Device Mapper Multipath

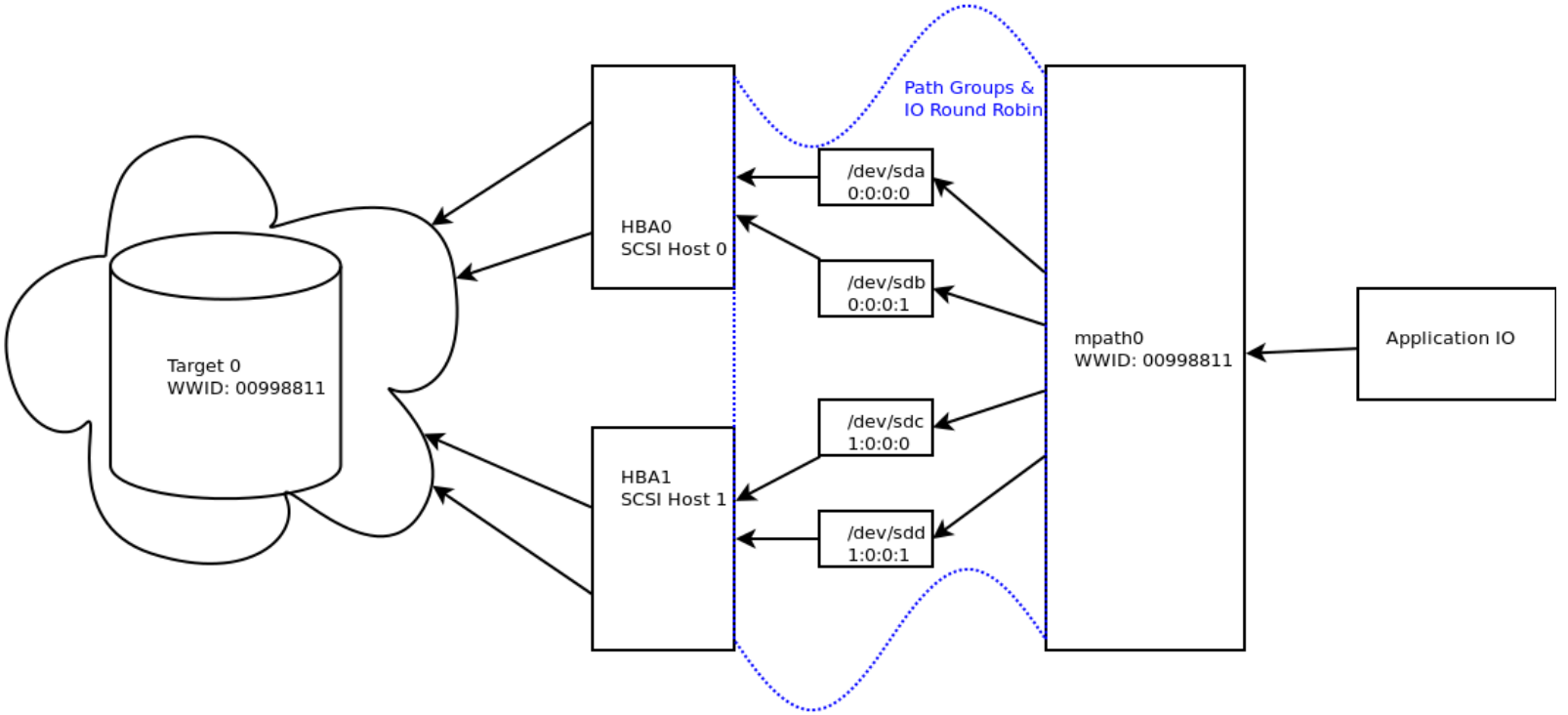
Adam Drew
Software Maintenance Engineer
Red Hat

Agenda

- What is Multipath?
- What is the Device Mapper?
- What is Device Mapper Multipath (dm-multipath)?
- What are the important dm-multipath terms & concepts?
- How do I configure dm-multipath?
- What are some differences between dm-multipath on RHEL 5 & RHEL 6?
- How can I diagnose dm-multipath problems and get help if I need it?

What is Multipath?

- Multipath is a storage network design technique that allows for fault tolerance or increased performance by providing multiple concurrent physical connections (paths) from the storage to the individual host systems

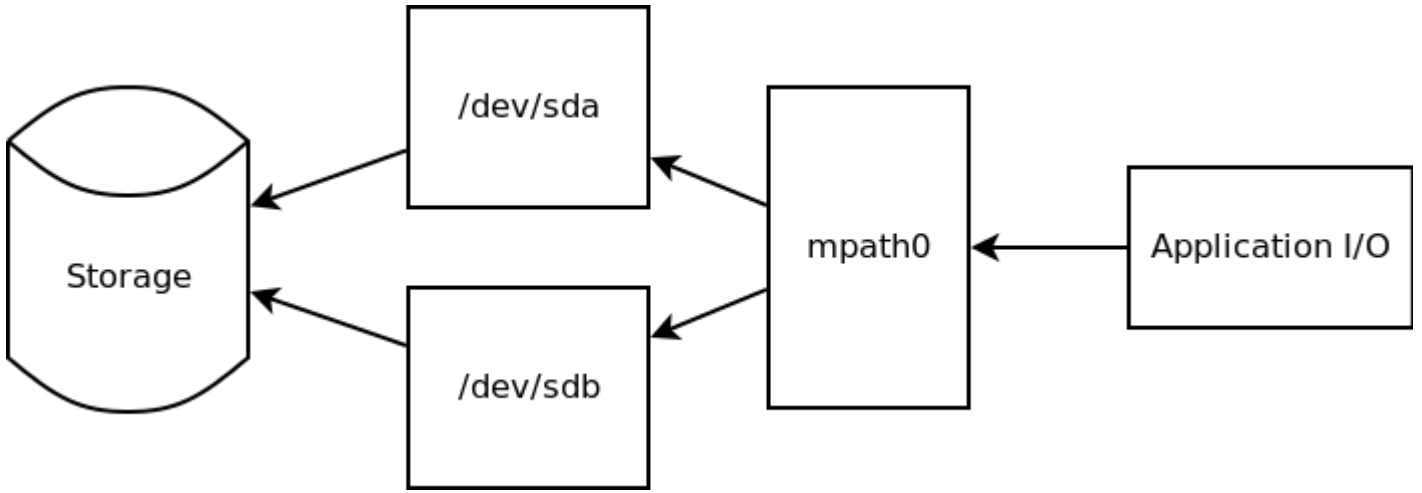


What is the Device Mapper?

- Device Mapper is the Linux kernel's device abstraction and mapping subsystem
- Device Mapper was introduced in the 2.6 Linux kernel.
- Device Mapper allows the Linux kernel to present virtual devices to user space
- These virtual devices are an abstraction that allows the kernel to route I/O to and from physical devices in a number of creative ways
- Device Mapper enables Logical Volume Management, copy-on-write snapshots, software RAID, device mirroring, device striping, and dm-multipath

Device Mapper Multipath

- Multiple paths to the storage result in multiple device nodes (e.g. /dev/sda, /dev/sdb, etc)
- dm-multipath identifies device nodes which correspond to the same LUN by their WWIDs
- Device Mapper creates a virtual device (e.g. /dev/mapper/mpath0) which routes I/O to and from the physical paths



Terms & Concepts

- LUN: Logical Unit Number. Though not an exact definition, in the common usage LUN means “the storage pool on the SAN side.”
- Path: A physical connection from the host to the LUN. An HBA with 2 ports would provide 2 paths back to the storage. 2 HBAs with 2 ports each would provide 4 paths.
- Dev Node: A Device Node. The virtual device “file” exposed to user space via /dev (e.g. /dev/sda). Each path will result in a separate dev node.
- WWID: A unique ID assigned to a LUN on the storage side. Each path's dev node for a given LUN will share the same WWID.

Terms & Concepts (cont'd)

- Mpath Device: Common term for the virtual device that dm-multipath creates to point back to the underlying path dev nodes (e.g. /dev/mapper/mpath0)
- Path Group: dm-multipath organizes paths into path groups. Path groups allow dm-multipath to intelligently route I/O between multiple active paths.
- Path Checker: A simple operation that dm-multipath performs on all paths to determine their viability. For example, reading sector 0 on all paths.
- Queuing: In the event of a loss of all paths dm-multipath can be configured to queue I/O indefinitely

Software Stack – The multipath Command

- The multipath command is the central point in the dm-multipath stack for working with dm-multipath managed devices
- Does the “dirty work” of identifying paths back to the same LUN and abstracting them behind an mpath device by way of a device mapper map
- Can be run to report valuable information on the current status of various mpath devices and their underlying paths

Software Stack – The mulipathd daemon

- The multipathd daemon runs in the background and should be configured to start on boot on any system that requires dm-multipath
- On start, multipathd reads the multipath configuration file at /etc/multipath/multipath.conf and executes the proper multipath commands to set up the mpath devices in accordance with the config file
- multipathd then sleeps in the background waking up on an interval to run the path checker against the dm-multipath managed paths.
- multipathd will re-enable any failed paths that return to service if the path checker succeeds on them

Software Stack - The Multipath Configuration File

- The dm-multipath config file is located at /etc/multipath/multipath.conf
- multipath.conf can be used to control all of dm-multipath's behaviors including how paths are grouped, what path checker is used, and how mpath device should be named
- Examples and explanations of every directive available for multipath.conf can be seen at /usr/share/doc/device-mapper-multipath-<version>/multipath.conf.annotated
- The default values used for various storage devices can be seen at /usr/share/doc/device-mapper-multipath-<version>/multipath.conf.defaults

The Software Stack – The Bindings File

- The bindings file is where dm-multipath stores its device alias bindings for later use
- If you allow dm-multipath to name your devices for you it will populate the bindings file with the alias maps it decides on
- The purpose of the bindings file is to ensure that device aliases are persistent, i.e. that a given mpath device will always correspond to the same LUN even across reboots
- The bindings file comes with a few “gotchas.” It was located under /var in RHEL 5 but moved to /etc in RHEL 6. The bindings file should not be manipulated by-hand. Use multipath.conf to customize aliases.

The Software Stack – Device Mapper & SCSI

- dm-multipath provides its functionality by creating and manipulating Device Mapper maps.
- You can see the multipath device maps in dmsetup output.
- dm-multipath relies on the SCSI layer for identifying devices by WWID and determining if a path has failed
- dm-multipath has no “special” way to know if a path has failed. dm-multipath “knows” a path has failed if it receives a SCSI error on that path. If something is hiding or blocking SCSI errors you will not get failover!

The Software Stack – The dm-multipath Module

- The dm-multipath kernel module is a plugin for Device Mapper
- The dm-multipath module contains the multipath specific logic that enables Device Mapper to provide multipath functionality
- It handles the “dirty work” of re-routing the I/O, failing over between path groups, and other low level operations

The Software Stack - How It All Fits Together

- multipathd starts and reads multipath.conf
- multipathd reads the bindings file to to see what the device alias maps were the last time it ran
- multipathd runs the multipath commands required to comply with the config file directives and the bindings file maps
- multipath command sets up the Device Mapper maps
- multipathd sleeps; It wakes to check path viability on the interval and in the manner specified in multipath.conf
- SCSI errors cause the dm-multipath kernel module to re-route I/O to available paths.

Configuring dm-multipath: RHEL 5

- Ensure that the device-mapper-multipath RPM is installed
- Edit /etc/multipath.conf and comment-out the blacklist
- Start the multitpathd service and chkconfig it on
- Run multipath -ll and verify you have a multipath device

Configuring dm-multipath demo: RHEL 5

Configuring dm-multipath: RHEL 6

- RHEL 6 uses the mpathconf command to configure dm-multipath
- mpathconf will automatically create a multipath.conf
- mpathconf can also automatically chkconfig the multipathd service on, and start it

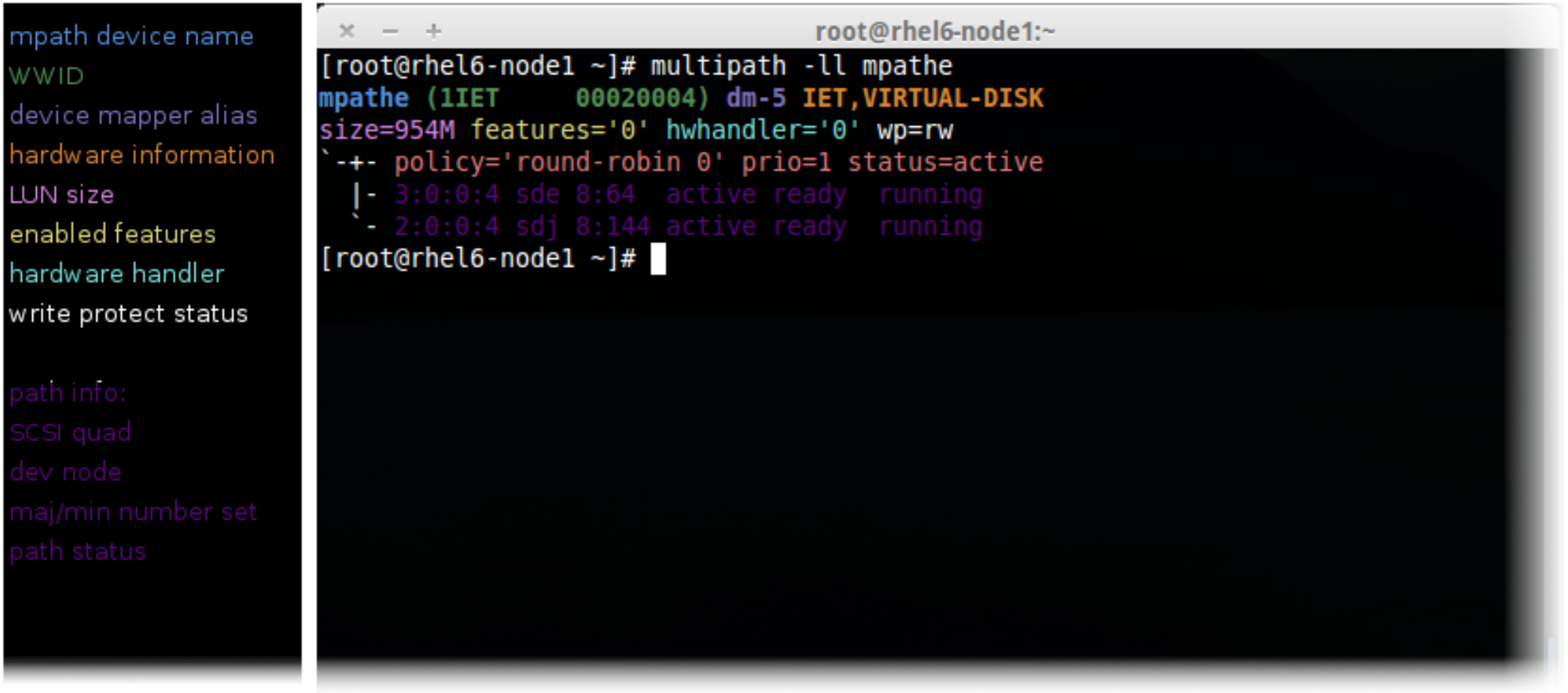
Configuring dm-multipath demo: RHEL 6

Differences in dm-multipath for RHEL 5 & RHEL 6

- RHEL 6 can use find_multipaths to more intelligently create mpath devices.
- RHEL 6 can more efficiently choose which paths to send IO down based on queue size or service time. RHEL 5 could only round-robin between paths.
- RHEL 6 sees its bindings file under /etc by default. Having the bindings file under /var caused problems for some folks in RHEL 5.
- The remote FC port device loss and SCSI I/O failure timeouts can be explicitly controlled by dm-multipath in RHEL 6 allowing for greater predictability and control for failover timing.

Getting in-depth: multipath command output

- multipath -ll provides valuable information on current multipath topology



Getting in-depth: Path Groups & Policies

- dm-multipath groups paths into path groups.
- Path group policies define the criteria by which dm-multipath groups paths:
  - Multibus : All paths in one path group
  - Failover : Each path in its own path group
  - Serial : Group by serial number
  - Prio : Group by path prio (determined by prio callout)
  - Node : Group by SCSI target number
- A single path group at a time is active
- Default is multibus

Getting in-depth: Path Selectors

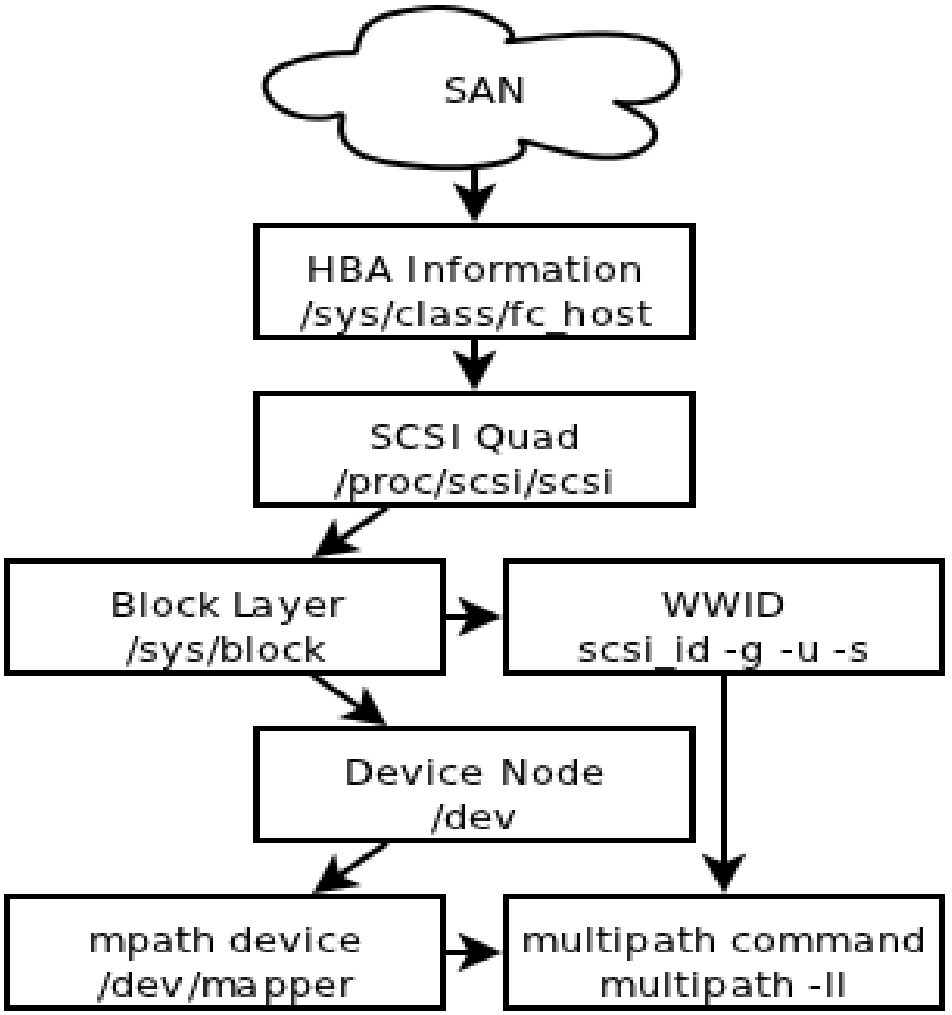
- dm-multipath determines which path within a path group to send IO down based on a path selector
- RHEL 6 has multiple path selectors, RHEL 5 only has round-robin
  - Round Robin: Loop through all paths in group
  - Queue Length: Select path with shallowest IO queue
  - Service Time: Select path with lowest estimated svctime
- Different path selectors can be configured for each multipath
- The default is round robin

Getting in-depth: Path Groups & Selectors Demo

Getting in-depth: Device Mapping

- Storage devices pass though multiple levels of abstraction between the hardware drivers and user space
- It is important to be able to clearly identify which device maps to which other device all the way through the stack so that in the event of a problem what is wrong can be clearly understood
- Understanding the device mapping and abstraction systems in use, and in what order they occur, is useful for “checking Device Mapper's work” and determining where in the stack a problem may be

Getting in-depth: Device Mapping (cont'd)



- The WWID and LUN # set on the SAN
- HBA info available in /sys/class/fc_host
- SCSI Quad H:B:T:L in /proc/scsi/scsi
- Double check WWID with /sys/block/
- Dev node under /dev
- multipath -ll output

Getting in-depth: Device Mapping example

Getting in-depth: Device Mapping example cont'd

```
mpath4 (3600a0b80001327510000bb14d11c49a) dm-2 IBM,1742-900
[size=8.0G][features=0][hwhandler=1 rdac][rw]
\_ round-robin 0 [prio=100][active]
 \_ 5:0:1:0 sdh 8:112 [active][ready]
\_ round-robin 0 [prio=0][enabled]
 \_ 5:0:0:0 sdb 8:16  [active][ghost]

[root@cs-rh5-1 ~]# ll /sys/class/fc_host
total 0
drwxr-xr-x 3 root root 0 Apr  1 14:00 host4
drwxr-xr-x 3 root root 0 Apr  1 14:00 host5

Host: scsi5 Channel: 00 Id: 00 Lun: 00
  Vendor: IBM      Model: 1742-900         Rev: 0914
  Type:   Direct-Access                    ANSI SCSI revision: 05
Host: scsi5 Channel: 00 Id: 01 Lun: 00
  Vendor: IBM      Model: 1742-900         Rev: 0914
  Type:   Direct-Access                    ANSI SCSI revision: 05

[root@cs-rh5-1 ~]# scsi_id -g -u -s /block/sdh
3600a0b80001327510000bb14d11c49a
[root@cs-rh5-1 ~]# scsi_id -g -u -s /block/sdb
3600a0b80001327510000bb14d11c49a
```


Troubleshooting Tips

- Check the basics first! Use a simple multipath.conf with nothing set other than the defaults and no black list. Make sure multipathd is chkconfigged on and running.
- Most suspected dm-multipath issues aren't dm-multipath issues at all! Check the storage stack all the way from the SAN through /proc/scsi/scsi, and the WWIDs. If /proc/scsi/scsi doesn't see your LUNs then dm-multipath wont either! WWIDs don't match? Then dm-multipath wont touch them!
- All LUNs in /proc/scsi/scsi, right # of dev nodes, path WWIDs match, and have a simple config? Check known issues. /var on a seperate filesystem from /?
- When in doubt reboot or rescan SCSI bus and restart mutlipathd.

Documentation

- Official Product Documentation: docs.redhat.com
  - RHEL 6 DM Multipath Guide
  - RHEL 5 DM Multipath Guide
- Customer Portal: access.redhat.com
  - Lots of solutions created directly from resolved support cases
  - Technical Briefs
  - User Groups
  - Videos
- /usr/share/doc/device-mapper-multipath-<version>/

Questions? Thanks!

- Thank you very much for working with Red Hat and attending this webinar. I hope it was interesting and proves useful!
- Need more help? Contact your TAM and Red Hat Support. We're here to help.