Configuring IBM General Parallel File System (GPFS) with Oracle RAC

On IBM pSeries with AIX 5L and Linux on POWER

Rick Piasecki
IBM eServer Solutions Enablement
January 2006

Table of contents

Abstract

This paper includes a review of key IBM General Parallel File System (GPFS) features for Oracle Real Application Clusters (RAC) Databases. This document also includes a summary of IBM High Availability Cluster Multi-Processing (HACMP) requirements and options with Oracle RAC or GPFS and the requirements for tuning GPFS and IBM AIX 5L for Oracle. There are two sample GPFS installation and configuration scenarios, as well as an example migration to GPFS Version 2.3 from a previous GPFS version.

Introduction

IBM® GPFS V2.3 has been verified for use with:

- Oracle® 9*i* Real Application Clusters (RAC) and Oracle Database 10*g* RAC (10.1.0.x and 10.2.0.1) on both the AIX 5L™ Version 5.3 and V5.2 operating systems
- Oracle Database 10g Release 2 (10.2.0.1) RAC on Linux™ on POWER™
  - Novell® SUSE® Linux Enterprise Server (SLES) 9 for IBM POWER with Service Pack 2 (SP2)
  - Red Hat® Enterprise Linux® (RHEL) 4 for POWER with Update 1

Note: Table 1, shown on the next page, provides software support details.

GPFS is the IBM high-performance parallel, scalable file system for UNIX® clusters and is capable of supporting multiple terabytes of storage within a single file system.

GPFS is a shared-disk file system where every cluster node can have parallel, concurrent read and write access to the same file. It is designed to provide high-performance I/O by striping data across multiple disks that are accessed from multiple servers. GPFS provides high availability through logging and replication. It can be configured for automatic failover from both disk and node malfunctions.

GPFS can be used for all components of an Oracle Database 10*g* RAC configuration, including the following:

- The shared Cluster Ready Services (CRS) home
- Oracle Home
- Oracle Cluster Registry (OCR) disk
- Voting disk
- The Oracle data and log files

GPFS can also be used to complement the Oracle Automatic Storage Management (ASM) feature in Oracle Database 10*g*; managing the shared CRS Home, Oracle Home, OCR disk, and voting disk while ASM manages the Oracle data and log files.

GPFS V2.1 and V2.2 were previously approved for Oracle RAC but GPFS V2.3 now offers several new key features, including the following:

- Support for the AIX® 5L V5.3 operating system
- Single-node quorum with tie-breaker disks
- Single GPFS cluster type
- More disaster recovery options

A summary of key GPFS features for Oracle RAC Databases is given later in this paper. Database administrators (DBAs) who are planning to use GPFS with any Oracle RAC configuration must select GPFS V2.3. DBAs who are dealing with existing GPFS and Oracle RAC installations need to consider upgrading to GPFS V2.3.

This document also includes:

- A Summary of HACMP™ requirements and options with Oracle RAC or GPFS
- GPFS and AIX tuning requirements for Oracle
- Sample GPFS installation and configuration scenarios
- Example migration to GPFS V2.3 from a previous GPFS version
- A list of GPFS references and additional information

GPFS and Oracle RAC: Supported configurations

The Oracle RAC server for Oracle Database 9*i* and Oracle Database 10*g* supports the software configurations shown in Table 1.

	Oracle RAC server		
	Oracle 9*i* 9.2.0.2 for RAC or higher	**Oracle Database 10*g* (10.1.0.x) for RAC**	**Oracle Database 10*g* (10.2.0.x) for RAC**
	HACMP is required for Oracle 9*i* RAC (See HACMP information below)	HACMP is not required for Oracle Database 10*g* RAC (See HACMP information below)	HACMP is not required for Oracle Database 10*g* RAC (See HACMP information below)
AIX 5L V5.2 ML 04, or later	GPFS V2.1, V2.2, V2.3.0.1 or higher	GPFS V2.1, V2.2, V2.3.0.1 or higher	GPFS V2.3.0.3 or higher
AIX 5L V5.3, or later	GPFS V2.3.0.1 or higher	GPFS V2.3.0.1 or higher	GPFS V2.3.0.3 or higher
Linux on POWER SLES 9 for IBM POWER with SP2, or RHEL 4 for POWER with Update 1	Not supported	Not supported	GPFS V2.3.0.6 or higher

Table 1: Oracle RAC server software configurations

Be aware of the following notes regarding the supported software configurations:

- The AIX 5L 64- bit kernel is advised for Oracle RAC and GPFS configurations.
- GPFS V2.3 requires the V2.3.0.1 upgrade (APAR IY63969) or later.
- GPFS V2.2 requires GPFS PTF 6 (V2.2.1.1) or later.
- GPFS V2.1 was withdrawn from marketing on April 29, 2005.
- GPFS for AIX 5L, V2.2 was withdrawn from marketing on June 30, 2005.
- See Oracle MetaLink note 282036.1 for the latest software requirements for AIX 5L and Oracle.
- See Oracle MetaLink note 341507.1 for the latest software requirements for Linux on POWER and Oracle.

HACMP considerations with Oracle RAC, AIX 5L, and GPFS

There are particular requirements and options for using the HACMP product with Oracle RAC, the AIX 5L operating system, and GPFS. Some of these considerations are as follows:

- Oracle 9*i* RAC always requires HACMP.
- HACMP is optional for Oracle Database 10*g* RAC.
- HACMP V5.1 and V5.2 are certified with both Oracle 9*i* and Oracle Database 10*g* on both the AIX 5L V5.2 and V5.3 operating systems. (**Note:** See Oracle MetaLink note 282036.1 for the latest complete set of patch requirements for HACMP, the AIX 5L operating system, and Oracle.)

In Oracle 9*i* RAC, there are additional considerations:

- HACMP is required as the Oracle 9*i* RAC clusterware.
- HACMP is required if using shared concurrent volume groups (raw logical volumes managed by HACMP).
- HACMP is optional for the GPFS V2.2 node set. Instead, it is possible to use IBM Reliable Scalable Cluster Technology (RSCT) Peer Domain (also known as RPD).
- HACMP and RPD are not required for GPFS V2.3.

In Oracle Database 10*g* RAC, there are also further considerations:

- HACMP is optional for Oracle Database 10*g* RAC CRS:
  - If HACMP is configured for Oracle, CRS will use the HACMP node names and numbers.
  - If HACMP is configured to provide high availability for other products, this is compatible with CRS for Oracle Database 10*g* RAC.
- HACMP is only required if using shared concurrent volume groups (raw logical volumes managed by HACMP).
- HACMP is optional for the GPFS V2.2 node set. Instead, it is possible to use RPD.
- HACMP and RPD are not needed for GPFS V2.3.

Therefore, it is possible to have a complete Oracle 10*g* RAC configuration with Oracle Database 10*g* RAC and GPFS V2.3.

Note: The previous information is for the AIX 5L operating system only. HACMP is not supported on Linux on POWER.

GPFS features for Oracle RAC

This section explains the key GPFS features for Oracle RAC databases on AIX 5L and Linux on POWER.

- In GPFS V2.3, new, single-node quorum support provides two-node Oracle high availability for all disk subsystems.
  - A new quorum type of **node quorum with tiebreaker disks** can be used with one or three tie-breaker disks.
  - There is no dependence on a particular storage architecture. (GPFS is designed to work with all storage architectures.)
    (**Note:** See the GPFS V2.3 FAQs [especially the first question within the **Disk-specific questions** section] for the currently verified IBM storage and a storage support statement.)
- In GPFS V2.3, a single **cluster type** removes the requirement for additional cluster software.
  - There are no more **hacmp**, **rpd**, or **lc** cluster types.
    (**Note:** The **lc**, or **loose cluster**, type is now implicit.)
  - There is no requirement for HACMP or RSCT for GPFS V2.3
    (**Note:** HACMP is required for Oracle 9*i* RAC configurations.)
  - The GPFS concept of **nodesets** is removed, which simplifies administration.
  - New GPFS **NSD** disk types support storage area network (SAN) and network-attached storage (NAS) configurations (or a combination of the two).
    (**Note:** NSD stands for Network Shared Disk.)
  - Migration from previous GPFS V2.2 cluster types is fully supported and documented.
- There are dynamic support capabilities for Oracle Database.
  - Disks can be dynamically added or removed from a GPFS file system.
  - Automatic rebalancing of the GPFS file system occurs after disks are added or removed.
  - Nodes can be dynamically added or removed from a GPFS cluster.
- It is possible to achieve optimal performance with Oracle Database by using best practices.
  - The use of direct I/O with asynchronous I/O is the Oracle default for Oracle data and log files.
  - GPFS provides a choice of large block sizes.
  - The new **mmpmon** command monitors GPFS performance details.
    Note: GPFS best practices tuning for Oracle is documented later in this document.
- High availability and backup support is provided for Oracle Databases
  - GPFS supports hardware Redundant Array of Independent Disks (RAID) configuration.
  - GPFS provides its own 2-way or 3-way replication of data, metadata, or both.
  - It is possible to exploit AIX 5L EtherChannel and IEEE 802.3ad link aggregation or Channel Bonding on Linux for the GPFS network.
  - The new high availability and backup support is compatible with standard file system backup and restore programs.

- There are multiple disaster recovery options for Oracle RAC Database when using GPFS.
  - Synchronous mirroring can utilize GPFS replication.
  - Synchronous mirroring can utilize IBM TotalStorage® Enterprise Storage Server® (ESS) Peer-to-Peer Remote Copy (PPRC).
  - Asynchronous mirroring can utilize IBM TotalStorage ESS FlashCopy®.
  - The GPFS **mmfsctl** command is used for disaster recovery management.

GPFS tuning requirements for Oracle

Both the AIX environment and the Oracle Database are well known for the robustness of tuning mechanisms that are available to DBAs for the purposes of sustaining maximum performance from their combined hardware, operating system, database, and application software assets. This section of the paper will highlight some of these tuning options.

AIO and DIO options

By default, Oracle uses the asynchronous I/O (AIO) and direct I/O (DIO) features of the AIX 5L operating system to do its own scheduling of I/O directly to disks, bypassing most of the GPFS caching and prefetching facilities. Therefore:

- Do not use the **dio** mount option for the GPFS file system or change the DIO attribute for any Oracle files.
- The Oracle init.ora parameter **filesystemio_options** setting will be ignored for Oracle files on GPFS.

Configuring LUNs for GPFS and Oracle

If using RAID devices, configure a single logical unit number (LUN) for each RAID device. Do not create LUNs across RAID devices for use by GPFS as this will ultimately result in a significant loss in performance. It will also make the removal of a bad RAID more difficult. GPFS will stripe across the multiple LUNs (RAIDs) using its own optimized method.

GPFS block size, Oracle db_block_size, and db_file_multiblock_read_count

For Oracle RAC Databases, set the GPFS file system block, using the **mmcrfs** command and the **-B** option, to a large value. Adhere to the following guidelines:

- 512 kilobytes is generally suggested.
- 256 kilobytes is suggested when there is significant activity (other than Oracle) using the file system and many small files exist that are not in the database.
- 1 megabyte is suggested for file systems of 100 terabytes or larger.

The large block size makes the allocation of space for the databases manageable and has no effect on performance when Oracle is using AIO and DIO. (**Note:** Do not set the GPFS block size equal to the Oracle **db_block_size**.)

- Set the Oracle **db_block_size** value so that it is equal to the LUN segment size or to a multiple of the LUN pdisk segment size.

- Set the Oracle init.ora parameter **db_file_multiblock_read_count** value to prefetch one or two full GPFS blocks.

For example, if the GPFS block size is 512 kilobytes and the Oracle block size is 16 kilobytes, set the Oracle **db_file_multiblock_read_count** to either 32 or 64.

GPFS and AIX 5L tuning for AIO

This section explains some guidelines for using AIX 5L tuning parameters and tunables to improve the performance of GPFS for asynchronous I/O.

GPFS threads

Use the following guidelines to set the GPFS **worker threads** to allow the maximum parallelism of the Oracle AIO threads, and the GPFS **prefetch threads** to benefit from Oracle sequential I/O.

On a 64-bit AIX kernel:

- GPFS worker threads can be less than or equal to 548.
- GPFS worker threads + GPFS prefetch threads less than or equal to 550.

When requiring GPFS sequential I/O, set the prefetch threads between 50 and 100 (the default is 64), and set the worker threads to have the remainder. For example:

```
mmchconfig prefetchThreads=75
mmchconfig worker1Threads=475
```

On a 32-bit AIX kernel:

- GPFS worker threads can be less than or equal to 162.
- GPFS worker threads + GPFS prefetch threads less than or equal to 164.

Note:

- The 64-bit AIX kernel is preferred for optimal performance with GPFS and Oracle RAC.
- These changes, via the **mmchconfig** command, require that GPFS be restarted. Refer to the **mmshutdown** and **mmstartup** commands.

Corresponding tuning of AIX AIO maxservers

The number of AIX AIO kprocs that are created must be approximately the same as the GPFS ***worker1Threads*** setting. For the AIO maxservers setting:

- On AIX 5L V5.1 systems, it is the total number of AIO kprocs.
- On AIX 5L V5.2 and V5.3 systems, it is the number of kprocs per CPU.
  Note: It is suggested that kprocs be set slightly larger than *worker1Threads* divided by the number of CPUs. For example if *worker1Threads* is set to 500 on a 32-way SMP:
  - On an AIX 5L V5.1 system, set maxservers to 640.
  - On AIX 5L V5.2 and V5.3 systems, **maxservers** is a per CPU parameter. Therefore, 640 AIO kprocs divided by 32 CPUs per system equals 20 for maxservers.

Use the **smit aio** configuration option or the **chdev -l aio0 -a maxservers=<value> -P** command to set the value. System reboot will be required for the changes to take affect.

The free **nmon** performance tool can be used to effectively monitor AIO kproc behavior. The tool can be downloaded from:

ibm.com/developerworks/eserver/articles/analyze_aix/index.html

GPFS and pinned SGA

Oracle databases requiring high performance will usually benefit from running with a pinned Oracle System Global Area (SGA). This is also true when running with GPFS, because GPFS uses DIO, which requires that the user I/O buffers (in the SGA) be pinned. GPFS will normally pin the I/O buffers on behalf of the application; but, if Oracle has already pinned the SGA, GPFS will recognize this and will not duplicate the pinning, which saves additional system resources.

Pinning the SGA on the AIX 5L operating environment requires the following three steps.

1. $ /usr/sbin/vmo -r -o v_pinshm=1
2. $ /usr/sbin/vmo -r -o maxpin%=percent_of_real_memory
3. Where percent_of_real_memory = ((size of SGA / size of physical memory) *100) + 3

Note: Set the LOCK_SGA parameter to TRUE in the init.ora.

Other important GPFS attributes

There are other important GPFS attributes, as is discussed below:

- If the GPFS contains a shared Oracle Home or CRS Home, the default value for the maximum number of inodes will probably be insufficient for the Oracle Universal Installer (OUI) installation process. Use a command, such as the following, to increase the inode value:

```
mmchfs /dev/oragpfs –F 50000
```

Inode consumption can be verified through the standard AIX system command:

```
root@raven:64bit /> df -g /oragpfs
Filesystem    GB blocks      Free %Used    Iused %Iused Mounted on
/dev/oragpfs     130.00    120.90    7%    32692    24% /oragpfs
```

Or, through the GPFS command:

```
root@raven:64bit /> mmdf /dev/oragpfs -F
Inode Information
-----------------
Total number of inodes: 139264
Total number of free inodes: 106572
```

- In order for Oracle RAC node recovery to work correctly, the DBA must configure the GPFS (1) to be loaded automatically at boot time and (2) to be mounted automatically. Use the following two AIX commands to configure this:

```
root@raven:64bit /> mmchconfig autoload=yes
```

```
      root@raven:64bit /> mmchfs /dev/oragpfs -A yes
mmchfs: 6027-1371 Propagating the changes to all affected nodes.
```

This is an asynchronous process.

GPFS V2.3 installation examples

Two GPFS installation examples are provided in this section. In the first example, the tie-breaker disks are part of the GPFS file system. In the second example, GPFS replication is used and the tie-breaker disks are separate from the GPFS file system.

Example 1: Create a GPFS file system where tie-breaker disks are part of the GPFS file system

Follow these steps to create a GPFS without replication and with tie-breaker disks that are a part of the GPFS:

1. Preinstallation steps:
   a. Set **chdev –l aio0 –a maxservers=20**.
   b. Select and configure the GPFS network.
   c. Add GPFS interface names to the **/.rhosts** file.
   d. Ensure that a properly configured .rhosts file exists in the root user's home directory on each node in the GPFS.

2. Install the GPFS software. Note that:
   a. The GPFS file set names have changed in GPFS V2.3.
   b. GPFS V2.3.0.1 update (APAR IY63969) is mandatory.

```
root@raven: /> lslpp -l |grep -i gpfs
  gpfs.base             2.3.0.1  APPLIED          GPFS File Manager
  gpfs.msg.en_US        2.3.0.0  COMMITTED        GPFS Server Messages - U.S.
  gpfs.base             2.3.0.1  APPLIED          GPFS File Manager
  gpfs.docs.data        2.3.0.1  APPLIED          GPFS Server Manpages and
```

   c. Add GPFS to $PATH:

```
export PATH=$PATH:/usr/lpp/mmfs/bin
```

3. Create a 2-node GPFS cluster. Note that:
   a. This cluster consists of two nodes, **raven** and **star**.
   b. Each node has a private interface that is also configured (**ravenp** and **starp**, respectively).

- Create the GPFS node list file.
  - For Oracle, both nodes will be of type **quorum**. Note that:

    Example: /tmp/gpfs/node_list contains:

```
ravenp:quorum
starp:quorum
```

  - The host name or IP address must refer to the communications adapter over which the GPFS daemons communicate. Alias interfaces are not allowed. Use the original address or a name that is resolved by the host command to that original address.

- Create the GPFS cluster.

- Note that the primary and secondary nodes specified in the **mmcrcluster** command are for managing the **cluster configuration** information, not the primary and secondary NSD servers. Because this is a 2-node configuration, both nodes will be quorum nodes.

```
root@raven: /tmp/gpfs> mmcrcluster -n /tmp/gpfs/node_list -p ravenp -s starp

Thu Jan  6 19:17:13 PST 2005: 6027-1664 mmcrcluster: Processing node ravenp
Thu Jan  6 19:17:16 PST 2005: 6027-1664 mmcrcluster: Processing node starp
mmcrcluster: Command successfully completed
mmcrcluster: 6027-1371 Propagating the changes to all affected nodes.
```

This is an asynchronous process.

c. Display the cluster configuration results as double check:

```
root@raven: /tmp/gpfs> mmlscluster

GPFS cluster information
========================
  GPFS cluster name:             ravenp
  GPFS cluster id:               10383406012703833913
  GPFS UID domain:               ravenp
  Remote shell command:          /usr/bin/rsh
  Remote file copy command:      /usr/bin/rcp

GPFS cluster configuration servers:
-----------------------------------
  Primary server:    ravenp
  Secondary server:  starp

Node number  Node name    IP address       Full node name           Remarks
-------------------------------------------------------------------------
       1         ravenp          144.25.68.193    ravenp        quorum node
       2         starp           144.25.68.192    starp         quorum node
```

and ...

```
root@raven: /tmp/gpfs> mmlsconfig
Configuration data for cluster ravenp:
--------------------------------------------------
clusterName ravenp
clusterId 10383406012703833913
clusterType lc
multinode yes
autoload no
useDiskLease yes
maxFeatureLevelAllowed 806

File systems in cluster ravenp:
---------------------------------------
(none)
```

4. Create the cluster-wide names for the NSDs to be used by GPFS.
   a. Create a file with the list of disks to be used by GPFS:

      Example: /tmp/gpfs/disk_list

      ```
      hdisk5
      hdisk6
      hdisk7
      hdisk8
      ```

      - Because hdisk numbers for the same disk can vary from node to node, these are the hdisk names on the node where the configuration is being done.
      - Use the physical volume identifier (PVID) to identify the same hdisk on each node. If necessary to help identify the same disk on all nodes, use the **chdev** command to assign missing PVIDs as follows:

      ```
      chdev -l hdisk9 -a pv=yes
      ```

      - Do not specify the primary and secondary NSD servers in the disk name file because all nodes will be SAN-attached in this Oracle configuration.
      - Make a copy of this file in case of problems, because it will be modified by the configuration process.

      ```
      cp /tmp/gpfs/disk_list /tmp/gpfs/disk_list_bak
      ```

      - In this example, the designated tie-breaker disks will be part of the file system; therefore, they are also included in this file.

   b. Use the **mmcrnsd** command and the GPFS disk descriptor file just created:

      ```
      root@raven: /tmp/gpfs> mmcrnsd -F /tmp/gpfs/disk_list
      mmcrnsd: Processing disk hdisk5
      mmcrnsd: Processing disk hdisk6
      mmcrnsd: Processing disk hdisk7
      mmcrnsd: Processing disk hdisk8
      mmcrnsd: 6027-1371 Propagating the changes to all affected nodes.
      ```

      This is an asynchronous process.

Note: Below are the new contents of the /tmp/gpfs/disk_list file:

```
# hdisk5
gpfs1nsd:::dataAndMetadata:-1
# hdisk6
gpfs2nsd:::dataAndMetadata:-1
# hdisk7
gpfs3nsd:::dataAndMetadata:-1
# hdisk8
gpfs4nsd:::dataAndMetadata:-1

root@raven: /tmp/gpfs> lspv
hdisk0          00204edabc20fc3f                    rootvg          active
hdisk1          none                                None
hdisk2          0006580c69653fa6                    None
hdisk3          0009005f04c37ea1                    raven_vg        active
hdisk4          00204eda3be7d2dd                    raven_raw_vg    active
hdisk5          000657fc4aa3d756                    None
hdisk6          000657fc4aa3e302                    None
hdisk7          000657fc4aa3f26e                    None
hdisk8          000657fc4aa2a0cc                    None
```

Note: No names are displayed in the volume group fields for the **lspv** command because a desired name was not specified in the original /tmp/gpfs/disk_list file.

Note: These are not actual AIX volume groups.

c. The **mmlsnsd** command can be used to identify the NSD formatted disks:

```
root@raven: /tmp/gpfs> mmlsnsd

 File system   Disk name    Primary node             Backup node
---------------------------------------------------------------------------
 (free disk)   gpfs1nsd     (directly attached)
 (free disk)   gpfs2nsd     (directly attached)
 (free disk)   gpfs3nsd     (directly attached)
 (free disk)   gpfs4nsd     (directly attached)
```

5. Further customize the cluster configuration and designate tie-breaker disks.

a. Change GPFS cluster attributes:

```
mmchconfig prefetchThreads=505
mmchconfig autoload=yes
```

b. Designate the tie-breaker disks:

```
root@raven: />mmchconfig tiebreakerDisks="gpfs1nsd;gpfs2nsd;gpfs3nsd"
Verifying GPFS is stopped on all nodes ...
mmchconfig: Command successfully completed
mmchconfig: 6027-1371 Propagating the changes to all affected nodes.
This is an asynchronous process.
```

c. Display the GPFS cluster configuration:

```
root@raven: /> mmlsconfig
Configuration data for cluster ravenp:
-----------------------------------------------
clusterName ravenp
clusterId 10383406012703833913
clusterType lc
multinode yes
autoload yes
useDiskLease yes
maxFeatureLevelAllowed 806
prefetchThreads 505
tiebreakerDisks gpfs1nsd;gpfs2nsd;gpfs3nsd

File systems in cluster ravenp:
-----------------------------------------
(none)
```

6. Start the GPFS on all nodes:

```
root@raven: />mmstartup -a
Thu Jan 20 19:16:46 PST 2005: 6027-1642 mmstartup: Starting GPFS ...
```

7. Create and mount the GPFS file system.
   a. Use the **mmcrfs** command and the disk descriptor file previously created.

```
root@raven: /> mmcrfs /oragpfs /dev/oragpfs -F /tmp/gpfs/disk_list -B 1024k –N
2000000 -n 8 -A yes
```

```
GPFS: 6027-531 The following disks of oragpfs will be formatted on node ravenp:
    gpfs1nsd: size 10485760 KB
    gpfs2nsd: size 10485760 KB
    gpfs3nsd: size 10485760 KB
    gpfs4nsd: size 104857600 KB
GPFS: 6027-540 Formatting file system ...
Creating Inode File
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
Flushing Allocation Maps
GPFS: 6027-535 Disks up to size 310 GB can be added to this file system.
GPFS: 6027-572 Completed creation of file system /dev/oragpfs.
mmcrfs: 6027-1371 Propagating the changes to all affected nodes.
This is an asynchronous process.
```

b. Display the GPFS file system attributes for verification:

```
root@raven: /> mmlsfs oragpfs
flag value          description
---- -------------- -----------------------------------------------------
 -s  roundRobin        Stripe method
 -f  32768             Minimum fragment size in bytes
 -i  512               Inode size in bytes
 -I  32768             Indirect block size in bytes
 -m  1                 Default number of metadata replicas
 -M  1                 Maximum number of metadata replicas
 -r  1                 Default number of data replicas
 -R  1                 Maximum number of data replicas
 -j  cluster           Block allocation type
 -D  posix             File locking semantics in effect
 -k  posix             ACL semantics in effect
 -a  1048576           Estimated average file size
 -n  8                 Estimated number of nodes that will mount file system
 -B  1048576           Block size
 -Q  none              Quotas enforced
     none              Default quotas enabled
 -F  139264            Maximum number of inodes
 -V  8.01              File system version. Highest supported version: 8.01
 -u  yes               Support for large LUNs?
 -z  no                Is DMAPI enabled?
 -d  gpfs1nsd;gpfs2nsd;gpfs3nsd;gpfs4nsd  Disks in file system
 -A  yes               Automatic mount option
 -E  yes               Exact mtime default mount option
 -S  no                Suppress atime default mount option
 -o  none              Additional mount options
```

c. Mount the GPFS file system:

The GPFS file system is mounted manually the first time by using the standard system **mount** command.

```
root@raven: /> mount /oragpfs
```

d. Allow Oracle user access to the GPFS file system:

```
root@raven: /> chown oracle dba /oragpfs
```

Example 2: Creating a GPFS file system where tie-breaker disks are not part of the GPFS file system

Follow these steps to create a GPFS using GPFS replication and with tie-breaker disks that are not a part of the GPFS:

1. Preinstallation steps:
   a. Set **chdev –l aio0 –a maxservers=20**.
   b. Select and configure the GPFS network.
   c. Add the GPFS interface names to the ***/*.rhosts** file.

2. Install the GPFS software. Note that:
   a. The GPFS fileset names have changed in GPFS V2.3.
   b. GPFS V2.3.0.1 update (APAR IY63969) or higher is mandatory.

```
root@raven: /> lslpp -l |grep -i gpfs
  gpfs.base                 2.3.0.1  APPLIED       GPFS File Manager
  gpfs.msg.en_US            2.3.0.0  COMMITTED     GPFS Server Messages - U.S.
  gpfs.base                 2.3.0.1  APPLIED       GPFS File Manager
  gpfs.docs.data            2.3.0.1  APPLIED       GPFS Server Manpages and
```

   c. Add GPFS to $PATH.
   `export PATH=$PATH:/usr/lpp/mmfs/bin`

3. Create a 2-node GPFS cluster.
   a. Create the node list file:
      - For Oracle, both nodes will be of type **quorum**.
        Example: /tmp/gpfs/node_list contains:
        ```
        ravenp:quorum
        starp:quorum
        ```
      - The host name or IP address must refer to the communications adapter over which the GPFS daemons communicate. Alias interfaces are not allowed. Use the original address or a name that is resolved by the host command to that original address. Because this is a 2-node Oracle configuration, both nodes will be quorum nodes.
   b. Create the GPFS cluster:
      Note: The primary and secondary nodes specified in the **mmcrcluster** command are to manage the **cluster configuration** information, not the primary and secondary NSD servers.

```
root@raven: /tmp/gpfs> mmcrcluster -n /tmp/gpfs/node_list -p ravenp -s starp

Thu Jan  6 19:17:13 PST 2005: 6027-1664 mmcrcluster: Processing node ravenp
Thu Jan  6 19:17:16 PST 2005: 6027-1664 mmcrcluster: Processing node starp
mmcrcluster: Command successfully completed
mmcrcluster: 6027-1371 Propagating the changes to all affected nodes.
```

      This is an asynchronous process.

c. Display the configuration results as double check:

```
root@raven: /tmp/gpfs> mmlscluster

GPFS cluster information
========================
GPFS cluster name:         ravenp
GPFS cluster id:           10383406012703833913
GPFS UID domain:           ravenp
Remote shell command:      /usr/bin/rsh
Remote file copy command:  /usr/bin/rcp

GPFS cluster configuration servers:
-----------------------------------
  Primary server:    ravenp
  Secondary server:  starp
```

```
 Node number  Node name    IP address       Full node name              Remarks
-------------------------------------------------------------------------------
    1         ravenp           144.25.68.193    ravenp             quorum node
    2         starp            144.25.68.192    starp              quorum node
```

```
root@raven: /tmp/gpfs> mmlsconfig
Configuration data for cluster ravenp:
--------------------------------------------------
clusterName ravenp
clusterId 10383406012703833913
clusterType lc
multinode yes
autoload no
useDiskLease yes
maxFeatureLevelAllowed 806

File systems in cluster ravenp:
-------------------------------------------(none)
```

4. Create the cluster-wide names for the NSDs to be used for the GPFS file system.
   a. Create the GPFS disk descriptor file:
      - The format for each entry is:

```
DiskName:PrimaryServer:BackupServer:DiskUsage:FailureGroup:DesiredName
```

      - The **FailureGroup** field is used to indicate that hdisk 21 and 22 will be in failure group **1**, while hdisk 22 and 23 will be in failure group **2**.
      - The **DesiredName** is specified and will appear in the volume group field when the **lspv** command is used. (**Note:** These are not actual AIX 5L volume groups.)
      - Contents of the file for the data disks /tmp/gpfs/disk_list_data are listed below:

```
hdisk21::::1:fg1a
hdisk22::::1:fg1b
hdisk23::::2:fg2a
hdisk24::::2:fg2b
```

      - Make a copy of this file in case of problems, because it will be modified during the configuration process.

```
cp /tmp/gpfs/disk_list_data /tmp/gpfs/disk_list_data_bak
```

   b. NSD format the data disks using the GPFS disk descriptor file created for data disks.

```
root@raven: />mmcrnsd -F /tmp/gpfs/disk_list_data

mmcrnsd: Processing disk hdisk21
mmcrnsd: Processing disk hdisk22
mmcrnsd: Processing disk hdisk23
mmcrnsd: Processing disk hdisk24
mmcrnsd: 6027-1371 Propagating the changes to all affected nodes.
```

      This is an asynchronous process.

c. Display the results of the command as a double check:

```
root@raven: />mmlsnsd
```

```
 File system   Disk name    Primary node                 Backup node
---------------------------------------------------------------------------
 (free disk)   fg1a         (directly attached)
 (free disk)   fg1b         (directly attached)
 (free disk)   fg2a         (directly attached)
 (free disk)   fg2b         (directly attached)

root@raven: />lspv

hdisk0          00c5430c0026114d                    rootvg          active
hdisk1          00c5430cbbcfe588                      None
hdisk2          00c5430c92db553f                      None
hdisk10         00c5430c0cdcd979                      None
hdisk11         00c5430c0cdcdc32                      None
hdisk12         00c5430c0cdcdedc                      None
hdisk13         00c5430c0cdce198                      None
hdisk14         00c5430c0cdce462                      None
hdisk15         00c5430c0cdce9da                      None
hdisk16         00c5430c0cdcec98                      None
hdisk17         00c5430c0cdcef4f                      None
hdisk18         00c5430c0cdcf20e                      None
hdisk19         00c5430c0cdcf4c9                      None
hdisk20         00c5430c0cdcf747                      None
hdisk21         00c5430c0cdcf976                      fg1a
hdisk22         00c5430c0cdcfbb3                      fg1b
hdisk23         00c5430c92db71ea                      fg2a
hdisk24         00c5430c0cdcfde3                      fg2b
```

5. Create the cluster-wide names for the NSDs to be used for the tie-breaker disks.
   a. Create the GPFS disk descriptor file for the tie-breaker disks.
      - The format for each entry is:

```
DiskName:PrimaryServer:BackupServer:DiskUsage:FailureGroup:DesiredName
```

      - If the **DesiredName** is specified, this name will appear in the volume group field when the **lspv** command is used. (**Note:** These are not actual volume groups.)
      - Contents of the file for the data disks /tmp/gpfs/disk_list_ are listed below:

```
hdisk18:::::tie1
hdisk19:::::tie2
hdisk20:::::tie3
```

b. NSD format the tie-breaker disks using the GPFS disk descriptor file that was created for the tie-breaker disks:

```
root@n80:64bit /tmp/gpfs> mmcrnsd -F /tmp/gpfs/disk_list_tie

mmcrnsd: Processing disk hdisk18
mmcrnsd: Processing disk hdisk19
mmcrnsd: Processing disk hdisk20
mmcrnsd: 6027-1371 Propagating the changes to all affected nodes.
```

This is an asynchronous process.

c. Display the results of the command as a double check:

```
root@n80:64bit /tmp/gpfs> mmlsnsd

 File system   Disk name    Primary node             Backup node
---------------------------------------------------------------------------
 (free disk)   fg1a         (directly attached)
 (free disk)   fg1b         (directly attached)
 (free disk)   fg2a         (directly attached)
 (free disk)   fg2b         (directly attached)
 (free disk)   tie1         (directly attached)
 (free disk)   tie2         (directly attached)
 (free disk)   tie3         (directly attached)
```

6. Identify the tie breaker disks to the cluster configuration by using the **mmchconfig** command:

```
root@raven: />mmchconfig tiebreakerDisks="tie1;tie2;tie3"

Verifying GPFS is stopped on all nodes ...
mmchconfig: Command successfully completed
mmchconfig: 6027-1371 Propagating the changes to all affected nodes.
```

This is an asynchronous process.

a. Display the tie-breaker disks in a new configuration:

```
root@raven: /> mmlsconfig

Configuration data for cluster ravenp:
--------------------------------------
clusterName ravenp
clusterId 13882357191189485225
clusterType lc
multinode yes
autoload yes
useDiskLease yes
maxFeatureLevelAllowed 806
tiebreakerDisks tie1;tie2;tie3
prefetchThreads 505

File systems in cluster ravenp:
-------------------------------
(none)
```

7. Start the GPFS on all nodes:

```
root@raven: />mmstartup -a
Thu Jan 20 19:16:46 PST 2005: 6027-1642 mmstartup: Starting GPFS ...
```

8. Create and mount the GPFS file system.
   a. Create the GPFS file system using the **mmcrfs** command and the disk descriptor file previously created:

```
root@raven: /> mmcrfs /oragpfs /dev/oragpfs -F /tmp/gpfs/disk_list_data –B
1024K –n 8 –A yes

GPFS: 6027-531 The following disks of oragpfs will be formatted on node n80:
    fg1a: size 17796014 KB
    fg1b: size 17796014 KB
    fg2a: size 17796014 KB
    fg2b: size 17796014 KB
GPFS: 6027-540 Formatting file system ...
Creating Inode File
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
Flushing Allocation Maps
GPFS: 6027-535 Disks up to size 84 GB can be added to this file system.
GPFS: 6027-572 Completed creation of file system /dev/oragpfs.
mmcrfs: 6027-1371 Propagating the changes to all affected nodes.
```

   This is an asynchronous process.

   b. Mount the GPFS file system:
      - The GPFS file system is mounted manually the first time by using the standard system **mount** command.

```
root@raven: /> mount /oragpfs
```

   c. Allow Oracle user access to the GPFS file system:

```
root@raven: /> chown oracle dba /oragpfs
```

Migration to GPFS V2.3 from a previous version

The following items must be considered before migrating to GPFS V2.3:

- In previous GPFS releases, it was possible to configure clusters using one of the following cluster types: sp, hacmp, rpd, or lc.
  - In GPFS V2.3, only the lc cluster type is supported implicitly.
- In previous GPFS releases, each cluster type supported different disk types, such as virtual shared disks, AIX 5L logical volumes, or NSDs.
  - In GPFS V2.3, only the NSD disk type is supported.
- Prior to GPFS V2.3, it was possible to divide a GPFS cluster into a number of node sets, which determined all the nodes of the GPFS cluster where a GPFS file system was to be mounted.
  - In GPFS V2.3, the concept of a node set is removed. All nodes in the GPFS V2.3 cluster are automatically members of one, and only one, node set.

These new features of GPFS V2.3 require that the GPFS cluster be rebuilt and migrated.

The following example migration is based on the migration instructions that can be found in the GPFS documentation.

Example migration steps

Follow these migration steps:

1. Ensure that all disks in the file systems to be migrated are in working order by issuing the **mmlsdisk** command and checking for the disk status to be **ready** and availability to be **up**:

```
root@kerma / > mmlsdisk /dev/gpfsdisk
disk         driver       sector failure holds    holds
name         type       size   group metadata data  status        availability
------------ -------- ------ ------- --------      ----- ------------ --------
gpfs35lv     disk        512       1      yes      yes   ready         up
gpfs36lv     disk        512       1      yes      yes   ready         up
gpfs37lv     disk        512       1      yes      yes   ready         up
```

   - Stop all user activity on the file systems to be migrated, and make a backup of critical user data to ensure protection in the event of a failure.
   - Cleanly unmount all mounted GPFSs file systems from all cluster nodes. Do not to use the **force** option to unmount the file system on any node.

2. Shut down the GPFS daemons on all nodes of the cluster:

```
root@kerma / > mmshutdown -a
Fri Mar 18 14:50:49 PST 2005: 6027-1341 mmshutdown: Starting force unmount of
GPFS file systems
Fri Mar 18 14:50:54 PST 2005: 6027-1344 mmshutdown: Shutting down GPFS daemons
jnanag:  Shutting down!
kermag:  Shutting down!
jnanag:  0513-044 The mmfs Subsystem was requested to stop.
kermag:  0513-044 The mmfs Subsystem was requested to stop.
jnanag:  Master did not clean up; attempting cleanup now
jnanag:  /var/mmfs/etc/mmfsdown.scr: /usr/bin/lssrc -s mmfs
jnanag:  Subsystem         Group            PID          Status
jnanag:   mmfs             aixmm                         inoperative
jnanag:  /var/mmfs/etc/mmfsdown.scr: /usr/sbin/umount -f -t mmfs
jnanag:  Fri Mar 18 14:51:37 2005: GPFS: 6027-311 mmfsd64 is shutting down.
jnanag:  Fri Mar 18 14:51:37 2005: Reason for shutdown: mmfsadm shutdown
command timed out
Fri Mar 18 14:51:58 PST 2005: 6027-1345 mmshutdown: Finished
```

3. Export the GPFS file systems (using the **mmexportfs** command).
   - This command creates a configuration output file that will be required, when finishing the migration, to import the file system to the new GPFS V2.3 cluster. Preserve this file. It will also be used in case there is a need to go back to older versions of GPFS.

```
root@kerma /tmp/gpfs22 > mmexportfs all -o gpfs22.con

mmexportfs: Processing file system gpfsdisk ...

mmexportfs: Processing disks that do not belong to any file system ...
mmexportfs: 6027-1371 Propagating the changes to all affected nodes.
```

   This is an asynchronous process.

4. Delete all existing nodes for each node set in the cluster (using the **mmdelnode** command):

```
root@kerma /tmp/gpfs22 > mmdelnode -a
Verifying GPFS is stopped on all affected nodes ...
mmdelnode: 6027-1370 Removing old nodeset information from the deleted nodes.
```

   This is an asynchronous process.

   In case there is more than one node set:

```
root@kerma / > mmdelnode –a –C nodesetid
```

5. Delete the existing cluster by issuing the **mmdelcluster** command. (This command is only available with GPFS V2.2 and earlier releases.)

```
root@kerma /tmp/gpfs22 > mmdelcluster -a
mmdelcluster: 6027-1371 Propagating the changes to all affected nodes.
```

   This is an asynchronous process.

```
mmdelcluster: Command successfully completed
```

6. Install the GPFS V2.3 software on all of the cluster nodes.
   - The GPFS V2.3 install images have been copied to /tmp/gpfslpp on all nodes.

```
root@kerma / > installp –agXYd /tmp/gpfslpp gpfs

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
gpfs.msg.en_US              2.3.0.1         USR         APPLY       SUCCESS
gpfs.docs.data              2.3.0.1         SHARE       APPLY       SUCCESS
gpfs.base                   2.3.0.1         USR         APPLY       SUCCESS
gpfs.base                   2.3.0.1         ROOT        APPLY       SUCCESS
```

7. Determine which nodes will be quorum nodes in the GPFS cluster and create a new GPFS cluster across all desired cluster nodes (using the **mmcrcluster** command):

```
root@kerma /var/mmfs/gen >  mmcrcluster -n  nodefile -p kermag -s jnanag
Mon Mar 21 11:25:08 PST 2005: 6027-1664 mmcrcluster: Processing node kermag
Mon Mar 21 11:25:10 PST 2005: 6027-1664 mmcrcluster: Processing node jnanag
mmcrcluster: Command successfully completed
mmcrcluster: 6027-1371 Propagating the changes to all affected nodes.
```

This is an asynchronous process.

Where the contents of the node file are:

```
root@kerma /var/mmfs/gen > cat nodefile
kermag:quorum
jnanag:quorum
```

8. Complete the movement of the GPFS file system to the new cluster (using the **mmimportfs** command):

```
root@jnana /tmp/gpfs22 > mmimportfs gpfsdisk  -i gpfs22.con
mmimportfs:  Attempting to unfence the disks.  This may take a while ...

mmimportfs: Processing file system gpfsdisk ...
mmimportfs: Processing disk gpfs35lv
mmimportfs: Processing disk gpfs36lv
mmimportfs: Processing disk gpfs37lv

mmimportfs: Committing the changes ...

mmimportfs: The following file systems were successfully imported:
        gpfsdisk
mmimportfs: The NSD servers for the following disks from file system gpfsdisk
were reset or not defined:
        gpfs35lv
        gpfs36lv
        gpfs37lv
mmimportfs: Use the mmchnsd command to assign NSD servers as needed.
mmimportfs: 6027-1371 Propagating the changes to all affected nodes.
```

This is an asynchronous process.

9. Start the GPFS on all nodes of the cluster (using the **mmstartup** command):

```
root@jnana / > mmstartup -a
Mon Mar 21 14:15:22 PST 2005: 6027-1642 mmstartup: Starting GPFS ...

root@jnana / > df /gpfs
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/gpfsdisk  106756608 106661888    1%       76     1% /gpfs
```

10. Complete the migration to the new level of GPFS (using the **mmchfs** command).

Mount the file system if it is not already mounted:

```
root@jnana / > mount /gpfs
```

Issue the following command to migrate the file system metadata to the new GPFS V2.3 format:

```
root@jnana / > mmchfs gpfsdisk –V
```

Summary

The IBM General Parallel File System (GPFS) is a high-performance shared-disk file system that can provide fast database access from all nodes in a homogenous or heterogeneous cluster of AIX systems. GPFS allows parallel applications simultaneous access to the same files, or different files, from any node that has the GPFS mounted. GPFS provides high availability, backup, and failover support

GPFS V2.3 for AIX 5L on POWER scales file system I/O to meeting the objectives of a wide range of applications, including (as discussed in this paper) Oracle RAC and Oracle Database 10*g* RAC (10.1.0.x and 10.2.0.1).

This paper has demonstrated two scenarios for creating a GPFS and has even stepped through the migration process to bring an older release of a GPFS up to the latest release. GPFS V2.3 offers many improvements over the previous release of this file system and is well worth investigating. These improvements include a new, streamlined cluster type, the ability to share file systems among multiple GPFS clusters through virtual connections, more powerful commands, and much more.

The **Resources** section of this paper is particularly robust in the number of touch points it provides for those who are ready to delve into this powerful product for the Oracle environment.

Resources

These Web sites provide useful references to supplement the information contained in this document:

GPFS

The GPFS FAQ site contains the latest information regarding GPFS including software requirements, supported hardware, and supported storage configurations.

- GPFS V2.3 FAQ
  publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq
- GPFS forum at IBM developerWorks
  ibm.com/developerworks/forums/dw_forum.jsp?forum=479&cat=13
- GPFS V2.3 summary of changes
  publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs23/bl1ins10/bl1ins1012
- GPFS V2.3 announcement details
  ibm.com/common/ssi/OIX.wss?DocURL=http://d03xhttpcl001g.boulder.ibm.com/common/ssi/rep_ca/4/897/ENUS204-294/index.html&InfoType=AN
- GPFS product home page
  ibm.com/servers/eserver/clusters/software/gpfs.html
- GPFS V2.3 publications
  publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfsbooks
- Direct to main pages for GPFS V2.3 commands
  publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs23/bl1adm10/bl1adm1069
- Migrating to GPFS V2.3 from previous GPFS versions
  publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs23/bl1ins10/bl1ins1037
- Establishing disaster recovery for your GPFS cluster
  publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs23/bl1adm10/bl1adm1060

EtherChannel and Link Aggregation with AIX 5L

- AIX EtherChannel and IEEE 802.3ad Link Aggregation
  publib.boulder.ibm.com/infocenter/pseries/index.jsp?topic=/com.ibm.aix.doc/aixbman/commadmn/tcp_etherchannel

General IBM information

- IBM eServer™ pSeries® and AIX 5L Information Center
  publib.boulder.ibm.com/infocenter/pseries/index.jsp
- IBM Publications Center
  www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US
- **nmon** performance tool: A free tool to analyze AIX 5L and Linux performance
  ibm.com/developerworks/eserver/articles/analyze_aix/index.htm

Oracle Metalink

- Oracle Metalink
  metalink.oracle.com

Oracle and AIX 5L

- Oracle 10*g* Release 1

  Oracle Database Administrator's Reference 10*g* Release 1 (10.1) for UNIX Systems: AIX-Based Systems, hp HP-UX PA-RISC (64-bit), hp Tru64 UNIX, Linux x86, and Solaris Operating System (SPARC) (Part Number: B10812-01)

  - Appendix A: Administering Oracle Database on AIX
    download-west.oracle.com/docs/html/B10812_01/appendix_a.htm#sthref575
  - Chapter 8: Tuning for Oracle Database on UNIX
    download-west.oracle.com/docs/html/B10812_01/chapter8.htm#sthref441
- Oracle 10*g* Release 2

  Oracle Database Administrator's Reference 10*g* Release 2 (10.2) for UNIX-Based Operating Systems (Part number: B15658-04)

  - Appendix A: Administering Oracle Database on AIX
    download-west.oracle.com/docs/cd/B19306_01/server.102/b15658/appa_aix.htm#sthref723
  - Chapter 8: Tuning for Oracle Database on UNIX
    download-west.oracle.com/docs/cd/B19306_01/server.102/b15658/tuning.htm#sthref573

Oracle and Linux on POWER

- Oracle 10*g* Release 2

  Oracle Database Administrator's Reference 10*g* Release 2 (10.2) for UNIX-Based Operating Systems (Part number: B15658-04)

  - Appendix C: Administering Oracle Database on Linux
    download-west.oracle.com/docs/cd/B19306_01/server.102/b15658/appc_linux.htm#sthref870

About the author

Rick Piasecki, IBM eServer Solution Enablement

Rick Piasecki is a senior software programmer in the IBM eServer Solutions Enablement organization, working onsite at Oracle Corporation. He is involved in the enablement and support of Oracle products on the IBM eServer pSeries platforms. He holds a Bachelor of Science in Biology degree from the University of Connecticut and a Bachelor of Computer Science degree from Florida Atlantic University. Rick can be contacted at piasecki@us.ibm.com.

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	RedBooks	AIX	TotalStorage
ibm.com	pSeries	AIX 5L	Enterprise Storage Server
the IBM logo	HACMP	POWER	FlashCopy

Red Hat, the Red Hat "Shadow Man" logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.