

Exercise #1A: Install and configure a GPFS cluster

Objectives:

Use the GPFS web based administration tool to install a GPFS cluster

Requirements:

Node names and IP address provided by instructor

Node1: _____

Node2: _____

Account/Cluster name used for exercise

Name: root

Password: _____

ClusterName: _____

Step 1: Initialize the Lab Environment

This lab assumes the rpm packages have already been installed on the servers.

The packages for linux are:

gpfs.base-3.2.1-0.i386.rpm

gpfs.docs-3.2.1-0.noarch.rpm

gpfs.gui-3.2.1-0.i386.rpm

gpfs.src-3.2.1-0.noarch.rpm

gpfs.base-3.2.1-0.i386.update.rpm

gpfs.gpl-3.2.1-0.noarch.rpm

gpfs.msg.en_US-3.2.1-0.noarch.rpm

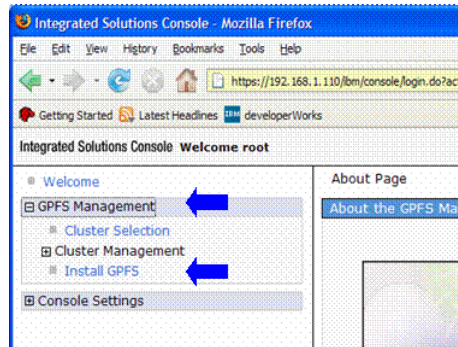
You can install the packages manually by copying them to each node and running rpm or the GPFS install wizard will distribute the packages for you. You need to manually install the packages on at least one node and start the GUI by running:

/etc/init.d/gpfs start

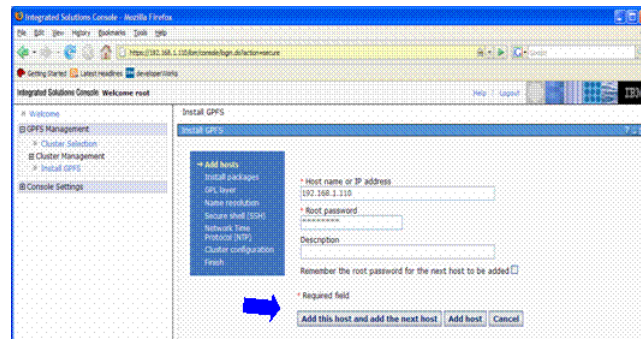
Step 2: Create the GPFS cluster

In this step you will create a GPFS cluster on two nodes using the GPFS web based administration interface.

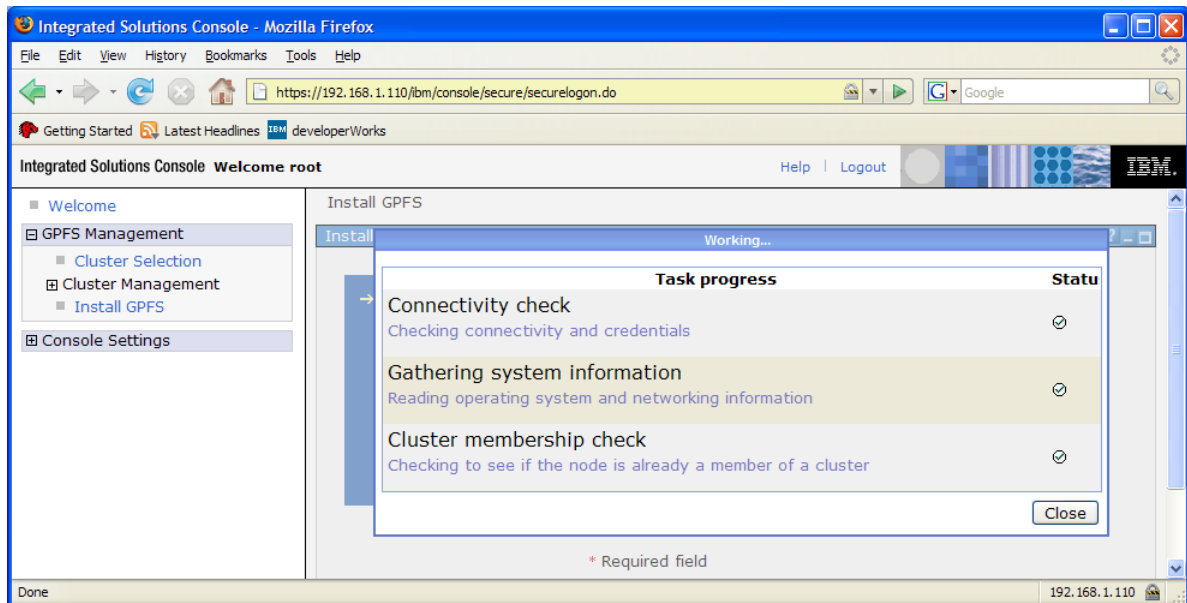
1. Open a web browser to the GPFS
`http://[node1 ip address]/ibm/console`
2. Login using the account and password provided by the instructor.
3. Account: root
Password:
 4. In the navigation pane select "*GPFS Management*" then "*Install GPFS*". This will start the GPFS configuration wizard.



5. Select "Create a new Session". This will take you to the Define hosts page.
6. Under "Defined hosts" click "Add"
 1. Enter the name or ipaddress of node1, and the root password.
 2. Select "Add this host and add the next host."



7. Close the Task process dialogue when completed.



8. Enter the IP address and root password for node2 and select "Add host." Close the Task process dialogue when completed.

9. The hosts have now been added. Select next to go to the Install and Verify Packages page. On this page select "*Check existing package installation.*" Close the task dialog when the check is complete.
10. GPFS Ships an open source component called the GPL layer that allows the support of a wide variety of Linux kernels. The GPL layer installation page checks that the GPL layer is built and installed correctly. If it is not, the installer will complete the build and install. Select "Check existing GPL layer installation". Close the "Task" dialog when the check is complete.
11. GPFS verifies the network configuration of all nodes in the cluster. Select "Check current settings" to verify the network configuration. Close the "Task" dialog when the check is complete.
12. GPFS uses ssh (Or other remote command tool) for some cluster operations. The installer will verify that ssh is configured properly for a GPFS cluster. Select "Check Current Settings" to verify the ssh config. Close the "Task" dialog when the check is complete.
13. It is recommended, though not required, that all the servers synchronize the time using a protocol such as NTP. For this lab we will skip the NTP setup. Choose "*Skip Setup*" to continue.
14. Next, you set the name the GPFS cluster. Enter the cluster name and select "*Next.*"
15. The last step is to define the primary and secondary cluster configuration servers. Since this is a two node cluster we will leave it at the defaults. Select "Next" to continue.
16. Select Next to complete the cluster configuration. Close the "Task" dialog when the configuration is complete.
17. When you select "Finish" you will be directed to the cluster management page.

Integrated Solutions Console - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://192.168.1.110/ibm/console/login.do?action=secure

Getting Started Latest Headlines IBM developerWorks

Integrated Solutions Console **Welcome root** Help Logout

- Welcome
- GPFS Management
 - Cluster Selection
 - Cluster Management
 - Install GPFS
- Console Settings

GPFS Cluster Page

Cluster Selection

This view lists all managed clusters

Remove

Selec	Name	Primary server	Secondary server
<input type="checkbox"/>	Demo1.bvrssg.net	gpfsdemo1.bvrssg.net	gpfsdemo2.bvrssg.net

Total: 1

Mon, 17 Mar 2008 15:46:02

Add an existing cluster to the list of managed clusters

[Add a cluster...](#)

Cluster Overview

Cluster settings

Name:	Demo1.bvrssg.net
ID:	13882347424533378077
Primary server:	gpfsdemo1.bvrssg.net
Secondary server:	gpfsdemo2.bvrssg.net
TCP port:	1191
Remote copy path (RCP):	/usr/bin/scp
Remote shell (RSH) path:	/usr/bin/ssh
User Interface Domain (UID):	Demo1.bvrssg.net

Cluster nodes

Host name	Is manager node	Is quorum node
gpfsdemo1.bvrssg.net	✓	✓
gpfsdemo2.bvrssg.net	✓	✓

Total: 2

Cluster file systems

Name	Mount point
Total: 0	

Cluster disks

Name	Usage	File system
Total: 0		

Done 192.168.1.110

18. The GPFS cluster is now installed and running.

Exercise #1B: Install and configure a GPFS cluster

Objectives

- Verify the system environment
- Create a GPFS cluster
- Define NSD's
- Create a GPFS file system

You will need

- An AIX 5.3 System
 - Very similar to linux config with rpm instead of AIX binary images and Linux admin commands are different
- At least 4 hdisks
- GPFS 3.3 Software with latest PTF

Step 1: Verify Environment

1. Verify nodes properly installed
 1. Check that the oslevel is supported
On the system run oslevel
Check the GPFS FAQ: <http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp>
 2. Is the installed OS level supported by GPFS? Yes No
 3. Is there a specific GPFS patch level required for the installed OS? Yes No
 4. If so what patch level is required? _____
2. Verify nodes configured properly on the network(s)
 1. Write the name of Node1: _____
 2. Write the name of Node2: _____
 3. From node 1 ping node 2
 4. From node 2 ping node 1

If the pings fail, resolve the issue before continuing.
3. Verify node-to-node ssh communications (For this lab you will use ssh and scp for communications)
 1. On each node create an ssh-key. To do this use the command ssh-keygen; if you don't specify a blank passphrase, -N, then you need to press enter each time you are prompted to create a key with no passphrase until you are returned to a prompt. The result should look something like this:

```
# ssh-keygen -t rsa -N "" -f $HOME/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/.ssh'.
Your identification has been saved in /.ssh/id_rsa.
Your public key has been saved in /.ssh/id_rsa.pub.
The key fingerprint is:
7d:06:95:45:9d:7b:7a:6c:64:48:70:2d:cb:78:ed:61
sas@perf3-c2-aix
```

2. On node1 copy the /.ssh/id_rsa.pub file to /.ssh/authorized_keys
cp /.ssh/id_rsa.pub /.ssh/authorized_keys
3. From node1 copy the /.ssh/id_rsa.pub file from node2 to /tmp/id_rsa.pub
scp node2:/.ssh/id_rsa.pub /tmp/id_rsa.pub
4. Add the public key from node2 to the authorized_keys file on node1
cat /tmp/id_rsa.pub >> /.ssh/authorized_keys
5. Copy the authorized key file from node1 to node2
scp /.ssh/authorized_keys node2:/.ssh/authorized_keys

6. To test your ssh configuration ssh as root from node 1 to node1 and node1 to node2 until you are no longer prompted for a password or for addition to the known_hosts file.


```
node1# ssh node1 date
node1# ssh node2 date
node2# ssh node1 date
node2# ssh node2 date
```
7. Suppress ssh banners by creating a .hushlogin file in the root home directory


```
touch /.hushlogin
```
4. Verify the disks are available to the system

For this lab you should have 4 disks available for use *hdiskn-hdiskt*.

 1. Use *lspv* to verify the disks exist
 2. Ensure you see 4 disks besides *hdisk0* talk.

Step 2: Install the GPFS software

On node1

1. Locate the GPFS software in */yourdir/software/base/*

```
cd /yourdir/software/base/
```
2. Run the *inutoc* command to create the table of contents


```
inutoc .
```
3. Install the base GPFS code using the *installp* command


```
installp -aXY -d/yourDir/software/basegpfs -f all
```
4. Locate the latest GPFS patch level in */yourdir/software/PTF/*

```
cd /yourdir/software/PTF/
```
5. Run the *inutoc* command to create the table of contents


```
inutoc .
```
6. Install the PTF GPFS code using the *installp* command


```
installp -aXY -d/yourdir/software/PTFgpfs -f all
```
7. Repeat Steps 1-7 on node28. On node1 and node2 confirm GPFS is installed using *lspp*

```
lspp -L gpfs.*
```

the output should look similar to this

```
# lspp -L gpfs.*
Fileset                Level State Type Description (Uninstaller)
-----
gpfs.base               3.3.0.3  A   F   GPFS File Manager
gpfs.docs.data          3.3.0.3  A   F   GPFS Server Manpages and Documentation
gpfs.gui                3.3.0.3  C   F   GPFS GUI
gpfs.msg.en_US          3.3.0.1  A   F   GPFS Server Messages U.S. English
```

Note: Exact versions of GPFS may vary from this example, the important part is that all three packages are present.
8. Confirm the GPFS binaries are in your path using the *mmfslcluster* command


```
# mmfslcluster
mmfslcluster: 6027-1382 This node does not belong to a GPFS cluster.
mmfslcluster: 6027-1639 Command failed. Examine previous error messages to determine cause.
```

Note: The path to the GPFS binaries is: */usr/lpp/mmfs/bin*

Step 3: Create the GPFS cluster

For this exercise the cluster is initially created with a single node. When creating the cluster make node1 the

primary configuration server and give node1 the designations quorum and manager. Use ssh and scp as the remote shell and remote file copy commands.

*Primary Configuration server (node1): _____

*Verify fully qualified path to ssh and scp: ssh path _____

scp path _____

1. Use the `mmcrcluster` command to create the cluster
`mmcrcluster -N _node01_:manager-quorum -p _node01_ -r /usr/bin/ssh -R /usr/bin/scp`
2. Run the `mmlscluster` command again to see that the cluster was created
`# mmlscluster`

GPFS cluster information

=====

```
GPFS cluster name:   node1.ibm.com
GPFS cluster id:    13882390374179224464
GPFS UID domain:    node1.ibm.com
Remote shell command: /usr/bin/ssh
Remote file copy command: /usr/bin/scp
```

GPFS cluster configuration servers:

```
Primary server:  node1.ibm.com
Secondary server: (none)
```

Node Daemon	node name	IP address	Admin node name	Designation
1	perf3-c2-aix.bvnssg.net	10.0.0.1	node1.ibm.com	quorum-manager

3. Set the license mode for the node using the `mmchlicense` command. Use a server license for this node.
`mmchlicense server --accept -N node01`

Step 4: Start GPFS and verify the status of all nodes

1. Start GPFS on all the nodes in the GPFS cluster using the `mmstartup` command
`mmstartup -a`
2. Check the status of the cluster using the `mmgetstate` command
`# mmgetstate -a`

Node number	Node name	GPFS state
1	node1	active

Step 5: Add the second node to the cluster

1. One node 1 use the `mmaddnode` command to add node2 to the cluster
`# mmaddnode -N node2`
2. Confirm the node was added to the cluster using the `mmlscluster` command
`# mmlscluster`
3. Use the `mmchcluster` command to set node2 as the secondary configuration server
`# mmchcluster -s node2`
4. Set the license mode for the node using the `mmchlicense` command. Use a server license for this node.
`mmchlicense server --accept -N node02`

5. Start node2 using the mmstartup command
mmstartup -N node2
6. Use the mmgetstate command to verify that both nodes are in the active state
mmgetstate -a

Step 6: Collect information about the cluster

Now we will take a moment to check a few things about the cluster. Examine the cluster configuration using the `mmcluster` command

1. What is the cluster name? _____
2. What is the IP address of *node2*? _____

What date was this version of GPFS "Built"? _____

Hint: look in the GPFS log file: `/var/adm/ras/mmfs.log.latest`

Step 7: Create NSDs

You will use the 4 *hdisks*.

- Make sure they can all hold data and metadata
 - Leave the storage pool column blank.
 - Leave the Primary and Backup server fields blank
 - Sample input files are in `/yourdir/samples`
1. On node 1 create directory `/yourdir/data`
 2. Create a disk descriptor file `/yourdir/data/diskdesc.txt` using the format:
#DiskName:PrimaryServer:BackupServer:DiskUsage:FailureGroup:DesiredName:StoragePool
hdiskw:::dataAndMetadata::nsd1:
hdiskx:::dataAndMetadata::nsd2:
hdisky:::dataAndMetadata::nsd3:
hdiskz:::dataAndMetadata::nsd4:
Note: hdisk numbers will vary per system.
 3. Create a backup copy of the disk descriptor file `/yourdir/data/diskdesc_bak.txt`
`cp /yourdir/data/diskdesc.txt /yourdir/data/diskdesc_bak.txt`
 4. Create the NSD's using the `mmcrnsd` command
`mmcrnsd -F /yourdir/data/diskdesc.txt`

Step 8: Collect information about the NSD's

Now collect some information about the NSD's you have created.

1. Examine the NSD configuration using the `mmlsnsd` command
 1. What `mmlsnsd` flag do you use to see the operating system device (`/dev/hdisk?`) associated with an NSD? _____

Step 9: Create a file system

Now that there is a GPFS cluster and some NSD's available you can create a file system. In this section we will create a file system.

- Set the file system blocksize to 64kb

- Mount the file system at /gpfs
- 1. Create the file system using the `mmcrfs` command
`mmcrfs /gpfs fs1 -F diskdesc.txt -B 64k`
- 2. Verify the file system was created correctly using the `mmlsfs` command
`mmlsfs fs1`
 Is the file system automatically mounted when GPFS starts? _____
- 3. Mount the file system using the `_mmmount_` command
`mmmount all -a`
- 4. Verify the file system is mounted using the `df` command

```
# df -k
Filesystem    1024-blocks    Free %Used    Iused %Iused Mounted on
/dev/hd4       65536         6508  91%      3375   64% /
/dev/hd2      1769472      465416  74%     35508   24% /usr
/dev/hd9var    131072        75660  43%       620    4% /var
/dev/hd3       196608      192864   2%        37    1% /tmp
/dev/hd1        65536        65144   1%         13    1% /home
/proc          -             -      -         -      - /proc
/dev/hd10opt   327680      47572  86%       7766   41% /opt
/dev/fs1      398929107 398929000  1%          1    1% /gpfs
```
- 5. Use the `mmddf` command to get information on the file system.
`mmddf fs1`
 How many inodes are currently used in the file system? _____

Exercise #2: Storage Pools, Filesets and Policies

Objectives:

- Create storage pools
- Create a file system
- Create filesets
- Implement a Placement Policy
- Define and execute a file management policy

Requirements:

Complete Exercise 1: Installing the cluster

List of available devices

```
/dev/sd__  
/dev/sd__  
/dev/sd__  
/dev/sd__  
/dev/sd__
```

Step 1: Create A File System With 2 Storage Pools

Storage pools are defined when an NSD (Network Shared Disk) is created. You will use the 4 *disks* (labelled sd_ through sd_) provided by your instructor to create 2 storage pools. Since the storage is all direct attached, no NSD servers are needed. Place the disks into two storage pools and make sure both pools can store file data.

hints:

- Create 2 disks per storage pool
- The first pool must be the *system* pool.
- The system pool should be able to store data and metadata
- Only the system pool can contain metadata # Create a disk descriptor file */gpfs-course/data/pooldesc.txt* using the format: #DiskName:serverlist::DiskUsage:FailureGroup:DesiredName:StoragePool

```
/dev/sd__:::dataAndMetadata::nsd1:system  
/dev/sd__:::dataAndMetadata::nsd2:system  
/dev/sd__:::dataOnly::nsd3:pool1  
/dev/sd__:::dataOnly::nsd4:pool1
```

The two storage pools will be system and pool1.

1. Create a backup copy of the disk descriptor file
/gpfs-course/data/pooldesc_bak.txt
2. Create the NSD's using the *mmcrnsd* command.

```
> mmcrnsd -F /gpfs-course/data/pooldesc.txt -v no  
mmcrnsd: Processing disk sdc  
mmcrnsd: Processing disk sdd  
mmcrnsd: Processing disk sde  
mmcrnsd: Processing disk sdf  
mmcrnsd: Propagating the cluster configuration data to all  
affected nodes. This is an asynchronous process.
```
3. Create a file system based on these NSD's using the *mmcrfs* command
 - * Set the file system blocksize to 64KB
 - * Mount the file system at */gpfs*

Command: "mmcrfs /gpfs fs1 -F */gpfs-course/data/pooldesc.txt* -B 64k -M 2 -R 2"

Example:

```
gpfs1:~ # mmcrfs /gpfs fs1 -F /gpfs-course/data/pooldesc.txt -B 64k -M2 -R2
```

The following disks of fs1 will be formatted on node gpfs1:

```

nsd1: size 20971520 KB
nsd2: size 20971520 KB
nsd3: size 20971520 KB
nsd4: size 20971520 KB
Formatting file system ...
Disks up to size 53 GB can be added to storage pool 'system'.
Disks up to size 53 GB can be added to storage pool 'pool1'.
Creating Inode File
 45 % complete on Wed Sep 26 10:05:27 2007
 89 % complete on Wed Sep 26 10:05:32 2007
100 % complete on Wed Sep 26 10:05:33 2007
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
 42 % complete on Wed Sep 26 10:05:52 2007
 83 % complete on Wed Sep 26 10:05:57 2007
100 % complete on Wed Sep 26 10:05:59 2007
 43 % complete on Wed Sep 26 10:06:04 2007
 85 % complete on Wed Sep 26 10:06:09 2007
100 % complete on Wed Sep 26 10:06:10 2007
Completed creation of file system /dev/fs1.
mmcrfs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

```

4. Verify the file system was created correctly using the *mmfsfs* command
 > mmfsfs fs1

5. Mount the file system using the *mmmount* command
 > mmmount fs1 -a

6. Verify the file system is mounted using the *df* command.
 > df

```

Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda3        32908108 10975916 21932192  34% /
tmpfs            2019884      4 2019880   1% /dev/shm
/dev/sda1         72248    45048  27200  63% /boot
/dev/fs1        3989291072 491992640 3497298432 13% /gpfs

```

7. Verify the storage pool configuration using the *mmddf* command
 > mmdf fs1

disk name	disk size in KB	failure group	holds metadata	holds data	free KB in full blocks	free KB in fragments
Disks in storage pool: system						
nsd1	102734400	-1 yes	yes		102565184 (100%)	90 (0%)
nsd2	102734400	-1 yes	yes		102564608 (100%)	96 (0%)
(pool total)	205468800				205129792 (100%)	186 (0%)
Disks in storage pool: pool1						
nsd3	102734400	-1 no	yes		102732288 (100%)	62 (0%)
nsd4	102734400	-1 no	yes		102732288 (100%)	62 (0%)
(pool total)	205468800				205464576 (100%)	124 (0%)
=====						
(data)	410937600				410594368 (100%)	310 (0%)
(metadata)	205468800				205129792 (100%)	186 (0%)
=====						
(total)	410937600				410594368 (100%)	310 (0%)

Inode Information

```

Number of used inodes:      4038

```

Number of free inodes: 397370
 Number of allocated inodes: 401408
 Maximum number of inodes: 401408

Step 2: Create Filesets

We are going to 5 filesets to organize the data.

1. Create 5 filesets fileset1-fileset5 using the *mmcrfileset* command

```
> mmcrfileset fs1 fileset1
> mmcrfileset fs1 fileset2
> mmcrfileset fs1 fileset3
> mmcrfileset fs1 fileset4
> mmcrfileset fs1 fileset5
```

2. Verify they were created using the *mmlsfileset* command

```
# mmlsfileset fs1
```

What is the status of fileset1-fileset5? _____

3. Link the filesets into the file system using the *mmlinkfileset* command

```
# mmlinkfileset fs1 fileset1 -J /gpfs/fileset1
# mmlinkfileset fs1 fileset2 -J /gpfs/fileset2
# mmlinkfileset fs1 fileset3 -J /gpfs/fileset3
# mmlinkfileset fs1 fileset4 -J /gpfs/fileset4
# mmlinkfileset fs1 fileset5 -J /gpfs/fileset5
```

Now what is the status of fileset1-fileset5? _____

Step 3: Create a file placement policy

Now that you have two storage pools and some filesets you need to define placement policies to instruct GPFS where you would like the file data placed. By default, if the system pool can accept data, all files will go to the system storage pool. You are going to change the default and create 3 placement rules. The rules will designate:

- Data in fileset1 to fileset4 go to the system storage pool
 - Data in fileset5 go to the pool1 storage pool
 - files that end in .dat go to pool1 storage pool
1. Start by creating a policy file */gpfs-course/data/placementpolicy.txt*
 /* The fileset does not matter, we want all .dat and .DAT files to go to pool1 */
 RULE 'datfiles' SET POOL 'pool1' WHERE UPPER(name) like '%.DAT'

 /* All non *.dat files placed in fileset5 will go to pool1 */
 RULE 'fs5' SET POOL 'pool1' FOR FILESET ('fileset5')

 /* Set a default rule that sends all files not meeting the other criteria to the system pool */
 RULE 'default' set POOL 'system'
 2. Install the new policy file using the *mmchpolicy* command
 > mmchpolicy fs1 placementpolicy.txt

Step 4: Testing the placement policies

Now you will do some experiments to see how policies work. Use this chart to track the experiments results. You can get the amount of free space by using the *mmdf* command.

Experiment	System Pool (Free KB)	Pool1 (Free KB)
Before		
Bigilfe1		
bigfile1.dat		
bigfile2		
Migrate/Delete		

- Record the "Before" free space in the chart
- Create a file in fileset1 called bigfile1
dd if=/dev/zero of=/gpfs/fileset1/bigfile1 bs=64k count=10000
- Record the free space in each pool using the *mmdf* command (Bigfile1)

```
> mmdf fs1
disk      disk size  failure holds   holds      free KB      free KB
name      in KB    group metadata data    in full blocks  in fragments
-----
Disks in storage pool: system
nsd1      20971520   -1 yes    yes    20588288 ( 98%)    930 ( 0%)
nsd2      20971520   -1 yes    yes    20588608 ( 98%)    806 ( 0%)
-----
(pool total)  41943040                      41176896 ( 98%)    1736 ( 0%)

Disks in storage pool: pool1
nsd3      20971520   -1 no     yes    20969408 (100%)    62 ( 0%)
nsd4      20971520   -1 no     yes    20969408 (100%)    62 ( 0%)
-----
(pool total)  41943040                      41938816 (100%)    124 ( 0%)

=====
(data)      83886080                      83115712 ( 99%)    1860 ( 0%)
(metadata)  41943040                      41176896 ( 98%)    1736 ( 0%)
=====
(total)      83886080                      83115712 ( 99%)    1860 ( 0%)

Inode Information
-----
Number of used inodes:    4044
Number of free inodes:   78132
Number of allocated inodes: 82176
Maximum number of inodes: 82176
```

- Create a file in fileset1 called bigfile1.dat
dd if=/dev/zero of=/gpfs/fileset1/bigfile1.dat bs=64k count=1000
Record the free space (bigfile1.dat)
- Create a file in fileset5 called bigfile2
dd if=/dev/zero of=/gpfs/fileset5/bigfile2 bs=64k count=1000
Record the free space (bigfile2)
- Questions
 - Where did the data go for each file?
 - Bigfile1 _____
 - Bigfile1.dat _____
 - Bigfile2 _____
 - Why?

7. Create a couple more files (These will be used in the next step)
 - > dd if=/dev/zero of=/gpfs/fileset3/bigfile3 bs=64k count=10000
 - > dd if=/dev/zero of=/gpfs/fileset4/bigfile4 bs=64k count=10000

Step 5: File Management Policies

Now that you have data in the file system you are going to manage the placement of the file data using file management policies. For this example your business rules say that all file names that start with the letters "big" need to be moved to pool1. In addition, all files that end in ".dat" should be deleted.

1. To begin, create a policy file /gpfs-course/data/managementpolicy.txt that implements the business rules.


```
RULE 'datfiles' DELETE WHERE UPPER(name) like '%.DAT'
RULE 'bigfiles' MIGRATE TO POOL 'pool1' WHERE UPPER(name) like 'BIG%'
```
2. Test the rule set using the *mmapplypolicy* command


```
> mmapplypolicy fs1 \-P managementpolicy.txt \-I test
```

This command will show you what *mmapplypolicy* will do but will not actually perform the delete or migrate.
3. Actually perform the migration and deletion using the *mmapplypolicy* command


```
> mmapplypolicy fs1 \-P managementpolicy.txt
```
4. Review:

Review the output of the *mmapplypolicy* command to answer these questions.

How many files were deleted? _____

How many files were moved? _____

How many KB total were moved? _____

Step 6: Using External Pools

In this step you will use the external pool interface to generate a report.

1. Create two files : expool1.bash, listrule1.txt:


```
File /tmp/expool1.bash
#!/bin/bash

dt=`date +%h%d%y-%H_%M_%S`
results=/tmp/FileReport_`${dt}`

echo one $1
if [[ $1 == 'MIGRATE' ]];then
  echo Filelist
  echo There are `cat $2 | wc -l` files that match >> ${results}
  cat $2 >> ${results}
  echo ----
  echo - The file list report has been placed in ${results}
  echo ----
fi
File listrule1.txt
RULE EXTERNAL POOL 'externalpoolA' EXEC '/tmp/expool1.bash'

RULE 'MigToExt' MIGRATE TO POOL 'externalpoolA'
WHERE FILE_SIZE > 2
```

Note: You may need to modify where clause to get a list of files on your file system.
2. Make the external pool script executable


```
chmod +x /tmp/expool1.bash
```

3. Execute the job

`mmapplypolicy test1 -P listrule1.txt`

It will print output to the screen. When it is done it will print the location of the results file. For example:

The file list report has been placed in `/tmp/FileReport_Jul3108-20_15_50`

4. What information do you see in the file?

Exercise #3: Using Replication

In this lab we will configure a file system for replication.

Objectives:

- Enable data and metadata replication
- Verify and monitor a file's replication status

Requirements:

1. Complete Exercise 1: Installing the cluster
2. List of available devices
/dev/sd__
/dev/sd__
/dev/sd__
/dev/sd__

Step 1: Enabling Replication

1. The max replication factor should have been set in Lab 1. Use the *mmfsfs* command to verify the file system is enabled for replication. A file system is enabled for replication when the maximum number of data and metadata replicas is set to 2.
2. If these parameters are not set to 2 you will need to recreate the file system. To recreate the file system
 - a. Umount the file system
 - b. Delete the file system
 - c. Create the file system and specify -M 2 and -R 2
> *mmcrfs* /gpfs fs1 -F pooldesc.txt -B 64k -M 2 -R 2

Where pooldesc.txt is the disk descriptor file from Lab 1

Step 2: Change the failure group on the NSDs

1. View the current failure group settings using the *mmddf* command
> *mmfsdisk* fs1
The failure group should be set to a value of -1
2. Change the failure group to 1 for nsd1 and nsd3 and to 2 for nsd2 and nsd4 using the *mmchdisk* command.
> *mmchdisk* fs1 change -d "nsd1:::dataAndMetadata:1:::"
> *mmchdisk* fs1 change -d "nsd2:::dataAndMetadata:2:::"
> *mmchdisk* fs1 change -d "nsd3:::dataOnly:1:::"
> *mmchdisk* fs1 change -d "nsd4:::dataOnly:2:::"
3. Verify the changes using the *mmddf* command
mmfsdisk fs1

Notice that data was not written because the default replication level is still set to 1. Now that there are two failure groups you can see how to change the replication status of a file.

Step 3: Replicate a file

Replication status can be set at the file level. In this step we will replicate the data and metadata of a single file in the file system.

1. Create a file in the GPFS file system, /gpfs, called bigfile10
dd if=/dev/zero of=/gpfs/fileset1/bigfile10 bs=64k count=1000

2. Use the *mm/sattr* command to check the replication status of the file bigfile10


```
> mmlsattr /gpfs/fileset1/bigfile10
replication factors
metadata(max) data(max) file [flags]
-----
1 ( 2) 1 ( 2) /gpfs/fileset1/bigfile10
```
3. Change the file replication status of bigfile10 so that it is replicated in two failure groups using the *mmchattr* command.


```
mmchattr \-m 2 \-r 2 /gpfs/fileset1/bigfile10
```

 Notice that this command takes a few moments to execute, as you change the replication status of a file the data is copied before the command completes unless you use the "-I defer" option.
4. Again use the *mm/sattr* command to check the replication status of the file bigfile10


```
> mmlsattr /gpfs/fileset1/bigfile10
```

 Did you see a change in the replication status of the file?

Step 4: Replicate all data in the file system

If desired you can replicate all of the data in the file system. In this step we will change the default replication status for the whole file system.

1. Create a file in fileset1 called bigfile11


```
dd if=/dev/zero of=/gpfs/fileset1/bigfile11 bs=64k count=1000
```
2. Use the *mm/sattr* command to check the replication status of the file bigfile11


```
mmlsattr /gpfs/fileset1/bigfile11
```
3. Using the *mmchfs* command change the default replication status for fs1.


```
mmchfs fs1 \-m 2 \-r 2
```
4. Use the *mm/sattr* command to check the replication status of the file bigfile11


```
mmlsattr /gpfs/fileset1/bigfile11
```

 Has the replication status of bigfile11 changed? _____
5. The replication status of a file does not change until *mmrestripefs* is run or a new file is created. To test this create a new file called bigfile12


```
dd if=/dev/zero of=/gpfs/fileset1/bigfile12 bs=64k count=1000
```
6. Use the *mm/sattr* command to check the replication status of the file bigfile12


```
mmlsattr /gpfs/fileset1/bigfile12
```

 Is the file replicated?
7. You can replicate the existing files in the file system using the *mmrestripefs* command


```
mmrestripefs fs1 \-R
```

Exercise #4: Snapshots

In this lab we will use the snapshot feature to create online copies of files.

Objectives:

- Create a file system snapshot
- Restore a user deleted file from a snapshot image
- Manage multiple snapshot images

Requirements:

1. Complete Exercise 1: Installing the cluster
2. A File System - Use Exercise 2 to create a file system if you do not already have one.

Step 1: Use a snapshot to backup a file

A snapshot is a point in time view of a file system. To see how snapshots operate you will create a file, take a snapshot, delete the file then restore the snapshot from the file.

1. Create a file for testing in the `/gpfs/fileset1` directory.
`echo "hello world:snap1" > /gpfs/fileset1/snapfile1`
2. Create a snapshot image using the *mmcrsnapshot* command
`> mmcrsnapshot fs1 snap1`

Writing dirty data to disk
Quiescing all file system operations
Writing dirty data to disk again
Creating snapshot.
Resuming operations.
3. Modify the file for testing in the `/gpfs/fileset1` directory.
`echo "hello world:snap2" >> /gpfs/fileset1/snapfile1`
4. Create a seconds snapshot image using the *mmcrsnapshot* command
`mmcrsnapshot fs1 snap2`
5. View the list of snapshots created using the *mmlssnapshot* command
Example:
`> mmlssnapshot fs1`
`gpfs1:~ # mmlssnapshot fs1`
Snapshots in file system fs1:

Directory	SnapId	Status	Created
snap1	2	Valid	Wed Sep 26 11:03:52 2007
snap2	3	Valid	Wed Sep 26 11:04:52 2007
6. Delete the file `/gpfs/fileset1/snapfile1` Now that the file is deleted let's see what is in the snapshots:
7. Take a look at the snapshot images. To view the image change directories to the `.snapshot` directory
`cd /gpfs/.snapshots`
What directories do you see? _____
8. Compare the `snapfile1` stored in each snapshot
`cat snap1/fileset1/snapfile1`
`cat snap2/fileset1/snapfile1`
Are the file contents the same? _____
9. To restore the file from the snapshot copy the file back into the original location
`cp /gpfs/.snapshots/snap2/fileset1/snapfile1 /gpfs/fileset1/snapfile1`
10. When you are done with a snapshot you can delete the snapshot. Delete both of these snapshots

using the *mmde/snapshot* command

```
> mmdelsnapshot fs1 snap1  
> mmdelsnapshot fs1 snap2
```

11. Verify the snapshots were deleted using the *mm/lssnapshot* command
mm/lssnapshot fs1

Exercise #5: Dynamically Adding a Disk to an Online File System

Objectives:

- Add a disk to a storage pool online
- Re-balance existing data in the file system

Requirements:

1. Complete Exercise 1: Installing the cluster
2. A File System (Use Exercise 2 to create a file system if you do not already have one).
3. Device to add
/dev/sd____

Step 1: Add a disk to the existing file system

1. Verify that GPFS is running and the file system is mounted using the *mmgetstate* command and the *df* command
 - a. The *mmgetstate* command will show the status of the nodes in the cluster.
> mmgetstate -a
 - b. The *df* command will display the mounted GPFS file system.
df
2. Create a disk descriptor file /gpfs-course/data/adddisk.txt for the new disk using the format
#DiskName:serverlist::DiskUsage:FailureGroup:DesiredName:StoragePool
/dev/sd_:::dataOnly::nsd5:pool1
3. Use the *mmcrnsd* command to create the NSD
> mmcrnsd -F /gpfs-course/data/adddisk.txt
4. Verify the disk has been created using the *mmfnsd* command
> mmfnsd
The disk you just added should show as a (free disk)
5. Add the new NSD to the fs1 file system using the *mmadddisk* command
> mmadddisk fs1 -F /gpfs-course/data/adddisk.txt
6. Verify the NSD was added using the *mmddf* command
> mmddf fs1

disk name	disk size in KB	failure group	holds metadata	holds data	free KB in full blocks	free KB in fragments

Disks in storage pool: system						
nsd1	20971520	1 yes	yes		20873984 (100%)	284 (0%)
nsd2	20971520	2 yes	yes		20873984 (100%)	202 (0%)

(pool total)	41943040				41747968 (100%)	486 (0%)
Disks in storage pool: pool1						
nsd3	20971520	1 no	yes		20969408 (100%)	62 (0%)
nsd4	20971520	2 no	yes		20969408 (100%)	62 (0%)
nsd5	20971520	-1 no	yes		20969408 (100%)	62 (0%)

(pool total)	62914560				62908224 (100%)	186 (0%)

=====

=====

(data)	104857600	104656192 (100%)	672 (0%)
(metadata)	41943040	41747968 (100%)	486 (0%)
=====			
(total)	104857600	104656192 (100%)	672 (0%)

Inode Information

```

-----
Number of used inodes:    4045
Number of free inodes:   78131
Number of allocated inodes: 82176
Maximum number of inodes: 82176

```

Step 2: Re-balancing the data

In some cases you may wish to have GPFS re-balance existing data over the new disks that were added to the file system. Often it is not necessary to manually re-balance the data across the new disks. New data that is added to the file system is correctly striped. Re-striping a large file system requires a large number of insert and delete operations and may affect system performance. Plan to perform this task when system demand is low.

1. To re-balance the existing data in the file system use the *mmrestripefs* command.
 > mmrestripefs fs1 -b

Use the *mmdf* command to view the utilization of each disk.