4.8.32.5.iv Viewing Jobs That Have Been Moved to Another Server

If you are connected to ServerA and a job submitted to ServerA has been moved from ServerA to ServerB through peer scheduling, in order to display it via `qstat`, give the job ID as an argument to `qstat`. If you only give the `qstat` command, the job will not appear to exist. For example, the job `123.ServerA` is moved to ServerB. In this case, use

```
qstat 123
```

or

```
qstat 123.ServerA
```

To list all jobs at ServerB, you can use:

```
qstat @ServerB
```

4.8.32.5.v Peer Scheduling and Hooks

When a job is pulled from one complex to another, the following happens:

- Hooks are applied at the new complex as if the job had been submitted locally
- Any movejob hooks at the furnishing server are run

4.8.32.6 Peer Scheduling Caveats

- Each complex can peer with at most 50 other complexes.
- When using peer scheduling, group hard limits may not be applied correctly. This can occur when the job owner's group is different on the furnishing and pulling systems. For help in avoiding this problem, see section 4.8.32.3.v, "Making User-to-group Mappings Consistent Across Complexes", on page 154.
- You cannot peer schedule between a Windows complex and a Linux complex.
- When the pulling scheduler runs the job the first time, the job is run as if the job still had all of the resources it had at the furnishing complex. If the job is requeued and restarted at the pulling complex, the job picks up new default resources from the pulling complex, and is scheduled according to the newly-inherited resources from the pulling complex.
- Peer scheduling is not supported for job arrays.

4.8.33 Placement Sets

Placement sets are the sets of vnodes within which PBS will try to place a job. PBS tries to group vnodes into the most useful sets, according to how well connected the vnodes are, or the values of resources available at the vnodes. Placement sets are used to improve task placement (optimizing to provide a "good fit") by exposing information on system configuration and topology. The scheduler tries to put a job in the smallest appropriate placement set.

4.8.33.1 Definitions

Task placement

The process of choosing a set of vnodes to allocate to a job that will satisfy both the job's resource request (select and place specifications) and the configured scheduling policy.

Placement Set

A set of vnodes. Placement sets are defined by the values of vnode-level string array resources. A placement set is all of the vnodes that have the same value for a specified defining resource substring. For example, if the defining resource is a vnode-level string array named "*switch*", which can have values "S1", "S2", or "S3": the set of vnodes which have a substring matching "switch=S2" is a placement set.

Placement sets can be specified at the server or queue level.

Placement Set Series

A set of placement sets; a set of sets of vnodes.

A placement set series is all of the placement sets that are defined by specifying one string array resource. Each placement set in the series is the set of vnodes that share one value for the resource. There is one placement set for each value of the resource. If the resource takes on N values at the vnodes, then there are N sets in the series. For example, if the defining resource is a string array named "*switch*", which can have values "S1", "S2", or "S3", there are three sets in the series. The first is defined by the value "S1", where all the vnodes in that set have the value "S1" for the resource switch. The second set is defined by "S2", and the third by "S3".

Each of the resources named in node_group_key specifies a placement series. For example, if the server's node_group_key attribute contains "router,switch", then the server has two placement set series.

Placement Pool

All of the placement sets that are defined; the server can have a placement pool, and each queue can have its own placement pool. If a queue has no placement pool, the scheduler uses the server's placement pool.

A placement pool is the set of placement set series that are defined by one or more string array resources named in node_group_key.

For example, if the server's node_group_key attribute contains "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 statically fits into a placement set if the job could fit into the placement set if the set were empty. It might not fit right now with the currently available resources.

Dynamic Fit

A job dynamically fits into a placement set if it will fit with the currently available resources (i.e. the job can fit right now).

4.8.33.2 Requirements for Placement Sets

- Placement sets are enabled by setting the server's node_group_enable attribute to *True*
- Server-level placement sets are defined by setting the server's node_group_key attribute to a list of vnode-level string array resources.
- Queue-level placement sets are defined by setting a queue's node_group_key attribute to a list of vnode-level string array resources.
- At least one vnode-level string array resource must exist on vnodes and be set to values that can be used to partition the vnodes.

4.8.33.3 Description of Placement Sets

4.8.33.3.i What Defines a Placement Set, Series, or Pool

Placement sets are defined by the values of vnode-level string array resources. You define placement sets by specifying the names of these resources in the node_group_key attribute for the server and/or queues. Each value of each resource defines a different placement set. A placement set is all of the vnodes that have the same value for a specified defining resource. For example, if the defining resource is a vnode-level string array named "*switch*", which has the values "S1", "S2", and "S3", the set of vnodes where switch has the value "S2" is a placement set. If some vnodes have more than one substring, and one of those substrings is the same in each vnode, those vnodes make up a placement set. For example, if the resource is "*router*", and vnode V0 has resources_available.router set to "r1i0,r1", and vnode V1 has resources_available.router set to "r1i1,r1", V0 and V1 are in the placement set defined by resources_available.router = "r1". If the resource has *N* distinct values across the vnodes, including the value zero and being unset, there can be *N-1* or *N* placement sets defined by that resource. If the only_explicit_psets scheduler attribute is *False*, there are *N* placement sets. If the only_explicit_psets scheduler attribute is *True*, there are *N-1* placement sets; see section 4.8.33.3.v, "Placement Sets Defined by Unset Resources", on page 159.

Each placement set can have a different number of vnodes; the number of vnodes is determined only by how many vnodes share that resource value.

Each placement set series is defined by the values of a single resource across all the vnodes. For example, if there are three switches, S1, S2 and S3, and there are vnodes with resources_available.switch that take on one or more of these three values, then there will be three placement sets in the series.

Whenever you define any placement sets, you are defining a placement pool. Placement pools can be defined for the server and for each queue. You define a server-level placement pool by setting the server's node_group_key to a list of one or more vnode-level string array resources. You define a queue-level placement pool by similarly setting the queue's node_group_key.

4.8.33.3.ii Vnode Participation in Placement Sets, Series, and Pools

Each vnode can be in multiple placement sets, placement set series, and placement pools.

A vnode can be in multiple placement sets in the same placement set series. For example, if the resource is called "*router*", and a vnode's router resource is set to "R1, R2", then the vnode will be in the placement set defined by router = R1 and the set defined by router = R2.

A vnode is in a placement series whenever the resource that defines the series is defined on the vnode. For example, if placement sets are defined by the values of the router and the switch resources, and a vnode has value *R1* for router, and *S1* for switch, then the vnode is in both placement series, because it is in the set that shares the *R1* value for router, and the set that shares the *S1* value for switch. Each of those sets is one of a different series.

The server has its own placement pool if the server's node_group_key attribute is set to at least one vnode-level string array resource. Similarly, each queue can have its own placement pool. A vnode can be in any placement pool that specifies a resource that is defined on the vnode.

4.8.33.3.iii Multihost Placement Sets

Placement sets, series, and pools can span hosts. Placement sets can be made up of vnodes from anywhere, regardless of whether the vnode is from a multi-vnode host.

To set up a multihost placement set, choose a string array resource for the purpose, and list it in the desired node_group_key attribute. For example, create a string_array resource called "span":

```
Qmgr: create resource span type=string_array, flag=h
```

Add the resource "span" to node_group_key on the server or queue. Use `qmgr` to give it the same value on all the desired vnodes. You can write a script that sets the same value on each vnode that you want in your placement set.

4.8.33.3.iv Machines with Multiple Vnodes

Machines with multiple vnodes are represented as a generic set of vnodes. Placement sets are used to allocate resources on a single machine to improve performance and satisfy scheduling policy and other constraints. Jobs are placed on vnodes using placement set information. For placement set generation information for SGI systems, see section 9.2.8.1, "Generation of Placement Set Information", on page 430.

4.8.33.3.v Placement Sets Defined by Unset Resources

The only_explicit_psets scheduler attribute controls whether unset resources define placement sets.

- If the only_explicit_psets scheduler attribute is *False*, vnodes where a defining resource is unset are grouped into their own placement set, for each defining resource. For example, if you have ten vnodes, on which there is a string resource COLOR, where two have COLOR set to "red", two are set to "blue", two are set to "green" and the rest are unset, there will be four placement sets defined by the resource COLOR. This is because the fourth placement set consists of the four vnodes where COLOR is unset. This placement set will also be the largest. Every resource listed in node_group_key can potentially define such a placement set.
- If the only_explicit_psets scheduler attribute is *True*, vnodes where a resource is unset are not grouped into placement sets.

4.8.33.3.vi Placement Sets and Node Grouping

Node grouping is the same as one placement set series, where the placement sets are defined by a single resource. Node grouping has been superseded by placement sets.

In order to have the same behavior as in the old node grouping, group on a single resource. If this resource is a string array, it should only have one value on each vnode. This way, each vnode will only be in one node group.

4.8.33.4 How Placement Sets Are Used

You use placement sets to partition vnodes according to the value of one or more resources. Placement sets allow you to group vnodes into useful sets.

You can run multi-vnode jobs in one placement set. For example, it makes the most sense to run a multi-vnode job on vnodes that are all connected to the same high-speed switch.

PBS will attempt to place each job in the smallest possible set that is appropriate for the job.

4.8.33.4.i Order of Placement Pool Selection

The scheduler chooses one placement pool from which to select a placement set.

Queue placement pools override the server's placement pool. If a queue has a placement pool, jobs from that queue are placed using the queue's placement pool. If a queue has no placement pool (the queue's node_group_key is not defined), jobs are placed using the server's placement pool, if it exists.

A per-job placement set is defined by the `-l place` statement in the job's resource request. Since the job can only request one value for the resource, it can only request one specific placement set. A job's `place=group` resource request overrides the sets defined by the queue's or server's node_group_key.

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

1. A per-job placement set (job's `place=group=` request)
2. A placement set from the placement pool for the job's queue
3. A placement set from the complex-wide placement pool

4.8.33.4.ii Order of Placement Set Consideration Within Pool

The scheduler looks in the selected placement pool and chooses the smallest possible placement set that is appropriate for the job. The scheduler examines the placement sets in the pool and orders them, from smallest to largest, according to the following rules:

1. Static total ncpus of all vnodes in set
2. Static total mem of all vnodes in set
3. Dynamic free ncpus of all vnodes in set
4. Dynamic free mem of all vnodes in set

If a job can fit statically within any of the placement sets in the placement pool, then the scheduler places a job in the first placement set in which it fits dynamically. This ordering ensures the scheduler will use the smallest possible placement set in which the job will dynamically fit. If there are multiple placement sets where the job fits statically, but some are being used, the scheduler uses the first placement set where the job can run now. If the job fits statically into at least one placement set, but these placement sets are all busy, the scheduler waits until a placement set can fit the job dynamically.

For example, we have the following placement sets, and a job that requests 8 CPUs:

Set1 ncpus = 4

Set2 ncpus = 12; this placement set is full

Set3 ncpus = 16; this placement set is not being used

The scheduler looks at Set1; Set1 is statically too small, and the scheduler moves to the next placement set. Set2 is statically large enough, but the job does not fit dynamically. The scheduler looks at Set3; Set3 is large enough, and the job fits dynamically. The scheduler runs the job in Set3.

If the job requests 24 CPUs, the scheduler attempts to run the job in the set consisting of all vnodes that are associated with a specific queue, if do_not_span_psets is *False*.

4.8.33.4.iii Determining Whether Job Can Run

Whether the job can run in the selected placement pool is determined by the value of the do_not_span_psets attribute.

- If this attribute is *False*, and a job cannot statically fit into any placement set in the selected placement pool, the scheduler ignores defined placement sets and uses all available vnodes as its placement set, and runs the job without regard to placement sets.
- If the attribute is *True*, the scheduler does not run the job.

4.8.33.4.iv Order of Vnode Selection Within Set

The scheduler orders the vnodes within the selected placement set using the following rules:

- If node_sort_key is set, vnodes are sorted by node_sort_key. See section 4.8.50, “Sorting Vnodes on a Key”, on page 208.
- If node_sort_key is not set, the order in which the vnodes are returned by `pbs_statnode()`. This is the default order the vnodes appear in the output of the `pbsnodes -a` command.

The scheduler places the job on the vnodes according to their ordering above.

4.8.33.5 Summary of Placement Set Requirements

The steps to configure placement sets are given in the next section. The requirements are summarized here for convenience:

- Definitions of the resources of interest
- Vnodes defining a value for each resource to be used for placement sets (e.g., rack)
  - If defined via vnode definition, you must HUP the MoMs involved
- The server's or queue's node_group_key attribute must be set to the resources to be used for placement sets. For example, if we have custom resources named "rack", "socket", "board", and "boardpair", which are to be used for placement sets:

  Qmgr: set server node_group_key = "rack,socket,board,boardpair"

  - No signals needed, takes effect immediately
- Placement sets must be enabled at the server by setting the server's node_group_enable attribute to *True*. For example:

  Qmgr: set server node_group_enable=True

  - No signals needed, takes effect immediately

Adding a resource to the 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>`.

4.8.33.6 How to Configure Placement Sets

The following steps show how to satisfy the requirements for placement sets:

1. If the vnodes that you will use in placement sets are not defined, define them. See section 3.1.5, "Creating Vnodes", on page 28.
2. If the vnode-level string array resources that you will use to define placement sets do not exist, create them. See section 5.14.5, "Configuring Host-level Custom Resources", on page 249.
3. Restart the server; see section 5.14.3.1, "Restarting the Server", on page 245.
4. If values for the vnode-level string array resources that you will use to define placement sets are not set at the vnodes you wish to use, set the values. See section 3.5, "How to Configure MoMs and Vnodes", on page 36.
5. If you use vnode definition files to set values for vnode-level string array resources, HUP the MoMs involved.
6. To create queue placement pools, set the node_group_key attribute to the name(s) of one or more vnode-level string array resources. Do this for each queue for which you want a separate pool. For example:

   ```
   Qmgr: set queue workq node_group_key = <router,switch>
   ```

7. To create a server placement pool, set the node_group_key server attribute to the name(s) of one or more vnode-level string array resources. For example:

   ```
   Qmgr: set server node_group_key = <router,switch>
   ```

   For example, to create a server-level placement pool for the resources host, L2 and L3:

   ```
   Qmgr: set server node_group_key = "host,L2,L3"
   ```

8. Set the server's node_group_enable attribute to *True*

   ```
   Qmgr: set server node_group_enable = True
   ```

9. Set the do_not_span_psets scheduler attribute to *True* if you don't want jobs to span placement sets.

   ```
   Qmgr: set sched do_not_span_psets = True
   ```

10. Set the only_explicit_psets attribute to *True* if you don't want the scheduler to create placement sets from unset resources.

    ```
    Qmgr: set sched only_explicit_psets = True
    ```

11. For ease of reviewing placement set information, you can add the name of each resource used in each vnode's pnames attribute:

    ```
    Qmgr: active node <vnode name>,<vnode name>,...
    Qmgr: set node pnames += <resource name>
    ```

    or

    ```
    Qmgr: set node pnames = <resource list>
    ```

    For example:

    ```
    Qmgr: set node pnames = "board,boardpair,iruquadrant,iruhalf,iru,rack"
    ```

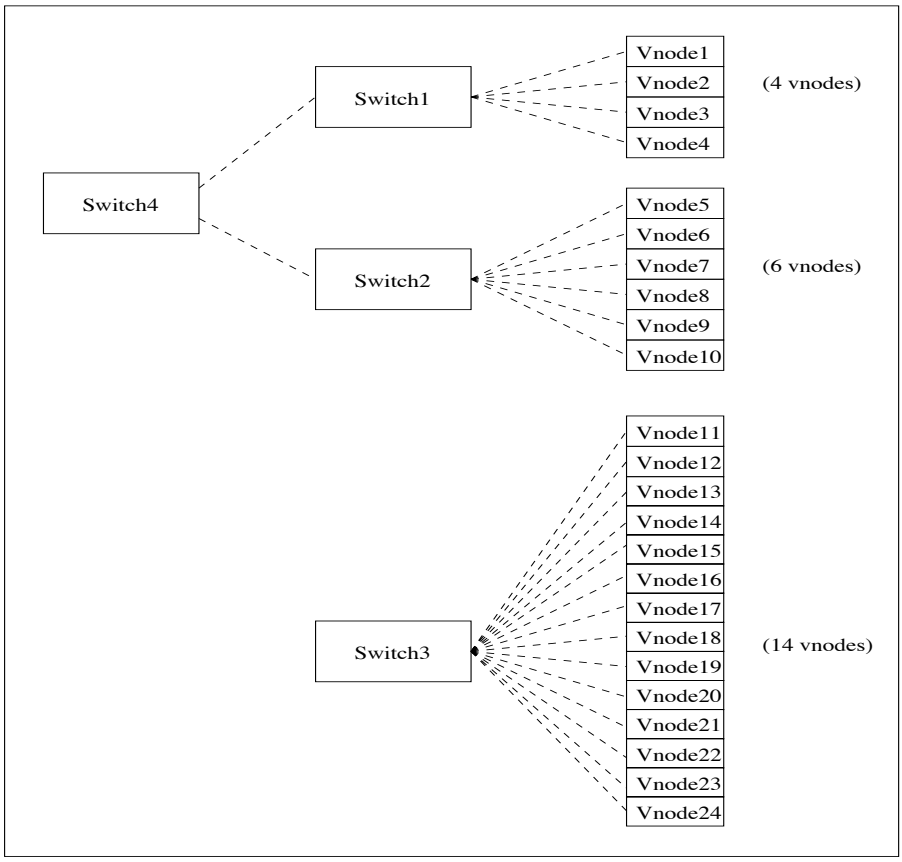
We recommend using the natural vnode for any placement set information that is invariant for a given host.

Resources used only for defining placement sets, and not for allocation to jobs, do not need to be listed in the `resources:` line in `PBS_HOME/sched_priv/sched_config`. So 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.

4.8.33.7 Examples of Creating Placement Sets

4.8.33.7.i Cluster with Four Switches

This cluster is arranged as shown with vnodes 1-4 on Switch1, vnodes 5-10 on Switch2, and vnodes 11-24 on Switch3. Switch1 and Switch2 are on Switch4.



To make the placement sets group the vnodes as they are grouped on the switches:

Create a custom resource called *switch*. The -h flag makes the resource requestable:

```
switch  type=string_array, flag=h
```

On vnodes[1-4] set:

```
resources_available.switch="switch1,switch4"
```

On vnodes[5-10] set:

```
resources_available.switch="switch2,switch4"
```

On vnodes[11-24] set:

```
resources_available.switch="switch3"
```

On the server set:

```
node_group_enable=True
node_group_key=switch
```

So you have 4 placement sets:

The placement set "switch1" has 4 vnodes

The placement set "switch2" has 6 vnodes

The placement set "switch3" has 14 vnodes

The placement set "switch4" has 10 vnodes

PBS will try to place a job in the smallest available placement set. Does the job fit into the smallest set (switch1)? If not, does it fit into the next smallest set (switch2)? This continues until it finds one where the job will fit.

4.8.33.7.ii Example of Configuring Placement Sets on a Multi-vnode Machine

For information on how to configure vnodes on a cpusetted machine in order to define new placement sets, use the instructions in section 3.5.2.3, "Configuring Machines with Cpusets", on page 37.

In this example, we define a new placement set using the new resource "NewRes". We create a file called `SetDefs` that contains the changes we want.

1. Create the new resource:
   ```
   Qmgr: create resource NewRes type=string_array, flag=h
   ```
2. Add NewRes to the server's node_group_key
   ```
   Qmgr: set server node_group_key+="NewRes"
   ```
3. Add NewRes to the value of the pnames attribute for the natural vnode. This makes the name of the resource you used easily available. Add a line like this to `SetDefs`:
   ```
   host3:  resources_available.pnames ="...,NewRes"
   ```
4. For each vnode, V, that's a member of a new placement set you're defining, add a line of the form:
   ```
   V: resources_available.NewRes = "<value1[,...]>"
   ```
   All the vnodes in the new set should have lines of that form, with the same resource value, in the new configuration file.

   Here the value of the resource is "P" and/or "Q".

   We'll put vnodes A, B and C into one placement set, and vnodes B, C and D into another.
   ```
   A:  resources_available.NewRes2 = P
   B:  resources_available.NewRes2 = "P,Q"
   C:  resources_available.NewRes2 = "P,Q"
   D:  resources_available.NewRes2 = Q
   ```
   For each new placement set you define, use a different value for the resource.
5. Add `SetDefs` and tell MoM to read it, to make a Version 2 MoM configuration file `NewConfig`.
   ```
   pbs_mom -s insert NewConfig SetDefs
   ```
6. Stop and restart the MoM. See "Starting and Stopping PBS: Linux" on page 137 in the PBS Professional Installation & Upgrade Guide.

4.8.33.7.iii Example of Placement Sets Using Colors

A placement pool is defined by two resources: `colorset1` and `colorset2,` by using "node_group_key=`colorset1,colorset2`".

If a vnode has the following values set:

```
resources_available.colorset1=blue, red
resources_available.colorset2=green
```

The placement pool contains at least three placement sets. These are:

{resources_available.colorset1=blue}

{resources_available.colorset1=red}

{resources_available.colorset2=green}

This means the vnode is in all three placement sets. The same result would be given by using one resource and setting it to all three values, e.g. `colorset=blue,red,green`.

Example: We have five vnodes `v1 - v5:`

```
v1 color=red host=mars
v2 color=red host=mars
v3 color=red host=venus
v4 color=blue host=mars
v5 color=blue host=mars
```

The placement sets are defined by

```
node_group_key=color
```

The resulting node groups would be: `{v1, v2, v3}, {v4, v5}`

4.8.33.7.iv Simple Switch Placement Set Example

Say you have a cluster with two high-performance switches each with half the vnodes connected to it. Now you want to set up placement sets so that jobs will be scheduled only onto the same switch.

First, create a new resource called "*switch*". See section 5.14.2, "Defining New Custom Resources", on page 234.

Next, we need to enable placement sets and specify the resource to use:

```
Qmgr: set server node_group_enable=True
Qmgr: set server node_group_key=switch
```

Now, set the value for switch on each vnode:

```
Qmgr: active node vnode1,vnode2,vnode3
Qmgr: set node resources_available.switch=A
Qmgr: active node vnode4,vnode5,vnode6
Qmgr: set node resources_available.switch=B
```

Now there are two placement sets:

switch=*A*: {vnode1, vnode2, vnode3}

switch=*B*: {vnode4, vnode5, vnode6}

4.8.33.8 Placement Sets and Reservations

When PBS chooses a placement set for a reservation, it makes the same choices as it would for a regular job. It fits the reservation into the smallest possible placement set. See section 4.8.33.4.ii, "Order of Placement Set Consideration Within Pool", on page 160.

When a reservation is created, it is created within a placement set, if possible. If no placement set will satisfy the reservation, placement sets are ignored. The vnodes allocated to a reservation are used as one single placement set for jobs in the reservation; they are not subdivided into smaller placement sets. A job within a reservation runs within the single placement set made up of the vnodes allocated to the reservation.

4.8.33.9 Placement Sets and Load Balancing

If you configure both placement sets and load balancing, the net effect is that vnodes that are over their load limit will be removed from consideration.

4.8.33.10 Viewing Placement Set Information

You can find information about placement sets in the following places:

- The server's node_group_enable attribute shows whether placement sets are enabled
- The server's node_group_key attribute contains the names of resources used for that queue's placement pool
- Each queue's node_group_key attribute contains the names of resources used for that queue's placement pool
- Each vnode's pnames attribute can contain the names of resources used for placement sets, if properly set
- The scheduler's do_not_span_psets attribute shows whether jobs are allowed to span placement sets
- The scheduler's only_explicit_psets attribute shows placement sets are created using unset resources
- PBS-generated MoM configuration files contain names and values of resources

4.8.33.11 Placement Set Caveats and Advice

- When you create a Version 2 configuration file for a pre-existing vnode, make sure it specifies all of the information about the vnode, such as resources and attribute settings. The creation of the configuration file overrides previous settings, and if the new file contains no specification for a resource or attribute, that resource or attribute becomes unset.
- If there is a vnode-level platform-specific resource set on the vnodes on a multi-vnode machine, then node_group_key should probably include this resource, because this will enable PBS to run jobs in more logical sets of vnodes.
- If the user specifies a job-specific placement set, for example `-lplace=group=switch`, but the job cannot statically fit into any switch placement set, then the job will still run, but not in a switch placement set.
- The pnames vnode attribute is for displaying to the administrator the resources used for placement sets. This attribute is not used by PBS.

4.8.33.11.i Non-backward-compatible Change in Node Grouping

Given the following example configuration:

vnode1: switch=A

vnode2: switch=A

vnode3: switch=B

vnode4: switch=B

vnode5: switch unset

`Qmgr: s s node_group_key=switch`

There is no change in the behavior of jobs submitted with `qsub -l ncpus=1`

version 7.1: The job can run on any node: node1, ..., node5

version 8.0: The job can run on any node: node1, ..., node5

Example of 8.0 and later behavior: jobs submitted with `qsub -lnodes=1`

version 7.1: The job can only run on nodes: node1, node2, node3, node4. It will never use node5

version 8.0: The job can run on any node: node1, ..., node5

Overall, the change for version 8.0 was to include every vnode in placement sets (when enabled). In particular, if a resource is used in node_group_key, PBS will treat every vnode as having a value for that resource, hence every vnode will appear in at least one placement set for every resource. For vnodes where a string resource is "unset", PBS will behave as if the value is “”.

4.8.33.12 Attributes and Parameters Affecting Placement Sets

do_not_span_psets
: Scheduler attribute. Specifies whether or not the scheduler requires the job to fit within one of the existing placement sets. When do_not_span_psets is set to *True*, the scheduler will require the job to fit within a single existing placement set. The scheduler checks all placement sets, whether or not they are currently in use. If the job fits in a currently-used placement set, the job must wait for the placement set to be available. If the job cannot fit within a single placement set, it will not run.

  When this attribute is set to *False*, the scheduler first attempts to place the job in a single placement set. All existing placement sets are checked. If the job fits in an occupied placement set, the job waits for the placement set to be available. If there is no existing placement set, occupied or empty, into which the job could fit, the job runs regardless of placement sets, running on whichever vnodes can satisfy the job’s resource request.

  Format: *Boolean*

  Default value: *False* (This matches behavior of PBS 10.4 and earlier)

  Example: To require jobs to fit within one placement set:

  ```
  Qmgr: set sched do_not_span_psets=True
  ```

node_group_enable
: Server attribute. Specifies whether placement sets are enabled.

  Format: *Boolean*

  Default: *False*

node_group_key
: Server and queues have this attribute. Specifies resources to use for placement set definition. Queue’s attribute overrides server’s attribute.

  Format: *string_array*

  Default: Unset

only_explicit_psets
: Scheduler attribute. Specifies whether placement sets are created using unset resources. If *False*, for each defining resource, if there are vnodes where the value of the resource is unset, PBS creates a placement set for the series defined by that resource. If *True*, PBS does not create placement sets for resources that are unset.

  Format: *Boolean*

  Default: *False*

4.8.33.13 Errors and Logging

If do_not_span_psets is set to *True*, and a job requests more resources than are available in one placement set, the following happens:

- The job's comment is set to the following:
  “Not Running: can't fit in the largest placement set, and can't span psets”
- The following message is printed to the scheduler’s log:
  “Can't fit in the largest placement set, and can't span placement sets”