Third Party Software: How do I integrate GPFS 4.x with Bright?

How do I integrate GPFS with Bright?

GPFS can be integrated with Bright in two ways. Either by mounting a GPFS filesystem on an existing Bright Cluster, or by adding GPFS cluster functionality to the Bright Cluster.

A. Mounting an External GPFS filesystem (GPFS over NFS)

An external GPFS filesystem can be mounted on a Bright Cluster via NFS by carrying out the following steps:

A1. Export the GPFS filesystem using NFS

- Edit `/etc/exports` and add an entry to export the GPFS filesystem to the network of the Bright Cluster:

```
/gpfs1/test <base network ip>/<netmask>(rw,fsid=745,no_root_squash,async)
```

- Make sure that the clocks of all nodes in the GPFS cluster are synchronized. If this is not done, then NFS access to the data, as well as other GPFS file system operations, may be disrupted.
- Restart NFS, and ensure that it is properly configured and running.
- Ensure that the firewall on the GPFS cluster will accept the incoming connections for NFS from the network that the Bright Cluster is on.

A2. Mount the GPFS filesystem on the Bright Cluster as NFS (when NSD is not directly attached to the nodes)

a. For the nodes that are supposed to mount the GPFS filesystem: inside their software image, make a new directory to mount the GPFS filesystem:

```
# mkdir /cm/images/default-image/gpfs1/test
```

b. Add an fsmount entry for the nodes (head and compute nodes) which are supposed to mount the GPFS filesystem:

for the head node:

```
# cmsh
% device fsmounts master
% add /gpfs1
% set device gpfs-test:/gpfs1/test
% set filesystem nfs
% commit
% !mount -a
```

for the compute nodes:

```
% category fsmounts default
% add /gpfs1
% set device gpfs-test:/gpfs1/test
% set filesystem nfs
% commit
```

A3. Mount the GPFS filesystem on the Bright Cluster using the mmmount command (when NSD is directly attached to the nodes)

Third Party Software: How do I integrate GPFS 4.x with Bright?

Follow the steps in Section B "Adding GPFS Cluster Functionality to Bright Cluster" section *without creating the NSD (steps B7 and B8).* Also step B4 can be skipped, as the cluster will already be created. You can use the `mmaddnode` command on the GPFS cluster to add the new nodes to the GPFS cluster.

B Adding GPFS Cluster Functionality to Bright Cluster

You can follow this brief guide to installing the GPFS file system on a RedHat-like system, and integrating it with Bright:

B1. Installing The GPFS Packages

- On the headnodes:

Run the installer and accept the license

```
# ./Spectrum_Scale_Protocols_Advanced-4.2.1.0-x86_64-Linux-install

Extracting License Acceptance Process Tool to /usr/lpp/mmfs/4.2.1.0 ..
.
tail -n +563 ./Spectrum_Scale_Protocols_Advanced-4.2.1.0-x86_64-Linux-
install | tar -C /usr/lpp/mmfs/4.2.1.0 -xvz --exclude=installer --excl
ude=*_rpms --exclude=*rpm  --exclude=*tgz --exclude=*deb 1> /dev/null

Installing JRE ...
tail -n +563 ./Spectrum_Scale_Protocols_Advanced-4.2.1.0-x86_64-Linux-
install | tar -C /usr/lpp/mmfs/4.2.1.0 --wildcards -xvz  ibm-java*tgz
1> /dev/null
tar -C /usr/lpp/mmfs/4.2.1.0/ -xzf /usr/lpp/mmfs/4.2.1.0/ibm-java*tgz
Defaulting to --text-only mode.
```

```
Invoking License Acceptance Process Tool ...
/usr/lpp/mmfs/4.2.1.0/ibm-java-x86_64-71/jre/bin/java -cp /usr/lpp/mmf
s/4.2.1.0/LAP_HOME/LAPApp.jar com.ibm.lex.lapapp.LAP -l /usr/lpp/mmfs/
4.2.1.0/LA_HOME -m /usr/lpp/mmfs/4.2.1.0 -s /usr/lpp/mmfs/4.2.1.0  -te
xt_only

LICENSE INFORMATION

The Programs listed below are licensed under the following
License Information terms and conditions in addition to the
Program license terms previously agreed to by Client and
IBM. If Client does not have previously agreed to license
terms in effect for the Program, the International Program
License Agreement (Z125-3301-14) applies.

Program Name: IBM Spectrum Scale V4.2.1
Program Number: 5641-GPF

Program Name: IBM Spectrum Scale V4.2.1
Program Number: 5725-Q01

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.
1

License Agreement Terms accepted.

Extracting Product RPMs to /usr/lpp/mmfs/4.2.1.0 ...
tail -n +563 ./Spectrum_Scale_Protocols_Advanced-4.2.1.0-x86_64-Linux-
install | tar -C /usr/lpp/mmfs/4.2.1.0 --wildcards -xvz  installer obj
ect_rpms ganesha_rpms gpfs_rpms rhel6 sles11 sles12 smb_rpms zimon_rpm
s manifest 1> /dev/null
   - installer
   - object_rpms
   - ganesha_rpms
   - gpfs_rpms
   - rhel6
   - sles11
   - sles12
   - smb_rpms
   - zimon_rpms
   - manifest

Removing License Acceptance Process Tool from /usr/lpp/mmfs/4.2.1.0 ..
.
```

```
rm -rf  /usr/lpp/mmfs/4.2.1.0/LAP_HOME /usr/lpp/mmfs/4.2.1.0/LA_HOME

Removing JRE from /usr/lpp/mmfs/4.2.1.0 ...
rm -rf /usr/lpp/mmfs/4.2.1.0/ibm-java*tgz

==================================================================
Product rpms successfully extracted to /usr/lpp/mmfs/4.2.1.0

   Cluster installation and protocol deployment
      To install a cluster or deploy protocols with the Spectrum Scale
 Install Toolkit:  /usr/lpp/mmfs/4.2.1.0/installer/spectrumscale -h
      To install a cluster manually:  Use the gpfs rpms located within
 /usr/lpp/mmfs/4.2.1.0/gpfs_rpms

      To upgrade an existing cluster using the Spectrum Scale Install
Toolkit:
      1) Copy your old clusterdefinition.txt file to the new /usr/lpp/
mmfs/4.2.1.0/installer/configuration/ location
      2) Then run:  /usr/lpp/mmfs/4.2.1.0/installer/spectrumscale upgr
ade -h

      To add nodes to an existing cluster using the Spectrum Scale Ins
tall Toolkit:
      1) Add nodes to the clusterdefinition.txt file:  /usr/lpp/mmfs/4
.2.1.0/installer/spectrumscale node add -h
      2) Install GPFS on the new nodes:  /usr/lpp/mmfs/4.2.1.0/install
er/spectrumscale install -h
      3) Deploy protocols on the new nodes:  /usr/lpp/mmfs/4.2.1.0/ins
taller/spectrumscale deploy -h

      To add NSDs or file systems to an existing cluster using the Spe
ctrum Scale Install Toolkit:
      1) Add nsds and/or filesystems with:  /usr/lpp/mmfs/4.2.1.0/inst
aller/spectrumscale nsd add -h
      2) Install the NSDs:  /usr/lpp/mmfs/4.2.1.0/installer/spectrumsc
ale install -h
      3) Deploy the new file system:  /usr/lpp/mmfs/4.2.1.0/installer/
spectrumscale deploy -h

======================================================================
============
To get up and running quickly, visit our wiki for an IBM Spectrum Scal
e Protocols Quick Overview:
https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki
/General%20Parallel%20File%20System%20%28GPFS%29/page/Protocols%20Quic
k%20Overview%20for%20IBM%20Spectrum%20Scale
======================================================================
=============
```

RPMs are now available in `/usr/lpp/mmfs/4.2.1.0/gpfs_rpms`

```
# cd /usr/lpp/mmfs/4.2.1.0/gpfs_rpms
```

```
# rpm -ivh gpfs.base-4.2.1-0.x86_64.rpm gpfs.docs-4.2.1-0.noarch.rpm
gpfs.gpl-4.2.1-0.noarch.rpm gpfs.msg.en_US-4.2.1-0.noarch.rpm
gpfs.gskit-8.0.50-57.x86_64.rpm
```

- On the compute nodes:

The best way to install GPFS RPMs on the regular nodes is to install them inside the software image. This guarantees that the packages persist on the nodes, and will not be erased after a sync update, a reboot, or even a full install.

```
# rpm --root /cm/images/default-image/ -ivh gpfs.base-4.2.1-0.x86_64.rpm gpfs.docs-4.2.1-0.noarch.rpm gpfs.gpl-4.2.1-0.noarch.rpm gpfs.msg.en_US-4.2.1-0.noarch.rpm gpfs.gskit-8.0.50-57.x86_64.rpm
```

The directory `default-image` in the preceding can be substituted with the appropriate software image name.

Notes:

- After installing the packages, you should append the path of GPFS binaries to the default path of the binaries:

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

- The export statement could be added to `/etc/profile.d/gpfs.sh` on the head node, and also inside the software image (for example `/cm/images/default-image/etc/profile.d/gpfs.sh`) so that the configuration becomes permanent

B2. Configure the node category exclude lists

After installing the packages, the `/var/mmfs` path should be added to the following lists as items to be excluded. This prevents the GPFS configuration files from being erased.

```
# cmsh
% category use default
% set excludelistsyncinstall
(add the following line)
no-new-files: - /var/mmfs

% set excludelistgrab
(add the following line)
- /var/mmfs

% set excludelistgrabnew
(add the following line)

- /var/mmfs


% set excludelistupdate
(add the following line)

no-new-files: - /var/mmfs

% commit
```

The "`default` category can be substituted with the appropriate category name.

Notes:

- The `/var/mmfs` shouldn't be added to the full exclude list (`excludelistfullinstall`). This is because provisioning a node in FULL mode will re-partition the hard drives and will re-create the filesystem, and will then synchronize the image, so that `/var/mmfs` will anyway already be destroyed on the node.
- In case a node was provisioned in a FULL install mode, the node should be re-added to the GPFS cluster.

B3. Building The Compatibility Layer

Before starting GPFS, the GPFS compatibility layer must be built. This layer is a set of binaries that need to be built locally from source code, to match the Linux kernel and configuration of the hosts. To build the layer:

```
# cd /usr/lpp/mmfs/src
# make LINUX_DISTRIBUTION=REDHAT_AS_LINUX Autoconfig
# make World
# make InstallImages
# make rpm
```

The "make rpm" step generates an RPM package for portability binaries, so that the compatibility layer can conveniently be deployed on other machines with an identical architecture, distribution level, and Linux kernel. (Ie: the binary is only "portable" to pretty much the same machines. That is because portability here means that it inserts modules into the kernel without any need to rebuild the kernel to support GPFS. Ie the administrator actually needs to restrict where it is deployed to ensure this kind of portability). The generated rpm, `gpfs.gplbin*.rpm`, can be installed on the software image with:

```
# rpm --root /cm/images/default-image -ivh /root/rpmbuild/RPMS/x86_64/gpfs.gplbin-3.10.0-514.2.2.el7.x86_64-4.2.1-0.x86_64.rpm
```

```
# rpm -ivh /root/rpmbuild/RPMS/x86_64/gpfs.gplbin-3.10.0-514.2.2.el7.x86_64-4.2.1-0.x86_64.rpms
```

if we are using default-image.

We also need to build the kernel modules with:

```
# LINUX_DISTRIBUTION=REDHAT_AS_LINUX mmbuildgpl

--------------------------------------------------------

mmbuildgpl: Building GPL module begins at Thu Oct 20 14:50:31 CEST 2016.

--------------------------------------------------------

Verifying Kernel Header...

  kernel version = 20632642 (2.6.32-642.6.1.el6.x86_64, 2.6.32-642.6.1)

  module include dir = /lib/modules/2.6.32-642.6.1.el6.x86_64/build/include

  module build dir   = /lib/modules/2.6.32-642.6.1.el6.x86_64/build

  kernel source dir  = /usr/src/linux-2.6.32-642.6.1.el6.x86_64/include

  Found valid kernel header file under /lib/modules/2.6.32-642.6.1.el6.x86_64/build/include

Verifying Compiler...

  make is present at /usr/bin/make

  cpp is present at /usr/bin/cpp

  gcc is present at /usr/bin/gcc

  g++ is present at /usr/bin/g++

  ld is present at /usr/bin/ld

Verifying Additional System Headers...

  Verifying kernel-headers is installed ...
```

```
    Command: /bin/rpm -q kernel-headers

    The required package kernel-headers is installed

make World ...

make InstallImages ...

--------------------------------------------------------

mmbuildgpl: Building GPL module completed successfully at Thu Oct 20 14:51:58 CEST 2016.

------------------------------------------------------
```

B4. Creating The GPFS Cluster

The command `mmcrcluster` is used to create the GPFS cluster. This command has two mandatory options:

- **-p** to specify the primary GPFS cluster configuration server
- **-N** to specify the nodes

To make things easier for installation, it is useful to create a file listing all of the nodes in the GPFS cluster using either host names or IP addresses. For example: create the file `gpfs.allnodes`, listing the nodes one per line:

```
gpfs-test.cm.cluster:quorum

node001

node002
```

Then run the `mmcrcluster` command:

```
# mmcrcluster -N gpfs.allnodes -p gpfs-test.cm.cluster -r /usr/bin/ssh

Warning: Permanently added 'gpfs-test.cm.cluster,10.141.255.254' (RSA)
 to the list of known hosts.

Thu May 23 17:18:40 CEST 2013: mmcrcluster: Processing node gpfs-test.
cm.cluster

Thu May 23 17:18:43 CEST 2013: mmcrcluster: Processing node node001.cm
.cluster mmcrcluster: Command successfully completed

mmcrcluster: Warning: Not all nodes have proper GPFS license designati
ons. Use the mmchlicense command to designate licenses as needed.

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

The **-r** option specifies the absolute path to the remote shell program. By default, rsh is used, but that is a bad idea, so we choose to replace it with ssh. To add another node to the cluster after the initial creation, the `mmaddnode` command is used.

After creating the cluster with `mmrcluster`, or after adding a node with *mmaddnode*, the `mmfsEnvLevel1`, `mmfsNodeData`, and `mmsdrfs` files get created under `/var/mmfs/gen`. These files are necessary for node identification to the GPFS server. A FULL provisioning of the nodes recreates the partitions and the filesystem of the nodes, so the configurations stored under `/var/mmfs` will be destroyed. In this case, the fully provisioned node should be re-added to the cluster. This can be done by rebooting the node to be unpingable for a few moments and then removing it with the *mmdelnode* command, and re-adding it with the *mmaddnode* command:

```
# mmdelnode -N node001.cm.cluster

Verifying GPFS is stopped on all affected nodes ...

mmdelnode: Command successfully completed

# mmaddnode -N node001.cm.cluster

Fri Nov 22 15:57:30 CET 2013: mmaddnode: Processing node node001.cm.cl
uster

mmaddnode: Command successfully completed

mmaddnode: Warning: Not all nodes have proper GPFS license designation
s. Use the mmchlicense command to designate licenses as needed.

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

B5. Assign The GPFS License

The *mmchlicense* command is used to assign the appropriate GPFS license to each of the nodes in the cluster.

```
# mmchlicense server -N gpfs-test.cm.cluster
```

```
The following nodes will be designated as possessing GPFS server licenses:

gpfs-test.cm.cluster

Please confirm that you accept the terms of the GPFS server Licensing Agreement. The full text can be found at www.ibm.com/software/sla

Enter "yes" or "no": yes

mmchlicense: Command successfully completed
```

The "`server` option can be changed to `client` according to the role of the node in the cluster.

B6. Start The GPFS Cluster

Start GPFS by issuing the *mmstartup* command.

```
# mmstartup

Fri May 24 17:08:01 CEST 2013: mmstartup: Starting GPFS ...
```

You can add the **-N** option to the *mmstartup* command to add a comma-separated list of nodes that should be started:

```
# mmstartup -N gpfs-test,node001
```

B7. Create a Network Shared Disk

Create new disks for use in your file systems by issuing the *mmcrnsd* command.

```
# mmcrnsd -F descfile
```

```
mmcrnsd: Processing disk vdc
```

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

After creating the NSD, the descfile will be modified and vdc will be renamed. The new name of the vdc should be used when creating a new Filesystem on vdc using the mmcrfs command.

Note

A line in descfile should be in the following format:

```
"DiskName:ServerList::DiskUsage:FailureGroup:DesiredName:StoragePool"
```

For example,

```
"sdb:::dataAndMetadata:5::"
```

After issuing the *mmcrnsd* command, the contents of the descfile will be altered automatically so that it can be used in creating the filesystem as follows:

```
# sdb:::dataAndMetadata:5::

gpfs1nsd:::dataAndMetadata:5::system
```

For more information about the descfile, please check the man pages of `mmcrnsd`

B8. Create a new Filesystem

Create a new filesystem by issuing the *mmcrfs* command.

```
# mmcrfs gpfs1nsd -F descfile -B 512K -Q yes -T /gpfs1

The following disks of gpfs1nsd will be formatted on node gpfs-test:

gpfs1nsd: size 96468992 KB

Formatting file system ...

Disks up to size 834 GB can be added to storage pool 'system'.

Creating Inode File

Creating Allocation Maps

Creating Log Files

Clearing Inode Allocation Map

Clearing Block Allocation Map

Formatting Allocation Map for storage pool 'system'

 Completed creation of file system /dev/gpfs1nsd.
```

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

Notes:

- The **-T** option specifies the mount point directory of the GPFS file system. If it is not specified, the mount point will be set to *DefaultMountDir/Device*. The default value for *DefaultMountDir* is /gpfs but, it can be changed with the *mmchconfig* command.
- The **-Q** option activates quotas automatically when the file system is mounted. The default is no. Issue the *mmdefedquota* command to establish default quota values. Issue the *mmedquota* command to establish explicit quota values.

B9. Mount the Filesystem on the Head Node

Assuming that the head node is directly attached to the NSD, mount the newly created filesystem on it by issuing the *mmmount* command:

```
# mmmount gpfs1nsd -a

Fri May 24 17:26:07 CEST 2013: mmmount: Mounting file systems ...
```

The *mmmount* command will append an entry in /etc/fstab for the mounted GPFS filesystem if it doesn't exist:

```
# cat /etc/fstab

[...]
```

```
/dev/gpfs1nsd /gpfs1 gpfs rw,mtime,atime,quota=userquota;groupquota;fi
lesetquota,dev=gpfs1nsd,noauto 0 0
```

Notes:

- The **-a** option in mmmount command will mount the not only the head node, but on every node attached directly to the NSD. Other nodes which are not attached directly to the mmmount command will fail.
- The filesystem will be mounted on the mount point specified with the **-T** option when creating the filesystem with *mmcrfs* command. If no value is specified, then the mount point will be what is defined by DefaultMountDir, which is "`/gpfs` by default. This DefaultMountDir can be changed with the *mmchconfig* command.

For more information, please refer to *GPFS: Administration and Programming Reference*.

B10. Auto Mount GPFS Filesystem on Reboot

The *mmchconfig* command can be used to configure GPFS to startup automatically on reboot:

```
# mmchconfig autoload=yes

mmchconfig: Command successfully completed

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

The **-N** option can be used to specify the node to be configured.

B11. Mount the GPFS filesystem on regular nodes

Assuming that the regular nodes are attached to the NSD via the head node**.**

- Make a new directory to mount the GPFS filesystem on it, inside the software image of the nodes which are supposed to mount the GPFS filesystem:

```
# mkdir /cm/images/default-image/gpfs1/test
```

- Add an fsmount entry on the nodes which are supposed to mount the GPFS filesystem:

```
# cmsh
% category fsmounts default
% add /gpfs1
% set device gpfs-test:/gpfs1/test
% set filesystem nfs
% commit
% !mount -a
```

Assuming that the regular nodes are attached to the NSD directly, the *mmmount* command will append an entry in `/etc/fstab` for the mounted GPFS filesystem, if it doesn't exist, and by default the filesystems of type `gpfs` get excluded by Bright so that they won't be touched by an `imageupdate` command:

```
# cat /etc/fstab

[...]

/dev/gpfs1nsd /gpfs1 gpfs rw,mtime,atime,quota=userquota;groupquota;filesetquota,dev=gpfs1nsd,noauto 0 0
```

Notes:

- The node-installer checks the disk layout XML schema and mounts the the filesystems specified in it, but it will not mount what is specified in `/etc/fstab` or what is defined in the fsmounts of the category or the node. So rebooting a node will not affect the GPFS filesystem as it will not be mounted at this stage, since it is not part of the disksetup XML schema.
- By default Bright excludes filesystems of type `gpfs` so that they won't get wiped by an "image update" and as a result, adding the GPFS mount points to the exclude lists is not needed.

C. Troubleshooting issues when doing these steps

Issue:

```
# mmcrcluster -N gpfs.allnodes -p gpfs-test.cm.cluster

mmdsh: rsh: gpfs-test.cm.cluster /usr/lpp/mmfs/bin/mmremote mmrpc:1:1:1302:mmrc_checkNewClusterNode_gpfs-test_22256_1367600794_: checkNewClusterNode lc/lc2 gpfs-test.cm.cluster _NOSECONDARY_ %%home%%:20_MEMBER_NODE::0:1:gpfs-test:10.141.255.254:gpfs-test.cm.cluster:client:::::gpfs-test.cm.cluster:gpfs-test:::Q:::::: _DEFAULT_ _DEFAULT_ 1402863967 1563841178:lc2: No such file or directory
```

```
mmdsh: rsh: gpfs-test.cm.cluster /usr/lpp/mmfs/bin/mmremote mmrpc:1:1:1302:: getRc mmrc_checkNewClusterNode_gpfs-test_22256_1367600794_: No such file or directory



mmcrcluster: Unexpected error from checkNewClusterNode gpfs-test.cm.cluster. Return code: 2



mmcrcluster: Command failed. Examine previous error messages to determine cause.
```

Resolution:

Use the "-r" option with the absolute path to ssh:

```
# mmcrcluster -N gpfs.allnodes -p gpfs-test.cm.cluster -r /usr/bin/ssh
```

Issue:

```
# make Autoconfig

[...]

Can't determine the distribution type. /etc/redhat-release is present, but the release name is not recognized. Please specify the distribution type explicitly.
```

Workaround:

Specify the appropriate Linux distribution:

```
# make LINUX_DISTRIBUTION=REDHAT_AS_LINUX Autoconfig
```

Issue:

```
# mmmount gpfs1nsd
```

```
Fri Nov 22 13:41:31 CET 2013: mmmount: Mounting file systems ...
```

```
mmremote: GPFS is not ready to handle commands yet.
```

```
mmmount: Command failed. Examine previous error messages to determine
cause.
```

Resolution:

GPFS needs to be started on the node:

```
# mmstartup
```

Issue:

```
# mmstartup
```

```
mmstartup: Required service not applied. Install GPFS 3.5.0.1 or later
.
```

```
mmstartup: Command failed. Examine previous error messages to determin
e cause.
```

Workaround:

```
# touch /var/mmfs/gen/IGNOREPTF1
```

Unique solution ID: #1327
Author: Michele Lamarca
Last update: 2017-02-22 14:07