Appendices
10.1 Hardware and Software Configuration - Grid Prototype Implementation

Hardware Configuration

Grid Master Node (Job Submission Node)

Host name: grid1.aes.edu
IP Address: 10.0.5.31
Processor: Intel Dual Core 2.6 GHz
RAM: 1024 MB
Storage Capacity: 80 GB
Network Ethernet card speed: 100 MBPS

Grid Worker Nodes (Processor Node/Worker Node for Job Submission)

Host name: grid2.aes.edu
IP Address: 10.0.5.32
Processor: Intel Dual Core 2.6 GHz
RAM: 512 MB
Storage Capacity: 80 GB
Network Ethernet card speed: 100 MBPS
IP Address: 10.0.5.32

Software Configurations

• Operating Systems used:
  – Windows XP Service Pack 2 (Job submission node)
  – Ubuntu Desktop OS Version 9.10
  – Ubuntu Desktop OS Version 9.04

• Grid Middle Ware Used:
  – Globus Toolkit Version 4.2.1

• Grid Monitoring:
  – WebMDS (Web-Browser Based Monitoring and Discovery Service)
  – Ganglia
  – Tomcat 6 (For WebMDS)

• System Development Tools:
  – IBM Rational Software Architect 7.0
  – Java SE 6.0
  – Eclipse Helios IDE for development of Grid User Interface and Scheduler component

• Mobile Agent Based Architecture Provider
  – JADE 3.6.1 Toolkit
10.2 Configuration of Grid Using Globus Toolkit 4.2

Pre-requisite software required for Globus Toolkit grid nodes installation

g++
gcc
tar
sed
make
Java (JDK 1.6 or higher)

Globus Installation Folder

If space is available in /usr/local then
mkdir /usr/local/globus

As there wasn’t enough space in / directory, we created /home/globus.
mkdir /home/globus
chown globus:globus /usr/local/globus

Contents of login file (~/.bashrc)

# Globus, java, ant, junit, ssl settings
export JAVA_HOME=/usr/lib/jdk1.6.0_16
export ANT_HOME=/usr/local/apache-ant-1.8.2
# export COG_INSTALL_PATH=/usr/local/cog-4_1_5
# export JUNIT_INSTALL_PATH=/usr/bin/junit
export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:/usr/bin:/bin:/usr/sbin:$PATH
export GLOBUS_LOCATION=/usr/local/globus
export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib
export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$GLOBUS_LOCATION/bin:$PATH
# $COG_INSTALL_PATH/bin:$PATH
# $GLOBUS_LOCATION/etc/globus-user-env.sh

As root login, go to desktop folder containing below .gz file and
tar -xvzf gt4.2.1-all-source-installer.tar.gz

chown -R globus:globus gt4.2.1-all-source-installer

globus@grid2-desktop:/home/grid2/Desktop/GridSoftware/gt4.2.1-all-source-installer$ 
./configure --prefix=/usr/local/globus
make | tee installer.log
make install
Appendices

Setup of Grid Security Infrastructure (GSI)

Setup of Simple CA (Certification Authority) on machine Grid2 desktop

The distribution package built for this CA is stored in
/home/globushome/.globus/simpleCA/globus_simple_ca_c5c448ae_setup-0.20.tar.gz

globus@grid2-desktop:/home/grid2/Desktop/GridSoftware/GridPrerequisiteSoftware$ $GLOBUS_LOCATION/sbin/gpt-build globus_simple_ca_c5c448ae_setup-0.20.tar.gz

gcc32dbg
gpt-build ===> CHECKING BUILD DEPENDENCIES FOR
globus_simple_ca_c5c448ae_setup

gpt-build ===> Changing to
/home/grid2/Desktop/GridSoftware/GridPrerequisiteSoftware/BUILD/globus_simple_ca_c5c448ae_setup-0.20/
gpt-build ===> BUILDING globus_simple_ca_c5c448ae_setup

gpt-build ===> Changing to
/home/grid2/Desktop/GridSoftware/GridPrerequisiteSoftware/BUILD

gpt-build ===> REMOVING empty package globus_simple_ca_c5c448ae_setup-noflavor-pgm_static

globus@grid2-desktop:/home/grid2/Desktop/GridSoftware/GridPrerequisiteSoftware$

root@grid2-desktop:/usr/local/globus/setup/globus_simple_ca_c5c448ae_setup# ./setup-gsi -default
setup-gsi: Configuring GSI security
Installing Globus CA certificate into trusted CA certificate directory...
Installing Globus CA signing policy into trusted CA certificate directory...
setup-gsi: Complete

Obtain host certificate for grid2 desktop

root@grid2-desktop:/usr/local/globus/setup/globus_simple_ca_c5c448ae_setup# /setup-gsi-cert-request -host grid2-desktop

The hostname grid2-desktop does not appear to be fully qualified.
Do you wish to continue? [n] y
Generating a 1024 bit RSA private key
writing new private key to '/etc/grid-security/hostkey.pem'
    
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
A private host key and a certificate request has been generated with the subject:
/O=Grid/OU=GlobusTest/OU=simpleCA-grid1-desktop/CN=host/grid2-desktop

The private key is stored in /etc/grid-security/hostkey.pem
The request is stored in /etc/grid-security/hostcert_request.pem

Please e-mail the request to the Globus Simple CA adityabpatel@yahoo.com
You may use a command similar to the following:
cat /etc/grid-security/hostcert_request.pem | mail adityabpatel@yahoo.com

Your certificate will be mailed to you within two working days.
If you receive no response, contact Globus Simple CA at adityabpatel@yahoo.com
root@grid2-desktop:/usr/local/globus/setup/globus_simple_ca_c5c448ae_setup#

************************************************************************
Then, copy the hostcert_request.pem file to machine having SIMPLE CA (grid1-desktop)
Login on grid1-desktop as user: globus (Simple CA owner)
grid-ca-sign -inhostcert_request.pem -out hostsigned.pem
Copy the hostsigned.pem to grid2-desktop machine

On grid2-desktop machine as root:
As root move the signed host certificate to /etc/grid-security/hostcert.pem.
The certificate should be owned by root(rw as root) and be read-only for other users.

Setting Up Grid Container Security

login as root
root@grid2-desktop:~# cd /etc/grid-security
root@elephant:/etc/grid-security# cp hostcert.pem containercert.pem
root@elephant:/etc/grid-security# cp hostkey.pem containerkey.pem
root@elephant:/etc/grid-security# chown globus:globus container*.pem
root@elephant:/etc/grid-security# ls -l *.pem
-rw-r--r-- 1 globus globus 2724 2008-06-16 14:26 containercert.pem
-r-------- 1 globus globus 887 2008-06-16 14:26 containerkey.pem
-rw-r--r-- 1 root root 2724 2008-06-16 14:26 hostcert.pem
-rw-r--r-- 1 root root 1404 2008-06-16 14:26 hostcert_request.pem
-r-------- 1 root root 887 2008-06-16 14:26 hostkey.pem
Setting up GRAM4

First we have to setup sudo so the globus user can start jobs as a different user. For reference, you can see the System Administrator's Guide.

root@elephant:~# visudo
root@elephant:~# cat /etc/sudoers
Runas_Alias GLOBUSUSERS = ALL, !root;
globus ALL=(GLOBUSUSERS) NOPASSWD: /sandbox/globus/globus-4.2.1/libexec/globus-gridmap-and-execute
-g /etc/grid-security/grid-mapfile /sandbox/globus/globus-4.2.1/libexec/globus-job-manager-script.pl *
globus ALL=(GLOBUSUSERS) NOPASSWD: /sandbox/globus/globus-4.2.1/libexec/globus-gridmap-and-execute
-g /etc/grid-security/grid-mapfile /sandbox/globus/globus-4.2.1/libexec/globus-gram-local-proxy-tool *

Configuration of GRAM2 on grid2 desktop

sudo apt-get install xinetd

As per the instruction given at URL:
http://www.globus.org/toolkit/docs/4.2/4.2.1/execution/gram2/admin/#gram2-admin-starting

You will need to create a file named /etc/grid-security/grid-mapfile which consists of single line entries listing a certificate subject and a username, like this:
"/O=Grid/OU=GlobusTest/OU=simpleCA-grid1-desktop/CN=bacon" bacon
"/O=Grid/OU=GlobusTest/OU=simpleCA-grid1-desktop/CN=aditya" aditya

You can check your subject name using grid-cert-info -subject.

Request a User Certificate

As your normal user account (i.e. bacon, not globus), run:
grid-cert-request

After you enter a passphrase, this creates

- $USER/.globus/usercert.pem (empty)
- $USER/.globus/userkey.pem
- $USER/.globus/usercert_request.pem

Your private key is stored in /home/bacon/.globus/userkey.pem
Your request is stored in /home/bacon/.globus/usercert_request.pem
As simple CA owner (globus) (on grid2-desktop machine)
grid-ca-sign -in usertcert_request.pem -out signed.pem

On Grid1-desktop machine
-rw-r--r-- 1 nobody nogroup 2648 2011-03-31 12:12 04.pem
root@grid1-desktop:~/Desktop/GridSoftware# chown bacon:bacon 04.pem

Verify the Simple CA certificate installation

To verify that the SimpleCA certificate is installed in /etc/grid-security/certificates and that your certificate is in place with the correct permissions, run:
bacon$ grid-proxy-init -debug -verify

After entering your passphrase, successful output will look like:

    [bacon@$] grid-proxy-init -debug -verify
User Cert File: /home/bacon/.globus/usercert.pem
User Key File: /home/bacon/.globus/userkey.pem

Trusted CA Cert Dir: /etc/grid-security/certificates
Output File: /tmp/x509up_u1002
Your identity: /O=Grid/OU=GlobusTest/OU=simpleCA-grid1-desktop/CN=bacon
Enter GRID pass phrase for this identity: bacon1978 or globus
Creating proxy .....++++++++++++
............++++++++++++
Done
Proxy Verify OK
Your proxy is valid until: Tue Feb 8 02:48:44 2011

Make the host credentials accessible by the container

As root, run:
root# cd /etc/grid-security
root# cp hostkey.pem containerkey.pem
root# cp hostcert.pem containercert.pem
root# chown globus.globus containerkey.pem containercert.pem
root# chown globus.globus containerkey.pem containercert.pem

At this point the certificates in /etc/grid-security should look something like:
root# ls -l *.pem
-rw-r--r-- 1 globus globus 1785 Oct 14 14:47 containercert.pem
-r-------- 1 globus globus  887 Oct 14 14:47 containerkey.pem
-rw-r--r-- 1 root root 1785 Oct 14 14:42 hostcert.pem
-r-------- 1 root root  887 Sep 29 09:59 hostkey.pem
Add authorization

Add authorizations for users: Create /etc/grid-security/grid-mapfile as root. You need two pieces of information:

- the subject name of a user
- the account name it should map to.

"/O=Grid/OU=GlobusTest/OU=simpleCA-grid2-desktop/CN=bacon" bacon
"/O=Grid/OU=GlobusTest/OU=simpleCA-grid2-desktop/CN=darshan" darshan
"/O=Grid/OU=GlobusTest/OU=simpleCA-grid2-desktop/CN=aditya" Aditya

bacon@grid2-desktop:~$ grid-cert-info –subject /O=Grid/OU=GlobusTest/OU=simpleCA-grid2-desktop/CN=bacon

Starting the web services container

go to the grid2-desktop machine

root@elephant:~# cp $GLOBUS_LOCATION/etc/init.d/globus-ws-java-container /etc/init.d

globus@elephant:~$ /etc/init.d/globus-ws-java-container start
Starting Globus container. PID: 29985

bacon@grid2-desktop:~$ globus-check-remote-environment -s https://localhost:8443
### Remote Endpoint Version Information ###
Apache Axis version: 1.4
Built on Mar 01, 2007 (10:42:15 CST)
Java WS Core Version on remote endpoint https://localhost:8443: 4.2.1

bacon@grid2-desktop:~$ globusrun-ws -submit -c /bin/date
Submitting job...Done.
Job ID: uuid:b6773138-5abd-11e0-9f1f-00016c8a00c3
Termination time: 03/30/3011 11:06 GMT
Current job state: Active
Current job state: Done

bacon@grid2-desktop:~/.globus$ globus-job-run grid1-desktop /bin/date
Tue Mar  1 18:04:27 IST 2011
Appendices

**Set up GridFTP**

apt-get install xinetd

[root@bigboy tmp]# service xinetd start

[root@bigboy tmp]# service xinetd stop

[root@bigboy tmp]# service xinetd restart

root@elephant:/etc/grid-security# **gedit /etc/xinetd.d/gridftp**

```
service gsiftp  
{  
  instances = 100  
  socket_type = stream  
  wait = no  
  user = root  
  env += GLOBUS_LOCATION=/usr/local/globus  
  env += LD_LIBRARY_PATH=/usr/local/globus/lib  
  server = /usr/local/globus/sbin/globus-gridftp-server  
  server_args = -i  
  log_on_success += DURATION  
  disable = no  
}
```

root@elephant:/etc/grid-security# **gedit /etc/services**

**Add below line as last line**

```
# Local services
# Local services
gsigatekeeper    2119/tcp                       # Globus Gatekeeper
gsiftp          2811/tcp
```

root@elephant:/etc/grid-security# **/etc/init.d/xinetd reload**

Reloading internet superserver configuration: xinetd.

root@elephant:/etc/grid-security# **netstat -an | grep 2811**

```
tcp 0 0 0.0.0.0:2811 0.0.0.0:* LISTEN
```
Appendices

Writing Job output to file (-so option)

If you want the output of the job to be written to a local file instead of the terminal you’ll have to add the -so option:

```
globusrun-ws -submit -F https://grid2-desktop:8443/wsrf/services/ManagedJobFactoryService -s -so job.out -c /bin/echo hello world!
```

cat job.out

hello world!

globusrun-ws -submit -s -so datejob.out -c /bin/date

Using a Job description

The specification of a job to submit is to be written by the user in a job description XML file.

Here is an example of a simple job description:

```
<job>
  <executable>/bin/echo</executable>
  <argument>this is an example_string</argument>
  <argument>Globus was here</argument>
  <stdout>${GLOBUS_USER_HOME}/stdout</stdout>
  <stderr>${GLOBUS_USER_HOME}/stderr</stderr>
</job>
```

Tell globusrun-ws to read the job description from a file, using the -f argument:

```
% bin/globusrun-ws -submit -f simple.xml
```

Submitting job...Done.
Job ID: uuid:c51fe35a-4fa3-11d9-9cfc-000874404099
Termination time: 12/17/2004 20:47 GMT
Current job state: Active
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
Appendices

```
bacon@grid2-desktop:~$ mv simple.xml simple.rsl

bacon@grid2-desktop:~$ globusrun-ws -submit -f simple.rsl

Submitting job...Done.
Job ID: uuid:a1d05796-5ac2-11e0-9dda-00016c8a00c3
Termination time: 03/30/3011 11:41 GMT
Current job state: Active
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
```
10.3 Configuration and setup of Hadoop Cluster

Install Hadoop for single node

Prerequisite: JDK 1.6

Steps:

1) Configure SSH i.e. Secure Shell
   It is used to remotely login to another node for running the application and is already installed.
   hduser@grid1-desktop:-ssh-keygen -t rsa -P “ ”
   This will generate an SSH key for hduser (empty password SSH)
   hduser@grid1-desktop:-
   $ cat $HOME/.ssh/id_rsa.pub>>$HOME/.ssh/authorized_keys
   (5)$ssh localhost

2) Hadoop installation:-
   Download Hadoop in /usr/local (as below)
   $cd /usr/local
   $ sudo tar –xvzf Hadoop-0.20.2.tar.gz
   $ sudo mv hadoop-0.20.2 hadoop-1.0.0
   $sudo chown –R hduser:hadoop Hadoop-1.0.0
   (this changes the owner to hduser)

3) Update $HOME/.bashrc for hduser
   Do the following changes in $HOME/.bashrc
   export HADOOP_HOME=/usr/local/Hadoop-1.0.0
   (to set the Hadoop-related environment variable)
   Export JAVA_HOME=/usr/lib/jdk1.6.0_16

4) Configuration
   Go to the conf directory and open Hadoop-env.sh
   hduser@grid1-desktop:-$ cd /usr/local/Hadoop-1.0.0/conf
   $gedit Hadoop-env.sh
   In this file, make the following changes:-
   export JAVA_HOME=/usr/lib/jdk1.6.0_16
5) Create a temp directory
   $sudo mkdir –p /app/Hadoop/tmp
   Change the owner of this directory to hduser:-
   $ sudo chown hduser:hadoop /app/Hadoop/tmp

   Change the /conf/core-site.xml file:

   <value> /app/hadoop/tmp </value>
   Change the /conf/mapred-site.xml file:-
   <value>grid1-desktop:54311</value>
   Change /conf/hdfs-site.xml file:-
   <value> 1 </value>

6) Format the HDFS via NameNode
   hduser@grid1-desktop:-$ /usr/local/hadoop-1.0.0/bin/hadoop  namenode –format
   (this will start the namenode)

7) Start Single-node cluster
   hduser@grid1-desktop:- $ /usr/local/hadoop-1.0.0/bin/start-all.sh
   (this will start NameNode, DataNode, JobTracker, TaskTracker on local machine)

8) Check whether Hadoop processes are running:-
   $cd /usr/local/hadoop-1.0.0
   $ jps
   Output:-

9) Check if Hadoop is listening on the configured ports.
Hadoop Configuration for multi-node cluster

Prerequisite:
Configuring single-node clusters first.

Steps:

1) Configure the /etc/hosts file on both the machines.

   # /etc/hosts (for master AND slave)
   10.0.5.31    master
   10.0.5.38    slave

2) hduser@master:~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@slave

   This command will prompt you for the login password for user hduser on slave, then copy the public SSH key for you, creating the correct directory and fixing the permissions as necessary.

   This command will prompt you for the login password for user hduser on slave, then copy the public SSH key for you, creating the correct directory and fixing the permissions as necessary.

   So, connecting from master to master…

       hduser@grid1-desktop:~ $ ssh master

   and from master to slave.

       hduser@grid2-desktop:~$ ssh slave

3) Configure the conf/masters (master only)

   update /conf/masters and enter the IP address of the master node:
   10.0.5.31

   If you have additional slave nodes, just add them to the conf/slaves file, one per line:

   10.0.5.38
   10.0.5.39
   10.0.5.40
   10.0.5.41
4) Change the fs.default.name variable (in conf/core-site.xml) which specifies the NameNode (the HDFS master) host and port. In our case, this is the master machine.

5) Change the mapred.job.tracker variable (in conf/mapred-site.xml) which specifies the JobTracker (MapReduce master) host and port. Again, this is the master in our case.

6) Start the multi-node cluster.

At this point, the following Java processes should run on master…

/usr/local/hadoop/bin:~$ startall.sh