IBM® DB2® for Linux®, UNIX®, and Windows®

Configuring Geographically Dispersed DB2® pureScale™ Clusters

Jeremy Brumer
DB2 pureScale System Verification Test

Roy Cecil
DB2 Performance Team

Fabrizio Fabbri
DB2 Software Development Team

Steve Rees
Senior Performance Manager
DB2 Software Development

Last updated: April 2011

Introduction

The DB2® pureScale™ Feature provides outstanding database scalability, availability and application transparency on AIX® and Linux® platforms, building on the data sharing architecture of the 'gold standard', DB2 for z/OS® Parallel Sysplex®. However, any single-site system, even DB2 pureScale systems or the DB2 for z/OS Parallel Sysplex, can be vulnerable to external events that might compromise power or communications, for example, or even cause physical damage to the system.

Because disasters like power failures and fires might easily disable a single data center, many large IT organizations configure two sites, far enough apart to be on separate power grids. This configuration minimizes the risk of total outage, and allows business to carry on at one site, even if the other is impacted by a disaster. This paper describes the geographically dispersed DB2 pureScale cluster (or GDPC for short). Like the Geographically Dispersed Parallel Sysplex™ configuration of DB2 for z/OS, GDPC provides the scalability and application transparency of a regular single-site DB2 pureScale cluster, but in a cross-site configuration which enables 'active/active' system availability, even in the face of many types of disaster.

The active/active part is important because it means that during normal operation, the DB2 pureScale members at both sites are sharing the workload between them as usual, with workload balancing (WLB) maintaining an optimal level of activity on all members, both within and between sites. This means that the second site is not a standby site, waiting for something to go wrong. Instead, the second site is pulling its weight, returning value for investment even during day-to-day operation.

This paper describes the prerequisites for a geographically dispersed DB2 pureScale cluster, followed by the steps to one deploy one, as well as some of the performance implications of different site-to-site distances and different workload types. It is assumed that the reader is familiar with the prerequisites for a non-GDPC pureScale cluster, and that those are met.

GDPC Concepts

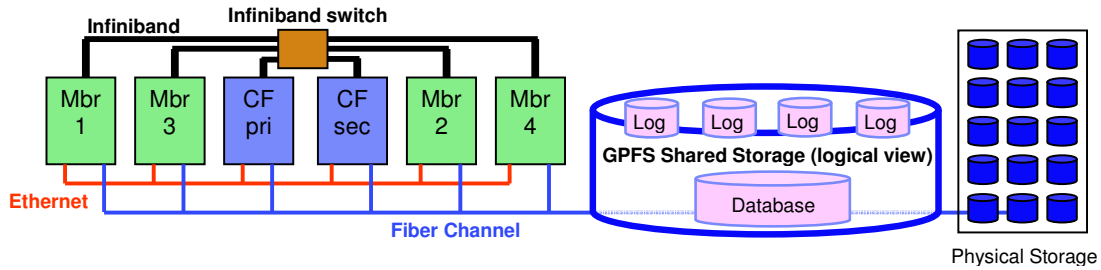
A typical DB2 pureScale cluster consists of, among other things:

- two[1] or more DB2 pureScale members
- two cluster caching facilities (CF)
- SAN-attached cluster storage running IBM® General Parallel File System (GPFS™)
- a high-speed, low-latency cluster interconnect such as InfiniBand (IB).

Figure 1 shows such a configuration, with four members and two CFs using InfiniBand for low-latency communications. The DB2 pureScale Feature is a shared-data architecture, in which all members are operating on a single copy of the database, communicating with each other via the CF to synchronize activities and to ingest, modify and retrieve data as required by the application. Messages between the members and CF use the remote direct memory access (RDMA) capability in the cluster interconnect, which provides extremely low communication latencies as well as very low CPU utilization per message.

[1] DB2 pureScale Feature supports clusters as small as one member and one CF, but typical clusters have at least two of each.

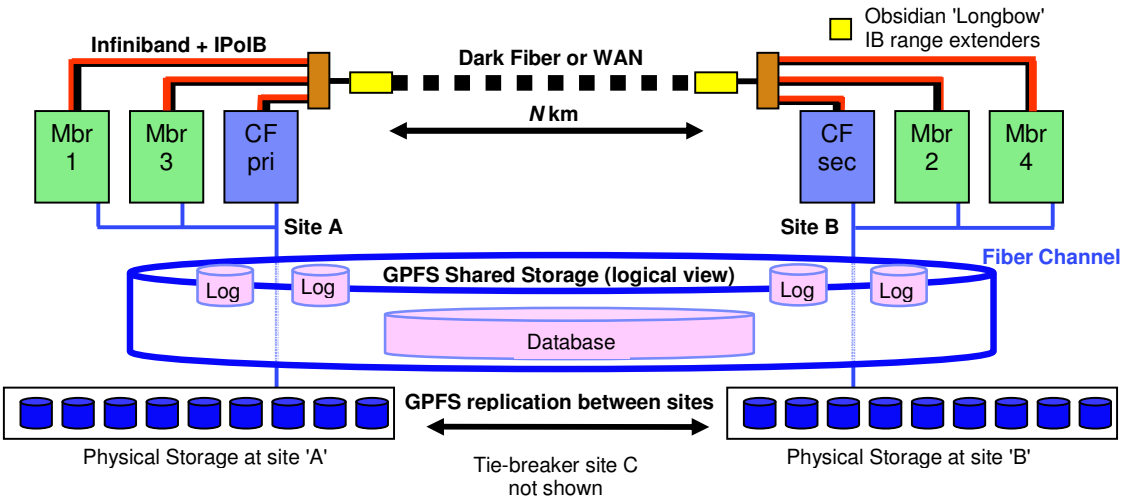
Figure 1. Typical DB2 pureScale configuration



Splitting a DB2 pureScale cluster in half across two sites A & B implies that half of the member systems will be physically located at site A and half at site B. (A third site with minimal configuration is actually required for tie-breaking purposes. See below for details.) Of course, one CF should be placed at each of the two main sites as well, to avoid a single point of failure (SPOF). In order to maintain the DB2 pureScale software's excellent performance and scalability, we need to continue to use an RDMA-capable interconnect between sites, so that messages from a member at one site to the CF at the other site are as fast and inexpensive as possible. The spanning distance of an InfiniBand network is typically measured in tens or maybe hundreds of meters, however devices such as the Obsidian Longbow InfiniBand extender allow the reach of an IB network to span greater distances, over wide-area networks or dedicated fiber optic links.

In addition to the dispersal of computing resources such as members and CFs, a disaster recovery (DR) cluster configuration also requires storage to be replicated across sites. Building on the standard DB2 pureScale cluster design, the GDPC configuration uses GPFS synchronous replication between sites to keep all disk write activity up-to-date across the cluster. This includes both table space writes and transaction log writes. At a high level, a GDPC cluster might look similar to the one illustrated in figure 2.

Figure 2. Typical geographically dispersed DB2 pureScale cluster



Client applications connecting to the DB2 pureScale cluster typically have workload balancing (WLB) enabled, which will transparently route work to the member with the most available capacity. WLB maintains optimal use of resources during normal operation, and also reroutes connections in case of member downtime (planned or unplanned), or even site failure. The client systems, often configured as application servers in a multitier environment, are often configured with redundancy across sites, providing fault tolerance at the upper layers as well.

GDPC Infrastructure and Prerequisites

Site-to-site Connectivity

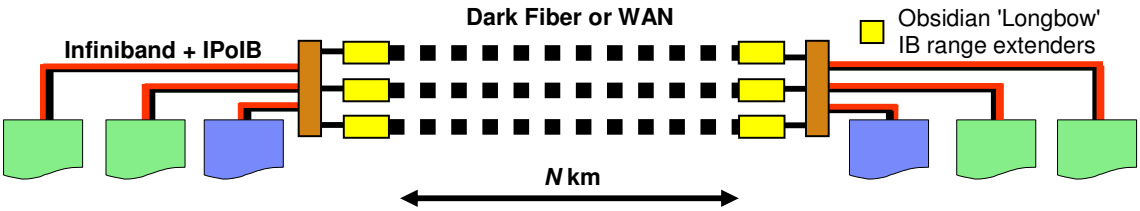
The connection between sites is a key piece of infrastructure in a GDPC. As described above, DB2 pureScale software uses low-latency, high-bandwidth RDMA messaging between members and CFs, and in a dispersed configuration, many such messages traverse the link from one site to the other. 'Longbow' InfiniBand extender technology from Obsidian Strategics[2] provides a transparent connection between the two portions of the IB network which are located at the two sites, and maintains the ability to execute RDMA operations across the GDPC, even at relatively large distances. Used in pairs at either end of the site-to-site interconnect, the extenders accept an IB connection to a site-local IB switch, and through it, to the members and CF. The extender translates IB traffic to and from packets that are sent and received over the site-to-site interconnect (either a 'dark fiber' or 10Gb WAN connection).

The Longbow IB extenders themselves add only a very small amount of extra latency to the DB2 pureScale message protocol. The bulk of the extra site-to-site message latency, when compared to a single-site DB2 pureScale cluster, arises from the simple fact of distance: each kilometer of transmission in glass fiber adds an additional 5 microseconds of delay. So for example, a 10km distance between sites would add (10km x 5 microseconds/km) x 2 for round trip = 100 microseconds of extra latency for most types of messages. In practical terms, workloads that have higher ratios of read activity compared to write activity tend to trigger fewer message exchanges with the CF, and so would be less likely to be impacted by additional site-to-site latency.

Current Longbow IB extenders operate at the 4X width SDR (Single Data Rate, or 10Gbit) data rate between end points (subject to the capacity of the dark fiber / WAN link.) This represents one half of the peak capacity of the IBM IB network adapters currently supported by the DB2 pureScale Feature on AIX. If redundancy or additional cross-site capacity is required, Longbow units can be 'ganged' in multiple pairs between sites (see Figure 3). As well, different Longbow models provide different features which may be useful in certain circumstances, such as encryption in the E-100 and X-100 models, which might be important if the site-to-site interconnect is shared or public and encryption is required by security policies. All Longbow models are supported with dispersed DB2 pureScale clusters. Particular configurations, such as the choice of model, use of WAN or fiber, or choice of transceiver wavelength, and other characteristics, are not specified here, and should be selected based on the physical infrastructure to be used, IT policies in effect, and so on. For more information about Longbow IB extenders, please contact Obsidian Research (http://www.obsidianresearch.com/).

[2] http://www.obsidianresearch.com/

Figure 3. Using multiple IB extender pairs



Two Site / Three Site Configurations

A dispersed DB2 pureScale cluster is composed of two main sites A and B, with each typically having an equal number of members and CFs. For example, if site A has two members and one CF, site B will also generally have two members and one CF. One additional tiebreaker host T is required to maintain quorum in the event of site failure. This additional host does not run any DB2 members or CFs, and is preferably located at a separate third site (site C), but may optionally be located at site A if a third site is not available. In such a two-site configuration, when the tiebreaker host T is located at site A, we refer to that site as the primary site.

A three-site configuration is resilient to failure of any one site. This means that if a failure should occur at site A or B, or even at the tie-breaker site C, the DB2 pureScale instance continues to operate. This is because the two surviving sites are able to maintain quorum. The tiebreaker site requires only TCP/IP access to the two main sites A & B (not SAN access), so the three-site topology is strongly recommended.

Comparatively, a two-site configuration will permit a loss of availability of the DB2 pureScale instance if there is a total site failure at the primary site A. Primary site failure will require manual steps to bring the instance back online at site B if primary site A remains offline. In a two-site configuration, it is recommended that the tie-breaker host T be configured on a separate physical computers from that of the members and CFs at site A, to ensure that quorum can be maintained even in the event of a complete failure of the computers used for DB2 members and CFs at site A.

Zoned SAN storage

GDPC requires that both sites A and B have direct access to each others' disks. To this end, a number of options are available for extending a SAN across two data centers. Options include transmitting Fibre Channel (FC) traffic directly over ATM or IP networks, or using iSCSI to transmit SCSI commands over IP. Dark fiber is likely to be the fastest but also the most expensive option. Note that the tiebreaker site in a GDPC configuration does not require a SAN connection to sites A and B; it needs only a TCP/IP connection to the other sites.

GPFS Synchronous Replication

A typical non-dispersed DB2 pureScale cluster uses GPFS in a non-replicated configuration. In such a case, all GPFS disk activity for a given file system goes to a single GPFS failure group. When disks are not replicated, a disk failure can leave some of the file system data inaccessible. For a dispersed DB2 pureScale cluster, however, GPFS replication is used between sites A & B in order to avoid storage becoming a single point of failure.

The GDPC configuration leverages GPFS replication[3], by configuring each site to maintain an entire copy of the file system data in its own failure group. As long as quorum is maintained on a separate host, in the event of a site failure (one of the failure groups are lost or inaccessible), the other site can continue with read/write access to the file system.

Host T requires a small disk or partition for each replicated GPFS file system in use by DB2 pureScale software, to be used as a file system quorum disk. The amount of storage for each disk or partition is approximately 50 MB, and these disks or partitions only need to be accessible by host T. Using a full physical volume for this purpose is wasteful and not necessarily practical; configuring a small LV is sufficient for this case..

GDPC Setup and Configuration

Let us assume the following hardware configuration:

Hosts:

Site A: Hosts	**hostA1, hostA2, hostA3**
Site B: Hosts	**hostB1, hostB2, hostB3**
Site C: Host	**T**

Equally-sized LUNs have been provisioned on storage at sites A and B, and all LUNs are accessible by all hosts at sites A & B.

For our examples, LUNs on disks located at Site A are:

```
/dev/hdiskA1
/dev/hdiskA2
/dev/hdiskA3
/dev/hdiskA4
/dev/hdiskA5
/dev/hdiskA6
/dev/hdiskA7
```

Note: for clarity, we use hdisk names which indicate the site where the disk is located

LUNs on disks located at Site B are:

```
/dev/hdiskB1
/dev/hdiskB2
```

[3] For more information on GPFS synchronous replication, see the Advanced Administration Guide at http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=%2Fcom.ibm.cluster.gpfs33.advanceadm.doc%2Fbl1adv00_xtoc.html

```
/dev/hdiskB3
/dev/hdiskB4
/dev/hdiskB5
/dev/hdiskB6
/dev/hdiskB7
```

LUNs on disks located at Site C are as follows. Theses disks can be 50MB logical volumes.

```
/dev/hdiskC1
/dev/hdiskC2
/dev/hdiskC3
```

Other than minimal changes to **HostFailureDetectionTime** and SCSI-3 PR capability discussed below (which account for the long-distance communication topology), the only other difference between GPFS as used in a regular DB2 pureScale instance and in a GDPC is the enablement of synchronous replication.

To enable GPFS synchronous replication, first assign all disks within site A to one GPFS failure group, and all disks within site B to another. This example uses the following setup:

Disks on Site A to the GPFS failure group 1
Disks on Site B to the GPFS failure group 2
Disks on Site C to the GPFS failure group 3

The GDPC is setup in the following way:

- Database MYDB is to be created on instance db2inst1
- MYDB will have three file systems
    1. logfs for transaction logs and database metadata
    2. datafs for database containers
    3. db2fs1 the shared file system for instances.

A three-site configuration and two-site configuration are similar in that they both have a tie-breaker host **T** – however in the 3-site configuration, **T** is located at a separate third site C. In a two-site configuration, the host **T** is co-located within site A. A three-site configuration provides simplified recoverability, so it is highly preferred over a 2-site configuration. In either case, each of site A & B usually each has a CF, and an equal number of members.

Each command is specified with the following format:

```
uid@host> command
```

uid
the user ID that executes the command.
host
where the command should be executed.
command
the command to execute.

STEP 1: Install the DB2 pureScale Feature on all hosts.

Install the DB2 pureScale software on all the hosts (**hostA1**, **hostA2**, **hostB1**, **hostB2**, **hostA3**, **hostB3**) using **db2setup**. Note that even though DB2 pureScale code is installed on host **T**, its only participation is as a GPFS tie-breaker.

Using the Advanced Configuration menu, designate **hostA3** and **hostB3** as the CFs and (optionally) one of the two to be the preferred primary CF. **Note:** a DB2 Cluster Services Tie-breaker Disk is not required since this is a GDPC. Instead we will use a host as a tie-breaker.

The file system that **db2setup** will create for the shared instance metadata will initially be a non-replicated GPFS file system, because **db2setup** does not yet have an option that allows specifying a replicated file system. We will convert this later to a replicated file system across the sites.

At this point, you have achieved the following implicitly via **db2setup**:

- Created an RSCT domain across hosts **hostA1**, **hostA2**, **hostA3**, **hostB1**, **hostB2**, **hostB3**
- Created a GPFS cluster across hosts **hostA1**, **hostA2**, **hostA3**, **hostB1**, **hostB2**, **hostB3**.
- Created a non-replicated GPFS file system **db2fs1** using the disk **/dev/hdiskA1** which is mounted on a path like **/db2sd_20110224005651.** The following operation can be used to verify the mount point:

```
root@hostA1:/> db2cluster -cfs -list -filesystem
File system NAME        MOUNT_POINT
--------------------    -------------------------
db2fs1                  /db2sd_20110224005651
```

STEP 2: Adding tie-breaker host T to the cluster.

Install on host T the IBM DB2 pureScale Feature for Enterprise Server Edition using the **db2_install** command

```
root@T> db2_install
```

You need to ensure that the SCSI-3 PR is disabled in the GPFS cluster configuration.
Check if SCSI-3 is enabled:

```
root@hostA1> /usr/lpp/mmfs/bin/mmlsconfig usePersistentReserve
```

If enabled, then disable it:

```
root@hostA1> db2cluster -cfs -stop -all

root@hostA1> /usr/lpp/mmfs/bin/mmchconfig usePersistentReserve=no

root@hostA1> db2cluster -cfs -start -all
```

Each site must contain a GPFS cluster configuration server, which will preserve the configuration in case of a disaster on one site. Use the command **mmchcluster** to change the configuration servers, where **hostA1** and **hostB1** are located at different sites.

```
root@hostA1> /usr/lpp/mmfs/bin//mmchcluster -p hostA1
-s hostB1
```

Next, **HostFailureDetectionTime** is increased to a higher value than what would be set on a non-GDPC DB2 pureScale cluster.

```
root@hostA1> db2cluster -cfs -stop -all

root@hostA1> db2cluster -cm -set -option HostFailureDetectionTime -
value 16

root@hostA1> db2cluster -cfs -start -all
```

Changing this value allows for the increased communication lag between sites that isn't present in a non-GDPC DB2 pureScale cluster. If unexpected host down events are still triggered due to large inter-site distances, higher parameter values may be used, however this will increase the DB2 pureScale crash recovery time.

The tiebreaker host provides cluster quorum, ensuring that during normal operation, the cluster contains an odd number of hosts. In case of a network outage between sites, only the site which can communicate with tiebreaker host **T** will gain cluster quorum.

Change the quorum type to majority node.

```
root@hostA1> db2cluster -cm -set -tiebreaker -majority

root@hostA1> db2cluster -cfs -set -tiebreaker -majority
```

The command **preprpnode** is run on host **T**, to add it into the RSCT domain, and then T is added to the cluster.

```
root@T> preprpnode hostA1 hostA2 hostB1 hostB2 hostA3 hostB3

root@hostA1> db2cluster -cm -add -host T
```

Host **T** is added into the GPFS cluster. This is done directly with the GPFS **mmaddnode** command, because we want to mark this host as a quorum client so that it will never run as file system manager, token manager, or other role.

```
root@hostA1> /usr/lpp/mmfs/bin/mmaddnode T:quorum-client

root@T> db2cluster -cfs -add -license
```

Next, ensure that we do not get false disk errors since **T** is not connected to the SAN.

```
root@hostA1> /usr/lpp/mmfs/bin/mmchconfig unmountOnDiskFail=yes -N T
```

The GPFS cluster is changed to use the InfiniBand private network to communicate between sites A and B. This enables the clustering software to detect network issues between sites, and trigger failover accordingly. The value provided should be the subnetwork IP address of the private network.

```
root@hostA1> db2cluster -cfs -set -option subnets
-value 10.1.1.0
```

In this example, 10.1.1.0 includes all the IP addresses from 10.1.1.0 through 10.1.1.255.

STEP 3: Prepare the `sqllib_shared` file system for replication.

To enable replication, we need to change the failure group of the non-replicated GPFS file system to #1. This would typically be the failure group on site A.

To permit that operation, the DB2 instance is stopped for each host, so the file system can be unmounted.

```
db2inst1@hostA1> db2stop instance on hostA1
db2inst1@hostA2> db2stop instance on hostA2
db2inst1@hostA3> db2stop instance on hostA3
db2inst1@hostB1> db2stop instance on hostB1
db2inst1@hostB2> db2stop instance on hostB2
db2inst1@hostB3> db2stop instance on hostB3
```

To ensure the sqllib_shared file system is cleanly unmounted, the cluster is put in maintenance mode.

```
root@hostA1:/> db2cluster -cm -enter -maintenance -all
```

Changing the failure group of the disk requires us to find out the Network Shared Disk (NSD) name that GPFS assigned to the disk. In the following sample output, the column **`Device`** contains the actual device path and the column **`Disk name`** contains the NSD name that GPFS assigned to that device.

```
root@hostA1:/> mmlsnsd -m

Disk name  NSD volume ID     Device         Node name
gpfs1nsd   091A33584D65F2F6  /dev/hdiskA1   hostA1
```

Create a file **`/tmp/nsdAddFGroup.txt`** containing a line describing the disk, and which indicates it is part of failure group 1.

```
root@hostA1:/> cat /tmp/nsdAddFGroup.txt
gpfs1nsd:::dataAndMetadata:1
```

The file should list all the NSD disks that belong to site A and that will belong to the **`db2fs1`** file system. In this example, there is just one disk.

```
root@hostA1:/> mmchdisk db2fs1 change
-F /tmp/nsdAddFGroup.txt
```

Modify the file system configuration to become a replicated file system.

```
root@hostA1:/> mmchfs db2fs1 -m 2 -r 2
```

STEP 4: Create an affinity between the NSD that you created for the sqllib_shared file system and the hosts on the site where the LUN is located.

Although some physical storage is local to each site, GPFS does not know which LUN is locally accessible (over the SAN) at each site. However we can indicate to GPFS that it should prefer going to local LUNs for read operations.

Create a file **/tmp/affinitizensd.txt** to contain a line that indicates the disk is part of site A, and then use **mmchnsd** to create the affinity between the NSD and a site.

```
root@hostA1:/> cat /tmp/affinitizensd.txt
gpfs1nsd:hostA1,hostA2,hostA3

root@hostA1:/> mmchnsd -F /tmp/affinitizensd.txt
```

Restarting the instance completes the change.

```
root@hostA1:/> db2cluster -cm -exit -maintenance -all

root@hostA1:/> db2cluster -cfs -mount -filesystem db2fs1

db2inst1@hostA1> db2start instance on hostA1
db2inst1@hostA2> db2start instance on hostA2
db2inst1@hostA3> db2start instance on hostA3
db2inst1@hostB1> db2start instance on hostB1
db2inst1@hostB2> db2start instance on hostB2
db2inst1@hostB3> db2start instance on hostB3
```

STEP 5: Replication of sqllib_shared file system.

Now add the replica disk and file system quorum disk to the existing **sqllib_shared** file system. Note that we also add information about the affinity of the LUNs to their local hosts.

Create a file **/tmp/nsdfailuregroup2.txt** that describes the replica disk(s) at site B and the tiebreaker disk on host **T.** In the following example **hdiskB1** on Site B will hold the data replica for the **sqllib_shared** file system, while the **hdiskC1** on host **T** will act as a file system disk quorum.

```
root@hostA1:/> cat /tmp/nsdfailuregroup2.txt
/dev/hdiskB1:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskC1:T::descOnly:3
```

Next, use this file to create the NSDs and add replica disks to the file system.

```
root@hostA1> /usr/lpp/mmfs/bin/mmcrnsd
-F /tmp/nsdfailuregroup2.txt

root@hostA1> /usr/lpp/mmfs/bin/mmadddisk  db2fs1
-F /tmp/nsdfailuregroup2.txt
```

After it is set up for replication, the file system needs to be rebalanced to actually replicate the data to the newly added disk.

```
root@hostA1> /usr/lpp/mmfs/bin/mmrestripefs db2fs1 -R
```

At this point we have:

- A GPFS and RSCT cluster across sites A, B and C
- A tie-breaker host **T** that is part of the RSCT domain and GPFS cluster but is not part of the DB2 instance.
- A DB2 pureScale cluster spanning sites A and B, with the instance shared metadata **sqllib_shared** file system being a replicated GPFS file system across sites A and B.

In the example above, the data in **sqllib_shared** is stored on both **/dev/hdiskA1** and **/dev/hdiskB1**. They are in separate replicated failure groups, so any data stored on **/dev/hdiskA1** is replicated on **/dev/hdiskB1**. The file descriptor quorum for **sqllib_shared** is handled through **/dev/hdiskC1**.

STEP 6: Creating file systems for the database.

To this point, we have storage replication configured for **sqllib_shared**, but we also need to configure it for the database and transaction logs. Next, we create NSDs using the disks for **logfs**, ensuring they are assigned to the correct failure groups.

Create a file **/tmp/nsdForLogfs.txt.**

```
root@hostA1:/> cat /tmp/nsdForLogfs.txt
/dev/hdiskA2:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskB2:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskC2:T::descOnly:3
```

Create the (replicated) **logfs** file system.

```
root@hostA1> /usr/lpp/mmfs/bin/mmcrnsd
-F /tmp/nsdForLogfs.txt

root@hostA1> /usr/lpp/mmfs/bin/mmcrfs logfs -F /tmp/nsdForLogfs.txt -m
2 -M 2 -r 2 -R 2 -B 1M -n 255
-T /gpfslog1RR
```

We also create NSDs with the disks for **datafs**, ensuring they are assigned to the correct failure groups.

Create a file **/tmp/nsdForDatafs.txt**

```
root@hostA1:/tmp> cat nsdForDatafs.txt

/dev/hdiskA3:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskA4:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskA5:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskA6:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskA7:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskB3:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskB4:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskB5:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskB6:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskB7:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskC3:T::descOnly:3
```

Create the replicated **datafs** file system.

```
root@hostA1> /usr/lpp/mmfs/bin/mmcrnsd
-F /tmp/nsdForDatafs.txt

root@hostA1> /usr/lpp/mmfs/bin/mmcrfs datafs
-F /tmp/nsdForDatafs.txt -m 2 -M 2 -r 2 -R 2 -B 1M
-n 255 -T /gpfstbsRR
```

Mount **datafs** and **logfs** and make the file systems writeable.

```
root@hostA1> db2cluster -cfs -mount -filesystem logfs

root@hostA2> db2cluster -cfs -mount -filesystem datafs

root@hostA1:/> chmod 777 /gpfslog1RR
root@hostA1:/> chmod 777 /gpfstbsRR
```

Now that all the replicated storage is in place, create the database.

```
db2inst1@hostA1> db2 create database MYDB ON /gpfstbsRR        DBPATH
ON /gpfslog1RR
```

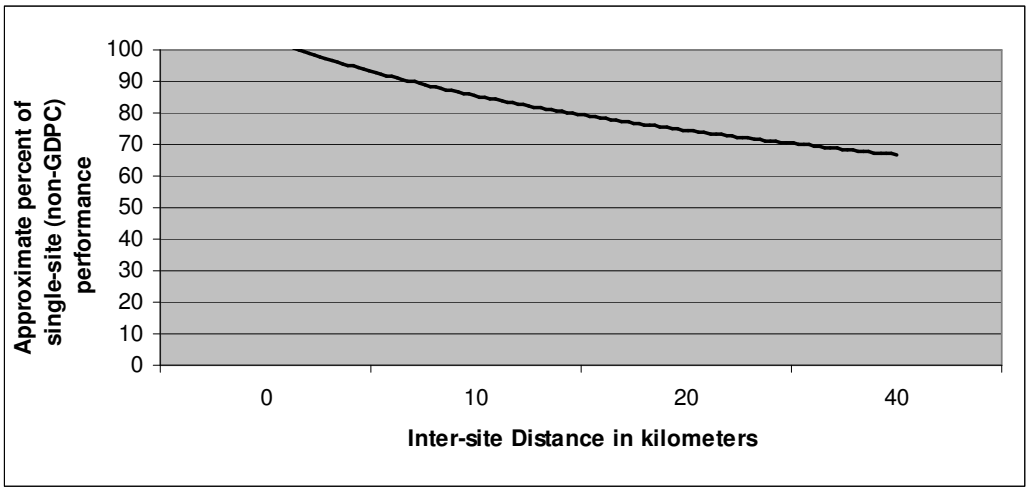
Performance Factors

As previously discussed, the introduction of significant distances between cluster members at different sites increases DB2 pureScale message latency in the amount of about 5 microseconds per kilometer of glass fiber. In some cases, the amount can be higher, if the connection includes signal repeaters, or is shared with other applications. Distance also adds latency to disk I/O requests over the zoned SAN. However, since the base duration of disk operations (around 1000 microseconds or more) tends to be much higher than that of DB2 pureScale messages, the relative impact to disk operations is generally less of a factor.

Besides distance, the performance overhead experienced by a GDPC configuration also depends on the workloads in use. The greater the portion of write activity (INSERT, UPDATE, DELETE) in the workload, the more messages need to be sent from members to the CFs, and the more disk writes (especially to the transaction logs) need to be made. This typically leads to higher perceived overhead at a given distance. Conversely, a greater portion of read (SELECT) activity means fewer messages and fewer disk writes, and reduced overhead.

Because all workloads are different, it is difficult to predict exactly what overhead any given GDPC might experience, relative to a single-site DB2 pureScale cluster. However, figure 4 provides an example of the impact to system throughput that increased cluster span might have, based on empirical tests in the IBM development lab.

Figure 4. Approximate impact to cluster throughput running e-commerce workload with increased distance



DB2 pureScale software is designed to have minimal downtime if a host fails due to hardware or software faults. In the event of failure, a system must be 'I/O fenced' to prevent it from corrupting the data. A key piece of technology that DB2 pureScale software uses to minimize downtime is SCSI-3 Persistent Reserve (PR). Once a host is I/O fenced, it can no longer access the storage device, and any I/O attempt is blocked. This technology enables recovery times on the order of 20 seconds from a member system failure.

In a GDPC, however, SCSI-3 PR cannot be enabled, since the tie-breaker site does not have access to the SAN storage. Instead, the GPFS disk lease expiry mechanism is used to fence a failed system. Depending on the nature of the failure, this typically results in a recovery time on the order of 60-90 seconds. This time is largely independent of the distance between sites.

Summary

While DB2 pureScale software in a regular single-site configuration provides outstanding availability, scalability and application transparency, events such as power or communication disruption, fires and floods can cause severe system outages, dramatically impacting users. By using technology to extend the reach of InfiniBand to many kilometers, and by employing well-known techniques to synchronously replicate GPFS file systems across sites, DB2 pureScale dispersed clusters are able to tolerate many common disaster scenarios that might otherwise compromise the entire system.

Appendix A – Detailed configuration steps

Prerequisites:

1. Sites A, B and C can communicate between each other through reliable TCP/IP links.
2. Other DB2 pureScale installation prerequisites have been satisfied across all hosts to be used in the cluster, such as passwordless ssh access for the root user.
3. Sites A and B are connected via a WAN or dark fiber with Obsidian Longbow IB range extenders, so that a single IB subnet can be configured across the sites.
4. Sites A and B each have a local SAN controller, and the SAN is zoned such that LUNs used for the DB2 pureScale instance are directly accessible from both sites. A one-to-one mapping between LUNs is required across sites so each LUN on site A has a corresponding equally sized LUN on site B.

To describe in detail each scenario we assume the following hardware configuration:

Site A: Hosts hostA1, hostA2, hostA3
Site B: Hosts hostB1, hostB2, hostB3
Site C: Host T

Equal sized LUNs have been provisioned on sites A and B, as follows

LUNs located on disk within Site A:

```
/dev/hdiskA1
/dev/hdiskA2
/dev/hdiskA3
/dev/hdiskA4
/dev/hdiskA5
/dev/hdiskA6
/dev/hdiskA7
```

LUNs located on disk within Site B:

```
/dev/hdiskB1
/dev/hdiskB2
/dev/hdiskB3
/dev/hdiskB4
/dev/hdiskB5
/dev/hdiskB6
/dev/hdiskB7
```

LUNs located on disk within Site C. These disks could be just 50MB logical volumes.

```
/dev/hdiskC1
/dev/hdiskC2
/dev/hdiskC3
```

Step 1: Install the DB2 pureScale Feature on Sites A and B.

Install the DB2 pureScale Feature on sites A and B using **db2setup**. Using the Advanced Configuration menu, designate **hostA3** and **hostB3** as the CFs and (optionally) one of the two to be the preferred primary CF.

Output from db2setup:

```
Product to install: DB2 Enterprise Server Edition with the
pureScale Feature

Previously Installed Components:
Components to be installed:
      Base client support
      Java support
      SQL procedures
      Base server support
      IBM Software Development Kit (SDK) for Java(TM)
      DB2 LDAP support
      DB2 Instance Setup wizard
      Control Server
      Communication support - TCP/IP
      Base application development tools
      Cluster caching facility
      Sample database source
      IBM Tivoli System Automation for Multiplatforms
      (Tivoli SA MP)
      IBM General Parallel File System (GPFS)

Languages:
      English
            All Products

Target directory: /opt/IBM/db2/V9.8

Maximum space required on each host: 3300 MB

DB2 Cluster Services:
      DB2 Cluster Services tiebreaker disk device path:
      /dev/hdiskC1
      DB2 clustered file system device path: /dev/hdiskA1

New instances:
      Instance name: db2inst1
            FCM port range: 60000-60003
            CF port: 56001
            CF Management port: 56000
            TCP/IP configuration:
                  Service name: db2c_db2inst1
                  Port number: 50002
            Instance user information:
                  User name: db2inst1
                  UID: 26131
```

```
                Group name: build
                GID: 200
                Home directory: /home/db2inst1
          Fenced user information:
                User name: db2inst1
                UID: 26131
                Group name: build
                GID: 200
                Home directory: /home/db2inst1

Cluster caching facilities:
      Preferred primary cluster caching facility:
      hostB3
      Preferred secondary cluster caching facility:
      hostA3

DB2 members:
      hostA1
      hostA2
      hostB1
      hostB2

New Host List:
      Host        Cluster Interconnect Netname
      hostA1      hostA1-ib0.torolab.ibm.com
      hostA2      hostA2-ib0.torolab.ibm.com
      hostA3      hostA3-ib0.torolab.ibm.com
      hostB1      hostB1-ib0.torolab.ibm.com
      hostB2      hostB2-ib0.torolab.ibm.com
      hostB3      hostB3-ib0.torolab.ibm.com
```

Required post-install steps following `db2setup`:

In order to start using the DB2 instance you need to logon using a valid user ID, such as the DB2 instance owner ID in this case, **`db2inst1`**.

Clients can connect to the DB2 pureScale instance **`db2inst1`** at any DB2 member using the host name and the port number, for example, 50002 in this case. Record this information for future reference.

Optional post-install steps following `db2setup`:

The network monitor configuration file for the DB2 pureScale instance, **`/var/ct/cfg/netmon.cf`**, was updated on each host to include the IP address of the hosts' gateway. In this case, the configuration file was modified on the following hosts: **`hostA2 hostA3 hostB1 hostB2 hostB3`**. This file must be modified in the event of future changes to the hosts' gateway IP address.

You should ensure that you have the correct license entitlements for DB2 products and features installed on this computer. Each DB2 product or feature comes with a license certificate file (also referred to as a license key) that is distributed on an Activation CD, which also includes instructions for applying the license file. If you purchased a base DB2 product, as well as separately priced features, you might need to install more than one license certificate. The Activation CD for your product or feature can be downloaded from Passport Advantage if it is not part of the physical media pack you received from IBM. For more information about licensing, search the Information Center (http://publib.boulder.ibm.com/infocenter/db2luw/v9r8) using terms such as "licensing" or "db2licm".

Refer to "What's New" http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/topic/com.ibm.db2.luw.wn.doc/doc/c0052035.html in the DB2 Information Center to learn about the new functions for DB2 V9.8.

Next, you should confirm that the tiebreaker is set to majority node set.

```
root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cm -list
-tiebreaker
```

In this case, the current quorum device is of type disk with the following specifics: PVID=00f60428281f7a31.

```
root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cm -set -tiebreaker -
majority
Configuring quorum device for domain 'db2domain_20110224005525' ...
Configuring quorum device for domain 'db2domain_20110224005525' was
successful.
```

Verify that the tiebreaker has been changed to majority node set:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cm -list
-tiebreaker
```

The current quorum device is of type Majority Node Set.

Step 2 – Disable SCSI-3 PR:

SCSI-3 PR needs to be disabled in GDPC. In the output below, **pr=yes** means that SCSI-3 PR is enabled.

```
root@hostA1:/opt/IBM/db2/V9.8/bin> mmlsnsd -X

Disk name NSD volume ID Device Devtype Node name Remarks
---------------------------------------------------------
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostA1 pr=yes
```

In this case, it is enabled, so stop DB2 and GPFS, and disable SCSI-3 PR:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> su - db2inst1

db2inst1@hostA1:/home/db2inst1> db2stop force

02/24/2011 01:24:16 0 0 SQL1064N DB2STOP processing was successful.
02/24/2011 01:24:19 1 0 SQL1064N DB2STOP processing was successful.
02/24/2011 01:24:21 3 0 SQL1064N DB2STOP processing was successful.
02/24/2011 01:24:22 2 0 SQL1064N DB2STOP processing was successful.
SQL1064N DB2STOP processing was successful.

db2inst1@hostA1:/home/db2inst1> exit

root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cfs -stop -all
All specified hosts have been stopped successfully.
```

Verify that GPFS is stopped on all hosts:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> mmgetstate -a

Node number Node name GPFS state
------------------------------------------
1           hostA1      down
2           hostA2      down
3           hostA3      down
4           hostB1      down
5           hostB2      down
6           hostB3      down
```

GPFS is down, so disable SCSI-3 PR:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> /usr/lpp/mmfs/bin/mmchconfig
usePersistentReserve=no

Verifying GPFS is stopped on all nodes ...
mmchconfig: Processing the disks on node hostA1.torolab.ibm.com
mmchconfig: Processing the disks on node hostA2.torolab.ibm.com
mmchconfig: Processing the disks on node hostA3.torolab.ibm.com
mmchconfig: Processing the disks on node hostB1.torolab.ibm.com
mmchconfig: Processing the disks on node hostB2.torolab.ibm.com
mmchconfig: Processing the disks on node hostB3.torolab.ibm.com
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all affected
nodes. This is an asynchronous process.
```

Verify that SCSI-3 PR has been disabled: (PR=yes is not displayed)

```
root@hostA1:/opt/IBM/db2/V9.8/bin> mmlsnsd -X
```

```
Disk name NSD volume ID Device Devtype Node name Remarks
---------------------------------------------------------
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostA1
```

Verify that **usePersistentReserve** has been set to no:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> mmlsconfig

Configuration data for cluster
db2cluster_20110224005554.torolab.ibm.com:
------------------------------------------------------------
clusterName db2cluster_20110224005554.torolab.ibm.com
clusterId 655893150084494058
autoload yes
minReleaseLevel 3.3.0.2
dmapiFileHandleSize 32
maxFilesToCache 10000
pagepool 256M
verifyGpfsReady yes
assertOnStructureError yes
worker1Threads 150
sharedMemLimit 2047M
usePersistentReserve no
failureDetectionTime 35
leaseRecoveryWait 35
tiebreakerDisks gpfs1nsd
[hostA1]
psspVsd no
adminMode allToAll

File systems in cluster db2cluster_20110224005554.torolab.ibm.com:
------------------------------------------------------------------
/dev/db2fs1
```

Step 3: Increase HostFailureDetectionTime to a higher value:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cm -list
-hostfailuredetectiontime
```

The host failure detection time is 4 seconds. Change it to 16 seconds and verify.

```
root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cm -set -option
hostfailuredetectiontime -value 16

root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cm -list
-hostfailuredetectiontime
```

Step 4: Add tiebreaker host into cluster.

Install DB2 software on the tiebreaker host:

```
root@T:/devinst/db2_v98fp3/aix64/s101210/ese_dsf> ./db2_install
WARNING:
package uDAPL(udapl.rte) not found.
Required minimum level is 6.1.0.2.

Default directory for installation of products - /opt/IBM/db2/V9.8
***********************************************************
Do you want to choose a different directory to install [yes/no] ?
no


Specify one of the following keywords to install DB2 products.
ESE_DSF

Enter "help" to redisplay product names.
Enter "quit" to exit.
***********************************************************
ESE_DSF

WARNING:
package uDAPL(udapl.rte) not found.
Required minimum level is 6.1.0.2.
DB2 installation is being initialized.

Total number of tasks to be performed: 44
Total estimated time for all tasks to be performed: 2850 second(s)

Task #1 start
Description: Checking license agreement acceptance
Estimated time 1 second(s)
Task #1 end

Task #2 start
Description: Base Client Support for installation with root privileges
Estimated time 3 second(s)
Task #2 end

Task #3 start
Description: Product Messages - English
Estimated time 12 second(s)
Task #3 end

Task #4 start
Description: Base client support
Estimated time 219 second(s)
Task #4 end

Task #5 start
Description: Java Runtime Support
Estimated time 3 second(s)
Task #5 end
```

```
Task #6 start
Description: Java Help (HTML) - English
Estimated time 3 second(s)
Task #6 end

Task #7 start
Description: Base server support for installation with root privileges
Estimated time 7 second(s)
Task #7 end

Task #8 start
Description: Integrated Flash Copy Support
Estimated time 354 second(s)
Task #8 end

Task #9 start
Description: Global Secure ToolKit
Estimated time 38 second(s)
Task #9 end

Task #10 start
Description: Java support
Estimated time 11 second(s)
Task #10 end

Task #11 start
Description: SQL procedures
Estimated time 3 second(s)
Task #11 end

Task #12 start
Description: ICU Utilities
Estimated time 48 second(s)
Task #12 end

Task #13 start
Description: Java Common files
Estimated time 19 second(s)
Task #13 end

Task #14 start
Description: Base server support
Estimated time 425 second(s)
Task #14 end

Task #15 start
Description: IBM Software Development Kit (SDK) for Java(TM)
Estimated time 222 second(s)
Task #15 end

Task #16 start
Description: Connect support
Estimated time 3 second(s)
Task #16 end

Task #17 start
```

```
Description: Relational wrappers common
Estimated time 3 second(s)
Task #17 end

Task #18 start
Description: DB2 data source support
Estimated time 12 second(s)
Task #18 end

Task #19 start
Description: DB2 LDAP support
Estimated time 3 second(s)
Task #19 end

Task #20 start
Description: DB2 Instance Setup wizard
Estimated time 8 second(s)
Task #20 end

Task #21 start
Description: Control Server
Estimated time 3 second(s)
Task #21 end

Task #22 start
Description: Spatial Extender client
Estimated time 3 second(s)
Task #22 end

Task #23 start
Description: Communication support - TCP/IP
Estimated time 3 second(s)
Task #23 end

Task #24 start
Description: Base application development tools
Estimated time 42 second(s)
Task #24 end

Task #25 start
Description: ese dsf common
Estimated time 7 second(s)
Task #25 end

Task #26 start
Description: DB2 Update Service
Estimated time 4 second(s)
Task #26 end

Task #27 start
Description: Parallel Extension
Estimated time 3 second(s)
Task #27 end

Task #28 start
Description: EnterpriseDB code
Estimated time 3 second(s)
```

```
Task #28 end

Task #29 start
Description: Replication tools
Estimated time 23 second(s)
Task #29 end

Task #30 start
Description: Cluster caching facility
Estimated time 15 second(s)
Task #30 end

Task #31 start
Description: Sample database source
Estimated time 4 second(s)
Task #31 end

Task #32 start
Description: Informix data source support
Estimated time 5 second(s)
Task #32 end

Task #33 start
Description: Product Signature for DB2 Enterprise Server Edition with
the pureScale Feature
Estimated time 3 second(s)
Task #33 end

Task #34 start
Description: IBM Tivoli System Automation for Multiplatforms (Tivoli SA
MP)
Estimated time 600 second(s)
Task #34 end

Task #35 start
Description: IBM General Parallel File System (GPFS)
Estimated time 600 second(s)
Task #35 end

Task #36 start
Description: Setting DB2 library path
Estimated time 180 second(s)
Task #36 end

Task #37 start
Description: Copying DB2 image
Estimated time 600 second(s)
Task #37 end

Task #38 start
Description: Installing or updating DB2 HA scripts for Tivoli SA MP
Estimated time 40 second(s)
Task #38 end

Task #39 start
Description: Installing or updating DB2 Cluster Scripts for GPFS
Estimated time 40 second(s)
```

```
Task #39 end

Task #40 start
Description: Executing control tasks
Estimated time 20 second(s)
Task #40 end

Task #41 start
Description: Updating global registry
Estimated time 20 second(s)
Task #41 end

Task #42 start
Description: Updating the db2ls link
Estimated time 1 second(s)
Task #42 end

Task #43 start
Description: Registering DB2 licenses
Estimated time 5 second(s)
Task #43 end

Task #44 start
Description: Setting default global profile registry variables
Estimated time 1 second(s)
Task #44 end

Task #45 start
Description: Initializing instance list
Estimated time 5 second(s)
Task #45 end

Task #46 start
Description: Updating global profile registry
Estimated time 3 second(s)
Task #46 end

The execution completed successfully.
```

For more information see the DB2 installation log at
/tmp/db2_install.log.*<nnnnnnnn>*

Change the GPFS quorum type for the cluster to majority node set and verify:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cfs -set -
tiebreaker -majority

root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cfs -list
-tiebreaker
```

Add the tiebreaker site to the RSCT cluster:

```
root@T> preprpnode hostA1 hostA2 hostB1 hostB2 hostA3 hostB3

root@hostA1:/opt/IBM/db2/V9.8/bin> lsrpnode
Name OpState RSCTVersion
hostB2 Online 3.1.0.3
hostB3 Online 3.1.0.3
hostA3 Online 3.1.0.3
hostB1 Online 3.1.0.3
hostA2 Online 3.1.0.3
hostA1 Online 3.1.0.3

root@hostA1:/opt/IBM/db2/V9.8/bin> db2cluster -cm -add -host T
Adding node 'T' to the cluster ...
Trace spooling could not be enabled on the local host.
Adding node 'T' to the cluster was successful.
```

Verify that the tiebreaker has been added to the RSCT cluster:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> lsrpnode
Name OpState RSCTVersion
T Online 3.1.0.3
hostB3 Online 3.1.0.3
hostB2 Online 3.1.0.3
hostB1 Online 3.1.0.3
hostA3 Online 3.1.0.3
hostA2 Online 3.1.0.3
hostA1 Online 3.1.0.3
```

Add the tiebreaker site to the GPFS cluster:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> mmlsnode

GPFS nodeset Node list
------------- -------------------------------------------------
db2cluster_20110224005554 hostA1 hostA2 hostA3 hostB1 hostB2 hostB3

root@hostA1:/opt/IBM/db2/V9.8/bin> /usr/lpp/mmfs/bin/mmaddnode
T:quorum-client

Thu Feb 24 01:49:38 EST 2011: mmaddnode: Processing node
T.torolab.ibm.com
mmaddnode: Command successfully completed
mmaddnode: Warning: Not all nodes have proper GPFS license
designations.
```

Use the **mmchlicense** command to designate licenses as needed.

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

Verify that the tiebreaker site has been added to the GPFS cluster:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> mmlsnode
```

```
============================================================
| Warning: |
| This cluster contains nodes that do not have a proper GPFS license |
| designation. This violates the terms of the GPFS licensing agreement.
|
| Use the mmchlicense command and assign the appropriate GPFS licenses
|
| to each of the nodes in the cluster. For more information about GPFS
|
| license designation, see the Concepts, Planning, and Installation
Guide. |
============================================================

GPFS nodeset Node list
------------- -------------------------------------------
db2cluster_20110224005554 hostA1 hostA2 hostA3 hostB1 hostB2 hostB3 T
```

On the tiebreaker site add the GPFS license:

```
root@T:/opt/IBM/db2/V9.8/bin> db2cluster -cfs -add -license

The license for the shared file system cluster has been successfully
added.
```

Verify the license warning message is gone:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> mmlsnode

GPFS nodeset Node list
------------- -------------------------------------------------
db2cluster_20110224005554 hostA1 hostA2 hostA3 hostB1 hostB2 hostB3 T
```

Ensure we don't get false errors due to the fact that the tiebreaker site cannot directly access some of the disks:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> /usr/lpp/mmfs/bin/mmchconfig

unmountOnDiskFail=yes -N T
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

root@hostA1:/opt/IBM/db2/V9.8/bin> mmlsconfig
Configuration data for cluster
db2cluster_20110224005554.torolab.ibm.com:
----------------------------------------------------------
clusterName db2cluster_20110224005554.torolab.ibm.com
clusterId 655893150084494058
autoload yes
minReleaseLevel 3.3.0.2
dmapiFileHandleSize 32
maxFilesToCache 10000
pagepool 256M
verifyGpfsReady yes
assertOnStructureError yes
worker1Threads 150
```

```
sharedMemLimit 2047M
usePersistentReserve no
failureDetectionTime 35
leaseRecoveryWait 35
[T]
unmountOnDiskFail yes
[common]
[hostA1]
psspVsd no
adminMode allToAll

File systems in cluster db2cluster_20110224005554.torolab.ibm.com:
------------------------------------------------------------------
/dev/db2fs1


root@hostA1:/opt/IBM/db2/V9.8/bin> mmgetstate -a

Node number Node name GPFS state
-----------------------------------------
1 hostA1 down
2 hostA2 down
3 hostA3 down
4 hostB1 down
5 hostB2 down
6 hostB3 down
7 T down
```

Add the IB subnet to the GPFS list of networks. First, check the subnet for the IB network:

```
root@hostA1:/opt/IBM/db2/V9.8/bin> ping hostA1-ib0
PING hostA1-ib0.torolab.ibm.com (10.1.1.1): 56 data bytes
64 bytes from 10.1.1.1: icmp_seq=0 ttl=255 time=0 ms

root@hostA1:/opt/IBM/db2/V9.8/bin> mmchconfig subnets=10.1.1.0
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

root@hostA1:/opt/IBM/db2/V9.8/bin> mmlsconfig
Configuration data for cluster
db2cluster_20110224005554.torolab.ibm.com:
----------------------------------------------------------
clusterName db2cluster_20110224005554.torolab.ibm.com
clusterId 655893150084494058
autoload yes
minReleaseLevel 3.3.0.2
dmapiFileHandleSize 32
maxFilesToCache 10000
pagepool 256M
verifyGpfsReady yes
assertOnStructureError yes
worker1Threads 150
sharedMemLimit 2047M
usePersistentReserve no
failureDetectionTime 35
leaseRecoveryWait 35
```

```
[T]
unmountOnDiskFail yes
[common]
subnets 10.1.1.0
[hostA1]
psspVsd no
adminMode allToAll

File systems in cluster db2cluster_20110224005554.torolab.ibm.com:
------------------------------------------------------------------
/dev/db2fs1
```

You need to ensure that each site contains a shared file system configuration server so that the GPFS configuration files will be preserved in case of a disaster on one site.
Change the configuration servers so that **hostA1** is the primary config server and **hostB1** the secondary config server:

```
root@hostA1> /usr/lpp/mmfs/bin/mmchcluster -p hostA1 -s hostB1

root@hostA1:/> mmlscluster

GPFS cluster information
========================
GPFS cluster name: db2cluster_20110224005554.torolab.ibm.com
GPFS cluster ID: 655893150084494058
GPFS UID domain: db2cluster_20110224005554.torolab.ibm.com
Remote shell command: /usr/bin/ssh
Remote file copy command: /usr/bin/scp

GPFS cluster configuration servers:
-----------------------------------
Primary server: hostA1.torolab.ibm.com
Secondary server: hostB1.torolab.ibm.com
```

Step 5: Prepare the sqllib_shared file system for replication:

Assign the existing disk to failure group 1:

```
root@hostA1:/> mmlsnsd -m

Disk name NSD volume ID Device Node name Remarks
-------------------------------------------------------------------
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hostA1.torolab.ibm.com


root@hostA1:/> cat /tmp/nsdAddFGroup.txt
gpfs1nsd:::dataAndMetadata:1

root@hostA1:/> db2cluster -cfs -list -filesystem
File system NAME MOUNT_POINT
-------------------------------- ------------------------
db2fs1 /db2sd_20110224005651
```

```
root@hostA1:/> mmlsdisk db2fs1 -L
disk driver sector failure holds holds storage
name type size group metadata data status availability disk ID pool
remarks
------------ -------- ------ ------- -------- ----- -------------
gpfs1nsd nsd 512 -1 yes yes ready up 1 system desc
Number of quorum disks: 1
Read quorum value: 1
Write quorum value: 1

root@hostA1:/> mmchdisk db2fs1 change -F /tmp/nsdAddFGroup.txt
Verifying file system configuration information ...
mmchdisk: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

root@hostA1:/> mmlsdisk db2fs1 -L
disk driver sector failure holds holds storage
name type size group metadata data status availability disk ID pool
remarks
------------ -------- ------ ------- -------- ----- -------------
gpfs1nsd nsd 512 1 yes yes ready up 1 system desc
Number of quorum disks: 1
Read quorum value: 1
Write quorum value: 1
Attention: Due to an earlier configuration change the file system
is no longer properly replicated.
```

Note that the disk gpfs1nsd is now assigned to failure group 1 (previously, it was -1)

Change the replication settings for the file system to enable replication:

```
root@hostA1:/> mmlsfs db2fs1
flag value description
---- --------------- -----------------------------------------
-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 2 Maximum number of metadata replicas
-r 1 Default number of data replicas
-R 2 Maximum number of data replicas

root@hostA1:/> mmchfs db2fs1 -m 2 -r 2
The desired replication factor exceeds the number of available metadata
failure groups.
Allowed, but files will be unreplicated and hence at risk.
Attention: The desired replication factor exceeds the number of
available data failure groups in storage pool system.
This is allowed, but files in this storage pool will not be replicated
and will therefore be at risk.
```

Verify that the file system settings have been changed to enable replication:

```
root@hostA1:/> mmlsfs db2fs1
flag value description
---- --------------- -----------------------------
```

```
-f 32768 Minimum fragment size in bytes
-i 512 Inode size in bytes
-I 32768 Indirect block size in bytes
-m 2 Default number of metadata replicas
-M 2 Maximum number of metadata replicas
-r 2 Default number of data replicas
-R 2 Maximum number of data replicas
```

Step 6 - Create an affinity between the NSD that you created for the `sqllib_shared` file system and the hosts on the site where the LUN is located.

GPFS will not know which sites a LUN is locally accessible (over a SAN). By informing GPFS about this, GPFS will prefer going to the local LUNs for reads, providing better performance.

```
root@hostA1:/> cat /tmp/affinitizensd.txt
gpfs1nsd:hostA1,hostA2,hostA3
```

Unmount the file system on all hosts in order to make the changes:

```
root@hostA1:/> mmgetstate -a

Node number Node name GPFS state
------------------------------------------
1 hostA1 active
2 hostA2 active
3 hostA3 active
4 hostB1 active
5 hostB2 active
6 hostB3 active
7 T active
```

Ensure that RSCT does not automatically remount the file system:

```
db2inst1@hostA1:/home/db2inst1> db2stop instance on hostA1
SQL1064N DB2STOP processing was successful.
db2inst1@hostA2:/home/db2inst1> db2stop instance on hostA2
SQL1064N DB2STOP processing was successful.
db2inst1@hostA3:/home/db2inst1> db2stop instance on hostA3
SQL1064N DB2STOP processing was successful.
db2inst1@hostB1:/home/db2inst1> db2stop instance on hostB1
SQL1064N DB2STOP processing was successful.
db2inst1@hostB2:/home/db2inst1> db2stop instance on hostB2
SQL1064N DB2STOP processing was successful.
db2inst1@hostB3:/home/db2inst1> db2stop instance on hostB3
SQL1064N DB2STOP processing was successful.

root@hostA1:/> export PATH=$PATH:/home/db2inst1/sqllib/bin
root@hostA1:/> db2cluster -cm -enter -maintenance -all
Domain 'db2domain_20110224005525' has entered maintenance mode.
```

Verify the file system is not mounted. If it is, then unmount it:

```
root@hostA1:/> mmlsmount db2fs1
File system db2fs1 is not mounted.

root@hostA1:/> cat /tmp/affinitizensd.txt
gpfs1nsd:hostA1,hostA2,hostA3
root@hostA1:/> mmchnsd -F /tmp/affinitizensd.txt
mmchnsd: Processing disk gpfs1nsd
mmchnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

Verify that the site A computers (hostA2*) have become the server hosts for the disk:

```
root@hostA1:/> mmlsnsd -X

Disk name NSD volume ID Device Devtype Node name Remarks
---------------------------------------------------------------------------
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostA1.torolab.ibm.com
server node
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostA2.torolab.ibm.com
server node
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostA3.torolab.ibm.com
server node
```

Restart everything:

```
root@hostA1:/> db2cluster -cm -exit -maintenance -all

Host 'hostA1' has exited maintenance mode. Domain
'db2domain_20110224005525' has been started.
```

Verify that the file system has been remounted:

```
root@hostA1:/> mmlsmount db2fs1
```

File system db2fs1 is mounted on 6 hosts.

```
db2inst1@hostA1:/home/db2inst1> db2start instance on hostA1
SQL1063N DB2START processing was successful.
db2inst1@hostA2:/home/db2inst1> db2start instance on hostA2
SQL1063N DB2START processing was successful.
db2inst1@hostA3:/home/db2inst1> db2start instance on hostA3
SQL1063N DB2START processing was successful.
db2inst1@hostB1:/home/db2inst1> db2start instance on hostB1
SQL1063N DB2START processing was successful.
db2inst1@hostB2:/home/db2inst1> db2start instance on hostB2
SQL1063N DB2START processing was successful.
db2inst1@hostB3:/home/db2inst1> db2start instance on hostB3
SQL1063N DB2START processing was successful.
```

Verify with **`lssam`** that the host resources are now online for all 6 computers:

```
Online IBM.Equivalency:instancehost_db2inst1-equ
|- Online IBM.Application:instancehost_db2inst1_hostB3:hostB3
|- Online IBM.Application:instancehost_db2inst1_hostB2:hostB2
|- Online IBM.Application:instancehost_db2inst1_hostB1:hostB1
```

```
|- Online IBM.Application:instancehost_db2inst1_hostA2:hostA2
|- Online IBM.Application:instancehost_db2inst1_hostA1:hostA1
'- Online IBM.Application:instancehost_db2inst1_hostA3:hostA3
```

Step 7: Add the replica disk from site B and the file system quorum disk from the tiebreaker site.

Create the NSD for the replica disk from site B:

```
root@hostA1:/> cat /tmp/nsdfailuregroup2.txt
/dev/hdiskB1:hostB1,hostB2,hostB3::dataAndMetadata:2
root@hostA1:/> mmcrnsd -F /tmp/nsdfailuregroup2.txt
mmcrnsd: Processing disk hdiskB1
mmcrnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

Create the NSD for the disk from the tiebreaker site:

```
root@T:/tmp> cat nsdfailuregroup3.txt
/dev/hdiskC1:T::descOnly:3

root@T:/tmp> mmcrnsd -F /tmp/nsdfailuregroup3.txt
```

Verify that the NSDs have been created with the **mmlsnsd** command:

```
gpfs1001nsd 091A336D4D674B1E /dev/hdiskB1 hdisk hostA1.torolab.ibm.com
gpfs1001nsd 091A336D4D674B1E /dev/hdiskB1 hdisk hostA2.torolab.ibm.com
gpfs1001nsd 091A336D4D674B1E /dev/hdiskB1 hdisk hostA3.torolab.ibm.com
gpfs1001nsd 091A336D4D674B1E /dev/hdiskB1 hdisk hostB1.torolab.ibm.com
server node
gpfs1001nsd 091A336D4D674B1E /dev/hdiskB1 hdisk hostB2.torolab.ibm.com
server node
gpfs1001nsd 091A336D4D674B1E /dev/hdiskB1 hdisk hostB3.torolab.ibm.com
server node
gpfs1002nsd 091A33434D674B57 /dev/hdiskC1 hdisk
T.torolab.ibm.com server node
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostA1.torolab.ibm.com
server node
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostA2.torolab.ibm.com
server node
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostA3.torolab.ibm.com
server node
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostB1.torolab.ibm.com
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostB2.torolab.ibm.com
gpfs1nsd 091A33584D65F2F6 /dev/hdiskA1 hdisk hostB3.torolab.ibm.com
```

Add the disk at site B to a file system:

```
root@hostA1:/> db2cluster -cfs -add -filesystem db2fs1 -disk
/dev/hdiskB1
```

If the hdisk specified in the command is in use on any host in the cluster you might see the following error:

```
Disk '/dev/hdiskB1' is already in use.

There is a problem with the disks specified in the operation. Check the
diagnostic log (db2diag.log or /tmp/ibm.db2.cluster.*) for more
information. Correct the problem and re-issue the command.

A diagnostic log has been saved to '/tmp/ibm.db2.cluster.CJeoEa'.
```

In that case the disk can be added to the file system with the GPFS **mmaddisk** command:

```
root@hostA1:/> cat /tmp/nsdfailuregroup2.txt
# /dev/hdiskB1:hostB1,hostB2,hostB3::dataAndMetadata:2
gpfs1001nsd:::dataAndMetadata:2::

root@hostA1:/> mmadddisk db2fs1 -F /tmp/nsdfailuregroup2.txt

The following disks of db2fs1 will be formatted on node hostA1:
gpfs1001nsd: size 34603008 KB
Extending Allocation Map
Checking Allocation Map for storage pool 'system'
Completed adding disks to file system db2fs1.
mmadddisk: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

Verify that the disk has been added to the file system with the correct failure group:

```
root@hostA1:/> mmlsdisk db2fs1 -L

disk driver sector failure holds holds storage
name type size group metadata data status availability disk ID pool
remarks
------------ -------- ------ ------- -------- ----- -------------
gpfs1nsd nsd 512 1 yes yes ready up 1 system desc
gpfs1001nsd nsd 512 2 yes yes ready up 2 system desc
Number of quorum disks: 2
Read quorum value: 2
Write quorum value: 2
Attention: Due to an earlier configuration change the file system
is no longer properly replicated.
```

Add the disk at the tiebreaker site to the file system:

```
root@T:/> cat /tmp/nsdfailuregroup3.txt
# /dev/hdiskC1:T::descOnly:3
gpfs1002nsd:::descOnly:3::

root@T:/> mmadddisk db2fs1 -F /tmp/nsdfailuregroup3.txt

The following disks of db2fs1 will be formatted on node hostA1:
gpfs1002nsd: size 1048576 KB
Extending Allocation Map
Checking Allocation Map for storage pool 'system'
Completed adding disks to file system db2fs1.
```

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

Verify that the disk has been added to the file system and to the correct failure group:

```
root@T:/> mmlsdisk db2fs1 -L

disk driver sector failure holds holds storage
name type size group metadata data status availability disk ID pool
remarks
------------ -------- ------ ------- -------- ----- -------------
gpfs1nsd nsd 512 1 yes yes ready up 1 system desc
gpfs1001nsd nsd 512 2 yes yes ready up 2 system desc
gpfs1002nsd nsd 512 3 no no ready up 3 system desc
Number of quorum disks: 3
Read quorum value: 2
Write quorum value: 2
Attention: Due to an earlier configuration change the file system
is no longer properly replicated.
```

Step 8 - Rebalance the file system to replicate the data on the newly added disks.

```
root@hostA1:/> mmrestripefs db2fs1 -R

root@hostA1:/> mmlsdisk db2fs1 -L
disk driver sector failure holds holds storage
name type size group metadata data status availability disk ID pool
remarks
------------ -------- ------ ------- -------- ----- -------------
gpfs1nsd nsd 512 1 yes yes ready up 1 system desc
gpfs1001nsd nsd 512 2 yes yes ready up 2 system desc
gpfs1002nsd nsd 512 3 no no ready up 3 system desc
Number of quorum disks: 3
Read quorum value: 2
Write quorum value: 2
```

Verify that the message about the file system not being replicated is gone.

Step 9: Create NSDs for the disks to be used for the log file system.

```
cat /tmp/nsdForLogfs1.txt

/dev/hdiskA2:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskB2:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskC2:T::descOnly:3

root@hostA1:/> mmcrnsd -F /tmp/nsdForLogfs1.txt
mmcrnsd: Processing disk hdiskA2
mmcrnsd: Processing disk hdiskB2
```

```
mmcrnsd: Processing disk hdiskC2
mmcrnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

root@hostA1:/> cat /tmp/nsdForLogfs1.txt
# /dev/hdiskA2:hostA1,hostA2,hostA3::dataAndMetadata:1
gpfs1004nsd:::dataAndMetadata:1::
# /dev/hdiskB2:hostB1,hostB2,hostB3::dataAndMetadata:2
gpfs1005nsd:::dataAndMetadata:2::
# /dev/hdiskC2:T::descOnly:3
gpfs1006nsd:::descOnly:3::
```

Verify that the NSDs have been created:

```
root@hostA1:/> mmlsnsd -X:

gpfs1004nsd 091A33584D675EDA /dev/hdiskA2 hdisk hostA1.torolab.ibm.com
server node
gpfs1004nsd 091A33584D675EDA /dev/hdiskA2 hdisk hostA2.torolab.ibm.com
server node
gpfs1004nsd 091A33584D675EDA /dev/hdiskA2 hdisk hostA3.torolab.ibm.com
server node
gpfs1004nsd 091A33584D675EDA /dev/hdiskA2 hdisk hostB1.torolab.ibm.com
gpfs1004nsd 091A33584D675EDA /dev/hdiskA2 hdisk hostB2.torolab.ibm.com
gpfs1004nsd 091A33584D675EDA /dev/hdiskA2 hdisk hostB3.torolab.ibm.com
gpfs1005nsd 091A336D4D675EDC /dev/hdiskB2 hdisk hostA1.torolab.ibm.com
gpfs1005nsd 091A336D4D675EDC /dev/hdiskB2 hdisk hostA2.torolab.ibm.com
gpfs1005nsd 091A336D4D675EDC /dev/hdiskB2 hdisk hostA3.torolab.ibm.com
gpfs1005nsd 091A336D4D675EDC /dev/hdiskB2 hdisk hostB1.torolab.ibm.com
server node
gpfs1005nsd 091A336D4D675EDC /dev/hdiskB2 hdisk hostB2.torolab.ibm.com
server node
gpfs1005nsd 091A336D4D675EDC /dev/hdiskB2 hdisk hostB3.torolab.ibm.com
server node
gpfs1006nsd 091A33434D675EE0 /dev/hdiskC2 hdisk T.torolab.ibm.com
server node
```

Step 10: Create the replicated logfs system:

```
root@hostA1:/> cat /tmp/nsdForLogfs1.txt
# /dev/hdiskA2:hostA1,hostA2,hostA3::dataAndMetadata:1
gpfs1004nsd:::dataAndMetadata:1::
# /dev/hdiskB2:hostB1,hostB2,hostB3::dataAndMetadata:2
gpfs1005nsd:::dataAndMetadata:2::
# /dev/hdiskC2:T::descOnly:3
gpfs1006nsd:::descOnly:3::

root@hostA1:/> mmcrfs gpfslog1RR -F /tmp/nsdForLogfs1.txt -m 2 -M 2 -r
2 -R 2 -B 1M -n 255 -T /gpfslog1RR

The following disks of gpfslog1RR will be formatted on node hostB2:
gpfs1004nsd: size 438304768 KB
gpfs1005nsd: size 34603008 KB
gpfs1006nsd: size 57344 KB
```

```
Formatting file system ...
Disks up to size 6.7 TB can be added to storage pool 'system'.
Creating Inode File
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
Formatting Allocation Map for storage pool 'system'
Completed creation of file system /dev/gpfslog1RR.
mmcrfs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

Verify that the file system has been created with the disks in the proper failure groups:

```
root@hostA1:/> mmlsdisk gpfslog1RR -L

disk driver sector failure holds holds storage
name type size group metadata data status availability disk ID pool
remarks
------------ -------- ------ ------- -------- ----- -------------
gpfs1004nsd nsd 512 1 yes yes ready up 1 system desc
gpfs1005nsd nsd 512 2 yes yes ready up 2 system desc
gpfs1006nsd nsd 512 3 no no ready up 3 system desc
Number of quorum disks: 3
Read quorum value: 2
Write quorum value: 2
```

Step 11: Create NSDs for datafs, create the datafs file system:

```
root@hostA1:/tmp> cat nsdForDatafs.txt
/dev/hdiskA3:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskA4:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskA5:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskA6:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskA7:hostA1,hostA2,hostA3::dataAndMetadata:1
/dev/hdiskB3:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskB4:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskB5:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskB6:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskB7:hostB1,hostB2,hostB3::dataAndMetadata:2
/dev/hdiskC3:T::descOnly:3

root@hostA1:/tmp> mmcrnsd -F /tmp/nsdForDatafs.txt
mmcrnsd: Processing disk hdiskA3
mmcrnsd: Processing disk hdiskA4
mmcrnsd: Processing disk hdiskA5
mmcrnsd: Processing disk hdiskA6
mmcrnsd: Processing disk hdiskA7
mmcrnsd: Processing disk hdiskB3
mmcrnsd: Processing disk hdiskB4
mmcrnsd: Processing disk hdiskB5
mmcrnsd: Processing disk hdiskB6
mmcrnsd: Processing disk hdiskB7
mmcrnsd: Processing disk hdiskC3
mmcrnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

```
root@hostA1:/tmp> mmcrfs gpfstbsRR -F /tmp/nsdForDatafs.txt -m 2
-M 2 -r 2 -R 2 -B 1M -n 255 -T /gpfstbsRR

The following disks of gpfstbsRR will be formatted on node hostA3:
gpfs1016nsd: size 438304768 KB
gpfs1017nsd: size 438304768 KB
gpfs1018nsd: size 438304768 KB
gpfs1019nsd: size 1462220800 KB
gpfs1020nsd: size 1462220800 KB
gpfs1021nsd: size 157286400 KB
gpfs1022nsd: size 157286400 KB
gpfs1023nsd: size 157286400 KB
gpfs1024nsd: size 157286400 KB
gpfs1025nsd: size 157286400 KB
gpfs1026nsd: size 57344 KB
Formatting file system ...
Disks up to size 18 TB can be added to storage pool 'system'.
Creating Inode File
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
Formatting Allocation Map for storage pool 'system'
Completed creation of file system /dev/gpfstbsRR.
mmcrfs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

Step 12: Mount log file systems and data file system:

```
root@hostA1:/tmp> mmlsmount gpfslog1RR
File system gpfslog1RR is not mounted.

root@hostA1:/tmp> db2cluster -cfs -mount -filesystem gpfslog1RR
File system 'gpfslog1RR' was successfully mounted.

root@hostA1:/tmp> mmlsmount gpfslog1RR
File system gpfslog1RR is mounted on 7 nodes.

root@hostA1:/tmp> db2cluster -cfs -mount -filesystem gpfstbsRR
File system 'gpfstbsRR' was successfully mounted.
```

Step 13: Create the database.

As root, change the permissions on the file systems to allow the instance user to write to them:

```
root@hostA1:/> chmod 777 /gpfslog1RR

root@hostA1:/> ls -ld gpfslog1RR
drwxrwxrwx 2 root system 32768 Feb 25 02:59 gpfslog1RR

root@hostA1:/> chmod 777 /gpfstbsRR

db2inst1@hostA1:/home/db2inst1> db2start
02/25/2011 04:00:44 0 0 SQL1063N DB2START processing was successful.
02/25/2011 04:00:46 3 0 SQL1063N DB2START processing was successful.
```

```
02/25/2011 04:00:47 2 0 SQL1063N DB2START processing was successful.
02/25/2011 04:00:47 1 0 SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.

db2inst1@hostA1:/home/db2inst1/sqllib/db2dump> db2 create database MYDB
on /gpfstbsRR dbpath on /gpfslog1RR

DB20000I The CREATE DATABASE command completed successfully.
```

Step 14: Update storage failure timeouts.

Ensure that in the case of storage controller or site failure, an error is returned quickly to GPFS by setting the relevant device driver parameters. Note that the relevant parameters will differ for different device drivers. For DS3K/DS4K using the default AIX PCM, the updates should be as follows:

```
chdev -l hdisk<X> -a 'cntl_delay_time=20 cntl_hcheck_int=2' -P

<repeat for every hdisk<X> used by pureScale>

chdev -l fscsi<Y> -a dyntrk=yes -a fc_err_recov=fast_fail -P

<repeat for every fscsi<Y> adapter>

reboot the host

<repeat chdevs for every host in the cluster>
```

Verify the attributes have been set correctly on every computer:

```
root> lsattr -El fscsi0
attach          switch     How this adapter is CONNECTED           False
dyntrk          yes        Dynamic Tracking of FC Devices          True
fc_err_recov    fast_fail  FC Fabric Event Error RECOVERY Policy   True

root> lsattr -El hdiskA1
PCM             PCM/friend/otherapdisk  Path Control Module              False
PR_key_value    none                    Persistent Reserve Key Value     True
Algorithm       fail_over               Algorithm                        True
autorecovery    no                      Path/Ownership Autorecovery      True
clr_q           no                      Device CLEARS its Queue on error True
cntl_delay_time 20                      Controller Delay Time            True
cntl_hcheck_int 2                       Controller Health Check Interval True
```

Appendix B - Reintegrating a Failed Site

This scenario describes how to reintegrate a site back into the cluster after a temporary failure, such as a power outage or network outage. When the outage is resolved the cluster membership is reinstated on the hosts of the failed site.

For this scenario, we will assume that site B is being reintegrated.

1. Ensure that all the hosts have the latest GPFS configuration

   ```
   root@hostA1> /usr/lpp/mmfs/bin/mmchcluster -p LATEST
   ```

2. Ensure that both sites contain a GPFS server.

   ```
   root@hostA1> /usr/lpp/mmfs/bin/mmchcluster -p hostA1 -s hostB1
   ```

   Note that hostA1 and hostB1 are at different sites. Verify with **mmlscluster** that the primary and secondary configuration servers have been properly set.

3. For each host at the reintegrated site, ensure that it is designated as a GPFS quorum node using the **mmlscluster** command.

   ```
   Node  Daemon node name  IP address  Admin node name Designation
   ------------------------------------------------------------------
      1  host1A.ibm.com    9.26.51.88  host1A.ibm.com  quorum-manager
   ```

4. If a host on site A or B is not designated as a quorum node, designate it as such:

   ```
   root@hostA1> /usr/lpp/mmfs/bin/mmchnode --quorum -N hostB1
   ```

5. For each host on the reintegrated site ensure that it is designated as an RSCT quorum node using the **lsrsrc** command:

```
root@coralperf15a:/> lsrsrc -s "Name == 'host1A'" IBM.PeerNode
IsQuorumNode

Resource Persistent Attributes for IBM.PeerNode
resource 1:
        IsQuorumNode = 1
```

6. If a particular host is not set as an RSCT quorum node, then designate it as such:

```
root@hostA1> export CT_MANAGEMENT_SCOPE=2

root@hostA1> chrsrc -s "Name == 'hostB1'" IBM.PeerNode IsQuorumNode=1
```

7. After the hosts are restarted, ensure that all cluster services on all the hosts are online

```
root@hostA1> db2cluster -cm -start -domain <domain_name>
root@hostA1> db2cluster -cfs -start -all
```

8. Verify that the CM and CFS are online on all hosts

```
root@hostA1> db2cluster -cm -list -host -state
root@hostA1> db2cluster -cfs -list -host -state
```

9. Re-enable file system replication by starting the failed disks and restriping the file system

   Ensure that for each file system all of the disks are started:

```
root@hostA1> /usr/lpp/mmfs/bin/mmchdisk db2fs1 start -a
root@hostA1> /usr/lpp/mmfs/bin/mmchdisk logfs start -a
root@hostA1> /usr/lpp/mmfs/bin/mmchdisk datafs start -a
```

   Validate that all the disks for all file systems are ready and available:

```
root@hostA1> /usr/lpp/mmfs/bin/mmlsdisk db2fs1
root@hostA1> /usr/lpp/mmfs/bin/mmlsdisk logfs
root@hostA1> /usr/lpp/mmfs/bin/mmlsdisk datafs
```

10. For each file system the expectation is that all of the NSD disks show a status of ready and availability as up.

```
disk driver sector failure holds    holds storage
name type   size   group   metadata data  status  avail. pool
---- ------ ------ ------- -------- ----- ------- ------ ----
nsd1 nsd    512    1       yes      yes   ready   up     system
```

    Ensure the replication is correctly reinitialized and the data and metadata is correctly replicated on both site.

```
root@hostA1>/usr/lpp/mmfs/bin/mmrestripefs instfs -R
root@hostA1>/usr/lpp/mmfs/bin/mmrestripefs logfs -R
root@hostA1>/usr/lpp/mmfs/bin/mmrestripefs datafs -R
```

11. Verify that there are no alerts on the DB2 pureScale cluster.

```
db2inst1@hostA1> db2cluster -list -alert
```

Appendix C - Troubleshooting

1. *A computer's IB address is not pingable after a reboot*

Ensure the IB-related devices are available:

```
root> lsdev -C | grep ib
ib0         Available      IP over InfiniBand Network Interface
iba0        Available      InfiniBand host channel adapter
icm         Available      InfiniBand Communication Manager
```

If the devices are not available, bring them online with chdev:

```
chdev -l ib0 -a state=up
```

Ensure that the ib0 , icm and iba0 properties are set correctly, that ib0 references an IB adapter such as iba0, and that properties are persistent across reboots. Use the -p option of chdev to make changes persistent across reboots. For more information, see "Setting up uDAPL and InfiniBand" in the DB2 pureScale Information Center.[4]

2. *Access to the GPFS file systems hangs for a long time on a storage controller failure*

Ensure the device driver parameters are set properly on each machine in the cluster. See *appendix A, step 14: Update storage failure timeouts* for details.

3. *The cluster comes down following a site failure*

[4] http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/index.jsp?topic=/com.ibm.db2.luw.sd.doc/doc/t0056052.html

Check the system logs on the surviving site to see if GPFS has triggered a kernel panic due to outstanding I/O requests:

```
GPFS Deadman Switch timer has expired and there are still outstanding
I/O requests
```

If this is the case, then ensure that the device driver parameters have been properly set. See *appendix A, step 14: Update storage failure timeouts* for details.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment do so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

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

Linux is a registered 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.