GPGPU Cluster

Pierre Schweitzer

October 2012

Contents

Listings

1 Global Overview

The cluster is made of one master machine that is used to dispatch the jobs over the eight nodes. The master (using IP: 172.16.67.206) cannot run any job and is the controling machine for all the shared deamons, like file system, job deamon, but also DNS, DHCP and LDAP. All the nodes managed by the deamons are in the same gigabit subnetwork (192.168.1.0/24) and the master also acts as a gateway for them.

Machine are affected IPs and names that way:
geforce{id} → 192.168.1.{50 + id}
apu{id} → 192.168.1.{100 + id}

2 Hardware

2.1 Complete configuration

The cluster is made of eight nodes. Seven are using the following configuration:

- CORSAIR 650D Obsidian Series
- ASUS-P8P67-DE
- INTEL Core i7 2600K - 3.4 GHz
- GAINWARD GTX590-3072 - GeForce GTX 590[1]
- CORSAIR Vengeance bleue DDR3 PC3-15000 4GB x2
- NZXT HALE90 - 1000W - 80+ Gold
- SAMSUNG-HD10 Disque dur interne 3.5” 1000 go 32 mo 7200 tr/min - SATA

Those are refered as geforce/cuda machines.

The eighth machine is the following:

- CORSAIR 650D Obsidian Series
- GIGABYTE GA-A75M-UD2H
- AMD A-Series A8 3850 - 2.9 GHz - AMD Radeon HD 6550D
- CORSAIR Vengeance bleue DDR3 PC3-15000 4GB x2
- COOLERMASTER Alimentation Modulaire Silent Pro 700 Watts
- SAMSUNG-HD10 Disque dur interne 3.5” 1000 go 32 mo 7200 tr/min - SATA
- D-LINK DGE-528T

It is refered as apu.

The master will not be described as it has nothing to do with computation.

[1] This is seen as two cards

2.2 MAC addresses

- apu0
  - Realtek: 50:e5:49:52:df:92
  - Intel: fc:75:16:56:f6:5f
- dolly
  - Broadcom: 00:23:ae:8a:fd:06
- geantvert
  - Broadcom: bc:30:5b:b0:8a:95
  - D-link: 01:1b:21:4f:06:49
- geforce0
  - Realtek: f4:6d:04:e4:41:5c
  - Intel: f4:6d:04:e4:41:f9
- geforce1
  - Realtek: f4:6d:04:e4:4e:3f
  - Intel: f4:6d:04:e4:56:06
- geforce2
  - Realtek: f4:6d:04:e4:41:69
  - Intel: f4:6d:04:e4:41:f5
- geforce3
  - Realtek: f4:6d:04:e4:41:9a
  - Intel: f4:6d:04:e4:41:b5
- geforce4
  - Realtek: f4:6d:04:e4:40:62
  - Intel: f4:6d:04:e4:41:a7
- geforce5
  - Realtek: f4:6d:04:e4:4e:26
  - Intel: f4:6d:04:e4:41:a8
- geforce6
  - Realtek: f4:6d:04:e4:41:66
  - Intel: f4:6d:04:e4:41:f7
- master
  - Broadcom: 00:1d:09:03:33:3e
  - D-link: fc:75:16:56:f4:ce

- Intel: 00:1b:21:62:28:08
- CNet: 00:80:ad:77:29:35

See 4.2 for an helper script to wake all the machines using the wake on lan protocol.

3 Software

3.1 DNS & DHCP

The nodes are strictly the same regarding software and configuration. Their only difference comes from their hostname. That one is transmitted by the DHCP server when they boot.

The master is then running a DNS & DHCP server, using the dnsmasq deamon. This one gives an IP address in the cluster subnet to each node and its name. The name being either geforceID or apuID. It also enables the domain name to the machine.

This allows connecting to the machine from anywhere using its name. As long as the machine did not request its IP address and its hostname, it is not set.

The server also sets domains for static machines.

To get this working properly, a fix to /sbin/dhclient-script has been required to get the hostname being really set on the nodes. This is due to a Debian bug left unfixed[2]

See 5.2 for the configuration files.

3.2 GlusterFS

The home directories are set as a RAID-1 volume shared between all the nodes. This means that the volumes (names homes) aggregates all the spare space on all the disks on the nodes and create a complete volume which is redundant. Then, it is mounted through GlusterFS on all the nodes as home. This offers the following features:

- Shared homes on all nodes and cluster
- Redundant data preventing disk failures

This is administrated through the gluster CLI. Each spare disk has to be partitionned. Then, this is referenced as a "brick". A brick being a partition on a node. Then, the bricks are merged on the master using the CLI, while setting the redundancy. Because we set 2 replications (original and replicate) the number of bricks as to be even.

3.3 Slurm

Slurm is the "resource manager" for the cluster. It means that it controls all the jobs that run on the cluster. Its head is on the master and jobs have to be queued from the master. Slurm was choosen over Torque (which is another

[2] http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=604883#20

manager widely spread over the clusters) because it allows managing GPUs as a resource and also it offers a compatibility with Torque (same commands).

The master is the head of slurm, but this is fully transparent to slurm. All the machines (master and node) have the same configuration. Slurm makes the difference when it starts by looking at the hostname, seeing whether it is running on master or on a node. Only difference is that the nodes must also ship the number of GPU they provide. GPUs are seen as Generic Resources (GRES) and thus require their own configuration.

One drawback of Slurm is that it requires that all the nodes have their name (and IP defined) before it starts on master. One workaround to this is to define the nodes statically in the DNS server (and not only in the DHCP server).

See 5.3 for the configuration files.

See 4.3 for an helper script to dispatch configuration.

3.4 LDAP

3.5 ifenslave

Because all the machines have two network cards and because of the huge internal traffic (jobs, data - with GlusterFS), it has been decided to practise the inteface bonding. This makes the system see the two interfaces as one and to transfer data over the single interface it sees while data are dispatched over the two. This allows higher transfer rate.

Software used is ifenslave with the bonding driver. An init script has been written to automatically bond the interfaces. It is run after the DHCP. It gets the IP address affected to the machine, gives it to the bond interface and enslaves the two network cards.

It has been choosen that on all the machines, the realtek card, which is receiving the DHCP address is eth0 and master of the bonding. eth1 is then the intel card.

See 5.1 for the configuration file.

3.6 iptables

To allow the master to be a gateway for all the nodes, it requires a small iptables configuration. It is used to route all the traffic for the nodes over the master. To make it automatically at boot, a small script has been written and put in /etc/network/if-pre-up.d

See 4.1 for the full script.

3.7 Held packages

In order to ensure correct functionning of the cluster, some packages have been held in aptitude to prevent them from being updated.

First package is isc-dhcp-client. Upgrading it would led to erase the fix which has been done on DHCP init script with hostnames. See 3.1 for more information about the fix.

Second package is opencl-headers. It has to remain in 1.1 release. The 1.2 standard does not come with C++ support.

4 Scripts

4.1 iptables

Listing 1: /etc/network/if-pre-up.d/gateway

```
#!/bin/sh
iptables --table nat --append POSTROUTING --out-interface eth1 -j MASQUERADE
iptables --append FORWARD --in-interface eth3 -j ACCEPT
echo 1 > /proc/sys/net/ipv4/ip_forward
```

4.2 wakeonlan

Listing 2: /root/utils/broadcast_wake_on_lan.sh

```
#! /bin/sh

wakeonlan -i 192.168.1.255 \
50:e5:49:52:df:92 \
fc:75:16:56:f6:5f \
00:23:ae:8a:fd:06 \
bc:30:5b:b0:8a:95 \
f4:6d:04:e4:4e:3f \
f4:6d:04:e4:56:06 \
f4:6d:04:e4:41:69 \
f4:6d:04:e4:41:f5 \
f4:6d:04:e4:40:62 \
f4:6d:04:e4:41:a7 \
f4:6d:04:e4:4e:26 \
f4:6d:04:e4:41:a8 \
f4:6d:04:e4:41:66 \
f4:6d:04:e4:41:f7 \
00:1d:09:03:33:3e \
fc:75:16:56:f4:ce \
f4:6d:04:e4:41:9a \
f4:6d:04:e4:41:b5
```

4.3 Slurm

Listing 3: /root/utils/broadcast_slurm_config.sh

```
#! /bin/sh

for i in $*; do
        scp /etc/slurm-llnl/slurm.conf $i:/etc/slurm-llnl/slurm.conf
        ssh $i "/etc/init.d/slurm-llnl␣restart"
done
```

5 Configuration files

5.1 Bonding

Listing 4: /etc/network/interfaces

```
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.

auto lo
iface lo inet loopback

auto bond0
iface bond0 inet dhcp
        slaves eth0 eth1
        bond-mode active-backup
        bond-miimon 100
        bond-downdelay 200
        bond-updelay 200
```

5.2 dnsmasq

Listing 5: /etc/dnsmasq.d/pxe-boot

```
dhcp-boot=pxelinux.0
enable-tftp
tftp-root=/var/lib/tftpboot
no-dhcp-interface=eth1
dhcp-range=192.168.1.50,192.168.1.150,255.255.255.0,12h

no-hosts
cache-size=1000
dns-forward-max=150

# Geforce hosts
# Allow RLT, disallow Intel
dhcp-host=f4:6d:04:e4:41:5c,192.168.1.50,geforce0,infinite
dhcp-host=f4:6d:04:e4:41:f9,ignore
dhcp-host=f4:6d:04:e4:4e:3f,192.168.1.51,geforce1,infinite
dhcp-host=f4:6d:04:e4:56:06,ignore
dhcp-host=f4:6d:04:e4:41:69,192.168.1.52,geforce2,infinite
dhcp-host=f4:6d:04:e4:41:f5,ignore
dhcp-host=f4:6d:04:e4:41:9a,192.168.1.53,geforce3,infinite
dhcp-host=f4:6d:04:e4:41:b5,ignore
dhcp-host=f4:6d:04:e4:40:62,192.168.1.54,geforce4,infinite
dhcp-host=f4:6d:04:e4:41:a7,ignore
dhcp-host=f4:6d:04:e4:4e:26,192.168.1.55,geforce5,infinite
dhcp-host=f4:6d:04:e4:41:a8,ignore
dhcp-host=f4:6d:04:e4:41:66,192.168.1.56,geforce6,infinite
dhcp-host=f4:6d:04:e4:41:f7,ignore

# APU hosts
dhcp-host=50:e5:49:52:df:92,192.168.1.100,apu0,infinite
dhcp-host=fc:75:16:56:f6:5f,ignore
```

```
# Others hosts
dhcp-host=bc:30:5b:b0:8a:95,192.168.1.150,geantvert,infinite
dhcp-host=01:1b:21:4f:06:49,ignore

# Other hosts
address=/dolly/172.16.66.65
address=/master/172.16.67.206
address=/monmasteramoiavecdesgpus/172.16.67.206
address=/gmaster/172.16.67.206

# Fixed DNS entries
address=/geforce0/192.168.1.50
address=/geforce1/192.168.1.51
address=/geforce2/192.168.1.52
address=/geforce3/192.168.1.53
address=/geforce4/192.168.1.54
address=/geforce5/192.168.1.55
address=/geforce6/192.168.1.56
address=/apu0/192.168.1.100
address=/geantvert/192.168.1.150

log-dhcp
```

5.3 Slurm

Listing 6: /etc/slurm-llnl/slurm.conf

```
# slurm.conf file generated by configurator.html.
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.
#
# Master is the control machine of the cluster
ControlMachine=Master
#ControlAddr=
#BackupController=
#BackupAddr=
#
AuthType=auth/none
CacheGroups=0
#CheckpointType=checkpoint/none
CryptoType=crypto/openssl
#DisableRootJobs=NO
#EnforcePartLimits=NO
#Epilog=
#EpilogSlurmctld=
#FirstJobId=1
#MaxJobId=999999
#GresTypes=
#GroupUpdateForce=0
#GroupUpdateTime=600
#JobCheckpointDir=/var/slurm/checkpoint
```

```
JobCredentialPrivateKey=/etc/slurm-llnl/slurm.key
JobCredentialPublicCertificate=/etc/slurm-llnl/slurm.cert
#JobFileAppend=0
#JobRequeue=1
#JobSubmitPlugins=1
#KillOnBadExit=0
#Licenses=foo*4,bar
MailProg=/bin/mail
#MaxJobCount=5000
#MaxStepCount=40000
#MaxTasksPerNode=128
MpiDefault=none
#MpiParams=ports=#-#
#PluginDir=
#PlugStackConfig=
#PrivateData=jobs
ProctrackType=proctrack/pgid
#Prolog=
#PrologSlurmctld=
#PropagatePrioProcess=0
#PropagateResourceLimits=
#PropagateResourceLimitsExcept=
#ReturnToService=1
ReturnToService=2
#SallocDefaultCommand=
SlurmctldPidFile=/var/run/slurmctld.pid
SlurmctldPort=6817
SlurmdPidFile=/var/run/slurmd.pid
SlurmdPort=6818
SlurmdSpoolDir=/tmp/slurmd
SlurmUser=slurm
#SlurmdUser=root
#SrunEpilog=
#SrunProlog=
StateSaveLocation=/tmp
SwitchType=switch/none
#TaskEpilog=
TaskPlugin=task/none
#TaskPluginParam=
#TaskProlog=
#TopologyPlugin=topology/tree
#TmpFs=/tmp
#TrackWCKey=no
#TreeWidth=
#UnkillableStepProgram=
#UsePAM=0
#
#
# TIMERS
#BatchStartTimeout=10
```

```
#CompleteWait=0
#EpilogMsgTime=2000
#GetEnvTimeout=2
#HealthCheckInterval=0
#HealthCheckProgram=
InactiveLimit=0
KillWait=30
#MessageTimeout=10
#ResvOverRun=0
MinJobAge=300
#OverTimeLimit=0
SlurmctldTimeout=120
SlurmdTimeout=300
#UnkillableStepTimeout=60
#VSizeFactor=0
Waittime=0
#
#
# SCHEDULING
#DefMemPerCPU=0
FastSchedule=1
#MaxMemPerCPU=0
#SchedulerRootFilter=1
#SchedulerTimeSlice=30
SchedulerType=sched/backfill
SchedulerPort=7321
SelectType=select/cons_res
#SelectTypeParameters=CR_Core_Memory
SelectTypeParameters=CR_CPU
#
#
# JOB PRIORITY
#PriorityType=priority/basic
#PriorityDecayHalfLife=
#PriorityCalcPeriod=
#PriorityFavorSmall=
#PriorityMaxAge=
#PriorityUsageResetPeriod=
#PriorityWeightAge=
#PriorityWeightFairshare=
#PriorityWeightJobSize=
#PriorityWeightPartition=
#PriorityWeightQOS=
#
#
# LOGGING AND ACCOUNTING
#AccountingStorageEnforce=0
#AccountingStorageHost=
#AccountingStorageLoc=
#AccountingStoragePass=
#AccountingStoragePort=
AccountingStorageType=accounting_storage/none
#AccountingStorageUser=
AccountingStoreJobComment=YES
ClusterName=gpucluster
#DebugFlags=
#JobCompHost=
#JobCompLoc=
#JobCompPass=
#JobCompPort=
#JobCompType=jobcomp/filetxt
#JobCompUser=
JobAcctGatherFrequency=30
JobAcctGatherType=jobacct_gather/none
SlurmctldDebug=3
SlurmctldLogFile=/var/log/slurm-llnl/slurmctld.log
SlurmdDebug=3
SlurmdLogFile=/var/log/slurm-llnl/slurm.log
#SlurmSchedLogFile=/var/log/slurm-llnl/sched.log
#SlurmSchedLogLevel=
#
#
# POWER SAVE SUPPORT FOR IDLE NODES (optional)
#SuspendProgram=
#ResumeProgram=
#SuspendTimeout=
#ResumeTimeout=
#ResumeRate=
#SuspendExcNodes=
#SuspendExcParts=
#SuspendRate=
#SuspendTime=
#
#
# COMPUTE NODES
GresTypes=gpu
#NodeName=geforce[0-6] RealMemory=7984 Sockets=1 CoresPerSocket=4 ThreadsPerCor
#NodeName=apu0 RealMemory=7499 Sockets=1 CoresPerSocket=4 ThreadsPerCore=1 Stat
#NodeName=GeantVert RealMemory=12041 Sockets=2 CoresPerSocket=4 ThreadsPerCore=
NodeName=geforce[0-6] RealMemory=7984 CPUs=8 Feature=”geforce” Gres=gpu:2 State
NodeName=apu0 RealMemory=7499 CPUs=4 Gres=gpu:0 State=UNKNOWN
NodeName=GeantVert RealMemory=12041 CPUs=16 Feature=”Fermi” Gres=gpu:2 State=UN
PartitionName=queue Nodes=geforce[0-6],GeantVert,apu0 Default=YES MaxTime=INFIN
```

Listing 7: /etc/slurm-llnl/gres.conf

```
Name=gpu File=/dev/nvidia0
Name=gpu File=/dev/nvidia1
```