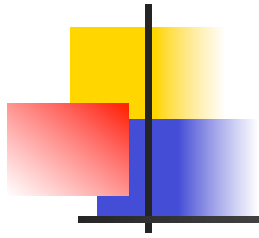




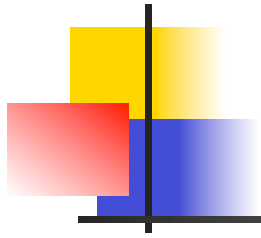
GPFS for advanced users

GPFS Callbacks



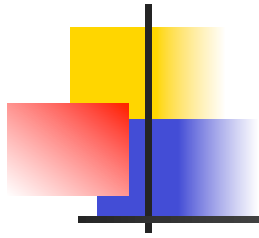
Callback

- was added in GPFS 3.3 to help users automate certain tasks
- GPFS can run user scripts when certain events occurs
 - called automatically by GPFS after the defined event happens.
- Commands:
 - `mmaddcallback` - Sets up a new pair-event to monitor executables.
 - `mmcallback` - Lists already configured callbacks.
 - `mmdelcallback` - Deletes incorrect or old callback conditions



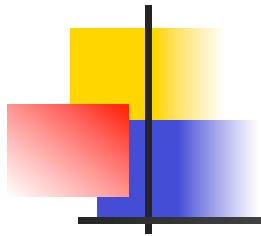
Events

- GPFS can monitor two types of events:
 - Global events trigger a callback on all nodes that are configured in the cluster.
 - Local events trigger a callback script only on the node where the event occurs.



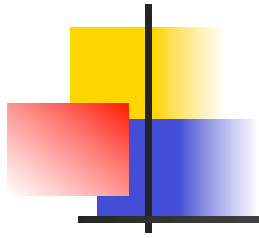
Global events

Global event	Description
nodeJoin	Triggered when one or more nodes join the cluster.
nodeLeave	Triggered when one or more nodes leave the cluster.
quorumReached	Triggered when a quorum has been established in the GPFS cluster.
quorumLoss	Triggered when a quorum has been lost in the GPFS cluster.
quorumNodeJoin	Triggered when one or more quorum nodes join the cluster.
quorumNodeLeave	Triggered when one or more quorum nodes leave the cluster.
clusterManagerTakeover	Triggered when a new cluster manager node has been elected. This event occurs when a cluster first starts or when the current cluster manager fails or resigns and a new node takes over as cluster manager.



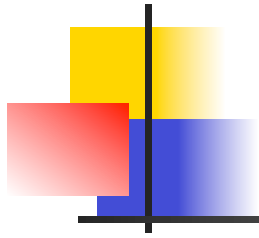
Local events

Local event	Description
lowDiskSpace	Triggered when the file system manager detects that disk space is running below the low threshold that is specified in the current policy rule.
noDiskSpace	Triggered when the file system manager detects that a disk ran out of space.
softQuotaExceeded	Triggered when the file system manager detects that a user or file set quota has been exceeded.
preMount, preUnmount, mount, unmount	Triggered when a file system is about to be mounted or unmounted or has been mounted or unmounted successfully. These events are generated for explicit <code>mount</code> or <code>unmount</code> commands, a remount after GPFS recovery, and a forced unmount when GPFS shuts down.
preStartup	Triggered after the GPFS daemon completes its internal initialization and joins the cluster, but before the node runs recovery for any VFS (GPFS-specific extensions) mount points that were already mounted, and before the node starts accepting user-initiated sessions.
startup	Triggered after a successful GPFS startup and when the node is ready for user initiated sessions.
preShutdown	Triggered when GPFS detects a failure and is about to shutdown.
shutdown	Triggered when GPFS completed the shutdown.



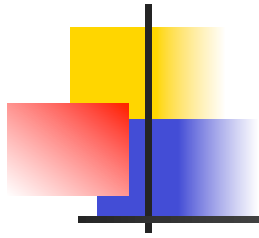
Command: mmcallback

- `mmaddcallback CallbackId --command CommandPath -event Event[,Event...] [--priority Value] [-async | --sync [--timeout Seconds] [--onerror Action]] [-N {Node[,Node...] | NodeFile | NodeClass}][--parms ParameterString ...]`
- Primarily, you set the following parameters:
 - `CallbackId` , - name that you give to your callback
 - `CommandPath` , - complete path of your script
 - `Event` , - event for which GPFS has to run your command



Example: log “nodeLeave” events

- `cat callback/nodeDown.ksh`
`#!/bin/ksh`
`echo "`date` Node: " $1 >> /root/callback/nodedown.log`
- `mmaddcallback NodeDownCallback --command /root
/callback/nodeDown.ksh --event nodeLeave --parms
%eventNode -N server1,server2`



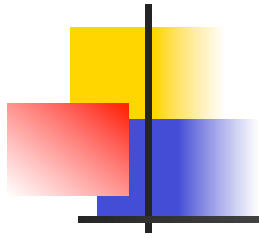
Threshold based migration using callbacks

- Policy:

```
RULE 'theshold1' MIGRATE FROM POOL 'system'  
THRESHOLD(50,40) WEIGHT(ACCESS_TIME) TO POOL 'pool1'  
RULE 'default' SET POOL 'system'
```

- Create a callback:

- The callback intercepts the *lowDiskSpace* event and calls your script. For threshold based migration the script can be pretty simple, all it needs to do is call *mmapplypolicy*.
- use the *--single-instance* option to *mmapplypolicy* to make sure multiple policies are not running at the same time on the file system.



Callback example

■ Callback script:

```
#!/bin/bash
```

```
# Parameters passed by callback
```

```
# $1 = %fsName, The name of the file system experiencing the event
```

```
fsname=$1 echo `date` > /callback/threshold.log 2>&1
```

```
/usr/lpp/mmfs/bin/mmapplypolicy $fsname --single-instance >> /callback/threshold.log 2>&1
```

■ Install callback:

- Script to call: /callback/threshold.bash

- Parameters to pass to script: %fsName (File system name)

- Node on which to execute the callback script: node01

```
mmaddcallback thresholdCallback --command /callback/threshold.bash --event lowDiskSpace --parms fsName -N  
node01
```