



PARALLEL DISTRIBUTED AND COMPUTING

NAME:YASHRAJ AGARWAL
REGISTRATION NUMBER:18BCI0183



Introduction

To understand about the distributed computing we will use AWS. We create two node cluster. We use starcluster for this purpose. Using NFS and sshing into slave node will helps us to gain this.

We will be running a sample hello world program to understand this.

REQUIREMENT FOR STARCLUSTER

- Firstly for pip-installing 'Starcluster', we require Python 2 .So make sure you have that first, before proceeding.
- The default cluster is a cluster of 2 instances of type m1.medium, which is characterized by a 32-bit or 64-bit processor, 1 Intel dual-core processor, 3.75GB or RAM, and 410 GB of disk storage (for OS + data). This should be enough for executing an MPI program.



Creating a Keypair

Command:

```
starcluster createkey mykey -o ~/.ssh/mykey.rsa
```

The above command will create a keypair called mykey on Amazon EC2 and save the private key to ~/.ssh/mykey.rsa(Can't display the execution of this command, due to privacy reasons.)

Setting up Starcluster:

```

>>> Using default cluster template: smallcluster
>>> Validating cluster template settings...
>>> Cluster template settings are valid
>>> Starting cluster...
>>> Launching a 2-node cluster...
>>> Creating security group sg-mycluster...
ReservationID=ba838658a9c898745
>>> Waiting for instances to propagate...
2/2 |||||||||||||||||||||||||||||||||||||||||||| 100%
>>> Waiting for cluster to come up... (updating every 30s)
/home/karthik/.local/lib/python2.7/site-packages/boto/ec2/connection.py:1399: UserWarning: The 'launch.group-id' filter now requires a security group id (sg-*) and no longer supports filtering by group name. Please update your filters accordingly.
  UserWarning)
>>> Waiting for all nodes to be in a 'running' state...
2/2 |||||||||||||||||||||||||||||||||||||||||||| 100%
>>> Waiting for SSH to come up on all nodes...
2/2 |||||||||||||||||||||||||||||||||||||||||||| 100%
>>> Waiting for cluster to come up took 1.687 mins
>>> The master node is ec2-52-284-4-241.compute-1.amazonaws.com
>>> Configuring cluster...
>>> Running plugin starcluster.clustersetup.DefaultClusterSetup
>>> Configuring hostnames...
2/2 |||||||||||||||||||||||||||||||||||||||||||| 100%
>>> Creating cluster user: sgadmdm (uid: 1001, gid: 1001)
2/2 |||||||||||||||||||||||||||||||||||||||||||| 100%
>>> Configuring scratch space for user(s): sgadmdm
2/2 |||||||||||||||||||||||||||||||||||||||||||| 100%
>>> Configuring /etc/hosts on each node
2/2 |||||||||||||||||||||||||||||||||||||||||||| 100%
>>> Starting NFS server on master.
>>> Configuring NFS exports path(s):
/home
>>> Mounting all NFS export path(s) on 1 worker node(s)
1/1 |||||||||||||||||||||||||||||||||||||||||||| 100%
>>> Setting up NFS took 0.356 mins
>>> Configuring passwordless ssh for root
>>> Configuring passwordless ssh for sgadmdm
>>> Running plugin starcluster.plugins.sge.SGEPlugin
>>> Configuring SGE...
>>> Configuring NFS exports path(s):
/oph/age6
>>> Mounting all NFS export path(s) on 1 worker node(s)

```

```
>>> Mounting all NFS export path(s) on 1 worker node(s)
1/1 |=====| 100%
>>> Setting up NFS took 0.263 mins
>>> Installing Sun Grid Engine...
1/1 |=====| 100%
>>> Creating SGE parallel environment 'orte'
2/2 |=====| 100%
>>> Adding parallel environment 'orte' to queue 'all.q'
>>> Configuring cluster took 2.392 mins
>>> Starting cluster took 4.196 mins
```

The cluster is now ready to use. To login to the master node as root, run:

```
$ sstercluster sstercluster mycluster
```

If you're having issues with the cluster you can reboot the instances and completely reconfigure the cluster from scratch using:

```
$ sstercluster restart mycluster
```

When you're finished using the cluster and wish to terminate it and stop paying for service:

```
$ sstercluster terminate mycluster
```

Alternatively, if the cluster uses EBS instances, you can use the 'stop' command to shutdown all nodes and put them into a 'stopped' state preserving the EBS volumes backing the nodes:

```
$ sstercluster stop mycluster
```

WARNING: Any data stored in ephemeral storage (usually /dev) will be lost!

You can activate a 'stopped' cluster by passing the -x option to the 'start' command:

```
$ sstercluster start -x mycluster
```

This will start all 'stopped' nodes and reconfigure the cluster.

Connecting to Master Code

```
karthi@DESKTOP-1FFJPM6:~$ starcluster sshmaster mycluster
/home/karthik/.local/lib/python2.7/site-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptograph
y, and will be removed in a future release.
  from cryptography.hazmat.backends import default_backend
StarCluster - (http://star.mit.edu/cluster) (v. 0.90.0)
Software Tools for Academics and Researchers (STAR)
Please submit bug reports to starcluster@mit.edu

The authenticity of host 'ec2-52-204-4-241.compute-1.amazonaws.com (52.204.4.241)' can't be established.
ECDSA key fingerprint is SHA256:NDY/TC+22gM6G8859umHMWpvc8B0KtK8Z17TuET1s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-52-204-4-241.compute-1.amazonaws.com,52.204.4.241' (ECDSA) to the list of known hosts.

StarCluster
StarCluster Ubuntu 13.04 AMI
Software Tools for Academics and Researchers (STAR)
Homepage: http://star.mit.edu/cluster
Documentation: http://star.mit.edu/cluster/docs/latest
Code: https://github.com/jtrilay/StarCluster
Mailing list: http://star.mit.edu/cluster/maillinglist.html

This AMI Contains:
* Open Grid Scheduler (OGS - formerly SGE) queuing system
* Condor workload management system
* OpenMPI compiled with Open Grid Scheduler support
* OpenBLAS - Highly optimized Basic Linear Algebra Routines
* NumPy/SciPy linked against OpenBLAS
* Pandas - Data Analysis library
* IPython 1.1.0 with parallel and notebook support
* Julia 0.3pre
* and more! (use 'dpkg -l' to show all installed packages)
```

Open Grid Scheduler/Condor cheat sheet:

- * `qstat/condor_q` - show status of batch jobs
- * `qhost/condor_status` - show status of hosts, queues, and jobs
- * `qsub/condor_submit` - submit batch jobs (e.g. `qsub -cwd ./job.sh`)
- * `qdel/condor_rm` - delete batch jobs (e.g. `qdel 7`)
- * `qconf` - configure Open Grid Scheduler system

Current System Stats:

System load:	0.04	Processes:	92
Usage of /:	34.6% of 7.84GB	Users logged in:	0
Memory usage:	7%	IP address for eth0:	172.31.46.87



Verify 2 cluster nodes within /etc/hosts

```
root@master:~# cat /etc/hosts
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
172.31.46.87 master
172.31.36.141 node001
```


Mount /home in NFS Shared

```
root@master:~# ssh node001 mount
/dev/xvda1 on / type ext4 (rw)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/cgroup type tmpfs (rw)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
none on /run/user type tmpfs (rw,noexec,nosuid,nodev,size=104857600,mode=0755)
rpc_pipefs on /run/rpc_pipefs type rpc_pipefs (rw)
/dev/xvdaa on /mnt type ext3 (rw,_netdev)
master:/home on /home type nfs (rw,vers=4,addr=172.31.46.87,clientaddr=172.31.36.141)
master:/opt/sge6 on /opt/sge6 type nfs (rw,vers=4,addr=172.31.46.87,clientaddr=172.31.36.141)
```

Logging into sgeadmin from Master node

```
root@master:~# ssh node001 hostname  
node001  
root@master:~# su - sgeadmin  
sgeadmin@master:~$ ssh node001 hostname  
node001
```

Importing file created through vi editor to
cluster(using put command already done

```
starcluster put mycluster helloworldmpi.c /home/sgeadmin
```

before)

```
sgeadmin@master:~$ vi helloworldmpi.c
sgeadmin@master:~$ mpicc helloworldmpi.c -o hellompi
sgeadmin@master:~$ mpirun -n 2 -host master,node001 ./hellompi
master: hello world from process 0 of 1
node001: hello world from process 0 of 1
```

Terminating the cluster

```
karthik@DESKTOP-1FF3PQG:~$ starcluster terminate mycluster
/home/karthik/.local/lib/python2.7/site-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in a future release.
  from cryptography.hazmat.backends import default_backend
StarCluster - (http://star.mit.edu/cluster) (v. 0.95.6)
Software Tools for Academics and Researchers (STAR)
Please submit bug reports to starcluster@mit.edu

Terminate EBS cluster mycluster (y/n)? y
>>> Running plugin starcluster.plugins.sgs.SGSEPlugin
>>> Running plugin starcluster.clustersetup.DefaultClusterSetup
>>> Terminating node: master (i-0a937d7f440f62a31)
>>> Terminating node: node001 (i-05f67283537d0d3f)
/home/karthik/.local/lib/python2.7/site-packages/boto/ec2/connection.py:1339: UserWarning: The 'launch_group_id' filter now requires a security group id (sg-*) and no longer supports filtering by group name. Please update your filters accordingly.
  UserWarning)
>>> Waiting for cluster to terminate...
>>> Removing security group: sg-mycluster
karthik@DESKTOP-1FF3PQG:~$
```