Native Linux Device-Mapper Multipath for HP StorageWorks Disk Arrays reference guide

Introduction

This guide is supplanting and successor of *HP Device-Mapper Multipath Enablement Kit*. In the past, *HP Device-Mapper Multipath Enablement Kit* had provided HP StorageWorks customers ease and promptness to adopt Linux open-source multipathing solution. HP has provided lots of technical feedback and enhancement requests to the community. Over time, all these changes are now well integrated and distributed by most of the latest major Linux releases. *HP Device-Mapper Multipath Enablement Kit* will not be updated after v4.4.1 but will be kept available at both external/internal SPOCK for download. Instead this document will be updated as necessary.

Overview

Device-Mapper (DM) is an infrastructure in the Linux kernel. It provides a generic way to create virtual layers of block devices. It supports striping, mirroring, snapshots, concatenation, and multipathing. The multipath feature is provided with combination of DM Multipath kernel modules and multipath-tools user-space package.

DM Multipath enables hosts to route I/O over the multiple paths available to an end storage unit (LUN). A path refers to the connection from an HBA port to a storage controller port. When an active path through which I/O happens fails, DM Multipath reroutes the I/O over other available paths. On a Linux host, when there are multiple paths to a storage controller, each path appears as a separate block device and hence results in multiple block devices for single LUN. DM Multipath creates a new Multipath block device for those devices that have the same LUN WWN.

For example, a host with two HBAs when attached to a storage controller with two ports through a single FC switch provides four block devices: `/dev/sda`, `/dev/sdb`, `/dev/sdc`, and `/dev/sdd`. DM Multipath creates a single block device, `/dev/mapper/mpath1`, that reroutes I/O through these four underlying block devices.

DM Multipath consists of following components:

- **dm-multipath kernel module** - Routes I/O and provides failover to paths and path groups.
- **multipath configuration tool** - Provides commands to configure, list, and flush Multipath devices.
- **multipathd daemon** - Monitors path status. When paths revert, `multipathd` daemon may also initiate path group switches to ensure that the optimal path group is used.
- **kpartx utility** - Reads partition tables on the specified device and creates device maps over the detected partitions. The **kpartx utility** is called from `hotplug` whenever device maps are created and deleted.
- **devmap-name** - Provides a meaningful device name to `udev` for device maps (devmaps).

Features

DM Multipath offers the following features:

- **I/O failover and failback**: Provides transparent failover and failback of I/Os by rerouting I/Os automatically to an alternative path when a path failure is sensed, and routing them back when the path is restored.
- **Path grouping policies**: Paths are coalesced based on the following path-grouping policies:
  - Group by prio – Paths with same priority are grouped together
  - Multibus – All paths are grouped under a single path group

    - Group by serial – Paths are grouped together based on controller serial number
    - Failover only – Provides failover without load balancing by grouping the paths into individual path groups
- **I/O load balancing policies**: Provides the following load balancing policies within a path group:
    - Weighted round robin – This round-robin algorithm routes `rr_min_io` number of I/Os on a selected path before switching to the next path.
    - Least pending I/O path – This determines the number of non-serviced requests pending on a path and selects the path which has the least number of pending requests for service.
    - DM service time – This is a service time oriented dynamic load balancer, which selects a path to complete the incoming I/O with the shortest time.
- **Device name persistence**: Device names are persistent across reboots and Storage Area Network (SAN) reconfigurations. DM also provides configurable device name aliasing feature for easier management.
- **Persistent device settings**: All the device settings such as load balancing policies, path grouping policies are persistent across reboots and SAN reconfigurations.
- **Device exclusion**: Provides device exclusion feature through blacklisting of devices.
- **Path monitoring**: Periodically monitors each path for status and enables faster failover and failback.
- **Online device addition and deletion**: Devices can be added to or deleted from DM Multipath without rebooting the server or disrupting other devices or applications.
- **Management Utility**: Provides Command Line Interface (CLI) to manage Multipath devices.
- **Boot from SAN**: Provides multipathing for operating system installation partitions on SAN devices.
- **Cluster support**: Provides multipathing in HP Serviceguard and SteelEye LifeKeeper clustering environment.
- **Volume Manager support**: Provides support for multipathing devices to be configured under Logical Volume Manager.

Note

1. Path grouping policy support is based on the HP StorageWorks array product type
2. At this time, Least pending I/O and DM service time are supported on SLES11 distribution only

Installing DM Multipath

For supported HP StorageWorks arrays, HP is supporting DM Multipath which is bundled with the OS distribution or patch release. In this and other document, sometimes this is referred as “native” or “inbox” DM Mulitpath.

To make DM Mulitpath available, ensure following RPMs are installed:

- For latest RHEL4, such as Update 7 or later
    - device-mapper
    - device-mapper-multipath
- For latest RHEL5, such as Update 2 or later
    - device-mapper
    - device-mapper-multipath
- For RHEL6, such as Update 1 or later
    - device-mapper

    - device-mapper-multipath
- For latest SLES10, such as ServicePack 3 or later
    - deivce-mapper
    - device-mapper-devel
    - multipath-tools
- For SLES11, or any later ServicePack
    - deivce-mapper
    - multipath-tools

Configuration file /etc/multipath.conf

Support for many devices is already built-in the user space component of DM Multipath. HP has been providing supported arrays profile or stanza to be included as soon as it's released.

On a fresh new install, RHEL system has a basic /etc/multipath.conf and SLES system does not have /etc/multipath.conf. To list all device types that are already supported by default, do following
Make sure multipathd is running
For RHEL

```
# multipathd –k
multipathd> show config
```

For SLES

```
# multipath –t
```

In order to customize number of DM Multipath features or to add support of HP devices which is not built-in, user needs to modify /etc/multipath.conf. It is advisable to include array which is already built-in as well. The file has 5 sections:

1. System defaults ("**defaults**") – define value for attributes to use whenever the attribute is not specified for a particular device
2. Black listed device ("**blacklist**") – specify list of devices to be excluded from DM Multipath control
3. Black list exception ("**blacklist_exceptions**") – specify devices to be under DM Multipath control despite being listed in blacklist
4. Storage controller settings ("**devices**") – define the device specific settings to be applied based on "Vendor" and "Product" values
5. Device specific setting ("**multipaths**") – to fine tune each individual LUN settings

Table 1 List of some attributes

Attribute	Description
`path_grouping_policy`	Used for applying the policy to the multipath device hosted by this storage controller
`path_checker`	Used for determining the state of the path
`path_selector`	Used to select the path selector algorithm to be used for mpath. These algorithms are offered by the kernel mpath target
`failback`	Used to manage the time during path group failback
`prio_callout`	Executable to obtain a path weight for a block device. Weights are

Attribute	Description
	summed for each path group to determine the next path group to be used in case of path failure
rr_weight	Used to assign weights to the path
rr_min_io	The number of IOs to route to a path before switching to the next in the same path group
no_path_retry	(n =18) indicates the number of retries until queuing is disabled (queues till n number of polling), ***fail*** indicates immediate failure (no queuing), or ***queue*** indicates never stop queuing (queue forever till the path comes alive)
getuid_callout	A standalone program that returns a globally unique identifier for a path. multipath/multipathd invokes this callout and uses the ID returned to coalesce multiple paths to a single multipath device

Note

Different Linux distribution may use different attribute names but the fundamental function of the attribute is same.

HP arrays parameter values

It is recommended to add the following HP array profiles to `/etc/multipath.conf` file under "**devices**" section and use these values:

For MSA2012fc/MSA2212fc/MSA2012i

```
device {
      vendor                  "HP"
      product                 "MSA2[02]12fc|MSA2012i"
      getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
      hardware_handler        "0"
      path_selector           "round-robin 0"
      path_grouping_policy    multibus
      failback                immediate
      rr_weight               uniform
      rr_min_io               100
      no_path_retry           18
      path_checker            tur
}
```

For EVA4x00/EVA6x00/EVA8x00/P6300/P6500

```
device {
      vendor                  “HP”
      product                 "HSV2[01]0|HSV3[046]0|HSV4[05]0"
      getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
      prio_callout            "/sbin/mpath_prio_alua /dev/%n"
      hardware_handler        "0"
      path_selector           "round-robin 0"
      path_grouping_policy    group_by_prio
      failback                immediate
      rr_weight               uniform
      rr_min_io               100
      no_path_retry           18
      path_checker            tur
```

```
}
```

For P2000 FC / P2000 FC/iSCSI / P2000 SAS

```
device {
      vendor                  “HP”
      product     "P2000 G3 FC|P2000G3 FC/iSCSI|P2000 G3 SAS|P2000 G3 iSCSI"
      getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
      prio_callout            "/sbin/mpath_prio_alua /dev/%n"
      hardware_handler        "0"
      path_selector           "round-robin 0"
      path_grouping_policy    group_by_prio
      failback                immediate
      rr_weight               uniform
      rr_min_io               100
      no_path_retry           18
      path_checker            tur
}
```

For MSA2012sa/MSA2312sa/MSA2324sa

```
device{
      vendor                  "HP"
      product                 "MSA2012sa|MSA2312sa|MSA2324sa"
      getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
      prio_callout            "/sbin/mpath_prio_alua /dev/%n"
      hardware_handler        "0"
      path_selector           "round-robin 0"
      path_grouping_policy    group_by_prio
      failback                immediate
      rr_weight               uniform
      rr_min_io               100
      no_path_retry           18
      path_checker            tur
}
```

For XP/P9500

```
device{
      vendor                  "HP"
      product                 "OPEN-.*"
      getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
      hardware_handler        "0"
      path_selector           "round-robin 0"
      path_grouping_policy    multibus
      failback                immediate
      rr_weight               uniform
      rr_min_io               1000
      no_path_retry           18
      path_checker            tur
}
```

For MSA2312fc/MSA2324fc/MSA2312i/MSA2324i

```
device{
      vendor                  "HP"
      product                 "MSA2312fc|MSA2324fc|MSA2312i|MSA2324i"
      getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
      prio_callout            "/sbin/mpath_prio_alua /dev/%n"
```

```
        hardware_handler        "0"
        path_selector           "round-robin 0"
        path_grouping_policy    group_by_prio
        failback                immediate
        rr_weight               uniform
        rr_min_io               100
        no_path_retry           18
        path_checker            tur
}
```

For SVSP

```
device{
        vendor                  "HP"
        product                 "HSVX700|HSVX740"
        getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
        prio_callout            "/sbin/mpath_prio_alua /dev/%n"
        hardware_handler        "1 alua"
        path_selector           "round-robin 0"
        path_grouping_policy    group_by_prio
        failback                immediate
        rr_weight               uniform
        rr_min_io               100
        no_path_retry           18
        path_checker            tur

}
```

Note

- For SLES 11, replace
  `getuid_callout "/sbin/scsi_id -g -u -s /block/%n"`
  with
  `getuid_callout "/lib/udev/scsi_id -g -u /dev/%n"`
- For SLES 10 SP2 or later and SLES 11, replace
  `prio_callout "/sbin/mpath_prio_alua %n"`
  with
  `prio alua`
- For RHEL 6 U1 or later, replace
  `getuid_callout "/sbin/scsi_id -g -u -s /block/%n"`
  with
  `getuid_callout "/lib/udev/scsi_id –-whitelisted –-device=/dev/%n"`
  and
  `prio_callout "/sbin/mpath_prio_alua %n"`
  with
  `prio alua`

- For RHEL 5 U3 or earlier, there is problem with native scsi_id tool to work correctly for MSA2012sa, MSA2312sa, and MSA2324sa through P700m SAS adapter in c-class server. In this situation, download HPDM kit v4.4.1 which will provide hp_scsi_id tool and replace
  `getuid_callout "/sbin/scsi_id -g -u -n -s /block/%n"`
  with
  `getuid_callout "/sbin/hp_scsi_id -g -u -s /block/%n"`

DM Multipath daemon

To enable DM Multipath daemon to start at boot time,

For RHEL hosts,

1. Run the following command to check if the daemon is configured to start at boot time:
   ```
   # chkconfig --list multipathd
   ```
2. Run the following commands to enable the Device Mapper Multipath daemon:
   ```
   # chkconfig multipathd on
   ```

For SLES hosts,

1. Run the following commands to check if the daemon is configured to start at boot time:
   ```
   # chkconfig --list boot.multipath
   # chkconfig --list multipathd
   ```
2. Run the following commands to enable the Device Mapper Multipath daemons:
   ```
   # chkconfig boot.multipath on
   # chkconfig multipathd on
   ```

To start DM Multipath daemon one time or while the system is running,

```
# /etc/init.d/multipathd start
```

Additional setup

These are recommended HBA and iSCSI parameters for use in DM Multipath environment

Configuring for Qlogic FC HBA

1. Edit the `/etc/modprobe.conf` file in RHEL hosts or `/etc/modprobe.conf.local` file in SLES hosts with the following values:
   ```
   options qla2xxx ql2xmaxqdepth=16 qlport_down_retry=10
   ql2xloginretrycount=30
   ```
2. Rebuild the initrd by performing the following:
   a. Backup the existing initrd image by executing the following command:
      ```
      #mv /boot/initrd-<version no.>.img /boot/initrd-<version
      no.>.img.old
      ```
   b. Make a new initrd image by executing the following command:
      - For SLES 10/SLES 11:
        ```
        #mkinitrd -k <kernal> -i <initrd>
        ```
      - For other operating systems:
        ```
        #mkinitrd /boot/initrd-<version no.>.img `uname -r
        ```
   c. Edit the value for default parameter in /boot/grub/menu.lst file to boot with the new initrd image.

Configuring for Emulex FC HBA

1. Edit the `/etc/modprobe.conf` file in RHEL hosts or `/etc/modprobe.conf.local` file in SLES hosts with the following values:
   ```
   options lpfc lpfc_nodev_tmo=14 lpfc_lun_queue_depth=16
   lpfc_discovery_threads=32
   ```
2. Rebuild the initrd as above

Configuring for Brocade FC HBA

It is recommended to set the timeout value to 14 seconds as follows:

For driver version 1.x.x.x

```
# bcu fcpim --mpiomode <port_ID> off 14
```

For driver version 2.x.x.x
```
# bcu fcpim –pathtov port_ID 14
```

Configuring for mptsas (SC08Ge) adapter

1. Edit the `/etc/modprobe.conf` file in RHEL hosts or `/etc/modprobe.conf.local` file in SLES hosts with the following values:
   ```
   options mptsas mpt_cmd_retry_count=10 mpt_disable_hotplug_remove=1
   ```
2. Rebuild the initrd as above

Configuring for software iSCSI initiator

1. Update the iSCSI configuration file
   - In RHEL 5, SLES 10, and SLES 11 hosts, edit the `/etc/iscsi/iscsid.conf` file with the following value:
     ```
     node.session.timeo.replacement_timeout=15
     node.startup=automatic
     ```
   - In RHEL 4 hosts, edit the `/etc/iscsi.conf` file with the following value:
     ```
     ConnFailTimeout=15
     ```
2. Restart the iSCSI service by executing the following command:
   - In RHEL 4/RHEL 5 hosts,
     ```
     #/etc/init.d/iscsi restart
     ```
   - In SLES 10/SLES 11 hosts,
     ```
     #/etc/init.d/open-iscsi restart
     ```

Basic Operation

For example, a Linux host with a dual port HBA is connected to 2 SAN fabrics with XP24000 (2 array ports on each fabric) and EVA8000 (2 array ports on each fabric). In this case, if all the paths are available, the host has four I/O paths for any LUN presented from either array.

To ensure DM Multipath daemon is running, execute
```
#/etc/init.d/multipathd status
```

To get detail listing of multipath devices
```
# multipath –ll
```

```
36001438002a56fd60000600001c60000 dm-250 HP,HSV450
[size=5.0G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=10][enabled]
 \_ 2:0:5:4   sdck 69:128  [active][ready]
 \_ 5:0:5:4   sdhz 134:128 [active][ready]
\_ round-robin 0 [prio=50][enabled]
 \_ 2:0:0:4   sdz  69:14   [active][ready]
 \_ 5:0:0:4   sdhk 134:44  [active][ready]
360060e801439ba00000139ba00002209 dm-62 HP,OPEN-V
[size=8.0G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=2][active]
 \_ 5:0:4:9   sdlv 128:27  [active][ready]
 \_ 2:0:2:9   sdaq 134:0   [active][ready]
 \_ 5:0:7:9   sdnv 128:272 [active][ready]
 \_ 2:0:11:9  sdhq 134:0   [active][ready]
```

The output is presented by grouping paths for a LUN with unique identifiers, such as UID/WWN. Vendor, product ID, size, features, and corresponding hwhandler are displayed following the unique LUN identifier.

Then paths are grouped based on the path grouping policy. In above example, for the XP LUN, paths are grouped with multibus policy and devices sdlv, sdaq, sdnv, sdhq belong to the same path group. For the EVA LUN, the grouping is done with group_by_prio policy. Devices sdck, sdhz, belong to one path group, and devices sdz, sdhk belong to a different path group. This grouping is based on one path group per path priority value. Path priority value is determined by ALUA state of the path. I/O always happens in the path group with higher priority. If all paths in the active group fail, failover occurs to the other path group. When paths are up again and the failback parameter is set as immediate, failback occurs to the earlier group and I/O occurs through the earlier group.

The state of the path is given as [active][ready] if the path is up, and ready for I/O. If the path is down, this state is shown as [failed][faulty]. The path states are updated by the multipathd daemon periodically based on the polling interval set in the /etc/multipath.conf file.

To update the multipath maps in the kernel after a new LUN is added or deleted at the Linux host:

1. `/etc/init.d/multipathd restart`
2. `multipath -v`

Command	Description
`# multipath –F`	Deletes all DM Multipath devices.
`# multipath –d`	Displays potential paths, but does not create any device.
`# multipath`	Creates DM Multipath devices.
`# multipath -l` `# multipath –ll`	Displays the list of device status. Displays the detailed list of device status.
`# multipath -v0`	Configures multipath map information

Notables

- For RHEL6 system using software iSCSI initiator, system will fail to reboot or shutdown if dm-multipath is being used. In /etc/multipath.conf "**defaults**" section, add this line
  `queue_without_daemon no`
- For RHEL5 update 6, RedHat is aware of a bindings files out-of-sync issue which if the server's boot lun is multipath, device mappings may get confused. After any lun presentation change (ie. addition, removal, or order change), make sure execute following command so that bindings copy in initrd image is in-sync with copy in /var/lib/multipath
  `# cp /boot/<your current initrd image> /boot/initrd.bak`
  `# mkinitrd –f /boot/<your current initrd image> <kernel version>`
- By default, RHEL5 and RHEL6 `/etc/multipath.conf` blacklist all devices. So make sure to make change appropriately in "**blacklist**" section to enable multipath to device you desire.
- In some cases if SELinux is enabled, device maps may not be created and labeling problem can cause SELinux denies access requested by `/sbin/multipathd`. Try to restore the default system file context for bin by running the following commends: `#restorecon -v bin`

If this does not work, there is currently no automatic way to allow this access. Instead, you can generate a local policy module to allow this access. For more information, see the following website: http://fedora.redhat.com/docs/selinux-faq-fc5/#id2961385
You can also disable SELinux protection altogether. However, disabling SELinux protection is not recommended.

- If an existing LUN is deleted and a new LUN is presented back in the same SCSI slot, LUN collision may occur. This results in the creation of a new LUN through old device special files. This may lead to data corruption. To avoid this situation, perform the following steps:
  1. `multipath –f <device>`
  2. `echo "- - -" > /sys/class/scsi_host/<host instance>/scan`
  3. `/etc/init.d/multipathd restart`
  4. `/sbin/multipath –v0`
- multipath commands may take longer time to execute on heavily loaded servers or under path failure conditions.
- Blacklisting the multipath device in the `/etc/multipath.conf` file and restarting the multipath service may not remove the device on RHEL 4 distributions. Execute the following command to remove the blacklisted device: `# multipath -f <device>`
- Using fdisk command to create partitions may fail to create Multipath device for the partition device. It is recommended to use parted command to create partitions for the device.
- `multipath -l` command may not reflect the correct path status for Logical Units presented from MSA2xxxsa array when paths fail or are restored under heavy load conditions. To refresh the path status, execute the `# multipath -v0` command.
- multipathd daemon crashes on systems configured with device paths more than the system open file limits (default system open file limit =1024). It is recommended to change the system open file limits by using either the 'max_fds' parameter in the /etc/multipath.conf file or by using the ulimit -n command and restart the multipathd demon.
- Multipath devices may not be created for logical units when the system disks or internal controllers are cciss devices. It is recommended to blacklist these non multipath devices in the /etc/multipath.conf file and restart the multipathd daemon.
- For LUNs greater than 2TB in RHEL 4 operating systems, DM multipath devices may not be created with appropriate size.
- On RHEL 4 operating systems with large number of iSCSI devices, not all multipath devices may get created after a reboot. It is recommended to increase the ESTABLISHTIMEOUT value in the /etc/sysconfig/iscsi file depending on the number of LUNs, or run the multipath -v0 command after the reboot.