HP-UX 11i v3 Native Multi-Pathing for Mass Storage

Abstract

This document describes the Native Multi-Pathing feature offered by the mass storage subsystem on HP-UX 11i v3. It is intended for system administrators or operators.

For a comprehensive description of all the features of the HP-UX 11i v3 mass storage subsystem refer to *The Next Generation Mass Storage Stack HP-UX 11i v3* white paper.

Introduction

HP-UX 11i v3 has re-architected the mass storage subsystem to significantly enhance scalability, performance, availability, manageability, and serviceability in a SAN environment. The re-architected subsystem increases maximum configuration limits to address very large SAN configurations while delivering performance and availability with automatic Native Multi-pathing, parallel I/O scan, optimized I/O forwarding and CPU locality.

Multi-pathing is native to HP-UX 11i v3. It is built in to the mass storage subsystem and it is available to applications without any special configuration. Pre-11i v3 multi-pathing add-on products are no longer necessary on HP-UX 11i v3.

Native multi-pathing offers the following features:

- optimal distribution of I/O traffic across lunpaths to LUNs,
- dynamic discovery of lunpaths and LUNs,
- automatic monitoring of lunpaths,
- automatic lunpath failover and recovery,
- intelligent I/O retry algorithms to deal with failed lunpaths,
- lunpath authentication to avoid data corruption.

This paper presents an overview of native multi-pathing and describes its principal benefits and features. It also provides details on managing the solution.

Publication History

- February 2007: Initial Publication
- September 2007: Addition of load balancing policies closest_path and pref_tport, and support of T10 Asymmetric Logical Unit Access (ALUA), coincident with the September 2007 release of HP-UX 11i v3
- March 2008: Addition of weighted_rr load balancing policy, automatic lunpath recovery configuration policies, and expanded examples on multi-pathing with legacy DSFs
- September 2009: Addition of the lpt_to_lockdown attribute for the path_lockdown load balancing policy

Terms and Definitions

Agile Addressing	The ability to address a logical unit with the same DSF regardless of the location of the LUN, that is, the device file for a LUN remains the same even if the LUN is moved from one Host Bus Adapter (HBA) to another, from one switch/hub port to another or presented via a different target port to the host. The LUN device file stays the same even if the N-Port ID of the target port changes in the case of Fibre Channel LUNs.
Agile Naming Model	DSF naming model introduced in 11i v3 to support agile addressing.
Agile View of The I/O Tree	View of the I/O tree using agile addressing to represent elements of the I/O tree.
DSF	Device Special File.
EVM	Event Management System.
Hard Partition	Feature part of the HP Partitioning Continuum for HP-UX 11i on HP 9000 and HP Integrity servers. (Complete documentation available at http://www.hp.com).
Hardware Path (H/W Path)	A series of numbers representing the physical or virtualized location of a device. Before 11i v3, the hardware path contained a maximum of 14 elements of 8 bits each. Starting with 11i v3, the hardware path contains up to 64 elements of 64 bits each.
HBA	Host Bus Adapter.
(Legacy) H/W Path	H/W Path in the legacy format (pre HP-UX 11i v3). Used for all components, mass storage or not.
I_T nexus	Defined by the T10 SAM standard as a nexus between a SCSI initiator port and a SCSI target port.
I_T_L nexus	Defined by the T10 SAM standard as a nexus between a SCSI initiator port, a SCSI target port, and a LUN.
Legacy or agile I/O Tree	The I/O Tree is the HP-UX representation of physical and virtual devices discovered by the operating system. The pre-11i v3 I/O tree view is present on HP-UX 11i v3 for backward compatibility reasons. The legacy I/O tree is limited to the pre-11i v3 I/O tree view. The agile I/O tree is limited to the agile view.
Legacy Naming Model / Legacy Format	DSF format convention used prior to HP-UX 11i v3. This model is maintained in HP-UX 11i v3 for backward compatibility purpose.
Legacy DSF	A lunpath dependent DSF that follows the legacy naming model conventions, wherein the DSF embeds the bus/target/lun/option for a specific lunpath to a mass storage device (Ex. /dev/dsk/c#t#d#).

LUN	Logical unit that refers to an end storage device such as disk, tape, floppy, cdrom, changer, and so forth. This is the logical unit itself and does not represent the path to the logical unit.
LUN Hardware Path	Hardware path to the LUN in the agile view of the I/O Tree. It has the form `64000/0xfa00/0x*`
LUN id	Equivalent to the logical unit number defined by the T10 SAM standard.
Lunpath	Path to a LUN. It is also known as an I_T_L nexus.
Lunpath Hardware Path	The hardware path of a Lunpath. In the agile view of the I/O tree, the lunpath hardware path format is: `<hba_path>.<Target port WWN or Target Id>.<0xLUN id>`
Path	An I_T or I_T_L nexus as defined in the T10 SAM standard.
Persistent DSF	DSF following the agile naming model conventions. Does not contain any encoding of bus, target identifier or device specific option information.
SAM	T10 SCSI Architecture Model standard (available at http://www.t10.org)
SCSI Class Driver	An HP-UX device driver which manages one or more specific classes of mass storage devices. For example: Disk Driver, Tape Driver, Changer Driver, Pass-Through Driver.
SCSI Interface Driver	An HP-UX device driver which manages one or more specific types of Host Bus Adapters. For example: Tachlite FC Driver, Qlogic FC Driver.
SPC-3	T10 "*SCSI Primary Commands – 3*" (T10 SPC-3) standard (available at http://www.t10.org)
Target Id	Target port identifier as defined in SCSI transport protocols.
Target Path	Path to a target port. It is also known as an I_T nexus.
Target Path Hardware Path	The hardware path of a Target Path. In the agile view of the I/O tree, the target path hardware path has the following format: `<hba_path>.<Target port WWN or Target Id>`
VPD	Vital Product Data – term defined by the T10 SPC-3 standard (section 7.6) (available at http://www.t10.org).
WWID	SCSI Logical Unit World Wide Identifier as defined by the SCSI standard and obtained from VPD INQUIRY Page 83h.
WWN	Worldwide Name as defined in SCSI transport protocols. It is an identifier that is worldwide unique.

Native Multi-Pathing Features and Benefits Overview

Native Multi-Pathing Features Overview

Multi-pathing is the ability to manage the various paths to a LUN device. A path to a LUN is also known as an I_T_L nexus. It corresponds to the route taken by an I/O request issued from an HP-UX host to a SCSI device.

Here is an overview of the features that native multi-pathing encompasses. For more details refer to the Native Multi-Pathing Features section below.

- **LUN WWID and DSF** - The various paths to LUN devices are correlated based on a world-wide unique identifier (WWID) that is used to create a persistent DSF offering agile addressing.

- **Dynamic I/O load-distribution** - The I/O load to a LUN is optimally and transparently distributed across the available lunpaths.

- **High availability of SCSI devices** – To optimize access to LUNs, the mass storage stack transparently performs lunpaths error detection, recovery and automated failover.

- **LUN failure management** - When LUN failure conditions occur, the mass storage stack automatically recovers whenever possible. If necessary, administrators can use the new `scsimgr` command to recover from the LUN failure.

- **Active-Passive device support**

- **SAN dynamic discovery and reconfiguration** – Proactive and automatic discovery of new paths allow applications' I/O traffic to be dynamically balanced across the most up-to-date set of active lunpaths. New DSFs are automatically created for newly discovered LUN devices. The health of LUNs and lunpaths is also dynamically updated as LUNs and lunpaths availability changes. The LUNs and lunpaths health is also made available to user applications via tools like `ioscan`.

- **Full integration with HP-UX 11i v3** – Existing tools (for instance `ioscan`, `sar`, and `smh`) and HP-UX subsystems (for instance boot, dump, file systems (HFS), the Logical Volume Manager (LVM)) have been enhanced to use native multi-pathing. Furthermore new tools like `scsimgr` are provided to manage the native multi-pathing solution.

- **Compatibility with the legacy naming model** – Legacy lunpath DSFs are available on HP-UX 11i v3 for backward compatibility reasons. High availability is ensured by default on legacy DSFs through native multi-pathing. A tunable is provided to enable/disable native multi-pathing on legacy DSF.

- **Pre-HP-UX 11i v3 multi-pathing add-on products are no longer required**.

Benefits of Native Multi-Pathing

Optimal Use of System Resources

The mass storage subsystem optimally uses system resources such as memory and processors to perform I/O operations as efficiently as possible. For instance, on systems supporting cell local memory, you can tune the I/O load distribution to reduce memory, interrupt and device latencies using the "cell local round robin" I/O load balancing policy (see section Disk LUN load balancing selection for more details).

Optimal use of SAN resources

The I/O path selection is optimized to achieve best I/O data throughput. For each SCSI device class, the mass storage subsystem offers multiple I/O load balancing policies. A load balancing policy can be selected and tuned to take into account the idiosyncrasies of the SAN configuration.

New SAN resources are readily made available to the host thanks to the automated discovery of SCSI devices. Stale SAN resources can be deleted to free up system resources.

Scalability

The mass storage subsystem can theoretically address up to 2^{24} (16 million) LUNs. There is no architectural limit on the number of lunpaths allowed to a LUN. However, you may refer to the release notes of HP-UX 11i v3 to find out the maximum number of LUNs and lunpaths that were tested (http://docs.hp.com/en/oshpux11iv3.html).

However, the mass storage subsystem has not removed any scalability limitations present in previous releases of HP-UX for legacy DSFs.

Built-in High Availability of SAN Resources

The mass storage subsystem quickly identifies failing SAN components and action is taken to recover from the error to minimize the impact on applications using the SAN resources. The mass storage subsystem proactively monitors and reports lunpath error conditions and provides error recovery mechanisms. When a lunpath fails, the mass storage subsystem switches the I/O flow to other available lunpaths (**path failover**) while recovery is attempted on the failing lunpath. Depending upon the circumstances, a failing lunpath component can either automatically recover (**self-healing**), or the mass storage subsystem notifies the system administrator who can take appropriate action.

Increased I/O performance

The optimal use of system and SAN resources in the selection of lunpaths leads to significant improvements in I/O throughput. See the *HP-UX 11i v3 Mass Storage I/O Performance Improvements* white paper for detailed information.

Ease of Management

Native multi-pathing is easy to manage for the following reasons:

- The agility of LUN addressing achieved through the use of LUN WWID makes LUN DSFs immune to certain SAN topology changes, such as for instance the addition of new lunpaths to a LUN device.

- Existing HP-UX commands such as `ioscan`, `sar`, `insf`, and `rmsf` have been enhanced to handle LUN multi-pathing. For instance, `ioscan` has a new option, the map option specified by `-m`, to display the mapping between lunpath hardware paths and LUN hardware paths. The `sar` command has a new option (-L) to display data throughput of lunpaths.

- New tools have been introduced. The new command line utility `scsimgr` manages the mass storage subsystem. It is integrated with HP System Management Homepage to offer a graphical user interface. See the *scsimgr SCSI Management and Diagnostics Utility* white paper.

- System administrators no longer have to learn commands specific to multi-pathing add-on products.

- The mass storage subsystem performs quick components fault notification via statistics, diagnostics messages, tunables.

- Tools such as `ioscan` and `scsimgr`, and mechanisms internal to the mass storage subsystem, give an up-to-date view of the SAN components connected to the host at any time. As the SAN topology evolves, the host system view of the SAN is updated. `ioscan` and `scsimgr` also provide easy ways to display the relationships between SAN components.

- Dynamic discovery of SCSI devices automatically provides system administrators with the most up-to-date view of SAN resources.

Native Multi-Pathing Features

LUN WWID and DSF

The various lunpaths to LUN devices are correlated based on a world-wide unique identifier (WWID) that is associated with each LUN device. Indeed, the host can determine that different lunpaths lead to the same LUN only if a LUN can be uniquely identified by the operating system. The LUN WWID is based on SPC-3 commands Inquiry and VPD Inquiry Page 83h data.

Based on this LUN WWID, the mass storage subsystem creates one persistent DSF for each LUN irrespective of the number of lunpaths to the LUN device. The association between a LUN WWID and a LUN DSF is persistently stored on the host.

Using LUN WWID and LUN DSFs has the following advantages:

- **LUN addressing scalability**: Since LUN DSF use opaque minor numbers, the mass storage susystem can theoretically address up to 2^{24} (16 million) LUNs. The pre-11.31i v3 limitations of 32k lunpaths only exists for the legacy naming model.
- **LUN agile addressing**: A LUN DSF remains unaffected by certain SAN topology changes, such as the LUN device being moved from one target port to another target port or, in the case of a Fibre Channel port, a target N-Port ID changing.
- **Abstraction of lunpaths management**: If additional lunpaths (via the addition of new SCSI controller or new SCSI target ports) are offered to a given LUN, or if existing paths components to the same LUN are removed, the LUN DSF remains unaffected.
- **One DSF per LUN irrespective of the number of lunpaths**. Compared to 11i v2, the number of LUN DSFs is no longer proportional to the number of lunpaths. It reflects the number of SCSI devices connected to the host.

Some old SCSI devices do not support VPD Inquiry Page 83h data. For such devices, the mass storage subsystem creates one LUN DSF per lunpath to the LUN device.

See the *HP-UX 11i v3 Mass Storage Device Naming* white paper for more information on persistent DSFs.

Dynamic I/O Load Distribution

The mass storage subsystem dynamically distributes the I/O load across the available lunpaths to a LUN according to a settable policy. This distribution is transparent to the application generating the I/O load.

Dynamic I/O load balancing leads to significant improvements in I/O throughput. See the *HP-UX 11i v3 Mass Storage I/O Performance Improvements* white paper for more information.

To aid in optimal resource utilization, the SCSI stack supports several load balancing policies and tunables settable with `scsimgr`. For instance, disk devices use the round-robin load balancing policy as the default policy. However, administrators have the flexibility to change and tune the default load balancing policy of LUNs using `scsimgr`.

The `sar` command has been enhanced to display details about the I/O performance details on each lunpath. The `scsimgr` command shows the I/O distribution based on both the number of I/O operations and the amount of data transferred for each lunpath.

Policies Supported for Disk Devices

Disk devices support the following I/O load balancing policies:

- **Round-robin (*round_robin*)** – This policy distributes the I/O load equally across all active lunpaths irrespective of the current load on each lunpath. It is suitable when lunpaths have similar I/O operation turnaround characteristics.

- **Least command load (*least_cmd_load*)** – This policy selects the lunpath with the least number of pending I/O requests for the next I/O operation. It is suitable when lunpaths have asymmetric performance characteristics.

- **Cell aware round robin (*cl_round_robin*)** – This policy is applicable to servers supporting hard partitions, which have high latencies for non-local memory access operations. The lunpath chosen to issue an I/O operation is in the same locality in which the I/O is issued. This policy helps optimize memory access latency.

- **Closest path (*closest_path*)** – This policy selects the lunpath based on its affinity with the CPU processing the I/O operation so as to minimize memory access latency. This policy is more appropriate for cell-based platforms. The affinity between the lunpath and CPU is determined based on the relative locations of the CPU processing the I/O operation and the CPU to which the HBA used by the lunpath is bound. The lunpath is selected by order of preference depending on whether both CPUs share:

  1. the same core,
  2. the same socket,
  3. the same Front Side Bus (FSB),
  4. the same cell,
  5. different cells.

- **Preferred path (*preferred_path*), Preferred target port (*pref_tport*)** – These two policies apply to certain types of targets that present an optimized/un-optimized controller model (different from active-passive). An optimized/un-optimized controller pair is one in which the optimized controller is favored for accessing that LUN since it yields better performance.
  - With the preferred path, you specify a lunpath to the optimized controller. This lunpath is used preferably for I/O transfers to the disk device.
  - With the preferred target port, you specify an optimized target port. Lunpaths to this target port are used preferably for I/O transfers to the disk device.

  Note: Refer to the disk device's user documentation for information on determining or configuring the optimized/un-optimized controller or target port. Based on this information you can set the *preferred_path* and the *pref_tport* attributes appropriately on the host system.

- **Weighted round-robin (*weighted_rr*)** – This policy distributes the I/O load across all active lunpaths in a round-robin manner and according to the weight assigned to each lunpath. A number of I/O operations corresponding to the weight of the lunpath is transferred on the same lunpath before another lunpath is selected. Lunpaths with a weight of 0 are excluded from I/O transfer.

The default disk device load balancing policy is round-robin, even on servers supporting hard partitioning. However, the administrator can change it to any of the above listed.

Note: For devices implementing the T10 ALUA (Asymmetric Logical Unit Access) standard (refer to SPC-3 section 5.8), the SCSI stack on HP-UX 11i v3 automatically detects the set of lunpaths to the optimized target port groups and uses only these lunpaths for I/O transfer according to the I/O load balancing policy set for the disk device. Lunpaths to un-optimized target port groups are put in standby state and are not used for I/O transfer. You can disable this behavior by setting the *alua_enabled* attribute to "false" for the disk device. In this case the support of the ALUA standard is ignored by the SCSI stack and all lunpaths are used for I/O transfer.

The administrator has a lot of flexibility in setting the scope of load balancing policies for disks. A load balancing policy can be set for all disk devices, or for a set of disk devices meeting some specific criteria, such as vendor id, product id, and firmware revision, or for a specific disk device.

Policies Supported for Tapes, Changer, and Pass-Through Devices

Because of the serial nature of the I/O flow to tape and changer devices, only one path selection mechanism is supported by `estape` and `eschgr` drivers: **path lock down (*path_lockdown*)**. When opening a tape or changer LUN DSF, the lunpath onto which all the I/O operations are sent can be selected by the mass storage subsystem or set by the administrator. By default, the mass storage subsystem internally selects the optimal lunpath. The administrator can manually set the lunpath with the lpt_to_lockdown attribute. The lpt_to_lockdown attribute must only be set when the device is in the UNOPEN state.

Devices claimed by the pass-through driver (`esctl`) also use path lockdown as the I/O load balance policy.

High Availability of SCSI devices

To provide applications with continuous access to the LUNs, the mass storage subsystem has built-in LUN access high availability.

By default, without requiring special upper layers, the mass storage subsystem increases the availability of LUN devices to applications by doing lunpath failovers, detecting and removing failing lunpaths from the application I/O traffic, by monitoring offline lunpaths, and automatically making lunpaths that come online available to application I/O traffic.

The SCSI subsystem monitors SCSI components, such as SCSI controllers, target ports, lunpaths and LUNs. As failing components are identified, the mass storage subsystem reports errors by logging messages in the system log, then the SCSI subsystem resorts to path failover and recovery mechanisms to provide applications with continuous access to the LUN end devices; only when certain critical errors happen, is administrator intervention required.

In summary, after detecting a SCSI component error, the operating system reports the error to system administrators and offers a palette of pro-active recovery actions: automatic path failover or dynamic replacement of the failing component.

Path Error Reporting

HP-UX 11i v3 offers new and enhanced mechanisms to report failures on SCSI components. The goal is to assist administrators in performing quick and efficient diagnostic to take the most appropriate action.

- **Error messages** - A comprehensive set of error messages of various severity levels are used to report a wide range of errors. These messages can be monitored in syslog and STM.

- **Statistics** - A detailed set of statistics is available for each SCSI component to help troubleshooting, and to quickly identify a faulting component. Administrators can use `scsimgr` to display these statistics.

- **EVM event**s – The mass storage subsystem generates events to which other modules in the kernel or user space can subscribe, to get notified about changes on a LUN and lunpath properties. The SCSI stack monitors every LUN and every lunpath availability change. It also monitors LUN property changes such as LUN size.

- **I/O error triggered events** – The mass storage subsystem also reports failures on the lunpaths and LUN upon detection of certain I/O errors.

Path Failover

When a lunpath goes offline, I/O operations issued on that lunpath fail. The policy of how the mass storage subsystem deals with this scenario is dependent upon three factors:

- **Path bound I/O operations**
  User applications can request the mass storage subsystem to issue I/O operations on a specific lunpath. Such I/O operations are called path bound I/O operations. When a path bound I/O fails, the SCSI stack retries it a certain number of times on the same lunpath before failing it back to the upper layer (for example applications, file systems, volume managers). There is no path failover. I/O operations sent to a LUN using the path lock down load balancing policy are path bound I/O operations.

- **I/O operations not bound to a path**
  Path failover is applied to I/O operations that are not bound to a lunpath . When an not bound to a path I/O operation fails, if the I/O can be retried (see I/O retry policy below), the I/O is failed over the next lunpath selected by the LUN load balancing policy path selection algorithm.

- **I/O retry policy**
  The mass storage subsystem retries a failing I/O operation a certain number of times before returning failure to the application using one of two retry policies:

  - **time based** - The mass storage subsystem retries the I/O operation within a certain time interval which is either set by upper layer modules such as the volume manager or the file system, or determined by a default LUN attribute. For disk LUNs, the *esd_secs* LUN attribute holds the time credit for an I/O operation across different retries. For tape LUNs, the *read_secs* attribute and *write_secs* attribute hold the read and write time credits.

  - **count based** - The I/O operation is retried less than a high-water mark threshold . For disk LUNs, I/O operations can be retried indefinitely (if disk LUN *infinite_retries_enable* attribute is set), or a finite number of times (corresponding to the disk LUN *max_retries* attribute value).

Path Recovery Policies

When a disk device no longer responds to I/O operations sent through a lunpath for a configurable period of time (attribute *path_fail_secs*), this lunpath is considered offline, and is no longer used for I/O transfer. Prior to the March 2008 release of HP-UX 11i v3, the system automatically monitors the lunpath by periodically sending inquiry requests to the device through it, and declares the lunpath back online when the inquiry requests succeed.

Starting with the March 2008 release of HP-UX 11i v3, you have more control on automatic path recovery. You can configure path ping policies and path recovery policies.

The following path ping policies can be configured:

- **None** – No connectivity test is performed when a lunpath is declared offline.
- **Basic** – The connectivity of the lunpath is tested by periodically sending INQUIRY SCSI commands to the device through the lunpath.
- **Extended** – The connectivity of the lunpath is tested by periodically sending INQUIRY and TUR (Test Unit Ready) SCSI commands through the lunpath.

The following path recovery policies can be configured:

- **Immediate** – The lunpath is considered back online upon a successful connectivity test (ping).
- **Time-based threshold** – The lunpath is considered back online upon sustained and consecutive successful connectivity tests during a configured period of time.
- **Count-based threshold** – The lunpath is considered back online after a specified number of consecutive successful connectivity tests.

LUN Failure Management

Managing Loss of Accessibility to a LUN

When all the lunpaths to a LUN are taken offline, the LUN becomes inaccessible. The mass storage subsystem still accepts I/O operations to this LUN during a **transient grace period**. This mechanism gives some time for the LUN to recover and shields applications from SAN transient conditions that may cause a temporary loss of accessibility to the LUN. If at least one lunpath becomes active before the end of the grace period, incoming I/O operations can flow again. After this grace period expires, pending I/O operations and further incoming I/O operations to this LUN are returned with a failure indication until a lunpath comes back online. Furthermore the stack provides more flexibility to the customer by allowing administrators to tune the value of the transient grace period. This allows the mass storage subsystem to provide high availability customized to the local SAN.

Managing Authentication Failure

Whenever a LUN is opened, the SCSI stack authenticates that the LUN represented by the DSF is still the same device. The goal is to prevent data written on the physical LUN device from being corrupted by mistake when LUN devices are swapped. The authentication consists of getting the WWID of the LUN from the LUN device and comparing it with the WWID associated with the DSF at the time of the DSF creation.

If any change in LUN behavior is detected, a LUN authentication failure is printed on the console. To prevent data corruption, any pending I/O operation to the LUN is failed and no further I/O operation can be sent to the LUN until the LUN is re-authenticated properly or until the LUN identifier is changed by the administrator. System administrators can use "`scsimgr replace_wwid`" to change the "identifier" associated with a given LUN DSF (See *"scsimgr SCSI Management and Diagnostics Utility"* white paper).

Media Based Errors

If the SCSI stack detects the LUN disk device could not commit a write from its cache to the media, then it freezes the I/O traffic to the disk to prevent data corruption. When the error condition on the LUN has been cleared, the system administrator must run "`scsimgr enable -D <lun DSF>`" to resume I/O operations on the LUN.

Active-Passive Device Support

Active-passive arrays are supported if any of the following conditions is true:

- The vendor has developed and provided an Active/Passive Switch (APSW) plug-in module on HP-UX 11i v3. Refer to the "*Writing Active/Passive Switch Plug-Ins*" white paper for more information.
- The disk array supports the T10 Asymmetric Logical Unit Access (ALUA) standard. Refer to SPC-3 Section 5.8 for more information.
- The disk array is now active-active symmetric and no loss of performance will occur on a host performing multi-path load balancing simultaneously via different paths to the same LUN.
- The LUN or disk array is connected to a single host OS instance, the *load_bal_policy* attribute is set to *preferred_path* or *pref_tport,* and the array is in auto-trespass mode.

SAN Dynamic Discovery and Reconfiguration

SCSI Devices Dynamic Discovery

At boot time, SCSI devices connected to the host are automatically discovered and corresponding SCSI components are instantiated by the operating system. Once the system is up and running, administrators can add new SCSI devices, disconnect existing SCSI devices, or replace path components, such as a failing Fibre Channel HBA. Certain topology change events are automatically detected by the mass storage subsystem while others require the system administrator's intervention.

The mass storage subsystem asynchronously detects some topology changes in the SAN. For instance, the addition of a Fibre Channel port to a Fibre Channel switch triggers a Fibre Channel Request Status Change Notifications (RSCN) that is automatically detected by the SCSI stack. Upong receiving the RSCN, the SCSI services automatically discover the LUNs connected to the newly added target port.

However, some SAN topology changes are not automatically detected by the operating system. They require the administrator to run `ioscan`. For instance, after adding Parallel SCSI LUNs, the administrator must run `ioscan` to discover the newly added LUNs.

Both automatic SCSI device discovery and the `ioscan` command have the following results:

- The mass storage subsystem registers the corresponding paths components in both the legacy and the agile I/O tree views.
- Legacy lunpaths DSF and LUN DSFs are automatically created to make the new devices readily accessible to applications.

- The mass storage subsystem makes the new lunpaths available to I/O traffic with no interruption to applications, and, in the case of automatic device discovery, without requiring special actions from the system administrator.

The complexity of the SAN components under the LUN is hidden to the administrator:

- The dynamic discovery of components is transparent to the administrator. If needed, running `ioscan` (without the –k option) automatically creates the new LUN DSFs.
- Newly discovered active lunpaths are automatically used in the lunpath selection scheme of I/O load balancing. Therefore, applications sending I/O operations through existing LUN DSFs are not affected by the discovery of new lunpaths.
- New LUN DSFs corresponding to newly discovered LUNs are seemlessly created.

SAN Component Deletion

Throughout the life of a system, SAN components may be replaced or removed. For instance, lunpaths to a LUN device may become stale because the target port involved has been replaced. Administrators can delete SCSI target paths and SCSI lunpaths using `rmsf`. They can also replace and delete SCSI controllers using the OL* process.

The I/O stack transparently handles deletion or disabling of lunpaths. As long as there is at least one available lunpath to a LUN, I/O traffic is not affected. This allows customers to replace faulty SCSI HBAs (via OL* operations) or faulty SCSI target ports without impacting applications.

See corresponding man pages on `rmsf(1M)` and `pdweb(1M)`.

Easy Monitoring of LUNs and Lunpaths Accessiblity

To provide administrators with a real-time status of the availability of lunpath or a LUN components, the mass storage subsystem dynamically updates the "health" property of lunpath and LUN I/O nodes. The "health" property conveys availability status values such as online, offline, unusable, disable or standby. Without performing a full scan of the SAN configuration, the administrator can display the health property of lunpaths and LUN components by running "`ioscan -P health`".

See section Lunpath and LUN Health for more details.

Full Integration with HP-UX 11i v3

Multi-pathing is fully integrated with HP-UX 11i v3. Here are a few examples of subsystems and utilities that have evolved toward using multi-pathing.

crashconf(1M)

The dump subsystem is aware of multi-pathed devices and supports automatic dump device path failover. If a configured lunpath goes offline, the dump subsystem automatically selects an alternate available hardware lunpath and reconfigures the dump device. The dump subsystem also takes advantage of multi-pathing through the concurrent dump functionality which significantly reduces dump time.

For more information on changes to the dump subsystem, see the *"HP-UX 11i v3 Crash Dump Improvement"* white paper.

setboot(1M)

The `setboot` command is aware of multi-pathed devices and supports automatic boot path failover. If the hardware lunpath written into stable storage goes offline, `setboot` retrieves an alternate available lunpath to the LUN and writes it into the system stable storage.

Refer to the "*setboot(1M) in HP-UX 11i v3*" white paper for more information.

LVM

In the HP-UX 11i v3 release, LVM supports both legacy and persistent DSFs. Since multi-pathing applies by default to legacy and persistent LUN DSFs, LVM uses native multi-pathing by default. All the LVM commands (`pvcreate, vgcreate` and `vgextend`) are backward compatible, and work transparently with both legacy and persistent DSFs. New options have been added `to vgimport` and `vgscan` to select the persistent DSF naming model. HP highly recommends using persistent DSFs for LVM configurations. The LVM alternate links (PVlinks) feature is still supported in HP-UX 11i v3.

See the "*the LVM Migration from Legacy to Agile Naming Model HP-UX 11i v3*" white paper for more information.

File Systems

In HP-UX 11i v3, HFS supports both legacy and persistent LUN DSFs. Therefore, HFS takes advantage of multi-pathing by default. For support of VxFS on HP-UX 11iv3, see the release notes from Symantec.

Commands

Existing commands such as `ioscan` and `sar` have evolved to use multi-pathing. New options have been introduced. For information on the changes, see the command's associated man page and the HP-UX 11i v3 Release Notes.

Compatibility with the Legacy Naming Model

By default, native multi-pathing applies to legacy DSFs even though a legacy DSF corresponds to a lunpath to a LUN. When an application accesses a legacy DSF, the mass storage subsystem internally applies the configured LUN level multi-pathing (I/O load balancing, automated path failure detection and recovery) to send I/O requests to the end LUN. This offers applications migrating to HP-UX 11i v3 the opportunity to easily take advantage of multi-pathing: no special configuration is needed.

At the same time the mass storage subsystem is flexible to accommodate the strict backward compatibility needs of some applications by allowing disabling of native multi-pathing on legacy DSFs via a tunable. See section "Setting Native Multi-Pathing support on legacy DSFs".

Pre-HP-UX 11i v3 Multi-Pathing Add-on Products are no longer required

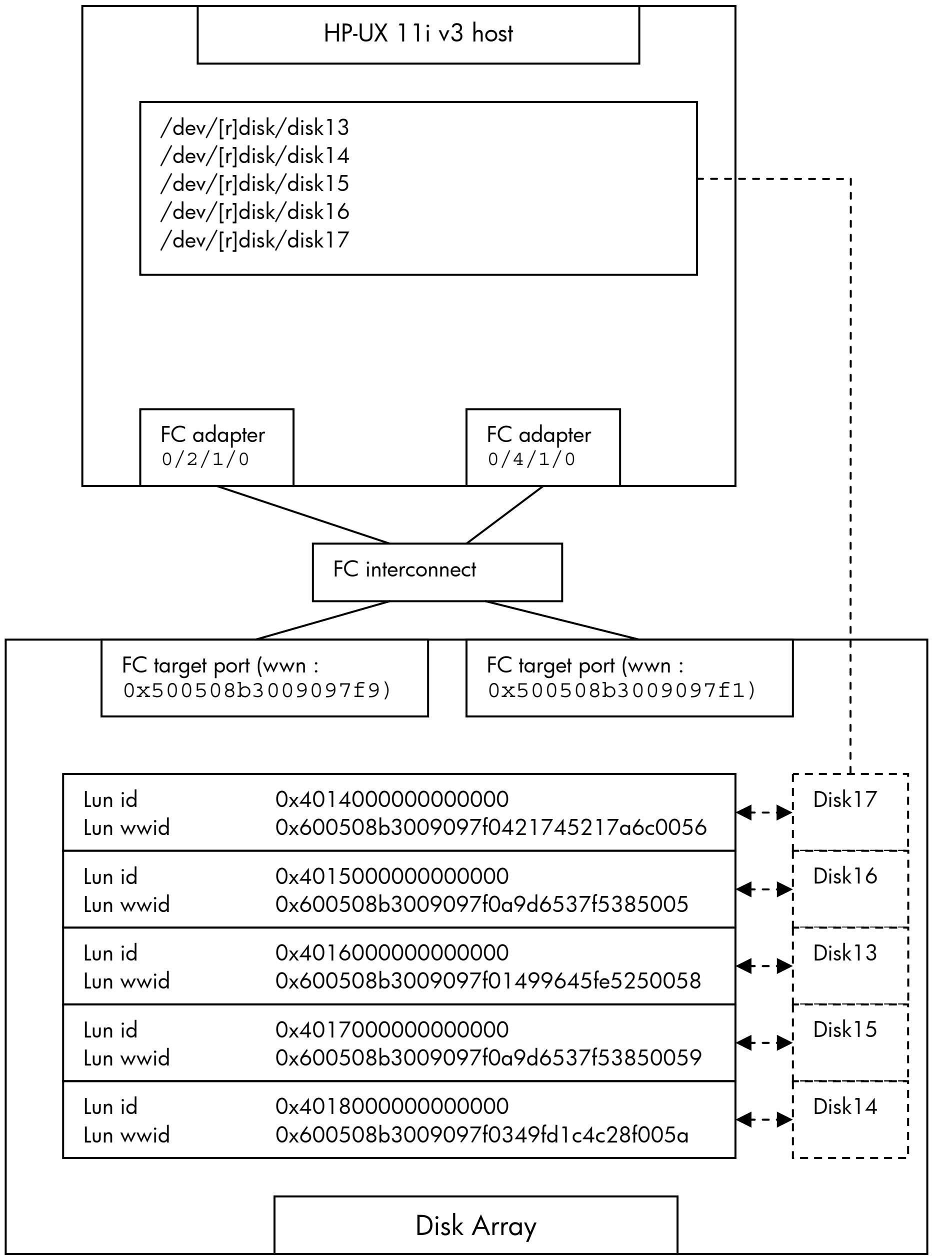
HP add-on AutoPath® and SecurePath® products are no longer supported with HP-UX 11i v3. For more information, see the "*Migrating from HP Storage Works Secure Path for Active-Active Disk Arrays to Native Multipathing in HP-UX 11i v3*" white paper.

To determine if other add-on multi-pathing products are supported on HP-UX 11i v3, see their associated release notes.

Managing HP-UX 11i v3 Native Multi-Pathing

To illustrate the management of multi-pathing on HP-UX 11iv3, consider a real world example. The SAN is comprised of:

- a server running HP-UX 11i v3. It is connected to the SAN via two host bus adapters (HBAs)
- a disk array with five LUN devices connected to the SAN via two target ports.



Provided that all the LUNs are configured to be visible to the host via each HBA and each target port, each LUN can be accessed via four different lunpaths corresponding to all the possible combinations of HBAs and target ports. In HP-UX 11i v3, only one persistent DSF per LUN is exported to applications. For instance, the following persistent DSFs exist:

- /dev/[r]disk/disk13
- /dev/[r]disk/disk14
- /dev/[r]disk/disk15
- /dev/[r]disk/disk16
- /dev/[r]disk/disk17

For each LUN device, the `esdisk` driver selects by default the round-robin dynamic I/O load balancing policy.

Viewing Native Multi-pathing Related Information

Hardware Paths to Lunpaths, Targetpaths and LUNs

Run `ioscan` to see the hardware paths for SCSI HBA, target paths, lunpaths and LUNs in the **agile view**.

There are two Fibre Channel SCSI HBAs:

```
# ioscan -kfnC fc
Class     I  H/W Path  Driver S/W State   H/W Type     Description
================================================================
fc        0  0/2/1/0   td     CLAIMED     INTERFACE    HP Tachyon XL2 Fibre Channel Mass
Storage Adapter
fc        3  0/4/1/0   td     CLAIMED     INTERFACE    HP Tachyon XL2 Fibre Channel Mass
Storage Adapter
```

The **agile view** shows two target ports with world-wide names **0x500508b3009097f1** and **0x500508b3009097f9** and four target paths:

```
# ioscan -kfnNC tgtpath
Class     I  H/W Path       Driver S/W State   H/W Type     Description
...
tgtpath   7  0/2/1/0.0x500508b3009097f1  estp   CLAIMED     TGT_PATH     fibre_channel
target served by td driver
tgtpath   3  0/2/1/0.0x500508b3009097f9  estp   CLAIMED     TGT_PATH     fibre_channel
target served by td driver
tgtpath   8  0/4/1/0.0x500508b3009097f1  estp   CLAIMED     TGT_PATH     fibre_channel
target served by td driver
tgtpath   6  0/4/1/0.0x500508b3009097f9  estp   CLAIMED     TGT_PATH     fibre_channel
target served by td driver
```

The lunpaths are represented by a concatenation of the target path hardware path and the LUN 64 bit address (eg. `0/2/1/0.0x500508b3009097f9 + LUN address` **`0x4014000000000000).`** The following `ioscan` output shows a few lunpaths:

```
# ioscan -kfnNC lunpath
Class     I  H/W Path  Driver S/W State   H/W Type     Description
...
lunpath  15  0/2/1/0.0x500508b3009097f1.0x0                 eslpt   CLAIMED     LUN_PATH
LUN path for ctl2
lunpath  16  0/2/1/0.0x500508b3009097f1.0x4014000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk17
lunpath  19  0/2/1/0.0x500508b3009097f1.0x4015000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk16
lunpath  17  0/2/1/0.0x500508b3009097f1.0x4016000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk13
lunpath  18  0/2/1/0.0x500508b3009097f1.0x4017000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk15
lunpath  20  0/2/1/0.0x500508b3009097f1.0x4018000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk14
lunpath   0  0/2/1/0.0x500508b3009097f9.0x0                 eslpt   CLAIMED     LUN_PATH
LUN path for ctl2
```

```
lunpath    5  0/2/1/0.0x500508b3009097f9.0x4014000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk17
lunpath    4  0/2/1/0.0x500508b3009097f9.0x4015000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk16
lunpath    1  0/2/1/0.0x500508b3009097f9.0x4016000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk13
lunpath    3  0/2/1/0.0x500508b3009097f9.0x4017000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk15
lunpath    2  0/2/1/0.0x500508b3009097f9.0x4018000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk14
lunpath   21  0/4/1/0.0x500508b3009097f1.0x0                 eslpt    CLAIMED      LUN_PATH
LUN path for ctl2
lunpath   22  0/4/1/0.0x500508b3009097f1.0x4014000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk17
lunpath   23  0/4/1/0.0x500508b3009097f1.0x4015000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk16
lunpath   26  0/4/1/0.0x500508b3009097f1.0x4016000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk13
lunpath   24  0/4/1/0.0x500508b3009097f1.0x4017000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk15
lunpath   25  0/4/1/0.0x500508b3009097f1.0x4018000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk14
lunpath    8  0/4/1/0.0x500508b3009097f9.0x0                 eslpt    CLAIMED      LUN_PATH
LUN path for ctl2
lunpath   13  0/4/1/0.0x500508b3009097f9.0x4014000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk17
lunpath   11  0/4/1/0.0x500508b3009097f9.0x4015000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk16
lunpath   12  0/4/1/0.0x500508b3009097f9.0x4016000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk13
lunpath   10  0/4/1/0.0x500508b3009097f9.0x4017000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk15
lunpath    9  0/4/1/0.0x500508b3009097f9.0x4018000000000000  eslpt    CLAIMED      LUN_PATH
LUN path for disk14
```

The above shows that five disk LUNs are connected to the target port of world-wide name **`0x500508b3009097f9`** (shown in green) and that `ioscan` displays the DSF of the LUN (for example `disk17`) to which each lunpath maps. The LUN corresponding to `/dev/[r]disk/disk17` has four lunpaths.

LUN hardware paths and the corresponding LUN DSFs in the **agile view** look like:

```
# ioscan -kfnNC disk
Class      I  H/W Path  Driver S/W State   H/W Type     Description
===================================================================
disk      13  64000/0xfa00/0x1  esdisk   CLAIMED     DEVICE       COMPAQ  MSA1000 VOLUME
                       /dev/disk/disk13   /dev/rdisk/disk13
disk      14  64000/0xfa00/0x2  esdisk   CLAIMED     DEVICE       COMPAQ  MSA1000 VOLUME
                       /dev/disk/disk14   /dev/rdisk/disk14
disk      15  64000/0xfa00/0x3  esdisk   CLAIMED     DEVICE       COMPAQ  MSA1000 VOLUME
                       /dev/disk/disk15   /dev/rdisk/disk15
disk      16  64000/0xfa00/0x4  esdisk   CLAIMED     DEVICE       COMPAQ  MSA1000 VOLUME
                       /dev/disk/disk16   /dev/rdisk/disk16
disk      17  64000/0xfa00/0x5  esdisk   CLAIMED     DEVICE       COMPAQ  MSA1000 VOLUME
                       /dev/disk/disk17   /dev/rdisk/disk17
...
```

<u>LUN WWID</u>

The *wwid* attribute holds the LUN VPD page 83h WWID:

```
# scsimgr get_attr -D /dev/rdisk/disk17 -a wwid

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk1

name = wwid
current = 0x600508b3009097f0421745217a6c0056
default =
saved =
```

LUN-to-lunpaths Mapping

ioscan(1M)

The `ioscan` command is now equipped with a powerful set of new options to query the relationship among various SAN components. The following sections demonstrate the use of these new options in the context of the above example.

The `ioscan command` can query the mapping combinations between LUN and lunpaths using DSF or hardware paths.

- *To which legacy DSF does `/dev/rdisk/disk17` map?*

```
# ioscan -m dsf /dev/rdisk/disk17
Persistent DSF          Legacy DSF(s)
========================================
/dev/rdisk/disk17       /dev/rdsk/c5t2d4
                        /dev/rdsk/c7t2d4
                        /dev/rdsk/c9t2d4
                        /dev/rdsk/c11t2d4
```

- *To which persistent DSF does `/dev/rdsk/c5t2d4` map?*

```
# ioscan -m dsf /dev/rdsk/c5t2d4
Persistent DSF          Legacy DSF(s)
========================================
/dev/rdisk/disk17       /dev/rdsk/c5t2d4
```

- *To which lunpath and legacy hardware paths does LUN hardware path `64000/0xfa00/0x5` map?*

```
# ioscan -m hwpath -H 64000/0xfa00/0x5
Lun H/W Path      Lunpath H/W Path                 Legacy H/W Path
====================================================================
64000/0xfa00/0x5
                  0/2/1/0.0x500508b3009097f9.0x4014000000000000   0/2/1/0.3.8.0.0.2.4
                  0/4/1/0.0x500508b3009097f9.0x4014000000000000   0/4/1/0.3.8.0.0.2.4
                  0/2/1/0.0x500508b3009097f1.0x4014000000000000   0/2/1/0.3.14.0.0.2.4
                  0/4/1/0.0x500508b3009097f1.0x4014000000000000   0/4/1/0.3.14.0.0.2.4
```

- *To which lunpath hardware path and LUN hardware path does legacy lunpath hardware path `0/2/1/0.3.8.0.0.2.4` map?*

```
# ioscan -m hwpath -H 0/2/1/0.3.8.0.0.2.4
Lun H/W Path      Lunpath H/W Path                 Legacy H/W Path
====================================================================
64000/0xfa00/0x5
                  0/2/1/0.0x500508b3009097f9.0x4014000000000000   0/2/1/0.3.8.0.0.2.4
```

- *To which LUN and legacy hardware path does new lunpath hardware path `0/2/1/0.0x500508b3009097f9.0x4014000000000000` map?*

```
# ioscan -m hwpath -H 0/2/1/0.0x500508b3009097f9.0x4014000000000000
Lun H/W Path      Lunpath H/W Path                 Legacy H/W Path
====================================================================
64000/0xfa00/0x5
                  0/2/1/0.0x500508b3009097f9.0x4014000000000000   0/2/1/0.3.8.0.0.2.4
```

For more information, see the [HP-UX 11i v3 Mass Storage Device Naming]() white paper.

scsimgr(1M)

The `scsimgr` command displays the mapping between LUNs and lunpaths.

```
# scsimgr lun_map -D /dev/rdisk/disk17

        LUN PATH INFORMATION FOR LUN : /dev/rdisk/disk17

Total number of LUN paths     = 4
World Wide Identifier(WWID)    = 0x600508b3009097f0421745217a6c0056

LUN path : lunpath5
Class                         = lunpath
Instance                      = 5
Hardware path                 = 0/2/1/0.0x500508b3009097f9.0x4014000000000000
SCSI transport protocol       = fibre_channel
State                         = UNOPEN
Last Open or Close state      = ACTIVE

LUN path : lunpath13
Class                         = lunpath
Instance                      = 13
Hardware path                 = 0/4/1/0.0x500508b3009097f9.0x4014000000000000
SCSI transport protocol       = fibre_channel
State                         = UNOPEN
Last Open or Close state      = ACTIVE

LUN path : lunpath16
Class                         = lunpath
Instance                      = 16
Hardware path                 = 0/2/1/0.0x500508b3009097f1.0x4014000000000000
SCSI transport protocol       = fibre_channel
State                         = UNOPEN
Last Open or Close state      = ACTIVE

LUN path : lunpath22
Class                         = lunpath
Instance                      = 22
Hardware path                 = 0/4/1/0.0x500508b3009097f1.0x4014000000000000
SCSI transport protocol       = fibre_channel
State                         = UNOPEN
Last Open or Close state      = ACTIVE
```

The above output shows the lunpath states as `UNOPEN`: it means that the LUN is currently not open. Furthermore, the last open or close state is `ACTIVE`: the LUN is available for I/O traffic.

Statistics

The effects of the distribution of the I/O load across lunpaths can be observed using I/O transfer statistics. Using `scsimgr`, administrators can gather the following I/O transfer statistics on lunpath and LUNs. The `sar` command can also display I/O throughput on LUN and lunpaths (see the Native multi-pathing: an example section below).

In the context of LUN multi-pathing, it is also very useful to be able to monitor the behavior of lunpaths and LUN components. Using error related statistics such as I/O errors and open errors system administrators can readily identify failing components and understand the cause of the failure. For instance, the administrator can gather the following lunpath open failure statistics and I/O errors statistics:

```
# scsimgr get_stat -H 0/4/1/0.0x500508b3009097f9.0x4014000000000000

        STATISTICS FOR LUN path : 0/4/1/0.0x500508b3009097f9.0x4014000000000000

Generic Statistics:
…
Open failures due to inquiry failure             = 0
```

```
Open failures due to insufficient inquiry data   = 0
Open failures due to invalid peripheral qualifier= 0
EVPD page 83h inquiries                          = 2757
EVPD page 83h inquiry failures                   = 0
insufficient EVPD page 83h inquiry data          = 0
Mismatches of peripheral qualifiers              = 0
Page 83h ids not associated with the LUN         = 0
Open failures due to no page 83h response        = 0
Open failures due to page 83h code mismatches    = 0
Open failures due page 83h id mismatches         = 0
Open failures due to page 83h length mismatches  = 0
Open failures due to page 83h content mismatches = 0
…
I/O transfer Statistics:
…
Check condition status                           = 43
Busy status                                      = 0
Queue full status                                = 0
Reservation conflicts                            = 0
Invalid Request status                           = 0
Select Timeout status                            = 0
Incomplete status                                = 0
No Resource status                               = 0
Target Path Offline status                       = 0
IO Timeout status                                = 0
IO Aborted status                                = 0
Reset Occured status                             = 0
Unrecognized CDB status                          = 0
Bad Sense data status                            = 0
Deferred errors                                  = 0
Recovered errors                                 = 0
NOT READY sense status                           = 0
Medium errors                                    = 0
Hardware errors                                  = 0
Illegal request sense status                     = 42
Unit Attentions                                  = 1
```

Lunpath and LUN Health

The "health" property of an I/O node is new in HP-UX 11i v3. It can be displayed with "`ioscan –P health`" (see the *"Next Gen Mass storage stack white-paper"* white paper).

The table below explains the "health" property values of a disk LUN I/O node.

online	The LUN is accessible through one or more lunpaths
offline	All the lunpaths to the LUN have gone offline. As a result, the LUN is not accessible
limited	Not all paths to the LUN are active : some paths are failing
unusable	Not Applicable
disabled	The LUN has been suspended as a result of one of the following errors: • LUN disabled due to unrecoverable deferred error. • LUN disabled administratively using the '`scsimgr disable`' command. Note that this is currently supported for disk devices only. • Change in block size (for disk only) As a result of being disabled, all pending I/O requests to the LUN are aborted and are returned with an error to upper layers and applications. No further I/O operation can be issued to the LUN.
standby	Not Applicable

The table below explains the "health" property values of a lunpath I/O node.

online	The lunpath is active

offline	The lunpath is reported offline as a result of the following conditions:
	• The underlying target path becoming offline (see target path offline)
	• The lunpath open failed for some of the following reasons:
	  o INQUIRY command failure
	  o WWID authentication failure

limited	Not Applicable
unusable	Lunpath authentication failed
disable	The lunpath has been suspended as a result of one of the following errors: • Lunpath disabled administratively using the '`scsimgr disable`' command. • The corresponding LUN has been disabled
standby	For active/passive devices, the path is a standby path

See Native Multi-Pathing: A Concrete Example section for an example on how the "health" property changes when a target port becomes unavailable.

LUN and Lunpaths Attributes

`The scsimgr` command reads and sets the multi-pathing related attributes of LUN components. All the LUN and lunpath attributes are described in details in the *scsimgr SCSI Management and Diagnostics Utility* white paper. The following table summarizes LUN and lunpath attributes of interest for multi-pathing:

Component	Attributes
LUN	Generic: *hw_path, wwid, serial_number, vid, pid, total_path_cnt, transient_secs, leg_mpath_enable, lpt_lockdown, uniq_name* Disk LUNs: *path_fail_secs, load_bal_policy, infinite_retries_enable, preferred_path, pref_tport, alua_enabled*
lunpath	Generic: *lunid, hw_path*

Setting I/O Load Balancing for Disk Devices

The following examples show how the system administrator can set the I/O load balancing policy and other related attributes. The attribute values can be modified temporary with the "`scsimgr set_attr`" command or persistently across system reboots with the "`scsimgr save_attr`" command. The latter command can also be used to persistently restore attribute default values. For more information, see scsimgr(1M).

Disk LUN Load Balancing Selection

The default load balancing policy for disk devices is "round_robin":

```
# scsimgr get_attr -D /dev/rdisk/disk17 -a load_bal_policy

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = load_bal_policy
current = round_robin
default = round_robin
```

This can be changed to "cell locality based round robin":

```
# scsimgr set_attr -D /dev/rdisk/disk17 -a load_bal_policy=cl_round_robin
# scsimgr get_attr -D /dev/rdisk/disk17 -a load_bal_policy

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = load_bal_policy
current = cl_round_robin
default = round_robin
saved =
```

To change the *load_bal_policy* attribute to "preferred_path", the *preferred_path* attribute must be set to the hardware path of the preferred lunpath. The following commands show how to find the lunpaths hardware paths mapping to /dev/rdisk/disk17 and how to select one of them as the preferred path. For optimal performance, the preferred lunpath should correspond to an optimized controller or target port group. In the example below, we assume the target port 0x500508b3009097f9 is optimized for I/O transfer to disk17, and we want to make the change persistent across system reboots:

```
# ioscan -m hwpath -H 64000/0xfa00/0x5
Lun H/W Path      Lunpath H/W Path                  Legacy H/W Path
====================================================================
64000/0xfa00/0x5
                  0/2/1/0.0x500508b3009097f9.0x4014000000000000   0/2/1/0.3.8.0.0.2.4
                  0/4/1/0.0x500508b3009097f9.0x4014000000000000   0/4/1/0.3.8.0.0.2.4
                  0/2/1/0.0x500508b3009097f1.0x4014000000000000   0/2/1/0.3.14.0.0.2.4
                  0/4/1/0.0x500508b3009097f1.0x4014000000000000   0/4/1/0.3.14.0.0.2.4

# scsimgr save_attr -D /dev/rdisk/disk17 -a load_bal_policy=preferred_path -a
preferred_path=0/2/1/0.0x500508b3009097f9.0x4014000000000000
Value of attribute load_bal_policy saved successfully
Value of attribute preferred_path saved successfully

# scsimgr get_attr -D /dev/rdisk/disk17
…
name = load_bal_policy
current = preferred_path
default = round_robin
saved = preferred_path
…
name = preferred_path
current = 0/2/1/0.0x500508b3009097f9.0x4014000000000000
default =
saved = 0/2/1/0.0x500508b3009097f9.0x4014000000000000
…
```

Note: The *preferred_path* attribute is only relevant when the *load_bal_policy* attribute is set to "preferred_path".

In the above example, if you want the two paths to the optimized target port to be used preferably for I/O transfer to disk17, set the I/O load balancing policy to "pref_tport" and the *pref_tport* attribute to 0x500508b3009097f9. For example:

```
# scsimgr save_attr -D /dev/rdisk/disk17 -a load_bal_policy=pref_tport -a
pref_tport=0x500508b3009097f9
Value of attribute load_bal_policy saved successfully
Value of attribute pref_tport saved successfully

# scsimgr get_attr -D /dev/rdisk/disk17 –a load_bal_policy –a pref_tport

name = load_bal_policy
current = pref_tport
default = round_robin
```

```
saved = pref_tport

name = pref_tport
current = 0x500508b3009097f9
default =
saved = 0x500508b3009097f9
```

Note: The *pref_tport* attribute is only relevant when the *load_bal_policy* attribute is set to “pref_tport”.

To reset *load_bal_policy*, *pref_tport* and *preferred_path* to their default values, run the following command:

```
# scsimgr save_attr default -D /dev/rdisk/disk17 -a load_bal_policy –a pref_tport –a
preferred_path

Value of attribute load_bal_policy saved successfully
Value of attribute pref_tport saved successfully
Value of attribute preferred_path saved successfully

# scsimgr get_attr –D /dev/rdisk/disk17 –a load_bal_policy –a pref_tport –a preferred_path

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = load_bal_policy
current = round_robin
default = round_robin
saved =

name = pref_tport
current =
default =
saved =

name = preferred_path
current =
default =
saved =
```

To change the *load_bal_policy* attribute to “least_cmd_load”, run `scsimgr` as follows:

```
# scsimgr set_attr -D /dev/rdisk/disk17 -a load_bal_policy=least_cmd_load
Value of attribute load_bal_policy set successfully
# scsimgr get_attr -D /dev/rdisk/disk17
…
name = load_bal_policy
current = least_cmd_load
default = round_robin
saved
…
```

<u>Pass-Through LUN Load Balancing</u>

The pass-through device driver (`esctl`) only supports the path_lockdown multi-pathing policy. By default, the SCSI stack internally selects the path on which to send I/O operations and sets the LUN *lpt_lockdown* attribute to the hardware path of the locked down path. Run `scsimgr` to see the *lpt_lockdown* attribute value:

```
# scsimgr get_attr -D /dev/pt/pt2 –a lpt_lockdown
…
name = lpt_lockdown
current = 0/4/1/0.0x500508b3009097f9.0x0
default =
saved =
```

This shows that for device /dev/pt/pt2, the OS internally selected the lunpath of hardware path `0/4/1/0.0x500508b3009097f9.0x0.`

To change the lunpath, run scsimgr as follows. The device must be in the UNOPEN astate before executing the command.

```
# scsimgr save_attr –D /dev/pt/pt2 –a lpt_to_lockdown=0/4/1/0/0.0x500508b3009097f1.0x0
```

To display the attribute value, enter the following command:

```
# scsimgr get_attr –D /dev/pt/pt2 –a lpt_to_lockdown
.
.
.
name = lpt_to_lockdown
current = 0/4/1/0.0x0x500508b3009097f1.0x0
default =
saved =
```

Setting Lunpath Failure Detection Tunables

The mass storage subsystem monitors the I/O error statistics on lunpaths to detect failing lunpaths to LUNs.

If the lunpath continues to see I/O errors with no successful completions for a certain time interval, the SCSI stack takes the broken lunpath offline. This time interval value is defined by the *path_fail_secs* LUN attribute:

```
# scsimgr get_attr -D /dev/rdisk/disk17 –a path_fail_secs
...
name = path_fail_secs
current = 120
default = 120
saved =
```

This attribute can be set to a different value:

```
# scsimgr set_attr -D /dev/rdisk/disk17 -a path_fail_secs=100
Value of attribute path_fail_secs set successfully
```

Setting Native Multi-Pathing Support on Legacy DSFs

Applications accessing devices through legacy DSFs can take advantage of native muti-pathing. That means even if the lunpath corresponding to the legacy DSF is not available, the application can still access the device if there is at least one lunpath available.

To give more control to users seeking stricter backward compatibility, for instance those using 3rd party multi-pathing products with legacy DSFs, the settable attribute *leg_mpath_enable* enables or disables multi-pathing on legacy DSFs.

Setting the *leg_mpath_enable* attribute affects multi-pathing on legacy DSFs as follows:

- If a legacy DSF is opened when *leg_mpath_enable* is set to 'true' for the device, I/O transfer is performed on all available lunpaths according to the I/O load balancing policy. An attempt to open a legacy DSF succeeds even if the corresponding lunpath is not available.

- If a legacy DSF is opened when *leg_mpath_enable* is set to 'false' for the device, I/O transfer is only performed on the lunpath corresponding to the legacy DSF. If this lunpath becomes unavailable, I/O transfers fail. If this lunpath is not available when the legacy DSF is opened, the open fails.
- Changing the setting of *leg_mpath_enable* does not affect legacy DSFs that are already open. If multi-pathing was enabled when a legacy DSF was opened, and *leg_mpath_enable* is set to 'false' while the legacy DSF is still open, I/O transfers continue to use all available lunpaths. Likewise if multi-pathing was disabled when a legacy DSF was opened, and leg_mpath_enable is subsequently set to 'true' while the legacy DSF is still open, I/O transfers continue to use only the lunpath corresponding to the legacy DSF.

In the SAN example used in this document, disk17 has 4 lunpaths to which correspond 4 legacy DSFs. The legacy DSF `/dev/rdsk/c5t2d4` corresponds to the lunpath with the hardware path `0/2/1/0.0x500508b3009097f1.0x4014000000000000`. The following examples illustrate how multi-pathing works, depending on the setting of *leg_mpath_enable*, when an application accesses disk17 through the legacy DSF `/dev/rdsk/c5t2d4`.

All lunpaths of disk17 are used when `/dev/rdsk/c5t2d4` is opened while *leg_mpath_enable* is set to 'true'.

Display current value of *leg_mpath_enable* for disk17:

```
# scsimgr get_attr -D /dev/rdisk/disk17 -a leg_mpath_enable
        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = leg_mpath_enable
current = true
default = true
saved =
```

Clear statistics for all lunpaths of disk17:

```
# scsimgr clear_stat -D /dev/rdisk/disk17 all_lpt
scsimgr: Cleared statistics successfully
```

Check that total I/O processed on each lunpath of disk17 is now 0:

```
# scsimgr get_stat -D /dev/rdisk/disk17 all_lpt | grep "Total I/Os processed"
Total I/Os processed                             = 0
Total I/Os processed                             = 0
Total I/Os processed                             = 0
Total I/Os processed                             = 0
```

Display the I/O load balancing policy applicable for disk17:

```
# scsimgr get_attr -D /dev/rdisk/disk17 -a load_bal_policy

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = load_bal_policy
current = round_robin
default = round_robin
saved =
```

Open `/dev/rdsk/c5t2d4`, and send 120 I/O operations to disk17:

```
# dd if=/dev/rdsk/c5t2d4 of=/dev/null count=120
120+0 records in
120+0 records out
```

Check the number of I/Os sent on each lunpath of disk17:

```
# scsimgr get_stat -D /dev/rdisk/disk17 all_lpt | grep "Total I/Os processed"
Total I/Os processed                            = 40
Total I/Os processed                            = 40
Total I/Os processed                            = 40
Total I/Os processed                            = 40
```

Notes:

- About the same number of I/O transactions are processed on each lunpath of disk17. This is consistent with the round_robin I/O load balancing policy.
- The total number of I/O transfers processed on each lunpath is greater than 120/4 due to the overhead associated with opening a device.

<u>It is possible to open `/dev/rdsk/c5t2d4` and transfer data even if the corresponding lunpath is not available</u>

The cable to the HBA with hardware path 0/2/1/0 is disconnected. Lunpaths using this HBA are now unavailable as shown in the following output of `ioscan`. Their "S/W state" is now NO_HW.

```
# ioscan -kfnNC lunpath
Class     I  H/W Path  Driver S/W State   H/W Type     Description
===================================================================================
…
lunpath  16  0/2/1/0.0x500508b3009097f1.0x4014000000000000  eslpt   NO_HW     LUN_PATH
LUN path for disk17
lunpath   5  0/2/1/0.0x500508b3009097f9.0x4014000000000000  eslpt   NO_HW     LUN_PATH
LUN path for disk17
lunpath  22  0/4/1/0.0x500508b3009097f1.0x4014000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk17
lunpath  13  0/4/1/0.0x500508b3009097f9.0x4014000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk17
```

The *dd* command can open `/dev/rdsk/c5t2d4` and transfer data even if the corresponding lunpath is not available.

```
# dd if=/dev/rdsk/c5t2d4 of=/dev/null count=120
120+0 records in
120+0 records out
```

<u>Only the lunpath corresponding to `/dev/rdsk/c5t2d4` is used if *leg_mpath_enable* is set to 'false' when `/dev/rdsk/c5t2d4` is opened</u>

Disable multi-pathing for legacy DSFs of disk17:

```
# scsimgr set_attr -D /dev/rdisk/disk17 -a leg_mpath_enable=false
Value of attribute leg_mpath_enable set successfully

# scsimgr get_attr -D /dev/rdisk/disk17 –a leg_mpath_enable
        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = leg_mpath_enable
current = false
default = true
saved =
```

Clear statistics:

```
# scsimgr clear_stat -D /dev/rdisk/disk17 all_lpt
scsimgr: Cleared statistics successfully

# scsimgr get_stat –D /dev/rdisk/disk17 all_lpt | grep "Total I/Os processed"
Total I/Os processed                            = 0
Total I/Os processed                            = 0
Total I/Os processed                            = 0
Total I/Os processed                            = 0
```

Transfer data through `/dev/rdsk/c5t2d4`:

```
# dd if=/dev/rdsk/c5t2d4 of=/dev/null count=120
120+0 records in
120+0 records out

# scsimgr get_stat -D /dev/rdisk/disk17 all_lpt | grep "Total I/Os processed"
Total I/Os processed                          = 130
Total I/Os processed                          = 0
Total I/Os processed                          = 0
Total I/Os processed                          = 0
```

Notes: I/O transfers are performed only on the lunpath corresponding to `/dev/rdsk/c5t2d4`.

`/dev/rdsk/c5t2d4` cannot be opened if the corresponding lunpath is not available and leg_mpath_enable is set to 'false'

While multi-pathing is disabled for disk17, the cable to HBA 0/2/1/0 is disconnected. As shown below *dd* can not open `/dev/rdsk/c5t2d4`.

```
# scsimgr get_attr -D /dev/rdisk/disk17 -a leg_mpath_enable
        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = leg_mpath_enable
current = false
default = true
saved =

# ioscan -kfnNC lunpath
Class     I  H/W Path  Driver S/W State   H/W Type     Description
=====================================================================
…
lunpath  16  0/2/1/0.0x500508b3009097f1.0x4014000000000000  eslpt   NO_HW     LUN_PATH
LUN path for disk17
lunpath   5  0/2/1/0.0x500508b3009097f9.0x4014000000000000  eslpt   NO_HW     LUN_PATH
LUN path for disk17
lunpath  22  0/4/1/0.0x500508b3009097f1.0x4014000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk17
lunpath  13  0/4/1/0.0x500508b3009097f9.0x4014000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk17

# dd if=/dev/rdsk/c5t2d4 of=/dev/null count=120
/dev/rdsk/c5t2d4: No such device or address
dd: cannot open /dev/rdsk/c5t2d4
```

Changes to *leg_mpath_enable* do not affect legacy DSF that is already open

The following example shows how changing *leg_mpath_enable* while `/dev/rdsk/c5t2d4` is open does not affect the lunpaths used for I/O transfer on `/dev/rdsk/c5t2d4` as long as it remains opened.

Show current value of *leg_mpath_enable* for disk17:

```
# scsimgr get_attr -D /dev/rdisk/disk17 -a leg_mpath_enable

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = leg_mpath_enable
current = true
default = true
saved =
```

Clear statistics:

```
# scsimgr clear_stat -D /dev/rdisk/disk17 all_lpt
scsimgr: Cleared statistics successfully
```

```
# scsimgr get_stat -D /dev/rdisk/disk17 all_lpt | grep "Total I/Os processed"
Total I/Os processed                               = 0
Total I/Os processed                               = 0
Total I/Os processed                               = 0
Total I/Os processed                               = 0
```

Open `/dev/rdsk/c5t2d4` and transfer data:

```
# dd if=/dev/rdsk/c5t2d4 of=/dev/null count=100000000 &
[1]     17998
#
# scsimgr get_stat -D /dev/rdisk/disk17 all_lpt | grep "Total I/Os processed"
Total I/Os processed                               = 18071
Total I/Os processed                               = 17988
Total I/Os processed                               = 17988
Total I/Os processed                               = 17988
```

Set *leg_mpath_enable* to 'false' while `/dev/rdsk/c5t2d4` is open:

```
# scsimgr set_attr -D /dev/rdisk/disk17 -a leg_mpath_enable=false
Value of attribute leg_mpath_enable set successfully

# scsimgr get_attr -D /dev/rdisk/disk17 -a leg_mpath_enable

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk17

name = leg_mpath_enable
current = false
default = true
saved =
```

Check whether I/O transfers are performed on all lunpaths of disk17 even if multi-pathing is now disabled for legacy DSFs of disk17.

```
# scsimgr get_stat -D /dev/rdisk/disk17 all_lpt | grep "Total I/Os processed"
Total I/Os processed                               = 73881
Total I/Os processed                               = 73780
Total I/Os processed                               = 73880
Total I/Os processed                               = 73780
```

Configuring Automatic Path Recovery

By default failed lunpath automatic recovery uses the basic ping and immediate path recovery policies. This section gives examples of how the default configuration can be changed to meet requirements for different use cases.

Five settable attributes control how automatic path recovery is performed. For more information see the *scsimgr_esdisk*(7) man page:

- *ping_type*: path ping policy (none, basic, or extended)
- *ping_recovery*: path recovery policy (immediate, count_based, or time_based)
- *ping_count_threshold*: minimum number of successful pings before the path is considered back online, when the ping recover policy is count-based.
- *ping_time_threshold*: minimum time in seconds pings should consistently succeed before the path is considered back online.

Note that this is only valid for the March 2008 release of HP-UX 11i v3 or later.

Disabling Automatic Path Recovery

To disable automatic path recovery for disk16, use the following command:

```
# scsimgr set_attr -D /dev/rdisk/disk16 -a ping_type=none
Value of attribute ping_type set successfully
```

Configuring Extended Path Ping

In this example the user wants to configure extended path ping for all disk devices with the same vendor identifier as disk17. For that purpose the user has registered a settable attribute scope for disk devices with the appropriate vendor identifier and set the ping_type attribute for this scope.

Determine the settable attribute scope name:

```
# scsimgr ddr_name -D /dev/rdisk/disk17 vid
        SETTABLE ATTRIBUTE SCOPE

"/escsi/esdisk/0x0/COMPAQ  "
```

Register the settable attribute scope name if it is not yet registered:

```
# scsimgr ddr_add -N "/escsi/esdisk/0x0/COMPAQ  "
scsimgr:WARNING: Adding a settable attribute scope may impact system operation if some
attribute values are changed at this scope.
Do you really want to continue? (y/n)? y
scsimgr: DDR entry '/escsi/esdisk/0x0/COMPAQ  ' added successfully
```

Change the value of the *ping_type* attribute to 'extended' for "/escsi/esdisk/0x0/COMPAQ ":

```
# scsimgr set_attr –N "/escsi/esdisk/0x0/COMPAQ  " -a ping_type=extended
Value of attribute ping_type set successfully
```

Setting a Count-based Deferred Path Recovery Policy

In this example the user wants failed lunpaths of disk17 to be considered back online only after 20 successful pings. We assume that the *ping_type* attribute is already set to 'basic' or 'extended' for disk17.

```
# scsimgr set_attr -D /dev/rdisk/disk17 –a ping_recovery=count_based -a
ping_count_threshold=20
Value of attribute ping_recovery set successfully
Value of attribute ping_count_threshold set successfully
```

Setting a Time-based Deferred Path Recovery Policy

In this example the user wants failed lunpaths of disk17 to be considered back online only if pings continually succeed for 10 seconds. We assume that the *ping_type* attribute is already set to 'basic' or 'extended' for disk17.

```
# scsimgr set_attr -D /dev/rdisk/disk17 –a ping_recovery=time_based -a
ping_time_threshold=10
Value of attribute ping_recovery set successfully
Value of attribute ping_time_threshold set successfully
```

Native Multi-Pathing: A Concrete Example

This example illustrates what happens when a target port is disconnected and reconnected while I/O operations are flowing. This example illustrates:

- Error reporting
- Asynchronous target port offline notification
- `ioscan` and `scsimgr` use cases to display multi-pathing information
- Lunpath failover
- Lunpath recovery

First, initiate some I/O traffic:

```
# dd if=/dev/rdisk/disk15 of=/dev/rdisk/disk16 &
```

Then, the target port of world wide name `0x500508b3009097f9` is physically disconnected.

The administrator is notified via a message in `/var/adm/syslog/syslog`:

```
Jan 31 02:08:50 hpfcs835 vmunix: class : tgtpath, instance 3
Jan 31 02:08:50 hpfcs835 vmunix: Target path (class=tgtpath, instance=3) has gone offline.
The target path h/w path is 0/2/1/0.0x500508b3009097f9
Jan 31 02:08:50 hpfcs835 vmunix:
Jan 31 02:08:50 hpfcs835 vmunix: class : tgtpath, instance 6
Jan 31 02:08:50 hpfcs835 vmunix: Target path (class=tgtpath, instance=6) has gone offline.
The target path h/w path is 0/4/1/0.0x500508b3009097f9
```

Since the target port SAN element is involved in the definition of two target paths, both target paths went offline.

The `ioscan` command shows that two target paths have gone to `NO_HW` state:

```
# ioscan -fnNC tgtpath

Class      I  H/W Path       Driver S/W State   H/W Type     Description

…

tgtpath    7  0/2/1/0.0x500508b3009097f1  estp   CLAIMED     TGT_PATH
fibre_channel target served by td driver

tgtpath    3  0/2/1/0.0x500508b3009097f9  estp   NO_HW       TGT_PATH
fibre_channel target served by td driver

tgtpath    8  0/4/1/0.0x500508b3009097f1  estp   CLAIMED     TGT_PATH
fibre_channel target served by td driver

tgtpath    6  0/4/1/0.0x500508b3009097f9  estp   NO_HW       TGT_PATH
fibre_channel target served by td driver
```

Using `scsimgr`, gather statistics for offline events on the target path:

```
# scsimgr get_stat -H 0/4/1/0.0x500508b3009097f9
...
Target path offline events from I/F driver       = 1
```

As a result of the target port `0x500508b3009097f9` being disconnected, all the lunpaths using this SAN component become unavailable. The `ioscan` command shows these lunpaths with a NO_HW state:

```
# ioscan -kfnNC lunpath
Class      I  H/W Path  Driver S/W State   H/W Type     Description
==================================================================
lunpath  14  0/0/2/0.0.0x0.0x0                         eslpt   CLAIMED     LUN_PATH
LUN path for disk20
lunpath   7  0/1/1/0.0x0.0x0                           eslpt   CLAIMED     LUN_PATH
LUN path for disk19
lunpath   6  0/1/1/0.0x1.0x0                           eslpt   CLAIMED     LUN_PATH
LUN path for disk18
lunpath  15  0/2/1/0.0x500508b3009097f1.0x0            eslpt   CLAIMED     LUN_PATH
LUN path for ctl2
```

```
lunpath  16  0/2/1/0.0x500508b3009097f1.0x4014000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk17
lunpath  19  0/2/1/0.0x500508b3009097f1.0x4015000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk16
lunpath  17  0/2/1/0.0x500508b3009097f1.0x4016000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk13
lunpath  18  0/2/1/0.0x500508b3009097f1.0x4017000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk15
lunpath  20  0/2/1/0.0x500508b3009097f1.0x4018000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk14
lunpath   0  0/2/1/0.0x500508b3009097f9.0x0                 eslpt   NO_HW       LUN_PATH
LUN path for ctl2
lunpath   5  0/2/1/0.0x500508b3009097f9.0x4014000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk17
lunpath   4  0/2/1/0.0x500508b3009097f9.0x4015000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk16
lunpath   1  0/2/1/0.0x500508b3009097f9.0x4016000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk13
lunpath   3  0/2/1/0.0x500508b3009097f9.0x4017000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk15
lunpath   2  0/2/1/0.0x500508b3009097f9.0x4018000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk14
lunpath  21  0/4/1/0.0x500508b3009097f1.0x0                 eslpt   CLAIMED     LUN_PATH
LUN path for ctl2
lunpath  22  0/4/1/0.0x500508b3009097f1.0x4014000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk17
lunpath  23  0/4/1/0.0x500508b3009097f1.0x4015000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk16
lunpath  26  0/4/1/0.0x500508b3009097f1.0x4016000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk13
lunpath  24  0/4/1/0.0x500508b3009097f1.0x4017000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk15
lunpath  25  0/4/1/0.0x500508b3009097f1.0x4018000000000000  eslpt   CLAIMED     LUN_PATH
LUN path for disk14
lunpath   8  0/4/1/0.0x500508b3009097f9.0x0                 eslpt   NO_HW       LUN_PATH
LUN path for ctl2
lunpath  13  0/4/1/0.0x500508b3009097f9.0x4014000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk17
lunpath  11  0/4/1/0.0x500508b3009097f9.0x4015000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk16
lunpath  12  0/4/1/0.0x500508b3009097f9.0x4016000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk13
lunpath  10  0/4/1/0.0x500508b3009097f9.0x4017000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk15
lunpath   9  0/4/1/0.0x500508b3009097f9.0x4018000000000000  eslpt   NO_HW       LUN_PATH
LUN path for disk14
```

The health of the affected lunpaths and LUNs changes too. The affected lunpaths' health becomes "offline" and affected LUNs' health becomes "limited" (see section Lunpath and LUN Health for more information on health).

```
# ioscan -P health
Class        I  H/W Path        health
=======================================================================================
…
tgtpath      7  0/2/1/0.0x500508b3009097f1                     online
lunpath     15  0/2/1/0.0x500508b3009097f1.0x0                 online
lunpath     16  0/2/1/0.0x500508b3009097f1.0x4014000000000000  online
lunpath     19  0/2/1/0.0x500508b3009097f1.0x4015000000000000  online
lunpath     17  0/2/1/0.0x500508b3009097f1.0x4016000000000000  online
lunpath     18  0/2/1/0.0x500508b3009097f1.0x4017000000000000  online
lunpath     20  0/2/1/0.0x500508b3009097f1.0x4018000000000000  online
tgtpath      3  0/2/1/0.0x500508b3009097f9                     offline
lunpath      0  0/2/1/0.0x500508b3009097f9.0x0                 offline
lunpath      5  0/2/1/0.0x500508b3009097f9.0x4014000000000000  offline
lunpath      4  0/2/1/0.0x500508b3009097f9.0x4015000000000000  offline
lunpath      1  0/2/1/0.0x500508b3009097f9.0x4016000000000000  offline
lunpath      3  0/2/1/0.0x500508b3009097f9.0x4017000000000000  offline
lunpath      2  0/2/1/0.0x500508b3009097f9.0x4018000000000000  offline
ba           3  0/3             N/A
slot         1  0/3/1           N/A
fc           1  0/3/1/0         online
fc           2  0/3/1/1         online
ba           4  0/4             N/A
slot         2  0/4/1           N/A
fc           3  0/4/1/0         online
```

```
tgtpath      8  0/4/1/0.0x500508b3009097f1                     online
lunpath     21  0/4/1/0.0x500508b3009097f1.0x0                 online
lunpath     22  0/4/1/0.0x500508b3009097f1.0x4014000000000000  online
lunpath     23  0/4/1/0.0x500508b3009097f1.0x4015000000000000  online
lunpath     26  0/4/1/0.0x500508b3009097f1.0x4016000000000000  online
lunpath     24  0/4/1/0.0x500508b3009097f1.0x4017000000000000  online
lunpath     25  0/4/1/0.0x500508b3009097f1.0x4018000000000000  online
tgtpath      6  0/4/1/0.0x500508b3009097f9                     offline
lunpath      8  0/4/1/0.0x500508b3009097f9.0x0                 offline
lunpath     13  0/4/1/0.0x500508b3009097f9.0x4014000000000000  offline
lunpath     11  0/4/1/0.0x500508b3009097f9.0x4015000000000000  offline
lunpath     12  0/4/1/0.0x500508b3009097f9.0x4016000000000000  offline
lunpath     10  0/4/1/0.0x500508b3009097f9.0x4017000000000000  offline
lunpath      9  0/4/1/0.0x500508b3009097f9.0x4018000000000000  offline
…
disk        13  64000/0xfa00/0x1                               limited
disk        14  64000/0xfa00/0x2                               limited
disk        15  64000/0xfa00/0x3                               limited
disk        16  64000/0xfa00/0x4                               limited
disk        17  64000/0xfa00/0x5                               limited
```

Using the `scsimgr` command, gather statistics for offline events and I/O errors on the lunpath path:

```
# scsimgr get_stat -H 0/2/1/0.0x500508b3009097f9.0x4014000000000000
...
Offline events                                   = 1
Online events                                    = 0
...
I/O transfer Statistics:

Bytes read                                       = 1426794
Bytes written                                    = 0
Total I/Os processed                             = 12215
I/O failures                                     = 1
Retried I/Os                                     = 2
Retried I/O failures                             = 0
...
```

Some I/O operation errors are expected since I/O operations may have been pending on the lunpaths that were taken offline when the target port was disconnected. However, **it is critical to observe that the transfer of data is uninterrupted due to lunpath failover**. This can be verified running the `sar` command which shows I/O activity on the remaining two active lunpaths for disk15 and disk16:

```
# sar -R -L -d 1 5

HP-UX hpfcs835 B.11.31 U ia64    02/01/07

03:38:16        lunpath  %busy   avque    r/s     w/s  blks/s  avwait  avserv
                          %age     num    num     num     num    msec    msec
                 device  %busy   avque    r/s     w/s  blks/s  avwait  avserv
       disk15_lunpath18  27.00    0.50    534       0     534    0.00    0.51
       disk16_lunpath19  17.00    0.50      0     533     533    0.00    0.33
       disk16_lunpath23  18.00    0.50      0     534     534    0.00    0.32
       disk15_lunpath24  27.00    0.50    533       0     533    0.00    0.51
                 disk15  55.00    0.50   1066       0    1066    0.00    0.51
                 disk16  35.00    0.50      0    1067    1067    0.00    0.33
```

Now, the target port of world wide name `0x500508b3009097f9` is reconnected.

The following messages are immediately logged in `/var/adm/syslog/syslog.log`:

```
Jan 31 02:41:48 hpfcs835 vmunix: class : tgtpath, instance 3
Jan 31 02:41:48 hpfcs835 vmunix: Target path (class=tgtpath, instance=3) has gone online.
The target path h/w path is 0/2/1/0.0x500508b3009097f9
Jan 31 02:41:48 hpfcs835 vmunix: class : tgtpath, instance 6
Jan 31 02:41:48 hpfcs835 vmunix: Target path (class=tgtpath, instance=6) has gone online.
The target path h/w path is 0/4/1/0.0x500508b3009097f9
~
```

As a result of the target port `0x500508b3009097f9` coming back online, all the lunpaths using this SAN component become available again. The `ioscan` command shows these lunpaths with a CLAIMED state. Using `scsimgr`, gather some statistics for online events on the target path and lunpath path:

```
# scsimgr get_stat -H 0/4/1/0.0x500508b3009097f9.0x4014000000000000

        STATISTICS FOR LUN path : 0/4/1/0.0x500508b3009097f9.0x4014000000000000

Generic Statistics:
…
Offline events                                   = 1
Online events                                    = 1
…
```

Using `sar`, observe that I/O operations are still going from disk15 to disk16 on all lunpaths:

```
# sar -L 1 5

HP-UX hpfcs835 B.11.31 U 9000/800    02/01/07

16:41:10       lunpath   %busy   avque     r/s     w/s  blks/s  avwait   avser
                          %age     num     num     num     num    msec    msec
       disk15_lunpath18    7.92    0.50     122       0     122    0.00     0.6
        disk15_lunpath3    5.94    0.50       0     122     122    0.00     0.4
       disk15_lunpath24   23.76    0.50       0     121     121    0.00     1.9
       disk15_lunpath10    9.90    0.50     122       0     122    0.00     0.8
       disk16_lunpath19    5.94    0.50     122       0     122    0.00     0.4
        disk16_lunpath4   15.84    0.50     121       0     121    0.00     1.2
       disk16_lunpath23   21.78    0.50       0     122     122    0.00     1.7
       disk16_lunpath11    7.92    0.50       0     122     122    0.00     0.6
…
```

The `ioscan` command shows that LUN and lunpath health is set back to online automatically.

```
# ioscan -P health
Class        I  H/W Path       health
================================================================================
…
tgtpath      7  0/2/1/0.0x500508b3009097f1                          online
lunpath     15  0/2/1/0.0x500508b3009097f1.0x0                      online
lunpath     16  0/2/1/0.0x500508b3009097f1.0x4014000000000000  online
lunpath     19  0/2/1/0.0x500508b3009097f1.0x4015000000000000  online
lunpath     17  0/2/1/0.0x500508b3009097f1.0x4016000000000000  online
lunpath     18  0/2/1/0.0x500508b3009097f1.0x4017000000000000  online
lunpath     20  0/2/1/0.0x500508b3009097f1.0x4018000000000000  online
tgtpath      3  0/2/1/0.0x500508b3009097f9                          online
lunpath      0  0/2/1/0.0x500508b3009097f9.0x0                      online
lunpath      5  0/2/1/0.0x500508b3009097f9.0x4014000000000000  online
lunpath      4  0/2/1/0.0x500508b3009097f9.0x4015000000000000  online
lunpath      1  0/2/1/0.0x500508b3009097f9.0x4016000000000000  online
lunpath      3  0/2/1/0.0x500508b3009097f9.0x4017000000000000  online
lunpath      2  0/2/1/0.0x500508b3009097f9.0x4018000000000000  online
ba           3  0/3            N/A
slot         1  0/3/1          N/A
fc           1  0/3/1/0        online
fc           2  0/3/1/1        online
ba           4  0/4            N/A
slot         2  0/4/1          N/A
fc           3  0/4/1/0        online
tgtpath      8  0/4/1/0.0x500508b3009097f1                          online
lunpath     21  0/4/1/0.0x500508b3009097f1.0x0                      online
lunpath     22  0/4/1/0.0x500508b3009097f1.0x4014000000000000  online
lunpath     23  0/4/1/0.0x500508b3009097f1.0x4015000000000000  online
lunpath     26  0/4/1/0.0x500508b3009097f1.0x4016000000000000  online
lunpath     24  0/4/1/0.0x500508b3009097f1.0x4017000000000000  online
lunpath     25  0/4/1/0.0x500508b3009097f1.0x4018000000000000  online
tgtpath      6  0/4/1/0.0x500508b3009097f9                          online
lunpath      8  0/4/1/0.0x500508b3009097f9.0x0                      online
lunpath     13  0/4/1/0.0x500508b3009097f9.0x4014000000000000  online
lunpath     11  0/4/1/0.0x500508b3009097f9.0x4015000000000000  online
lunpath     12  0/4/1/0.0x500508b3009097f9.0x4016000000000000  online
lunpath     10  0/4/1/0.0x500508b3009097f9.0x4017000000000000  online
lunpath      9  0/4/1/0.0x500508b3009097f9.0x4018000000000000  online
```

```
…
disk        13  64000/0xfa00/0x1                                  online
disk        14  64000/0xfa00/0x2                                  online
disk        15  64000/0xfa00/0x3                                  online
disk        16  64000/0xfa00/0x4                                  online
disk        17  64000/0xfa00/0x5                                  online
```

This illustrates the fact that as the target port comes back online, the related lunpaths automatically become active. As a result, they are automatically used again for I/O traffic.

For More Information

To learn more about mass storage on HP-UX 11i v3 see the following documents on the HP documentation websites (http://docs.hp.com and http://docs.hp.com/en/netsys.html#Storage%20Area%20Management):

- The Next Generation Mass Storage Stack
- HP-UX 11i v3 Mass Storage Device Naming
- HP-UX 11i v3 Native Multi-Pathing for Mass Storage
- HP-UX 11i v3 Mass Storage I/O Performance Improvements
- scsimgr SCSI Management and Diagnostics Utility
- HP-UX 11i v3 Persistent DSF Migration Guide
- HP-UX 11i v3 Writing Active/Passive Switch Plug-Ins
- LVM Online Disk Replacement (LVM OLR)
- When Good Disks Go Bad: Dealing with Disk Failures under LVM
- Migrating from HP Storage Works Secure Path for Active-Active Disk Arrays to Native Multi-pathing in HP-UX 11i v3

Call to Action

HP welcomes your input. Please give us comments about this white paper, or suggestions for mass storage or related documentation, through our technical documentation feedback website: http://docs.hp.com/en/feedback.html

09/2009