



ISBN 978-89-5884-987-2 98560

Global File System 설치 가이드

ver.1.0

부 서: 슈퍼컴퓨팅센터
작성자: 조혜영, 홍태영

한국과학기술정보연구원

Korea Institute of Science and Technology Information



목 차

1. 설치 전 준비	3
1.1 시스템 사양	3
1.2 OS	3
 2. GFS 6.0 패키지 준비 및 설치	6
2.1 RPM으로 설치하는 방법	6
2.2 SOURCE로 설치하는 방법	11
 3. GFS 구성	14
 4. GFS CONFIGURATION	16
4.1 사전 작업	16
4.2 GFS CONFIGURATION 과정	18
 5. GIGABIT ETHERNET로 SETUP	37

1. 설치 전 준비

1.1 시스템 사양

- node 수 : 9
- CPU : P4 XEON 2.8C (Hyperthreading ON)
- Mem : 1GB
- HDD : 80GB SATA
- NIC : e100(100Mbps) & tg3(1000Mbps)

1.2 OS

- 설치 OS : Linux frontend-0.public 2.4.21-15.ELsmp #1 SMP Mon May 17 21:03:21 GMT 2004 i686 i686 i386 GNU/Linux
- OS 설치과정 : Rocks 3.2.0 를 사용하여 9node 에 설치,
Rocks 3.2.0 8node 에 Rocks 로 클러스터 설정 참조
- OS 파티션
현재 디스크 상황 check

```
hycho@compute-0-1 hycho]$ df

Filesystem 1K-blocks Used Available Use% Mounted on
/dev/sda1 61906641750356412584030% /
/dev/sda5 2064178832828195603201% /home1
/dev/sda3 2064178832828195603201% /home2
/dev/sda2 2064178832828195603201% /home3
/dev/sda7 77638563282873366361% /home4
none 50578805057880% /dev/shm
frontend-0.local:/home1/hycho
20161172757424183796084% /home/hycho
```

=> 각 계산 노드의 /home/~hycho 은 frontend 의 /home1/hycho 에 automount 되어 있다.

* 커널 버전이 설치하려는 GFS 버전과 같으므로 커널 컴파일 과정 생략

(GFS-6.0.0-1.2.src.rpm 와 kernel 2.4.21-15.ELsmp 가 맞고, GFS-6.0.0-7.1.src.rpm 와 2.4.21-15.0.4.EL 와 맞다)

-. kernel update

```
[root@compute-0-5 root]# cd /install/GFS

[root@compute-0-5 GFS]# ls

kernel-source-2.4.21-15.0.4.EL.i386.rpm

[root@compute-0-5 GFS]# rpm -ivh kernel-source-2.4.21-15.0.4.EL.i386.rpm

Preparing...                               ##### [100%]

 1:kernel-source                           ##### [100%]

rpm -ivh kernel-source-2.4.21-15.0.4.EL.i386.rpm

[root@compute-0-5 GFS]# cd /usr/src/linux-2.4
```

※ make mrproper 이 필요.

이전에 설정한 정보를 지우거나 새로운 커널 소스를 가져다 컴파일 할 때에는 이 명령을 먼저 사용해야 한다. make mrproper 명령은 이전 커널을 컴파일할 때 만들어진 오브젝트 파일(*.o)과 의존성 설정, 컴파일 환경 설정값, 버전 정보 등 새로 시작하는 컴파일에 영향을 주는 이전 정보들을 삭제한다.

2. GFS 6.0 패키지 준비 및 설치

2.1 rpm 으로 설치하는 방법

(a) `ftp://ftp.redhat.com/pub/redhat/linux/enterprise/3/en/RHGFS/i386/SRPMS`

(`http://ftp.freshrpms.net/pub/redhat /linux/enterprise/3/en/RHGFS/i386/`)에서 `perl-`

`Net-Telnet-3.03-2.src.rpm`, `rh-gfs-en-6.0-4.src.rpm` 다운로드

(b)

[ftp://ftp.redhat.com/pub/redhat/linux/updates/enterprise/3AS/en/RHGFS/SRPMS](http://ftp.redhat.com/pub/redhat/linux/updates/enterprise/3AS/en/RHGFS/SRPMS) 에서

`GFS -6.0.0-1.2.src.rpm` 다운로드

(`GFS-6.0.0-1.2.src.rpm` 와 `kenel 2.4.21-15.ELsmp` 가 맞고, `GFS-6.0.0-`

`7.1.src.rpm` 와 `2.4.21-15.0.4.EL` 와 맞다)

(c) 소스 RPM 설치

```
#cd /install/GFS/SRPMS
```

```
# rpm -Uvh perl-Net-Telnet-3.03-2.src.rpm
```

```
# rpm -Uvh rh-gfs-en-6.0-4.src.rpm
```

```
# rpm -UVh GFS-6.0.0-1.2.src.rpm GFS-6.0.0-7.1.src.rpm 다운로드
```

(d) GFS RPM build

```
# cd /usr/src/redhat/SPECS
```

```
# rpmbuild -ba perl-Net-Telnet.spec
```

```
# rpmbuild -ba rh-gfs-en.spec
```

```
# rpmbuild -ba gfs-build.spec --target i686
```

(GFS-6.0.0-1.2.src.rpm 버전문제-smp 버전에서 할 때는 rpm 에 문제가 있는 것으로 보임, smp 커널 위치를 잘 못 찾는 것으로 보임, c06 에만 rpm 에 SPEC 파일과 kernel MakeFile 을 수정하여 rpmbuild 가 되게 했음)

※ "# rpmbuild -ba perl-Net-Telnet.spec"를 실행하기 위해서는

perl-Digest-HMAC-1.01-11.1.src.rpm 와 perl-Digest-SHA1-2.01-15.1.src.rpm 가 필요하다.

이 rpm 은 [ftp://ftp.redhat.com/pub/redhat/linux/enterprise/3/en/os/i386/SRPMS/](http://ftp.redhat.com/pub/redhat/linux/enterprise/3/en/os/i386/SRPMS/)에서 구할 수 있다. ftp 를 사용하여, prompt, hash, mget 등의 명령어를 사용하면 편리하게 down 받을 수 있다.

※ perl-Digest-HMAC-1.01-11.1.src.rpm 와 perl-Digest-SHA1-2.01-15.1.src.rpm 설치

(dependency 문제로 perl-Digest-SHA1-2.01-15.1.src.rpm 를 먼저 설치한 후, perl-Digest-SHA1-2.01-15.1.src.rpm 를 설치해야 한다.)

(가) 소스 RPM 설치

```
[root@compute-0-0 SRPMS]# rpm -Uvh perl-Digest-SHA1-2.01-15.1.src.rpm
```

```
warning: perl-Digest-SHA1-2.01-15.1.src.rpm: V3 DSA signature: NOKEY, key ID db42a60e
```

```
1:perl-Digest-SHA1 #####
```

```
[100%]
```

```
[root@compute-0-0 SRPMS]# rpm -Uvh perl-Digest-HMAC-1.01-11.1.src.rpm
```

```
warning: perl-Digest-HMAC-1.01-11.1.src.rpm: V3 DSA signature: NOKEY, key ID db42a60e
```

```
1:perl-Digest-HMAC #####
```

```
[100%]
```

(나) perl-Digest-SHA1-2.01-15.1.src.rpm RPM build & 설치

```
# cd /usr/src/redhat/SPECS
```

```
# rpmbuild -ba perl-Digest-SHA1.spec
```

```
# cd /usr/src/redhat/RPMS/i386
```

```
# rpm -Uvh perl-Digest-SHA1-2.01-15.1.i386.rpm
```

(다) perl-Digest-HMAC-1.01-11.1.src.rpm RPM build & 설치

```
# cd /usr/src/redhat/SPECS
```

```
# rpmbuild -ba perl-Digest-HMAC.spec
```

```
# cd /usr/src/redhat/RPMS/noarch
```

```
# rpm -Uvh perl-Digest-HMAC-1.01-11.1.noarch.rpm
```


(e) GFS RPM 설치

```
# cd /usr/src/redhat/RPMS/noarch  
  
# rpm -Uvh perl-Net-Telnet-3.03-2.noarch.rpm  
  
# rpm -Uvh rh-gfs-en-6.0-4.noarch.rpm  
  
# cd /usr/src/redhat/RPMS/i686  
  
# rpm -Uvh GFS-6.0.0-7.1.i686.rpm  
  
# rpm -Uvh GFS-modules-smp-6.0.0-7.1.i686.rpm
```

※ rh-gfs-en-6.0-4.noarch.rpm 은 htmlview 를 필요로 함으로 설치한다.

※ Test 로 front, c00, c06 만 위의 절차대로 설치하고 나머지는 마지막 rpm 만 copy 해서 아래와 같이 설치했음

```
# scp -r c00:/usr/src/redhat/RPMS/ /usr/src/redhat/
```

(scp 명령은 RPMS 디렉토리가 대상서버에 있으면 RPMS 디렉토리를 또 만든다. 즉 "#
scp -r c00:/usr/src/redhat/RPMS/ /usr/src/redhat/RPMS"로 실행하면
/usr/src/redhat/RPMS/RPMS 로 만들어진단.)

```

pvfs-kernel-1.6.0-1.i386.rpm          100%  79KB  0.0KB/s  00:00

perl-Digest-SHA1-2.01-15.1.i386.rpm    100%  33KB  0.0KB/s  00:00

perl-Digest-HMAC-1.01-11.1.noarch.rpm  100%  11KB  0.0KB/s  00:00

perl-Net-Telnet-3.03-2.noarch.rpm      100%  51KB  0.0KB/s  00:00

rh-gfs-en-6.0-4.noarch.rpm             100% 415KB  0.0KB/s  00:00

htmlview-2.0.0-10.noarch.rpm           100% 4805   0.0KB/s  00:00

GFS-6.0.0-1.2.i686.rpm                 100% 532KB  0.0KB/s  00:00

GFS-devel-6.0.0-1.2.i686.rpm           100%  27KB  0.0KB/s  00:00

GFS-modules-6.0.0-1.2.i686.rpm         100% 231KB  0.0KB/s  00:00

GFS-modules-smp-6.0.0-1.2.i686.rpm     100% 237KB  0.0KB/s  00:00

# cd /usr/src/redhat/RPMS/i386

# rpm -Uvh perl-Digest-SHA1-2.01-15.1.i386.rpm

Preparing... ##### [100%]

    1:perl-Digest-SHA1 ##### [100%]

# cd /usr/src/redhat/RPMS/noarch/

# rpm -Uvh perl-Digest-HMAC-1.01-11.1.noarch.rpm

Preparing... ##### [100%]

    1:perl-Digest-HMAC ##### [100%]

# rpm -Uvh perl-Net-Telnet-3.03-2.noarch.rpm

Preparing... ##### [100%]

    1:perl-Net-Telnet ##### [100%]

# rpm -Uvh htmlview-2.0.0-10.noarch.rpm

```

```

Preparing... ##### [100%]

    1:htmlview ##### [100%]

# rpm -Uvh rh-gfs-en-6.0-4.noarch.rpm

Preparing... ##### [100%]

    1:rh-gfs-en ##### [100%]

    # cd /usr/src/redhat/RPMS/i686/

# rpm -Uvh GFS-6.0.0-1.2.i686.rpm

Preparing... ##### [100%]

    1:GFS ##### [100%]

# rpm -Uvh GFS-modules-smp-6.0.0-1.2.i686.rpm

Preparing... ##### [100%]

    1:GFS-modules-smp ##### [100%]

```

2.2 source 로 설치하는 방법

```

* scp c03:/usr/src/redhat/BUILD/gfs-build.tar.gz

* cd /usr/src/redhat/BUILD/

* cp -r /usr/src/linux-2.4.21-15.0.4.EL/ .

* mkdir GFS-6.0.0

* mv gfs-build.tar.gz ./GFS-6.0.0/

* cd GFS-6.0.0/

* tar zxvf gfs-build.tar.gz

* cd linux-2.4.21-15.0.4.EL/

```

```
* for f in ../GFS-6.0.0/GFS/kernel_patches/2.4.21-15.0.4.EL/*; do patch -p1 <
$f;done
```

```
patching file Documentation/Configure.help
```

```
patching file fs/Config.in
```

```
patching file fs/Makefile
```

```
patching file fs/gfs_locking/Makefile
```

```
patching file fs/gfs_locking/lock_harness/Makefile
```

```
patching file Documentation/Configure.help
```

```
patching file fs/Config.in
```

```
patching file fs/Makefile
```

```
.....
```

```
* for f in ../GFS-6.0.0/gnbd/kernel_patches/2.4.21-15.0.4.EL/*;do patch -p1
<$f;done
```

```
* make oldconfig
```

```
* make menuconfig
```

```
/usr/src/redhat/BUILD/linux-2.4.21-15.0.4.EL/에서
```

```
make menuconfig 에서
```

```
FS 에서 GFS 관련 추가(GFS Lock Harness, GFS file system support, Lock Nolock,
Lock GULM).. block device 에서 gnbd 추가, pool 추가
```

```
또는 /usr/src/redhat/BUILD/linux-2.4.21-15.0.4.EL/.config 파일에 아래 내용을 추가
```

```
CONFIG_BLK_DEV_GNBD=m
```

```
CONFIG_BLK_DEV_POOL=m
```

CONFIG_LOCK_HARNESS=m

CONFIG_GFS_FS=m

CONFIG_LOCK_NOLOCK=m

CONFIG_LOCK_GULM=m

* make dep

* cd ../GFS-6.0.0/

* ls

*for i in GFS gnbd bedrock/;do cd \$i; ./configure --mandir=/usr/share/man/ --
kernel_src=/usr/src/redhat/BUILD/linux-2.4.21-15.0.4.EL/ ;cd ../done

* cd gulm/

*./configure --mandir=/usr/share/man/ --kernel_src=/usr/src/redhat/BUILD/linux-
2.4.21-15.0.4.EL/ --libdir=/usr/lib

* cd ../

* vi Makefile 에서 LINUX_DIR 을 수정

#LINUX_DIR=linux

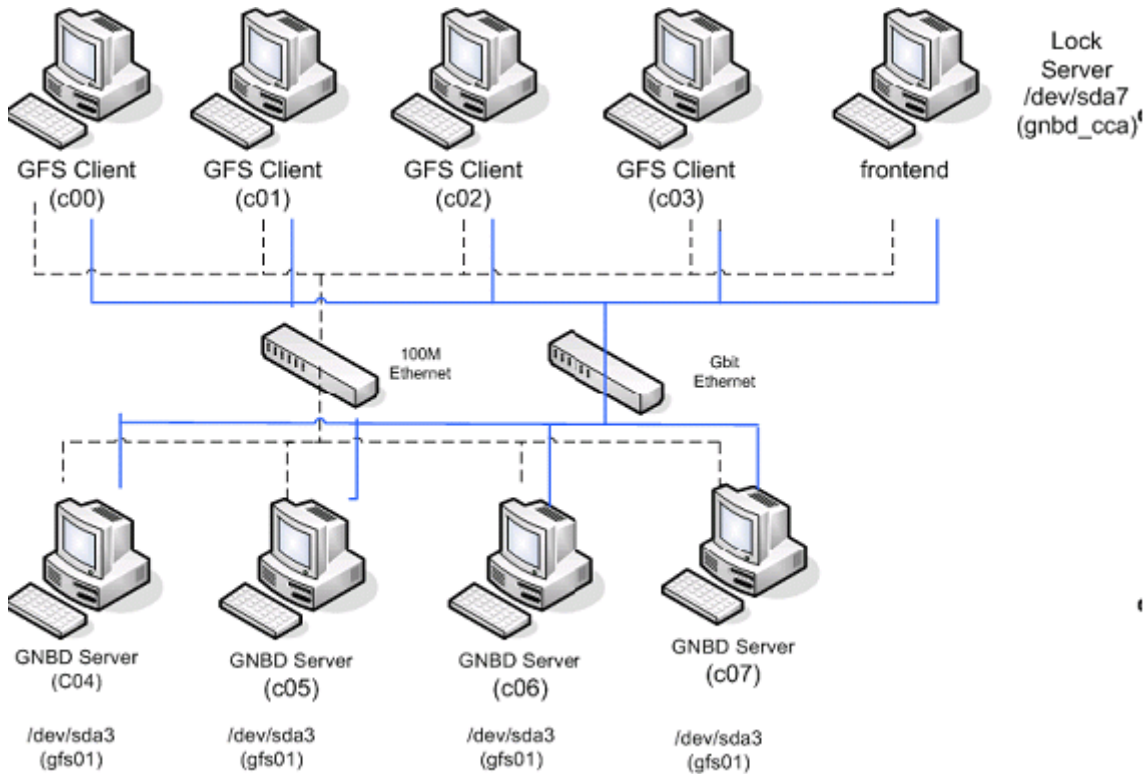
LINUX_DIR=/usr/src/redhat/BUILD/linux-2.4.21-15.0.4.EL/

* make

* make install

3. GFS 구성

* 구성도 개요



(a) 그림에서 보는 바와 같이 GNBD(GFS Network Block Device) 서버로 사용될 compute node00,02,04,06 의 /dev/sda3 을 gnbd_server module 을 이용하여 Network Block Device 로 액세스 가능하도록 함

- 이 때, 각각의 C00, C02, C04, C06 의 /dev/sda3 는 gfs01, gfs01, gfs03, gfs04 로 gnbdname 을 설정함.

(b) GFS 의 pool 로 구성 /dev/pool/pool_gfs01 라는 device 를 생성

(d) /dev/pool/pool_gfs01 을 각 GFS Client 의 /gfs01 과 /gfs02, /gfs03, /gfs04 에 마운트

(e) frontend 를 GFS Lock Server 의 역할을 겸하게 함

-
- single point of failure 를 방지하고자 할 경우 복수의 lock server 를 설정할 필요가 있음
 - (f) frontend 의 /dev/sda7 은 CCS(Cluster Configuration System) archive 로 사용
 - 한 개의 lock server 를 사용할 경우, CCS archive 를 Lock Server 에 두는 것이 더
좋다고 함
 - 복수의 CCS archive 를 사용하여 single point of failure 를 방지하고자 할 경우에
대해서는 GFS Administrator's Guide Ch. 7 의 내용을 참조할 것

4. GFS Configuration

4.1 사전 작업

(a) 모든 노드에서 실행

```
depmod -a  
  
modprobe pool  
  
modprobe lock_gulm  
  
modprobe gfs  
  
lsmod
```

(b) Create file system pools

c00,c02,c04,c6 에서 pool 상태 확인(처음 설치가 아닐 경우 아래와 같이 이전 pool 보임)


```
[root@compute-0-0 root]# pool_tool -s
```

Device	Pool Label
=====	=====
/dev/sda	<- partition information ->
/dev/sda1	<- EXT2/3 filesystem ->
/dev/sda2	<- EXT2/3 filesystem ->
/dev/sda3	<- EXT2/3 filesystem ->
/dev/sda4	<- partition information ->
/dev/sda5	<- EXT2/3 filesystem ->
/dev/sda6	<- swap device ->
/dev/sda7	<- EXT2/3 filesystem ->

```
[root@compute-0-0 root]# df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda1	6190664	2274044	3602152	39%	/
/dev/sda5	20641788	32828	19560320	1%	/home1
/dev/sda3	20641788	32828	19560320	1%	/home2
/dev/sda2	20641788	32828	19560320	1%	/home3
/dev/sda7	7763856	32828	7336636	1%	/home4
none	505788	0	505788	0%	/dev/shm
pvfs2	78368768	188416	78180352	1%	/mnt/pvfs2

4.2 GFS Configuration 과정

(a) Create and export GNBD devices

```
- C01 CCS 로 하기 위해서

# modprobe gnbd_serv

[root@compute-0-1 root]# gnbd_export -e gnbd_cca -d /dev/sda7 -c

gnbd_export: created GNBD cca serving file /dev/sda7

[참고] gnbd_cca 이름이 너무 길면 나중에 pool_tool -c 할 때 error 남

예) gnbd_cca_G 경우

[root@compute-0-3 giga_setup]# pool_tool -c gnbd-cca_G.cfg

line 4: Unable to get file status: No such file or directory at "/dev/gnbd/gnbd_cca_g"

- C00, C02, C04, C06 에서

# modprobe gnbd_serv

# gnbd_export -e gfs00 -d /dev/sda5 -c

gnbd_export: created GNBD gfs00 serving file /dev/sda5
```

※ -c 옵션 의문. Chapter 11. P119 참조

-c 없이 하면

```
[root@compute-0-2 root]# gnbd_export -e gfs02 -d /dev/sda5 -c
```

gnbd_export error: error connecting to gnbd_servd: Connection refused 과 같은
Error 가 뜬다.

gnbd_export -R : exprot 없애기

(b) Import GNBD devices on all GFS nodes and the lock server node

- 모든 노드(c00-c07,front)에서 gkbd_import 할 차례

```
[root@compute-0-0 root]# insmod gkbd
Using /lib/modules/2.4.21-15.ELsmp/kernel/drivers/block/gkbd/gkbd.o
[root@compute-0-0 root]# gkbd_import -i c00
gkbd_import: created directory /dev/gkbd
gkbd_import: created GNBD device "gfs00"
[root@compute-0-0 root]# gkbd_import -i c02
gkbd_import: created GNBD device "gfs01"
[root@compute-0-0 root]# gkbd_import -i c04
gkbd_import: created GNBD device "gfs02"
[root@compute-0-0 root]# gkbd_import -i c06
gkbd_import: created GNBD device "gfs03"
[root@compute-0-0 root]# gkbd_import -i c01
gkbd_import: created GNBD device "cca"
```

위의 결과로 /dev/gkbd/아래에 cca, gfs00, gfs01, gfs02, gfs03 디렉토리가 생긴다.

- (c) Create pool configuration for the four file systems

c00,02,04,06 에 /opt/gfs/gfs.cfg 를 다음과 같이 작성한다.

```
# vi /opt/gfs/gfs.cfg

#polmane      name

poolname gfs

#subpools      number(1 unless speciall data or journal subpools are used)

subpools 1

#subpool      id stripe devices

#나중에 stripe 부분을 수정해 보자. (ex. 512byte per sector)

subpool 0 0 4

#pooldevice    subpool id device

pooldevice 0 0 /dev/gnbd/gfs00

pooldevice 0 1 /dev/gnbd/gfs01

pooldevice 0 2 /dev/gnbd/gfs02

pooldevice 0 3 /dev/gnbd/gfs03
```

(d) Create a pool configuration for the CCS data

c00,02,04,06 에 /opt/gfs/gnbd-cca.cfg 를 다음과 같이 작성한다.

```
poolname gnbd_cca

subpools 1

subpool 0 0 1

pooldevice 0 0 /dev/gnbd/cca
```

(e) Create the pools using the pool_tool command

이 작업은 GNBD client node 하나에서 수행해야 한다.(c03 에서만 수행했음)

```
[root@compute-0-3 ccs_back]# pool_tool -c gfs.cfg

Pool gfs exists. Do you want to erase it? (y/n)y

Erased gfs

Pool label written successfully from gfs.cfg

[root@compute-0-3 ccs_back]# pool_tool -c gnbd-cca.cfg

Pool gnbd_cca exists. Do you want to erase it? (y/n)y

Erased gnbd_cca

Pool label written successfully from gnbd-cca.cfg

[root@compute-0-3 gfs]# pool_tool -c gfs.cfg

/dev/gnbd/gfs00 has "EXT2/3 filesystem" on it.

Are you sure you want to overwrite "EXT2/3 filesystem" with a pool label on
/dev/gnbd/gfs00? (y/n)y

/dev/gnbd/gfs01 has "EXT2/3 filesystem" on it.

Are you sure you want to overwrite "EXT2/3 filesystem" with a pool label on
/dev/gnbd/gfs01? (y/n)y

/dev/gnbd/gfs02 has "EXT2/3 filesystem" on it.

Are you sure you want to overwrite "EXT2/3 filesystem" with a pool label on
/dev/gnbd/gfs02? (y/

/dev/gnbd/gfs03 has "EXT2/3 filesystem" on it.

Are you sure you want to overwrite "EXT2/3 filesystem" with a pool label on
```

```

/dev/gnbd/gfs03? (y/n)y

Pool label written successfully from gfs.cfg

[root@compute-0-3 gfs]# pool_tool -c gnbd-cca.cfg

/dev/gnbd/cca has "EXT2/3 filesystem" on it.

Are you sure you want to overwrite "EXT2/3 filesystem" with a pool label on

/dev/gnbd/cca? (y/n)y

Pool label written successfully from gnbd-cca.cfg

```

아래와 같이 pool 의 상황을 확인할 수 있다.

```

[root@frontend-0 opt]# pool_tool -s

Device                                Pool Label
=====
/dev/gnbd/cca                        gnbd_cca
/dev/gnbd/gfs00                      gfs
/dev/gnbd/gfs01                      gfs
/dev/gnbd/gfs02                      gfs
/dev/gnbd/gfs03                      gfs

/dev/sda                            <- partition information ->
/dev/sda1                          <- EXT2/3 filesystem ->
/dev/sda2                          <- EXT2/3 filesystem ->
/dev/sda3                          <- swap device ->

```

```
/dev/sda4          <- partition information ->

/dev/sda5          <- EXT2/3 filesystem ->

/dev/sda6          <- EXT2/3 filesystem ->

/dev/sda7          <- EXT2/3 filesystem ->
```

(f) Active the pools on all nodes

* 이 단계에서 어떤 한 node 가 reboot 될 때마다 수행되어야 한다. 그렇지 않으면 pool devices 를 access 할 수 없을지도 모른다.

* pool_assemble -a 를 c00-c07,front 에서 수행한다.

* c01(ccs)에서 아래와 같이 ccscd 가 먼저 떠 있어서 c01 의 gnbd_cca pool 이 active 안될 수도 있으므로 확인하고, ccscd 가 아래와 같이 떠 있을 때, ccscd processor 를 죽이고 "pool_assemble -r"하고 난후 "pool_assemble -a"를 실행한다.

```
[root@compute-0-1 root]# ps -ef | grep gnbd

root      5318      1  0 16:43 ?          00:00:00 ccscd -d /dev/pool/gnbd_cca
root      5961      1  0 16:47 ?          00:00:00 [gnbd_servd]
root      6612    6553  0 17:19 pts/1    00:00:00 grep gnb

[root@frontend-0 opt]# pool_assemble -a

gfs assembled.

gnbd_cca assembled.
```

```
[root@frontend-0 opt]# pool_tool -s
```

Device	Pool Label
=====	=====
/dev/pool/gfs	<- unknown ->
/dev/pool/gnbd_cca	<- unknown ->
/dev/gnbd/cca	gnbd_cca
/dev/gnbd/gfs00	gfs
/dev/gnbd/gfs01	gfs
/dev/gnbd/gfs02	gfs
/dev/gnbd/gfs03	gfs
/dev/sda	<- partition information ->
/dev/sda1	<- EXT2/3 filesystem ->
/dev/sda2	<- EXT2/3 filesystem ->
/dev/sda3	<- swap device ->
/dev/sda4	<- partition information ->
/dev/sda5	<- EXT2/3 filesystem ->
/dev/sda6	<- EXT2/3 filesystem ->
/dev/sda7	<- EXT2/3 filesystem ->

(g) Create CCS files

c01 에서 실행[참고: Giga Test 는 c03 에서 했음]


```
# mkdir /root/gnbd
# cd /root/gnbd
# vi cluster.ccs
cluster {
  name = "gnbd"
  lock_gulm {
    servers = [ "c01" ] # lockserver 의 이름을 적는다
    heartbeat_rate = 30.0
    allowed_misses = 2
  }
}
또는
cluster {
  name = "gnbd"
  lock_gulm {
    servers = [ "compute-0-1.local" ]
    heartbeat_rate = 30.0
    allowed_misses = 2
  }
}
로 작성한다.
```

```
# vi /root/gnbd/node.ccs
nodes {
  front {
    ip_interfaces {
      eth0 = "10.0.0.1"
    }
  }
  c00 {
    ip_interfaces {
      eth0 = "10.255.255.254"
    }
    fence {
      server {
        gnbd {
          ipaddr = "10.255.255.254"
        }
      }
    }
  }
}
```

```

    }
    }
}
c01 {
    ip_interfaces {
        eth0 = "10.255.255.253"
    }
    fence {
        server {
            gnbd {
                ipaddr = "10.255.255.253"
            }
        }
    }
}
c02 {
    ip_interfaces {
        eth0 = "10.255.255.252"
    }
    fence {
        server {
            gnbd {
                ipaddr = "10.255.255.252"
            }
        }
    }
}
c03 {
    ip_interfaces {
        eth0 = "10.255.255.251"
    }
    fence {
        server {
            gnbd {

```

```

                                ipaddr = "10.255.255.251"
                                }
                            }
                        }
                    }

```

```

c04 {
    ip_interfaces {
        eth0 = "10.255.255.250"
    }
    fence {
        server {
            gnbd {
                ipaddr = "10.255.255.250"
            }
        }
    }
}

```

```

c05 {
    ip_interfaces {
        eth0 = "10.255.255.249"
    }
    fence {
        server {
            gnbd {
                ipaddr = "10.255.255.249"
            }
        }
    }
}

```

```

c06 {
    ip_interfaces {
        eth0 = "10.255.255.248"
    }
}

```

```
fence {  
    server {  
        gnbd {
```

```

                                ipaddr = "10.255.255.248"
                                }
                            }
                        }
                    }
                }
            }
        c07 {
            ip_interfaces {
                eth0 = "10.255.255.247"
            }
            fence {
                server {
                    gnbd {
                        ipaddr = "10.255.255.247"
                    }
                }
            }
        }
    }
}

#vi fence.ccs
fence_devices {
    gnbd {
        agent = "fence_gnbd"
        server = "c00"
        server = "c01"
        server = "c02"
        server = "c03"
        server = "c04"
        server = "c05"
        server = "c06"
        server = "c07"
    }
}

```

(h) Create the CCS Archive on the CCA Device

하나의 single Node(c03 에서)에서 한 번만 필요함 , CCS 서버에서 하면 안됨

```
[root@compute-0-3 root]# ccs_tool create /root/gnbd/
/dev/pool/gnbd_cca
Initializing device for first time use... done
```

(i) Start the CCS daemon (ccsd) on all the nodes

```
[root@frontend-0 root]# cluster-fork ccsd -d /dev/pool/gnbd_cca

compute-0-0:
compute-0-1:
compute-0-2:
compute-0-3:
compute-0-4:
compute-0-5:
compute-0-6:
compute-0-7:

[root@frontend-0 root]# cluster-fork ps -ef | grep
ccsd

root      17166      1  0 09:29 ?        00:00:00 ccsd -d /dev/pool/gnbd_cca
root      17159      1  0 09:29 ?        00:00:00 ccsd -d /dev/pool/gnbd_cca
```

root	10790	1	0 09:29 ?	00:00:00 ccsd -d /dev/pool/gnbd_cca
root	10335	1	0 09:29 ?	00:00:00 ccsd -d /dev/pool/gnbd_cca
root	10625	1	0 09:29 ?	00:00:00 ccsd -d /dev/pool/gnbd_cca
root	9740	1	0 09:29 ?	00:00:00 ccsd -d /dev/pool/gnbd_cca
root	10561	1	0 09:29 ?	00:00:00 ccsd -d /dev/pool/gnbd_cca
root	9641	1	0 09:29 ?	00:00:00 ccsd -d /dev/pool/gnbd_cca

(j) At each node, start the the LOCK_GULM server

모든 노드에서 실행

```
[root@frontend-0 root]# cluster-fork lock_gulmd
```

c01: LOCK_GULM 서버의 상태는 gulm_tool 명령을 이용하여 볼 수 있다.

```
[root@compute-0-3 gnbld]# gulm_tool nodelist c03
```

```
Name: compute-0-2.local
```

```
ip      = 10.255.255.252
```

```
state = Logged in
```

```
mode = Client
```

```
missed beats = 0
```

```
last beat = 1106923370760569
```

```
delay avg = 0
```

```
max delay = 0
```

```
Name: compute-0-0.local
```

```
ip      = 10.255.255.254
```

```
state = Logged in
```

```
mode = Client
```

```
missed beats = 0
```

```
last beat = 0
```

```
delay avg = 0
```

```
max delay = 0
```

```
Name: compute-0-6.local
```

```
ip      = 10.255.255.248
```

```
state = Logged in
```


mode = Client

missed beats = 0

last beat = 0

delay avg = 0

max delay = 0

Name: compute-0-4.local

ip = 10.255.255.250

state = Logged in

mode = Client

missed beats = 0

last beat = 0

delay avg = 0

max delay = 0

Name: compute-0-5.local

ip = 10.255.255.249

state = Logged in

mode = Client

missed beats = 0

last beat = 0

delay avg = 0

max delay = 0

Name: compute-0-3.local

ip = 10.255.255.251

state = Logged in

mode = Client

missed beats = 0

last beat = 1106923370850224

delay avg = 0

max delay = 0

Name: compute-0-1.local

ip = 10.255.255.253

state = Logged in

mode = Master

```
missed beats = 0  
  
last beat = 1106923370659607  
  
delay avg = 0  
  
max delay = 0
```

```
Name: compute-0-7.local
```

```
ip      = 10.255.255.247
```

```
state = Logged in
```

```
mode = Client
```

```
missed beats = 0
```

```
last beat = 0
```

```
delay avg = 0
```

```
max delay = 0
```

```
[root@compute-0-3 gkbd]#
```

(k) Create the GFS file systems

C01 에서 실행해야함

```
[root@compute-0-1 root]# gfs_mkfs -p lock_gulm -t gnbd:gfs -j 4  
/dev/pool/gfs
```

gnbd 는 cluster.ccs 의 Name 과 맞아야 함

This will destroy any data on /dev/pool/gfs.

It appears to contain a GFS filesystem.

Are you sure you want to proceed? [y/n] y

Device: /dev/pool/gfs

Blocksize: 4096

Filesystem Size: 20938428

Journals: 1

Resource Groups: 320

Locking Protocol: lock_gulm

Lock Table: Adam:gfs

Syncing...

All Done

(l) Mount the GFS file systems on all the nodes

모든 노드에서 실행

```
[root@compute-0-6 root]# mount -t gfs /dev/pool/gfs /gfs
```

5. Gigabit Ethernet 로 Setup 하기

설정 환경:

```
가. ethernet Test
GPS Server : c00, c02
CCS Server : c01
설정파일 위치 :c03: /root/gnbd/test00_ethernet
나. giga Test
GPS Server : c04, c06
CCS Server : c01
설정파일 위치:c03: /root/gnbd/test01_giga
```

(a) Giga 로 setup 할 때 hostname 은 `uname -n` 에서 나오는 hostname 그대로 해야한다.

P.63 참고

NodeName : The Linux hostname of the node.

Note: Make sure that you use the Linux hostname and that the primary IP address of the node is associated with the hostname. Specifying a *NodeName* other than the Linux hostname (for example the interface name) can cause unpredictable results .especially if the node is connected to multiple networks. To determine the hostname of a node, you can use the `uname -n` command at the node. To verify the IP address associated with the hostname, you can issue a ping command to the hostname.

따라서 /etc/hosts 파일을 수정하여야 한다.

```
#
```

```
# Do NOT Edit (generated by dbreport)
```

```
#
127.0.0.1      frontend-0.public      localhost.localdomain  localhost
10.0.0.1      frontend-0.local frontend-0 front

# warning: should be frontend-0-0

10.255.255.254 compute-0-0g.local compute-0-0g c00g
10.255.255.253 compute-0-1g.local compute-0-1g c01g
10.255.255.252 compute-0-2g.local compute-0-2g c02g
10.255.255.251 compute-0-3g.local compute-0-3g c03g
10.255.255.250 compute-0-4g.local compute-0-4g c04g
10.255.255.249 compute-0-5g.local compute-0-5g c05g
10.255.255.248 compute-0-6g.local compute-0-6g c06g
10.255.255.247 compute-0-7g.local compute-0-7g c07g

#

# Giga Port

#

20.0.0.1      compute-0-0.local compute-0-0 c00
20.0.0.2      compute-0-1.local compute-0-1 c01
20.0.0.3      compute-0-2.local compute-0-2 c02
20.0.0.4      compute-0-3.local compute-0-3 c03
20.0.0.5      compute-0-4.local compute-0-4 c04
20.0.0.6      compute-0-5.local compute-0-5 c05
```

20.0.0.7 compute-0-6.local compute-0-6 c06

20.0.0.8 compute-0-7.local compute-0-7 c07

(b) c03:/root/gnbd/nodes.ccs 에서 ethernet interface 를 eth0 에서 eth1 으로 수정한다.

```
nodes {
  compute-0-0.local {
    ip_interfaces {
      eth1 = "20.0.0.1"
    }
    fence {
      server {
        gnbd {
          ipaddr = "20.0.0.1"
        }
      }
    }
  }
  compute-0-1.local {
    ip_interfaces {
      eth1 = "20.0.0.2"
    }
    fence {
      server {
        gnbd {
          ipaddr = "20.0.0.2"
        }
      }
    }
  }
  compute-0-2.local {
    ip_interfaces {
      eth1 = "20.0.0.3"
    }
    fence {
      server {
```

```

                                gnbd {
                                    ipaddr = "20.0.0.3"
                                }
                            }
                        }
                    }
compute-0-3.local {
    ip_interfaces {
        eth1 = "20.0.0.4"
    }
    fence {
        server {
            gnbd {
                ipaddr = "20.0.0.4"
            }
        }
    }
}
compute-0-4.local {
    ip_interfaces {
        eth1 = "20.0.0.5"
    }
    fence {
        server {
            gnbd {
                ipaddr = "20.0.0.5"
            }
        }
    }
}
compute-0-5.local {
    ip_interfaces {
        eth1 = "20.0.0.6"
                                21,6-41      37%

        server {
            gnbd {

```



```
                                ipaddr = "20.0.0.4"
                                }
                                }
                                }
}
compute-0-4.local {
    ip_interfaces {
        eth1 = "20.0.0.5"
    }
    fence {
        server {
            gnbd {
                ipaddr = "20.0.0.5"
            }
        }
    }
}
compute-0-5.local {
    ip_interfaces {
        eth1 = "20.0.0.6"
    }
    fence {
        server {
            gnbd {
                ipaddr = "20.0.0.6"
            }
        }
    }
}
compute-0-6.local {
    ip_interfaces {
        eth1 = "20.0.0.7"
    }
    fence {
        server {
            gnbd {
```

```

                                ipaddr = "20.0.0.7"
                                }
                            }
                        }
                    }
                compute-0-7.local {
                    ip_interfaces {
                        eth1 = "20.0.0.7"
                    }
                    fence {
                        server {
                            gnbd {
                                ipaddr = "20.0.0.7"
                            }
                        }
                    }
                }
            }
        }
    }

```

(c) reboot 하고 다시 GFS 설정을 실행한다.

```

umount

gulfm_tool shutdown Rabbit

service ccscd stop

pool_assemble -r

```

hosts 파일과 cluster.ccs 를 수정한 후 다음을 실행한다.

```

pool_assemble -a

ccs_tool create /root/gnbd/ /dev/pool/gnbd_cca

cluster-fork ccscd -d /dev/pool/gnbd_cca

cluster-fork lock_gulmd

```

```
gulm_tool nodelist c03
```

위의 작업 수행 후 mount 로 확인한다.