April 2014 AIX EXTRA

Implementing a 3 node redundant GPFS cluster

By Jaqui Lynch

The IBM General Parallel File System (GPFS) is a highly available scalable clustered filesystem that provides both availability and performance when it is necessary to share files between LPARs. It provides concurrent shared disk access to a single global namespace and can be set up in multiple configurations, each providing different levels of redundancy. IBM has an excellent document available (ref 1) on setting up a 2 node cluster, however, in this article we will discuss setting up a 3 node cluster for full redundancy.

GPFS provides a single global namespace for data across platforms, eliminating the need for NFS and also eliminating multiple copies of the data. It can handle direct SAN connections from clients and servers or they can connect over TCP/IP. The data access appears to be the same to the client as GPFS will transparently send the block I/O request over the TCP/IP network instead of the SAN, if the client is network attached.

GPFS is very scalable and currently supports up to 2 billion files per filesystem, up to 256 filesystems and up to 8192 nodes in the cluster. AIX, Linux and Windows systems can all participate in the cluster.

Cluster Description

The current AIX setup we are migrating from has two disk subsystems, each providing disk to the LPAR and those disks are the LVM mirrored to provide cross disk subsystem redundancy. GPFS does not support LVM mirroring so we decided to take advantage of using failure groups for redundancy. We put all the disks from one disk subsystem into one failure group and all the disks in the other disk subsystem into a second failure group. In our case we were using NPIV but this will work with vSCSI or direct SAN attached disks. The reason for using three nodes instead of two is that we did not want to use a tiebreaker disk – that meant that in order to have a quorum (majority of nodes present) we needed to have three nodes instead of two.

Implementation

To keep it simple we have 3 LPARs (gpfs1, gpfs2, gpfs3) and each will have a boot disk (hdisk0) and 2 GPFS disks (hdisk1 and hdisk2). We are using NPIV to provide the disks.

The first step was to build the AIX nodes with only the boot disks attached and get them installed all at the same version of AIX (7.1 TL03 SP1). Once the LPARs were booted we set the following on each of the fiber adapters:
chdev -l fcs0 -a max_xfer_size=0x200000 -a num_cmd_elems=1024 –P
Do this on each of the fcs adapters
It is important that these be set on the VIO servers first and the VIO servers should then be rebooted.

We also set the same values on the NPIV client and then zoned and mapped the disks from the first disk subsystem to that LPAR, and ran cfgmgr

We tried using rendev to rename the disks to make it obvious where they came from, however we ran into some errors with GPFS not recognizing the disk type when we ran mmcrnsd so we left them as the old names.

We did this for each disk and also set the following parameters and put a PVID on the disks:
chdev -l hdisk1 -a queue_depth=48 -a reserve_policy=no_reserve –P
chdev –l hdisk1 –a pv=yes
Perform this on every hdisk above;
At this point you should reboot the client LPARs. After the reboot check that you see all the disks and that they have the same PVIDs, etc.

The next step is to install the GPFS software and any fixes – in our case we installed GPFS 3.5 with FP15. Once it was installed we set up SSH with no password between all of the three LPARs. There are ways to strengthen SSH to ensure that only a subset of servers can use this facility – see reference 5.

On each node create a file called /usr/local/etc/gpfs-nodes.txt and put in it a list (one per line) of the nodes in the cluster. This is the point where you have to decide whether you will use fully qualified names or not. We used fully qualified names which we also used for the hostname.

On all of the nodes the following should be added to /etc/environment
/usr/lpp/mmfs/bin should be added to the end of the PATH
WCOLL=/usr/local/etc/gpfs-nodes.txt

Additionally /etc/security/limits should be updated with fsize=-1 and nofiles=20000 or some other large number. At this point you are ready to create the cluster.

Working on LPAR gpfs1 create a file /usr/local/etc/gpfs-nodesinit.txt that contains:
gpfs1.abc.com:quorum-manager
gpfs2.abc.com:quorum-manager
gpfs3.abc.com:quorum-manager

Now create the cluster with gpfs1 as primary and gpfs2 as secondary

mmcrcluster -C GPFSCLUST1 -p gpfs1.abc.com -s gpfs2.abc.com -r /usr/bin/ssh -R /usr/bin/scp -N /usr/local/etc/gpfs-nodesinit.txt -A

Then you need to accept the licenses
mmchlicense server --accept -N gpfs1.abc.com,gpfs2.abc.com,gpfs3.abc.com

I normally check the cluster at this point using mmlscluster and mmlsconfig

At this point we create the NSDs. In this case we plan to use two failure groups. The hdisks from subsystem A will be in failure group 2 and those from subsystem B will be in failure group 3. We create a file /usr/local/etc/gpfsdisks.txt that contains:

%nsd: nsd=vahdisk1 device=/dev/hdisk1 usage=dataAndMetadata failuregroup=2
%nsd: nsd=vahdisk2 device=/dev/hdisk2 usage=dataAndMetadata failuregroup=2

%nsd: nsd=vbhdisk3 device=/dev/hdisk3 usage=dataAndMetadata failuregroup=3
%nsd: nsd=vbhdisk3 device=/dev/hdisk4 usage=dataAndMetadata failuregroup=3

The above defines the nsds and names them – I made the names match the hdisk names. On gpfs1 we then run the following to create the NSDs:
mmcrnsd -F /usr/local/etc/gpfsdisks.txt
You can now run lspv and mmlspv to see how the disks are mapped. At this point you can startup GPFS on each node from gpfs1 using mmstartup –a.
mmlsnsd will show you the NSD mappings.

Final steps include creating the filesystem – we will create /gpfsa with a blocksize of 512 across the two disks in failure group 2 and replicated across the two disks in failure group 3
mmcrfs /gpfsa gpfsa -F /usr/local/etc/gpfsdisks.txt -m2 -M2 -r 2 -R 2
(-m2 says 2 replicas for metadata, -M2 says max of 2 metadata replicas, -r2 says 2 replicas for data, -R2 says max of 2 data replicas)

On gpfs1 run mmmount all –a - this will cause the filesystem to mount on all the nodes. You can see it with df or mmdf /gpfsa and should now be able to create and delete files within the filesystem. Because we are using replicas a du –sg will show the filesystem as twice the size it really is but an ls –al will show the real size that you are using. The df will also show the total space across all the disks but the %used will be correct.

At this point you are ready to test redundancy by unzoning or disconnecting some of the disks.

In order to support network based client nodes we need to convert our current NSDs to support network connectivity. The default order of access will be local block interfaces for SAN and then NSD servers (network based). So the first step is to unmount the filesystem and convert the current primary and secondary and NSDs to support network:
mmumount gpfsa -a
mmchnsd "vahdisk1:gpfs1.abc.com,gpfs2.abc.com,gpfs3.abc.com"
mmchnsd "vahdisk2:gpfs1.abc.com,gpfs2.abc.com,gpfs3.abc.com"
mmchnsd "vbhdisk3:gpfs1.abc.com,gpfs2.abc.com,gpfs3.abc.com"
mmchnsd "vbhdisk4:gpfs1.abc.com,gpfs2.abc.com,gpfs3.abc.com"

mmlsnsd will show the changes and you can now remount the filesystem from gpfs1 using mmmount gpfsa -a

You can now add network client nodes by installing GPFS on the client LPARs, setting up ssh and licensing them as clients. You will need to update the gpfs-nodes.txt file on all the nodes. Adding a network client (gpfs4) is done from gpfs1 as follows:
mmaddnode –N gpfs4.abc.com:client
mmchlicense client --accept -N gpfs4.abc.com
On the client you can now run mmstartup and then set the client up to mount gpfsa as network only:
mmmount gpfsa -o useNSDserver=always
The above causes it to always expect to do a network mount

Summary

This is a fairly simple implementation but it does provide redundancy and is very durable. GPFS is a great way to provide shared disks in a redundant fashion with better performance than that provided by NFS. It is well worth testing out as an alternative.

References

For more information:

1. Building a 2 node cluster
   https://www.ibm.com/developerworks/aix/library/au-aix-building-two-node-gpfs-cluster/

2. GPFS Documentation Home Page - links to Admin, etc guides
   http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfsbooks.html

3. GPFS Commands
   http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.v3r5.gpfs100.doc/bl1adm_command.htm

4. GPFS Backup Questions
   https://www.ibm.com/developerworks/community/forums/html/topic?id=77777777-0000-0000-0000-000014482845

5. Strengthening the ssh configuration for GPFS
   http://www.ibm.com/developerworks/aix/library/au-aix-modifying-ssh-configuration/

6. GPFS Implementation redbook
   http://publib-b.boulder.ibm.com/abstracts/sg247844.html?Open

Useful Commands

mmlscluster
mmlsconfig
mmgetstate -aLs
mmlsmgr
mmdsh command
mmdf filesystemname (i.e. gpfsa)
mmlsnsd
mmlsdisk
lspv
mmlspv
mmlsfs filesystem
mslsfileset filesystem -d -i (for filesystem use filesystem name i.e. gpfsa)
mmadddisk
mmlspool filesystem
mmlspool filesystem all
mmlspolicy filesystem
mmlsrecoverygroup
mmlssnapshot filesystem
lslpp -l | grep gpfs
mmbackup
mmbackupconfig
mmfsmount -L
mmmount filesystem
mmumount filesystem
mmlsmount all
mmdelfs filesystem
mmfsck filesystem
mmchfs filesystem -r 2   (changes filesystem to have 2 replicas)
mmpmon