

NAME**gqueue** — SGE Script Helper**SYNOPSIS****gqueue** [options] *input-file***DESCRIPTION**

The **gqueue** utility is used to simplify the task of submitting computational jobs to the Sun GridEngine (SGE) queue environment. SGE runs parallel jobs in a special context called a parallel environment. A parallel environment controls how a multi-node job is started and stopped. Serial jobs need not run inside a parallel environment. However, in all cases a job must be submitted to SGE as a queue script: an executable shell script that contains additional commands that influence how SGE will handle the job in addition to the actual commands associated with the computational task. **gqueue** removes the burden of writing an SGE submission script each time a user wishes to run a calculation: it writes the script for the user.

gqueue creates in the user's home directory a special directory to hold all of its generated queue scripts as they execute. By default, the directory is named ".gqueue" but the default name may be changed when the utility is compiled or the user may choose an alternate directory at runtime via a command-line option. Whatever the name of this directory (or directories) it (they) should not be moved, deleted, or renamed while it (they) contain(s) an active queue script.

The queue scripts are always named using the basename of the input file the user submits. SGE will not accept queue scripts which have filenames that begin with a numerical character; if this is the case, then the basename has a "gqs_" prefix and a unique eight-character suffix. Otherwise, the script is named by appending ".gqs_" and a unique eight-character suffix to the basename. Queue scripts automatically delete themselves at the end of a successful job unless you request otherwise via a command-line option. SGE output files (with a ".o#" suffix) are also automatically removed unless the user requests otherwise.

Each of the SGE job processing options that **gqueue** handles has an associated environment variable. If the environment variable is defined, then when **gqueue** is run it automatically starts with the parameter values provided in the shell environment. This can be helpful if, for example, the user wishes to have an email delivered after completion of every job without explicitly entering an email address on the command line, or if every job on a cluster of homogeneous dual-processor nodes should be run with two processors requested.

If **gqueue** is run without an input file provided on the command line, it attempts to read the computational body of the queue script from standard input: your terminal will hang waiting for you to type the body of the script followed by an EOF (control-D) character. You can press control-C to kill **gqueue** if you mistakenly end up in this mode.

OPTIONS**--verbose/-v**

Write as much information as possible as the SGE queue script is written and submitted.

--quiet/-q

Suppress output that would be produced in verbose mode.

--help Displays a short summary of the command-line options and usage of **gqueue****--saveoutput**

When the job has finished or failed, allow the SGE output file for it to remain in the queue script directory.

--savescript

When the job has finished or failed, do not expunge the SGE queue script from the queue script directory.

- queuedir/-Q *directory***
Store the generated queue script and SGE output files in the given directory.
- email-address/-e *address***
Requests that SGE send an email message when the job either completes, is killed, or is suspended.
- nproc/-n #**
Requests this number of processors for the job. An appropriate parallel environment from the selected p.e. type will be chosen that has at least this many processors associated with it in the **gqueue** XML configuration file.
- memsize/-m #{*LABEL*}**
Request the given memory allocation for the job. Without a label, the values is assumed to be in bytes. The unit prefixes "k", "K", "m", "M", "g", and "G" are valid and by themselves imply a base unit of bytes (label of "b" or "B" explicitly). The word (8 byte value) may also be used as the base unit, with a label of "w" or "W". The lowercase forms are metric prefixes; uppercase forms are binary prefixes.
- pe-type/-t *id***
Requests a specific parallel environment type be used for this job. Each type is defined within a `<pe_list>..</pe_list>` block in the **gqueue** XML configuration file.
- pe-summary**
Do not create the script or do anything, just summarize the specified parallel environment type. If no type was provided on the command line, then all of the available types are summarized.
- priority/-p {*low,medium,high*}**
Specify a scheduling priority for the job. By default, all jobs you submit are given medium priority. Note that it is up to users to police themselves on this feature: a malicious user would submit all his/her jobs at *high* priority and get a greater percentage of the system's CPU time.
- shell/-s *shell-path***
The script created by **gqueue** should be executed as a script interpreted by the shell specified by *shell-path* rather than the default (`/bin/sh`).

RETURN VALUES

gqueue returns non-zero when an error occurs.

ENVIRONMENT

The following environment variables may be assigned values by a user and **gqueue** will use these values by default. Any value provided on the command-line will, of course, override the default from the user's environment.

GQUEUE_VERBOSE=yes

Run the program in verbose mode.

GQUEUE_SAVEOUTPUT=yes

Save the SGE output file after the script completes.

GQUEUE_SAVESCRIPT=yes

Save the SGE queue script after the script completes.

GQUEUE_EMAILADDR=address

When the job finishes/fails send an email message to the given address.

GQUEUE_QUEUEDIR=directory

Store the generated queue script and SGE output files in the given directory.

GQUEUE_NPROC=#

Request the given number of processors for the job.

GQUEUE_MEMSIZE=#{LABEL}

Request the given memory allocation for the job. Without a label, the values is assumed to be in bytes. The unit prefixes "k", "K", "m", "M", "g", and "G" are valid and by themselves imply a base unit of bytes (label of "b" or "B" explicitly). The word (8 byte value) may also be used as the base unit, with a label of "w" or "W". The lowercase forms are metric prefixes; uppercase forms are binary prefixes.

GQUEUE_PE_TYPE=id

Selects the specified parallel environment type for the job.

GQUEUE_SHELLPATH=shell-path

The path to the shell used to execute this job. Note that you **MUST** use the command line option or specify this environment variable in your environment prior to running **gqueue** if you want to use anything other than /bin/sh to execute the script!

In addition to these variables, the following will be set once **gqueue** is running; add-on script-file processing scripts will have access to these values:

GQUEUE_INPUT_FILEPATH

Path to the input file associated with the queue script being generated. For example, the Gaussian pre-merge script, gaussian.pre, reads the input file for processor count lines that it passes back to **gqueue** in its output.

GQUEUE_SCRIPT_FILEPATH

Path to the queue script file itself, so the processing script can append information, for example.

FILES

The configuration for **gqueue** is by default stored in the /etc/gqueue.d directory. In that directory should be an XML file, pe.conf, that contains the set of parallel environment types available on the cluster. The DTD for the file is:

```
<!ELEMENT pe_lists      (pe_list+)>
<!ELEMENT pe_list      (pe+)>
<!ATTLIST pe_list
  id          CDATA #IMPLIED
  implicit-single CDATA #IMPLIED
  script-base CDATA #IMPLIED>
<!ELEMENT pe>
<!ATTLIST pe
  id          CDATA #IMPLIED
  cpu-count   CDATA #IMPLIED
  round-up    CDATA #IMPLIED>
```

An example configuration file:

```
<?xml version="1.0"?>
<pe_lists>
  <pe_list id="gaussian" script-base="/usr/local/gqueue/scripts" implicit-single="yes">
    <pe id="gaussian_dual" cpu-count="2"/>
    <pe id="gaussian_quad" cpu-count="4" round-up="true"/>
    <pe id="gaussian_linda" cpu-count="18"/>
  </pe_list>
</pe_lists>
```

```

</pe_list>
<pe_list id="lam-mpi" script-base="/usr/local/gqueue/scripts">
  <pe cpu-count="18"/>
</pe_list>
</pe_lists>

```

1. The script-base attribute specifies where the pre-merge, merge, and post-merge support scripts for the <pe_list> are stored. You should keep these under the /etc/gqueue.d directory.
2. The first <pe_list> is the default that will be chosen when the user does not specify a **--pe-type** on the command line.
3. Each <pe> should have a SGE parallel environment name as its id attribute. If no id is provided, that entry is named using the parent <pe_list> id (there can thus be only one default per <pe_list>).
4. The <pe> with the largest cpu-count attribute marks the maximum CPU allocation for that parallel environment type.
5. A <pe_list> with implicit-single set to true will treat any single-processor job as a job that can run outside of a parallel environment.

SUPPORT SCRIPTS

Each parallel environment type can have a set of support scripts/executables available that are called at specific times during the run of **gqueue** and are thus given a chance to modify the queue script and/or the behavior of the utility itself.

{<pe_list> id}.pre

The .pre script/executable is run just after the queue script has been created but before a target parallel environment is selected from the <pe_list> or any SGE-specific options are written to the queue script.

{<pe_list> id}.merge

The .merge script/executable is called when the input file itself is ready to be merged into the queue script. If no .merge script/executable is provided for a <pe_list> **gqueue** simply copies the contents of the user's input file verbatim into the queue script.

{<pe_list> id}.post

The .post script/executable is run after the input file merger but before the queue script footer is written (including the commands to remove the queue script and/or output file).

AUTHORS

The **gqueue** program is written, documented, and supported by Jeffrey Frey, Network & Systems Services, University of Delaware