

우 준

GPFS-MC 구축 지침서

<제목 차례>

제1장 개요	1
제2장 로컬 GPFS 클러스터 구축하기	3
1. GPFS 지원 환경	3
2. 다양한 GPFS 구성 방법	3
3. 로컬 GPFS 클러스터 구축	7
가. 구축 시스템 환경	7
나. GPFS 클러스터 구성	8
다. GPFS 설치하기	9
제3장 GPFS-MC 구축하기	21
1. GPFS-MC 구축 환경	21
2. GPFS-MC 구성하기	22
가. SSL 설정	22
나. 키 교환	24
다. 액세스 허용 및 리모트 클러스터와 파일시스템 설정	24
라. 리모트 파일시스템 마운트	26

<표 차례>

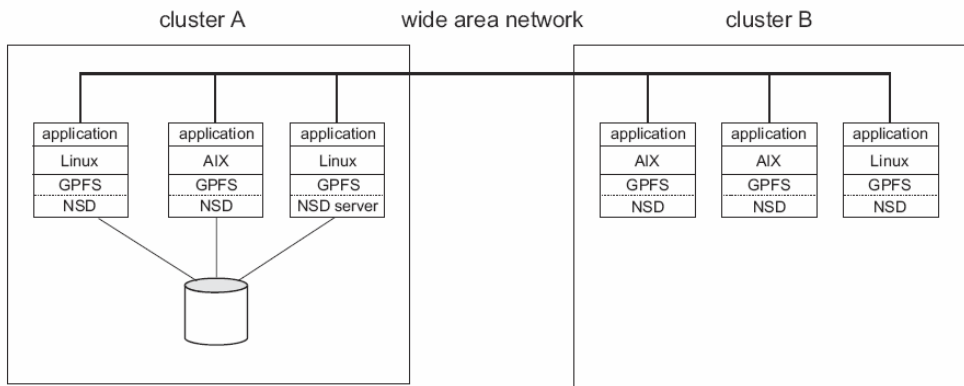
<표 2-1> GPFS 지원 환경	3
<표 3-1> 리모트 파일시스템 액세스 설정을 위한 GPFS-MC 명령어 요약	22

<그림 차례>

<그림 1-1> GPFS-MC 구성	1
<그림 1-2> DEISA GPFS-MC 구축 사례	2
<그림 2-1> SAN 디스크가 연결된 리눅스 클러스터	4
<그림 2-2> 네트워크로 연결된 리눅스 클러스터	4
<그림 2-3> 리눅스와 AIX 클러스터	5
<그림 2-4> 리눅스와 AIX 클러스터	5
<그림 2-5> 리눅스와 AIX 클러스터	6
<그림 2-6> HPS로 연결된 AIX 클러스터	6
<그림 2-7> GPFS 클러스터 구성	8
<그림 2-8> GPFS-MC 테스트 구성 환경	21

제1장 개요

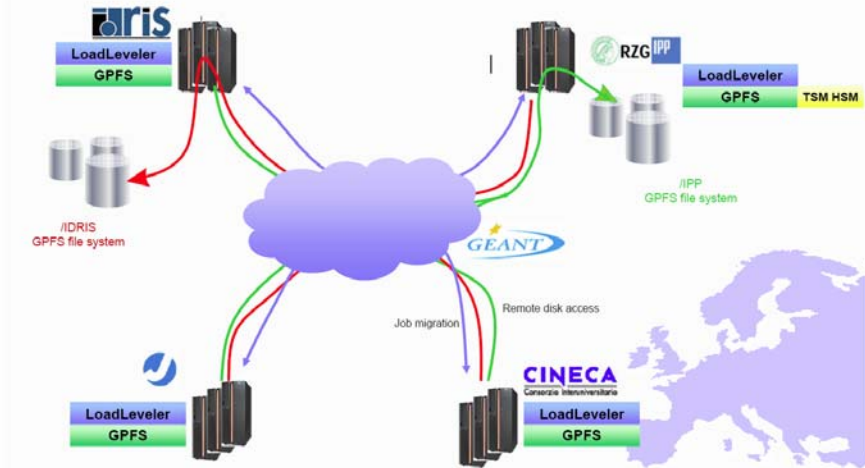
GPFS(General Parallel File System)는 IBM에서 개발하여 다수의 노드로부터 기존 POSIX API를 통해 동시적인 파일엑세스를 가능하게 하는 블록 I/O 기반의 병렬파일시스템이다. <그림 1-1>은 WAN 상의 GPFS cluster A의 NSD서버에 액세스하여 리모트 파일시스템을 GPFS cluster B가 마운트하여 사용하는 구성을 보여주고 있다. GPFS는 이러한 GPFS 멀티클러스터(MC)를 구축할 수 있도록 `mmauth`, `mmremotecluster`, `mmremotefs` 명령을 추가로 제공한다.



<그림 1-1> GPFS-MC 구성

GPFS는 미국의 TeraGrid, NERSC, NCAR, 유럽의 DEISA 등과 같은 해외의 그리드 및 멀티클러스터(MC) 환경에서 글로벌공유파일시스템으로 구축되어 각 서비스 모델의 핵심 구성 요소로 활용되고 있다. <그림1-2>는 유럽의 DEISA 프로젝트에서 GPFS-MC(멀티클러스터)를 통해서 구축된 글로벌공유파일시스템 구축 사례를 보여주고 있다.

LL/GPFS MC Architecture for DEISA



<그림 1-2> DEISA GPFS-MC 구축 사례

본서에서는 WAN 상의 멀티클러스터 환경에서 GPFS-멀티클러스터(MC)를 구축하기 위한 단계별 절차를 실제 구축환경을 예로 들어 이야기 하고 있다. 먼저 각 로컬 사이트에 GPFS 서버 및 클라이언트를 포함한 GPFS 클러스터를 구축하고, 이어서 각 사이트 간의 GPFS 클러스터를 GPFS-MC 환경으로 구성하기 위한 방법을 구체적으로 설명한다.

제2장 로컬 GPFS 클러스터 구축하기

1. GPFS 지원 환경

GPFS는 다양한 플랫폼과 리눅스 및 AIX 위에 설치하여 구동 할 수 있고, 대부분의 네트워크 인터페이스를 지원한다. 2007년 12월 현재 3.2 버전이 가장 최신 버전으로 <표 2-1> 같은 운영체제, 시스템 플랫폼, 네트워크를 지원한다.

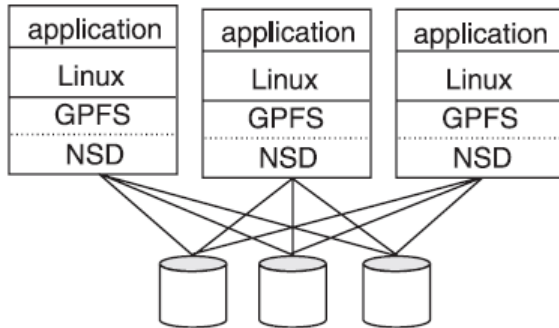
<표 2-1> GPFS 지원 환경

OS	<ul style="list-style-type: none"> ● AIX 5L v5.3, AIX 5L v5.2, ● Red Hat Enterprise Linux 4 and 5 ● SuSE Linux Enterprise Server 9 and 10
플랫폼	<ul style="list-style-type: none"> ● i386, x86_64, PowerPC
네트워크	<ul style="list-style-type: none"> ● Ethernet, 10GB Ethernet, Myrinet, InfiniBand, HPS, SP Switch2

2. 다양한 GPFS 구성 방법

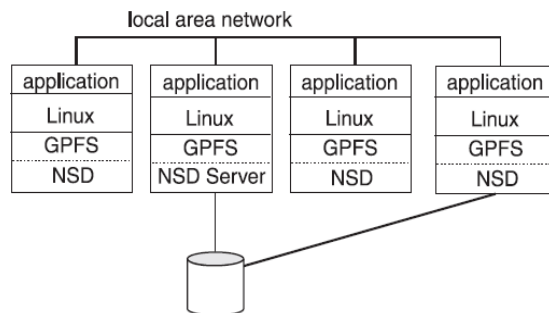
GPFS는 Linux와 AIX 클러스터를 모두 지원하며, Linux와 AIX간에도 NSD 서버를 통해서 I/O를 할 수 있다. <그림2-1>은 리눅스로만 구성되어 있는

GPFS 클라이언트로서 모든 NSD 노드가 SAN디스크에 물리적으로 연결되어 있어 I/O의 요청이 있을 때 물리적으로 직접 접근하게 된다. 따라서 네트워크를 통한 접근이 없으므로 NSD 서버는 존재하지 않는다.



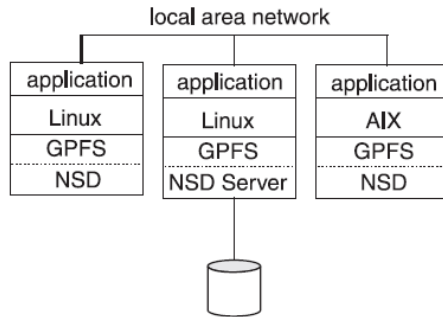
<그림 2-1> SAN 디스크가 연결된 리눅스 클러스터

<그림2-2>는 하나의 NSD 노드만 물리적으로 디스크와 연결되어 있어서 네트워크를 통해 I/O 서비스를 제공하기 위해 NSD 서버로서의 역할을 수행하고, 다른 노드들은 네트워크를 통해서 디스크에 접근을 하게 된다. NSD 서버의 역할을 수행하는 노드에 장애가 일어났을 경우를 대비하여 물리적인 연결을 하나 더 추가하여 NSD 서버에 대한 백업노드로 지정할 수 있으며 가장 우측에 연결되어 있는 NSD가 이것을 보여주고 있다.



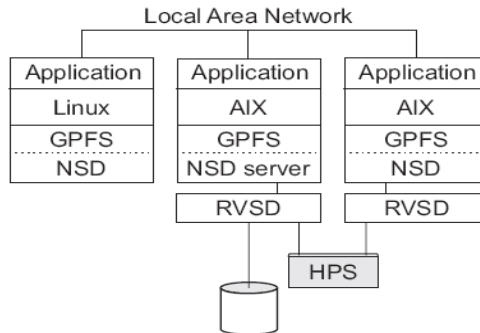
<그림 2-2> 네트워크로 연결된 리눅스 클러스터

<그림2-3>은 리눅스와 AIX가 함께 구성되어 있는 노드가 NSD 서버를 통해서 리눅스 노드에 연결되어 있는 디스크에 접근이 가능함을 보여준다. 이것은 다양한 환경으로 구성되어 있는 클러스터 환경에 호환성과 확장가능성을 높여준다.



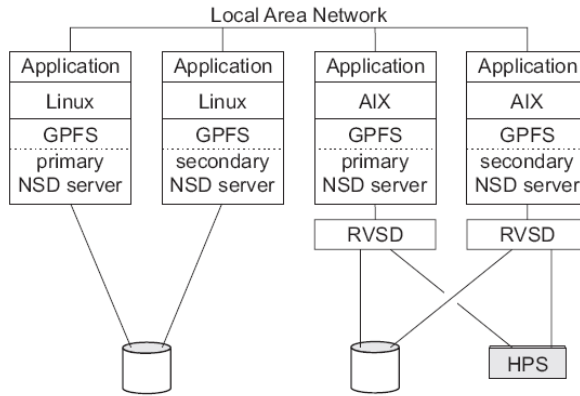
<그림 2-3> 리눅스와 AIX 클러스터

<그림2-4>은 RVSD를 갖는 AIX노드 NSD 서버에 다른 AIX 노드와 리눅스 노드가 네트워크를 통해서 접근 하는 구성을 보여준다. <그림2-3>에서 리눅스 NSD서버에 AIX 노드가 접근 할 수 있는 것처럼 AIX NSD 서버에도 리눅스 노드가 접근 가능하다. AIX NSD 서버도 HPS 스위치를 통해 디스크에 대한 직접 연결을 추가하여 백업노드를 구성할 수 있다.



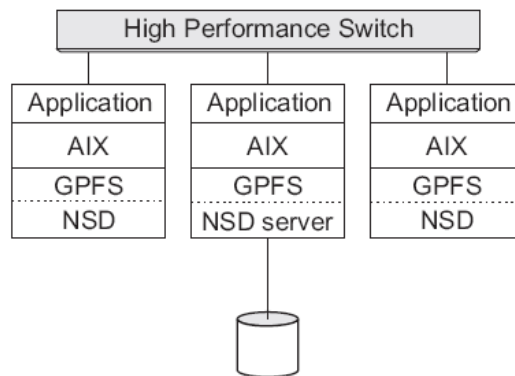
<그림 2-4> 리눅스와 AIX 클러스터

<그림2-5>는 리눅스와 AIX 노드가 모두 각각의 물리적인 디스크를 가지고 NSD서버로 구성되어 있는 것을보여주고 있으며, 상호간에 네트워크를 통해서 접근이 가능하다.



<그림 2-5> 리눅스와 AIX 클러스터

<그림2-6>은 IBM의 HPS(High performance Switch)를 통해 연결한 AIX노드의 구성이며 하나의 노드가 NSD 서버로서의 역할을 수행하며 나머지 노드들은 HPS를 통해서 디스크에 접근한다.



<그림 2-6> HPS로 연결된 AIX 클러스터

2. 로컬 GPFS 클러스터 구축

가. 구축 시스템 환경

로컬 GPFS 클러스터 구축 절차를 좀 더 현실적으로 설명하기 위해 테스트 구축 환경을 기반 위에 단계별로 구축 과정을 이야기 하고자 한다.

(1) 서버 및 클라이언트

① 서버

DELL PE 1950 5대

- CPU : Intel Xeon 2.33GHz 2 CPU(Quad core) X86_64
- MEM : 2GB
- OS : Red Hat Enterprise Linux ES(2.6.9-42.ELsmp)

② 클라이언트

SGI Altix1350 8대(로그인 노드 포함)

- CPU : Intel Itanium 2 1.5GHz 16 CPU IA-64
- MEM : 8GB
- OS : Red Hat Enterprise Linux AS 3.4 (2.4.21-sgi306rp13)

(2) 디스크 스토리지

본 GPFS 클러스터 구축에 사용된 디스크 스토리지는 EMC의 CX-3 기종

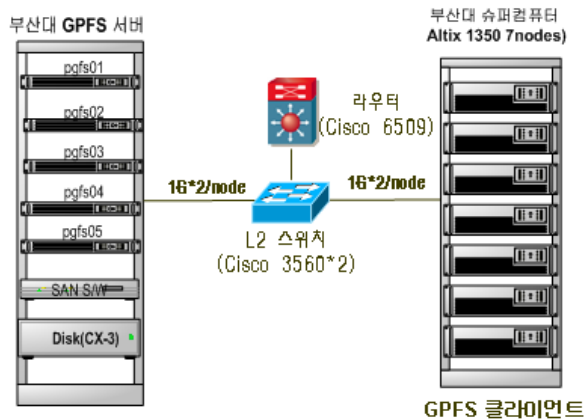
이다. 세부 사양은 아래와 같다.

- 물리디스크 16개(300GB, 15K RPM, FC4) : 4.8TB
- RAID Array : RAID5 [4+P], Global hot-spare 1개
- Logical 드라이브 : 6개
- 기타 RAID 설정
 - 세그먼트 사이즈 : 32KB
 - cache frame size : 16KB
 - Read(60%)/write(40%) cache : enables

(3) 네트워크

- Cisco 3560 스위치(1GBE 24포트) 2대
- 각 노드는 1 GbE 2포트로 로컬 네트워크에 연동됨

나. GPFS 클러스터 구성



<그림 2-7> GPFS 클러스터 구성

다. GPFS 설치하기

(1) GPFS 파일셋 설치하기

아래와 같이 서버에 x86_64 플랫폼을 위한 GPFS 3.2 버전 RPM 파일셋을 설치한다.

```
rpm -ivh gpfs.base-3.2.0-1.x86_64.rpm
rpm -ivh gpfs.docs-3.2.0-1.noarch.rpm
rpm -ivh gpfs.gpl-3.2.0-1.noarch.rpm
rpm -ivh gpfs.msg.en_US-3.2.0-1.noarch.rpm
```

☞ [참고] GPFS 업데이트 파일은 다음 웹페이지에서 다운로드 할 수 있다.

<http://www14.software.ibm.com/webapp/set2/sas/f/gpfs/download/systemx.html>

☞ [참고] 다수의 노드에 GPFS RPM 파일셋을 설치하기 위한 셸 명령어

```
# cat gpfs.allnodes |xargs -i ssh {}
/ rpm -Uvh /home2/gpfs/gpfs.base-3.2.0-1.x86_64.rpm
```

gpfs.allnodes 파일에는 아래와 같이 설치 대상 노드들의 호스트 이름을 가지고 있다.

```
gpfs01
gpfs02
gpfs03
.....
```

(2) GPFS portability layer 빌드하기

리눅스 커널과 GPFS 커널 간 통신이 가능하도록 설치 하드웨어 및 리눅스 배포판을 기반으로 하는 custom portability 모듈을 빌드해 주어야 한다.

```
cd /usr/lpp/mmfs/src
export SHARKCLONEROOT=/usr/lpp/mmfs/src
make Autoconfig
#Can check /usr/lpp/mmfs/src/config/site.mcr
make World
make InstallImages
```

☞ [참고] 컴파일 되어 설치되는 파일들

다수의 노드에 설치할 때는 하나의 노드에서 컴파일하여 다음 파일들을 다른 노드의 /usr/lpp/mmfs/bin 디렉토리에 복사하는 것이 편리하다.

```
mmfslinux
mmfs26
tracedev
dumpconv
lxtrace
```

(3) GPFS 클러스터 생성하기

아래와 같이 primary configuration 서버로 pgfs01 노드를 backup configuration 서버로 pgfs02로 지정하고, 노드 디스크립터 파일인 hosts 파일에 지정된 노드들을 GPFS 클러스터 노드로 구성한다. -C 옵션은 GPFS 클러스터의 이름을 지정한다.

```
[root@pgfs01 ~] mmcrcluster -n /home2/gpfs/hosts -p pgfs01 -s pgfs02 -r /usr/bin/ssh -R /usr/bin/scp -C pgfs -A
```

```
2007. 05. 21. (월) 11:12:43 KST: mmcrcluster: Processing node pgfs1
```

```
2007. 05. 21. (월) 11:12:43 KST: mmcrcluster: Processing node pgfs2
```

```
2007. 05. 21. (월) 11:12:50 KST: mmcrcluster: Processing node pgfs3
```

```
2007. 05. 21. (월) 11:12:52 KST: mmcrcluster: Processing node pgfs4
```

```
2007. 05. 21. (월) 11:12:52 KST: mmcrcluster: Processing node pgfs5
```

```
mmcrcluster: Command successfully completed
```

```
mmcrcluster: Propagating the cluster configuration data to all affected nodes. This is an asynchronous process.
```

☞ [참고] 노드 디스크립터 파일 [/home2/gpfs/hosts]

콜론(:)으로 구분된 첫 번째 필드는 노드이름을 표시하며 /etc/hosts 파일에 아래 정의된 호스트 이름에 대한 IP가 각각 한 개씩만 존재해야 한다. 두 번째 필드는 '-'로 구분되어 노드의 역할(manager|client)과 쿼럼노드 여부(quorum|noquorum)를 표시한다. 쿼럼(quorum) 노드는 GPFS가 정상적으로 운영되는 데 필요한 노드의 수를 결정하기 위해서 지정된다. 세 번째 필드는 노드들 사이에서 통신하기 위해 관리자 명령에 의해서 사용되는 노드의 이름으로 구성되는 옵션 필드이다.

```
pgfs1:manager-quorum:
```

```
pgfs2:manager-quorum:
```

```
pgfs3:manager-quorum:
```

```
pgfs4:manager-quorum:
```

```
pgfs5:manager-quorum:
```

(4) GPFS 클러스터 노드 추가

클라이언트 노드들을 별도로 추가하기 위해서는 아래와 같이 -N 옵션 뒤에 노드 디스크립터 파일 혹은 노드 이름을 명시한다.

```
[root@pgfs01 ~] mmaddnode -N hosts
```

☞ [참고] GPFS 클러스터 노드 구성 확인

```
[root@pgfs01 ~] mmlscluster
```

GPFS cluster information

```
=====
GPFS cluster name:      pgfs.pgfs01
GPFS cluster id:       9676957302179964142
GPFS UID domain:      pgfs.pgfs01
Remote shell command:  /usr/bin/ssh
Remote file copy command: /usr/bin/scp
```

GPFS cluster configuration servers:

```
-----
Primary server:  pgfs01
Secondary server: pgfs02
```

Node	Daemon node name	IP address	Admin node name	Designation
1	pgfs01	134.75.117.101	pgfs01	quorum-manager
2	pgfs02	134.75.117.102	pgfs02	quorum-manager
3	pgfs03	134.75.117.103	pgfs03	quorum-manager
4	pgfs04	134.75.117.104	pgfs04	quorum-manager
5	pgfs05	134.75.117.105	pgfs05	quorum-manager
6	pdaisy	134.75.117.130	pdaisy	
7	pdaisy1	134.75.117.131	pdaisy1	
8	pdaisy2	134.75.117.132	pdaisy2	
9	pdaisy3	134.75.117.133	pdaisy3	
10	pdaisy5	134.75.117.135	pdaisy5	
11	pdaisy6	134.75.117.136	pdaisy6	
12	pdaisy7	134.75.117.137	pdaisy7	
13	pdaisy4	134.75.117.134	pdaisy4	

(5) NSD 구성하기

RAID 혹은 LVM 등으로 구성된 디스크 디바이스로부터 NSD를 구성하기 위해서는 먼저 아래 [참고] 항목에서 설명한 바와 같이 디스크 디스크립터 파일을 생성하고, mmcrnsd 명령의 -F 옵션으로 지정하여 NSD를 생성한다. -v 옵션은 디스크 디바이스가 이미 NSD로 포맷되었는지를 확인할 지 여부를 지정한다.

```
[root@pgfs01 ~] mmcrnsd -F disk -v no
mmcrnsd: Processing disk sgd
mmcrnsd: Propagating the cluster configuration data to all
          affected nodes. This is an asynchronous process.
mmlnsd
```

☞ [참고] 디스크 디스크립터 파일

아래와 같은 디스크 디스크립터 파일을 생성하고 나서 NSD를 구성한다. 콜론(:)으로 구분되는 첫 번째 필드는 디스크 디바이스의 이름을 지정한다. 두 번째와 세 번째 필드는 각각 primary와 backup NSD 서버 리스트를 설정한다. 최근 GPFS 3.2 버전부터는 두 번째 필드에 콤마(,)로 구분하여 NSD 서버를 최대 8대 까지 지정하여 앞에서부터 차례로 서버 장애 시 백업 서버의 역할을 담당하도록 구성할 수 있다. 네 번째 필드는 4가지(dataAndMetadata|dataOnly|metadataOnly|descOnly)로 NSD의 역할을 정의할 수 있다.

```
# cat disk
emcpowerg:pgfs01:pgfs02:dataAndMetadata:::
emcpowerj:pgfs02:pgfs03:dataAndMetadata:::
emcpowerj:pgfs03:pgfs04:dataAndMetadata:::
emcpowerk:pgfs04:pgfs05:dataAndMetadata:::
emcpowerl:pgfs05:pgfs01:dataAndMetadata:::
```

☞ [참고] 디스크 디바이스를 포함한 NSD 구성확인 하기

```
[root@pgfs01 ~]# mmlsnsd
```

File system	Disk name	NSD servers
pgfs	gpfs20nsd	pgfs01,pgfs02
pgfs	gpfs21nsd	pgfs02,pgfs03
pgfs	gpfs22nsd	pgfs03,pgfs04
pgfs	gpfs23nsd	pgfs04,pgfs05
pgfs	gpfs24nsd	pgfs05,pgfs04

(6) pagepool 설정하기

GPFS는 버퍼 캐시의 용도로 페이지-풀(pagepool)을 사용하며, 64비트 LINUX/AIX O/S와 GPFS V.3.2에서는 최대 256GB까지 설정할 수 있다.

```
[root@pgfs01 ~] mmchconfig pagepool=1G
```

```
mmchconfig: Command successfully completed
```

```
mmchconfig: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

```
[root@pgfs01 ~]# mmlsconfig
```

```
Configuration data for cluster pgfs.pgfs01:
```

```
-----
clusterName pgfs.pgfs01
clusterId 9676957302179964142
clusterType lc
autoload yes
minReleaseLevel 3.2.0.1
dmapiFileHandleSize 32
pagepool 1G
cipherList AUTHONLY
maxMBpS 150
distributedTokenServer no
```

```
File systems in cluster pgfs.pgfs01:
```

```
-----
/dev/pgfs
```

☞ [참고] GPFS 구성 확인하기

```
[root@pgfs01 ~] mmlsconfig
Configuration data for cluster pgfs.pgfs01:
-----
clusterName pgfs.pgfs01
clusterId 9676957302179964142
clusterType lc
autoload yes
minReleaseLevel 3.2.0.1
dmapiFileHandleSize 32
pagepool 1G
cipherList AUTHONLY
maxMBpS 150
distributedTokenServer no

File systems in cluster pgfs.pgfs01:
-----
(none)
```

☞ [참고] NSD 제거하기

NSD를 제거하기 전에 GPFS 데몬을 shutdown 시켜야 한다.

```
[root@pgfs01 ~] mmdelnsd -F disk
mmdelnsd: Processing disk gpfs1nsd
mmdelnsd: Processing disk gpfs2nsd
mmdelnsd: Processing disk gpfs3nsd
mmdelnsd: Processing disk gpfs4nsd
mmdelnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

(8) GPFS 데몬 구동

```
[root@node01 gpfs]# mmstartup -a
2007. 04. 20. (금) 18:25:07 KST: mmstartup: Starting GPFS ...
takeOverSdrServ yes
```

☞ [참고] GPFS 데몬 구동 및 종료 관련 로그 확인

① 정상 구동시 로그 화면

```
[root@pgfs01 ~] cat /var/adm/ras/mmfs.log.latest

2007. 04. 23. (월) 14:40:21 KST runmmfs starting
Removing old /var/adm/ras/mmfs.log.* files:
Unloading modules from /usr/lpp/mmfs/bin
Loading modules from /usr/lpp/mmfs/bin
Module                Size  Used by
mmfslinux              189316  1 mmfs
tracedev               17952   2 mmfs,mmfslinux
Removing old /var/mmfs/tmp files:
Mon Apr 23 14:40:22 2007: mmfsd initializing. {Version: 3.1.0.11   Built: Apr  6 2007
11:07:44} ...
Mon Apr 23 14:40:24 2007: Connecting to 172.40.0.12 pgfs01
Mon Apr 23 14:40:24 2007: Connected to 172.40.0.12 pgfs02
.....
Mon Apr 23 14:40:25 2007: mmfsd ready
2007. 04. 23. (월) 14:40:25 KST: mmcommon mmfsup invoked
```

② 비정상 구동 시 로그 화면

```
[root@pgfs01 ~] cat /var/adm/ras/mmfs.log.latest
2007. 04. 20. (금) 18:44:50 KST runmmfs starting
Removing old /var/adm/ras/mmfs.log.* files:
Unloading modules from /usr/lpp/mmfs/bin
Loading modules from /usr/lpp/mmfs/bin
insmod: error inserting '/usr/lpp/mmfs/bin/tracedev': -1 Invalid module format
Unloading modules from /usr/lpp/mmfs/bin
2007. 04. 20. (금) 18:44:50 KST runmmfs: error in loading or unloading the mmfs
kernel extension
2007. 04. 20. (금) 18:44:50 KST runmmfs: stopping GPFS
```

(9) GPFS 파일시스템 생성

```
[root@pgfs01 ~] mmcrfs /mnt/pgfs pgfs -F disk -A yes -B 512K -v no
The following disks of pgfs will be formatted on node pgfs01:
  gdfs1nsd: size 224395264 KB
  gdfs2nsd: size 224395264 KB
  gdfs3nsd: size 224395264 KB
  gdfs4nsd: size 225096704 KB
Formatting file system ...
Disks up to size 1.1 TB can be added to storage pool 'system'.
Creating Inode File
  33 % complete on Mon May 21 11:47:23 2007
  67 % complete on Mon May 21 11:47:28 2007
 100 % complete on Mon May 21 11:47:33 2007
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
  34 % complete on Mon May 21 11:47:55 2007
  70 % complete on Mon May 21 11:48:00 2007
 100 % complete on Mon May 21 11:48:04 2007
Completed creation of file system /devpgfs.
mmcrfs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

※ mmcrfs 명령 관련 옵션 설명

```
-A (yes|no|automount)
  yes : GPFS daemon 시작 시 mount
  no  : manual mount
  automount : 파일시스템 처음 액세스 할 때 mount

-B Blocksize
  16, 64, 256KB(default), 512KB, 1MB, 2MB, 4MB

-v (yes|no)
  -F 옵션으로 지정된 디스크가 기존 파일시스템에 속해 있는지의 확인 여부 지정한다. 해당 디스크들이 다른 파일시스템에서 사용되었지만 새로운 파일시스템에서 사용하고자 할 때는 반드시 -v no 로 지정해야 한다.
```

(10) GPFS 파일시스템 삭제

파일시스템을 삭제하기 전에 반드시 전체 노드에서 파일시스템을 umount 해야 한다.

```
[root@pgfs01 ~] mmdelfs pgfs
All data on following disks of linuxlv will be destroyed:
  gpfs1nsd
  gpfs2nsd
  gpfs3nsd
  gpfs4nsd
Completed deletion of file system /dev/pgfs
mmdelfs: Propagating the cluster configuration data to all
  affected nodes. This is an asynchronous process.
```

[참고] GPFS 파일시스템 구성 확인하기

```
[root@pgfs01 ~] mmlsfs pgfs
flag value          description
-----
-f 8192             Minimum fragment size in bytes
-i 512              Inode size in bytes
-l 16384            Indirect block size in bytes
-m 2                Default number of metadata replicas
-M 2                Maximum number of metadata replicas
-r 1                Default number of data replicas
-R 2                Maximum number of data replicas
-j cluster          Block allocation type
-D nfs4             File locking semantics in effect
-k all              ACL semantics in effect
-a 1048576           Estimated average file size
-n 32               Estimated number of nodes that will mount file system
-B 262144            Block size
-Q user:group:fileset Quotas enforced
  none              Default quotas enabled
-F 2613091           Maximum number of inodes
-V 10.00 (3.2.0.0)  File system version
-u yes              Support for large LUNs?
-z no               Is DMAPI enabled?
-L 8388608           Logfile size
-E yes              Exact mtime mount option
-S no               Suppress atime mount option
-K whenpossible     Strict replica allocation option
-P system            Disk storage pools in file system
-d gpfs20nsd:gpfs21nsd:gpfs22nsd:gpfs23nsd:gpfs24nsd  Disks in file system
-A no               Automatic mount option
-o none              Additional mount options
-T /mnt/pgfs         Default mount point
```

(11) GPFS 파일시스템 마운트

```
[root@pgfs01 ~] mmmount pgfs -a
```

2007. 05. 21. (월) 11:51:34 KST: mmmount: Mounting file systems ...

```
# df -k
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/mapper/VolGroup00-LogVol00	138271784	88404252	42843712	68%	/
/dev/sda1	101086	14485	81382	16%	/boot
none	1027704	0	1027704	0%	/dev/shm
/dev/pgfs	2675799552	14019840	2661779712	1%	/mnt/pgfs
/dev/kgfs	6920601600	47813376	6872788224	1%	/mnt/kgfs
/dev/kigfs	9358213120	1720819712	7637393408	19%	/mnt/kigfs

☞ [참고] GPFS 데몬 종료

```
[root@pgfs01 ~] mmshutdown -a
```

☞ [참고] GPFS 파일시스템 umount

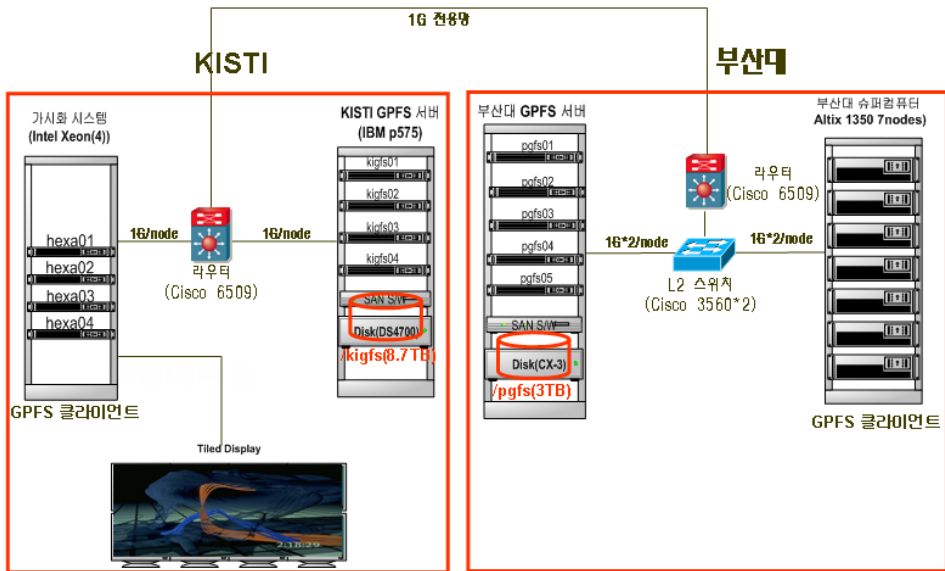
```
[root@pgfs01 ~] mmumount pgfs -a
```

2007. 05. 21. (월) 13:58:56 KST: mmumount: Unmounting file systems ...

제3장 GPFS-MC 구축하기

1. GPFS-MC 구축 환경

GPFS 멀티클러스터(MC) 구축을 위해 <그림 3-1>과 같은 테스트 화면을 테스트 구성 환경을 예제로 하여 부산대와 KISTI에 각각 구축된 GPFS 클러스터 간 리모트 클러스터의 파일시스템을 상호 마운트 하기 위한 구성을 설명하고자 한다. 제2장에서 설명한 바에 따라 KISTI와 부산대에 이미 GPFS 클러스터가 각각 구축되어 있다고 가정하고, 멀티 클러스터 구축 과정을 하나씩 살펴보자.



<그림 2-8> GPFS-MC 테스트 구성 환경

2. GPFS-MC 구성하기

<표 3-1>은 본서에서 예제로 사용하고 있는 테스트 환경에 대한 멀티 클러스터를 구성을 위한 명령을 요약하여 설명하고 있다. 이어서 각 과정을 하나씩 살펴보자.

KISTI GPFS 클러스터(kigfs.kigfs01)	부산대 GPFS 클러스터(pgfs.pgfs01)
● mmauth genkey new	● mmauth genkey new
● mmchconfig cipherList=AUTHONLY	● mmchconfig cipherList=AUTHONLY
● mmauth add pgfs.pgfs01 ...	● mmremotecluster add kigfs.kigfs01 ...
● mmauth grant pgfs.pgfs01 -f kigfs ...	● mmremotefs add kigfs -f kigfs -C kigfs.kigfs01 -T /mnt/kigfs

<표 3-1> 리모트 파일시스템 액세스 설정을 위한 GPFS-MC 명령어 요약

가. SSL 설정

리모트 파일시스템을 마운트하기 위해서는 SSL에서 제공하는 공개키를 기반으로 한 상호 인증 절차가 필요하다. 이를 위해서 먼저 KISTI와 부산대 GPFS 클러스터에서 각각 공개키를 생성하고 인증방식을 설정해야 한다.

(1) 공개키 생성

공개키는 /var/mmfs/ssl 디렉터리에 id_rsa.pub라는 이름으로 생성된다.

```
[root@pgfs01 ~] mmauth genkey new
Generating RSA private key, 512 bit long modulus
.....+++++
....+++++
e is 65537 (0x10001)
writing RSA key
genkeyData1          100% 1317      1.3KB/s   00:00

mmauth: Command successfully completed
```

☞ [참고] 공개키(id_rsa.pub) 내용

```
[root@pgfs01 ssl]# cat id_rsa.pub
clusterName=pgfs.pgfs01
clusterID=9676957302179964142
genkeyFormat=2
genkeyCompatibleFormat=2
keyGenNumber=1
keyDigest=16a535a38822bd73daa591235d6d5029aabc318b
publicKey=
-----BEGIN PUBLIC KEY-----
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBALmDXVqMuvn404U6il1Uidd7Ztp/fzXg
GAyBTTtq5JkZKwyVTONJJd7taCLP4ucJmH8r5vCxSy3LmIhaqDbW20sCAwEAAQ==
-----END PUBLIC KEY-----
certificate=
-----BEGIN CERTIFICATE-----
MIIBHDCBx6ADAgECAgEAMA0GCSqGSIb3DQEBAUAMBYxFDASBgNVBAMTC3BnZnMu
cGdmczAxMCAxDTA3MTAxMjA3NDIzOFoYDzlwNTIxMDAyMDc0MjM0WjAwMRQwEgYD
VQQDEwtwZ2ZzLnBnZnMwMTBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQC5g11ajLr5
+NOFOoiNVJXXe2baf3814BgMgU07auSZGSsMIUzpySxe7Wgiz+LnCZh/K+bwsUst
y5pYWqg21ttLAgMBAAEwDQYJKoZIhvcNAQEEBQADQQAktZFbnfI8jjRAKlhf4X0n
1ee3RXUtl/HTLkZv45FL1ZjdmTr86XaWEYgoIL4qgGkQ7TxbSA8IliuWyvXItJW2
-----END CERTIFICATE-----
```

(2) authentication 방식 변경

```
[root@pgfs01 ~] mmchconfig cipherList=AUTHONLY
Verifying GPFS is stopped on all nodes ...
mmchconfig: Command successfully completed
mmchconfig: 6027-1371 Propagating the cluster configuration
data to all affected nodes. This is an asynchronous process.
```

☞ [참고] 권한(인증) 정보 확인하기

```
[root@pgfs01 ~] testbed2:/var/mmfs/ssl> mmauth show
Cluster name:      pgfs.pgfs01 (this cluster)
Cipher list:      AUTHONLY
SHA digest:       7a4a5d068d323da3fe6b7b60936b73ab30a8cbc5
File system access: (all rw)
```

나. 키 교환

앞서 생성된 공개키를 scp 명령어 등을 사용하여 GPFS 클러스터 간에 교환한다.

```
[root@pgfs01 ssl] scp id_rsa.pub kigfs01:/tmp/id_rsa.pub.pgfs
id_rsa.pub          100% 808    0.8KB/s  00:00

[root@pgfs01 ssl] scp kigfs01:/var/mmfs/ssl/id_rsa.pub /tmp/id_rsa.pub.kigfs
id_rsa.pub          100% 817    0.8KB/s  00:00
```

다. 액세스 허용 및 리모트 클러스터와 파일시스템 설정

(1) KISTI GPFS 클러스터 설정

부산대 GPFS 클러스터(pgfs.pgfs01)에서 KISTI GPFS 클러스터(kigfs.kigfs01)의 파일시스템을 리모트 마운트하는 구성을 가정하여 KISTI GPFS 클러스터에서 파일시스템에 대한 리모트 액세스 허용 설정을 한다.

```
[root@kigfs01 ]# mmauth add pgfs.pgfs01 -k /tmp/id_rsa.pub.pgfs
mmauth: Command successfully completed
mmauth: Propagating the cluster configuration data to all affected
nodes.
This is an asynchronous process.

[root@kigfs01 ]# mmauth grant pgfs.pgfs01 -f kigfs

mmauth: Granting cluster pgfs.pgfs01 access to file system kigfs:
      access type rw; root credentials will not be remapped.

mmauth: Command successfully completed
mmauth: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

mmauth: Command successfully completed
mmauth: Propagating the cluster configuration data to allaffected
nodes.
This is an asynchronous process.
```

(2) 부산대 GPFS 클러스터 설정

KISTI 파일시스템에 액세스하기 위해서 리모트 클러스터를 설정한다. -n 옵션에 리모트 NSD 서버를 콤마(,)로 구분하여 설정한다.

```
[root@pgfs01 ~] mmremotecluster add kigfs.kigfs01 -k /tmp/id_rsa.pub.kigfs
-n kigfs01,kigfs02,kigfs03,kigfs04

[root@pgfs01 ~] mmremotecluster show all
Cluster name:      kigfs.kigfs01
Contact nodes:    kigfs01,kigfs02,kigfs03,kigfs04
SHA digest:       1ca0a49b43c9b136e5b966a03240ed414ac04e1b
File systems:     kigfs (kigfs)
```

다음은 리모트 클러스터의 파일시스템을 설정한다. 설정된 리모트 파일시스템은 로컬 서버의 파일시스템과 마찬가지로 로컬 GPFS 클러스터의 각 노드에 파일시스템으로 설정된다.

```
[root@pgfs01 ~] mmremotefs add kigfs -f kigfs -C kigfs.kigfs01 -T /mnt/kigfs -A no
```

```
mmremotefs: 6027-1371 Propagating the cluster configuration data to all affected nodes. This is an asynchronous process.
```

```
[root@pgfs01 ~] mmremotefs show all
```

Local Name	Remote Name	Cluster name	Mount Point	Mount Options	Automount
kgfs	kgfs	kgfs.kgfs01	/mnt/kgfs	rw	no
kigfs	kigfs	kigfs.kigfs01	/mnt/kigfs	rw	no

라. 리모트 파일시스템 마운트

```
[root@pgfs01 ssl] mmmount kigfs -a
```

```
2007. 05. 22. (화) 11:56:54 KST: mmmount: Mounting file systems ...
```

☞ [참고] 리모트 파일시스템 마운트 상태 확인

굵은 글씨체로 표시된 노드들이 KISTI의 리모트 파일시스템(kigfs)를 마운트하고 있는 부산대 GPFS 서버 및 클라이언트의 목록이다.

```
[root@pgfs01 ssl] mmlsmount kigfs -L

File system kigfs (kigfs.kigfs01:kigfs) is mounted on 19 nodes:
  134.75.117.51    kigfs02                kigfs.kigfs01
134.75.117.137  pdaisy7                pgfs.pgfs01
134.75.117.101  pgfs01                 pgfs.pgfs01
  134.75.117.50    kigfs01                kigfs.kigfs01
  134.75.117.11    kgfs01                 kigfs.kgfs01
134.75.117.132  pdaisy2                pgfs.pgfs01
  134.75.117.53    kigfs04                kigfs.kigfs01
134.75.117.135  pdaisy5                pgfs.pgfs01
134.75.117.130  pdaisy                 pgfs.pgfs01
134.75.117.131  pdaisy1                pgfs.pgfs01
  134.75.117.52    kigfs03                kigfs.kigfs01
134.75.117.133  pdaisy3                pgfs.pgfs01
134.75.117.105  pgfs05                 pgfs.pgfs01
  134.75.117.12    kgfs02                 kigfs.kgfs01
  134.75.117.13    kgfs03                 kigfs.kgfs01
  134.75.117.14    kgfs04                 kigfs.kgfs01
134.75.117.102  pgfs02                 pgfs.pgfs01
134.75.117.103  pgfs03                 pgfs.pgfs01
134.75.117.104  pgfs04                 pgfs.pgfs01
```

참고문헌

1. GPFS V 3.1 : Concept, Planning, and Installation Guide, IBM
2. GPFS V 3.1 : Advanced Administration Guide, IBM