Configuring Linux to Enable Multipath I/O

Storage is an essential data center component, and storage area networks can provide an excellent way to help ensure high availability and load balancing over multiple redundant data paths. To take advantage of these benefits in Linux® OS environments, enterprise IT organizations can use applications to set up multipath I/O configurations.

BY TESFAMARIAM MICHAEL, REZWANUL KABIR, JOSHUA GILES, AND JOHN HULL

Related Categories:

Fibre Channel switches

Storage

Storage area network (SAN)

Visit www.dell.com/powersolutions for the complete category index.

In the data center environment, to minimize downtime and service disruptions, IT departments must avoid single points of failure in any highly available system. For storage area networks (SANs), administrators can set up multiple redundant data paths (multipaths) between servers and storage systems to help avoid interruptions in data flow should a hardware failure occur.

To manage a multipath I/O configuration, administrators should ensure that the server OS supports multipath I/O and is configured properly to access data from the storage system and fail over to secondary data paths when necessary. For Linux operating systems, two multipath I/O applications are available: device mapper multipath and EMC® PowerPath® software. This article provides an overview of each application and highlights the advantages and disadvantages of each.

Understanding the basics of multipath I/O

A typical highly available SAN configuration may include a Dell™ PowerEdge™ server containing several host bus adapters (HBAs), two Fibre Channel switches, and a Dell/EMC CX series storage array, as shown in Figure 1; a cluster configuration would include multiple PowerEdge servers. As the figure shows, multiple data paths are configured between the server and the storage system to provide the necessary redundancy. In such a configuration—for example, a PowerEdge server running Red Hat® Enterprise Linux 4—a logical unit (LUN) on the CX storage that is assigned to the server is detected as many times as there are paths available. When an HBA driver loads, the SCSI midlayer initiates a scan of its bus and detects all assigned storage LUNs through every available path. Accordingly, that many SCSI disk devices are registered by the OS. In Figure 1, there are four paths to the storage system, so a LUN assigned to the attached server is detected four times, and four SCSI disk devices are detected by the HBA driver and registered with the server OS.

Despite the benefits of redundant paths, there are challenges to consider. These challenges include identifying a particular device for I/O and managing multiple devices of the same physical device.

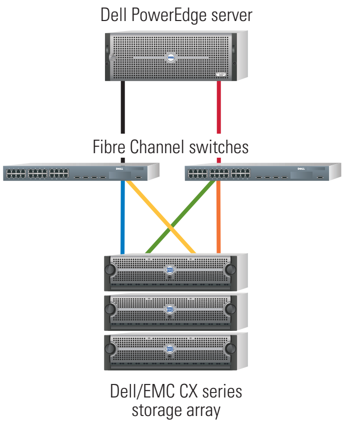


Figure 1. Basic highly available SAN configuration

Different storage systems manage paths to a particular LUN in different ways. Some systems provide an *isotropic* or *symmetric* view of the paths, where all paths are treated as equal. In these cases, all paths are active, and I/Os can be directed to any of them. Other storage systems, such as Dell/EMC CX series Fibre Channel systems, implement *asymmetric* arrays. In this case, paths to the same LUN are divided into active/passive groups, limiting the number of accessible devices at any given time by half.

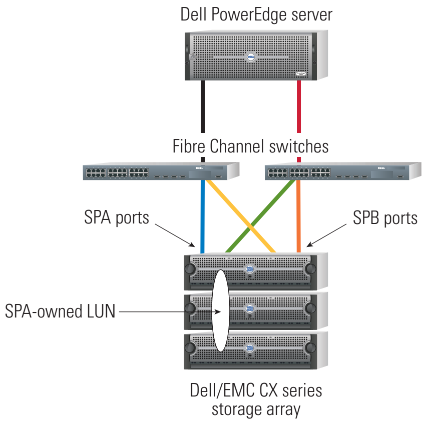
Active/passive cluster formations allow only one storage processor at a time to be actively performing I/Os to its assigned LUNs. The processors in Dell/EMC storage systems are grouped as storage processor A (SPA) and storage processor B (SPB). A particular port in these systems is associated with only one of these storage processors, and a LUN can be owned by only one of these processors at any given time. Default LUN ownership is specified during its creation. When a failure occurs, the ownership of a LUN can be changed to the other storage processor; this process is known as LUN trespassing. LUN trespassing is achieved by sending a device-dependent trespass command to the storage system. All asymmetric arrays require special hardware handlers to implement this mechanism to either fail over or fail back.

From the server side, a device is referred to as active if its path can be traced to a port owned by the storage processor that owns the LUN. A passive device's path to the LUN can be traced to a port not owned by the storage processor that owns the LUN. To differentiate active devices from passive devices, administrators can issue the command `sfdisk -l /dev/sdX`, where `/dev/sdX` is the SCSI disk device. Devices that do not return an I/O error are active, and those that do return an I/O error are passive. Under the current implementation of multipath I/O in Dell/EMC storage systems, Linux can use only active devices for I/O; passive devices can be used only when the LUN trespasses to the other storage processor. Once a LUN trespasses, the passive devices become active and vice versa. I/Os that were issued before the trespass but did not complete, along with all future I/Os, are then redirected to the devices that just became active.

When the Linux multipath application tools are used, a physical LUN on the storage system that is registered multiple times to the server OS—including active and passive devices—is bound into a single device, providing applications on the server side with a single point to perform I/Os. For instance, for the server shown in Figure 2, if the internal drives are combined in one logical drive in a RAID configuration, Linux would register it as /dev/sda. A LUN on the storage system that is assigned to the attached server would be detected four times by the HBA driver because there are four paths to it. These four devices are registered as /dev/sdb, /dev/sdc, /dev/sdd, and /dev/sde. In this case, because the LUN is owned by SPA, /dev/sdb and /dev/sdd are active and the other two devices are passive. Thus, only /dev/sdb and /dev/sdd can be used for I/O. Both /dev/sdc and /dev/sde become usable only if the LUN trespasses to SPB.

On the server, because the LUN has two active devices associated with it, administrators could mistakenly try to use them as two different devices by mounting these two active devices separately, which can allow data corruption or loss. For instance, if /dev/sdb is used for some data (after partitioning, creating a file system, and mounting it), and then /dev/sdd is later formatted with a file system treating it as a free device, all data will be wiped out in the LUN.

To help avoid this type of situation, administrators should use Linux multipath I/O applications. A multipath application provides a single of point of access by binding the four devices into a single



Block device	DM multipath device	Data path	Mode
/dev/sdb	/dev/dm1	Black, blue	Active
/dev/sdc		Black, yellow	Passive
/dev/sdd		Red, green	Active
/dev/sde		Red, orange	Passive

Figure 2. Basic highly available SAN configuration with a LUN owned by SPA

device, which can be partitioned, formatted, and mounted. This single device can then be used to distribute I/Os onto all the underlying active devices using a given set of algorithms.

Using device mapper for multipath I/O

Native Linux multipath I/O support has been added to the Linux 2.6 kernel tree with the release of 2.6.13, and has been backported into Red Hat Enterprise Linux 4 in Update 2 and into Novell® SUSE® Linux Enterprise Server 9 in Service Pack 2. It relies on device mapper (DM), a tool for mapping block devices that provides logical volume management, software RAID, and multipath functionality. Combining DM with the multipath user-space application can help create a native Linux multipath I/O configuration.

The overall architecture for DM multipath support in Linux is flexible and modular. DM multipath has a convenient plug-in design that allows administrators to enhance functionality by plugging in a module that achieves the desired result. For example, the DM multipath module has two hooks built into it: path selector and hw handler. The path selector hook is used to determine how I/Os should be distributed among various available paths, and the hw handler hook is used to take hardware-specific actions (for example, LUN trespassing).

Because of this modular architecture, administrators can implement a path selection algorithm (currently only a round-robin algorithm is supported) and register it with the path selector hook to use that particular algorithm to select paths. Similarly, administrators can implement a hardware-specific handler (for example, dm_emc) and register it with the hw handler hook of the DM multipath module to allow hardware-specific actions. For example, Dell/EMC CX series systems require the dm_emc handler to perform LUN trespassing for failover or failback.

In addition to these packages, some DM kernel modules, such as dm_multipath, dm_round_robin, and dm_emc, are also required. DM includes a user-space configuration tool (dmsetup) and a library (libdevmapper). DM multipath support also includes a multipath configuration file (multipath.conf), an init script (multipathd), udev rules, a device map creation tool from partitions (kpartx), and a multipath executable binary, among others. Udev is a recent Linux user-space application that manages devices (/dev/ directory) dynamically.

When DM multipath starts, it retrieves the universally unique identifier (UUID) of all the block devices in /proc/partitions (except those excluded in its configuration file) by issuing the `scsi_id -eg -s /block/sdX` command. It then groups all the block devices with the same UUID and creates a single device for them in /dev/mapper/. When this device is created, it can be partitioned with `fdisk` or `parted`. The partitions can be registered in /dev/mapper/ using `kpartx`, formatted with a file system, and mounted for usage.

DM multipath uses round-robin algorithms to balance I/Os across all active paths. If it experiences a failure when performing I/Os on the active devices because of a path disconnection, the DM kernel module (dm_emc in the case of Dell/EMC CX series systems) issues a trespass command (`switch-over`) to the system to switch over ownership of the LUN. Until the LUN trespasses successfully, all I/Os are queued. Once the trespass is successful, the passive devices become active and the active devices become passive, and DM multipath shifts I/Os (including those queued) to the new active devices.

Setting up the multipath configuration

To set up a multipath I/O configuration, administrators must first gather the UUIDs of the block devices. As mentioned earlier, the `scsi_id` command can be used to obtain the UUID of a block device. The default device naming can be changed by specifying aliases to UUIDs. These aliases as well as other settings for the multipath I/O configuration are set in the configuration file. The following steps describe how to configure systems for multipath I/O; these steps use the sample configuration shown in Figure 2.

The block devices have the same UUID, because they are all devices for one physical LUN. Administrators can issue the following commands to obtain the UUID for the four block devices in Figure 2:

```
scsi_id -g -s /block/sdb
scsi_id -g -s /block/sdc
scsi_id -g -s /block/sdd
scsi_id -g -s /block/sde
```

The output of all four commands is the same: a long hexadecimal number. A multipath configuration file has four sections: `devnode_blacklist`, `defaults`, `multipaths`, and `devices`. Visit *Dell Power Solutions* online at www.dell.com/powersolutions to see the sample multipath configuration file referred to in this article.

The `devnode_blacklist` section lists devices to be excluded from the multipath, which thus will not be probed for UUIDs. In the sample file online, all IDE devices (/dev/hd[a-z]) are excluded. When DM multipath starts, it will not issue any commands to these devices.

The `defaults` section assigns the default values to the specified multipath parameters. In the sample multipath configuration file online, these parameters include `multipath_tool`, which passes any argument to the `multipath` command; `polling_interval`, which dictates how often the devices should be pinged; and `default_selector`, which specifies the algorithms. Note that `default_hwhandler` should be set to `1 emc` to load the dm_emc module and issue all the necessary commands, including the trespass command, to the Dell/EMC CX series systems.

```
fdisk /dev/mapper/dm1
kpartx -l /dev/mapper/dm1       # lists all partitions on this device
kpartx -a /dev/mapper/dm1       # adds all partitions on this device in /dev/mapper/
```

Figure 3. Commands to create and add partitions

```
mke2fs -j /dev/mapper/dm1p1     # creates a file system
mkdir /data
mount /dev/mapper/dm1p1  /data  # mounts device on /data
df -h /data                     # displays device properties
```

Figure 4. Commands to create a file system on partitions and mount a device

The multipaths section embeds as many entries of the multipath stanza as there are available LUNs assigned to the server. The internal multipath stanza specifies the UUID (or wwid, as shown in the sample figure online) and the alias to the LUN. This figure includes only one multipath entry, which sets the wwid, alias, and path_checker (to check the path regularly) variables. It is important that these variables are set. For the wwid value shown in the sample multipath configuration file online, an alias device dm1 is created in /dev/mapper/ when DM multipath starts.

The devices section, similar to the multipaths section, also embeds the device stanza. In an environment with multiple SAN storage systems, several device entries are necessary. This internal stanza shows vendor-specific SAN settings.

After setting the multipath configuration file, administrators can start DM multipath by issuing the multipath command. To generate detailed return messages, they can issue the command as multipath -v3 -ll. This command displays useful information such as the size of the LUN, the alias, a list of active and passive devices, and other settings. It also displays the alias devices created (/dev/mapper/dm1 in the sample multipath configuration file online).

Next, administrators should create a partition on /dev/mapper/dm1 using fdisk or parted and add the partitions to /dev/mapper/ using the commands shown in Figure 3.

Finally, administrators should create the file system on the partitions (/dev/mapper/dm1p1) and mount the device using the commands shown in Figure 4.

The LUN can then be accessed using the /data mounting point, and data can be read from and written to it. To verify that the LUN can be used as expected, administrators should perform some I/O activities by copying files to /data. At the same time, they should confirm the I/O activity by issuing the command iostat -d 1.

Using EMC PowerPath for multipath I/O

EMC PowerPath provides similar functionality as DM multipath, but also includes features not present in DM multipath, such as a variety of algorithms (including round-robin), the ability to set priority for its devices, and the ability to report current configurations.

PowerPath for Linux is packaged in the Red Hat Package Manager (RPM™) format. EMC releases new or updated versions regularly. Usually a particular version supports a specific Linux release, such as Red Hat Enterprise Linux or SUSE Linux. The package can be downloaded from the EMC Web site at www.emc.com.

Once downloaded to the server, the package can be installed using the rpm -ivh command. For example, if EMCpower.LINUX-4.4.0-337.rhel.i386.rpm is downloaded for use on a 32-bit Intel architecture (IA-32) system running Red Hat Enterprise Linux 4, administrators can install this package by issuing the following command (the majority of the package's files are copied to /etc/opt/emcpower):

```
rpm -ivh EMCpower.LINUX-4.4.0-337.rhel.i386.rpm
```

PowerPath includes powermt, a powerful management utility for its devices. Its man page (man powermt) provides specific information about the utility. Among other features, powermt allows administrators to display the current settings; set priority, policy (algorithms), and mode; remove a particular HBA or device; and restore a removed HBA or device.

In addition, PowerPath comes with its own init script and can be started and stopped from the command line. When stopping PowerPath, administrators should be sure that there is no I/O activity—that is, PowerPath should not be in use by any application. After installation, administrators can start PowerPath by issuing the command service PowerPath start.

As with DM multipath, once started PowerPath gathers the UUIDs of the block devices and bundles the devices with the same UUID into a single device, /dev/emcpower*X*. However, it does not use a configuration file. As PowerPath identifies the LUNs, it enumerates them as /dev/emcpowera, /dev/emcpowerb, and so on. Because PowerPath relies on how the HBA driver has detected the LUN and created the block devices, and does not use an administrator-supplied configuration file, its enumeration of LUNs can vary from one node to the next in clustered servers. For the configuration in Figure 2, /dev/sdb, /dev/sdc, /dev/sdd, and /dev/sde are all bundled to /dev/emcpowera. This device can be partitioned, formatted with a file system, and mounted using the commands shown in Figure 5.

```
fdisk /dev/emcpowera            # partitions the device
mke2fs -j /dev/emcpowera1       # formats with ext3 file system
mkdir /data
mount /dev/emcpowera1 /data     # mounts the partition
df -h
```

Figure 5. Commands to partition, format, and mount a device

To stop PowerPath, administrators should first confirm that all of the /dev/emcpower*X* devices are not in use (that is, they must stop all I/Os to the devices and unmount them). They can then issue the command `service PowerPath stop`.

Comparing DM multipath with EMC PowerPath

When considering which multipath application to deploy, IT departments must take into consideration the features, level of manageability, and type of support. Given that DM multipath is relatively new, PowerPath is much more feature rich. For example, DM multipath provides only round-robin algorithms, but PowerPath provides nine different policies, including round-robin, adaptive, and basic failover. PowerPath also supports dynamic load balancing, automatic path failover, and online recovery. In addition, PowerPath allows administrators to set different priority levels for its devices, benefiting applications that use the devices with higher priorities.

> Although DM multipath is relatively new compared with PowerPath, it has solid backing in the Linux community and is expected to develop into an even stronger alternative to PowerPath in the future.

For manageability, PowerPath has an advantage in heterogeneous OS environments, because it is supported on the Microsoft® Windows®, Linux, UNIX®, and Novell NetWare® operating systems. DM multipath is available only on Linux and has relatively immature management support. However, DM multipath does allow for a consistent mapping of devices to LUNs in a cluster environment, which PowerPath does not.

Support for PowerPath is limited to the specific Linux operating systems supported by EMC, which typically includes only Red Hat Enterprise Linux and SUSE Linux. Because PowerPath is proprietary software, administrators must be running both a supported OS and a supported kernel to have PowerPath support, which can be inconvenient because of the large number of Linux distributions unsupported by PowerPath. Also, when new kernels are released by Linux vendors, there may be a lag between the kernel release and PowerPath support for that release. DM multipath does not have these limitations, because of its GNU General Public License and inclusion with most Linux distributions. Any new kernel released by a Linux vendor includes DM multipath support by default.

Choosing the appropriate multipath I/O application

Linux device mapper multipath and EMC PowerPath both provide viable and robust multipath I/O capability for Linux operating systems on Dell PowerEdge servers and Dell/EMC storage systems. Choosing the appropriate application depends on the specific data center environment and the necessary features and support. Although DM multipath is relatively new compared with PowerPath, it has solid backing in the Linux community and is expected to develop into an even stronger alternative to PowerPath in the future.

Tesfamariam Michael is a software engineer in the Dell Database and Application Engineering Department of the Dell Product Group. Tesfamariam has a B.S. in Electrical Engineering from the Georgia Institute of Technology, and a B.S. in Mathematics and an M.S. in Computer Science from Clark Atlanta University.

Rezwanul Kabir is a systems engineer in the Dell Linux Development Group. He has a B.S. in Computer Science and Engineering from Bangladesh University of Engineering and Technology and an M.S. in Computer Science from New Mexico State University.

Joshua Giles is a software engineer at Red Hat. His interests include operating systems, grammars, automata-based programming, and support vector machines (machine learning). Joshua has a B.S. in Computer Science from the New Mexico Institute of Mining and Technology.

John Hull is the manager of the Linux OS Development team at Dell. He has a B.S. in Mechanical Engineering from the University of Pennsylvania and an M.S. in Mechanical Engineering from the Massachusetts Institute of Technology.

FOR MORE INFORMATION

EMC PowerPath:
software.emc.com/products/software_az/powerpath.htm