

HP-UX 11i v3 Mass Storage I/O Performance Improvements

February, 2007



1	Introduction	2
2	Terms and Definitions	2
3	Native Multi-Pathing	3
4	Boot/Scan Improvements	6
5	Crash Dump Performance Improvements	7
6	Improved Performance Tracking Tools	8
7	New SCSI Tunables	9
8	LVM	10
9	Async Disk	10
10	References	11
11	For more information	11

1 Introduction

HP-UX 11i v3 introduces a new mass storage subsystem which provides improvements in manageability, availability, scalability, and performance. This paper discusses the corresponding mass storage I/O performance improvements and related capabilities, including:

- Native multi-pathing, which automatically takes advantage of multiple hardware paths to increase I/O throughput
- Boot/scan improvements, which decrease boot and scan times
- Crash dump performance improvements, via parallelization of the dump process
- Improved performance tracking, reporting tools, and statistics
- New and more flexible performance related tunables
- Logical Volume Manager (LVM) performance improvements
- Async Disk driver performance improvements

These performance improvements are built-in to HP-UX 11i v3 and do not require the purchase or installation of add-on products to obtain the performance benefits. Each of these will be discussed in the sections which follow.

2 Terms and Definitions

Async Disk Driver The Asynchronous Disk Pseudo Driver in HP-UX, named `asyncdsk` in the OS, is used to perform asynchronous I/O in a user application for improved performance. User processes or threads in which asynchronous I/O is performed do not need to wait for the I/O to complete before going on to do other work. Instead they are asynchronously notified of I/O completions. The Async Disk Driver is commonly used with the Oracle data base application on HP-UX.

DSF Device Special File.

HBA Host Bus Adapter. E.g., an I/O card with one or more ports on it for attachment to Fibre Channel, parallel SCSI, or other mass storage connectivity to a device.

LUN An end device in a mass storage interconnect.

LUN path A path to a LUN through an HBA port and a target port. Also known as a "lunpath".

Multi-pathing The ability to find the various paths to a LUN, and failover to an alternate path when a given path fails, and/or to load-balance across the various paths. HP-UX 11i v3 provides native multi-pathing which is built-in to the mass storage stack.

Parallel Dump A new crash dump feature in HP-UX 11i v3 which allows the dump process to be parallelized to produce significantly reduced dump times.

Target A storage device attachment to a mass storage interconnect such as Fibre Channel or SCSI.

3 Native Multi-Pathing

Multi-pathing is the ability to use multiple paths to a LUN to provide the following benefits:

- Availability: transparent recovery from path failures via failover to an alternate path.
- Performance: increased I/O performance via load-balancing of I/O requests across available paths.

HP-UX 11i v3 provides native multi-pathing built-in to the operating system. Native multi-pathing has additional manageability benefits, such as transparently handling changes in the SAN without the need for reconfiguration. See the Native Multi-Pathing white paper referenced in section 10 for functional and usage details. This paper discusses the performance benefits of the HP-UX 11i v3 native multi-pathing.

If there are multiple hardware paths from the host to a LUN (e.g., via multiple HBA ports or multiple target ports), the native multi-pathing will transparently distribute I/O requests across all available paths to the LUN¹, using a choice of load-balancing policies. Load-balancing policies determine how a path is chosen for each I/O request, and include the following:

- “round-robin” policy: a path is selected in a round-robin fashion from the list of available paths. This is the default policy for random access devices.
- “least-command-load” policy: a path with the least number of outstanding I/O requests is selected.
- “cell-local round-robin” policy: a path belonging to the cell (in cell-based servers) on which the I/O request was initiated is selected.
- “path lock down” policy: a single specified path is selected for all I/O requests to the LUN. This is primarily used for sequential access (e.g., tape) devices.
- “preferred path” policy: a path belonging to a user-specified “preferred” path is selected. This is similar to “path lockdown” except that it also provides for automatic path failover when the preferred path fails.

The load-balancing policies operate on the list of available paths to a LUN, which can change as paths go offline or online due to error conditions or by a user disabling paths via the `scsimgr` command.²

The performance benefits of load-balancing include the following:

- Increased I/O throughput via balanced utilization of all available paths.
- Decreased CPU utilization (and increased aggregate throughput capability) on cell-based machines using cell-local load-balancing.

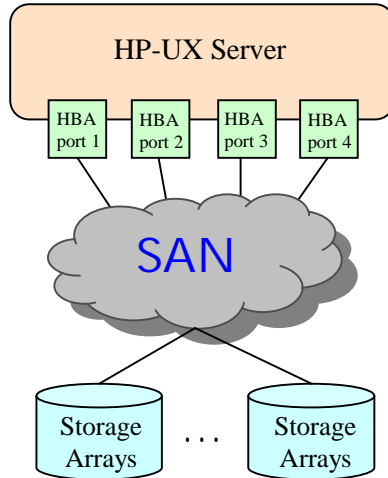
3.1 Increased I/O Throughput

The use of load-balancing can significantly increase the I/O throughput capability of a system. Consider, for example, a system with four HBA ports, each of which is connected to a set of LUNs in a set of storage arrays, as depicted in Figure 1.

¹ The distribution of I/O requests across all available paths occurs by default for random access type devices (e.g., disks). The “round-robin” policy is default for random access devices. The path lock down policy is the default for tapes, autochangers, and other types of devices.

² See the `scsimgr(1m)` man page or the white paper on the `scsimgr` utility referenced in section 10.

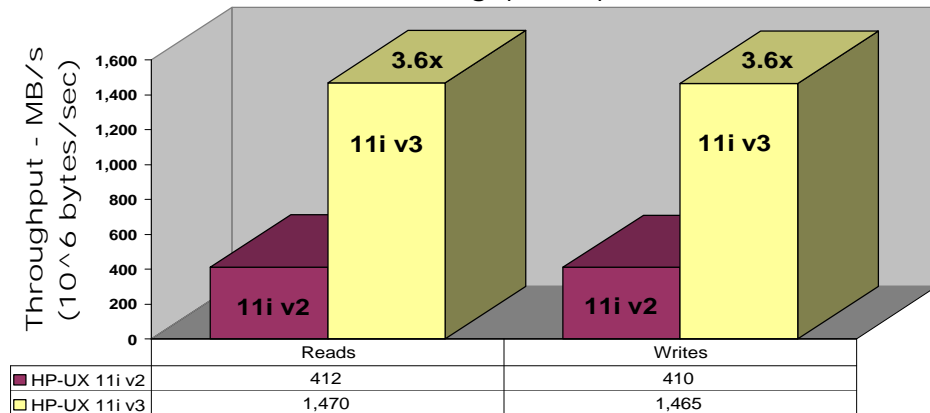
Figure 1 – System with 4 paths per LUN



Without load-balancing, I/O requests may be overloaded on some paths while other paths are under-utilized. Load-balancing spreads the load evenly across the available paths to make use of the full bandwidth of the various paths and avoid degradations associated with overloading a specific path. This can be especially important in high workload situations or where consistent service levels are required.

HP testing shows significant improvements in throughput and I/Os per second (IOPS) when load-balancing policies are enabled (which they are by default in HP-UX 11i v3).³ Chart 1 shows I/O throughput results on a system with a configuration similar to that in Figure 1. On 11i v2, or with load-balancing disabled in 11i v3, all the I/O goes through the one HBA port. On 11i v3, with load-balancing enabled, the I/O gets balanced automatically across paths through all four HBA ports, producing in this case about 3.6 times the throughput of the load-balancing disabled case. Careful static balancing of LUN path usage in an application or workload can produce similar results, but requires much more configuration and storage management work and is not guaranteed to continue to provide optimal results across configuration or workload changes.

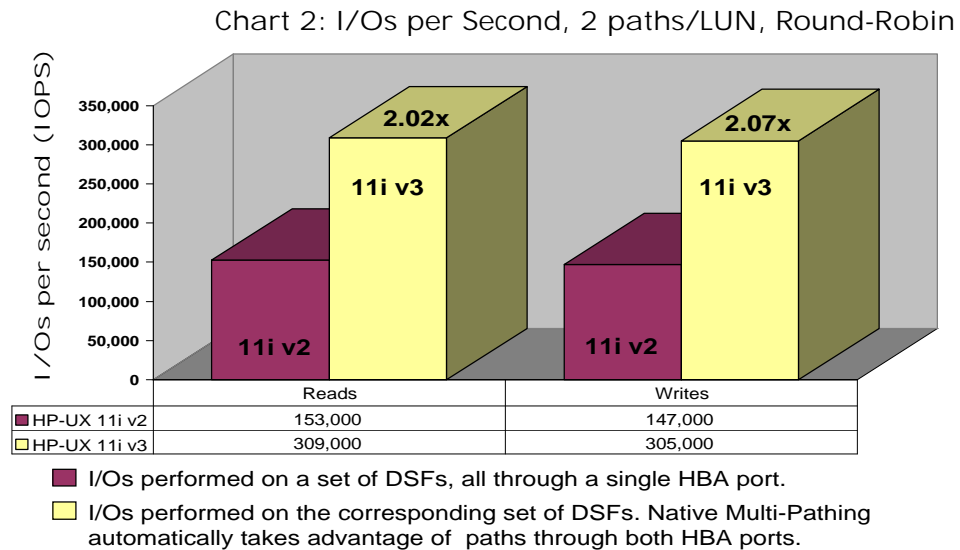
Chart 1: I/O Throughput, 4 paths/LUN, Round-Robin



- I/Os performed on a set of DSFs, all through a single HBA port.
- I/Os performed on the corresponding set of DSFs. Native Multi-Pathing automatically takes advantage of paths through the other 3 HBA ports.

³ These tests in Charts 1 and 2 were performed using the diskbench I/O benchmark tool on an rx6600 server with 4Gb/s AB379B Fibre Channel adapters connected to MSA1500 Fibre Channel disks. For comparison purposes, the set of DSFs used on 11i v2 were also used to perform the tests on 11i v3. The 11i v3 tests were also performed using the corresponding new persistent DSFs, without any change in the results. See the Device Naming white paper referenced in section 10 for details regarding the DSF naming in HP-UX 11i v3. The actual I/O performance increase experienced on a customer system will depend on the system workload and the mass storage hardware and software configuration.

Chart 2 shows I/Os per second (IOPS) results on a system with 2 paths (via 2 HBA ports) per LUN, resulting in a doubling of the performance with load-balancing enabled as compared to HP-UX 11i v2. These results show slightly more than 2x improvement due to cache locality and other tuning in the mass storage stack and the kernel in HP-UX 11i v3.



3.1.1 Least-Command-Load versus Round-Robin

The HP-UX 11i v3 results in Charts 1 and 2 used the default load-balancing policy, round-robin.

The least-command-load policy generally has higher CPU utilization than round-robin, but this is typically only significant in small I/O size workloads of 8K I/O size or less. The CPU utilization difference increases somewhat as the I/O load (average number of I/O requests outstanding) per LUN increases and as the number of paths increase.

Least-command-load can have an advantage on workloads with a mixture of I/O sizes which are in progress simultaneously, or in configurations with significant variations in path performance. This is due to the fact that least-command-load tends to balance the load better in the presence of such inequalities than the round-robin approach. For example, a workload with a significant mixture of I/O requests of different sizes (e.g., 4K, 8K, and 16K) which tend to be in progress simultaneously may see increased I/Os per second with least-command-load. Similarly, a configuration similar to Figure 1 in which HBA ports 1 and 2 are on 2Gb/s HBAs and ports 3 and 4 are on 4Gb/s HBAs may benefit from least-command-load if the workload is sufficient to sustain I/O on multiple paths simultaneously.

Neither least-command-load nor round-robin is recommended on cell-based machines when the paths are spread across cells. Cell-local round robin should be used instead, as discussed in the next section.

3.2 Decreased CPU Utilization in Cell-based Systems

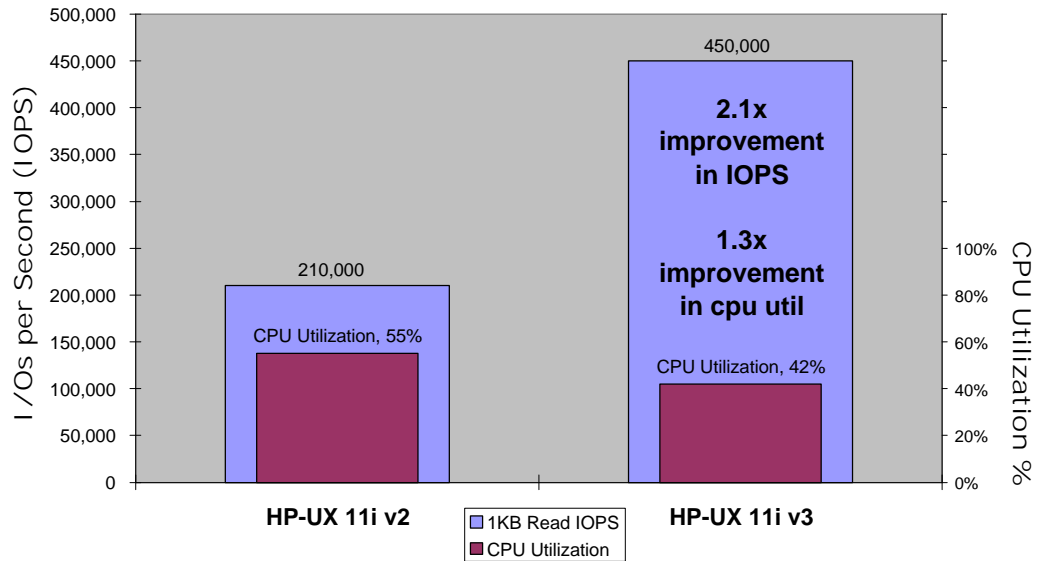
On a cell-based system, inter-cell overheads make the use of cell-local round-robin in HP-UX 11i v3 a significant advantage even in a completely statically balanced configuration and workload. Chart 3 compares IOPS on HP-UX 11i v2 with cell-local round-robin on HP-UX 11i v3. In these HP tests on a cell-based server⁴, HP-UX 11i v3 showed significant IOPS improvements:

⁴ The tests in Chart 3 were performed using the diskbench I/O benchmark tool on a 16-core, 4-cell rx8640 server with eight 4Gb/s Fibre Channel adapter ports connected to MSA1000 Fibre Channel Disks (8 paths/LUN).

2.1 times the IOPS of 11i v2. This is due to the decreased CPU utilization that results from better memory locality in the I/O path. In addition to the benefits of cell-local round-robin, which is recommended on cell-based systems, a significant portion of the speed-up is also due to better cache-locality algorithms in the new mass storage stack and the rest of the HP-UX 11i v3 kernel.

The cell-local round robin policy requires at least one LUN path per cell for optimal performance. When this requirement is met the cell-local round-robin can significantly improve the overall I/O performance while decreasing the CPU overhead and scaling well with the number of cells.

Chart 3: I/Os per second, 8 paths/LUN, Cell-Local Round-robin



Note: Chart 3, which used a statically balanced workload, cannot be compared with Chart 2, which used a statically unbalanced workload, and had a number of other differences in the servers and mass storage configurations. In a statically unbalanced workload the improvement from HP-UX 11i v2 to 11i v3 in Chart 3 would have been much larger.

4 Boot/Scan Improvements

The HP-UX 11i v3 mass storage stack fully parallelizes the probing of HBAs, targets, and LUNs to reduce I/O scan and system boot times. LUN opens and closes are also parallelized to further reduce scan and boot times and other operations which are open/close intensive. The mass storage stack also enables asynchronous detection of SAN configuration changes and dynamic re-scanning of the SAN along with automatic creation of new device files to more quickly enable new hardware.

Chart 4 and Chart 5 display the results of HP testing on an rx8620 server with 19,200 lunpaths, showing significant improvements in scan and boot times on HP-UX 11i v3.⁵ The shutdown and reboot time in Chart 4 is the total time from initiating the reboot (e.g. via the reboot command) until the system is booted back up to the console login. The use of the new `-N` option to `ioscan`, `'ioscan -kfnN'`, is included in the table for comparison with the legacy `'ioscan -kfn'` command

⁵ These boot and scan times were obtained on an rx8620 server with 8 processor cores and 2.8GB memory, with two 2Gb/s A6795A and six 4Gb/s AB379B Fibre Channel adapter ports each connected to 600 LUNs in an XP disk array. The 8 FC HBA ports were connected through a switch to 4 target ports on the XP array, for a total of 8x4=32 paths per LUN, and a total of 600x32=19,200 LUN paths on the system. The actual boot and scan times experienced on a customer system depend on the mass storage hardware and system configuration.

option. The 'ioscan -kfnN' takes less time because the -N output results in fewer DSFs to display: 600 new persistent DSFs (corresponding to the 600 LUNs on the system) versus 19,200 legacy DSFs (corresponding to the 19,200 lunpaths).⁶

Chart 4: Boot/Scan Times

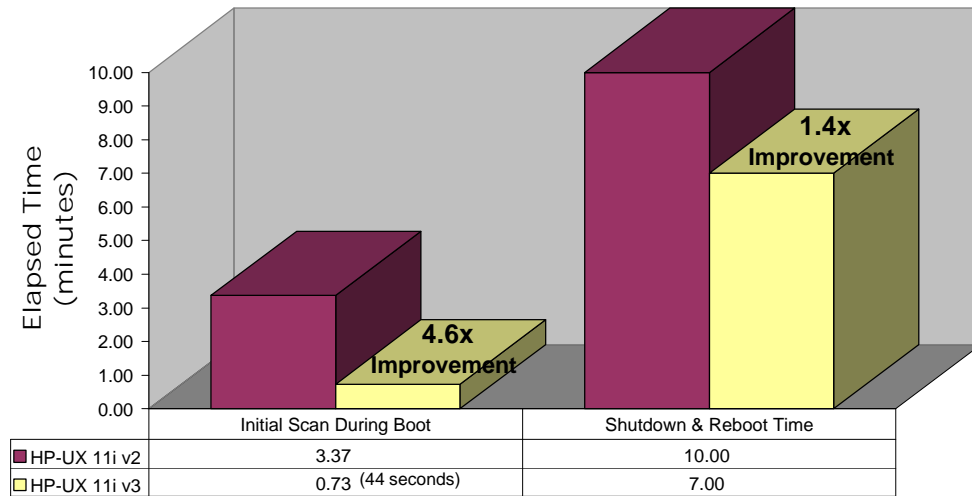
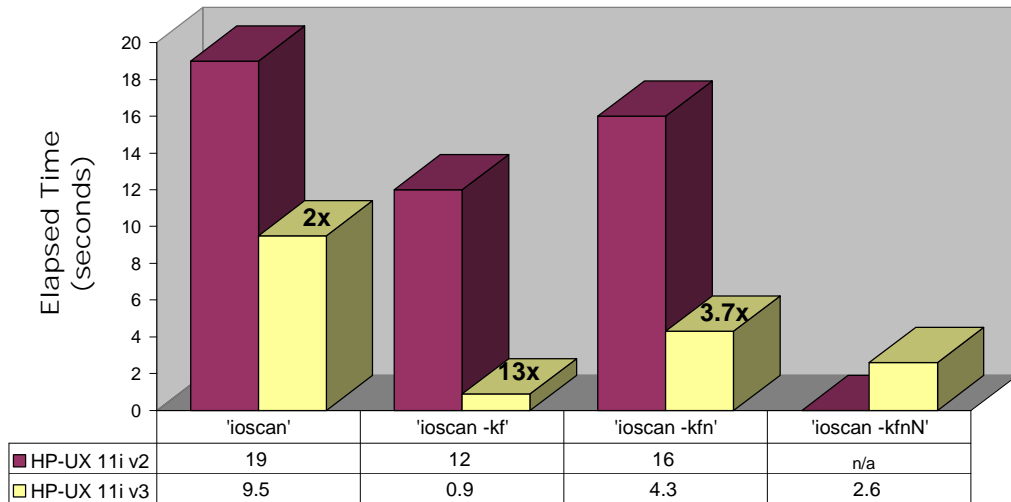


Chart 5: Run-Time ioscan times



Tests: ioscan Hardware scan
ioscan -kf Scan kernel I/O system data structures
ioscan -kfn Kernel scan + legacy DSFs
ioscan -kfnN Kernel scan + new persistent DSFs

5 Crash Dump Performance Improvements

Parallelism features have been added to the crash dump utility in HP-UX 11i v3 to significantly reduce the dump time. HP testing shows that dump times can be reduced to less than one-quarter of HP-UX 11i v2 dump times on equivalent configurations. See the HP-UX 11i v3 Crash Dump Improvements white paper referenced in section 10 for details.

⁶ See the HP-UX 11i v3 Mass Storage Device Naming white paper referenced in section 10 for details regarding the 11i v3 device file formats.

6 Improved Performance Tracking Tools

The HP-UX 11i v3 mass storage stack provides new or improved performance tracking capabilities, including information regarding:

- Per HBA port performance
- Tape performance
- Read/Write performance
- Per lunpath performance
- Statistics at various levels of the mass storage stack

The per HBA port performance data can be displayed via the new `-H` option to the `sar(1m)` command, or via new options in the GlancePlus system performance monitor. The per HBA port performance data includes

- I/Os per second (IOPS) per HBA port, including read versus write breakdown
- I/O throughput per HBA port, including read vs write breakdown
- HBA port utilization %
- Average number of I/O requests outstanding on the HBA port
- Average queue time (in milliseconds) in the HBA driver
- Average service time (in milliseconds) in the HBA driver

The queue time is the time that an I/O request sits in the HBA driver waiting for resources before being started in the hardware. The service time is the round-trip time to service an I/O request in the hardware.

The HBA port utilization % is the percentage of time that the HBA port is "busy" with I/O requests outstanding on it. For example, the `'sar -H 5'` command will display in the HBA port utilization % value, for each HBA port, the percentage of time I/O requests were outstanding on the HBA port during the specified 5 second interval.

In addition to the new `-H` option, the `sar` command provides the following new options in HP-UX 11i v3:

- `-L` report per lunpath performance data
- `-R` used with `-d` to add read vs write breakdown on per LUN reporting
- `-t` report tape device performance data

The `iostat(1m)` command provides a new option, `-L`, to display I/O statistics for each active lunpath.

Lastly, a new command, `scsimgr`, provides options to display and clear a variety of I/O statistics globally or with respect to various components in the mass storage stack (e.g., HBA ports, targets, LUNs, LUN paths).

These new capabilities allow the system administrator to more granularly track performance at various components and levels in the stack and to thus more readily identify hot spots and adjust the configuration to better distribute the workload.

See the `sar(1m)`, `iostat(1m)`, `scsimgr(1m)`, and `glance(1)` man pages for additional information.

7 New SCSI Tunables

The following new performance-related SCSI tunables are available in HP-UX 11i v3 via the `scsimgr(1m)` command.

- `escsi_maxphys`: the maximum I/O size allowed on the system. This tunable replaces the system tunable, `scsi_maxphys`, available on previous releases of HP-UX. The default setting has been increased from 1MB to 2MB in HP-UX 11i v3, with corresponding interface driver enhancements to support the larger size.

The Logical Volume Manager (LVM) has been enhanced in HP-UX 11i v3 to support the larger I/O sizes, as discussed in section 8.

- `max_q_depth`: the maximum number of I/O requests outstanding per LUN path. This replaces the previously available system tunable, `scsi_max_qdepth`, available on previous releases of HP-UX. The new `max_q_depth` SCSI tunable provides more flexibility in setting the queue depth. It can be set globally, as before, or per device or per device type. It can also be set based on other criteria such as vendor ID, product ID, and so forth.

This flexibility is important because the performance characteristics of increasing or decreasing this tunable are device specific. Some devices or device configurations have more internal I/O resources than others and have varying algorithms in how their resources are used. Setting the `max_q_depth` too high can result in increased CPU utilization; setting it too low can unnecessarily limit the I/O rate. The default value is 8. HP testing with MSA1500 disks⁷ obtained IOPS improvements when `max_q_depth` increased from 8 to 16, particularly with reads. Increasing `max_q_depth` from 16 to 32 in these tests produced either no increase in performance or a slight decrease. Table 1 shows these results:

Table 1: IOPS as `max_q_depth` increases

Test	Qdepth = 4	Qdepth = 8	Qdepth = 16	Qdepth = 32
1K Read IOPS, mpath off	251,000	297,000	304,000	304,000
1K Read IOPS, round-robin	294,000	299,000	302,000	302,000
1K Read IOPS, least-cmd-load	290,000	301,000	304,000	302,000
1K Write IOPS, mpath off	211,000	278,000	295,000	284,000
1K Write IOPS, round-robin	280,000	298,000	296,000	296,000
1K Write IOPS, least-cmd-load	279,000	296,000	290,000	288,000

The “mpath off” label in Table 1 refers to tests with native multi-pathing disabled, while “round-robin”, and “least-cmd-load” refer to tests with multi-pathing enabled using the respective load-balancing policies. Disabling native multi-pathing is only available with legacy DSFs, so the “mpath off” tests were performed using legacy DSFs and the other two sets of tests were performed with both legacy and new persistent DSFs. There was no difference in results between the use of legacy versus new DSFs.

⁷ These tests were performed using the `diskbench` I/O benchmark tool on an rx6600 server with a 2-port 4Gb/s AB379B Fibre Channel adapter connected to MSA1500 Fibre Channel disks (2 paths per LUN). The load factor per LUN (number of processes performing I/O to each LUN) was 32.

8 LVM Improvements

A number of LVM performance improvements have been provided in HP-UX 11i v3, including:

- Large I/O support. LVM now supports I/O sizes up to the extent size, within the limit of the `escsi_maxphys` setting and the HBA driver support as discussed in section 7.
- Faster resynchronization via an enhanced Mirror Write Cache (MWC). The MWC has been enhanced in HP-UX 11i v3 to support larger I/O sizes and to increase the size of the cache.
- Native Multi-Pathing support. LVM is designed to work seamlessly with the native multi-pathing, enabling all of the corresponding performance benefits without having to be concerned with the specifics of LUN paths.
- Fast `vgscan`. With the new `-k` option, `vgscan` can recover activated volume groups in `/etc/lvmtab` quickly (in seconds), even if the system has large number of LVM disks configured.
- Striping + mirroring support. Previously a logical volume could be mirrored or striped but not both. On HP-UX 11i v3 LVM introduces support for striped and mirrored logical volumes at a smaller granularity than the extent size (allowing stripe sizes as small as 4KB), offering increased opportunities for better performance.

See the “LVM New Features in HP-UX 11i v3” white paper referenced in Section 10 for additional information on each of these improvements.

9 Async Disk Driver Improvements

The Async Disk driver has been enhanced in HP-UX 11i v3 to provide the following:

- Dynamic scaling. The `max_async_ports` tunable, which specifies the maximum number of open `asyncdsk` ports, has been enhanced to be dynamically changeable in HP-UX 11i v3. In prior releases a reboot was required to change the number of ports, but in 11i v3 it can be changed via the `kctune` command at any time, even while the Async Disk driver is in use. This allows the system administrator to much more readily tune or re-tune the Async Disk driver as needed.
- More efficient, just-in-time, port allocation. In prior releases all ports up to the maximum specified by the `max_async_ports` tunable were pre-allocated and the required memory consumed whether the port was used or not. In HP-UX 11i v3 memory is not consumed until a port is actually opened.
- Higher `max_async_ports` default. The default value for the tunable has been increased from 50 to 4096, making it much less likely that the administrator will need to adjust the value. This much higher default can be provided on HP-UX 11i v3 without wasting memory due to the more efficient port allocation described above.
- No longer need to set 0x100 in minor number. HP-UX 11i v3 has been enhanced to no longer require the setting of 0x100 in the Async Disk driver minor number as had been required in previous releases to avoid long memory lock-down times.

10 References

<http://docs.hp.com/en/netsys.html#Storage%20Area%20Management>
(HP-UX 11i v3 Native Multi-Pathing for Mass Storage white paper)

<http://docs.hp.com/en/netsys.html#Storage%20Area%20Management>
(scsimgr SCSI Management and Diagnostics Utility on HP-UX 11i v3 white paper)

<http://docs.hp.com/en/netsys.html#Storage%20Area%20Management>
(HP-UX 11i v3 Mass Storage Device Naming White Paper)

<http://docs.hp.com/en/netsys.html#Storage%20Area%20Management>
(HP-UX 11i v3 Crash Dump Improvements white paper)

<http://www.docs.hp.com/en/oshpux11iv3.html#LVM%20Volume%20Manager>
(LVM New Features in HP-UX 11i v3 white paper)

11 For more information

<http://www.hp.com/go/hpux11i>