# USING PLACEMENT SETS FOR ALTAIR® PBS PROFESSIONAL®

Lynne Nelson – Senior Application Engineer, Altair / July 14, 2020

## Introduction

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. PBS Professional automates job scheduling, management, monitoring, and reporting, and it's the trusted solution for complex Top500 systems as well as smaller clusters.

The PBS Professional scheduler provides many different methods for optimizing scheduling. Placement sets, also referred to as topology-aware scheduling, can be used to group nodes together based on a shared characteristic to help optimize job performance as well as the overall use of your PBS Professional complex. Common usages include taking advantage of interconnect or hardware topology including IRUs or racks, system resources, access to shared storage, system location, and application availability.

## Challenges

You may have a variety of system resources under the control of a single PBS Professional scheduler and many jobs trying to utilize those resources. Improving application performance and maximizing system performance is a key benefit of using placement sets. A placement set can take advantage of interconnect and hardware topology to minimize hops and message latency for MPI-type applications by scheduling multi-node jobs on nodes that are topologically close to one another.

Using placement sets allows you to run a job on a set of nodes that share a common characteristic, but the user doesn't need to know what the value of that characteristic is. For example, you want all your nodes running a job to be on the same switch, but you don't care if it's switch "A" or "B." Or you want all nodes of a job in a heterogeneous cluster to have the same processor type, but you don't care if that type is "Intel" or "AMD."

The primary advantage of using placement sets is that while users can request specific placement sets, they do not need to know hardware details to obtain optimal job placement.

## Definition of Terms

- **Vnode** – A virtual node, or vnode, is an abstract object representing a set of resources which form a usable part of a machine. This could be an entire host, a nodeboard, or a blade. A single host can be made up of multiple vnodes. Each vnode can be managed and scheduled independently. PBS Professional views hosts as being composed of one or more vnodes.

- **Placement set** – A placement set is a logical grouping of vnodes based on the same value for a resource which is defined by a single value of a multi-valued string resource.

  For example, if you have a “switch” resource with values of S1, S2, and S3, a placement set would contain vnodes with the same “switch” definition. In this case there would be three different placement sets: resources_available.switch=S1, resources_available.switch=S2, and resources_available.switch=S3.

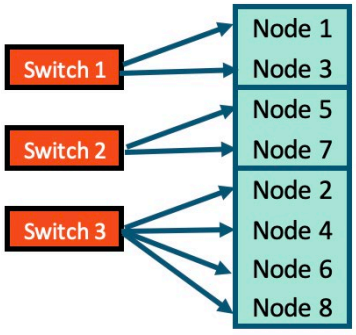


Figure 1

Examples of resources that are frequently grouped together include system resources, interconnect, rack, IRU, application, scratch space, and system location.

  

Note: Resources must be of type string or string_array. Placement sets can overlap; a single vnode may belong to multiple placement sets. If no placement set is defined, there is one placement set consisting of all the nodes in the PBS Professional complex.

- **Placement set series** – placement set series is all placement sets that are defined by one string array resource. In the example (Figure 1 above), "switch" is the string array resource, and it has three defined values. The placement series for the switch resource contains the three placement sets: switch=S1, switch=S2, and switch=S3.

- **Placement pool** – The placement pool is all placement sets that are defined on the system. The server can have a placement pool and each queue can have its own placement pool. If a queue has no placement pool, a scheduler uses the server's placement pool.

  For example, if the server's defined resources are "router,switch," and router can take the values "R1" and "R2" and switch can take the values "S1," "S2," and "S3," then there are five placement sets, in two placement series, in the server's placement pool.

- **Static fit** – A job fits statically into a placement set if the job could fit into the placement set if the set of nodes were empty. This does not mean that it will fit or execute with the currently available resources.

- **Dynamic fit** – A job fits dynamically into a placement set if it will fit with the currently available resources (i.e., the job can fit and run right now).

### Why Use Placement Sets

Following are several instances in which placement sets can aid in efficient scheduling.

### Example 1: Different vnode types

If you have a system with vnodes of different speeds (perhaps some older vnodes and some newer vnodes), it is preferred to run a multi-node job all on the same speed of vnodes. Another example is having different system types from different vendors running on the same PBS Professional complex. In this case, you want to group multi-node jobs by system type rather than spanning system types.

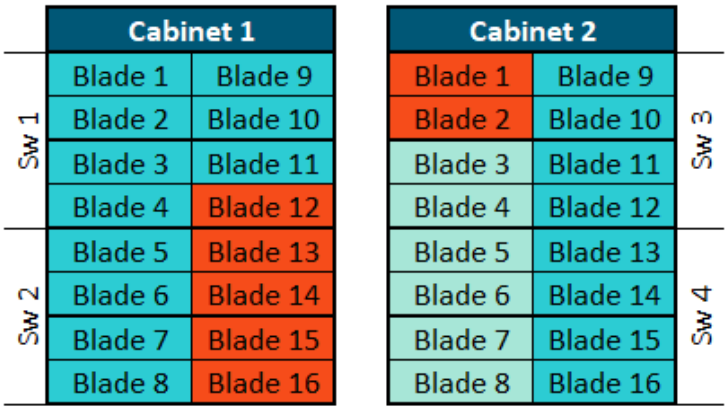


Figure 2: Multiple clusters of different types and different types of nodes

### Example 2: Interconnect topology

If you have vnodes connected to different switches and are running an MPI job, it is usually more efficient to run your job on vnodes that are connected to the same switch, which will reduce the number of hops required for inter-process communication (as shown above in Figure 2).

### Example 3: Shared storage

If your system has shared storage, jobs that do I/O will perform better if run on vnodes that have best access to the storage.

### Example 4: System location

Your PBS Professional complex may span multiple physical locations. In this case, it makes sense to create placement sets for each location rather than have jobs span different physical locations.

### Order of Placement Set Consideration Within a Placement Pool

The first step in the scheduling process is for the scheduler to order placement sets from smallest to largest, according to the following rules:

1. Static total ncpus of all vnodes in the set
2. Static total mem of all vnodes in the set
3. Dynamic free ncpus of all vnodes in the set
4. Dynamic free mem of all vnodes in the set (i.e., total amount of unused memory)

The scheduler then runs through the placement sets where the job statically fits, in order of smallest to largest, to place the job. The job is placed in the smallest set in which it dynamically fits. This ordering ensures that the scheduler will use the smallest possible placement set in which the job will fit dynamically.

In the example below, there are 24 nodes available to run jobs. There are two network switches in the configuration, Switch 1 and Switch 2, so the system has two placements sets:

- One where value of whatever = Switch 1
- One where value of whatever = Switch 2

There are four jobs currently running on the system and a user submits a job with the following request: qsub -l select=4:ncpus=8 my_test_job.

First the scheduler orders the placement sets from smallest to largest, so it will first look at the Switch 1 placement set, then the Switch 2 placement set. Per the steps above:

1. Yes, the job would fit in the Switch 1 set if all nodes were free (static check)
2. Ditto for the memory requirements (as none were specified)
3. Looking for dynamic fit to run job
    a. Look at Switch 1 (smallest) – 3 nodes available, job won't fit
    b. Look at Switch 2 – 6 nodes available, job will fit so it is run here

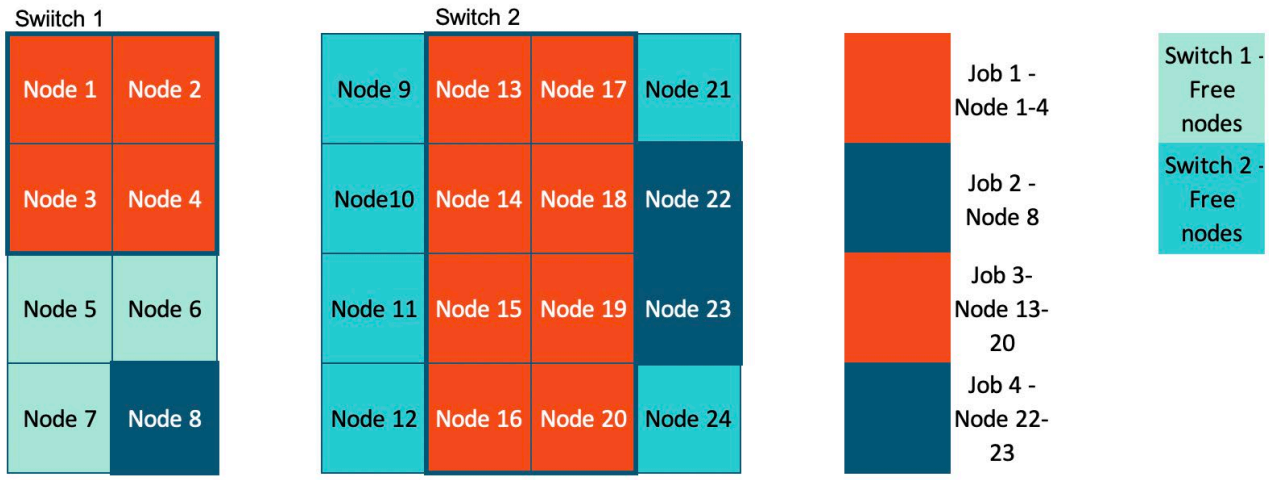


Figure 3: Sample 2-switch configuration

If no dynamic fit is possible, the scheduler waits until the job can fit. If the job can't statically fit in any defined placement set, the scheduler attempts to place it in the implicit placement set consisting of all vnodes.

### Scheduling Using Placement Sets

There are two primary methods for placement selection. The user can request a specific set using the -l place=group=<resource> statement at job submission, or the PBS Professional scheduler can determine best fit automatically.

A scheduler chooses the most specific placement pool available, following this order of precedence:

1. A per-job placement set if requested (the job includes -l place=group=<resource>)
2. A placement pool for the job's queue if one has been specified
3. A placement set from the placement pools in a scheduler's partition(s)
4. The default placement pool consisting of all vnodes if none of the above are defined

### Enabling and Configuring Placement Sets

There are a few basic steps to get placement sets configured and running using system-wide placement sets:

1. Determine which resources you want to use, create them, and enable the node grouping feature.
2. Add the resources to the node_group_key so the scheduler knows how to group the nodes.
3. Define the value of the resources for all relevant nodes.

| Create server placement sets | Commands |
|---|---|
| Create resources | `# qmgr -c "create resource switch type=string,flag=h"` |
| Enable node grouping | `# qmgr -c "set server node_group_enable = True` |
| Add the resource names to the node_group_key | `# qmgr -c "set server node_group_key="switch""` |
| Define the value of the resources for each node; can be done using qmgr command or in v2 config file for vnodes | `#qmgr -c "set node compute_1 resources_available.switch = 'infiniBandB'"` |

Adding a resource to a scheduler's resources: Line is required only if the resource is to be specifically requested by jobs. It is not required for -lplace=group=<resource name>. Resources used only for defining placement sets, and not for allocation to jobs, do not need to be listed in the "resources:" line in <sched_priv directory>/sched_config. For example, if you create a resource just for defining placement sets and jobs will not be requesting this resource, you do not need to list it in the "resources:" line.

If you want users to be able to request specific placement sets, follow the steps above and do the following additional steps:

| Additional steps | Commands |
|---|---|
| Add resources to sched_config file | Edit file: ***$PBS_HOME/sched_priv/sched_config*** and add your new resource to the "resources:" line:<br>`resources: "ncpus, mem, arch, host, vnode, aoe, eoe, switch"` |
| Restart the scheduler to re-read the config | `# kill -HUP $(pgrep -f pbs_sched)` |

### Creating Placement Sets for Queues

Queue-level placement sets are defined by setting a queue's node_group_key attribute to a list of vnode-level string array resources.

| Queue-level placement sets | Commands |
|---|---|
| Create resources as above | `Qmgr: set queue workq node_group_key = <router,switch>` |
| Set the node_group_key attribute to the name(s) of one or more vnode-level resources | `Qmgr: set queue workq node_group_key = <router,switch>` |

### Conclusion

Placement sets are an easy way to optimize job placement and performance. You can configure them to understand hardware topology, provide differentiation between different nodes, and improve overall system performance.

### References

PBS Professional 19.2.2 Admin Guide section 4.9.32