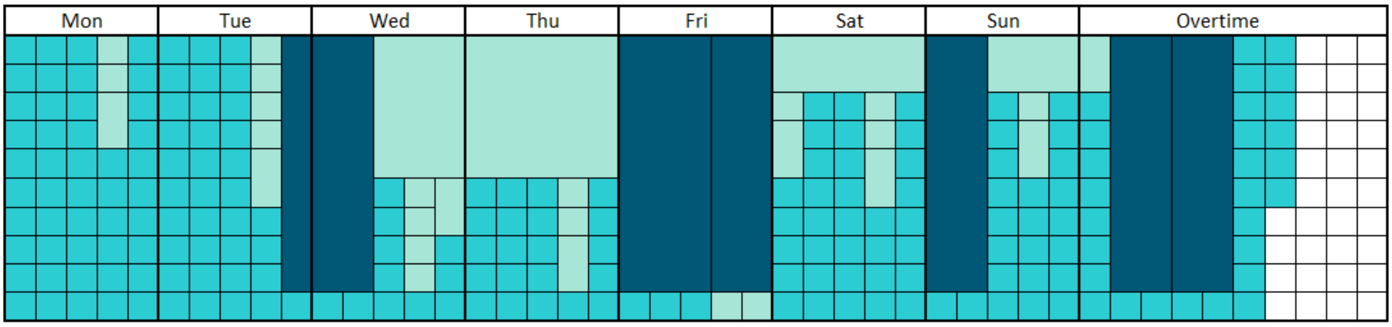


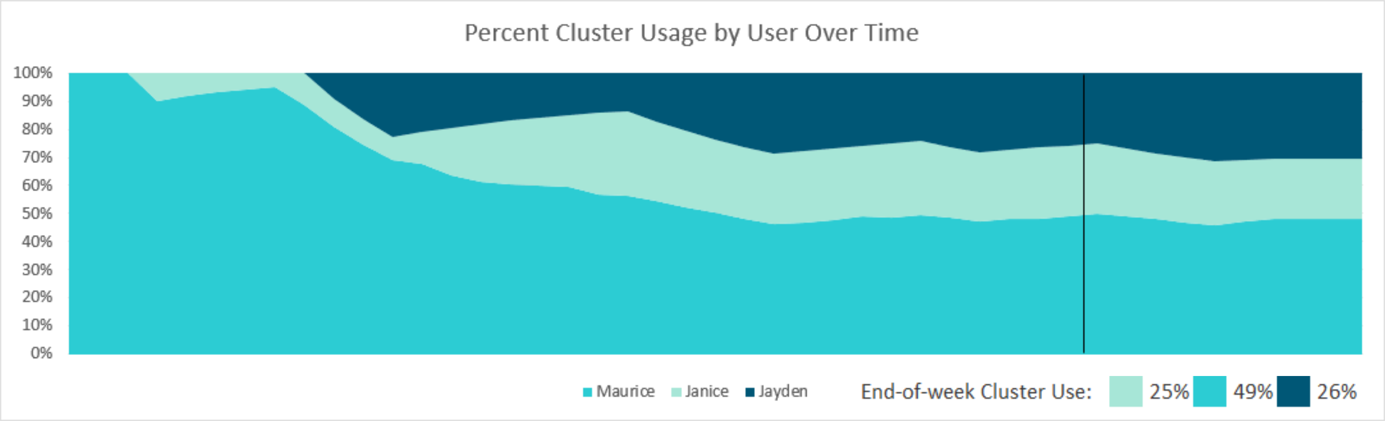


Wasted resources, extended completion time, and both Janice’s and Jayden’s jobs are pushed back; least fair.

Example 3: Desired Scheduling Outcome

Below is a representation of how scheduling against resources in a cluster would occur in the fairest way possible. The PBS Professional scheduler attempts to balance requested and used resources to match expectations of fairness as quickly as possible.





Everyone receives resources based on what they “deserve” or what is “fair,” the most balanced approach in terms of running times and cluster load.

Introduction to Fairshare

Fairshare is a scheduling tool used to share a cluster’s limited resources according to all entities’ history of usage and a weighted, rooted tree structure. Entity priority depends on usage history and the fairshare tree. Fairshare is the most direct option to grant a percentage of the cluster to an entity while sharing access to resources.

Enabling and Configuring Basic Fairshare

Basic fairshare will grant all system users equal shares of the cluster. There are four steps to enabling basic fairshare:

Steps	Commands
Stop Scheduling	`# qmgr -c “set sched scheduling = False”`

Set Scheduler Attributes	Edit File: ***$PBS_HOME/sched_priv/sched_config:***
	`fair_share: true      ALL`
	`fairshare_enforce_no_shares: FALSE`

Send HANGUP Signal to Scheduler	`# kill -HUP $(pgrep -f pbs_sched)`
Start Scheduling	`# qmgr -c “set sched scheduling = True”`

The `pbsfs` command will show updates after usage (CPU time) is accrued and recorded.

Example output of `pbsfs` command

```
[root@minimal ~]# pbsfs

Fairshare usage units are in: cput

TREEROOT   : Grp: -1      cgrp: 0    Shares: -1     Usage: 8      Perc: 100.000%

unknown    : Grp: 0       cgrp: 1    Shares: 0      Usage: 8      Perc:   0.000%

joe        : Grp: 1       cgrp: -1   Shares: 1      Usage: 7      Perc:   0.000%
```

You will need to set up the fairshare tree to specify what percentage of CPU time (or other resource defined in the sched_config file) to give to each entity. Using our three-person case study, Maurice is allocated 50% of the cluster, and Janice and Jayden each get 25%:

Steps	Commands
Stop Scheduling	`# qmgr -c “set sched scheduling = False”`

Add Entities to Fairshare Tree	Append to File: ***$PBS_HOME/sched_priv/resource_group:***
	`maurice 1 root 50`
	`janice 2 root 25`
	`jayden 3 root 25`

Send HANGUP Signal to Scheduler	`# kill -HUP $(pgrep -f pbs_sched)`
Start Scheduling	`# qmgr -c “set sched scheduling = True”`

Now when Maurice, Janice, and Jayden start using the cluster, they will each be given a weighted amount.

All fairshare data, including current known usage, can be viewed using the `pbsfs` command.

Conclusion

Fairshare is a powerful tool for allocating a set amount of your cluster to entities while sharing available resources. Following the basic setup steps, you should be able to quickly get fairshare working to grant your cluster users fair and equitable access. More advanced configurations and detailed documentation about each attribute, command, and feature can be found in the PBS Professional Administrator’s Guide.

This is a very small example of what fairshare can do. Other examples will be outlined in other white papers and will encompass a much wider range of fairshare's capabilities.