



OpenPBS

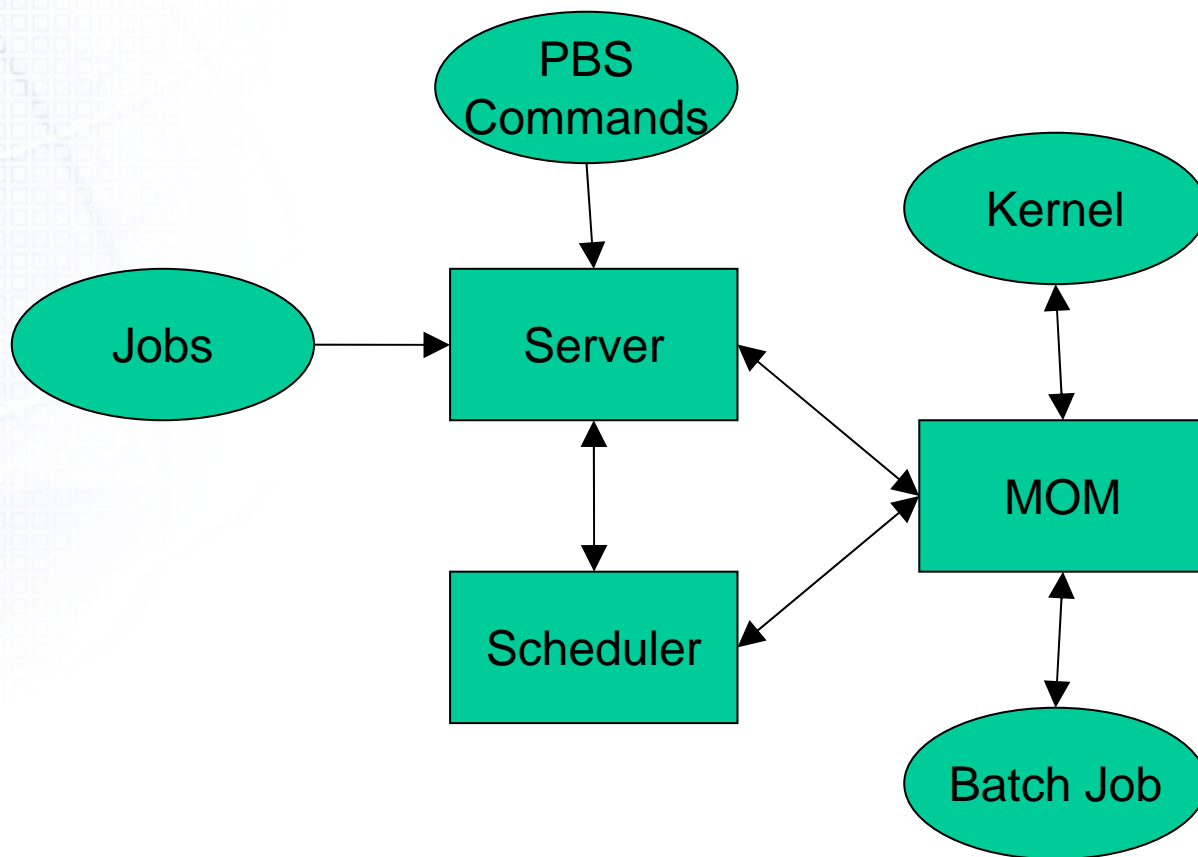


Features

- Job Priority
 - Users can specify the priority of their jobs.
- Job-Interdependency
 - OpenPBS enables the user to define a wide range of interdependencies between batch jobs such as execution order, synchronization, and execution conditioned on the success or failure of a specified other job.
- Automatic File Staging
 - OpenPBS provides users with the ability to specify any files that need to be copied onto the execution host before the job runs, and any that need to be copied off after the job completes.
- Single or Multiple Queue Support
 - OpenPBS can be configured with as many queues.
- Multiple Scheduling Algorithms
 - With OpenPBS you can select the standard first-in, first-out scheduling, or a more sophisticated scheduling algorithm.



OpenPBS Components





OpenPBS Components

- **Commands**
 - There are three command classifications: user commands, which any authorized user can use, operator commands, and manager (or administrator) commands.
- **Job Server**
 - The Server's main function is to provide the basic batch services such as receiving/creating a batch job, modifying the job, protecting the job against system crashes, and running the job. Typically there is one Server managing a given set of resources.



OpenPBS Components

- Job Executor (MOM)
 - The Job Executor is the daemon which actually places the job into execution. This daemon is informally called MOM as it is the mother of all executing jobs.
 - MOM places a job into execution when it receives a copy of the job from a Server. MOM creates a new session that is as identical to a user login session as is possible.
 - MOM also has the responsibility for returning the job's output to the user when directed to do so by the Server.
- Job Scheduler
 - The Job Scheduler daemon implements the site's policy controlling when each job is run and on which resources.
 - The Scheduler communicates with the various MOMs to query the state of system resources and with the Server for availability of jobs to execute.
 - Note that the Scheduler interfaces with the Server with the same privilege as the PBS manager.



Submit a PBS Job



A Sample PBS Job

- Example PBS job:

```
#!/bin/sh
```

```
#PBS -l walltime=1:00:00
```

```
#PBS -l mem=400mb
```

```
#PBS -l ncpus=4
```

```
#PBS -j oe
```

```
./subrun
```



A Sample PBS Job

- In our example above, lines 2-4 specify the “-l” resource list option, followed by a specific resource request. Specifically, lines 2-4 request 1 hour of wall-clock time, 400 megabytes (MB) of memory, and 4 CPUs.
- Line 5 is not a resource directive. Instead it specifies how PBS should handle some aspect of this job. (Specifically, the “-j oe” requests that PBS join the stdout and stderr output streams of the job into a single stream.)
- Finally line 7 is the command line for executing the program we wish to run.



Submitting a PBS Job

- Let's assume the above example script is in a file called "mysubrun". We submit this script using the qsub command:

```
% qsub mysubrun  
16387.cluster.pbspro.com
```
- You can also specify the option or directive on the qsub command line. This is particularly useful if you just want to submit a single instance of your job, but you don't want to edit the script. For example:

```
% qsub -l ncpus=16 -l walltime=4:00:00 mysubrun  
16388.cluster.pbspro.com
```
- In this example, the 16 CPUs and 4 hours of wallclock time will override the values specified in the job script.



Submitting a PBS Job

- Note that you are *not* required to use a separate “-l” for each resource you request. You can combine multiple requests by separating them with a comma, thusly:
- The same rule applies to the job script as well, as the next example shows.

```
% qsub -l ncpus=16,walltime=4:00:00 mysubrun  
16389.cluster.pbspro.com
```

```
#!/bin/sh  
#PBS -l walltime=1:00:00,mem=400mb  
#PBS -l ncpus=4  
#PBS -j oe  
./subrun
```



How PBS Parses a Job Script

- An initial line in the script that begins with the characters "#" or the character ":" will be ignored and scanning will start with the next line.
- A line in the script file will be processed as a directive to qsub if and only if the string of characters starting with the first non white space character on the line and of the same length as the directive prefix matches the directive prefix (i.e. "#PBS").
- The option character is to be preceded with the "-" character.



PBS System Resources

- Resources are specified using the “-l resource_list” option to qsub or in your job script.
- The resource_list argument is of the form:
 - resource_name[=[value]][,resource_name[=[value]],...]



PBS System Resources

- The resource values are specified using the following units:
 - node_spec (Node Specification Syntax)
 - a job with a -l nodes=nodespec resource requirement may now run on a set of nodes that includes time-shared nodes
 - and a job without a -l nodes=nodespec may now run on a cluster node
 - syntax for node_spec is any combination of the following separated by colons ':'
 - number {if it appears, it must be first}
 - node name
 - property
 - ppn=number
 - cpp=number
 - number:any other of the above[:any other]
 - where ppn is the number of processes (tasks) per node (defaults to 1) and cpp is the number of CPUs (threads) per process (also defaults to 1).
 - The 'node specification' value is one or more node_spec joined with the '+' character. For example, node_spec[+node_spec...][#suffix
 - The node specification can be followed by one or more global modifiers. E.g. "#shared" (requesting shared access to a node)



PBS System Resources

- `resc_spec` (Boolean Logic in Resource Requests)
 - It offers the ability to use boolean logic in the specification of certain resources (such as architecture, memory, wallclock time, and CPU count) within a single node.
 - Note that at this time, this feature controls the selection of single
 - nodes, not multiple hosts within a cluster, with the meaning
 - of “give me a node with the following properties”.
 - For example, say you wanted to submit a job that can run on either the Solaris or Irix operating system, and you want PBS to run the job on the first available node of either type. You could add the following “resc” specification to your `qsub` command line (or your job).



PBS System Resources

- Example

```
% qsub -l resc="(arch=='solaris7') || (arch=='irix')" mysubrun
% qsub -l resc="((arch=='solaris7') || (arch=='irix')) && (mem=100MB)
    &&(ncpus=4)"
#!/bin/sh
#PBS -l resc="(arch=='solaris7')||(arch=='irix')"
#PBS -l mem=100MB
#PBS -l ncpus=4
...
```

- The following example shows requesting different memory amounts depending on the architecture that the job runs on:

```
%qsub -l resc="( (arch=='solaris7') &&
(mem=100MB)||((arch=='irix')&&(mem=1GB) )"

```



PBS System Resources

- Time
 - [[hours:]minutes:]seconds[.milliseconds]
- Size
 - specifies the maximum amount in terms of bytes (default) or words
 - b or w bytes or words.
 - kb or kw Kilo (1024) bytes or words.
 - mb or mw Mega (1,048,576) bytes or words.
 - gb or gw Giga (1,073,741,824) bytes or words.
- String
 - comprised of a series of alpha-numeric characters containing no white space, beginning with an alphabetic character.
- Unitary
 - expressed as a simple integer



PBS Resources Available

Resource	Meaning	Units
arch	System architecture needed by job.	string
cput	Total amount of CPU time required by all processes in job.	Time
file	Maximum disk space requirements for a single file to be created by job.	Size
mem	Total amount of RAM memory required by job.	Size
ncpus	Number of CPUs (processors) required by job.	Unitary
nice	Requested “nice” (UNIX priority) value for job.	Unitary



PBS Resources Available

Resource	Meaning	Units
nodes	Number and/or type of nodes needed by job.	node_spec
pcput	Maximum amount of CPU time used by any single process in the job.	Time
pmem	Maximum amount of physical memory (workingset) used by any single process of the job.	Size
pvmem	Maximum amount of virtual memory used by any single process in the job.	size
vmem	Maximum amount of virtual memory used by all concurrent processes in the job.	Size
Walltime	Maximum amount of real time during which the job can be in the running state.	Time



Job Submission Options

Option	Function
-A account_string	Specifying a local account
-a date_time	Deferring execution
-c interval	Specifying job checkpoint interval
-e path	Redirecting output and error files
-h	Holding a job (delaying execution)
-l	Interactive-batch jobs
-j join	Merging output and error files
-k keep	Retaining output and error files on execution host



Job Submission Options

Option	Function
-l resource_list -l node_spec -l resc_spec	PBS System Resources Node Specification Syntax Boolean Logic in Resource Requests
-M user_list	Setting e-mail recipient list
-m MailOptions	Specifying e-mail notification
-N name	Specifying a job name
-o path	Redirecting output and error files
-p priority	Setting a job's priority
-q destination	Specifying Queue and/or Server
-r value	Marking a job as "rerunnable" or not



Job Submission Options

Option	Function
-S path_list	Specifying which shell to use
-u user_list	Specifying job userID
-V	Exporting environment variables
-v variable_list	Expanding environment variables
-W depend=list	Specifying Job Dependencies
-W group_list=list	Specifying job groupID
-W stagein=list	Input/Output File Staging
-W stageout=list	Input/Output File Staging
-z	Suppressing job identifier



Specifying Queue and/or Server

- If the `-q` option is not specified, the `qsub` command will submit the script to the default queue at the default server. The destination specification takes the following form:
 - `-q [queue[@host]]`
- Examples
 - `% qsub -q queue mysubrun`
 - `% qsub -q @server mysubrun`
 - `% qsub -q queueName@serverName mysubrun`
 - `% qsub -q queueName@serverName.domain.com mysubrun`
 - `#!/bin/sh`
 - `#PBS -q queueName`
 - `...`



Redirecting output and error files

- The “-o path” and “-e path” options to qsub allows you to specify the name of the files to which the standard output (stdout) and the standard error (stderr) file streams should be written.
- The path argument is of the form: [hostname:]path_name

- Examples

```
% qsub -o myOutputFile mysubrun
```

```
% qsub -o /u/james/myOutputFile mysubrun
```

```
% qsub -o myWorkstation:/u/james/myOutputFile mysubrun
```

```
#!/bin/sh
```

```
#PBS -o /u/james/myOutputFile
```

```
#PBS -e /u/james/myErrorFile
```

```
...
```



Exporting environment variables

- The “-V” option declares that all environment variables in the qsub command’s environment are to be exported to the batch job.

- Examples

```
% qsub -V mysubrun
```

```
#!/bin/sh
```

```
#PBS -V
```

```
...
```




Expanding environment variables

- The “-v variable_list” option to qsub expands the list of environment variables that are exported to the job.
- The variable_list is a comma separated list of strings of the form variable or variable=value. These variables and their values are passed to the job.

```
% qsub -v DISPLAY,myvariable=32 mysubrun
```



Specifying e-mail notification

- The “-m MailOptions” defines the set of conditions under which the execution server will send a mail message about the job.
- MailOptions
 - “a” send mail when job is aborted by batch system
 - “b” send mail when job begins execution
 - “e” send mail when job ends execution
 - “n” do not send mail

```
% qsub -m ae mysubrun
```

```
#!/bin/sh
```

```
#PBS -m b
```

```
...
```



Setting e-mail recipient list

- The “-M user_list” option declares the list of users to whom mail is sent by the execution server when it sends mail about the job. The user_list argument is of the form:
 - user[@host][,user[@host],...]
- If unset, the list defaults to the submitting user at the qsub host, i.e. the job owner.
- Example
 - `% qsub -M james@pbspro.com mysubrun`



Specifying a job name

- The “-N name” option declares a name for the job. The name specified may be up to and including 15 characters in length. It must consist of printable, non white space characters with the first character alphabetic.
- If the -N option is not specified, the job name will be the base name of the job script file specified on the command line.
- If no script file name was specified and the script was read from the standard input, then the job name will be set to STDIN.

- Example

```
% qsub -N myName mysubrun
```

```
#!/bin/sh
```

```
#PBS -N myName
```

```
...
```



Marking a job as “rerunnable” or not

- The “-r y|n” option declares whether the job is rerunnable.
- To rerun a job is to terminate the job and requeue it in the execution queue in which the job currently resides.

- Example

```
% qsub -r n mysubrun
```

```
#!/bin/sh
```

```
#PBS -r n
```

```
...
```



Specifying which shell to use

- The “-S path_list” option declares the shell that interprets the job script.
- The option argument path_list is in the form:
path[@host][,path[@host],...]
- If no matching host is found, then the path specified without a host will be selected, if present.
- If the -S option is not specified, the option argument is the null string, or no entry from the path_list is selected, then PBS will use the user’s login shell on the execution host.
- Example
 - `% qsub -S /bin/tcsh mysubrun`
 - `% qsub -S /bin/tcsh@mars,/usr/bin/tcsh@jupiter mysubrun`



Setting a job's priority

- The “-p priority” option defines the priority of the job.
- The priority argument must be an integer between -1024 and +1023 inclusive. The default is no priority which is equivalent to a priority of zero.
- Note that it is only advisory—the Scheduler may choose to override your priorities in order to meet local scheduling policy.

- Example

```
% qsub -p 120 mysubrun
```

```
#!/bin/sh
```

```
#PBS -p -300
```

```
...
```



Deferring execution

- The “-a date_time” option declares the time after which the job is eligible for execution.
- The date_time argument is in the form: [[[[[CC]YY]MM]DD]hhmm[.SS]]
 - CC is the first two digits of the year (the century),
 - YY is the second two digits of the year,
 - MM is the two digits for the month,
 - DD is the day of the month,
 - hh is the hour,
 - mm is the minute,
 - and the optional SS is the seconds.
- If the month, MM, is not specified, it will default to the current month if the specified day DD, is in the future. Otherwise, the month will be set to next month.
- Likewise, if the day, DD, is not specified, it will default to today if the time hhmm is in the future. Otherwise, the day will be set to tomorrow.



Deferring execution

- For example, if you submit a job at 11:15am with a time of “1110”, the job will be eligible to run at 11:10am tomorrow.
- Example

```
% qsub -a 0700 mysubrun
```

```
#!/bin/sh
```

```
#PBS -a 10220700
```

```
...
```



Holding a job (delaying execution)

- The “-h” option specifies that a user hold be applied to the job at submission time. The job will be submitted, then placed in a hold state. The job will remain ineligible to run until the hold is released.

- Example

```
% qsub -h mysubrun
```

```
#!/bin/sh
```

```
#PBS -h
```

```
...
```



Specifying job checkpoint interval

- The “-c interval” option defines the interval at which the job will be checkpointed, if this capability is provided by the operating system (e.g. under SGI IRIX and Cray Unicos). If the job executes upon a host which does not support checkpointing, this option will be ignored.
- The interval argument is specified as:
 - “n” No checkpointing is to be performed.
 - “s” Checkpointing is to be performed only when the server executing the job is shutdown.
 - “c” Checkpointing is to be performed at the default minimum time for the server executing the job.
 - “c=minutes” Checkpointing is to be performed at an interval of minutes, which is the integer number of minutes of CPU time used by the job. This value must be greater than zero.
 - “u” Checkpointing is unspecified. Unless otherwise stated, "u" is treated the same as "s".
- If “-c” is not specified, the checkpoint attribute is set to the value “u”.



Specifying job checkpoint interval

- In our cluster, checkpointing is not supported.

- Example

```
% qsub -c s mysubrun
```

```
#!/bin/sh
```

```
#PBS -c=10:00
```

```
...
```



Specifying job userID

- The “-u user_list” option defines the user name under which the job is to run on the execution system.
- If unset, the user_list defaults to the user who is running qsub.
- The user_list argument is of the form: user[@host][,user[@host],...]
- Only one user name may be given per specified host
- A named host refers to the host on which the job is queued for execution, not the actual execution host. Authorization must exist for the job owner to run as the specified user.



Specifying job userID

- Example

```
% qsub -u james@jupiter,barney@purpleplanet  
mysubrun
```



Specifying job groupID

- The “-W group_list=g_list” option defines the group name under which the job is to run on the execution system.
- The g_list argument is of the form:
group[@host][,group[@host],...]
- Only one group name may be given per specified host.
- Example

```
% qsub -W group_list=grpA,grpB@jupiter mysubrun
```



Specifying a local account

- The “-A account_string” option defines the account string associated with the job.
- The account_string is an opaque string of characters and is not interpreted by the Server which executes the job. This value is often used by sites to track usage by locally defined account names.

- Example

```
% qsub -A acct# mysubrun
```

```
#!/bin/sh
```

```
#PBS -A accountNumber
```

```
...
```




Merging output and error files

- The “-j join” option declares if the standard error stream of the job will be merged with the standard output stream of the job.
- A join argument value of oe directs that the two streams will be merged, intermixed, as standard output.
- If the join argument is n or the option is not specified, the two streams will be two separate files.
- Example

```
% qsub -j oe mysubrun
```

```
#!/bin/sh
```

```
#PBS -j eo
```

```
...
```



Retaining output and error files on execution host

- The “-k keep” option defines which (if either) of standard output or standard error will be retained on the execution host.
- If not set, neither stream is retained on the execution host. The argument is either the single letter "e" or "o", or the letters "e" and "o" combined in either order. Or the argument is the letter "n". If “-k” is not specified, neither stream is retained.



Retaining output and error files on execution host

- “e” The standard error stream is to be retained on the execution host. The stream will be placed in the home directory of the user under whose user id the job executed. The file name will be the default file name given by: `job_name.esequence` where `job_name` is the name specified for the job, and `sequence` is the sequence number component of the job identifier.
- “o” The standard output stream is to be retained on the execution host. The stream will be placed in the home directory of the user under whose user id the job executed. The file name will be the default file name given by: `job_name.osequence` where `job_name` is the name specified for the job, and `sequence` is the sequence number component of the job identifier.
- “eo” Both standard output and standard error will be retained.
- “oe” Both standard output and standard error will be retained.
- “n” Neither stream is retained.



Retaining output and error files on execution host

- Example

```
% qsub -k oe mysubrun
```

```
#!/bin/sh
```

```
#PBS -k oe
```

```
...
```



Suppressing job identifier

- The “-z” option directs the qsub command to not write the job identifier assigned to the job to the command’s standard output.

- Example

```
% qsub -z mysubrun
```

```
#!/bin/sh
```

```
#PBS -z
```

```
...
```



Interactive-batch jobs

- The “-I” option declares that the job is to be run "interactively". The job will be queued and scheduled as any PBS batch job, but when executed, the standard input, output, and error streams of the job are connected through qsub to the terminal session in which qsub is running.
- If a script is given, it will be processed for directives, but no executable commands will be included with the job.
- When the job begins execution, all input to the job is from the terminal session in which qsub is running.
- When an interactive job is submitted, the qsub command will not terminate when the job is submitted. qsub will remain running until the job terminates, is aborted, or the user interrupts qsub with a SIGINT (the control-C key).
- If qsub is interrupted prior to job start, it will query if the user wishes to exit. If the user responds "yes", qsub exits and the job is aborted.



Interactive-batch jobs

- Keyboard-generated interrupts are passed to the job. Lines entered that begin with the tilde ('~') character and contain special sequences are interpreted by qsub itself.
- The recognized special sequences are:
 - ~. qsub terminates execution. The batch job is also terminated.
 - ~susp Suspend the qsub program if running under the C shell. "susp" is the suspend character, usually CNTL-Z.
 - ~asusp Suspend the input half of qsub (terminal to job), but allow output to continue to be displayed. Only works under the C shell.
 - "asusp" is the auxiliary suspend character, usually CNTL-Y.



Case Studies

- It is possible to specify multiple resource specification strings. The first resc specification will be evaluated. If it can be satisfied, then it will be used. If not, then next resc string will be used.

```
% qsub \
```

```
-l resc="(ncpus=16)&& (mem=1GB) &&(walltime=1:00)" \
```

```
-l resc="(ncpus=8) && (mem=512MB)&&(walltime=2:00)" \
```

```
-l resc="(ncpus=4) && (mem=256MB)&&(walltime=4:00)" ...
```

- Indicates that you want 16 CPUs, but if you can't have 16 CPUs, then give you 8 with half the memory and twice the wall-clock time. But if you can't have 8 CPUs, then give you four and 1/4 the memory, and four times the walltime.



Case Studies

- This is different than putting them all into one resc specification. If you were to do

```
% qsub -l resc= "(ncpus=16)||(ncpus=8)||(ncpus=4)" ...
```

- you would be requesting the first available node which has either 16, 8, or 4 CPUs. In this case, PBS doesn't go through all the nodes checking for 16 first, then 8, then 4, as it does when using multiple resc specifications.



Case Studies

- You can do more than just using the equality and assignment operators. You can describe the characteristics of a node, but not request them. For example, if you were to specify:

```
% qsub \
```

```
-l resc="(ncpus>16)&&(mem>=2GB)" -lncpus=2
```

```
-lmem=100MB
```

- you are indicating that you want a node with more than 16 CPUs but you only want 2 of them allocated to your job.



Job Attributes

- A PBS job has the following public attributes.
 - Account_Name
 - Reserved for local site accounting.
 - Checkpoint
 - If supported by the server implementation and the host operating system, the checkpoint attribute determines when checkpointing will be performed by PBS on behalf of the job.
 - depend
 - The type of inter-job dependencies specified by the job owner.
 - Error_Path
 - The final path name for the file containing the job's standard error stream.



Job Attributes

- Execution_Time
 - The time after which the job may execute.
- group_list
 - A list of group_names@hosts which determines the group under which the job is run on a given host.
- Hold_Types
 - The set of holds currently applied to the job. If the set is not null, the job will not be scheduled for execution and is said to be in the hold state. Note, the hold state takes precedence over the wait state.
- Job_Name
 - The name assigned to the job by the qsub or qalter command.



Job Attributes

- Join_Path
 - If the Join_Paths attribute is TRUE, then the job's standard error stream will be merged, inter-mixed, with the job's standard output stream and placed in the file determined by the Output_Path attribute. The Error_Path attribute is maintained, but ignored.
- Keep_Files
 - The corresponding streams of the batch job will be retained on the execution host upon job termination. Keep_Files overrides the Output_Path and Error_Path attributes.
- Mail_Points
 - Identifies the state changes at which the server will send mail about the job.
- Mail_Users
 - The set of users to whom mail may be sent when the job makes certain state changes.



Job Attributes

- Output_Path
 - The final path name for the file containing the job's standard output stream.
- Priority
 - The job scheduling priority assigned by the user.
- Rerunable
 - The rerunable flag given by the user.
- Resource_List
 - The list of resources required by the job.
- Shell_Path_List
 - A set of absolute paths of the program to process the job's script file.



Job Attributes

- stagein
 - The list of files to be staged in prior to job execution.
- stageout
 - The list of files to be staged out after job execution.
- User_List
 - The list of user@hosts which determines the user name under which the job is run on a given host.
- Variable_List
 - This is the list of environment variables passed with the Queue Job batch request.
- comment
 - An attribute for displaying comments about the job from the system. Visible to any client.



Job Attributes

- The following attributes are read-only, they are established by the Server and are visible to the user but cannot be set by a user.
 - alt_id
 - For a few systems, such as Irix 6.x running Array Services, the session id is insufficient to track which processes belong to the job. Where a different identifier is required, it is recorded in this attribute. If set, it will also be recorded in the end-of-job accounting record. For Irix 6.x running Array Services, the alt_id attribute is set to the Array Session Handle (ASH) assigned to the job.
 - ctime
 - The time that the job was created.
 - etime
 - The time that the job became eligible to run, i.e. in a queued state while residing in an execution queue.
 - exec_host
 - If the job is running, this is set to the name of the host or hosts on which the job is executing. The format of the string is "node/ N[*C][+...]", where "node" is the name of a node, "N" is process or task slot on that node, and "C" is the number of CPUs allocated to the job. C does not appear if it is one.



Job Attributes

- egroup
 - If the job is queued in an execution queue, this attribute is set to the group name under which the job is to be run. [This attribute is available only to the batch administrator.]
- euser
 - If the job is queued in an execution queue, this attribute is set to the user name under which the job is to be run. [This attribute is available only to the batch administrator.]
- hostname
 - The name used as a basename for various files, such as the job file, script file, and the standard output and error of the job. [This attribute is available only to the batch administrator.]
- interactive
 - True if the job is an interactive PBS job.
- Job_Owner
 - The login name on the submitting host of the user who submitted the batch job.
- job_state
 - The state of the job.



Job Attributes

- mtime
 - The time that the job was last modified, changed state, or changed locations.
- qtime
 - The time that the job entered the current queue.
- queue
 - The name of the queue in which the job currently resides.
- queue_rank
 - An ordered, non-sequential number indicating the job's position within the queue. This is provided as an aid to the Scheduler. [This attribute is available to the batch manager only.]
- queue_type
 - An identification of the type of queue in which the job is currently residing. This is provided as an aid to the Scheduler. [This attribute is available to the batch manager only.]



Job Attributes

- resources_used
 - The amount of resources used by the job. This is provided as part of job status information if the job is running.
- server
 - The name of the server which is currently managing the job.
- session_id
 - If the job is running, this is set to the session id of the first executing task.
- substate
 - A numerical indicator of the substate of the job. The substate is used by the PBS Server internally. The attribute is visible to privileged clients, such as the Scheduler.



Checking Job / System Status

The `qstat` Command



Checking Job Status

- Executing the `qstat` command without any options displays job information in the default format.
 - The job identifier assigned by PBS
 - The job name given by the submitter
 - The job owner
 - The CPU time used
 - The job state
 - The queue in which the job resides



The qstat Command

- The job state is abbreviated to a single character:
 - “E” Job is exiting after having run
 - “H” Job is held
 - “Q” Job is queued, eligible to run or be routed
 - “R” Job is running
 - “S” Job is suspended
 - “T” Job is in transition (being moved to a new location)
 - “W” Job is waiting for its requested execution time to be reached



The qstat Command

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat
```

Job id	Name	User	Time Use	S	Queue
1378.tdgrocks	Set1	obraun	342:00:1	R	default
1379.tdgrocks	Set2	obraun	418:05:2	R	default
1380.tdgrocks	Set3	obraun	418:49:2	R	default
1381.tdgrocks	Set4	obraun	418:18:2	R	default
1382.tdgrocks	Set5	obraun	418:26:3	R	default
1564.tdgrocks	sin	lyang	320:26:5	R	default
1565.tdgrocks	sin	lyang	295:24:3	R	default
1566.tdgrocks	sin	lyang	157:53:1	R	default
1567.tdgrocks	sin	lyang	159:18:4	R	default
1568.tdgrocks	sin	lyang	158:15:2	R	default
1569.tdgrocks	sin	lyang	160:25:5	R	default
1574.tdgrocks	fk	lyang	171:58:2	R	default
1575.tdgrocks	fk	lyang	172:09:5	R	default
1576.tdgrocks	fk	lyang	119:08:1	R	default
1577.tdgrocks	fk	lyang	119:04:5	R	default
1620.tdgrocks	gamess	justin	416:10:2	R	default
1702.tdgrocks	zsrk2_512n4	01400037	04:47:39	R	default
1735.tdgrocks	sin	lyang	154:56:5	R	default
1736.tdgrocks	sin	lyang	154:57:1	R	default
1737.tdgrocks	sin	lyang	58:54:24	R	default
1738.tdgrocks	sin	lyang	41:56:09	R	default
1739.tdgrocks	sin	lyang	41:44:51	R	default
1740.tdgrocks	sin	lyang	42:32:39	R	default
1741.tdgrocks	sin	lyang	42:15:13	R	default
1742.tdgrocks	sin	lyang	41:20:10	R	default
1743.tdgrocks	sin	lyang	41:28:40	R	default
1744.tdgrocks	sin	lyang	33:43:04	R	default



The qstat Command

- An alternative display (accessed via the “-a” option) is also provided that includes extra information about jobs, including the following additional fields:
 - Session ID
 - Number of nodes requested
 - Number of parallel tasks (or CPUs)
 - Requested amount of memory
 - Requested amount of wallclock time
 - Elapsed time in the current job state.



The qstat Command

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat -a  
tdgrocks.sci.hkbu.edu.hk:  
  
Job ID      Username Queue      Jobname      SessID  NDS  TSK  Req'd  Req'd  Elap  
-----  
1378.tdgrocks.s obraun  default  Set1         8810    1  --   --    --    R 420:5  
1379.tdgrocks.s obraun  default  Set2        10387    1  --   --    --    R 420:5  
1380.tdgrocks.s obraun  default  Set3        30190    1  --   --    --    R 420:5  
1381.tdgrocks.s obraun  default  Set4        29002    1  --   --    --    R 420:5  
1382.tdgrocks.s obraun  default  Set5        29038    1  --   --    --    R 420:5  
1564.tdgrocks.s lyang   default  sin         17884    1  --   --    --    R 321:4  
1565.tdgrocks.s lyang   default  sin         21257    1  --   --    --    R 321:4  
1566.tdgrocks.s lyang   default  sin         21598    1  --   --    --    R 321:4  
1567.tdgrocks.s lyang   default  sin         17340    1  --   --    --    R 321:3  
1568.tdgrocks.s lyang   default  sin         24172    1  --   --    --    R 321:3  
1569.tdgrocks.s lyang   default  sin         28719    1  --   --    --    R 321:3  
1574.tdgrocks.s lyang   default  fk          18904    1  --   --    --    R 320:5  
1575.tdgrocks.s lyang   default  fk          18939    1  --   --    --    R 320:5  
1576.tdgrocks.s lyang   default  fk          26513    1  --   --    --    R 320:5  
1577.tdgrocks.s lyang   default  fk          26546    1  --   --    --    R 320:5  
1620.tdgrocks.s justin  default  gamess      29386    1  --   --    --    R 212:1  
1702.tdgrocks.s 01400037 default  zzrk2_512n  13561    1  --   --    --    R 159:2  
1735.tdgrocks.s lyang   default  sin         15989    1  --   --    --    R 155:2  
1736.tdgrocks.s lyang   default  sin         16024    1  --   --    --    R 155:2  
1737.tdgrocks.s lyang   default  sin         15350    1  --   --    --    R 155:2  
1738.tdgrocks.s lyang   default  sin         14506    1  --   --    --    R 155:2  
1739.tdgrocks.s lyang   default  sin         16318    1  --   --    --    R 155:2  
1740.tdgrocks.s lyang   default  sin         16353    1  --   --    --    R 155:2  
1741.tdgrocks.s lyang   default  sin         9024     1  --   --    --    R 155:2
```



Viewing Specific Information

- If the operand is a job identifier, it must be in the following form:
 - `sequence_number[.server_name][@server]`
- where `sequence_number.server_name` is the job identifier assigned at submittal time, see `qsub`.
- If the operand is a destination identifier, it takes one of the following three forms:
 - `queue`
 - `@server`
 - `queue@server`



Checking Server Status

- The “-B” option to qstat displays the status of the specified PBS Batch Server. The three letter abbreviations correspond to various job limits and counts as follows: Maximum, Total, Queued, Running, Held, Waiting, Transiting, and Exiting. The last column gives the status of the server itself: active, idle, or scheduling.



Checking Server Status

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat -B  
Server          Max Tot Que Run Hld Wat Trn Ext Status  
-----  
tdgrocks.sci.hkb 0  41  0  41  0  0  0  0 Active  
[02404435@tdgrocks 02404435]$
```



Checking Server Status

- When querying jobs, servers, or queues, you can add the “-f” option to qstat to change the display to the full or long display. For example, the Server status shown above would be expanded using “-f” as shown below:



Checking Server Status

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat -Bf  
Server: tdgrocks.sci.hkbu.edu.hk  
server_state = Active  
scheduling = True  
total_jobs = 41  
state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:41 Exiting:0  
acl_host_enable = False  
managers = maui@tdgrocks.sci.hkbu.edu.hk,root@tdgrocks.sci.hkbu.edu.hk  
default_queue = default  
log_events = 511  
mail_from = adm  
query_other_jobs = True  
resources_assigned.nodect = 30  
scheduler_iteration = 600  
node_ping_rate = 300  
node_check_rate = 600  
pbs_version = torque_1.0.1p4  
[02404435@tdgrocks 02404435]$ █
```



Checking Queue Status

- The “-Q” option to qstat displays the status of all (or any specified) queues at the (optionally specified) PBS Server. One line of output is generated for each queue queried.
- The three letter abbreviations correspond to limits, queue states, and job counts as follows: Maximum, Total, Enabled Status, Started Status, Queued, Running, Held, Waiting, Transiting, and Exiting. The last column gives the type of the queue: routing or execution.



Checking Queue Status

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat -Q  
Queue           Max Tot Ena Str Que Run Hld Wat Trn Ext Type  
-----  
default         0  41 yes yes  0  41  0  0  0  0 Execution  
[02404435@tdgrocks 02404435]$  
[02404435@tdgrocks 02404435]$ qstat -Qf  
Queue: default  
  queue_type = Execution  
  total_jobs = 41  
  state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:41 Exiting:0  
  resources_assigned.nodect = 30  
  enabled = True  
  started = True  
[02404435@tdgrocks 02404435]$
```




Viewing Job Information

- By specifying the “-f” option and a job identifier, PBS will print all information known about the job (e.g. resources requested, resource limits, owner, source, destination, queue, etc.) as shown in the following example. (See “Job Attributes” on the slides before.)



Viewing Job Information

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat -f 1928  
Job Id: 1928.tdgrocks.sci.hkbu.edu.hk  
Job_Name = gs_64  
Job_Owner = lhtang@tdgrocks.sci.hkbu.edu.hk  
resources_used.cput = 00:00:00  
resources_used.mem = 3484Kb  
resources_used.vmem = 12740Kb  
resources_used.walltime = 00:41:39  
job_state = R  
queue = default  
server = tdgrocks.sci.hkbu.edu.hk  
Checkpoint = u  
ctime = Tue Apr 20 00:08:47 2004  
Error_Path = tdgrocks.sci.hkbu.edu.hk:/home/lhtang/projects/coulomb_gas/gs_  
_64.err  
exec_host = comp-pvfs-0-62.local/1+comp-pvfs-0-52.local/1+comp-pvfs-0-46.lo  
cal/1+comp-pvfs-0-32.local/1+comp-pvfs-0-31.local/0+comp-pvfs-0-28.loca  
l/1+comp-pvfs-0-28.local/0+comp-pvfs-0-27.local/1+comp-pvfs-0-27.local/  
0+comp-pvfs-0-26.local/1+comp-pvfs-0-26.local/0+comp-pvfs-0-25.local/1+  
comp-pvfs-0-25.local/0+comp-pvfs-0-18.local/1+comp-pvfs-0-15.local/1+co  
mp-pvfs-0-15.local/0+comp-pvfs-0-11.local/1+comp-pvfs-0-11.local/0+comp  
-pvfs-0-9.local/1+comp-pvfs-0-9.local/0+comp-pvfs-0-8.local/1+comp-pvfs  
-0-8.local/0+comp-pvfs-0-6.local/1+comp-pvfs-0-6.local/0+comp-pvfs-0-4.  
local/1+comp-pvfs-0-4.local/0+comp-pvfs-0-3.local/1+comp-pvfs-0-3.local  
/0+comp-pvfs-0-2.local/1+comp-pvfs-0-2.local/0  
Hold_Types = n  
Join_Path = n  
Keep_Files = n  
Mail_Points = a  
mtime = Tue Apr 20 00:27:17 2004  
Output_Path = tdgrocks.sci.hkbu.edu.hk:/home/lhtang/projects/coulomb_gas/gs  
_64.log  
Priority = 0  
qtime = Tue Apr 20 00:08:47 2004  
Rerunnable = False  
Resource_List.nodect = 30  
Resource_List.nodes = 30  
Resource_List.walltime = 96:00:00  
session_id = 27121  
Variable_List = PBS_O_HOME=/home/lhtang,PBS_O_LANG=en_US.iso885915,  
PBS_O_LOGNAME=lhtang,  
PBS_O_PATH=/u1/local/pgi/linux86/bin:/usr/bin:/u1/local/hpjdk/bin:/u1/
```



List User-Specific Jobs

- The “-u” option to qstat displays jobs owned by any of a list of user names specified.
- The syntax of the list of users is:
 - user_name[@host][,user_name[@host],...]
- Host names are not required, and may be “wild carded” on the left end, e.g. “*.pbspro.com”. user_name without a “@host” is equivalent to “user_name@*”, that is at any host.



List User-Specific Jobs

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat -u 01400037  
tdgrocks.sci.hkbu.edu.hk:  
Job ID          Username Queue      Jobname      SessID NDS  TSK  Req'd  Req'd  Elap  
-----  
1702.tdgrocks.s 01400037 default    zrk2_512n   13561  1  --    --    --  R 159:4  
1825.tdgrocks.s 01400037 default    zrk2_256p   16618  1  --    --    --  R 85:29  
1827.tdgrocks.s 01400037 default    zrk2_512n   14987  1  --    --    --  R 85:09  
1894.tdgrocks.s 01400037 default    zrk2_256p   30236  1  --    --    --  R 30:52  
1895.tdgrocks.s 01400037 default    zrk2_512p   30404  1  --    --    --  R 30:20  
[02404435@tdgrocks 02404435]$ qstat -u 01400037,kiliu  
tdgrocks.sci.hkbu.edu.hk:  
Job ID          Username Queue      Jobname      SessID NDS  TSK  Req'd  Req'd  Elap  
-----  
1702.tdgrocks.s 01400037 default    zrk2_512n   13561  1  --    --    --  R 159:4  
1825.tdgrocks.s 01400037 default    zrk2_256p   16618  1  --    --    --  R 85:29  
1827.tdgrocks.s 01400037 default    zrk2_512n   14987  1  --    --    --  R 85:09  
1853.tdgrocks.s kiliu     default    hima.sobol   4295  1  --    --    --  R 47:14  
1854.tdgrocks.s kiliu     default    hima.nied    13183  1  --    --    --  R 47:13  
1894.tdgrocks.s 01400037 default    zrk2_256p   30236  1  --    --    --  R 30:52  
1895.tdgrocks.s 01400037 default    zrk2_512p   30404  1  --    --    --  R 30:20  
[02404435@tdgrocks 02404435]$
```



List Running Jobs

- The “-r” option to qstat displays the status of all running jobs at the (optionally specified) PBS Server. Running jobs include those that are running and suspended.



List Non-Running Jobs

- The “-i” option to qstat displays the status of all non-running jobs at the (optionally specified) PBS Server. Non-running jobs include those that are queued, held, and waiting.



Display Size in Gigabytes

- The “-G” option to qstat displays all jobs at the requested (or default) Server using the alternative display, showing all size information in gigabytes (GB) rather than the default of smallest displayable units.



Display Size in Megawords

- The “-M” option to qstat displays all jobs at the requested (or default) Server using the alternative display, showing all size information in megawords (MW) rather than the default of smallest displayable units. A word is considered to be 8 bytes.



List Nodes Assigned to Jobs

- The “-n” option to qstat displays the nodes allocated to any running job at the (optionally specified) PBS Server, in addition to the other information presented in the alternative display.
- The node information is printed immediately below the job and includes the node name and number of virtual processors assigned to the job.
- A text string of “--” is printed for non-running jobs.



List Nodes Assigned to Jobs

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat -n  
tdgrocks.sci.hkbu.edu.hk:  
Job ID      Username Queue   Jobname   SessID  NDS  TSK  Req'd  Req'd  Elap  
-----  
1378.tdgrocks.s obraun  default  Set1      8810    1  --   --    --  R 421:1  
  comp-pvfs-0-64  
1379.tdgrocks.s obraun  default  Set2     10387    1  --   --    --  R 421:1  
  comp-pvfs-0-52  
1380.tdgrocks.s obraun  default  Set3     30190    1  --   --    --  R 421:1  
  comp-pvfs-0-46  
1381.tdgrocks.s obraun  default  Set4     29002    1  --   --    --  R 421:1  
  comp-pvfs-0-45  
1382.tdgrocks.s obraun  default  Set5     29038    1  --   --    --  R 421:1  
  comp-pvfs-0-45  
1564.tdgrocks.s lyang   default  sin      17884    1  --   --    --  R 322:0  
  comp-pvfs-0-32  
1565.tdgrocks.s lyang   default  sin      21257    1  --   --    --  R 322:0  
  comp-pvfs-0-24  
1566.tdgrocks.s lyang   default  sin      21598    1  --   --    --  R 322:0  
  comp-pvfs-0-13  
1567.tdgrocks.s lyang   default  sin      17340    1  --   --    --  R 322:0  
  comp-pvfs-0-10  
1568.tdgrocks.s lyang   default  sin      24172    1  --   --    --  R 322:0  
  comp-pvfs-0-7
```



Display Job Comments

- The “-s” option to qstat displays the job comments, in addition to the other information presented in the alternative display.
- The job comment is printed immediately below the job.
- By default the job comment is updated by the Scheduler with the reason why a given job is not running, or when the job began executing.
- A text string of “--” is printed for jobs whose comment has not yet been set.



Display Job Comments

```
& qstat -s
```

Job ID	User	Queue	Jobname	Sess	NDS	TSK	Mem	Req'd Time	S	Elap Time
16.	south james	workq	aims14	--	--	1	--	0:01	H	--
Job held by james on Wed Aug 22 13:06:11 2001										
18.	south james	workq	aims14	--	--	1	--	0:01	W	--
Waiting on user requested start time										
51.	south barry	workq	airfoil	930	--	1	--	0:13	R	0:01
Job run on node south - started Thu Aug 23 at 10:56										
52.	south james	workq	subrun	--	--	1	--	0:10	Q	--
Not Running: No available resources on nodes										
57.	south susan	workq	solver	--	--	2	--	0:20	Q	--
--										



Display Queue Limits

- The “-q” option to qstat displays any limits set on the requested (or default) queues.
- Since PBS is shipped with no queue limits set, any visible limits will be site-specific. The limits are listed in the format shown below.



Display Queue Limits

```
02404435@tdgrocks:~  
[02404435@tdgrocks 02404435]$ qstat -q  
  
server: tdgrocks.sci.hkbu.edu.hk  
  
Queue          Memory CPU Time  Walftime Node  Run  Que  Lm  State  
-----  
default        --    --    --    --    40   0  --  E R  
                ---  ---  
                40   0  
  
[02404435@tdgrocks 02404435]$ █
```



Checking Job / System Status

The `qselect` Command



The qselect Command

- The qselect command provides a method to list the job identifier of those jobs which meet a list of selection criteria.
- Optional op component:
 - .eq. equal
 - .ne. not equal
 - .ge. greater than or equal to
 - .gt. greater than
 - .le. less than or equal to
 - .lt. less than



The qselect Command

- The available options to qselect are:
 - -a [op]date_time
 - Restricts selection to a specific time, or a range of times. The date_time argument is in the POSIX date format:
 - [[CC]YY]MMDDhhmm[.SS]
 - If op is not specified, jobs will be selected for which the Execution_Time and date_time values are equal.
 - -A account_string
 - Restricts selection to jobs whose Account_Name attribute matches the specified account_string.
 - -c [op] interval
 - Restricts selection to jobs whose Checkpoint interval attribute matches the specified relationship. The values of the Checkpoint attribute are defined to have the following ordered relationship:
 - $n > s > c = \text{minutes} > c > u$
 - If the optional op is not specified, jobs will be selected whose Checkpoint attribute is equal to the interval argument.



The qselect Command

- -h hold_list
 - Restricts the selection of jobs to those with a specific set of hold types. The hold_list argument is a string consisting of one or more occurrences the single letter n, or one or more of the letters u, o, or s in any combination. The letters represent the hold types:
 - n none
 - u user
 - o operator
 - s system
- -l resource_list
 - Restricts selection of jobs to those with specified resource amounts. The resource_list is in the following format:
 - resource_nameopvalue[,resource_nameopval,...]
 - The relation operator op must be present.



The qselect Command

- -N name
 - Restricts selection of jobs to those with a specific name.
- -p [op]priority
 - Restricts selection of jobs to those with a priority that matches the specified relationship.
- -q destination
 - Restricts selection to those jobs residing at the specified destination. The destination may be one of the following three forms:
 - queue
 - @server
 - queue@server
 - If the -q option is not specified, jobs will be selected from the default server. If the destination describes only a queue, only jobs in that queue on the default batch server will be selected. If the destination describes only a server, then jobs in all queues on that server will be selected. If the destination describes both a queue and a server, then only jobs in the named queue on the named server will be selected.



The qselect Command

- -r rerun
 - Restricts selection of jobs to those with the specified Rerunable attribute. The option argument must be a single character. The following two characters are supported by PBS: y and n.
- -s states
 - Restricts job selection to those in the specified states. The states argument is a character string which consists of any combination of the characters: E, H, Q, R, T, and W. The characters in the states argument have the following interpretation:
 - E the Exiting state.
 - H theHeldstate.
 - Q the Queued state.
 - R the Running state.
 - S the Suspended state
 - T the Transiting state.
 - W theWaiting state.



The qselect Command

– -u user_list

- Restricts selection to jobs owned by the specified user names. The syntax of the user_list is:
 - user_name[@host][,user_name[@host],...]
- Host names may be wild carded on the left end, e.g. "*.pbspro.com". User_name without a "@host" is equivalent to "user_name@*", i.e. at any host. Jobs will be selected which are owned by the listed users at the corresponding hosts.



qselect Example

- For example, say you want to list all jobs owned by user “barry” that requested more than 16 CPUs. You could use the following qselect command syntax:

```
% qselect -u barry -l ncpus.gt.16
```
- Pass the list of job identifiers directly into qstat for viewing purposes

```
% qstat -a ' qselect -u barry -l ncpus.gt.16 '
```



Working With PBS Jobs



The qalter Command

- There may come a time when you need to change an attribute on a job you have already submitted.
- Most attributes can be changed by the owner of the job while the job is still queued. However, once a job begins execution, the resource limits cannot be changed. These include:
 - cputime
 - walltime
 - number of CPUs
 - Memory
- Syntax for qalter is:
 - qalter job-resources job-list



The qalter Command

- Example

```
qalter -l walltime=20:00 -N engine 54
```



The qdel Command

- PBS provides the qdel command for deleting jobs from the system.
- Example
 - `% qdel 17`



The qhold Command

- PBS provides a pair of commands to hold and release jobs. To hold a job is to mark it as ineligible to run until the hold on the job is “released”.
- A job that has a hold is not eligible for execution.
- There are three types of holds: user, operator, and system. A user may place a user hold upon any job the user owns. An “operator”, who is a user with “operator privilege”, may place either a user or an operator hold on any job. The PBS Manager may place any hold on any job.
- Syntax of the qhold command is:
 - qhold [-h hold_list] job_identifier ...
 - hold_list characters
 - n none
 - u user
 - o operator
 - s system



The qhold Command

- If no -h option is given, the user hold will be applied to the jobs described by the job_identifier operand list.
- If the job identified by job_identifier is in the queued, held, or waiting states, then all that occurs is that the hold type is added to the job. The job is then placed into held state if it resides in an execution queue.
- If the job is in running state, then the following additional action is taken to interrupt the execution of the job.
- If checkpoint / restart is supported by the host system, requesting a hold on a running job will cause (1) the job to be checkpointed, (2) the resources assigned to the job be released, and (3) the job to be placed in the held state in the execution queue.
- If checkpoint / restart is not supported, qhold will only set the requested hold attribute. This will have no effect unless the job is rerun with the qrerun command.
- Example
 - `% qhold 54`



The qrls Command

- The qrls command releases the hold on a job.
- However, the user executing the qrls command must have the necessary privilege to release a given hold. The same rules apply for releasing holds as exist for setting a hold.
- The usage syntax of the qrls command is:
 - qrls [-h hold_list] job_identifier ...
- Example
 - `% qrls -h u 54`



The qmsg Command

- To send a message to a job is to write a message string into one or more output files of the job. Typically this is done to leave an informative message in the output of the job.
- Message can only be sent to running jobs.
- Syntax of the qmsg command is:
 - qmsg [-E][-O] message_string job_identifier
- Example
 - % qmsg -E “hello to my error (.e) file” 55
 - % qmsg -O “hello to my output (.o) file” 55
 - % qmsg “this too will go to my error (.e) file” 55



The qsig Command

- The qsig command requests that a signal be sent to executing PBS jobs.
- Syntax of the qsig command is:
 - qsig [-s signal] job_identifier
- If the -s option is specified, it declares which signal is sent to the job. The signal argument is either a signal name, e.g. SIGKILL, the signal name without the SIG prefix, e.g. KILL, or a unsigned signal number, e.g. 9. The signal name SIGNULL is allowed; the server will send the signal 0 to the job which will have no effect.



The qsig Command

- Two special signal names, "suspend" and "resume", (note, all lower case), are used to suspend and resume jobs. When suspended, a job continues to occupy system resources but is not executing and is not charged for walltime.
- Manager or operator privilege is required to suspend or resume a job.

- Example

```
% qsig -s SIGKILL 34
```

```
% qsig -s KILL 34
```

```
% qsig -s 9 34
```




The qorder Command

- PBS provides the qorder command to exchange the positions of 2 jobs in the queue or queues in which the jobs resides.
- The two jobs must be located at the same server, and both jobs must be owned by the user.
- Usage of the qorder command is:
 - `qorder job_identifier job_identifier`



The qmove Command

- PBS provides the qmove command to move jobs between different queues (even queues on different servers).
- To move a job is to remove the job from the queue in which it resides and instantiate the job in another queue.
- A job in the running state cannot be moved.
- The usage syntax of the qmove command is:
 - qmove destination job_identifier(s)
- The first operand is the new destination for
 - queue
 - @server
 - queue@server



Advance PBS Features

Coming soon...



Running Parallel Jobs

Parallel Jobs



Requesting Nodes

- The nodes resources_list item is set by the user (via the qsub command) to declare the node requirements for the job. It is a string of the form
 - -l nodes=node_spec[+node_spec...]
 - where node_spec can be any of the following: number, property[:property...], or number:property[:property...]. The node_spec may have an optional global modifier appended. This is of the form #property.



Requesting Nodes

- For example:
 - `6+3:fat+2:fat:hippi+disk#prime`
 - Where fat, hippy, disk, and prime are examples of property names assigned by the administrator in the `/var/spool/PBS/server_priv/nodesfile/nodes`
 - The above example translates as the user requesting six plain nodes plus three “fat” nodes plus two nodes that are both “fat” and “hippy” plus one “disk” node, a total of 12 nodes. Where `#prime` is appended as a global modifier, the global property, “prime” is appended by the Server to each element of the node specification.
 - It would be equivalent to
 - `6:prime+3:fat:prime+2:fat:hippy:prime+disk:prime`



Parallel Jobs and Nodes

- A user may request multiple processes per node by adding the terms `ppn=#` (for processor per node) or `cpp=#` (CPUs per process) to each node expression. For example, to request 2 VPs on each of 3 nodes and 4 VPs on 2 more nodes, the user can request
 - `-l nodes=3:ppn=2+2:ppn=4`
 - If `-lnodes=A:ppn=2+B:ppn=3` is given, then the ordering in the `PBS_NODEFILE` is A, B, A, B, B.



Running Parallel Jobs

MPI Jobs with PBS



MPI Jobs with PBS

- On a typical system, to execute a Message Passing Interface (MPI) program you would use the mpirun command. For example, here is a sample PBS script for a MPI job:

```
#!/bin/sh
#PBS -l nodes=32
#
mpirun -np 32 -machinefile $PBS_NODEFILE ./a.out
```

- Or, when using a version of MPI that is integrated with PBS:

```
#!/bin/sh
#PBS -l nodes=32
#
mpirun -np 32 ./a.out
```



Running Parallel Jobs

PVM Jobs with PBS



PVM Jobs with PBS

- On a typical system, to execute a Parallel Virtual Machine (PVM) program you would use the `pvmexec` command. For example, here is a sample PBS script for a PVM job:

```
#!/bin/sh
```

```
#PBS -l nodes=32
```

```
#
```

```
pvmexec ./a.out -inputfile datain
```



PBS Environment Variables



PBS

Environment Variables

Variable	Meaning
PBS_O_HOME	Value of HOME from submission environment.
PBS_O_LANG	Value of LANG from submission environment.
PBS_O_LOGNAME	Value of LOGNAME from submission environment
PBS_O_PATH	Value of PATH from submission environment
PBS_O_MAIL	Value of MAIL from submission environment
PBS_O_SHELL	Value of SHELL from submission environment
PBS_O_TZ	Value of TZ from submission environment
PBS_O_HOST	The host name on which the qsub command was executed.
PBS_O_QUEUE	The original queue name to which the job was submitted.



PBS

Environment Variables

Variable	Meaning
PBS_O_SYSTEM	The operating system name where qsub was executed.
PBS_O_WORKDIR	The absolute path of directory where qsub was executed.
PBS_ENVIRONMENT	Indicates if job is a batch job, or a PBS interactive job.
PBS_JOBID	The job identifier assigned to the job by the batch system.
PBS_JOBNAME	The job name supplied by the user.
PBS_NODEFILE	The filename containing a list of nodes assigned to the job.
PBS_QUEUE	The name of the queue from which the job is executed.
BEOWULF_JOB_MAP	Scyld systems only: list of node numbers separated by “.”
ENVIRONMENT	Provided for NQS migration; same as PBS_ENVIRONMENT



END