

Opensquad (<http://www.opensquad.com>)

SUN GRIDENGINE INSTALLATION ON UBUNTU 14.04

Posted on 16 novembre 2015 (<http://www.opensquad.com/blog/sun-gridengine-installation-on-ubuntu-14-04/>) by abrignon
(<http://www.opensquad.com/blog/author/abrignon/>)

A client asked me to write a quick documentation on Howto deploy a computing grid on a Ubuntu distro. I've never worked on grid solutions, so the challenge was interesting (while a bit painfull reading the various tutos laying around).

Theory of operations

The user gadmin will be the job submitter on master node and the job executor on remote nodes.

For convenience we will use a shared home directory to ensure that code is the same on every node and will gather the output on the master node (assuming code outputs into its home directory).

SSH connection :

The user gadmin on the master should connect to worker nodes without password via SSH.

So, for NFS to work correctly we must ensure that the user gadmin will have the UID/GID on every node of the grid.

Preparing the master node

Create the user with fixed UID/GID

```
groupadd -g 600 gsadmin
```

```
useradd \  
  --gid gsadmin \  
  -d /home/gsadmin \  
  -m -N --shell /bin/bash \  
gsadmin
```

Prepare the SSH passwordless connection by generating a ssh key pair and setting a password for the gsadmin user.

```
passwd gsadmin  
su - gsadmin  
# generate and copy ID locally  
ssh-keygen  
ssh-copy-id gsadmin@master # enter the password  
# check the success with :  
ssh gsadmin@master
```

Now install the NFS Server package and configure it with the home directory of the new user "gsadmin" for sharing.

```
apt-get install nfs-kernel-server
```

At this point NFS kernel server has been successfully installed, now we will make use of few commands as shown below to configure the default exports file and then restart nfs services.

```
echo "/home/gsadmin *(rw, sync, no_subtree_check)" >> /etc/exports
exportfs -a
service nfs-kernel-server restart
```

Install relevant packages

```
# Install software suite on master
apt-get install gridengine-master
apt-get install gridengine-client
```

start the daemon and fix any problem that arises by checking the output

```
/etc/init.d/gridengine-master start  
tail -f /var/spool/gridengine/qmaster/messages
```

Configure the permissions for gsadmin user and create a queue and a hostlist and link them together.
Every configuration in gridEngine is done via qconf.
We begin by allowing the root user to interact with the configuration.

```
sudo -u sgeadmin qconf -am root
sudo -u sgeadmin qconf -au root users
# Now we can keep working as root
qconf -am gsadmin
qconf -au gsadmin users
# we configure the hosts and the queue with 2 slots each
# 2 correspond to the number of CPU on the worker node
qconf -as master
# create a hostlist named allhosts
qconf -ahgrp @allhosts # save and exit
# add the master into the list
qconf -aattr hostgroup hostlist master @allhosts
# create a work queue main.q and link it to the @allhosts group
qconf -aq main.q # save and exit
qconf -aattr queue hostlist @allhosts main.q
# replace 2 with your worker CPU number
# and set this number to the work slots available
qconf -aattr queue slots "2" main.q
```

Now that the master works, we can export its data to the rest of the grid via NFS and prepare the user home directory to be mounted on remote nodes.

Of course, in real world, you should increase NFS security.

```
apt-get install nfs-kernel-server
echo "/home/gsadmin *(rw,sync,no_subtree_check)" >> /etc/exports
# export the cell configuration in read-only mode
echo "/var/lib/gridengine/default *(ro,sync,no_subtree_check)" >> /etc/exports
exportfs -a
service nfs-kernel-server restart
```

Worker node configuration

First of all, we must check the proper DNS configuration by issuing a ping master. If it fails, modify /etc/hosts by adding the master IP to the /etc/hosts.

Create the gsadmin user with the same characteristics than the one on the master

```
groupadd -g 600 gsadmin
useradd \
--gid gsadmin \
-d /home/gsadmin \
-m -N --shell /bin/bash \
gsadmin
apt-get install nfs-common
```

Configure gsdmin@worker shared home dir and ensure automounting upon reboot.

Add this line to the /etc/fstab file

```
master:/home/gsadmin /home/gsadmin nfs auto
```

and issue a **mount -a** to see if it's correctly mounted by entering "mount" at the prompt. You should see the /home/gsadmin listed as NFS.

Install and configure the grid software Executor

```
apt-get install gridengine-exec
# configure with the same information as the master (postix too)
# Anyway, we will reuse the cell configuration exported by the master via NFS
echo 'master:/var/lib/gridengine/default /var/lib/gridengine/default nfs auto,ro' >> /etc/fstab
mount -a
# check if all goes well
```

Add the worker in the master configuration (!! take care, this occurs on the {{master}} server !)

```
qconf -ah worker
qconf -aattr hostgroup hostlist worker @allhosts
```

Now the master is aware of the new worker node. We can start the executor in the node

```
/etc/init.d/gridengine-exec start
```

The new server should come up with information on mem/cpu/... on the master.

```
root@master:~# qhost -q
HOSTNAME ARCH NCPU LOAD MEMTOT MEMUSE SWAPTO SWAPUS
-----
global - - - - -
worker lx26-amd64 2 0.01 2.0G 99.7M 0.0 0.0
main.q BIP 0/0/2
```

Launching a job

We will use a simple bash script and check that it runs correctly.

We must use the gsadmin user to submit jobs to worker nodes so become user gsadmin (su – gsadmin) to edit and launch jobs.

Edit hello_world.sh and paste these lines in


```
#!/bin/bash
echo "Start Date : " `date`
# let's sleep a bit in order to show up as a running task
sleep 100
# issue a few information
echo "Working dir : " `pwd`
echo "Host : " `hostname`
echo "Stop Date : " `date`
```

Then issue a `chmod+x hello_world.sh`

```
gsadmin@master:~$ qsub hello_world.sh
Your job 3 ("hello_world.sh") has been submitted
gsadmin@master:~$ qstat
```

You will find the script's results the files named according to script name and Job ID.

- `scriptName.e+JobID` : contains Error (stderr) output
- `scriptName.o+JobID` : contains stdout output

Usefull links

- <http://informatics.malariagen.net/2011/06/01/gridengine-the-ubuntu-debian-way/>
(<http://informatics.malariagen.net/2011/06/01/gridengine-the-ubuntu-debian-way/>)

- Manual (PDF) (<http://www.bioteam.net/wp-content/uploads/2009/09/03-SGE-Admin-Config.pdf>)

Tagged grid computing (<http://www.opensquad.com/blog/tag/grid-computing/>), linux (<http://www.opensquad.com/blog/tag/linux/>), ubuntu (<http://www.opensquad.com/blog/tag/ubuntu/>)

← Linux and Acer Aspire 5683WLMI – Fedora Core 6
(<http://www.opensquad.com/blog/linux-and-acer-aspire-5683wlmi-fedora-core-6/>)

Testing IMAP and SMTP Authentication with Telnet →
(<http://www.opensquad.com/blog/testing-imap-and-smtp-authentication-with-telnet/>)

Leave a Comment

Message

Name

Email

Enregistrer mon nom, mon e-mail et mon site dans le navigateur pour mon prochain commentaire.

SUBMIT COMMENT

Articles récents

Ansible: processing data structures with Jinja2 (<http://www.opensquad.com/blog/ansible-processing-data-structures-with-jinja2/>)

Testing IMAP and SMTP Authentication with Telnet (<http://www.opensquad.com/blog/testing-imap-and-smtp-authentication-with-telnet/>)

Sun GridEngine Installation on Ubuntu 14.04 (<http://www.opensquad.com/blog/sun-gridengine-installation-on-ubuntu-14-04/>)

Linux and Acer Aspire 5683WLMi – Fedora Core 6 (<http://www.opensquad.com/blog/linux-and-acer-aspire-5683wlmi-fedora-core-6/>)

Variantes redhat (<http://www.opensquad.com/blog/variantes-redhat/>)

Étiquettes

[acer \(http://www.opensquad.com/blog/tag/acer/\)](http://www.opensquad.com/blog/tag/acer/)[ansible \(http://www.opensquad.com/blog/tag/ansible/\)](http://www.opensquad.com/blog/tag/ansible/)[devops \(http://www.opensquad.com/blog/tag/devops/\)](http://www.opensquad.com/blog/tag/devops/)[fedora \(http://www.opensquad.com/blog/tag/fedora/\)](http://www.opensquad.com/blog/tag/fedora/)[grid computing \(http://www.opensquad.com/blog/tag/grid-computing/\)](http://www.opensquad.com/blog/tag/grid-computing/)[imap \(http://www.opensquad.com/blog/tag/imap/\)](http://www.opensquad.com/blog/tag/imap/)

[linux \(http://www.opensquad.com/blog/tag/linux/\)](http://www.opensquad.com/blog/tag/linux/)[postgresql \(http://www.opensquad.com/blog/tag/postgresql/\)](http://www.opensquad.com/blog/tag/postgresql/)[smtp \(http://www.opensquad.com/blog/tag/smtp/\)](http://www.opensquad.com/blog/tag/smtp/)[spip \(http://www.opensquad.com/blog/tag/spip/\)](http://www.opensquad.com/blog/tag/spip/)[telnet \(http://www.opensquad.com/blog/tag/telnet/\)](http://www.opensquad.com/blog/tag/telnet/)[ubuntu \(http://www.opensquad.com/blog/tag/ubuntu/\)](http://www.opensquad.com/blog/tag/ubuntu/)

Archives

[mars 2017 \(http://www.opensquad.com/blog/2017/03/\)](http://www.opensquad.com/blog/2017/03/)[décembre 2015 \(http://www.opensquad.com/blog/2015/12/\)](http://www.opensquad.com/blog/2015/12/)[novembre 2015 \(http://www.opensquad.com/blog/2015/11/\)](http://www.opensquad.com/blog/2015/11/)[juillet 2006 \(http://www.opensquad.com/blog/2006/07/\)](http://www.opensquad.com/blog/2006/07/)[novembre 2005 \(http://www.opensquad.com/blog/2005/11/\)](http://www.opensquad.com/blog/2005/11/)[mars 2005 \(http://www.opensquad.com/blog/2005/03/\)](http://www.opensquad.com/blog/2005/03/)

Catégories

[Non classé \(http://www.opensquad.com/blog/category/non-classe/\)](http://www.opensquad.com/blog/category/non-classe/)

[Facebook \(\)](#) [Twitter \(\)](#) [Google+ \(\)](#) [Linkedin \(\)](#) [Behance \(\)](#) [Dribbble \(\)](#)

Copyright 2016 © **Opensquad**, All Rights Reserved

Cuda Free One page WordPress Theme (<http://www.codexcoder.com/products/cuda-free-one-page-wordpress-theme/>) Designed and Developed by CodexCoder (<http://www.codexcoder.com>) | Powered By WordPress (<http://wordpress.org>)