IBM Spectrum Scale FPO Cluster Maintenance Guide

IBM Spectrum Scale FPO Dev Team
Contents

1. Purpose ......................................................................................................................... 4
2. Performing the rolling upgrade of IBM Spectrum Scale .................................................. 4
   2.1 IBM Spectrum Scale Rolling Upgrade Procedure ....................................................... 4
   2.2 Upgrading other infrastructure .................................................................................. 7
3. Restarting the IBM Spectrum Scale FPO cluster ............................................................ 8
   3.1 Restarting one or several nodes ................................................................................ 8
   3.2 Restarting a large IBM Spectrum Scale cluster ......................................................... 8
4. Failure detection ............................................................................................................. 12
   4.1 Node state .................................................................................................................. 12
   4.2 Disk state .................................................................................................................. 12
   4.3 The IBM Spectrum Scale log .................................................................................... 12
5. Handling of disk failures ............................................................................................... 12
   5.1 Stopping the disk failure auto recovery operation ...................................................... 12
   5.2 Starting the disk failure recovery manually ............................................................... 13
   5.3 Handling of a physically broken disk ......................................................................... 14
6. Handling of a node failure ............................................................................................. 15
   6.1 Reboot node intentionally ......................................................................................... 15
      6.1.1 Automatic recovery of a node ............................................................................. 15
      6.1.2 Manual recovery of a node ................................................................................ 15
   6.2 Node crash and can boot up ..................................................................................... 15
      6.2.1 Automatically recovering a node ....................................................................... 15
      6.2.2 Manually recovering a node ............................................................................. 16
   6.3 Recovering from node crashes and reboot problems ................................................. 17
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Handling of multiple node failures</td>
<td>17</td>
</tr>
<tr>
<td>7.1</td>
<td>Recovering from a multiple node failure without SGPanic</td>
<td>18</td>
</tr>
<tr>
<td>7.2</td>
<td>Recovery from a multiple nodes failure with SGPanic</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>Handling of a network switch failure</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>Data locality restore</td>
<td>19</td>
</tr>
<tr>
<td>9.1</td>
<td>Checking for broken data locality</td>
<td>20</td>
</tr>
<tr>
<td>9.2</td>
<td>Restoration of the data locality</td>
<td>22</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Restoring the data locality of files without WADFG</td>
<td>22</td>
</tr>
<tr>
<td>9.2.2</td>
<td>Restoring the data locality of files with WADFG</td>
<td>23</td>
</tr>
<tr>
<td>10</td>
<td>Disk Replacement</td>
<td>24</td>
</tr>
<tr>
<td>10.1</td>
<td>Replacement of more than one active disk</td>
<td>26</td>
</tr>
<tr>
<td>10.2</td>
<td>Replacement of more than one broken disk</td>
<td>26</td>
</tr>
<tr>
<td>11</td>
<td>Reference</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>Appendix</td>
<td>28</td>
</tr>
<tr>
<td>10.1</td>
<td>A: how to collect IBM Spectrum Scale trace in large cluster</td>
<td>28</td>
</tr>
<tr>
<td>10.2</td>
<td>Appendix B: policy rule to generate a list of ill-replicated files</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>Revision History</td>
<td>30</td>
</tr>
</tbody>
</table>
1. Purpose

The purpose of this document is to describe the maintenance procedures of an IBM Spectrum Scale FPO cluster including some advanced FPO internal concepts and rolling upgrades for IBM Spectrum Scale and the system kernel in an FPO cluster, including the handling of component failures such as disk failure, node failure, and network failure. This document does not include an overall description of the IBM Spectrum Scale FPO product, instructions on how to deploy IBM Spectrum Scale, and details on how to create an IBM Spectrum Scale FPO file system for a big data solution. Before reading this document, see the IBM Spectrum Scale 4.2 Advanced Administration Guide (Chapter 17 and Chapter 18) in the IBM Spectrum Scale knowledge center and the IBM developerWorks IBM Spectrum Scale Hadoop WIKI for more information about IBM Spectrum Scale and the big data solution using IBM Spectrum Scale-FPO.

2. Performing the rolling upgrade of IBM Spectrum Scale

2.1 IBM Spectrum Scale Rolling Upgrade Procedure

If the application workload of the cluster can be suspended, you can shut down the entire cluster for upgrade. During this kind of upgrade, the IBM Spectrum Scale service is interrupted. In a production cluster when an interruption in service is not acceptable, a rolling upgrade is recommended. A rolling upgrade means upgrading node by node or failure group by failure group. In a rolling upgrade, IBM Spectrum Scale is stable on a subset of nodes.

While performing rolling upgrades:

- Only upgrade nodes from the same failure group at a time. Do not select nodes from two or more failure groups because offline nodes from two separate failure groups will leave only a single copy of data online (assuming that three-way replication has been configured).
- Do not break quorum while selecting nodes for the upgrade. An offline quorum node down will break the quorum relationship in the cluster. Therefore, you need to exclude the quorum node for the rolling upgrade of the failure group.

Prerequisites

Ensure that all the disks are in the Ready state and Up availability. You can check the status and the availability of the disks by running the mmlsdisk <fs-name> -L command.

Verify that the new IBM Spectrum Scale version is compatible with the running version by reviewing the FAQ. For instance, you cannot upgrade IBM Spectrum Scale from 3.4.0.x to 3.5.0.24 without first upgrading to 3.5.0.0 and then upgrading to the latest PTF.

Verify that the planned upgrade system kernel and Linux distro versions are compatible with IBM Spectrum Scale by reviewing the FAQ or IBM Support Center.
The IBM Spectrum Scale cluster manager and file system managers have more responsibility than other nodes. While performing maintenance on the cluster manager or the file system manager, the cluster manager or the file system manager will fail over automatically. Therefore, you must manually move the cluster manager or the file system manager to another node running the mmchmgr command and perform the maintenance of cluster manager or file system manager when the cluster is not busy.

Procedure

1. Disable auto recovery for disk failure.

To upgrade a node, you must shut down IBM Spectrum Scale running on the node. At this time, disks in the node become unreachable. Instead of letting the disks fail and automatic recovery initiate, temporarily disable auto recovery.

Run the “mmchconfig restripeOnDiskFailure=no –i” command to disable auto recovery for disk failure. The -i option means that the parameter change is permanent until changed by running the mmchconfig command and takes effect immediately. The time taken by IBM Spectrum Scale to synchronize the change to all nodes is impacted by cluster size. For instance, small clusters with less than 30 nodes synchronize very quickly. The restripeOnDiskFailure parameter is a cluster-wide configuration.

After you disable auto recovery, go to the file system manager to check whether there is auto recovery in running by the following commands:

- Run mmlsmgr to check the fs manager of one file system if you have many file systems in the cluster
- Log in to the fs manager of the file system and look for the tschdisk and tsrestripefs processing by running ps -elf | grep -e tschdisk -e tsrestripefs. If either of these processes running, it’s recommended to wait until these processes are no longer running.

2. Select the nodes that must be upgraded and plan the time of the upgrade.

In each upgrade cycle, you can only upgrade IBM Spectrum Scale on nodes with disks that have the same first two numbers in the failure group. Save the list of nodes in a file and name the file nodeList. The nodeList file must consist of one node name per line. Make a list of disks on these nodes to be upgraded in this cycle. Save the list of disks in a file and name the file diskList. The diskList file must consist of an NSD name per line. Run the “mmlsdisk Device -M” command to check which disk belongs to which node.

3. Stop the applications by using the IBM Spectrum Scale file system.

Confirm that there are no applications with open files in the file system before stopping IBM Spectrum Scale. Run the ls0f or fuse r command to check for open files.

4. Unmount the IBM Spectrum Scale file system on all nodes for this upgrade cycle by running the following command:

mmumount <fsName> -N <nodeList>
Confirm that the file system has been umounted on all related nodes by running the following command:

```
mmlsmount <fsName> -L
```

5. Suspend all disks in the nodes for this upgrade cycle so that IBM Spectrum Scale does not allocate new data blocks on these disks by running the following command:

```
mmchdisk <fsName> suspend -d <diskList>
```

IBM Spectrum Scale can still read data block from a suspended disk.

Confirm that the disks are suspended by running the following command:

```
mmlsdisk <fsName>
```

6. Shut down IBM Spectrum Scale on the nodes that must be upgraded by running the following command:

```
mmshutdown -N <nodeList>
```

Confirm IBM Spectrum Scale is "down" on these nodes.

```
mmgetstate -a
```

7. Upgrade IBM Spectrum Scale on each node in this upgrade cycle.

For information on how to install IBM Spectrum Scale packages on a node, see the IBM Spectrum Scale: Administration and Programming Reference (Chapter 2. Managing the GPFS cluster).

8. After the upgrade installation is completed on all nodes and the probability layer has been built, start IBM Spectrum Scale by running the following command:

```
mmstartup -N <nodeList>
```

Confirm that IBM Spectrum Scale is Active on all nodes by running the following command:

```
mmgetstate -a
```

9. Resume all disks that were suspended in Step 5 by running the following commands:

```
mmchdisk <fsName> resume -a or
mmchdisk <fsName> resume -d <diskList>
```

When the disks are in the Ready state but some disks have the Down availability, start these disks by running the following command:

```
mmchdisk <fsName> start -a or
mmchdisk <fsName> start -d <diskList>
```
This can take a while because IBM Spectrum Scale performs incremental data synchronization to keep the data in the suspended disks updated. The time required depends on the amount of data that has been changed while the disks were kept in the Suspended state.

Confirm that all the disks are in the Ready state with the Up availability by running the following command:

```bash
mmlsdisk <fsName>
```

10. Mount the IBM Spectrum Scale file system by running the following command:

```bash
mmmount <fsName> -N <nodeList>
```

Confirm that the IBM Spectrum Scale file system has mounted by running the following command:

```bash
mmlsmount <fsName> -L
```

11. Repeat Step 3 through Step 12 to upgrade IBM Spectrum Scale all nodes in the cluster.

12. Enable auto recovery for disk failure by running the following command:

```bash
mmchconfig restripeOnDiskFailure=yes -i
```

Ensure that you include the -i option to the mmchconfig command so that this change will be permanent and synchronized across the cluster with no requirement to restart IBM Spectrum Scale.

13. Run the applications on IBM Spectrum Scale for some time. If there are no unexpected issues, run the mmchconfig release=LATEST command to upgrade the cluster version. Run the mmchfs -V compat command to ensure that the upgrade is successful. Run mmchfs -V compat to enable backward-compatible format changes.

**Note:** After you run mmchconfig release=LATEST, you cannot revert the cluster release version back to the old version. Similarly, after you run mmchfs -V compat, you cannot downgrade the file system version to an older version.

For IBM Spectrum Scale upgrade, please check [IBM Spectrum Scale FAQ](https://www.ibm.com/support/knowledgecenter/SSSHSB_1.2.1/ibm_spectrum_scale/faq.html) or contact **scale@us.ibm.com** to verify compatibility between the different IBM Spectrum Scale major versions, before issuing mmchfs -V full. For information about specific file system format and function changes while upgrading to the current release, see the following [IBM Spectrum Scale: Administration and Programming Reference](https://www.ibm.com/support/knowledgecenter/SSSHSB_1.2.1/ibm_spectrum_scale/admin/programming.html) topic: Chapter 15, “File system format changes between versions of IBM Spectrum Scale”.

### 2.2 Upgrading other infrastructure

The same process of selecting nodes must be used while upgrading hardware firmware, operation system kernel, and other components that require stopping the IBM Spectrum Scale service on the node.
3 Repeating the IBM Spectrum Scale FPO cluster

When a node reboots due to a hardware or software issue, IBM Spectrum Scale can be automatically started and the file system can be mounted if autoload is configured as yes in mmchconfig. In a typical FPO deployment, each node has several locally attached disks. When a node stops functioning, disks attached to the node are unavailable from the cluster.

**Note:** A node must not be rebooted directly if the file system is still mounted.

After rebooting the node, the disk status of the node is uncertain. The node might be still Available and Up or Suspended and Down or Available and Down depending on the Auto Recovery configuration (mmlsconfig restripeOnDiskFailure) and the IO operations performed on the cluster.

3.1 Restarting one or several nodes

The restarting operations must break the quorum relationship, or the whole cluster will be not active because of lost quorum. See the section *Reboot node intentionally* for steps.

3.2 Restarting a large IBM Spectrum Scale cluster

About this task

Sometimes, there might be a need to restart the whole cluster because of an OS upgrade. On large FPO clusters, auto-recovery must be disabled while restarting IBM Spectrum Scale by performing the following steps:

Prerequisites

Ensure that all disks are in "ready" status and "up" availability (check this by “mmlsdisk <fs-name> -L”).

Verify that the planned upgrade system kernel and Linux distro versions are compatible with IBM Spectrum Scale by reviewing the FAQ or IBM Support Center.

1. Disable auto recovery for disk failure.

While shutting down IBM Spectrum Scale, some nodes might shut down quickly and some nodes might not. This can bring some disks Down from the fast nodes and trigger auto recovery. Therefore, temporarily disable auto recovery.

Run the “mmchconfig restripeOnDiskFailure=no -i” command to disable auto recovery for disk failure. Include the -i option so that the change is permanent and synchronized across the cluster with no requirement to restart IBM Spectrum Scale. The restripeOnDiskFailure parameter is a cluster-wide configuration.

After you disable auto recovery, you should go to the file system manager to check whether there is auto recovery in running by the following commands:

- Run mmlsmgr to check the fs manager of one file system if you have many file system in the cluster.
- Log in to the fs manager of the file system and run `ps -elf | grep -e tschdisk -e tsrestripefs`. If there are processes running, wait until they stop.

2. Stop applications by using the IBM Spectrum Scale file system.

Confirm that there are no applications with open files in the file system before stopping IBM Spectrum Scale. Use the `lsof` or `fuser` command to check for open files.

For example, to check if a given IBM Spectrum Scale file system has processes using it, run one of the following commands:

```
lsof +f --/dev/name_of_SpectrumScale_filesystem
```

or

```
fuser -m /mount_point_of_SpectrumScale_filesystem
```

3. Run `mmlsmgr` to check the file system manager node for each file system.

4. Unmount the IBM Spectrum Scale file system on all nodes for this upgrade cycle by running the following command:

```
mmumount <fsName> -a
```

Confirm that the file system has been umounted on all related nodes by running the following command:

```
mmlsmount <fsName> -L
```

5. Disable the Automatic mount option by running the following command:

```
mmchfs <fsName> -A no
```

Note: This is important for restarting the whole cluster. In a large cluster, one node might be started quickly than the others. If `-A` is not set as `no`, unnecessary disk IO operation might cause some disks from slow nodes being marked as Down.

6. Shut down IBM Spectrum Scale on the nodes that must be upgraded or otherwise worked upon by running the following command:

```
mmshutdown -N <nodeList>
```

Confirm that IBM Spectrum Scale is Down on these nodes by running the following command:

```
mgetstate -a
```

7. Upgrade IBM Spectrum Scale or perform the maintenance for the whole cluster.

8. Start the quorum nodes and the previous file system manager nodes first.

If all quorum nodes are active but the previous file system managers are not yet active, the file system manager takeover is triggered if you run some commands that will read file system descriptor, e.g.
mmllsmount, mmllsdisk, mmllspool. During the file system takeover, the newly selected file system manager will try to read the file system structure information from disks even if the file system is not mounted explicitly on any node. This might make some disks to be marked as Down.

Therefore, quorum nodes and the previous file system manager nodes (if you have more than one file system in the cluster, you must include all file system manager nodes) must be started together first. Only after all these nodes are active, go to Step 9.

9. Start the other nodes in the IBM Spectrum Scale cluster.

\texttt{mmstartup -a}

Confirm IBM Spectrum Scale is "Active" on these nodes

\texttt{mmgetstate -a}

Note: don’t run any commands (e.g. mmllsmount, mmllsdisk, mmllspool) that will read file system descriptor before all nodes are active.

10. Check the disk state and bring the disks to the Ready state with the Up availability by running the following command:

\texttt{mmllsdisk <fsName> -e}

The system will display disks that are not in the Ready state with the Up availability. Run \texttt{mmchdisk <fsName> start -a} to change the availability of the disks to Up. Similarly, run \texttt{mmchdisk <fsName> resume -a} to make the suspended and to-be-emptied disks available.

11. Mount the IBM Spectrum Scale file system

When all disks in the file system are "up", you can mount file system.

\texttt{mmllsmount <fsName> -N <nodeList>}

Confirm the IBM Spectrum Scale file system is mounted.

\texttt{mmllsmount <fsName> -L}

12. Enable auto recovery for disk failure by running the following command:

\texttt{mmchconfig restripeOnDiskFailure=yes -i}

Make sure you include the '-i' option so this change takes effect immediately (it’s impacted by cluster size, usually, for small clusters, e.g. node number less than 30 nodes, it will take short time for Spectrum Scale to sync the configuration. For large clusters, e.g. hundreds of node in the cluster, it will take some time to sync the configuration and then take effect) and permanently.

13. [Optional] Enable the Automatic mount option by running the following command:

\texttt{mmchfs <fsName> -A yes}
14. Upgrade the IBM Spectrum Scale cluster version and file system version if you have upgraded Spectrum Scale version in step 7.

Usually, after the above steps (from step 1 to step 10) are done, it’s suggested to run the applications over the Spectrum Scale for some time. If such application runs don’t show any unexpected issues, we can issue ‘mmchconfig release=LATEST’ to upgrade the cluster version to the latest. Then, run ‘mmchfs -V compat’ commands to ensure the upgrade is successful and we recommend ‘mmchfs -V compat’ to enable backward-compatible format changes. Note: after you run “mmchconfig release=LATEST”, you can’t revert the cluster release version back to the old version; after you run “mmchfs -V compat”, you can’t downgrade your file system version to old version. Usually, after the above steps (from step 1 to step 10) are done, it’s suggested to run the applications over the Spectrum Scale for some time. If such application runs don’t show any unexpected issues, we can issue ‘mmchconfig release=LATEST’ to upgrade the cluster version to the latest. Then, run ‘mmchfs -V compat’ commands to ensure the upgrade is successful and we recommend ‘mmchfs -V compat’ to enable backward-compatible format changes. Note: after you run “mmchconfig release=LATEST”, you can’t revert the cluster release version back to the old version; after you run “mmchfs -V compat”, you can’t downgrade your file system version to old version.

For major IBM Spectrum Scale upgrade, please check IBM Spectrum Scale FAQ or contact scale@us.ibm.com to verify compatibility between the different IBM Spectrum Scale major versions, before issuing ‘mmchfs -V full’. For information about specific file system format and function changes when upgrading to the current release, see the following IBM Spectrum Scale: Administration and Programming Reference (Chapter 15, “File system format changes between versions of IBM Spectrum Scale”).
4 Failure detection

4.1 Node state
To check the state of the nodes in a Spectrum Scale cluster, run mmgetstate or mmgetstate -a. A node that has the Down availability is reachable but the Spectrum Scale daemon of the node is not functioning. A node in the Unknown state cannot be reached from the node on which mmgetstate is run.

To investigate the state of a node, check if the node is not functioning or has a network issue.

4.2 Disk state
To check the state of the disks in a Spectrum Scale cluster, run mmlsdisk <fsname> -e to list all disks that are not in the Available or Up state.

4.3 The IBM Spectrum Scale log
IBM Spectrum Scale log files are saved in the /var/adm/ras/ directory on each node. Each time the IBM Spectrum Scale daemon starts, a new log file is created. The mmfs.log.latest log file is the link to the latest log. On Linux, critical information is also sent to the system log in the /var/log/messages folder.

Because the IBM Spectrum Scale cluster manager and file system managers handle cluster issues such as node leaves and disk down events, monitor the IBM Spectrum Scale log on the cluster manager and the file system manager to get the best view of the cluster and file system status.

5 Handling of disk failures
In an FPO deployment model with IBM Spectrum Scale, the restripeOnDiskFailure configuration parameter must be set to yes. This means that when a disk stops functioning, auto recovery tries to bring the disk back to the functional state. Auto recovery can enlist the help of any node in the cluster to help recover data. This might affect the file system I/O performance on all nodes because it may need to copy data from a valid disk to recover the non-functional disk.

5.1 Stopping the disk failure auto recovery operation
About this task
Auto recovery satisfies most cases. When auto recovery is running it can impact the I/O performance across the cluster. When that happens, you can stop auto recovery manually and restart it when the cluster is not busy. You must bring the non-functional disks to the functional state to protect your data. This section outlines how to stop and restart auto recovery.
**Prerequisites**

Check for disks in an abnormal state by running the `mmlsdisk <fsname> -e` command. The system displays all the disks in the file system that are not in the Ready state and do not have the Up availability. If all disks in the file system are functional, the system displays the following message: 6027-623 All disks up and ready.

1. If there is a disk in the abnormal state, auto recovery can be used to recover it.
2. To stop auto recovery, kill the tschdisk and tsrestripesfs processes on the file system manager node.
3. Log in to the IBM Spectrum Scale file system manager node (you can find the file system manager node by running `mmlsmgr`) and get the tschdisk and tsrestripesfs command processor ID through the `ps -elf | grep -e tschdisk -e tsrestripesfs` command.

Alternatively, you can check the IBM Spectrum Scale log (/var/adm/ras/mmfs.log.latest) in the file system manager node to see if the tschdisk command is still running. When the restripefs command is run by auto recovery, the command log message is redirected to /var/adm/ras/restripefsOnDiskFailure.log.<timestamp>(IBM Spectrum Scale 4.1 and IBM Spectrum Scale 4.1.1) or /var/adm/ras/autorecovery.log.<timestamp>(IBM Spectrum Scale 4.1.1 PTF1 and later).

4. Kill the tschdisk and tsrestripesfs processes by running `kill -9 <pid>`. Verify that the tschdisk and tsrestripesfs processes have been killed by running the following command: `ps -elf | grep -e tschdisk -e tsrestripesfs`

**5.2 Starting the disk failure recovery manually**

**About this task**

1. Run the “`mmlsdisk <fsname> -e`” command to check the disks that are not in the Ready state.
2. Resume all disks that are in the Suspended state.

   If there are many suspended disks, you can create a file that contains a list of all the suspended disks, one nsdname per line. Resume all the suspended disks by running the following command:

   
   ```bash
   mmchdisk <fsName> resume -d <suspendDiskList>
   ```

   Check the disk state again by running the `mmlsdisk -e` command to confirm that all disks are now in the Ready state. Disks that are still in the Suspended state, might have a hardware media or connection problem. Save the names of these disks in a file and name it `brokenDiskList`.

3. Start all disks that are not functioning.

   Save all the disks that do not have the Up availability in a file and name it `downDiskList`. Each line in the `downDiskList` file consists of a disk name. Start these disks by running the following command:
mmchdisk <fsName> start -d <downDiskList>

Check the disk status again by running the mmldisk –e command to confirm that all disks have the Up availability. Disks that still do not have the Up availability, might have a hardware media or connection problem. Save the names of these disks in a file and name it tobeSuspendDiskList. Suspend these disks by running the following command:

mmchdisk <fsName> suspend -d <tobeSuspendDiskList>

4. Migrate the data off the broken disks.

If a disk is still in the Suspended state after attempting to restart it, it might have a hardware media or connection problem. To keep your data safe, migrate the data off these suspended disks by running the following command:

mmrestripefs <fsName> -r

After a file system restripe has completed, all the data in suspended disks is migrated to other disks. At this point all the data in the file system has the desired level of protection.

5. Repair broken disks.

Verify the disk connections and disk media for disks that are in the Suspended state and perform Step xx through Step yy again to make them functional. If a failure occurs again, delete these disks from the file system by running the mmdeldisk command (e.g. mmdeldisk <fs-name> “broken-disk1;broken-disk2”). If you are unable to delete a broken disk, contact IBM support for guidance.

5.3 Handling of a physically broken disk

If a disk is physically broken, it cannot be recovered by auto recovery or manual recovery. Broken disks must be deleted from the file system.

(a) Run mmIsconfig restripeOnDiskFailure to check if auto-recovery is enabled. When auto recovery is not enabled:

Deleting NSD disks from the file system might trigger disk or network traffic because of data protection. If the cluster is busy with application IO and the application IO performance is important, schedule a maintenance window to delete the broken disks from the file system. Perform the steps in section 5.2 to check if a disk is physically broken and handle the broken disks.

(b) When auto recovery is enabled:

When IO is being performed on the physically broken disks, IBM Spectrum Scale marks these disks as Down. After that, auto recovery suspends the disks if it fails to recover them and then restripes the data off the suspended disks. If you are using IBM Spectrum Scale 4.1.0.4 or earlier, deleting the disks that are marked as Down triggers heavy IO traffic, especially for meta data disks. On Spectrum Scale 4.1.0.4, mmdeldisk is improved. If the data on the disks that are marked as Down have been
restriped, the disk status will change to Emptied. For disks that are marked as Down in the Emptied state, mmeldisk deletes the disks directly without involving additional IO traffic.

6 Handling of a node failure

In an FPO deployment, each node has locally attached disks. When a node fails or has a connection problem with other nodes in a cluster, disks in the node become unavailable. You might have to reboot a node to repair a hardware issue or patch the operating system kernel. IBM Spectrum Scale considers both these scenarios as a node failure.

If auto recovery is enabled and a failing node is recovered within the auto recovery wait time, auto recovery handles the node failure automatically, recovering the disks that are marked as Down and ensuring all data has the desired replication. If a node is not recovered within the auto recovery wait time, auto recovery migrates the data off the disks in the failed node to other disks in cluster.

6.1 Reboot node intentionally

6.1.1 Automatic recovery of a node

To reboot a node (for example, to enable some configuration change requiring a reboot) and have it recovered without any auto recovery, check the auto recovery wait time. The auto recovery wait time is decided by the minimum value of minDiskWaitTimeForRecovery, metadataDiskWaitTimeForRecovery, and dataDiskWaitTimeForRecovery. By default, minDiskWaitTimeForRecovery is 1800 seconds, metadataDiskWaitTimeForRecovery is 2400 seconds, and dataDiskWaitTimeForRecovery is 3600 seconds. If the reboot can be accomplished within the auto recovery wait time, unmount the file system, shut down IBM Spectrum Scale, and reboot the node without disabling auto recovery.

6.1.2 Manual recovery of a node

To perform a hardware maintenance on a node, the node will have to be shut down for a long time. Perform the steps in IBM Spectrum Scale Rolling Upgrade Procedure

6.2 Node crash and can boot up

6.2.1 Automatically recovering a node

About this task

When a node crashes because of a kernel or some other critical issue and is recovered shortly within the auto recovery wait period, the IBM Spectrum Scale cluster manager automatically adds this node and auto recovery recovers the disks and repairs the dirty data in disks attached in the node. Verify if the node and disks in the node function properly and the data in these disks is updated.
Procedure

1. Check if the state of the IBM Spectrum Scale service is Active on all nodes in the cluster by running the mmgetstate -a command. If there is a node that is functioning but the IBM Spectrum Scale service state is not Active, check the IBM Spectrum Scale log (/var/adm/ras/mmfs.log.latest) in the node and run the mmstartup command on that node after resolving the error in the IBM Spectrum Scale log.

2. Check if there is a disk that is not in the Ready state and does not have the Up availability by running the mmlsdisk -e command. If all disks in the file system are in the Ready state with the Up availability, the mmlsdisk -e command displays the following message: 6027-623 All disks up and ready. If the system displays a disk name when the mmlsdisk -e command is run, see the Disk failure section to recover the disk.

3. Run the mmlsdisk command to confirm that the following warning message is not present at the end of output:

   Attention: Due to an earlier configuration change the file system may contain data that is at risk of being lost.

   If the system displays this message, there are data replicas on Suspended and To-be-emptied disks. If the Suspended and To-be-emptied disks are not physically broken, you can recover these disks and resume them. Run the mmmrestripefs -r to fix the warning message. If the disks are physically broken, you must suspend them and run mmmrestripefs -r to fix the warning message.

6.2.2 Manually recovering a node

About this task

When a node crashes and is not recovered within the short period that auto recovery is temporarily suspended, you must start the node and manually recover the disks in the node.

In this case, auto recovery migrates all data from the disks in this node to other valid disks in the cluster.

Procedure

1. Find the root cause of the node crash, resolve it, and recover the node so that it is online.

2. If the IBM Spectrum Scale autoload configuration is disabled, start the IBM Spectrum Scale daemon by running the mmstartup command. Verify that the node is in the Active state. If the node is not in the Active state, check the IBM Spectrum Scale log (/var/adm/ras/mmfs.log.latest) on the node and run the mmstartup command after the resolving the error in the log.

3. When the node is in the Active state, auto recovery recovers all the Down disks in the node. Check the IBM Spectrum Scale log (/var/adm/ras/mmfs.log.latest) and the restripefs log (/var/adm/ras/restripefsOnDiskFailure.log.latest) on file system manager node for more details.
6.3 Recovering from node crashes and reboot problems

About this task

If a node fails and cannot be recovered (a severe hardware failure, for example) auto recovery migrates all data from the disks in the node to other disks in the cluster. If the system is not recovering, delete the disks in the node and, if possible, delete the node.

Procedure

1. Log in to another cluster node and run the mmlsdisk <fs-name> -M command to get a list of disks that are attached to the failed nodes. Save the disk list in a file and name the file diskList. Each line in the diskList file consists of a disk name.

2. Run the mmdeldisk <fsName> -F <diskList> command to delete the disks attached to the failed node.

3. Run the mmdelnspd –F <diskList> command to delete NSDs that are attached to the failed node. Run mmdelnode to remove the node. If you are replacing the node with new hardware, use the same name and IP address to continue.

4. To replace the failed node with a new node, start the replacement node with the same hostname and IP address as the failed node. Install IBM Spectrum Scale packages and configure the SSH authorization with other nodes in cluster. Run the following command to restore IBM Spectrum Scale configurations to the replacement node:

   mmsdrrestore -p <cluster manager> -R <remoteFileCopyCommand> -N <replacement node>

   Run the mmlsmgr command to identify the cluster manager node. The Remote file copy command is the same as what is configured for the cluster. You can check by running mmlscluster.

5. Start IBM Spectrum Scale on the replacement node by running the mmstartup –N <replacement node> command. Confirm that the IBM Spectrum Scale state is Active by running the mmgetstate –N <replacement node command>.

6. Prepare a stanza file to create NSDs by running the mmcrnsd command and add the disks to the file system by running the mmadddisk command. For more information, see the man page of mmadddisk.

7 Handling of multiple node failures

Auto recovery must be enabled in an FPO cluster to protect data from multiple node failures. To enable auto recovery in a cluster, set mmchconfig restripeOnDiskFailure to yes -N all. If the number of concurrent failed nodes is less than the value of maxFailedNodesForRecovery, auto recovery protects data against node failure and disk failure. If the number of concurrent failed nodes is greater than the value of maxFailedNodesForRecovery, auto recovery exits immediately without any action and the administrator must take actions to recover it.
7.1 Recovering from a multiple node failure without SGPanic

Step 1) Recover the failed nodes.

Step 2) If all nodes are recovered within a short period of time, run mmlsdisk <fs-name> -e to view the down disk list.

Step 3) Run “mmlsnnsd –X” to check if there are disks that are unrecognized by the operating system of nodes.

The following is an example output of the mmlsnnsd –X command:

```
# mmlsnnsd -X

Disk name   NSD volume ID   Device         Devtype   Node name                Remarks
--------------------------------------------------------------------------
mucxs131d01 AC170E46561E7A8F /dev/sdb       generic mucxs131.cn.ibm.com server node
mucxs131d02 AC170E46561E7A90 /dev/sdc       generic mucxs131.cn.ibm.com server node
mucxs531d07 AC170E4B5612838E /dev/sdh       generic mucxs531.cn.ibm.com server node
mucxs531d08 AC170E4B56128391 -              mucxs531.cn.ibm.com (not found) server node
```

In this output, the physical disk for the mucxs531d08 NSD is not recognized by the OS. Check the corresponding node to see if the disk is physically broken or if the cable of disk adapter is loose. If the unrecognized disks cannot be recovered, remove them from the down disk list.

Step 4) Run mmchdisk <fs-name> start -d <down disk list in step3>.

**Note:** If the command does not run successfully, open a PMR.

Step 5) If the “not found” disks in step3) can’t be brought back(e.g. physically broken), then you need to run “mmrestripefs <fs-name> -r” to fix the replica of data whose part of replica are located in these “not found” disks.

7.2 Recovery from a multiple nodes failure with SGPanic

For internal disk rich storage, typically FPO clusters, unmountOnDiskFail must be configured as meta

If the file system SGPanic reports inactive nodes, there are more than three (assuming that the file system meta replica is 3) nodes with meta data disk down or there are more than 3 disks with meta data down.

Perform the steps in the section 8.1 to fix the issues. Run mmfsck -n to scan the file system to ensure that mmfsck reports File system is clean. If mmfsck -n does not report File system is clean, open a PMR to report the issue.
8 Handling of a network switch failure

In an FPO cluster, if auto recovery is enabled and the number of inactive nodes is greater than the value of `maxFailedNodesForRecovery`, auto recovery will not take any action because it requires the manual intervention of an administrator. By default, the value of `maxFailedNodesForRecovery` is 3 nodes. You can change this value depending on your cluster configuration by running the `mmchconfig` command.

If there is a switch network failure, nodes might be reported as inactive. If you want auto recovery to enable recovery from switch network failures, careful planning is required in setting up the FPO cluster. For example, one network switch failure must not make disks with metadata inactive from 3 or more failure groups, and the value of `maxFailedNodesForRecovery` must be set to a number that is greater than the number of inactive nodes that will result from a switch network failure.

9 Data locality restore

In an FPO cluster, if the data storage pool is enabled with `allowWriteAffinity=yes`, the data locality will be decided in the following order:

1) WADFG is explicitly set by `mmchattr` or set by policy

2) default WAD or WAD set by policy and data ingesting node

If the file is set with WADFG, the locality will comply with WADFG irrespective of where the data is ingested. If the file is not set with WADFG, the locality is decided according to WAD and the data-ingesting node. Also, data locality configurations are not mandatory configurations. If there are no disks available to comply with the configured data locality, IBM Spectrum Scale FPO will take other disks to restore the data.

The data locality might be broken if there are `mmrestripefs -r` and `mmrestripefile` without careful consideration after disk failure or node failure. If applications need data locality for good performance, restore the data locality after a node failure or a disk failure.

Data locality impacted from inactive disks:

If you have multiple disks in the same node (all these disks must be configured in the same failure group or locality group), after a disk becomes inactive, `mmrestripefs -r` from auto recovery will suspend the disk and restripe the data on the inactive disks onto other disks in the same locality group. The restriping will not break the data locality because the data from local disks is present in that node. If you do not have other disks available in the same locality group (e.g. no other disks with the same failure group x,y,z), `mmrestripefs -r` from auto recovery will restripe the data on the inactive disks onto other nodes. The restriping might break the data locality for applications running on that node.

Data locality impacted from inactive nodes:
If an inactive node has no NSD disk in the file system, the data locality will not be impacted. If the inactive node has NSD disks in the file system and the node is not recovered within `dataDiskWaitTimeForRecovery` (if all down disks are dataOnly disks) and `metadatadiskWaitTimeForRecovery` (if there is met data NSD disk down), auto recovery will suspend these disks and perform mmrestripefs -r. Now, all disks from the inactive node are not available for write operations and the data from inactive disks is restriped onto other nodes. The restriping breaks the data locality on the inactive nodes.

**Data locality impacted from unintended mmrestripefile -b or mmrestripefs -b:**

If the file is not set with WADFG by policy or by mmchattr, both mmrestripefile -b and mmrestripefs -b can break the data locality.

**Data locality impacted from unintended “mmrestripefile -l”:**

If the file is not set with WADFG by policy or by mmchattr, mmrestripefs -l can break the data locality. The node running mmrestripefile -l is considered as the data writing node and the first replica of data is stored in the data-writing node for an FPO-enabled storage pool.

### 9.1. Checking for broken data locality

1. **Check for data locality:**

   ```
   cd /usr/lpp/mmfs/samples/fpo/
g++ -g -DGPF_SNC_FILEMAP -o tsGetDataBlk -I/usr/lpp/mmfs/include/ tsGetDataBlk.C -L/usr/lpp/mmfs/lib/ -lgpfs
   ./tsGetDataBlk <filename> -s 0 -f <data-pool-block-size * blockGroupFactor> -r 3
   ```

2. **Check the output of the tsGetDataBlk program:**

   ```
   [root@gpfstest2 sncfs]# /usr/lpp/mmfs/samples/fpo/tsGetDataBlk /sncfs/test -r 3
   File length: 1073741824, Block Size: 2097152
   Parameters: startoffset:0, skipfactor: META_BLOCK, length: 1073741824, replicas 3
   numReplicasReturned: 3, numBlksReturned: 4, META_BLOCK size: 268435456
   Block 0 (offset 0) is located at disks:  2   4   6
   Block 1 (offset 268435456) is located at disks:  2   4   6
   Block 2 (offset 536870912) is located at disks:  2   4   6
   Block 3 (offset 805306368) is located at disks:  2   4   6
   ```
In the above example, the block size of data pool is 2Mbytes, the blockGroupFactor of the data pool is 128. So, the META_BLOCK(or chunk) size is $2M * 128 = 256$Mbytes. Each output line represent one chunk, e.g. Block 0 in the above is located in the disks with disk id 2, 4 and 6 for 3 replica.

3. To know the node on which the three replicas of Block 0 are located, check the mapping between disks and nodes by running the `mmlsdisk` (the 9th column is the disk id of NSD) and `mmlsnsd`:

```bash
[root@gpfstest2 sncfs]# mmlsdisk sncfs -L

<table>
<thead>
<tr>
<th>disk</th>
<th>driver</th>
<th>sector</th>
<th>failure</th>
<th>holds</th>
<th>status</th>
<th>availability</th>
<th>disk id</th>
<th>pool</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>node1_sdb  nsd</td>
<td>512</td>
<td>1</td>
<td>Yes</td>
<td>No</td>
<td>ready</td>
<td>up</td>
<td>1</td>
<td>system</td>
<td>desc</td>
</tr>
<tr>
<td>node1_sdc  nsd</td>
<td>512</td>
<td>1,0,1</td>
<td>No</td>
<td>Yes</td>
<td>ready</td>
<td>up</td>
<td>2</td>
<td>datapool</td>
<td></td>
</tr>
<tr>
<td>node2_sda  nsd</td>
<td>512</td>
<td>1</td>
<td>Yes</td>
<td>No</td>
<td>ready</td>
<td>up</td>
<td>3</td>
<td>system</td>
<td></td>
</tr>
<tr>
<td>node2_sdb  nsd</td>
<td>512</td>
<td>2,0,1</td>
<td>No</td>
<td>Yes</td>
<td>ready</td>
<td>up</td>
<td>4</td>
<td>datapool</td>
<td></td>
</tr>
<tr>
<td>node6_sdb  nsd</td>
<td>512</td>
<td>2</td>
<td>Yes</td>
<td>No</td>
<td>ready</td>
<td>up</td>
<td>5</td>
<td>system</td>
<td>desc</td>
</tr>
<tr>
<td>node6_sdc  nsd</td>
<td>512</td>
<td>3,0,1</td>
<td>No</td>
<td>Yes</td>
<td>ready</td>
<td>up</td>
<td>6</td>
<td>datapool</td>
<td></td>
</tr>
<tr>
<td>node7_sdb  nsd</td>
<td>512</td>
<td>2</td>
<td>Yes</td>
<td>No</td>
<td>ready</td>
<td>up</td>
<td>7</td>
<td>system</td>
<td></td>
</tr>
<tr>
<td>node7_sdd  nsd</td>
<td>512</td>
<td>4,0,2</td>
<td>No</td>
<td>Yes</td>
<td>ready</td>
<td>up</td>
<td>8</td>
<td>datapool</td>
<td></td>
</tr>
<tr>
<td>node11_sdb nsd</td>
<td>512</td>
<td>3</td>
<td>Yes</td>
<td>No</td>
<td>ready</td>
<td>up</td>
<td>9</td>
<td>system</td>
<td>desc</td>
</tr>
<tr>
<td>node11_sdd nsd</td>
<td>512</td>
<td>1,1,1</td>
<td>No</td>
<td>Yes</td>
<td>ready</td>
<td>up</td>
<td>10</td>
<td>datapool</td>
<td>desc</td>
</tr>
<tr>
<td>node9_sdb  nsd</td>
<td>512</td>
<td>3</td>
<td>Yes</td>
<td>No</td>
<td>ready</td>
<td>up</td>
<td>11</td>
<td>system</td>
<td></td>
</tr>
<tr>
<td>node9_sdd  nsd</td>
<td>512</td>
<td>2,1,1</td>
<td>No</td>
<td>Yes</td>
<td>ready</td>
<td>up</td>
<td>12</td>
<td>datapool</td>
<td></td>
</tr>
<tr>
<td>node10_sdc nsd</td>
<td>512</td>
<td>4</td>
<td>Yes</td>
<td>No</td>
<td>ready</td>
<td>up</td>
<td>13</td>
<td>system</td>
<td>desc</td>
</tr>
<tr>
<td>node10_sdf nsd</td>
<td>512</td>
<td>3,1,1</td>
<td>No</td>
<td>Yes</td>
<td>ready</td>
<td>up</td>
<td>14</td>
<td>datapool</td>
<td></td>
</tr>
<tr>
<td>node12_sda nsd</td>
<td>512</td>
<td>4</td>
<td>Yes</td>
<td>No</td>
<td>ready</td>
<td>up</td>
<td>15</td>
<td>system</td>
<td></td>
</tr>
<tr>
<td>node12_sdb nsd</td>
<td>512</td>
<td>4,1,2</td>
<td>No</td>
<td>Yes</td>
<td>ready</td>
<td>up</td>
<td>16</td>
<td>datapool</td>
<td></td>
</tr>
</tbody>
</table>
```

```bash
[root@gpfstest2 sncfs]# mmlsnsd

<table>
<thead>
<tr>
<th>File system</th>
<th>Disk name</th>
<th>NSD servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>sncfs</td>
<td>node1_sdb</td>
<td>gpfstest1.cn.ibm.com</td>
</tr>
<tr>
<td>sncfs</td>
<td>node1_sdc</td>
<td>gpfstest1.cn.ibm.com</td>
</tr>
<tr>
<td>sncfs</td>
<td>node2_sda</td>
<td>gpfstest2.cn.ibm.com</td>
</tr>
<tr>
<td>sncfs</td>
<td>node2_sdb</td>
<td>gpfstest2.cn.ibm.com</td>
</tr>
<tr>
<td>sncfs</td>
<td>node6_sdb</td>
<td>gpfstest6.cn.ibm.com</td>
</tr>
<tr>
<td>sncfs</td>
<td>node6_sdc</td>
<td>gpfstest6.cn.ibm.com</td>
</tr>
<tr>
<td>sncfs</td>
<td>node7_sdb</td>
<td>gpfstest7.cn.ibm.com</td>
</tr>
<tr>
<td>sncfs</td>
<td>node7_sdd</td>
<td>gpfstest7.cn.ibm.com</td>
</tr>
</tbody>
</table>
```

The three replicas of Block 0 are located in disk ID 2 (NSD name node1_sdc, node name gpfstest1.cn.ibm.com), disk ID 4 (NSD name node2_sdb, node name gpfstest2.cn.ibm.com), and disk ID 6 (NSD name node6_sdc, node name gpfstest6.cn.ibm.com). Check each block of the file to see whether the blocks are located at the expected locations. If all blocks are not located in the expected locations, perform Step2 to fix the data locality.

9.2 Restoration of the data locality

If the blocks of the file are not located in the expected locations, restore or change the locality of the file. IBM Spectrum Scale FPO provides an interface to control all first replicas of the blocks, all second replicas of the blocks, or all third replicas of the blocks in specific nodes. For example, the first replica of all blocks is located in a specific node so that the applications running over the node can read all the data from the local disks.

Note: Spectrum Scale FPO does not support user control of the location of only one block or a part of blocks. For example, controlling the location of Block 1 or Block 2 and not changing the location of Block 3 is not supported.

9.2.1 Restoring the data locality of files without WADFG

About this task

Perform the following task to control the first replica of all blocks. If you want to control the second and third replicas of the blocks, perform the steps in section 9.2.2.

Step 1. Check if the file is configured with wadfg:

```
mlsattr -d -L /sncfs/test
file name:       /sncfs/test
metadata replication: 3 max 3
data replication:   3 max 3
immutable:          no
appendOnly:         no
flags:
storage pool name:   datapool
fileset name:        root
```
Step 2. Select the node on which you want to store all the blocks of the first replica of the data. One disk from this node will be used to store the first replica of the file, assuming that this node has at least one local disk that serves the GPFS filesystem.

Step 3. If you are using IBM Spectrum Scale 4.1.1.0 or later, perform the following steps:

   Step3.1) Perform ssh to the target node selected in Step 2.

   Step3.2) Run mmrestripefile –l filename for each filename that has been selected for data locality.

If your Spectrum Scale code is older than 4.1.1.0 perform the following steps:

   Step3.1) Perform ssh to the target node selected in Step 2.

   Step3.2) Run mmchdisk <fs-name> suspend -d “any-one-data-disk”.

   Step3.3) Run mmchdisk <fs-name> resume -d “any-one-data-disk”.

   Step3.4) Run mmrestripefile -b filename.

Step 4. Check the data locality by running the following command:

/usr/lpp/mmfs/samples/fpo/tsGetDataBlk /sncfs/test -r 3

The first replica of all blocks is located in the target node.

9.2.2 Restoring the data locality of files with WADFG

About this task
If you want to control the location for the first, second, or third replica, set the WADFG attributes of the files by running mmchattr. If you are using Spectrum Scale is older than Spectrum Scale 4.1.1.0, perform the steps in this section to restore the data locality.

Procedure

**Step 1.** Decide the location for data replica.

**Step 2.** Run mmchattr --write-affinity-failure-group to set or update the new WADFG of the file.

**Step 3.** (This step is not needed for Spectrum Scale 4.1.1.0+)
Run mmrestripefile –l filename or mmrestripefile –b filename.

Note: In IBM Spectrum Scale 4.1.1.0 and later, the default option for mmchattr is -I yes, and the restripe function of mmrestripefile -l is performed in Step 2 when mmchattr is run. Therefore, if you are using IBM Spectrum Scale 4.1.1.0 or later, Step 3 is not required.

**Step 4.** Check the data locality by referring to Section 9.1.

10 Disk Replacement

In a production cluster, you can replace physically broken and degraded disks with new disks. Adhere to the following rules while replacing a disk:

1st) If you have inactive disks from two failure groups (for replica 3), restripe the file system to protect the data to avoid data loss from a third inactive disk from the third failure group.

2nd) Replacing disks is time-consuming because the whole inode space is scanned. Therefore, IO traffic will be triggered in the cluster. Schedule the disk replacement when the cluster is not busy.

The mmrpldisk command can be used to replace one disk in file system with a new disk and can handle one disk (NSD) in one invocation. If you want to replace only one disk, see mmrpldisk command.

Note: mmrpldisk doesn’t migrate all data from the to-be-replaced disks into the new added disks. This is one defect impacting Spectrum Scale 4.1, 4.1.1, 4.2 and 4.2.1. See the following example:

```
[root@c8f2n03 ~]# mmlsdisk sncfs –L

<table>
<thead>
<tr>
<th>disk</th>
<th>driver</th>
<th>sector</th>
<th>failure holds</th>
<th>holds</th>
<th>storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>type</td>
<td>size</td>
<td>group</td>
<td>metadata</td>
<td>data</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------</td>
<td>---------------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>n03_0</td>
<td>nsd</td>
<td>512</td>
<td>1 Yes</td>
<td>Yes</td>
<td>ready</td>
</tr>
<tr>
<td>n03_1</td>
<td>nsd</td>
<td>512</td>
<td>1 Yes</td>
<td>Yes</td>
<td>ready</td>
</tr>
<tr>
<td>n04_0</td>
<td>nsd</td>
<td>512</td>
<td>2,0,0 Yes</td>
<td>Yes</td>
<td>ready</td>
</tr>
<tr>
<td>n04_1</td>
<td>nsd</td>
<td>512</td>
<td>2,0,0 Yes</td>
<td>Yes</td>
<td>ready</td>
</tr>
<tr>
<td>n05_1</td>
<td>nsd</td>
<td>512</td>
<td>4,0,0 No</td>
<td>Yes</td>
<td>ready</td>
</tr>
</tbody>
</table>
```
Number of quorum disks: 3
Read quorum value: 2
Write quorum value: 2

[root@c8f2n03 ~]# /usr/lpp/mmfs/samples/fpo/tsGetDataBlk /sncfs/log -s 0
File length: 1073741824, Block Size: 1048576
Parameters: startoffset:0, skipfactor: META_BLOCK, length: 1073741824, replicas 0
numReplicasReturned: 2, numBlksReturned: 8, META_BLOCK size: 134217728
Block 0 (offset 0) is located at disks: 2 5
Block 1 (offset 134217728) is located at disks: 2 3
Block 2 (offset 268435456) is located at disks: 2 5
Block 3 (offset 402653184) is located at disks: 2 3
Block 4 (offset 536870912) is located at disks: 2 5
Block 5 (offset 671088640) is located at disks: 2 3
Block 6 (offset 805306368) is located at disks: 2 5
Block 7 (offset 939524096) is located at disks: 2 3

[root@c8f2n03 ~]# mmrpldisk sncfs n03_1 n03_4

[root@c8f2n03 ~]# mmlsdisk sncfs –L
disk         driver   sector     failure holds    holds                                    storage
name         type       size       group metadata data  status        availability disk id pool         remarks
------------ ------ ------- -- -- -- ---------- ------- ------ ----------- ------- ------- ---------
n03_0        nsd      512     1  Yes Yes ready up 1 system        desc
n04_0        nsd      512 2,0,0  Yes Yes ready up 3 system        desc
n04_1        nsd      512 2,0,0  Yes Yes ready up 4 system
n05_1        nsd      512 4,0,0  No Yes ready up 5 system        desc
n03_4        nsd      512     1  Yes Yes ready up 6 system

Number of quorum disks: 3
Read quorum value: 2
Write quorum value: 2

[root@c8f2n03 ~]# /usr/lpp/mmfs/samples/fpo/tsGetDataBlk /sncfs/log -s 0
File length: 1073741824, Block Size: 1048576
Parameters: startoffset:0, skipfactor: META_BLOCK, length: 1073741824, replicas 0
numReplicasReturned: 2, numBlksReturned: 8, META_BLOCK size: 134217728
Block 0 (offset 0) is located at disks: 6 5
Block 1 (offset 134217728) is located at disks: 1 3
Block 2 (offset 268435456) is located at disks: 1 5
Block 3 (offset 402653184) is located at disks: 1 3
Block 4 (offset 536870912) is located at disks: 6 5
Block 5 (offset 671088640) is located at disks: 6 3
Block 6 (offset 805306368) is located at disks: 1 5
Block 7 (offset 939524096) is located at disks: 6 3

After replacing n03_1 with n03_4, part of data located in n03_1 are migrated into n03_4 and others are migrated into n03_0. Therefore, mmrpldisk doesn’t mean copy data from the to-be-replaced disks into new added disks. **mmrpldisk might break the data locality and you need to see the Section 9 to restore data locality if needed.**

If you have more than one disk to be replaced, run mmrpldisk multiple times. The PIT job is triggered to scan the whole inode space to migrate the data off the to-be-replaced disks. Therefore, it will trigger obvious IO traffic and is time-consuming. To accelerate the replacement process, see the following sub sections.

### 10.1 Replacement of more than one active disk

If you want to replace more than one disk used in a file system and if you have a lot of files and data in the file system, running mmrpldisk for each disk will be time consuming.

You can plug new disks into idle disk slots and run mmcrnsd to create NSD disks against the to-be-added disks. Run mmadddisk (without the option `-r`) to add the disks to the file system and then mmdeldisk for the to-be-replaced disks.

**Note:** If you place the new disks in the same failure groups as the to-be-replaced disks, the above operations maintain the data locality for data from the to-be-replaced disks. This means that IBM Spectrum Scale stores this data in the original failure group.

If you do not have additional idle disk slots, run mmdeldisk on the to-be-replaced disks. Run mmcrnsd to create NSD disks and run mmadddisk to add them to the file system. You might have to run mmrestripes -b to balance the file system but this will break the data locality.

### 10.2 Replacement of more than one broken disk

The broken disks have been restriped (then, it will become emptied/down), you could run "mmdeldisk" directly and it will be very fast since 4.1.1.0. Then, you could pull out the broken disks, pull in the new disks, run mmcrnsd and then run mmdeldisk to add them into the file system.

If the broken disks have not been restriped, (they will be marked as Ready/down or Ready/up), perform the following steps:

1. **step1:** Disable auto recovery temporarily (refer the section 2.1, step 2)
   - run mmlsnd -X to check these pulled-out disks:

   ```
   node7_sdn  COA80A0756FBAA89  -  -  gpfstest7.cn.ibm.com  (not found) server node
   ```

2. **step2:** Pull out the broken disks directly
   - run mmlsnd -X to check these pulled-out disks:

   ```
   node7_sdn  COA80A0756FBAA89  -  -  gpfstest7.cn.ibm.com  (not found) server node
   ```

3. **step3:** Pull in the new disks.
4. **step4:** Run mmcrnsd for the new disks (take new NSD name)
5. **step5:** Run mmadddisk `<fs-name>` -F `<new-nsd-file from step4>`
step6: refer See the Spectrum Scale *Problem Determination Guide*, Section **Disk Issues** → *Disk media failure* to delete the broken disks from the file system.

## 11 Reference

- IBM Spectrum Scale Knowledge Center
- White paper: deploying big data solution using IBM Spectrum Scale
- IBM Spectrum Scale FPO Data Locality
10 Appendix

10.1 A: how to collect IBM Spectrum Scale trace in large cluster

README:

This tool uses system crond service to collect IBM Spectrum Scale trace in huge cluster.

mmtrace or mmtracectl command always cannot response in time in huge cluster. This tool use system crontab service to cut IBM Spectrum Scale trace in each time period. This tool also uses mmdiag command to collect waiters, status and so on in each time period. All diag output are all collected in IBM Spectrum Scale.snap tarball.

Default time period is 1 second.

How to use it:

1. Copy this tool into the same dir in EACH node.
2. <prefix>on.sh # runs in one node, set gpfs trace level and related config
3. mmdsh -N all <prefix>start.sh # start gpfs trace.
4. mmdsh -N all <prefix>stop.sh # stop gpfs trace.
5. <prefix>off.sh $ turn off gpfs trace.

Limitation:

Only support AIX and Linux.

on.sh

```bash
on.sh

echo "set gpfs trace log quiet"
/usr/lpp/mmfs/bin/mmchconfig traceLogQuiet=yes > /dev/null

eecho "set gpfs trace buffer and level"
uname -a | grep Linux > /dev/null
if (( $? -eq 0 )); then
  # Linux
else
  # AIX
  mmtracectl --set --trace-def --trace-recycle-globalOnShutdown --aix-trace-buffer-size=128M --trace-file-size=256M --tracedev-write-mode=overwrite > /dev/null
fi
```

off.sh

```bash
off.sh

echo "turn off gpfs trace"
/usr/lpp/mmfs/bin/mmtracectl --off > /dev/null

/usr/lpp/mmfs/bin/mmdiag --trace | grep "trace mode" | grep none > /dev/null
if (( $? -eq 0 )); then
  echo "works"
else
  echo "Error"
```
echo "done"

**start.sh**

echo "create working dir /tmp/mmfs/gpfstrace"
mkdir -p /tmp/mmfs/gpfstrace

echo "link mmtrace command"
l -sf /usr/lpp/mmfs/bin/mmtrace /tmp/mmfs/gpfstrace/mmtrace

echo "generate diagnostic program"
rm -rf /tmp/mmfs/gpfstrace/diag.sh
touch /tmp/mmfs/gpfstrace/diag.sh
chmod +x /tmp/mmfs/gpfstrace/diag.sh

echo "date >> /tmp/mmfs/gpfstrace.diag.log" >> /tmp/mmfs/gpfstrace/diag.sh

echo "generate diagnostic program"
rm -rf /tmp/mmfs/gpfstrace/diag.sh

echo "backup /etc/crontab file"
l -s /var/spool/cron/root > /dev/null
if [[ $? -eq 0 ]]; then
    cp -rf /var/spool/cron/root /tmp/mmfs/gpfstrace/cronfile.root
    cp -rf /var/spool/cron/root /tmp/mmfs/gpfstrace/cronfile.root.bak
else
    echo "no cron file for root"
fi

echo "append new job in /etc/crontab"
echo "*/1 * * * * run-parts /tmp/mmfs/gpfstrace" >> /tmp/mmfs/gpfstrace/cronfile.root
/usr/bin/crontab /tmp/mmfs/gpfstrace/cronfile.root

echo "restart crond"
/etc/init.d/crond restart

echo "check it's working..."
/usr/bin/crontab -l | grep gpfstrace
if [[ $? -eq 0 ]]; then
    echo "done"
else
    echo "Error"
fi

**stop.sh**

echo "restore crontab"
l -s /tmp/mmfs/gpfstrace/cronfile.root.bak
if [[ $? -eq 0 ]]; then
    /usr/bin/crontab /tmp/mmfs/gpfstrace/cronfile.root.bak
else
    echo "no cron backup file for root"
fi

echo "restart crond"
/etc/init.d/crond restart
```
echo "check whether it's working..."
/usr/bin/crontab -l | grep gpfstrace
if [[ $? -ne 0 ]]; then
  echo "works"
else
  echo "Error"
fi

echo "remove temp binary in /tmp/mmfs/gpfstrace"
rm -rf /tmp/mmfs/gpfstrace/diag.sh
rm -rf /tmp/mmfs/gpfstrace/mmtrace

echo "done"
```

### 10.2 Appendix B: policy rule to generate a list of ill-replicated files

Policy example to generate a list of ill-replicated files:

```
/home/admin/cl1]# cat illRepl.pol
RULE EXTERNAL LIST 'missRepl' EXEC /usr/lpp/mmfs/samples/ilm/mmpolicyExec-list.sample'
RULE 'missReplRule' LIST 'missRepl' where MISC_ATTRIBUTES LIKE '%J%'

/home/admin/cl1]# mmapplypolicy /gpfs/bigfs/ingest -P ./illRepl.pol
```

Note: The sample program /usr/lpp/mmfs/samples/ilm/mmpolicyExec-list.sample needs to be modified per your needs to produce a list of files matching the rules. This program needs to be executable and available on all nodes in the cluster used by the mmapplypolicy command '-N' parameter.

### 11 Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date released</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>2015-09-01</td>
<td>Initial draft from Wei(<a href="mailto:gongwbj@cn.ibm.com">gongwbj@cn.ibm.com</a>)</td>
</tr>
<tr>
<td>0.5</td>
<td>2015-09-10</td>
<td>Yong(<a href="mailto:zhengzy@cn.ibm.com">zhengzy@cn.ibm.com</a>) did some changes and public for review</td>
</tr>
<tr>
<td>0.6</td>
<td>2015-09-17</td>
<td>Merge the comments from Scott(<a href="mailto:sfadden@us.ibm.com">sfadden@us.ibm.com</a>)</td>
</tr>
<tr>
<td>0.7</td>
<td>2015-11-03</td>
<td>Merge the comments from John(<a href="mailto:jlewars@us.ibm.com">jlewars@us.ibm.com</a>) and Merge the comments from Jim Lee(<a href="mailto:leejim@us.ibm.com">leejim@us.ibm.com</a>) and Alifiya(<a href="mailto:alohawal@in.ibm.com">alohawal@in.ibm.com</a>)</td>
</tr>
<tr>
<td>0.7.1/2</td>
<td>2015/11/29</td>
<td>Yong(<a href="mailto:zhengzy@cn.ibm.com">zhengzy@cn.ibm.com</a>) added the section 5.3 about handling broken disks</td>
</tr>
<tr>
<td>0.7.3</td>
<td>2016/03/04</td>
<td>Yong(<a href="mailto:zhengzy@cn.ibm.com">zhengzy@cn.ibm.com</a>) added the section 3.1 and 3.2</td>
</tr>
</tbody>
</table>
according to customer cases.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7.4</td>
<td>2016/03/09</td>
<td>John(<a href="mailto:jlewars@us.ibm.com">jlewars@us.ibm.com</a>) made minor changes to section 3.2 and 6.3.</td>
</tr>
<tr>
<td>0.8</td>
<td>2016/03/28</td>
<td>Yong added the section 9 for data locality restore</td>
</tr>
<tr>
<td>0.8.2</td>
<td>2016/4/25</td>
<td>Merged the comments from John(<a href="mailto:jlewars@us.ibm.com">jlewars@us.ibm.com</a>) and Yong added the section 10 according to the case from DashDB team.</td>
</tr>
<tr>
<td>0.9</td>
<td>2016/7/14</td>
<td>Merged the comments from ID team member</td>
</tr>
</tbody>
</table>