General Parallel File System Version 3.1 HOWTO for the IBM System Blue Gene Solution

Overview

Attention: This HOWTO provides General Parallel File System™ (GPFS™) information specific to the IBM® System Blue Gene® Solution. This document assumes that the reader understands the basic structure and components of the Blue Gene system and its software stack.

You can obtain additional Blue Gene documentation from the IBM Redbooks® Web site. Documentation at that location includes:

- *System Administration*, SG24-7178-01 at www.redbooks.ibm.com/abstracts/sg247178.html
- *Hardware Overview and Planning*, SG24-6796-01 at www.redbooks.ibm.com/abstracts/sg246796.html
- *Application Development*, SG24-7179-01 at www.redbooks.ibm.com/abstracts/sg247179.html
- *Unfolding the IBM* **@server** *Blue Gene Solution*, SG24-6686-01 at www.redbooks.ibm.com/abstracts/sg246686.html

Note: To verify that you have up-to-date information, refer to the Redbooks search page and enter Blue Gene as the search keywords.

For additional GPFS documentation, refer to the Cluster Information Center. Make certain that you retrieve the correct documentation for the GPFS version installed on your system. GPFS documentation includes:

- *Concepts, Planning, and Installation Guide*
- *Administration and Programming Reference*
  - Contains detailed information about commands noted in this HOWTO
- *Advanced Administration Guide*
- *Problem Determination Guide*
- *Data Management API Guide*
- Documentation updates
- Additional GPFS resources on the Cluster Information Center:
  - Search messages
  - FAQs
  - What's new
  - Overview
  - Tasks

Note: You can also retrieve the latest edition of this HOWTO and other IBM documents from the IBM Publication Center at http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi. From that page:

1. Select your country and click "Go"
2. On the new page, click "Search for publications"
3. The new page provides search options for keywords or publication numbers

- If you search on the publication number for this HOWTO and leave off the last two digits (search on SC23-5230), the search engine will return all available editions including the latest version.

GPFS on Blue Gene

GPFS runs on Blue Gene I/O nodes in much the same way that it does on any other GPFS cluster. The main difference is that the I/O nodes are diskless. This has two implications:

1. The files needed for the operation of GPFS are kept in NFS-mounted file systems. These files include:
   - GPFS code and utilities
   - GPFS configuration files
   - Console log files
   - syslog files
   - Traces
2. GPFS file systems use the Network Shared Disk (NSD) and remote cluster features of GPFS to access data through a TCP/IP network connection. As the Blue Gene I/O nodes are diskless, additional nodes with NSD disks attached must be used as NSD servers. These NSD server nodes are external to the Blue Gene machine itself and do not need to be dedicated to the Blue Gene machine; they may be used for other applications if desired.

   For ease of administration, place the Blue Gene I/O nodes in one GPFS cluster (hereafter called the **bgIO** cluster) and the NSD server nodes in another (hereafter called the **gpfsNSD** cluster). The nodes in the **bgIO** cluster use the remote cluster capability of GPFS to mount file systems in the **gpfsNSD** cluster.

   Utilizing the remote cluster feature:
   - Allows you to make configuration changes in the individual clusters without affecting operations in the other cluster
   - Prevents you from assigning the GPFS file system manager function to one of the Blue Gene I/O nodes which could degrade performance

The **bgIO** cluster, being composed of specialized Blue Gene I/O nodes, is somewhat unique, while the **gpfsNSD** cluster is a typical GPFS cluster. As such, much of this HOWTO covers information relating to unique aspects of setting up GPFS on the **bgIO** cluster:

Requirements for GPFS on Blue Gene

Requirements for GPFS running on Blue Gene include:

- Unique node names for each node interface used by GPFS

  These should either be listed in the **/etc/hosts** file or defined in the domain name server (DNS). This means that any node name used by GPFS cannot have more than one associated IP address listed in **/etc/hosts** or defined in the DNS. Check the **/etc/hosts** file and the DNS to make sure that this requirement is satisfied. If it is not, add unique hostnames with only one associated IP address to either the **/etc/hosts** file or the DNS, and use these hostnames when defining the GPFS cluster. Be sure to satisfy this requirement for both the **bgIO** and **gpfsNSD** clusters.

- Cluster node IP addresses

  The standard Blue Gene configuration assigns private, non-routable IP addresses to the Blue Gene I/O nodes and puts them and the service node on a private VLAN. If you want to dedicate an NSD cluster to the Blue Gene, it is possible to also assign private addresses to the nodes of the **bgNSD** cluster and put them on the virtual local area network (VLAN). However, if you want to also mount the **bgNSD** cluster from another GPFS client cluster, either all of the nodes in that cluster must also be on the VLAN, or you must give the Blue Gene I/O nodes public IP addresses and give them routes to the outside world.
- The use of **rsh** and **rcp**, or equivalent programs, for communication between the nodes of a GPFS cluster

  The **rsh** and **rcp** programs are not available on the Blue Gene I/O nodes, so the **ssh** and **scp** programs must be used instead. See"Getting ssh and scp working between the Service Node and the I/O nodes" on page 7.
- Specify the **no_root_squash** NFS export option

  The Blue Gene file system (/bgl) resides on the Service Node and it must be NFS-exported to the Blue Gene I/O nodes and any front end nodes. For this to work, you must specify the **no_root_squash** NFS export option in the **/etc/exports** file on the Service Node. This allows root access to /bgl on the Service Node from the I/O nodes and front end nodes. If you do not specify the **no_root_squash** NFS export option for /bgl, GPFS initialization will fail with permission problems. To specify the **no_root_squash** option in the **/etc/exports** file use the following format:

  ```
  /bgl    172.30.0.0/255.255.0.0(rw,no_root_squash,async)
  ```

Restrictions for GPFS on Blue Gene

In the **bgIO** cluster, the commands: **mmcrfs**, **mmchfs**, and **mmremotefs** do not support the **-A automount** option. The only supported values for the **-A** option with these commands are:

- **-A yes** (mount the file system when GPFS starts)
- **-A no** (mount the file system manually)

Blue Gene I/O node initialization

The Blue Gene Service Node initializes an I/O node by using the service network to load the operating system kernel and an initial RAM-based file system. This initial file system contains the bare minimum set of utilities and programs to allow NFS mounting of other file systems and the running of startup scripts.

Since the RAM file system contents are loaded through the service network, which is slow, it does not make sense to add all of the programs that would be needed to run GPFS into the initial RAM file system. Instead, the startup scripts change over to the *enhanced run-time environment* by renaming directories in the RAM file system to be symbolic links into an NFS-mounted directory. Thus, when GPFS is started, directories such as **/bin** and **/lib** are symbolic links into directories that have a more complete set of utilities than the minimal RAM file system.

Besides running utilities from NFS, a GPFS node also needs a place to keep configuration files, temporary working files, console output, dumps, and trace files. On the Blue Gene machine, this is done by having a startup script create a read/write directory in an NFS-mounted file system that is unique for each I/O node. This directory is symbolically-linked from the RAM file system.

The files that GPFS keeps in the per-node working directory do not necessarily need to be kept permanently. When an I/O node starts up, the directory will be recreated from scratch if it does not exist. However, startup will be faster if the contents can be saved between I/O node restarts. Also, when doing problem determination, GPFS console logs, dumps, and traces from previous runs may be helpful.

Blue Gene I/O node configuration for GPFS

Since GPFS use is optional, you must take several actions to:

- Cause GPFS to be started by the Blue Gene I/O nodes
- Ensure that GPFS is able to operate

For a full description of these actions, refer to the *Support for GPFS* section in the Blue Gene I/O node README file. This file is located in **/bgl/BlueLight/<driver>/ppc/docs/ionode.README** (where *<driver>* is the most recent Blue Gene driver). The actions described in the README ensure, among other things, that:

- The hostname for each I/O node is set from the **$BGL_SITEDISTDIR/etc/hosts** file
- The automatic startup of both the **ssh** daemon and the GPFS client on each I/O node

Installing GPFS on the nodes of the bgIO cluster

The **bgIO** cluster is composed of the Blue Gene Service Node, Blue Gene I/O nodes, and any Blue Gene front end nodes you want to include. To install GPFS on the nodes that will be in the **bgIO** cluster, follow these steps:

Step 1. "Creating the GPFS installation directory for the Service Node and any front end nodes"

Step 2. "Creating the GPFS installation directory for the I/O nodes" on page 5

Step 3. "Installing the GPFS man pages" on page 5

Step 4. "Installing GPFS on the Blue Gene Service Node and any front end nodes" on page 5

Step 5. "Installing GPFS on the Blue Gene I/O nodes" on page 6

Step 6. "Verifying the GPFS installation" on page 6

Creating the GPFS installation directory for the Service Node and any front end nodes

Attention: For all references to 3.1.0-N in this section, 'N' indicates the version of GPFS 3.1 available on the customer installation CD.

To create the GPFS installation directory:

1. On the Blue Gene Service Node, create a temporary subdirectory where GPFS installation images will be extracted. For example:

   ```
   mkdir  /tmp/gpfslpp_for_servicenode
   ```

2. Copy the self-extracting product image, **gpfs_install-3.1.0-N_sles9_ppc64**, from the GPFS for Linux® on POWER™ CD-ROM to the new directory. The image contains:
   - The GPFS product installation images.
   - The License Acceptance Process (LAP) Tool

     The LAP Tool is invoked for acceptance of the GPFS license agreements. The license agreements must be accepted to obtain access to the GPFS product installation images.
3. Verify that the self-extracting program has executable permissions.
4. Invoke the self extracting images and accept the license agreement:
   a. By default, the LAP Tool, JRE and GPFS installation images will be extracted to the target directory **/usr/lpp/mmfs/3.1.0-N_sles9_ppc64**.
   b. The license agreement files on the media may be viewed in either graphics or text-only modes. To view in graphics mode, simply invoke **gpfs_install-3.1.0-N_sles9_ppc64**. To view the license agreements in text only mode, use the **--text-only** option.

c. Use the **--silent** option to accept the license agreements.
d. Use the **--help** option to obtain usage information from the self extracting archive.

```
gpfs_install-3.1.0-N_sles9_ppc64 --silent
```

Upon license agreement acceptance, the GPFS product installation images will reside in the extraction target directory. Copy these images to the **/tmp/gpfslpp_for_servicenode** directory:

- gpfs.base-3.1.0-N.sles.ppc64rpm
- gpfs.gpl-3.1.0-N.noarch..rpm
- gpfs.msg.en_US-3.1.0-N.noarch.rpm
- gpfs.docs-3.1.0-N.noarch.rpm

The License agreements will remain available in the extraction target directory under the license subdirectory for future access. The license files are written using operating system-specific code pages. Accordingly, you may view the license in English and the local language configured on your machine. The other languages are not guaranteed to be viewable

Creating the GPFS installation directory for the I/O nodes

Attention: For all references to 3.1.0-N in this section, 'N' indicates the version of GPFS 3.1 available on the customer installation CD.

To create the GPFS installation directory for the I/O nodes :

1. On the Blue Gene Service Node, create a temporary subdirectory where GPFS installation images will be extracted. For example:

   ```
   mkdir  /tmp/gpfslpp_for_ionodes
   ```

2. Copy these GPFS images from the GPFS installation CD to the new directory:
   - gpfs.base-3.1.0-N.ppc.rpm
   - gpfs.docs-3.1.0-N.noarch.rpm
   - gpfs.gplbin-3.1.0-N.ppc.rpm
   - gpfs.msg.en_US-3.1.0-N.noarch.rpm

Installing the GPFS man pages

In order to use the GPFS man pages the **gpfs.docs** RPM Package Manager (RPM) must be installed. Once you have installed the **gpfs.docs** RPM, the GPFS manual pages will be located at **/usr/share/man/**.

Note: The **gpfs.docs** RPM need not be installed on all nodes if man pages are not desired or local file space on the node is limited.

Installing GPFS on the Blue Gene Service Node and any front end nodes

Install GPFS on the Blue Gene Service Node according to these directions:

Step 1. Issue the command:

```
cd /tmp/gpfslpp_for_servicenode
```

Step 2. Issue the command:

```
rpm -iv gpfs*.rpm
```

Step 3. Ensure you are at the current service level for GPFS for Linux on Power for SUSE LINUX ES 9 available at http://www14.software.ibm.com/webapp/set2/sas/f/gpfs/download/home.html.

Step 4. Create the GPFS binaries for the portability layer as described in **/usr/lpp/mmfs/src/README**. The files **mmfslinux**, **lxtrace**, **tracedev**, and **dumpconv** will be installed in **/usr/lpp/mmfs/bin** after you issue the commands:

```
su
make InstallImages
```

Installing GPFS on the Blue Gene I/O nodes

Install GPFS on the Blue Gene I/O nodes by issuing these commands on the Service Node:

```
cd /tmp/gpfslpp_for_ionodes
GPFS_BG=1 rpm --root /bgl/BlueLight/<driver>/ppc/bglsys/bin/bglOS --nodeps -ivh gpfs*.rpm
```

where *<driver>* is the most recent Blue Gene driver.

Note: The GPFS rpms for the Blue Gene I/O nodes include an rpm with a prebuilt GPFS Portability Layer. Therefore, you do not need to build the GPFS Portability Layer for the I/O nodes as you do when you install GPFS for SLES on the Service Node or front end nodes.

Verifying the GPFS installation

To verify the installation of the GPFS file sets:

- Check GPFS on the Service Node and all front end nodes.
  - On each of these nodes, run the command:

    ```
    rpm -qa | grep gpfs
    ```
- Check GPFS on the I/O nodes.
  - From the Service Node, run the command:

    ```
    rpm --root /bgl/BlueLight/<driver>/ppc/bglsys/bin/bglOS -qa | grep gpfs
    ```

    where *<driver>* is the most recent Blue Gene driver.

Output similar to the following should be returned:

```
gpfs.docs-3.1.0-0
gpfs.base-3.1.0-0
gpfs.msg.en_US-3.1.0-0
gpfs.gpl-3.1.0-0
```

Reinstalling GPFS after upgrading the Blue Gene driver

GPFS for the I/O nodes must be reinstalled if the Blue Gene driver is upgraded. Reinstall GPFS by following the steps in "Installing GPFS on the Blue Gene I/O nodes."

Applying service to GPFS

Attention: For all references to 3.1.0-N in this section, 'N' indicates the latest version of GPFS 3.1 available on the GPFS download website.

Attention: The update rpms that you use for GPFS service require a system with GPFS already installed. If your system requires the initial GPFS installation, you can obtain the full-install version of GPFS for Blue Gene from the customer installation CD.

GPFS service has two parts:

- Applying service to the Service Node and front end nodes

- Applying service to the Blue Gene I/O nodes

Applying service to the Service Node and front end nodes:

The current service level for GPFS for Linux on Power for SUSE LINUX ES 9 is available at http://www14.software.ibm.com/webapp/set2/sas/f/gpfs/download/home.html

Applying service to the Blue Gene I/O nodes:

1. From the GPFS download website, copy the current service level of the following GPFS 3.1 rpms for Blue Gene V1.3 I/O node images to the **/tmp/gpfslpp_for_ionodes** directory on the Service Node:
   - gpfs.base-3.1.0-N.ppc.update.rpm
   - gpfs.docs-3.1.0-N.noarch.rpm
   - gpfs.gplbin-3.1.0-N.ppc.rpm
   - gpfs.msg.en_US-3.1.0-N.noarch.rpm
2. Install GPFS service on the Blue Gene I/O nodes by issuing these commands on the Service Node:

```
cd /tmp/gpfslpp_for_ionodes
GPFS_BG=1 rpm --root /bgl/BlueLight/<driver>/ppc/bglsys/bin/bglOS --nodeps -Uvh gpfs*.rpm
```

   where <*driver*> is the most recent Blue Gene driver.

Getting ssh and scp working between the Service Node and the I/O nodes

Definitions of terms for this unit:

- **$BGL_SITEDISTDIR** is normally the **/bgl/dist** directory.
- **$BGL_SNIP** is the Service Node's IP address on the functional network.
- **$SN_HOSTNAME** is the Service Node's hostname on the functional network.
- **$IONODE_IPS** is a wildcarded IP address representing all I/O nodes.

  For example, if the I/O nodes have IP addresses 172.30.100.1 through 172.30.100.128, and 172.30.101.1 through 172.30.101.128, a reasonable value for **$IONODE_IPS** would be **172.30.10?.***
- **$IONODE_HOSTNAMES** is a wildcarded hostname representing all I/O nodes.

  For example, if the I/O nodes have host names such as **ionode1** or **ionode2**, a reasonable value for **$IONODE_HOSTNAMES** would be **ionode***.

To get **ssh** and **scp** working between the nodes of the **bgIO** cluster, these instructions should be performed by the root user on the Blue Gene Service Node:

Step 1. In the **/etc/hosts** file:

a. Find the IP address for the Service Node on the functional network.

b. Make certain that the hostname associated with the Service Node IP address, referred to here as **$SN_HOSTNAME**, is not used with any other IP address.
   - If the hostname is not unique, you can fix this by adding a new entry with: the IP address for the Service Node on the functional network, a new hostname, and a new short name.

c. Configure the **bgIO** GPFS cluster to use that unique hostname.

Step 2. In the **/etc/hosts** file, add an entry for each I/O node.

Note: It is recommended that you set up the I/O nodes of your Blue Gene system using fixed I/O address to physical location mappings. You can set up this mapping in the **/discovery/runPopIpPool** script by using the **(location, machineserialnumber, ipaddress)** form.

Step 3. Run the following commands to place a duplicate of the **/etc/hosts** file into a location where the I/O nodes can access it:

```
cp /etc/hosts $BGL_SITEDISTDIR/etc/hosts
chmod 644 $BGL_SITEDISTDIR/etc/hosts
```

Step 4. Run the following commands to create an **ssh** identity for the root user of the I/O node:

```
chown root:root /bgl
chmod 755 /bgl
chown root:root $BGL_SITEDISTDIR
chmod 755 $BGL_SITEDISTDIR
mkdir $BGL_SITEDISTDIR/root
chmod 700 $BGL_SITEDISTDIR/root
mkdir $BGL_SITEDISTDIR/root/.ssh
chmod 700 $BGL_SITEDISTDIR/root/.ssh
ssh-keygen -t rsa -b 1024 -f $BGL_SITEDISTDIR/root/.ssh/id_rsa -N ''
```

Step 5. Run the following commands to create an **ssh** identity for the I/O node:

```
chown root:root $BGL_SITEDISTDIR/etc
chmod 755 $BGL_SITEDISTDIR/etc
mkdir $BGL_SITEDISTDIR/etc/ssh
chmod 755 $BGL_SITEDISTDIR/etc/ssh
ssh-keygen -t rsa -b 1024 -f $BGL_SITEDISTDIR/etc/ssh/ssh_host_rsa_key -N ''
```

Note: This creates one host key and all I/O nodes share that key.

Step 6. Run the following commands to identify the Service Node and I/O nodes as known hosts to the I/O nodes:

```
echo "$BGL_SNIP,$SN_HOSTNAME $(cat /etc/ssh/ssh_host_rsa_key.pub)" >> \
  $BGL_SITEDISTDIR/root/.ssh/known_hosts
echo "$IONODE_IPS,$IONODE_HOSTNAMES $(cat $BGL_SITEDISTDIR/etc/ssh/ssh_host_rsa_key.pub)" >> \
  $BGL_SITEDISTDIR/root/.ssh/known_hosts
```

Step 7. Run the following command to identify the I/O nodes and the Service Node as known hosts to the Service Node:

```
echo "$IONODE_IPS,$IONODE_HOSTNAMES $(cat $BGL_SITEDISTDIR/etc/ssh/ssh_host_rsa_key.pub)" >> \
  /root/.ssh/known_hosts
echo "$BGL_SNIP,$SN_HOSTNAME $(cat /etc/ssh/ssh_host_rsa_key.pub)" >> /root/.ssh/known_hosts
```

Step 8. Run the following commands to enable the I/O nodes and Service Node to **ssh** between each other as the root user:

```
cat $BGL_SITEDISTDIR/root/.ssh/id_rsa.pub >> $BGL_SITEDISTDIR/root/.ssh/authorized_keys
cat /root/.ssh/id_rsa.pub >> $BGL_SITEDISTDIR/root/.ssh/authorized_keys
cat $BGL_SITEDISTDIR/root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys
```

Step 9. Test that **ssh** is working between all nodes of the **bgIO** cluster.

- Using **ssh**, check that every node of the cluster can run a simple command, such as **date**, on all of the other nodes of the cluster.
- The command should succeed without the need for any command line intervention. If the command is not successful without intervention, determine and resolve the cause of the problem by reviewing the previous steps.

Note: For GPFS to function, **ssh** and **scp** must work between all pairs of nodes in the **bgIO** cluster.

Getting ssh and scp working between a front end node and the other nodes of the bgIO cluster

Before you can add a front end node to the GPFS **bgl0** cluster, you must get **ssh** and **scp** working between the front end node, the Service Node, and the I/O nodes of the cluster. After you have **ssh** and **scp** functioning, you can add the front end node to the **bgl0** cluster using the **mmaddnode** command.

Definitions of terms for this unit:

- **$BGL_SITEDISTDIR** is normally the **/bgl/dist** directory
- **$BGL_FENIP** is the IP address for the front end node on the functional network
- **$FEN_HOSTNAME** is the hostname for the front end node on the functional network

Use the following procedure to setup **ssh** and **scp** on the front end node. This procedure assumes:

- That **ssh** and **scp** are already setup and operational between:
  - The Service Node and I/O nodes
  - The Service Node and the front end node
- That the **/bgl/dist** file system is mounted on both the Service Node and front end node

Step 1. On the front end node, login as root. If authorization keys do not exist, generate keys for the root user:

```
ssh-keygen -t rsa -b 1024 -f /root/.ssh/id_rsa -N ''
```

Step 2. On the Service Node, login as root and save a copy of existing **ssh** files:

```
cd /root/.ssh
cp -p authorized_keys authorized_keys.old
cp -p known_hosts known_hosts.old
cd $BGL_SITEDISTDIR/root/.ssh
cp -p authorized_keys authorized_keys.old
cp -p known_hosts known_hosts.old
```

Step 3. On the Service Node, retrieve the public keys for the front end node:

```
cd /root
scp -p $FEN_HOSTNAME:/root/.ssh/id_rsa.pub fen_id_rsa.pub
scp -p $FEN_HOSTNAME:/etc/ssh/ssh_host_rsa_key.pub fen_ssh_host_rsa_key.pub
```

Step 4. On the Service Node:

- Add the front end node to the known hosts
- Add authorized keys for the Service Node and I/O nodes

```
cd /root/.ssh
cat /root/fen_id_rsa.pub >> authorized_keys
echo "$BGL_FENIP,$FEN_HOSTNAME $(cat /root/fen_ssh_host_rsa_key.pub)" >> known_hosts

cd $BGL_SITEDISTDIR/root/.ssh
cat /root/fen_id_rsa.pub >> authorized_keys
echo "$BGL_FENIP,$FEN_HOSTNAME $(cat /root/fen_ssh_host_rsa_key.pub)" >> known_hosts
```

Step 5. On the Service Node, copy the known hosts and authorized keys to the front end node:

```
cd /root/.ssh
scp -p known_hosts $FEN_HOSTNAME:/root/.ssh/known_hosts
scp -p authorized_keys $FEN_HOSTNAME:/root/.ssh/authorized_keys
```

Note: For GPFS to function, **ssh** and **scp** must work between all pairs of nodes in the **bgIO** cluster.

Creating the clusters and establishing cross-cluster access

Here is an example of how you can create and configure the **gpfsNSD** and the **bgIO** clusters so that the **bgIO** cluster can access a file system on the **gpfsNSD** cluster. It is assumed you have installed GPFS on the nodes in the **gpfsNSD** cluster as per the instructions in the *GPFS: Concepts, Planning, and Installation Guide* (available from the Cluster Information Center).

Tips:

1. Create the file **gpfsNSD.nodes** which contains the list of the NSD nodes, one node per line.
2. Create the file **bgIO.nodes** which contains the list of the I/O nodes, one node per line.
3. If a command requires a cluster name, use the fully-qualified cluster name. A fully-qualified cluster name includes the domain ending such as **gpfsNSD.domain**.

4. Initially create the **bgIO** cluster with just the Blue Gene Service Node. Once the configuration work is done, add the Blue Gene I/O nodes and any front end nodes you want to use to the **bgIO** cluster. This greatly reduces the amount of internode communication needed to complete the work.
5. When setting up the **bgIO** cluster, configure the Blue Gene Service Node as the primary GPFS cluster configuration server and the only quorum node. The I/O nodes should be non-quorum nodes. Having a single quorum node does not decrease fault tolerance as:
   - The I/O nodes are not able to start if the Service Node is down.
   - If the Service Node fails after the I/O nodes have started, losing quorum in the **bgIO** cluster has no implications because the file systems belong to the **gpfsNSD** cluster where quorum is maintained.

   Note: For information about configuring the primary cluster configuration server and quorum nodes, refer to the *Concepts, Planning, and Installation Guide* available from the Cluster Information Center.

Step 1. On the node **gpfsNSD01**:

a. Create the **gpfsNSD** cluster designating node **gpfsNSD01** as the primary cluster configuration server and node **gpfsNSD02** as the secondary cluster configuration server:

```
mmcrcluster -n gpfsNSD.nodes -p gpfsNSD01 -s gpfsNSD02 -C gpfsNSD.domain  -A
```

b. Generate a security key:

```
mmauth genkey
```

c. Set the cluster configuration parameters as appropriate:

```
mmchconfig pagepool=128M,dataStructureDump=/var/mmfs/tmp
```

Note: If your nodes have sufficient memory, you may increase the `pagepool` value beyond `128M`.

Step 2. On the Blue Gene Service Node **bgservice**:

a. Create a file containing a node descriptor for the service node:

```
echo "bgservice:quorum" > service.node
```

b. Create the **bgIO** cluster with the Blue Gene Service Node **bgservice** as the only node in the cluster:

```
mmcrcluster -n service.node -p bgservice -C bgIO.domain -A -r /usr/bin/ssh -R /usr/bin/scp
```

c. Generate a security key:

```
mmauth genkey
```

d. Set the cluster configuration parameters as appropriate:

```
mmchconfig pagepool=128M,dataStructureDump=/var/mmfs/tmp
```

Note: A `pagepool` value of `128M` is recommended for I/O nodes with 512 MB of memory, while a `pagepool` value of `512M` is recommended for I/O nodes with 1 GB of memory.

Step 3. On the node **gpfsNSD01**:

a. Configure authentication on the **gpfsNSD** cluster:

```
mmchconfig cipherList=AUTHONLY
scp bgservice:/var/mmfs/ssl/id_rsa.pub /root/id_rsa.pub.bgservice
mmauth add bgIO.domain -k /root/id_rsa.pub.bgservice
```

b. Start the GPFS daemon:

```
mmstartup
```

c. Create NSDs for the cluster:

```
mmcrnsd -F disks.desc
```

d. Create the **fs0** file system within the **gpfsNSD** cluster:

```
mmcrfs /gpfs/fs0 fs0 -F disks.desc -A yes -B 1M -L 8M -N 10M -n numNodes -S yes
```

Note: **numNodes** is the total number of nodes that will mount the file system.

e. Grant remote access to **fs0** to the **bgIO** cluster:

```
mmauth grant bgIO.domain -f fs0
```

Step 4. On the node **bgservice**:

a. Configure authentication:

```
mmchconfig cipherList=AUTHONLY
scp gpfsNSD01:/var/mmfs/ssl/id_rsa.pub /root/id_rsa.pub.gpfsNSD01
```

b. Make the **gpfsNSD fs0** file system accessible from the **bgIO** cluster:

```
mmremotecluster add gpfsNSD.domain -k /root/id_rsa.pub.gpfsNSD01 -n \
    gpfsNSDcontactNode1,...,gpfsNSDcontactNodeN
mmremotefs add fs0 -f fs0 -C gpfsNSD.domain -T /gpfs/fs0 -A yes
```

where `gpfsNSDcontactNode1,...,gpfsNSDcontactNodeN` is a comma-separated list of contact nodes for the **gpfsNSD** cluster.

c. Start the GPFS daemon on the **bgservice** node:

```
mmstartup
```

d. Open a new window and start an **mmcs_db_console** session

e. From the new window, allocate the BG block

- For example, if the block is named **R11_GPFS**, issue the command:

```
allocate R11_GPFS
```

f. Before proceeding to sub-step 4h, make sure that the block allocation completed successfully.

- If the allocation completed successfully, you will be able to use **ssh** to run commands on the I/O nodes.
- Test this by issuing the command, **ssh ionodeN date** (where **ionodeN** is one of your I/O nodes).

Notes:

1) If you are not able to run commands on the I/O nodes using **ssh**, determine and resolve the cause of the problem by reviewing the previous steps.
2) Using **ssh**, you must be able to run commands on the I/O nodes before you can issue the **mmaddnode** command in sub-step 4h.

g. Add all of the I/O nodes to the DNS in preparation for running **mmaddnode**.

h. Add the Blue Gene I/O nodes to the **bgIO** cluster:

```
for i in $(seq 0 $numIOnodesMinus1); do echo bgIO$i:client-nonquorum >> bgIO.nodes; done
mmaddnode -n bgIO.nodes
```

Note: The **mmaddnode** command must be issued while the I/O nodes are up and running **ssh**.

i. Add any desired front-end nodes to the **bgIO** cluster, where the **bgFrontEnd.nodes** file contains the list of the front end nodes to be included in the cluster, one node per line:

```
mmaddnode -n bgFrontEnd.nodes
```

Deleting and re-adding Blue Gene I/O nodes

Follow these steps when deleting and re-adding Blue Gene I/O nodes to the GPFS **bgIO** cluster:

Step 1. Free any allocated blocks *XX_YY* associated with the I/O nodes being deleted and verify the nodes are not pingable after the block is freed. From the Midplane Management Control System (MMCS) database console environment, issue:

```
free XX_YY
```

Step 2. From the Service Node run the **mmdelnode** command using a text file such as **old_bgIO_nodes** containing the hostnames of the nodes to be deleted from the **bgIO** cluster, listing one hostname per line:

```
/usr/lpp/mmfs/bin/mmdelnode -n old_bgIO_nodes
```

Step 3. Remove the directory associated with *IP_address* for each of the nodes deleted from the **bgIO** cluster in the Step 3:

```
rm -rf /bgl/gpfsvar/IP_address
```

Step 4. Allocate the blocks *XX_YY* associated with the I/O nodes to be added to the **bgIO** cluster. From the MMCS database console environment, issue:

```
allocate XX_YY
```

Step 5. Add the I/O nodes back into the **bgIO** cluster, where the **new_bgIO_nodes** file contains the hostnames of the nodes to be added, one hostname per line:

```
/usr/lpp/mmfs/bin/mmaddnode -n new_bgIO_nodes
```

GPFS problem determination procedures for Blue Gene

Attention: If the GPFS (**mmfsd**) daemon crashes on a Blue Gene I/O node, you must reboot the I/O node before restarting GPFS.

The main unique aspect of Blue Gene is that the nodes are diskless. Therefore, all of the console log files, dumps, and traces are kept in a common NFS-mounted directory rather than in a local directory on each node. The root of the common directory is normally **/bgl/gpfsvar**, but may be overridden by setting the **GPFS_VAR_DIR** environment variable (see "Blue Gene I/O node configuration for GPFS" on page 4). Inside that directory is a separate directory for each I/O node, using the IP address of the node for the directory name. Inside these per-node directories are the **/var/mmfs** and **/var/adm/ras** directories with the same contents as for a standard GPFS node.

For example, here is part of the directory tree that might be present for an I/O node that had IP address 10.0.0.1:

```
/bgl
  /gpfsvar
    /10.0.0.1
      /var
        /adm
          /ras
            mmfs.log.latest
            mmfs.log.previous
            mmfs.log.2005.06.09.15.29.29.bgIO1
            mmfs.log.2005.06.09.15.51.26.bgIO1
        /mmfs
          /etc
            mmfs.cfg
          /gen
            mmsdrfs
            mmfsNodeData
          /ssl
            id_rsa
            id_rsa.pub
            openssl.conf
         /tmp
            complete.map.latest
            complete.map.previous
            complete.map.2005.06.15.17.21.13.bgIO1
            complete.map.2005.06.15.18.04.05.bgIO1
            internaldump.2005.06.14.12.28.37.signal.1037.bgIO1
            trcrpt.050602.15.20.50.bgIO1
```

In addition to log files that are kept in the per-node **/var/adm/ras** directory, the Blue Gene software keeps console logs in **/bgl/BlueLight/logs/BGL** which often contain useful information, particularly for solving initialization problems.

Note: Both the *GPFS: Administration and Programming Reference Guide* and the *GPFS: Problem Determination Guide* contain information that applies to GPFS running on a Blue Gene system. To obtain these documents, refer to Cluster Information Center.

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 IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights 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 give 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
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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.

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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Intellectual Property Law
Mail Station P300
2455 South Road,
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment or a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates 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. 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 interfaces 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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

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

Intel®, Intel Inside® (logos), MMX and Pentium® are trademarks of Intel Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

Second edition (July 2008)

This edition replaces SC23-5230-00 and it applies to version 3, release 1 of General Parallel File System for the IBM System Blue Gene Solution and to all subsequent releases and modifications until otherwise indicated in new editions.

SC23-5230-01