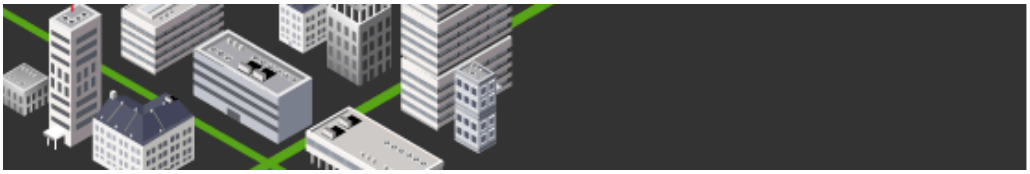
Sun Grid Engine installation on Ubuntu Server

How to install, configure and use Sun Grid Engine (SGE) for HPC

Last updated on May 18, 2016



This guide will help you set up and configure Sun Grid Engine (SGE) on Ubuntu Server 14.04 LTS.

Normally, the installation process will require your input several times, but by following this guide you will be able to perform an unattended installation which means that you can automate the setup of your cluster with a shell script. Alternatively, you can setup SGE manually by copy & pasting commands in this guide in the order that they are presented.

SGE is a task or job scheduler. You submit your typically long running tasks to a queue and the scheduler will try to run the task on one of the worker hosts when it is available.

Installation

A SGE cluster conceptually consists of a master host and one or several worker hosts. The master host can also function as a worker. Then there are also clients which submit jobs to the cluster.

Master

The commands below will perform an unattended installation. If you copy&paste them in the terminal, keep in mind that `apt-get` swallows pasted commands that follow that line.

Note that SGE will also install postfix (an SMTP server) which we will disable.

```
# Configure the master hostname for Grid Engine
echo "gridengine-master       shared/gridenginemaster string   $HOSTNAME" | sudo debconf-set-selections
echo "gridengine-master       shared/gridenginecell   string   default" | sudo debconf-set-selections
echo "gridengine-master       shared/gridengineconfig boolean false" | sudo debconf-set-selections
echo "gridengine-common       shared/gridenginemaster string   $HOSTNAME" | sudo debconf-set-selections
echo "gridengine-common       shared/gridenginecell   string   default" | sudo debconf-set-selections
echo "gridengine-common       shared/gridengineconfig boolean false" | sudo debconf-set-selections
echo "gridengine-client       shared/gridenginemaster string   $HOSTNAME" | sudo debconf-set-selections
echo "gridengine-client       shared/gridenginecell   string   default" | sudo debconf-set-selections
echo "gridengine-client       shared/gridengineconfig boolean false" | sudo debconf-set-selections
# Postfix mail server is also installed as a dependency
echo "postfix postfix/main_mailer_type        select  No configuration" | sudo debconf-set-selections

# Install Grid Engine
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y gridengine-master gridengine-client

# Set up Grid Engine
sudo -u sgeadmin /usr/share/gridengine/scripts/init_cluster /var/lib/gridengine default /var/spool/grideng
```

```
sudo service gridengine-master restart

# Disable Postfix
sudo service postfix stop
sudo update-rc.d postfix disable
```

Test that it works by running

```
$ qhost
HOSTNAME                ARCH         NCPU  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------
global                  -               -     -       -       -       -       -
```

If you see an error message like this

```
vagrant@master:~$ qhost
error: commlib error: access denied (client IP resolved to host name "localhost". This is not identical to
error: unable to send message to qmaster using port 6444 on host "master": got send error
```

it means that SGE is expecting `127.0.0.1` to resolve to `master` which is our hostname but in this case `master` resolves to `127.0.1.1` since that's what Ubuntu tends to put in `/etc/hosts`

```
vagrant@master:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 master master
```

In this case, I am going to solve this problem with

```
echo 127.0.0.1 localhost | sudo tee /etc/hosts
echo 192.168.9.10 master | sudo tee -a /etc/hosts
sudo service gridengine-master restart
```

but what it means is that you need to make sure that you have no problems resolving hostnames and IPs that you are going to use with SGE.

Worker

We need to know the master hostname before proceeding.

```
export MASTER_HOSTNAME=master
```

The following commands will perform an unattended installation on a worker host.

```
echo "gridengine-common       shared/gridenginemaster string   $MASTER_HOSTNAME" | sudo debconf-set-selecti
echo "gridengine-common       shared/gridenginecell   string  default"  | sudo debconf-set-selections
echo "gridengine-common       shared/gridengineconfig boolean false"  | sudo debconf-set-selections
echo "gridengine-client       shared/gridenginemaster string   $MASTER_HOSTNAME" | sudo debconf-set-selecti
echo "gridengine-client       shared/gridenginecell   string  default"  | sudo debconf-set-selections
echo "gridengine-client       shared/gridengineconfig boolean false"  | sudo debconf-set-selections
```

```
echo "postfix postfix/main_mailer_type        select  No configuration"  | sudo debconf-set-selections

sudo DEBIAN_FRONTEND=noninteractive apt-get install -y gridengine-exec gridengine-client

sudo service postfix stop
sudo update-rc.d postfix disable
```

Got errors about `/var/lib/gridengine/default/common/act_qmaster`?

```
echo $MASTER_HOSTNAME | sudo tee /var/lib/gridengine/default/common/act_qmaster
sudo service gridengine-exec restart
```

Test it with

```
vagrant@worker1:~$ qhost
error: denied: host "worker1" is neither submit nor admin host
```

which means that the installation was successful.

Otherwise you'd see errors about `communication error`.

(To get rid of this error, you can run `sudo qconf -ah worker1` on the master host to add this worker as an admin host. Read more in the `Hosts` section below.)

Note that `gridengine-exec` is the package to required to run SGE on a worker host. `gridengine-client` installs command line utilities like `qhost` and `qstat` that can help diagnose problems.

Need to reinstall SGE?

```
export MASTER_HOSTNAME=master
sudo rm -rf /var/lib/gridengine/
sudo apt-get remove gridengine-exec gridengine-client gridengine-common --purge -y
echo "gridengine-common       shared/gridenginemaster string   $MASTER_HOSTNAME" | sudo debconf-set-selecti
echo "gridengine-client       shared/gridenginemaster string   $MASTER_HOSTNAME" | sudo debconf-set-selecti
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y gridengine-exec gridengine-client
```

Configuration

You'll want to run these commands on the master host.

Users

Managers are like root users and are able to change SGE settings. Note that `sgeadmin` and `root` are already on the manager list.

```
# add yourself to the manager list
sudo qconf -am $USER
```

Operators are less privileged than managers and are able to add/remove workers.

```
# add yourself to the operator list (will be able to add/remove workers)
sudo qconf -ao $USER
```

Scheduler

You will probably want to adjust the scheduler configuration.

Here we are using the default settings except for `schedule_interval`. This setting specifies how often the scheduler checks for new jobs. By default, the value is 15 seconds which can be too high and cause delays if you submit jobs every second and they finish quickly.

Consult the man pages for more information.

```
# change scheduler config
cat > ./grid <<EOL
algorithm                         default
schedule_interval                 0:0:1
maxujobs                          0
queue_sort_method                 load
job_load_adjustments              np_load_avg=0.50
load_adjustment_decay_time        0:7:30
load_formula                      np_load_avg
schedd_job_info                   true
flush_submit_sec                  0
flush_finish_sec                  0
params                            none
reprioritize_interval             0:0:0
halftime                          168
usage_weight_list                 cpu=1.000000,mem=0.000000,io=0.000000
compensation_factor               5.000000
weight_user                       0.250000
weight_project                    0.250000
weight_department                 0.250000
weight_job                        0.250000
weight_tickets_functional         0
weight_tickets_share              0
share_override_tickets            TRUE
share_functional_shares           TRUE
max_functional_jobs_to_schedule   200
report_pjob_tickets               TRUE
max_pending_tasks_per_job         50
halflife_decay_list               none
policy_hierarchy                  OFS
weight_ticket                     0.500000
weight_waiting_time               0.278000
weight_deadline                   3600000.000000
weight_urgency                    0.500000
weight_priority                   0.000000
max_reservation                   0
default_duration                  INFINITY
EOL
sudo qconf -Msconf ./grid
rm ./grid
```

Queues

First, create a host list on which the jobs in the queue will run.

The name of the host list will be `allhosts` but in SGE configuration it is usually used with the `@` as a prefix: `@allhosts`.

```
# create a host list
echo -e "group_name @allhosts\nhostlist NONE" > ./grid
sudo qconf -Ahgrp ./grid
rm ./grid
```

Finally, create a queue for your jobs. There is a convention to add the `.q` suffix to your queue name. In this case, we will be creating a queue with the name `peteris.q`.

All settings have default values except for `qname`, `hostlist` and `load_thresholds`.

```
# create a queue
cat > ./grid <<EOL
qname                 peteris.q
hostlist              @allhosts
seq_no                0
load_thresholds       NONE
suspend_thresholds    NONE
nsuspend              1
suspend_interval      00:00:01
priority              0
min_cpu_interval      00:00:01
processors            UNDEFINED
qtype                 BATCH INTERACTIVE
ckpt_list             NONE
pe_list               make
rerun                 FALSE
slots                 2
tmpdir                /tmp
shell                 /bin/csh
prolog                NONE
epilog                NONE
shell_start_mode      posix_compliant
starter_method        NONE
suspend_method        NONE
resume_method         NONE
terminate_method      NONE
notify                00:00:01
owner_list            NONE
user_lists            NONE
xuser_lists           NONE
subordinate_list      NONE
complex_values        NONE
projects              NONE
xprojects             NONE
calendar              NONE
initial_state         default
s_rt                  INFINITY
h_rt                  INFINITY
s_cpu                 INFINITY
h_cpu                 INFINITY
s_fsize               INFINITY
h_fsize               INFINITY
s_data                INFINITY
h_data                INFINITY
s_stack               INFINITY
h_stack               INFINITY
s_core                INFINITY
h_core                INFINITY
s_rss                 INFINITY
h_rss                 INFINITY
s_vmem                INFINITY
h_vmem                INFINITY
```

```
EOL
sudo qconf -Aq ./grid
rm ./grid
```

Hosts

Allow a host to submit jobs to SGE.

```
# add the current host to the submit host list (will be able to do qsub)
sudo qconf -as $HOSTNAME
```

Allow a host to admin SGE, e.g., to see job statuses, etc.

```
# add to the admin host list so that we can do qstat, etc.
sudo qconf -ah $HOSTNAME
```

Add a worker

You can use the following bash script to add a worker to a queue.

```
#!/bin/bash

QUEUE=$1
HOSTNAME=$2
SLOTS=$3

# add to the execution host list
TMPFILE=/tmp/sge.hostname-$HOSTNAME
echo -e "hostname $HOSTNAME\nload_scaling NONE\ncomplex_values NONE\nuser_lists NONE\nxuser_lists NONE\npr
qconf -Ae $TMPFILE
rm $TMPFILE

# add to the all hosts list
qconf -aattr hostgroup hostlist $HOSTNAME @allhosts

# enable the host for the queue, in case it was disabled and not removed
qmod -e $QUEUE@$HOSTNAME

if [ "$SLOTS" ]; then
    qconf -aattr queue slots "[$HOSTNAME=$SLOTS]" $QUEUE
fi
```

Then use it as follows

```
$ sudo ./sge-worker-add.sh peteris.q worker1 4
root@master added "worker1" to exechost list
root@master modified "@allhosts" in host group list
Queue instance "peteris.q@worker1" is already in the specified state: enabled
root@master modified "peteris.q" in cluster queue list
```

You should now be able to see `worker1` in the output of `qhost`.

```
vagrant@master:~$ qhost
HOSTNAME                ARCH         NCPU  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------
global                  -               -     -       -       -       -       -
worker1                 -               -     -       -       -       -       -
```

But when you run `qstat -f` you may notice that `worker1` load average is `N/A` and the state is `u` which stands for unreachable.

```
vagrant@master:~$ qstat -f
queuename                      qtype resv/used/t
---------------------------------------------------------------------------------
peteris.q@worker1              BIP   0/0/4          -NA-     -NA-          u
```

To fix that, restart SGE on the worker host.

```
vagrant@worker1:~$ sudo service gridengine-exec
```

And the output of `qstat -f` should look like

```
vagrant@master:~$ qstat -f
queuename                      qtype resv/used/t
---------------------------------------------
peteris.q@worker1              BIP   0/0/4
```

Why do you need to run `sge-worker-add.sh` as `su`

`denied: "vagrant" must be manager for this oper` `-am $USER`.

Remove a worker

You can use the following bash script to remove a

Like the content?

Like the content?

Do you want to make a new friend (me)?

Enter your email and we can talk about software development or devops.

I can also send you a free PDF and ePUB/MOBI of all blog posts so that you can read them later.

First Name

E-mail

Let's be friends

POWERED BY DRIP

```
#!/bin/bash

QUEUE=$1
HOSTNAME=$2

# disable the host to avoid any jobs to be allocated to this host
qmod -d $QUEUE@$HOSTNAME

# remove it from the all hosts list
qconf -dattr hostgroup hostlist $HOSTNAME @allhosts

# remove it from the execution host list
qconf -de $HOSTNAME

# delete specific slot count for the host
qconf -purge queue slots $QUEUE@$HOSTNAME
```

Then use it as follows

```
vagrant@master:~$ sudo ./sge-worker-remove.sh peteris.q worker1
root@master changed state of "peteris.q@worker1" (disabled)
root@master modified "@allhosts" in host group list
root@master removed "worker1" from execution host list
root@master modified "peteris.q" in cluster queue list
```

Usage

Submit jobs

You can submit jobs to SGE with `qsub` which is in

Note that you need to be on a host that is allowed to submit jobs to SGE (run `sudo qconf -as $HOSTNAME` if you are not).

Let's submit a simple job that will execute the `hos`

```
$ qsub -b y hostname
Your job 1 ("hostname") has been submitted
```

It will be executed on one of the workers. In my ca

```
vagrant@worker1:~$ ls
hostname.e1  hostname.o1
vagrant@worker1:~$ cat hostname.*
worker1
```

The standard output was written to `hostname.o1` is the name of our command and `1` was our job I

You can change stdout/stderr filenames like this:

```
qsub -b y -o out.txt -e err.txt hostname
```

Both `out.txt` and `err.txt` will still be created on the worker host, so you'll typically may want to use a network share or something for them.

We can reduce the output of `qsub` to just the job number with the `-terse` flag:

```
$ qsub -b y hostname
Your job 3 ("hostname") has been submitted
$ qsub -terse -b y hostname
4
```

It will generally be useful to name your jobs with `-N` so that you can easily identify them in the queue:

```
$ qsub -b y -N this-job-has-a-name sleep 10
Your job 31 ("this-job-has-a-name") has been submitted
$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
```

Like the content?

Like the content?

Do you want to make a new friend (me)?

Enter your email and we can talk about software development or devops.

I can also send you a free PDF and ePUB/MOBI of all blog posts so that you can read them later.

First Name

E-mail

Let's be friends

POWERED BY DRIP

```
---------------------------------------------------------------------------------
peteris.q@worker1              BIP   0/1/4          0.01     lx26-amd64
     31 0.50000 this-job-h vagrant      r     02/30/2016 12:03:22     1
```

`qsub` will by default return immediately. Use `qsub -sync y` to wait until the job is completed:

```
$ date && qsub -b y sleep 10 && date
Wed Feb 30 12:07:13 UTC 2016
Your job 35 ("sleep") has been submitted
Wed Feb 30 12:07:13 UTC 2016

$ date && qsub -b y -sync y sleep 10 && date
Wed Feb 30 12:07:13 UTC 2016
Your job 36 ("sleep") has been submitted
Job 36 exited with exit code 0.
Wed Feb 30 12:07:24 UTC 2016
```

Like the content?

Sometimes you'll want a job to run after another o
`<id>,<id>`:

```
$ qsub -terse -b y -N date1 "date && sleep 10"
39
$ qsub -terse -b y -N date2 -hold_jid 39 date
40
$ cat date1*
Wed Feb 30 12:15:02 UTC 2016
$ cat date2*
Wed Feb 30 12:15:13 UTC 2016
```

Like the content?

Do you want to make a new friend (me)?

Enter your email and we can talk about software development or devops.

I can also send you a free PDF and ePUB/MOBI of all blog posts so that you can read them later.

First Name

E-mail

Let's be friends

POWERED BY DRIP

List jobs

You can generate lots of jobs with

```
for i in `seq 1 30`; do qsub -b y hostname; don
```

`qstat -f` will show you the currently running jobs:

```
$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
---------------------------------------------------------------------------------
peteris.q@worker1              BIP   0/2/4          0.01     lx26-amd64
     27 0.50000 hostname   vagrant      r     02/30/2016 11:55:26     1
     28 0.50000 hostname   vagrant      t     02/30/2016 11:55:26     1
---------------------------------------------------------------------------------
peteris.q@worker2              BIP   0/2/2          0.01     lx26-amd64
     26 0.50000 hostname   vagrant      r     02/30/2016 11:55:26     1
     29 0.50000 hostname   vagrant      t     02/30/2016 11:55:26     1
```

and `qstat -f -u \*` will also show pending jobs:

```
$ qstat -f -u \*
queuename                      qtype resv/used/tot. load_avg arch          states
---------------------------------------------------------------------------------
```

```
peteris.q@worker1              BIP   0/4/4          0.01     lx26-amd64
     20 0.50000 hostname   vagrant      r     02/30/2016 11:55:24     1
     23 0.50000 hostname   vagrant      t     02/30/2016 11:55:24     1
     24 0.50000 hostname   vagrant      t     02/30/2016 11:55:24     1
     25 0.50000 hostname   vagrant      t     02/30/2016 11:55:24     1
---------------------------------------------------------------------------
peteris.q@worker2              BIP   0/2/2          0.01     lx26-amd64
     21 0.50000 hostname   vagrant      r     02/30/2016 11:55:24     1
     22 0.50000 hostname   vagrant      t     02/30/2016 11:55:24     1

############################################################################
 - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
############################################
     26 0.50000 hostname   vagrant      qw    02
     27 0.50000 hostname   vagrant      qw    02
     28 0.50000 hostname   vagrant      qw    02/30/2016 11:55:23     1
     29 0.50000 hostname   vagrant      qw    02/30/2016 11:55:23     1
```

Like the content?

Note that the asterix `*` is needed to match all tas it with filenames in the current directory.

To see details of a job that is still in the queue, us

Like the content?

Do you want to make a new friend (me)?

Enter your email and we can talk about software development or devops.

I can also send you a free PDF and ePUB/MOBI of all blog posts so that you can read them later.

First Name

E-mail

Let's be friends

POWERED BY DRIP

```
$ qsub -terse -b y sleep 10
30
$ qstat -j 30
==============================================
job_number:                 30
exec_file:                  job_scripts/30
submission_time:            Wed Feb 30 12:00:00
owner:                      vagrant
uid:                        1000
group:                      vagrant
gid:                        1000
sge_o_home:                 /home/vagrant
sge_o_log_name:             vagrant
sge_o_path:                 /usr/local/sbin:/us                                    /usr/
sge_o_shell:                /bin/bash
sge_o_workdir:              /home/vagrant
sge_o_host:                 master
account:                    sge
mail_list:                  vagrant@master
notify:                     FALSE
job_name:                   sleep
jobshare:                   0
env_list:
job_args:                   10
script_file:                sleep
usage    1:                 cpu=00:00:00, mem=0.00000 GBs, io=0.00000, vmem=N/A, maxvmem=N/A
scheduling info:            There are no messages available
```

It is also possible to get the output as XML which will make it easier to process if you use a script or something to analyze the status of your cluster, for instance, to create a simple dashboard.

```
$ qstat -f -xml
<?xml version='1.0'?>
<job_info  xmlns:xsd="http://gridengine.sunsource.net/source/browse/*checkout*/gridengine/source/dist/util
  <queue_info>
    <Queue-List>
      <name>peteris.q@worker1</name>
```

```
        <qtype>BIP</qtype>
        <slots_used>0</slots_used>
        <slots_resv>0</slots_resv>
        <slots_total>4</slots_total>
        <arch>lx26-amd64</arch>
      </Queue-List>
      <Queue-List>
        <name>peteris.q@worker2</name>
        <qtype>BIP</qtype>
        <slots_used>0</slots_used>
        <slots_resv>0</slots_resv>
        <slots_total>2</slots_total>
        <arch>lx26-amd64</arch>
      </Queue-List>
    </queue_info>
    <job_info>
    </job_info>
</job_info>
```

Like the content?

Canceling jobs

Use `qdel`.

```
$ qsub -terse -b y sleep 1000
32
$ qdel 32
vagrant has registered the job 32 for deletion
```

Like the content?

Do you want to make a new friend (me)?

Enter your email and we can talk about software development or devops.

I can also send you a free PDF and ePUB/MOBI of all blog posts so that you can read them later.

First Name

E-mail

Let's be friends

POWERED BY DRIP

Restart SGE

If nothing is working, try restarting SGE.

```
sudo service gridengine-master restart
sudo service gridengine-exec restart
```

Vagrantfile

You can use the following `Vagrantfile` that will spin up a master node and two worker nodes for your experiments.

```
Vagrant.configure("2") do |config|
  # Ubuntu 14.04 LTS x64 official cloud image
  config.vm.box = "ubuntu/trusty64"

  # VirtualBox, common settings
  config.vm.provider "virtualbox" do |vb|
    vb.memory = 256
    vb.cpus = 1
    vb.customize ["modifyvm", :id, "--natdnshostresolver1", "on"] # fixes slow dns lookups
  end

  config.vm.define "master" do |srv|
    srv.vm.hostname = "master"
    srv.vm.network :private_network, ip: "192.168.9.10"
    srv.vm.provider "virtualbox" do |vb| vb.name = "SGE-Master"; end
  end
```

```
    config.vm.define "worker1" do |srv|
      srv.vm.hostname = "worker1"
      srv.vm.network :private_network, ip: "192.168.9.11"
      srv.vm.provider "virtualbox" do |vb| vb.name = "SGE-Worker1"; end
    end

    config.vm.define "worker2" do |srv|
      srv.vm.hostname = "worker2"
      srv.vm.network :private_network, ip: "192.168.9.12"
      srv.vm.provider "virtualbox" do |vb| vb.name = "SGE-Worker2"; end
    end
  end
```

Then

```
vagrant up
vagrant ssh master
vagrant ssh worker1
vagrant ssh worker2
vagrant destroy -f
```

Make sure that you change `/etc/hosts` to the fol

```
127.0.0.1 localhost
192.168.9.10 master
192.168.9.11 worker1
192.168.9.12 worker2
```

Additional links

- qconf man page
- qsub man page
- SGE QuickStart
- NYU HPC Tutorial

Final remarks

I hope you find this guide useful as it took me a long while to discover how to automate and debug everything.

The man pages are extensive but they serve as a reference rather and a step by step tutorial.

Like the content?

Like the content?

Do you want to make a new friend (me)?

Enter your email and we can talk about software development or devops.

I can also send you a free PDF and ePUB/MOBI of all blog posts so that you can read them later.

First Name

E-mail

Let's be friends

POWERED BY DRIP

Top posts

Recent posts

Like the content?

1 Comment peteris.rocks Login

Recommend Share

Join the discussion…

Chris Barbour • 2 months ago
This is a hugely helpful guide. One quick s
SSMTP instead. Postfix is being installed b
dependency of Grid Engine.

SSMTP satisfies this dependency in a sim
necessary.

• Reply • Share ›

Subscribe Add Disqus to your site Add Disqus Add

Like the content?

Do you want to make a new friend (me)?

Enter your email and we can talk about software development or devops.

I can also send you a free PDF and ePUB/MOBI of all blog posts so that you can read them later.

First Name

E-mail

Let's be friends

POWERED BY DRIP

© 2014, 2015, 2016 Pēteris Ņikiforovs RSS
All code written by me on this blog is licensed und
Built with VSCode, node.js, gulp, jade, less, markdown, coffeescript, highlight.js