

Guidelines for tuning HP-UX 11i v3 to handle transient SAN path failures



Table of contents

Abstract	2
IO Performance problems in SAN	3
Impact of transient SAN path failures on I/O retries	3
Role of timeout tunable parameters in LVM and Mass Storage Stack	3
Mechanism for increasing I/O success rate	3
Conclusion.....	5
Appendix A	6
Maximizing the success of IO completion.....	6
Mechanism for proactive path failure detection and recovery	6
Appendix B	8
Steps to determine if the system is experiencing transient SAN path failures	8
Glossary	10
References.....	11
Learn more at	11

Abstract

Failures in SAN environment can be permanent or transient in nature. Transient path failures can occur due to unreliable ports or unreliable interface cable connections, which may result in an application experiencing I/O performance issues. It could even lead to a major intermediate downtime of I/O processing which may appear as an I/O hang from the application perspective.

The HP-UX 11i v3 Mass Storage Stack is designed to provide optimal performance for most SAN configurations with default tunable settings. However, when a system is experiencing I/O performance issues due to transient path failures, applying some tuning recommendations may help in alleviating these performance problems. This paper provides recommendation to system administrators/users to tune the HP-UX Mass Storage Subsystem (Mass Storage Stack) and Logical Volume Manager (LVM), to handle transient path failures more robustly.

It must be noted that these recommendations need to be applied only on such systems which are experiencing transient SAN path failure conditions described in this paper. Also, these recommendations are specific to LVM managed storage configurations and do not apply to VxVM managed storage configurations.

IO Performance problems in SAN

Impact of transient SAN path failures on I/O retries

HP-UX 11i v3 provides native multi-pathing (NMP) built-in to the mass storage subsystem. NMP has manageability benefits, such as transparently handling changes in the SAN without the need for reconfiguration. If there are multiple hardware paths from the host to a LUN (for example, through multiple HBA ports or multiple target ports), the NMP transparently distributes I/O requests across all available paths to the LUN, using a choice of load balancing policies. For additional information, see [Native Multi-Pathing white paper](#) and [Mass Storage I/O Performance Improvements white paper](#) mentioned in the [References](#) section.

While retrying a failed I/O request, path selection logic in HP-UX 11i v3 Mass Storage Stack does not take into account the history of the paths where the I/O request was previously attempted. This could become an issue especially during a transient SAN path failure condition where a few paths to a LUN might be faulty, but a few other paths to the LUN might still be good (see [Appendix B](#) for more information in identifying transient SAN path failure condition). In such a condition, I/O requests may get tried/retried only on the faulty paths even though there are healthy paths available. As detailed in the subsequent section, there are timeout tunable parameters in LVM and Mass Storage Stack which define the time duration for completing an I/O request. Due to these timeout restrictions, in the above scenario, I/O requests may eventually fail back to the upper layer as there may not be enough time budget available to retry I/O on the healthy paths that are available.

This problem can be avoided if the timeout values are tuned in such a way that an I/O request is retried on maximum number of available paths, which in turn increases the chance of retrying I/O on available healthy paths. The following section explains this in more detail.

Role of timeout tunable parameters in LVM and Mass Storage Stack

The timeout tunable parameters in LVM and Mass Storage Stack, control the manner in which I/Os are retried on various paths. When LVM receives an I/O to a logical volume (LV), it converts this logical I/O into physical I/Os on one or more physical volumes (PV) from which the LV is allocated. There are two LVM timeout values which affect the I/O operation, namely, LV timeout and PV timeout. LV timeout controls how long LVM retries a logical I/O after a recoverable physical I/O error has occurred. PV timeout is the time budget set by LVM for each physical I/O originating from LVM. Mass Storage Stack must complete the physical I/O subject to PV timeout. The default PV timeout is 30 seconds while the default LV timeout is infinite. Refer to Appendix titled "LVM I/O Timeout Parameters" in [LVM System Administrator's Guide](#) for more details on LV timeout and PV timeout.

The mass storage subsystem defines a tunable **esd_secs** which is a LUN attribute that holds the time credit for an I/O operation across different retries. Default **esd_secs** value is 30 seconds. For an LVM I/O, both, **pv_timeout** and **esd_secs** values play a role in determining the IO timeout value at the physical link level.

Transient SAN failures can result in some paths to a LUN becoming unavailable. When an I/O request is sent on such paths, the I/O will time out at Mass Storage Stack layer after **pv_timeout** expires. These timed out I/Os are queued for retries in the LVM layer and the associated Physical Volume(s) are marked as unavailable. This results in performance degradation for application issuing the I/O.

Under such transient SAN failure conditions, the current I/O retry mechanism in Mass Storage Stack, with the default settings of I/O timeout tunable parameters (**pv_timeout** and **esd_secs**), allows time for the I/O to be tried on a maximum of 2 paths before exhausting the **pv_timeout** value. Therefore, even if additional healthy paths are available to the LUN, they will not be utilized for retrying the IO, because of the timeout restrictions. The I/O request will be reported as "failed" to the upper layer.

Mechanism for increasing I/O success rate

In order to maximize the success rate of an I/O, it is important that an I/O is retried on as many numbers of paths as possible. As the number of paths that can be tried is limited by the I/O timeout tunable parameters, the disk I/O timeout tunable (**esd_secs**) and the Physical Volume I/O timeout tunable (**pv_timeout**) values need to be tuned appropriately.

The following formula can be used to tune **pv_timeout** value in order to maximize the I/O success rate by increasing the number of paths retried.

$$pv_timeout = (n * esd_secs) + ((n-1) * \Delta)$$

Where **pv_timeout** is the LVM time budget available to Mass Storage Stack to attempt retries of an I/O that has failed to a given PV, **n** is the total number of paths configured to the PV, **esd_secs** is the MSS disk I/O time budget within which the I/O has to succeed on a given path and Δ is the delay (~2 seconds) between retries by Mass Storage Stack, each time the I/O fails.

Note

As the wire timeout value defined at interface driver is 10 seconds, the minimum tunable value possible for **esd_secs** is 10 seconds.

Tuning **pv_timeout** value as above helps to make sure that the I/O is attempted on maximum number of configured lun paths before it times out, thereby increasing the chance for I/O to succeed.

However when the number of paths is high, the above formula yields a large **pv_timeout** value, which may not be practical. For instance, when LV timeout is configured to a finite value, then it needs to be adjusted to match or exceed this large **pv_timeout**. This means that LV timeout also increases with PV timeout which may not meet the application I/O timeout requirements.

Further, LVM currently limits the **pv_timeout** value to a maximum of 300 seconds. Hence to keep the **pv_timeout** within acceptable limits, while maximizing the I/O success rate in transient conditions, the preference should be to tune **esd_secs** to a smaller value while applying the formula.

Caution

Lowering the **esd_secs** may result in lesser resiliency for non-LVM I/Os. For instance, on a disk with multiple partitions, with at least one partition managed by LVM, I/Os initiated to non-LVM partitions may be less resilient to transient SAN delays as **esd_secs** applies to the whole disk.

The examples below illustrate how the formula can be used to arrive at the timeout values for some specific cases.

Example-1: where **esd_secs** set to default (30 seconds) with 4 paths:

$$\begin{aligned} \text{pv_timeout} &= (n * \text{esd_secs}) + ((n-1) * \Delta) \\ \text{pv_timeout} &= (4 * 30) + ((4-1) * 2) = 126 \end{aligned}$$

By setting the **pv_timeout** tunable to 126, Mass Storage Stack gets enough time to retry the failed I/O on all the 4 available paths.

Example-2: where **esd_secs** is changed (to 10 seconds) from the default with 4 paths:

$$\text{pv_timeout} = 4 * 10 + (4-1) * 2 = 46$$

By setting the **pv_timeout** tunable to 46, Mass Storage Stack gets enough time to retry the failed I/O on all the 4 available paths.

Note

- In case at least one path is via a Qlogic HBA, managed by the fcd driver (driver delivered by FibreChnl-01bundle), the value of “ Δ ” is measured approximately as 13 seconds (this includes 2 sec delay as explained in earlier section + additional 11 sec which is specific to fcd).
 - LVM Logical volume (LV) can be mirrored on disks with multiple paths. In such configurations, write I/O's are sent simultaneously to all mirrored disks, therefore, if the I/O succeeds on at least one of the mirrored disk, LVM will take care of synchronizing the data on other disks. In case of read I/O's, it is desirable that the I/O fails back faster to LVM so that the I/O can be retried on another mirror disk, rather than on different paths of the same disk. Thus, when a mirror disk faces transient SAN failures, rather than retrying an I/O on all the paths of this disk, it is desirable to retry the I/O on other mirror disks. As mirror disks are typically configured through distinct SAN, chances of I/O success is higher when the I/O is retried on the other mirror disk. Therefore, the tuning suggestions explained above are not necessary for mirrored disks.
-

While the above guidelines can maximize the I/O success rate by allowing the I/O request to be tried on as many lunpaths as possible before timing out, it still cannot guarantee that all paths will be tried. However, the scenario can be improved

further by making use of an existing feature in Mass Storage Stack which temporarily eliminates the paths which are experiencing I/O failures from the I/O scheduling mechanism. This feature is described in [Appendix A](#).

Conclusion

HP recommends applying the tuning guidelines described in [Mechanism for increasing I/O success rate](#) in conjunction with path elimination and recovery feature described in [Appendix A](#) on systems with LVM, experiencing transient SAN path failures.

The steps to determine whether a system is experiencing transient SAN path failures is described in [Appendix B](#).

Tunable Parameter	Default Value	Recommended Value	Caution
esd_secs	30s	10s	I/O request that do not succeed within 10 seconds would be timed out.
pv_timeout	30s	$(n * esd_secs) + ((n-1) * \Delta)$	Where n is the number of paths to a lun and $\Delta = 2$ ($\Delta = 13$ in case at least one path is via a Qlogic HBA, managed by the fcd driver)

Appendix A

Maximizing the success of IO completion

While tuning the `pv_timeout` and `esd_secs`, will allow for the failed IO to be retried on additional paths, it does not guarantee that the IO will not be retried on those paths that have been tried so far. Due to the nature of IO scheduling, it is quite possible that by the time the IO to be retried is scheduled, Mass Storage Stack would have scheduled other IOs on the remaining paths regardless of the scheduling policy in effect. Thus, the retried IO may get tried again on the same path on which it already failed.

To eliminate this possibility, we recommend using an existing Mass Storage Stack feature in addition to the above tuning. This feature helps to mark a path offline preventing further I/O being scheduled on that path, in case of I/O failures. The feature is described below.

Mechanism for proactive path failure detection and recovery

To ensure continuous path availability, the Mass Storage Stack uses a combination of proactive failing of paths experiencing intermittent errors and a recovery mechanism to recover those paths. The following terms are used to describe the approach:

- `path_fail_secs (=X)`: This is the value in seconds for which the Mass Storage Stack will wait after the first I/O has failed on the path before marking the path as offline. At least one I/O request has to succeed within X seconds for the path to be not marked as offline. In other words, if a lunpath continues to see I/O errors with no successful I/O completions for X seconds, then Mass Storage Stack marks the lunpath as offline.
- `ping_type`: Polling method to test the health status of the offline path. Set to either `none`¹, `default`² or `extended`³.
- `ping_recovery`: This is a recovery policy either set to `time_based` or `count_based`
 - `ping_time_threshold (=Y)`: This tunable is set when the `ping_recovery` is set to `time_based` and the path will be brought online after continuous successful pings for Y seconds.
 - `ping_count_threshold (=Z)`: This tunable is set when the `ping_recovery` is set to `count_based` and the path will be brought online after Z continuous successful pings

Tuning the `path_fail_secs` to a lower value, causes Mass Storage Stack to mark the path offline, relatively early. Say we set the tunable to 10s, then, if a lunpath experiences an IO failure and no IO request succeeds on that path for the next 10 seconds, then the path will be marked as offline. The path will not be considered for any I/O until it is successfully recovered.

In the background, Mass Storage Stack will keep polling the lunpath to verify if the path has come back online. The criteria to mark the path online can be either time-based or count based.

Using the time-based approach, Mass Storage Stack will mark the lunpath online if ping IO to that path succeeds for `ping_time_threshold` seconds without any failure. In the count-based approach, Mass Storage Stack will mark the path online if `ping_count_threshold` number of consecutive pings on the lunpath return successfully.

Using this mechanism in conjunction with the tuning of `pv_timeout` and `esd_secs` tunable parameters, we can ensure that a failed IO gets retried on as many available paths as possible while not being retried on paths that have experienced failures.

Note

For more info on the `path_fail_secs` and ping recovery policy tunable parameters refer to the white paper at http://h20565.www2.hp.com/hpsc/doc/public/display?sp4ts.oid=3367813&docId=emr_na-c03469622&docLocale=en_US

¹ **none**: No attempt to recover paths marked offline.

² **default**: INQ cmd will be sent on the lunpath(s) marked offline.

³ **extended**: INQ, TUR and Read Capacity cmds will be sent on the lunpath(s) marked Offline

time_based ping recovery mechanism

Tuning the Mass Storage Stack tunable parameters (`path_fail_secs`) and the ping based path recovery mechanism gives the I/O a greater chance to succeed by eliminating paths that experience errors by marking them as failed. The following are the recommended values for the Mass Storage Stack tunable parameters to enable this mechanism.

- `path_fail_secs` = 10 seconds (default is 120 seconds)
- `ping_type` = extended (default is "default")
- `ping_recovery` = time_based (default is "immediate")
 - `ping_time_threshold` = 10 seconds

Note

Below is the command line example of how to set the above tunable parameters using `scsimgr (1M)` command.

```
# scsimgr set_attr -D <disk dsf> -a path_fail_secs=10 -a ping_type=extended
ping_recovery=time_based -a ping_time_threshold=10
```

As the path ping policy is set to extended, the offline paths will be marked online automatically if the real hardware status of the path changes to healthy condition. If the ping succeeds for ~10 seconds, the paths are made available for selection in the current scheduling policy for new I/Os.

Note

If the condition is not improved in terms of the I/O success rate with the retries we can alter the `path_fail_secs` and `ping_time_threshold` accordingly and observe the situation.

Count_based ping recovery mechanism

`count_based` tunable is used to mark the offline lun path back online after X (`ping_count_threshold`) successful pings. When choosing `count_based` recovery policy, the `ping_type` attribute should be set to either basic or extended for disk. The following are the recommended values for the Mass Storage Stack tunable parameters to enable this mechanism.

- `path_fail_secs`=10 seconds(default is 120 seconds)
- `ping_type`=extended (default is "default")
- `ping_recovery`=count_based (default is "immediate")
 - `ping_count_threshold`=5

Note

The below is the command line example of how to set the above tunable parameters using `scsimgr (1M)` command.

```
# scsimgr set_attr -D /dev/rdisk/disk18 -a path_fail_secs=10 -a ping_type=extended
ping_recovery=count_based -a ping_count_threshold=5
```

In this case as well, once the paths are moved to offline state due to transient failures, after 5 successful pings of these paths they are made available for selection in the current scheduling policy for new I/Os.

Note

- In an unlikely situation where all the paths are experiencing continuous transient failures mass storage subsystem may end up marking all the paths as offline. Hence this approach is to be used only when more than one path is configured and the paths are split across at least two or more independent SAN.
- Giving less value for `path_fail_secs` as described above, may not yield desired results in case of certain types of I/O failures. For example if the paths to the device are through a card managed by FCD/FCLP interface driver, when an I/O request fails with a specific error condition like `SCTL_IO_TIMEOUT` or `SCTL_IO_ABORT`, the above recommendations will not be useful to handle such failures.

Appendix B

Steps to determine if the system is experiencing transient SAN path failures

scsimgr(1M) utility along with fcmsutil (1M) utility provides command line options to gather statistics related to IO operations for a lun, lunpath or the HBA (FC card). Additionally LVM subsystem prints warning messages in the syslog file (/var/adm/syslog/syslog.log file) when there are I/O failures and LVM command utility vgdisplay(1M) has options to determine the Physical Volume status.

The following section describes the specific statistic counters, the error messages and the status information that can be gathered using the above mentioned utilities to determine whether the system is experiencing transient SAN path failures.

LVM subsystem warning messages and Physical Volume (PV) status:

If LVM subsystem is experiencing I/O failures, it may print the following warning message in the syslog (/var/adm/syslog/syslog.log file).

Syslog traces:

```
Apr 10 13:42:44 vmunix: LVM: WARNING: VG 64 0x020000: LV 1: Some I/O requests to
this LV are waiting
Apr 10 13:42:44 vmunix: indefinitely for an unavailable PV. These requests will be
queued until
Apr 10 13:42:44 vmunix: the PV becomes available (or a timeout is specified for
the LV).
```

vgdisplay(1M) command can be run to determine the Physical volume (PV) status. The status may be shown as “unavailable” when PV is experiencing large number of I/O failures.

```
Cmd: vgdisplay -v <Volume Group(VG) device path> | grep "PV Status"
"PV Status          unavailable"
```

The above warning messages from LVM subsystem together with the status information from the vgdisplay (1M) command may indicate that LVM subsystem is experiencing I/O failures.

To determine whether the I/O failures seen at LVM layer is due to transient path failures, additional I/O statistics information need to be gathered as stated below. The recommendation is to gather these statistics in a periodic manner (say every 5 minutes) and if the counters show an increasing trend then it is a clear indication that system is observing transient path failures.

scsimgr(1M) get_stat command (lun/ lunpaths statistics)

Given an agile lun Device Special file (DSF), get_stat option of scsimgr(1M) command can be run in the following manner to get I/O statistic information for all the lun paths associated with that LUN.

```
Cmd: scsimgr get_stat -D <lun DSF> -v all_lpt
```

For each of the lunpath associated with the LUN, the following I/O transfer statistics information from the above command output can be monitored for a period of time. If the statistics is showing significantly increasing trend only for some particular lunpaths compared to other lunpaths, then that is an indication that there is a transient path failure condition existing for some paths.

```
--- I/O transfer Statistics ---
I/O failures
Retried I/Os
Incomplete status
IO Timeout status
IO Aborted status
Asynchronous IO read failures
Asynchronous IO write failures
B_PFTIMEOUT I/O failures
```

Further to the above mentioned statistics at the lun/ lunpath level, interface driver level statistics can also be gathered to confirm that the system is experiencing transient path failures.

fcmsutil devstat command (fcd interface driver statistics)

For a given fcd HBA port, devstat option of fcmsutil (1M) command can be run as follows to gather statistic information

Cmd: `fcmsutil /dev/fcd<n> devstat all`

Specifically the following I/O transfer statistics information from the above command output can be gathered in a periodic manner. If the statistics is showing an increasing trend, then it confirms that the paths through this particular fcd HBA port are experiencing transient path failures.

Number of IOs timed out in fw

fcmsutil stat command (fclp/fcoc interface driver statistics)

For a given fclp or fcoc HBA port, stat option of fcmsutil (1M) command can be run as follows to gather statistic information

Cmd: `fcmsutil /dev/fclp<n> stat`

OR

Cmd: `fcmsutil /dev/fcoc<n> stat`

Specifically the following I/O transfer statistics information from the above command output can be gathered in a periodic manner. If the statistics is showing an increasing trend, then it confirms that the paths through this particular fclp/fcoc HBA port are experiencing transient path failures.

```
--- FCLP IO Module Statistics ---  
I/O completion failed  
I/O timeout  
Active I/O aborted  
Channel/NP is offline when starting I/O  
Channel/NP is transient when starting I/O  
Aborts issued  
Abort timeout
```

Glossary

Mass Storage Stack	Mass Storage Subsystem
LVM	Logical Volume Manager
VG	Volume Group
PV	Physical Volume
LV	Logical Volume
SAN	Storage Area Networks
FibreChnl-01	Fibre Channel Interface Driver bundle name.
lv_timeout	The LVM Logical Volume I/O timeout tunable
ping_type	Path ping policy (none, basic, or extended) The ping_type attribute should be set to either basic or extended for disk.
ping_time_threshold	It is the time based threshold tunable used to bring the offline path back to online after ping test continually succeeding for Y seconds

References

1. For more information on Native Multi-Pathing, path_fail_secs and ping recovery policy tunable parameters, see the following white paper at:
<http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c03469622/c03469622.pdf>
2. For more information on Mass Storage I/O Performance Improvements, see the following white paper at:
<http://bizsupport1.austin.hp.com/bc/docs/support/SupportManual/c01915741/c01915741.pdf>
3. For more info on the LVM I/O timeout tunable parameters, see the following man pages
 - `pv_timeout: pvchange(1M)`
 - `path_fail_secs: scsimgr(1M)`
4. HP-UX System Administrator's guide on LVM at:
<http://www.hp.com/go/hpux-LVM-VxVM-docs>

Learn more at

www.hp.com/go/HP-UX 11i v3

Sign up for updates
hp.com/go/getupdated



Share with colleagues



Rate this document

© Copyright 2012 – 2015 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Links to third-party websites take you outside the HP website. HP has no control over and is not responsible for information outside HP.com.

Intel® and Itanium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX® is a registered trademark of The Open Group.

828084-001, May 2015

