BROCADE
A Broadcom Inc. Company

VALIDATION TEST REPORT

Brocade Fabric Technology with the Infinidat InfiniBox Array with Flash Cache

Supporting Fabric OS 7.4.0, 8.0.1, 8.1.0, 8.2.0, and 8.2.1

53-1004092-04
12 November 2018

Contents

Preface

Document History

Date	Part Number	Description
December 2015	53-1004092-01	Supporting Brocade Fabric OS 7.4.0.
May 2017	53-1004092-02	Supporting Brocade Fabric OS 8.0.1 and 8.1.0.
March 2018	53-1004092-03	Supporting Brocade Fabric OS 8.2.0.
November 2018	53-1004092-04	Supporting Brocade Fabric OS 8.2.1.

Overview

The Storage Fabric Ready (SFR) program is a comprehensive testing and configuration initiative to validate the interoperability of Fibre Channel flash storage with a Brocade FC network infrastructure. This program provides testing of multiple fabrics, heterogeneous servers, and HBAs in a large-port-count Brocade environment. The SFR qualification program helps verify seamless interoperability and optimum performance with solid-state storage systems in Brocade FC storage fabrics.

Purpose of This Document

The goal of this document is to demonstrate the compatibility of the Infinidat InfiniBox FC storage array in a Brocade FC fabric containing Gen 5 and Gen 6 FC switches. This document provides a test report on the SFR qualification test plan executed on the Infinidat InfiniBox storage array.

Audience

The target audience for this document includes storage administrators, solution architects, system engineers, and technical development representatives.

Objectives

- Test the Infinidat InfiniBox array with the Brocade FC fabric in single and routed configurations for different stress and error recovery scenarios, and thereby validate the interoperability and integration of the Infinidat InfiniBox array with Brocade FC fabrics.
- Validate the performance of the Brocade FC fabric in a solid-state storage environment for high throughput and low latency applications.

Related Documents

- *Brocade Fabric OS Administration Guide*
- *Brocade Monitoring and Alerting Policy Suite Configuration Guide*
- *Brocade SAN Design and Best Practices*
- *Brocade Flow Vision Configuration Guide*
- *Brocade Fabric OS Command Reference*
- *Brocade SAN Fabric Resiliency and Administration Best Practices*
- Emulex ExpressLane Configuration

About Broadcom

Broadcom Inc. provides innovative storage networking solutions for data center, campus, and service provider networks, helping to reduce complexity and cost while enabling virtualization and cloud computing to increase business agility. To help ensure a complete solution, Broadcom partners with world-class IT companies and provides comprehensive education, support, and professional services offerings (www.broadcom.com).

About Infinidat

INFINIDAT was founded in 2011 by industry veterans focused on solving the problems that storage buyers face when they are forced to choose between cost, capacity, functionality, reliability, and performance. INFINIDAT enables you to have it all—today—and at a price that you can afford.

Businesses depend on data and rapid access to it, and the availability of that data is critical. With the growth of digitized data outpacing the growth of storage capacity, the market for storage has grown rapidly. Yet cost and complexity are forcing buyers to constantly make trade-offs. To solve this problem and capitalize on the exploding storage market, INFINIDAT has brought to market InfiniBox™, a new generation of highly reliable, scalable, and efficient storage systems.

InfiniBox™ is installed in companies around the world and supports a wide variety of applications and use cases. INFINIDAT is headquartered Herzliya, Israel, with US operations in Needham, Massachusetts.

To learn more, visit www.infinidat.com.

Configure DUT and Test Equipment

Task 1: Brocade FC Fabric Configuration

1. Enable MAPS and Fabric Performance Impact (FPI) monitoring on all switches in the fabric.

   MAPS enables health monitoring on the switches to detect potential faults and create alerts. MAPS FPI allows fabric monitoring for performance impacts, including timeouts, latency, and throughput. Detailed information on MAPS configuration and setup can be found in the *Brocade Monitoring and Alerting Policy Suite Configuration Guide*.

   - MAPS requires a "Fabric Vision License" to be installed.
   - Enable the desired MAPS policy using any of the available default policies, or create a custom policy.
   - FPI monitoring is enabled by default.

   ```
   > mapsconfig --enablemaps -policy dflt_aggressive_policy
   > mapsconfig --actions raslog,email,sw_marginal,sw_critical
   > mapsconfig --show
   Configured Notifications: RASLOG,EMAIL,SW_CRITICAL,SW_MARGINAL
   Mail Recipient: testuser@domain.com
   Paused members :
   ===============
   PORT :
   CIRCUIT :
   SFP :
   ```

   To view the summary of events and rules triggered:

   ```
   mapsdb –show

   1 Dashboard Information:
   ========================

   DB start time:                  Thu Feb 23 07:02:51 2017
   Active policy:                  ssr_aggressive_policy
   Configured Notifications:       RASLOG,EMAIL,SW_CRITICAL,SW_MARGINAL
   Fenced Ports :                  None
   Decommissioned Ports :          None
   ```

2. Configure Flow Monitoring with the I/O Insight feature on the Brocade G620 Switch in the fabric.

   The I/O Insight feature supported on the Brocade G620 Switch allows us to monitor the flow latency statistics at the SCSI I/O exchange level. The monitoring can be configured at an IT (Initiator-Target) flow level on fixed-port switches and at an ITL (Initiator-Target-LUN) flow level on chassis-based switches.

   Requires “Fabric Vision” and “IO Insight” licenses.

   a) Create a "Flow Monitor" flow at the source or destination device port on the Brocade G620 Switch.

```
> flow --create ios_infini_1 -fea mon -dstdev df1900 -egrport 25

> flow --show ios_infini_1
================================================================================================================
Name      : ios_infini_1        Features: mon(Activated)        noConfig: Off
Definition: EgrPort(25),DstDev(0xdf1900)
```

   b) Import the created flows into MAPS.

```
root> mapsconfig --import ios_infini_1

root> logicalgroup --show
----------------------------------------------------------------------------------------------------------------
Group Name                      |Predefined |Type              |Member Count |Members
----------------------------------------------------------------------------------------------------------------
ios_infini_1                    |No         |Flow              |1            |Monitored Flow
```

   c) Create MAPS rules to monitor the desired SCSI I/O latency statistics, and add them to a custom MAPS policy.

```
> mapspolicy --clone dflt_aggressive_policy -name ios_aggressive_policy

> mapsrule --create ios_infini_1_4k_wr_status -group ios_infini_1 -monitor WR_STATUS_TIME_LT_8K -timebase min
-op ge -value 1000 -action email,raslog -policy ios_aggressive_policy

> mapspolicy --show ios_aggressive_policy
Policy Name: ios_aggressive_policy

Rule Name                               |Condition                                                |Actions       |
-----------------------------------------------------------------------------------------------------------------
……..
ios_infini_1_4k_wr_status               |ios_infini_1(WR_STATUS_TIME_LT_8K/min>=1000)      |email,raslog  |
ios_infini_1_4k_rd_status               |ios_infini_1(RD_STATUS_TIME_LT_8K/min>=1000)      |email,raslog  |

……
```

   d) Enable the MAPS policy.

```
> mapspolicy --enable ios_aggressive_policy

> mapsdb --show

1 Dashboard Information:
========================
DB start time:                  Thu Mar  9 16:48:07 2017
Active policy:                  ios_aggressive_policy
Configured Notifications:       RASLOG,SW_CRITICAL,SW_MARGINAL
Fenced Ports :                  None
Decommissioned Ports :          None
Fenced circuits :               N/A
Quarantined Ports :             None
Top Zoned PIDs <pid(it-flows)>: 0xe01a01(30) 0xe01b01(30) 0xe01901(30) 0xe01801(30) 0xe02200(22)

2 Switch Health Report:
=======================
Current Switch Policy Status: HEALTHY
```

3. Configure zoning using the Peer Zoning feature in Brocade Fabric OS.

   Peer zoning allows a "principal" device to communicate with the other devices in the zone. The principal device manages a peer zone. Other "nonprincipal" devices in the zone can communicate with the principal device only; they cannot communicate with each other.

```
root> zonecreate –peerzone ssr_infinidat_1 –principal "57:42:b0:f0:00:04:2b:15; ….." –members
"10:00:8c:7c:ff:24:b6:00; …"

root> zoneshow --peerzone all
zone:  ssr_infinidat_1
   Property Member: 00:02:00:00:00:03:00:03
   Created by: User
   Principal Member(s):
               57:42:b0:f0:00:04:2b:15; 57:42:b0:f0:00:04:2b:25;
               57:42:b0:f0:00:04:2b:35
   Peer Member(s):
               10:00:8c:7c:ff:24:b6:00; 10:00:8c:7c:ff:24:b6:01;
               10:00:8c:7c:ff:07:49:00; 10:00:8c:7c:ff:07:49:01;
               10:00:8c:7c:ff:4f:ca:00; 10:00:8c:7c:ff:4f:ca:01;
               21:00:00:0e:1e:1b:f1:21; 21:00:00:0e:1e:1b:f1:20;
               21:00:00:0e:1e:18:99:80; 21:00:00:0e:1e:18:99:81;
               10:00:00:90:fa:61:92:3b; 10:00:00:90:fa:61:92:3c;
               10:00:8c:7c:ff:05:60:01; 10:00:8c:7c:ff:05:60:00;
               10:00:8c:7c:ff:14:e0:01; 10:00:8c:7c:ff:14:e0:00;
               10:00:8c:7c:ff:03:bc:01; 10:00:8c:7c:ff:03:bc:00;
               10:00:8c:7c:ff:03:9b:00; 10:00:8c:7c:ff:03:9b:01;
               10:00:8c:7c:ff:05:72:02; 10:00:8c:7c:ff:05:72:03

root> cfgactvshow
zone:  ssr_infinidat_1
               00:02:00:00:00:03:00:03
               57:42:b0:f0:00:04:2b:15
               57:42:b0:f0:00:04:2b:25
               57:42:b0:f0:00:04:2b:35
               10:00:8c:7c:ff:24:b6:00
               10:00:8c:7c:ff:24:b6:01
               10:00:8c:7c:ff:07:49:00
               10:00:8c:7c:ff:07:49:01
               10:00:8c:7c:ff:4f:ca:00
               10:00:8c:7c:ff:4f:ca:01
               21:00:00:0e:1e:1b:f1:21
. . .
```

4. Configure Fibre Channel Routing (an Integrated Routing license is required).

   The FC-FC routing service provides Fibre Channel routing between two or more fabrics without merging those fabrics. For example, using FC-FC routing, you can share tape drives across multiple fabrics without the administrative problems, such as change management, network management, scalability, reliability, availability, and serviceability, that might result from merging the fabrics. Detailed information on FCR setup can be found in the *Brocade Fabric OS Administration Guide*.

   An example FCR configuration is shown below.

   Enabling FCR on the backbone fabric switches.

   ```
   > fcrconfigure -bbfid 100
   > fosconfig --enable fcr
   ```

   Configuring EX ports connecting to edge fabrics.

   ```
   > portcfgexport [port#] -a1 -m0 -f 10
   > portcfgexport [port#] -a1 -m0 -f 20
   > portcfgexport [port#] -a1 -m5 -f 50
   ```

   An LSAN zone is created on both fabrics.

   ```
   > cfgactvshow
   zone:  LSAN_ssr_infinidat_peer
                  00:02:00:00:00:03:00:03; 57:42:b0:f0:00:04:2b:11;
                  57:42:b0:f0:00:04:2b:21; 57:42:b0:f0:00:04:2b:31;
                  10:00:8c:7c:ff:07:49:00; 10:00:8c:7c:ff:07:49:01;
                  10:00:8c:7c:ff:14:e0:00; 10:00:8c:7c:ff:14:e0:01
   ```

   Example output of exported devices.

   ```
   > fcrproxydevshow
     Proxy           WWN             Proxy      Device   Physical    State
    Created                           PID       Exists     PID
   in Fabric                                  in Fabric
   ------------------------------------------------------------------------
       10   21:00:00:24:ff:48:b9:6a  02f001       20      551a00   Imported
       10   21:00:00:24:ff:48:b9:6b  02f101       20      541e00   Imported
       10   52:4a:93:7d:f3:5f:61:00  02f201       20      550e00   Imported
       10   52:4a:93:7d:f3:5f:61:01  02f401       20      540400   Imported
   ```

Task 2: Infinidat InfiniBox Array Configuration

1. The tested InfiniBox array consists of three active controller nodes and two disk drive enclosures supplying 250 TB of usable capacity. There are 24x1TB SSD drives (8 per controller) used as cache drives. Each controller node has eight 8-Gb Fibre Channel ports for connectivity. The test bed is set up with one port from each controller node connected to each FC fabric.

   FIGURE 1 Infinidat Controller Nodes

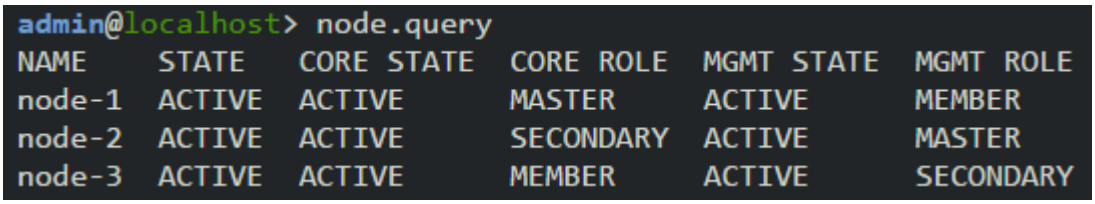


FIGURE 2 Infinidat System Setup

```
admin@localhost> system.info
Name                                ssr-infinibox
Serial                              1067
Product ID                          INFINIBOX
Model Name                          F2130
Version                             3.0.0.150
Uptime                              3 months 28 days 3 hours 47 minutes
Operational State                   ACTIVE
Disk Encryption                     no
WWNN                                57:42:b0:f0:00:04:2b:00
SSD Cache Drives                    24
Total Physical Capacity             249.73 TB
Free Physical Capacity              239.62 TB
Pools Physical Capacity             10.00 TB
Pools Allocated Physical Capacity   2.28 TB
Total Virtual Capacity              624.33 TB
Free Virtual Capacity               614.33 TB
Pools Virtual Capacity              10.00 TB
Pools Allocated Virtual Capacity    2.28 TB
Compression Savings                 1.00 : 1
Pools                               1
Volumes                             21 (0 snapshots)
Filesystems                         1 (0 snapshots)
Consistency Groups                  0 (0 snap-groups)
Replicas                            0
Exports                             3
Hosts                               8
Inactive Drives                     1
Rebuild 1 In Progress               no
Rebuild 2 In Progress               no
```

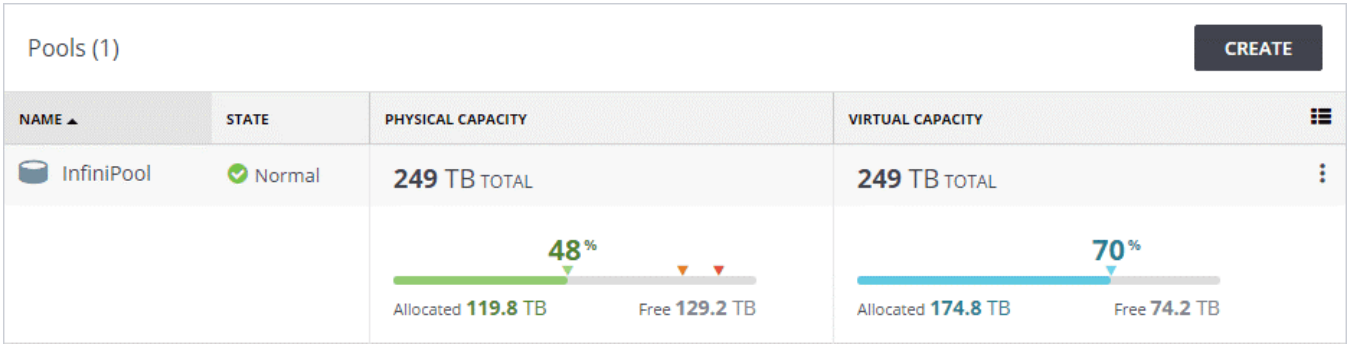
2. Connect Fibre Channel target ports on the array to the Brocade FC fabric, with two ports from each controller connected to FC Fabric 1 and FC Fabric 2. Additional FC ports are available for connection to the fabric to increase the bandwidth on the controllers and achieve higher performance.

FIGURE 3 Infinidat Fibre Channel Ports

```
admin@localhost> config.fc.port.query
NAME   ENABLED  STATE  STATE DESCRIPTION  LINK  SWITCH          WWPN                    ROLE        SPEED  MAX SPEED  PATCH PANEL
N1FC1  yes      OK     -                  UP    Brocade94       57:42:b0:f0:00:04:2b:11 HARD_PORT  8 Gbps     8 Gbps  1
N1FC2  yes      OK     -                  UP    G620_224        57:42:b0:f0:00:04:2b:12 HARD_PORT  8 Gbps     8 Gbps  2
N1FC3  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:13 HARD_PORT        -     8 Gbps  3
N1FC4  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:14 HARD_PORT        -     8 Gbps  4
N1FC5  yes      OK     -                  UP    B6510_74        57:42:b0:f0:00:04:2b:15 HARD_PORT  8 Gbps     8 Gbps  5
N1FC6  yes      OK     -                  UP    B6510_91_SSR2   57:42:b0:f0:00:04:2b:16 HARD_PORT  8 Gbps     8 Gbps  6
N1FC7  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:17 HARD_PORT        -     8 Gbps  7
N1FC8  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:18 HARD_PORT        -     8 Gbps  8
N2FC1  yes      OK     -                  UP    Brocade94       57:42:b0:f0:00:04:2b:21 HARD_PORT  8 Gbps     8 Gbps  1
N2FC2  yes      OK     -                  UP    G620_224        57:42:b0:f0:00:04:2b:22 HARD_PORT  8 Gbps     8 Gbps  2
N2FC3  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:23 HARD_PORT        -     8 Gbps  3
N2FC4  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:24 HARD_PORT        -     8 Gbps  4
N2FC5  yes      OK     -                  UP    B6510_74        57:42:b0:f0:00:04:2b:25 HARD_PORT  8 Gbps     8 Gbps  5
N2FC6  yes      OK     -                  UP    B6510_91_SSR2   57:42:b0:f0:00:04:2b:26 HARD_PORT  8 Gbps     8 Gbps  6
N2FC7  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:27 HARD_PORT        -     8 Gbps  7
N2FC8  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:28 HARD_PORT        -     8 Gbps  8
N3FC1  yes      OK     -                  UP    Brocade94       57:42:b0:f0:00:04:2b:31 HARD_PORT  8 Gbps     8 Gbps  1
N3FC2  yes      OK     -                  UP    G620_224        57:42:b0:f0:00:04:2b:32 HARD_PORT  8 Gbps     8 Gbps  2
N3FC3  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:33 HARD_PORT        -     8 Gbps  3
N3FC4  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:34 HARD_PORT        -     8 Gbps  4
N3FC5  yes      OK     -                  UP    B6510_74        57:42:b0:f0:00:04:2b:35 HARD_PORT  8 Gbps     8 Gbps  5
N3FC6  yes      OK     -                  UP    B6510_91_SSR2   57:42:b0:f0:00:04:2b:36 HARD_PORT  8 Gbps     8 Gbps  6
N3FC7  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:37 HARD_PORT        -     8 Gbps  7
N3FC8  yes      OK     -                  DOWN  -               57:42:b0:f0:00:04:2b:38 HARD_PORT        -     8 Gbps  8
```

3. All drives on the array are configured as part of a single pool.

FIGURE 4 Infinidat Disk Pool



4. Create host groups with the appropriate initiator WWNs associated with that host. To create cluster groups, create individual host groups for the cluster nodes and add them to the cluster group.

FIGURE 5 Infinidat Host Group Creation

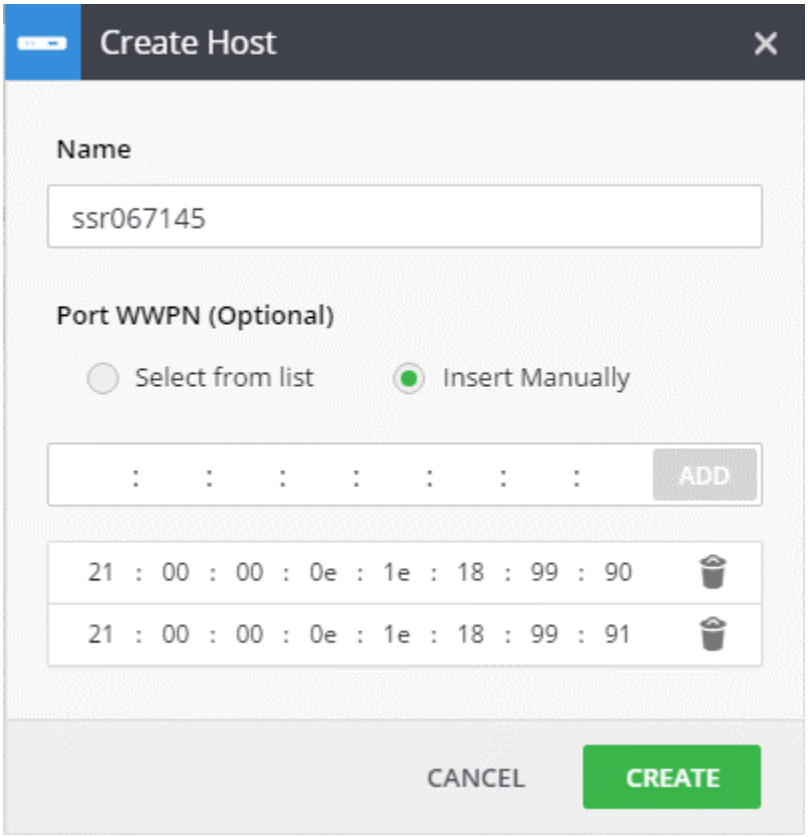


FIGURE 6 Infinidat Host Group Connectivity

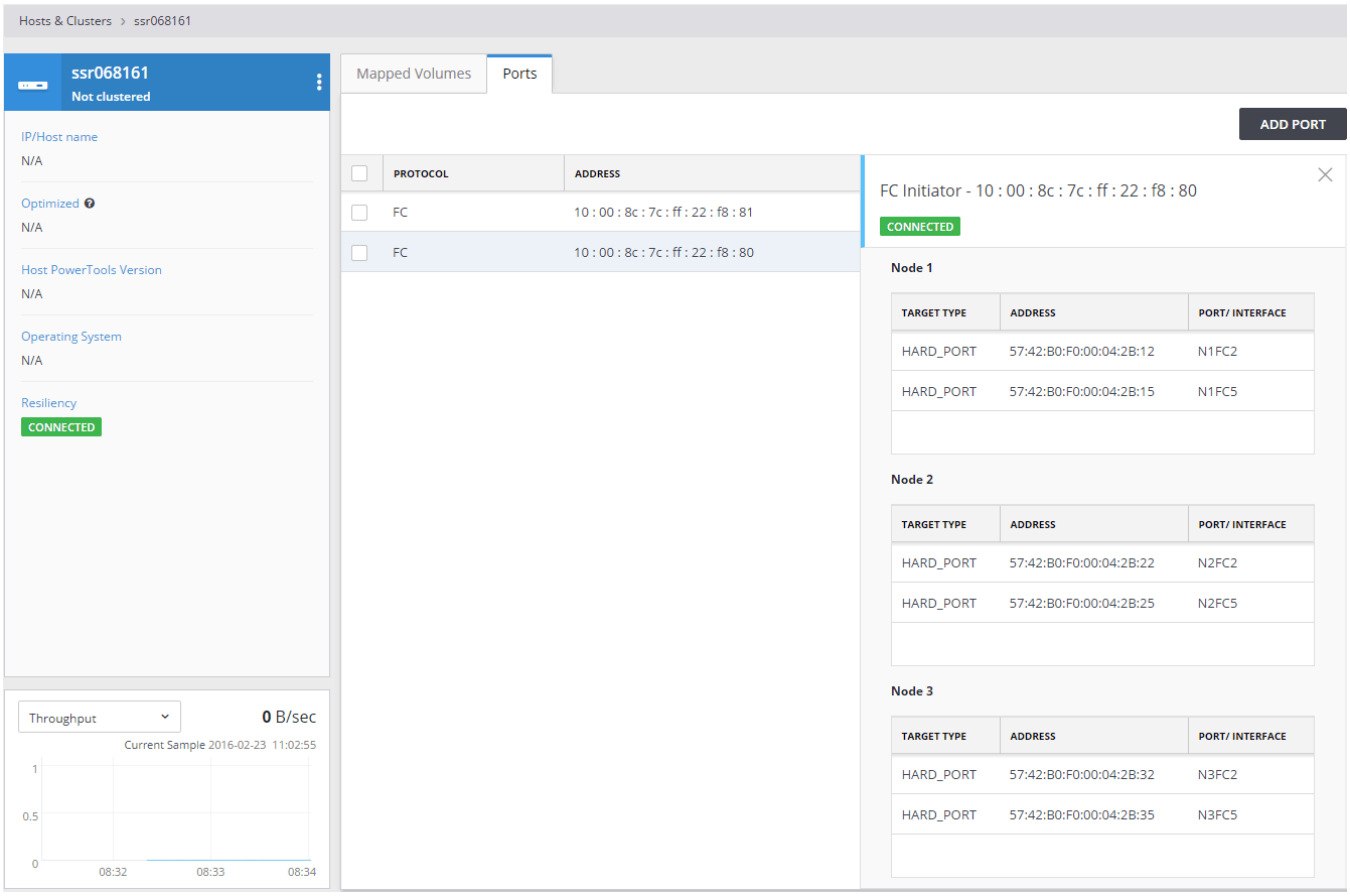


FIGURE 7 Infinidat Cluster Group Creation

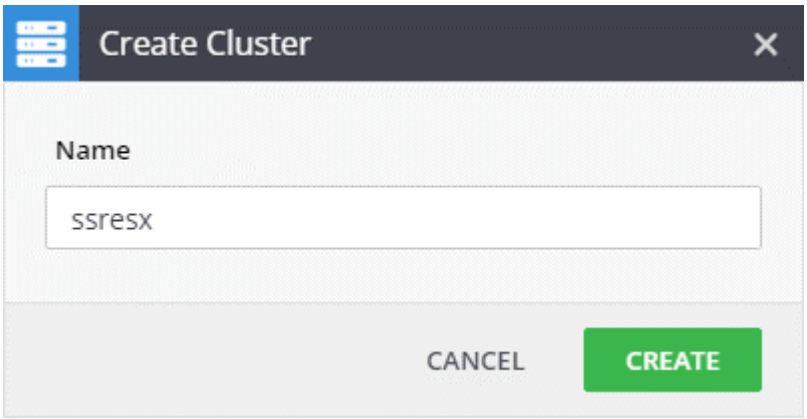
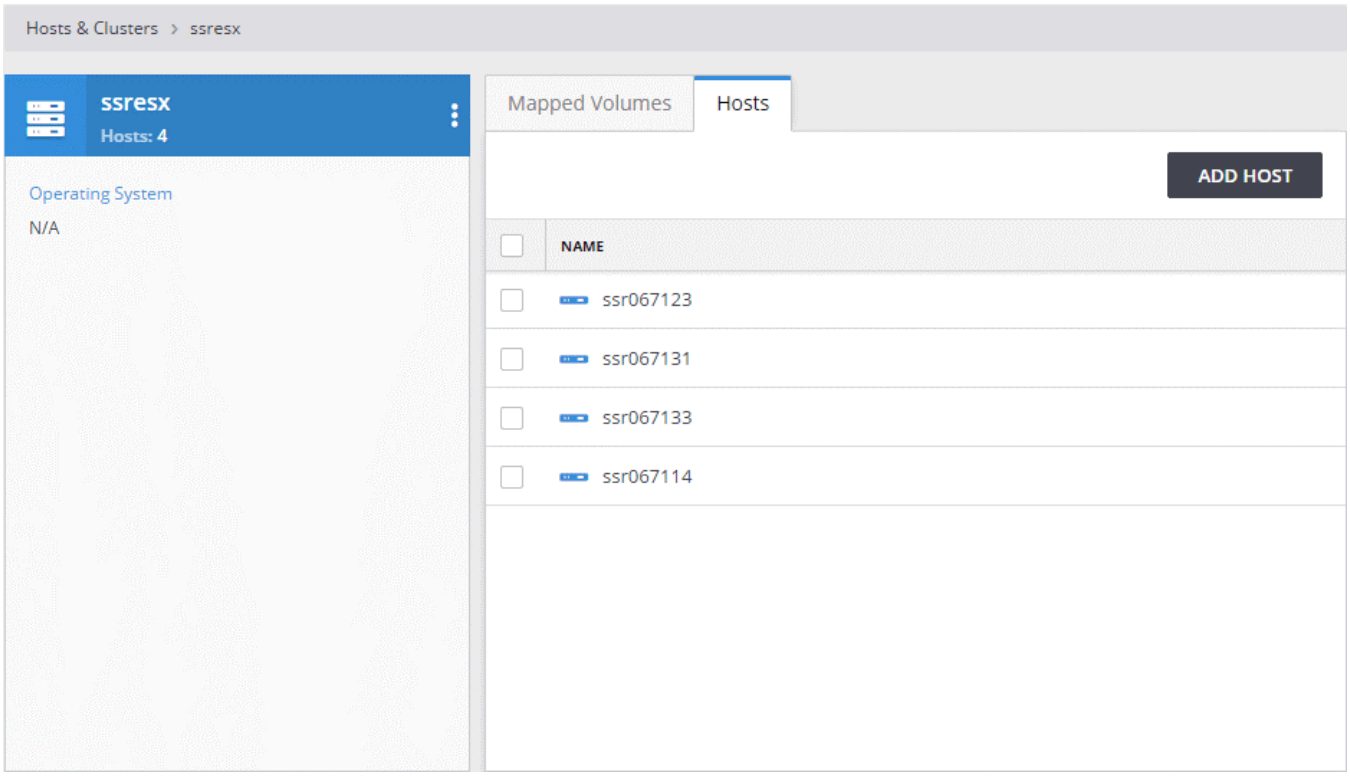


FIGURE 8 Infinidat Cluster Group Hosts



5. Create volumes and map them to the host/cluster groups. For the purpose of this testing, volumes are thick provisioned.

FIGURE 9 Infinidat Host Group Volume Mapping

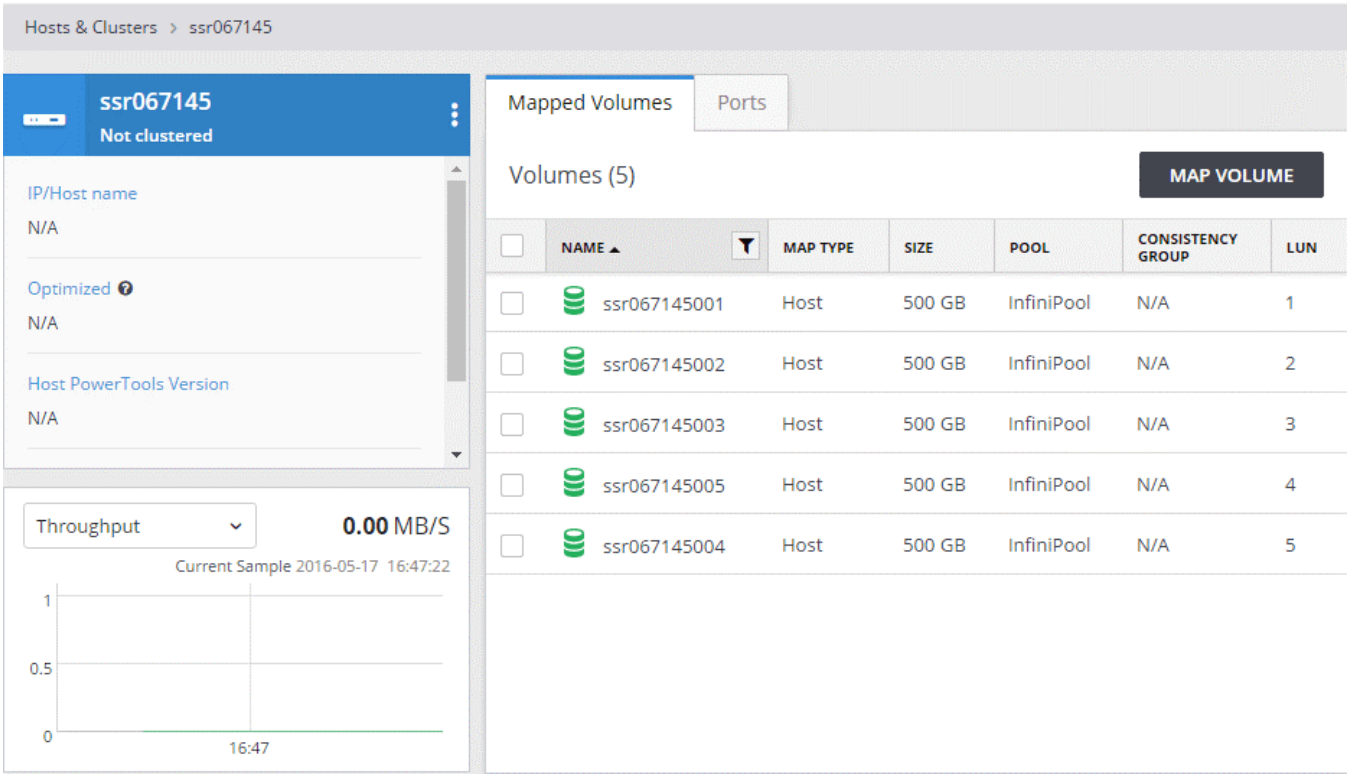
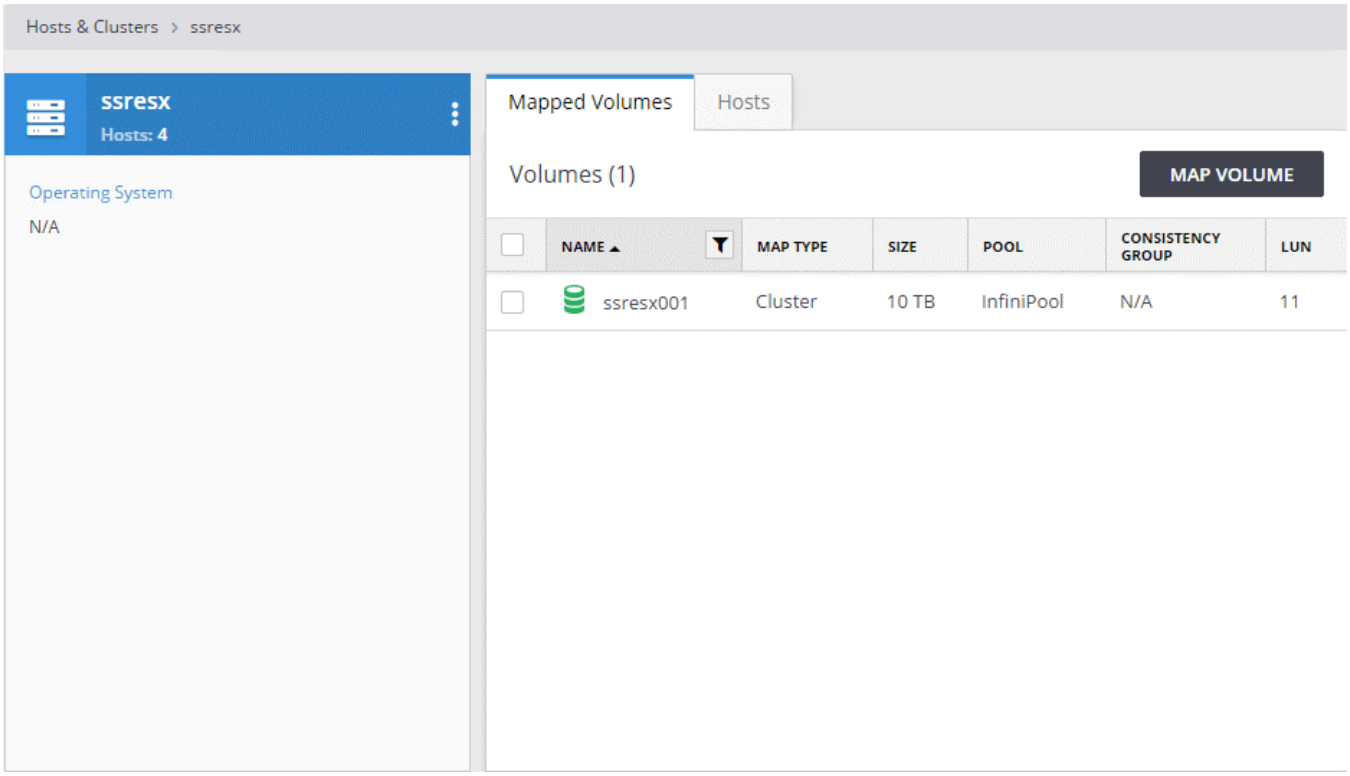


FIGURE 10 Infinidat Cluster Group Volume Mapping



Task 3: Host Setup

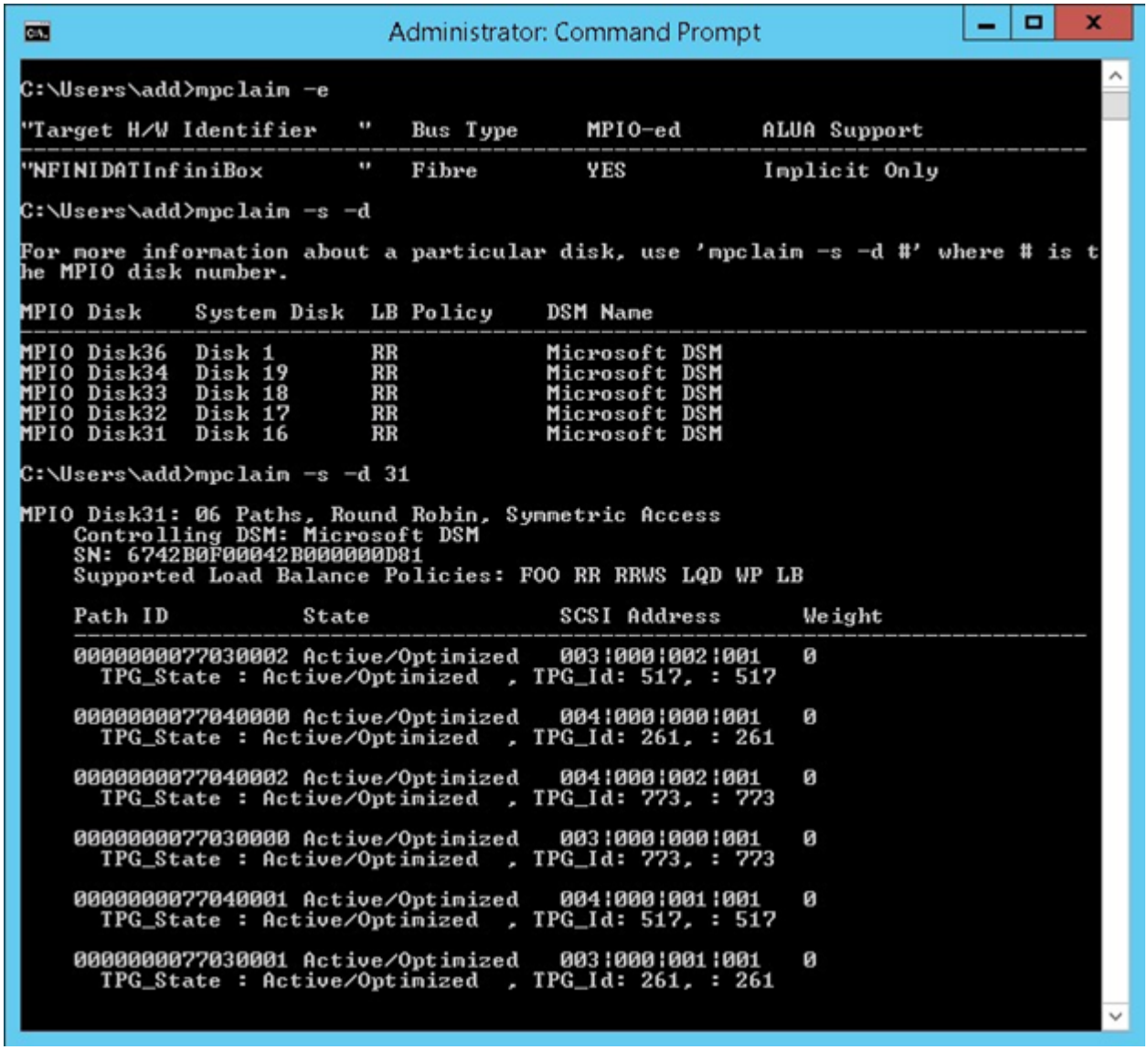
1. Provision a minimum of two uplinks from the host to the FC fabric for redundancy, and use native multipath tools to manage the available paths and load-balance across them.
2. Install the Infinidat Host Power Tools (HPT) application on all supported hosts. Run the "systems settings check"/"check host readiness" test from HPT. The test reports any parameters that are not per Infinidat recommendations, and it can be configured to automatically fix those settings. HPT can be used to configure the recommended multipath settings and any performance tuning parameters on the hosts and also to provision the storage LUNs.

3. Configuring the multipath settings allows for proper failover and load balancing across the available links. Multipath settings for Windows, Linux, and VMware as recommended by Infinidat are provided here. The following settings can be auto-configured using Infinidat Host Power Tools on supported OSs.

   - For Windows, native MPIO is used and the following parameters are tuned in the Windows Registry for optimal path recovery.

Registry	Value Name	Value
HKLM\SYSTEM\CurrentControlSet \Services\mpio\Parameters	PDORemovePeriod	20
HKLM\SYSTEM\CurrentControlSet \Services\mpio\Parameters	UseCustomPathRecoveryInterval	1
HKLM\SYSTEM\CurrentControlSet \Services\mpio\Parameters	PathRecoveryInterval	10

FIGURE 11 Windows MPIO Configuration



- For Linux, add the following to `/etc/multipath.conf`.

```
defaults {
        user_friendly_names yes
        max_fds                 8192
        find_multipaths yes
}
##
devices {
          device {
                vendor "NFINIDAT"
                product "InfiniBox.*"
                prio alua
                path_grouping_policy group_by_prio
                path_checker tur
                path_selector "round-robin 0"
                features "0"
                failback 30
```

```
                rr_weight priorities
                no_path_retry queue
        #       rr_min_io 1     # for kernels up to 2.6.31
                rr_min_io_rq 1 # for kernels 2.6.31 or newer
                flush_on_last_del yes
                fast_io_fail_tmo 5
                dev_loss_tmo 30
        }
}
```

Sample Output

```
# multipath -ll
mpathy (36742b0f00000042b0000000000000d93) dm-3 NFINIDAT,InfiniBox
size=466G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=50 status=active
  |- 8:0:0:4 sdo  8:224  active ready running
  |- 7:0:0:4 sdj  8:144  active ready running
  |- 7:0:2:4 sdw  65:96  active ready running
  |- 8:0:2:4 sdx  65:112 active ready running
  |- 8:0:1:4 sdu  65:64  active ready running
  `- 7:0:1:4 sdad 65:208 active ready running
```

- For VMware, the path selection policy is changed to Round Robin for the discovered Infinidat devices. The path selection policy for Infinidat LUNs can be set to the default, Round Robin, by creating a claiming rule on the VMware host.

FIGURE 12 Infinidat Devices Discovered on the VMware Host

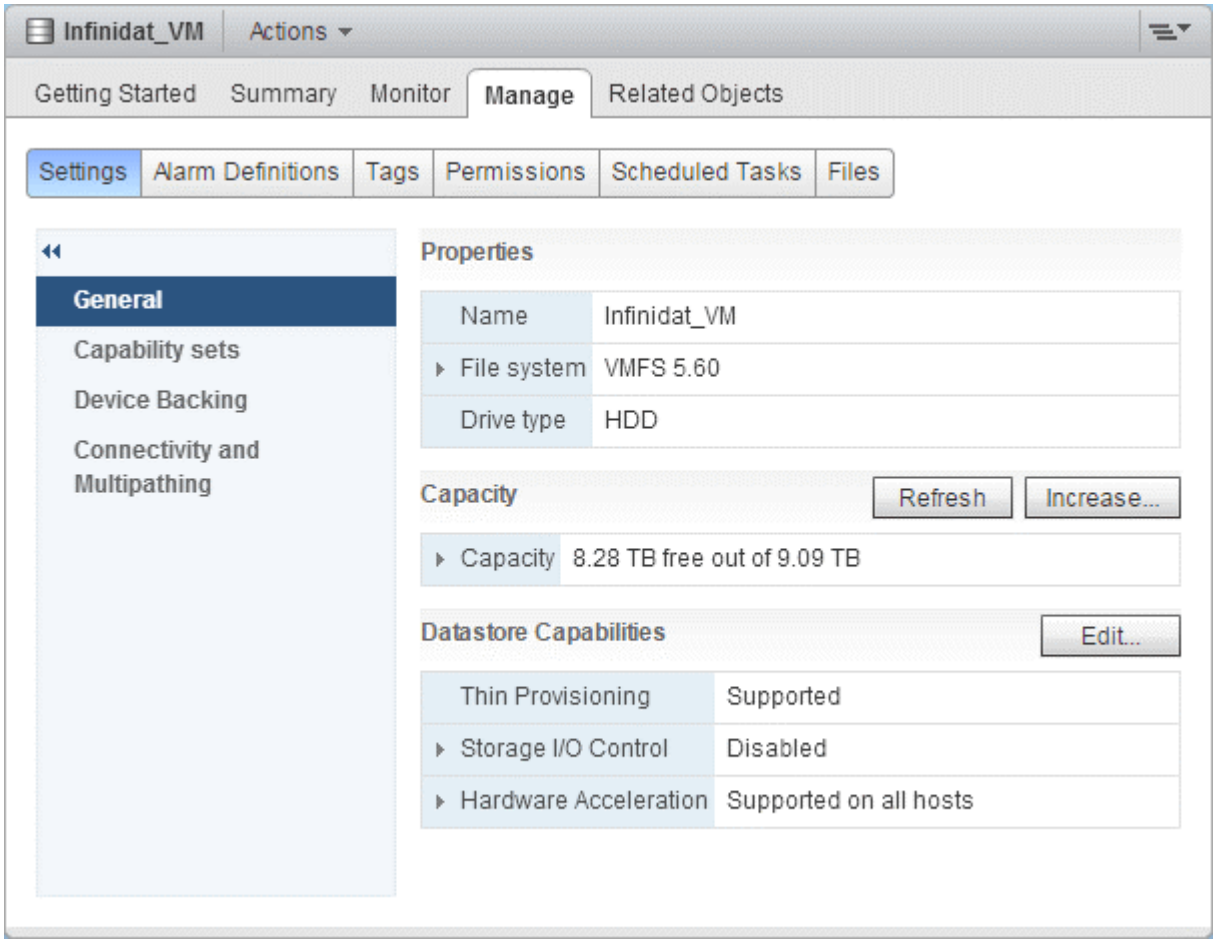
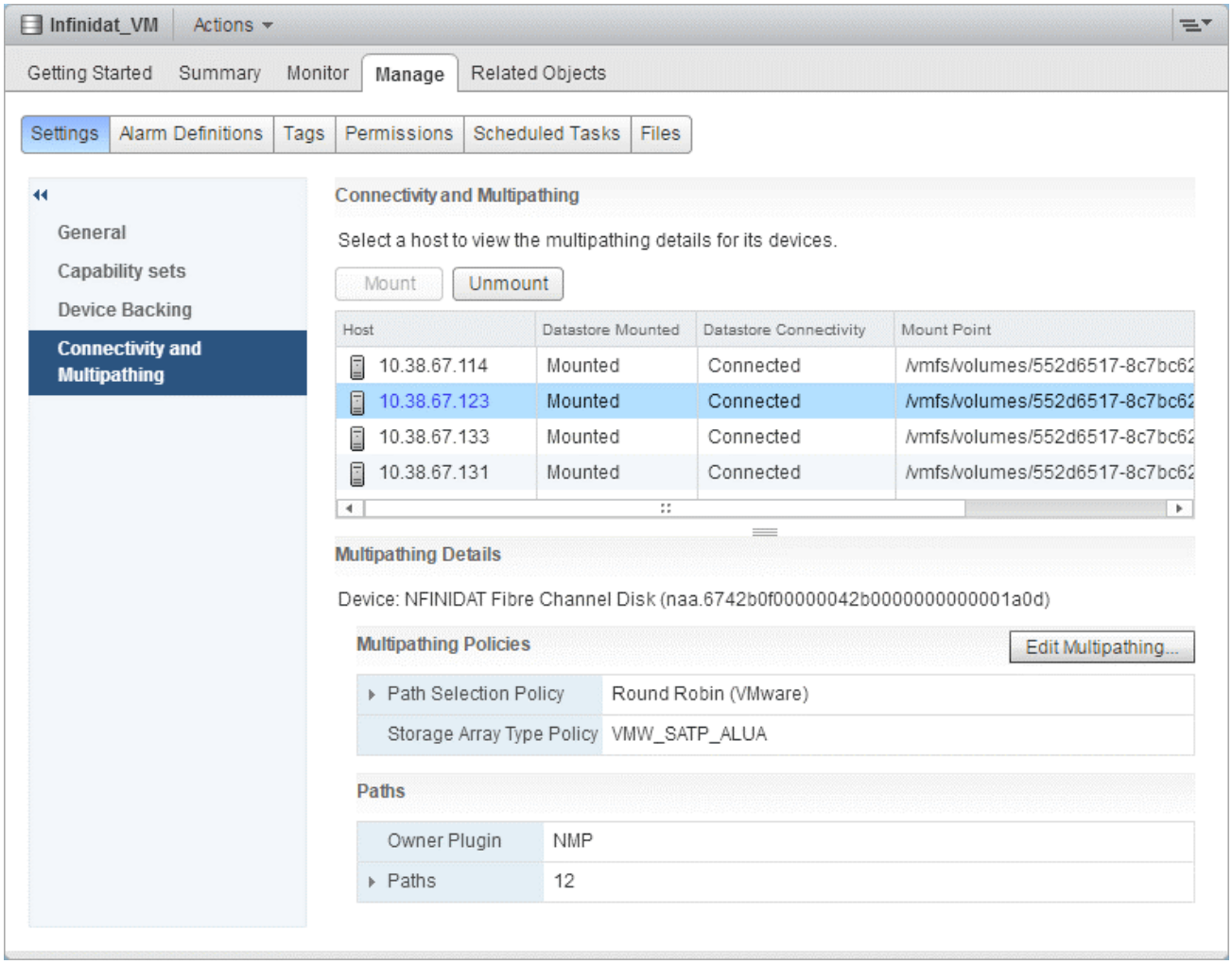


FIGURE 13 VMware Path Selection Policy for Infinidat Devices



4. Apply any host tuning parameters recommended by Infinidat for better operation with the storage array. The following settings can be auto-configured using Infinidat Host Power Tools on supported OSs.

   - For Linux, the following udev rule is created to persistently set the tuning parameters for Infinidat devices.

     On legacy udev systems, such as redhat-6, suse-11, and their equivalents, write the following contents to `/etc/udev/rules.d/41-infinidat-io-scheduler.rules`:

     ```
     ACTION=="add|change", KERNEL=="sd[a-z]*", SYSFS{vendor}=="NFINIDAT", RUN+="/bin/sh -c 'echo noop
     > /sys$DEVPATH/queue/scheduler'"
     ```

     On systemd-based systems, such as redhat-7, ubuntu-14.04, and their equivalents, write the following contents to `/lib/udev/rules.d/99-infinidat-io-scheduler.rules`:

     ```
     ACTION=="add|change", KERNEL=="sd[a-z]*", ENV{ID_VENDOR}=="NFINIDAT" RUN+="/bin/sh -c 'echo noop
     > /sys$env{DEVPATH}/queue/scheduler'"
     ```

   - For Windows and VMware, all default values are used and no other tuning parameters are applied.

Infinidat InfiniBox Test Report

What's New in This Report

- The array under test is the Infinidat InfiniBox running firmware version 4.0.5.30.
- The Brocade Fabric OS (FOS) version under test is 8.2.1.
- All HBAs have updated firmware and drivers.

Test History

Storage Model	Storage Firmware	Brocade FOS Version	Date
InfiniBox	1.7.1.2	FOS 7.4.0	December 2015
InfiniBox	2.2.1.1	FOS 8.0.1	December 2016
InfiniBox	3.0.0.150	FOS 8.1.0	March 2017
InfiniBox	3.0.20.100	FOS 8.2.0	February 2018
InfiniBox	4.0.5.30	FOS 8.2.1	November 2018

Test Plan Overview

The Infinidat InfiniBox array is connected to a Brocade FC fabric with two Fibre Channel target ports from each of the three controller nodes connected to the Brocade switches, as shown in the test configuration diagram.

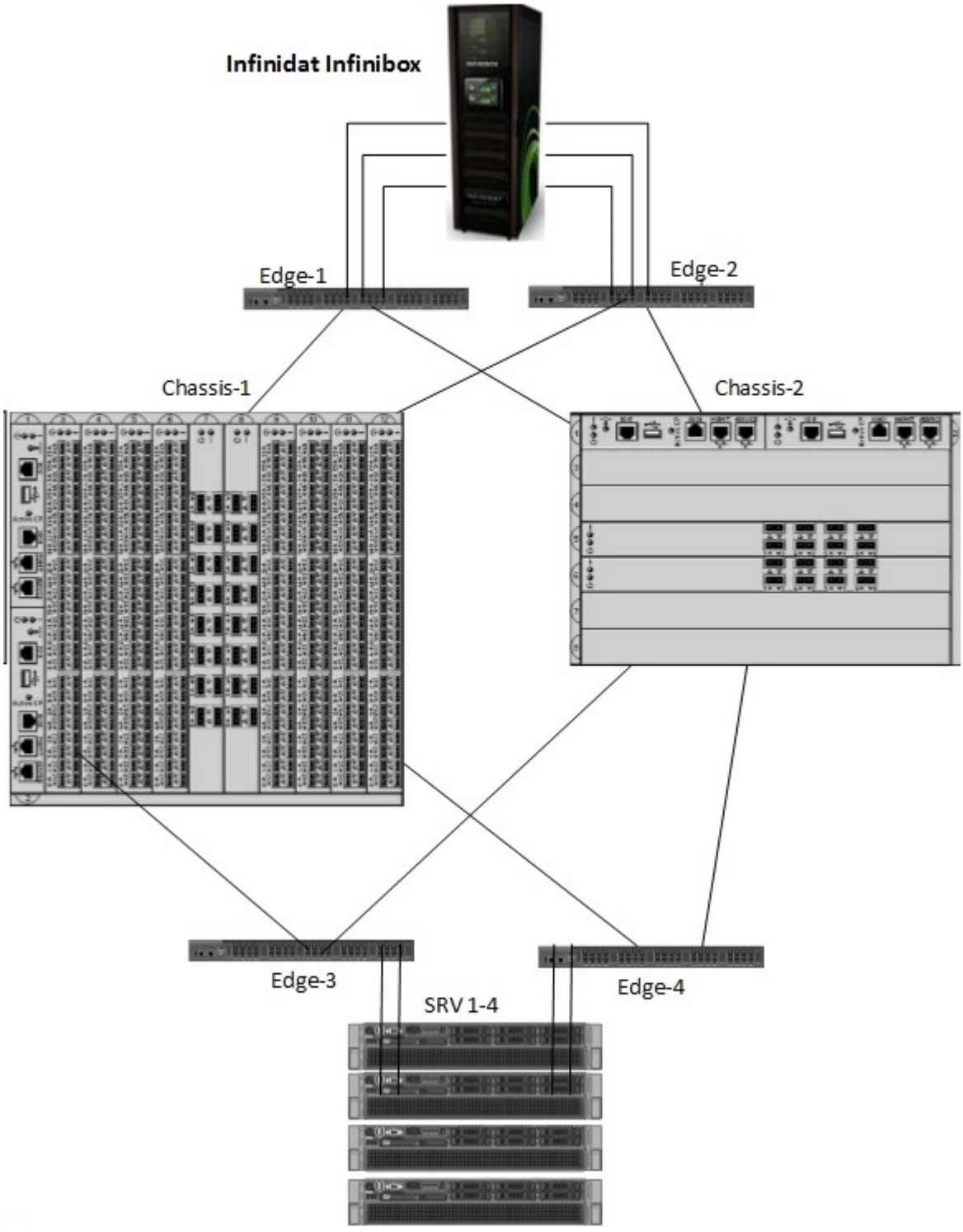
Scope

Testing focuses on interoperability of the Infinidat storage array and determining an optimal configuration for performance and availability.

Testing covers various I/O stress and error handling scenarios. Performance is observed within the context of best-practice fabric configuration; however, absolute maximum benchmark reporting of storage performance is beyond the scope of this test.

Details of the test steps are covered under "Test Cases" section. Standard test-bed setup includes IBM/HP/Dell servers with Brocade/QLogic/Emulex HBAs with two uplinks from every host to the Brocade FC fabric. I/O generator tools used include Medusa Labs Test Tools and VMware I/O Analyzer.

Test Configuration



DUT Descriptions

The following tables provide details about the devices under test (DUTs).

TABLE 1 Storage Array

DUT ID	Model	Vendor	Description
Infinidat InfiniBox	InfiniBox	Infinidat	The InfiniBox is an enterprise-class storage array with three active controller nodes having 24x8Gb FC ports for connectivity. The array has two drive enclosures, providing 250 TB of usable capacity. The array contains 24 SSD drives for cache on the 3 controllers.

TABLE 2 Switches

DUT ID	Model	Vendor	Description
Edge-1	Brocade 6510	Brocade	48-port Gen 5 16Gb FC switch
Edge-2	Brocade G620	Brocade	64-port Gen 6 32Gb FC switch
Edge-3	Brocade G610	Brocade	28-port Gen 6 32Gb FC switch
Edge-4	Brocade G620	Brocade	64-port Gen 6 32Gb FC switch
Chassis-1	Brocade X6-8	Brocade	8-slot Gen 6 32Gb director
Chassis-2	Brocade X6-4	Brocade	4-slot Gen 6 32Gb director

Test Cases

1.	**FABRIC INITIALIZATION—BASE FUNCTIONALITY**
1.1	Storage Device—Physical and Logical Login with Speed Negotiation
1.2	Zoning and LUN Mapping
1.3	Storage Device Fabric I/O Integrity
1.4	Storage Device Multipath Configuration—Path Integrity
2.	**FABRIC—ADVANCED FUNCTIONALITY**
2.1	Bottleneck Detection Using MAPS FPI—With Congested Host
2.2	Bottleneck Detection Using MAPS FPI—With Congested Fabric
2.3	Flow Monitoring with IO Insight and MAPS
2.4	QoS Integrity with QoS Zone-Based Traffic Prioritization
2.5	QoS Integrity with CS_CTL-Tag Based Frame Prioritization
2.6	Storage Device – FC Protocol Jammer Test Suite
2.7	Clear Link Diagnostics (D_Port) Test
3.	**STRESS AND ERROR RECOVERY WITH DEVICE MULTIPATH**
3.1	Storage Device Fabric I/O Integrity—Congested Fabric
3.2	Storage Device Integrity—Device Recovery from Port Toggle
3.3	Storage Device Integrity—Device Recovery from Device Relocation
3.4	Storage Device Stress—Device Recovery from Device Port Toggle—Extended Run
3.5	Storage Device Recovery—ISL Port Toggle (Sequential)
3.6	Storage Device Recovery—ISL Port Toggle (Entire Switch)
3.7	Storage Device Recovery—Director Blade Maintenance
3.8	Storage Device Recovery—Switch Offline
3.9	Storage Device Recovery—Switch Firmware Download
4.	**STORAGE DEVICE—FIBRE CHANNEL ROUTING (FCR) INTERNETWORKING TESTS**
4.1	Storage Device Internetworking Validation with the FC Host
4.2	Storage Device Internetworking Validation with FCoE Using VDX FlexPort
4.3	Storage Device Edge Recovery After FCR Disruptions
4.4	Storage Device Backbone Recovery After FCR Disruptions
5.	**OPTIONAL/ADDITIONAL TESTS**
5.1	Nondisruptive Firmware Upgrade on Storage Array
5.2	Workload Simulation Test Suite—Medusa
5.3	Workload Simulation Test Suite—VMware

1. Fabric Initialization—Base Functionality

1.1 Storage Device—Physical and Logical Login with Speed Negotiation

Test Objective

Verify device login to the switch and name server with all supported speed settings.

Test Configuration

Test Execution

Set switch ports to 4/8/16/Auto_Negotiate speed settings.

```
portcfgspeed <port> [4/8/16/0]
```

Result Validation

1. Validate link states on the array, and verify speed negotiation and device login at different speeds.
2. Check the switch port status and verify the "actual" and "configured" link speed. Check the name server for device login.

```
# nscamshow
# portshow X
```

Test Results

PASS. Speed negotiation, device login, and connectivity are verified.

1.2 Zoning and LUN Mapping

Test Objective

Verify that host-to-LUN access exists with valid zoning.

Test Configuration

Test Execution

1. Create peer zones on the fabric with the target WWNs as principal members and the initiator WWNs as peer members.
2. Create host groups and LUNs on the array with access to the initiator WWN.

Result Validation

Verify that LUNs are discovered on the hosts with host-specific tools.

- Linux: Check output of **lsscsi**.
- Windows: Check output at **Computer Management** > **Storage** > **Disk Management.**
- VMware: Check output at **Configuration** > **Storage** > **Devices.**

Test Results

PASS. Host has read/write access to the presented LUNs.

1.3 Storage Device Fabric I/O Integrity

Test Objective

Validate single-path host-to-LUN I/O with read/write/verify testing. Include short device cable pulls/port toggles to validate device recovery.

Test Configuration

Test Execution

1. Set up read/write I/O to the LUN using Medusa.
2. Perform link disruptions by port toggles, cable pulls.

Result Validation

Check Medusa I/O logs, and verify that I/O resumes after a short downtime. Medusa I/O will pause, but should recover without errors.

Test Results

PASS. I/O resumes after the disruption.

1.4 Storage Device Multipath Configuration—Path Integrity

Test Objective

Verify that multipath configures successfully. Each adapter and storage port to reside in different switches. For all device paths, consecutively isolate individual paths and validate I/O integrity and path recovery.

Test Configuration

Test Execution

1. Set up the host with at least two initiator ports zoned with two target ports on the array.
2. Set up multipath on the host, and start I/O.
3. Perform sequential port toggles across the initiator and target switch ports to isolate paths.

Result Validation

1. Check host multipath properties to verify that the toggled path recovers.

    - Windows: **mpclaim –s –d**
    - Linux: **multipath –ll**
    - VMware: Check the paths at **Configuration** > **Storage** > **Devices** > **Manage Paths**

2. Check the host and storage logs for failures.

3. Check the switch error logs and switch port status after toggling.

```
# errdumpall
# portstatsshow X
# portshow X
```

4. Check I/O logs, and verify that I/O continues without errors.

Test Results

PASS. I/O fails over to the remaining active paths and recovers when the disrupted path is restored.

2. Fabric—Advanced Functionality

2.1 Bottleneck Detection Using MAPS FPI—With Congested Host

Test Objective

Verify that congestion on host ports is detected. Verify storage device and fabric behavior during congestion.

Test Configuration

Test Execution

1. Enable MAPS monitoring and MAPS FPI on all switches. A Fabric Vision license is required.
2. Start I/O from a single host initiator to multiple targets.
3. Monitor the switch logs for congestion and latency (IO_PERF_IMPACT/IO_FRAME_LOSS) warnings.

Result Validation

Check the switch error logs and the MAPS dashboard for bottleneck warnings.

```
# errdumpall | grep IO_
# mapsdb --show all
```

Test Results

PASS. The bottlenecked ports are displayed on the MAPS dashboard, and a RASLog warning is created.

2.2 Bottleneck Detection Using MAPS FPI—With Congested Fabric

Test Objective

Create congestion on a switch ISL port. Verify that congestion in the fabric is detected. Verify storage device and fabric behavior during the congestion.

Test Configuration

Test Execution

1. Enable MAPS monitoring and MAPS FPI on all switches. A Fabric Vision license is required.
2. Isolate a single ISL in the fabric.
3. Start I/O from multiple host initiators to multiple targets.
4. Monitor the switch logs for congestion and latency (IO_PERF_IMPACT/IO_FRAME_LOSS) warnings.

Result Validation

Check the switch error logs and MAPS dashboard for bottleneck warnings.

```
# errdumpall | grep IO_

# mapsdb --show all
```

Test Results

PASS. The bottlenecked ports are displayed on the MAPS dashboard, and a RASLog warning is created.

2.3 Flow Monitoring with IO Insight and MAPS

Test Objective

Monitor IO latency statistics on the target ports. Verify that stats are reported accurately and that alerts are generated when thresholds are hit.

Test Configuration

1. Baseline a target LUN's latency running 4k reads from a workload generator. Here we are seeing about a 500–1000 microsecond range. Hence we set the rules to a 1000-µs or 1-ms threshold.

   ```
   Avg Completion Time0.000853
   ```

2. Set up flows and monitoring as per "Step 2 - Configure Flow Monitoring with I/O Insight" under "Task 1. Brocade FC Fabric Configuration."
3. Start I/O and adjust the traffic pattern to cause a rise in latency above the configured monitoring threshold; and confirm that RASLog, MAPS dashboard, and email notifications are generated.

Result Validation

1. Check flow statistics and MAPS alerts to verify the metrics are reported correctly and alerts are generated when thresholds are crossed.

```
# flow --show
```

```
RASLog:
> errdumpall
2017/02/08-08:46:58:582892, [MAPS-1003], 168254/165960, FID 128, WARNING, G620_066_223, Flow (ios_infini_3),
Condition=ios_infini_3(RD_STATUS_TIME_LT_8K/min>=1000), Current Value:[ RD_STATUS_TIME_LT_8K,1051
Microseconds], RuleName=ios_infini_3_4k_rd_status, Dashboard Category=Traffic Performance., raslogAction.c,
line: 96, comp:md, ltime: 2017/02/08-08:46:58:582642
```

```
Email:
Switch Time:Feb 08 08:46:58
Affected Entity:Flow (ios_infini_3)
Monitor:Read completion time (RD_STATUS_TIME_LT_8K)
Rule Name:ios_infini_3_4k_rd_status
Group:ios_infini_3
Condition:ios_infini_3(RD_STATUS_TIME_LT_8K/min>=1000)
Current Value:1051 Microseconds
Dashboard Category:Traffic Performance
Switch Name:G620_066_223
Switch WWN:10:00:c4:f5:7c:2a:8b:c8
Switch IP:10.38.66.223
Fabric Name:SSR
VFID:128

MAPS Dashboard:
> mapsdb –show

3.2 Rules Affecting Health:
==========================

Category(Rule Count)|Repeat|Rule Name                |Execution Time   |Object             |Triggered Value|
                    |Count |                         |                 |                   |(Units)        |
-------------------------------------------------------------------------------------------------------------
Traffic Performance(|2     |ios_infini_3_4k_rd_status|05/25/16 15:46:58|Flow (ios_infini_3)|1051
                                                                                          |Microseconds
```

Test Results

PASS. Verified that I/O stats are reported and that alerts are generated when thresholds are exceeded.

2.4 QoS Integrity with QoS Zone-Based Traffic Prioritization

Test Objective

Verify storage device behavior, and validate traffic characteristics with different QoS zones.

Test Configuration

Test Execution

1. Set up initiator-target pairs with Low/Medium/High QoS zones in the fabric.

```
zone:  QOSH_ssr_infinidat_2
                00:02:00:00:00:03:00:03; 57:42:b0:f0:00:04:2b:11;
                57:42:b0:f0:00:04:2b:21; 57:42:b0:f0:00:04:2b:31;
                10:00:8c:7c:ff:24:6d:00; 10:00:8c:7c:ff:24:6d:01;
                21:00:00:0e:1e:18:99:90; 21:00:00:0e:1e:18:99:91
```

2. Start I/O from all hosts, and verify I/O statistics.

Result Validation

1. 1. Check the I/O logs, and verify that I/O continues without error.
2. Check the switch error logs and switch port status for errors.

```
# errdumpall

# porterrshow

0    0    0    0    0    0    0    946    1    0    1    0    0    0    0    0
0    0    0    0    0    0    0    30     1    0    1    0    0    0    0    0
```

```
0    0    0    0    0    0    0    0      0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    0      0    0    0    0    0    0    0    0
```

Test Results

PASS. I/O from hosts in all QoS zones completes successfully without errors.

2.5 QoS Integrity with CS_CTL-Based Frame Prioritization

Test Objective

Verify CS_CTL I/O prioritization using the Emulex ExpressLane feature.

Test Configuration

Test Execution

1. Configure all switches in the fabric to be in "Auto" CS_CTL QoS mode.

```
root> configurechassis
Configure...
cfgload attributes (yes, y, no, n): [no]
Custom attributes (yes, y, no, n): [no]
  system attributes (yes, y, no, n): [no]
  fos attributes (yes, y, no, n): [no] y

   Reboot needed to effect new CSCTL Mode
        CSCTL QoS Mode (0 = default; 1 = auto mode): (0..1) [0] 1

root> configshow -all | grep csctlMode
fos.csctlMode:1
```

2. Enable CS_CTL mode on the initiator and target switch ports.

```
root> portcfgqos –enable [slot/]port csctl_mode

root> portcfgshow 22
. . .
CSCTL mode:              ON
. . .
```

3. Set up initiator-target zones in the fabric, and discover the LUNs on the host.
4. Enable the ExpressLane feature on the host Emulex ports, and set the ExpressLane priority (CS_CTL value) to high (3).
5. Enable ExpressLane on any discovered LUNs, and start write I/O from the host.
6. Verify I/O statistics and CS_CTL prioritization in the fabric.

Result Validation

1. Check I/O logs, and verify that I/O continues without errors for all LUNs.
2. Verify that I/O performance improves on ExpressLane-enabled LUNs.
3. Verify CS_CTL prioritization in the fabric by monitoring the high VC buffer credits on the ISLs.

```
root> portregshow 0 | grep -E "_trc |bbc_mbc"
0x88982800: bbc_trc4    0    2    2    2    2    1    1
0x88982820: bbc_trc2    2    2    2    2    2    2    0
0x88982840: bbc_trc0    0    0    0    0    0    0    0
```

```
0x88982860: bbc_trc0    0    0    0    0    0    0    0
0x88982880: bbc_trc0    0    0    0    0    0    0    0
```

4. Check the switch error logs and the switch port status for errors.

```
# errdumpall

# porterrshow

0:    1.7g    2.7g      0    0    0    0    0    0    0    946    1    0    1    0    0    0    0
0
1: 431.1m 431.4m  0    0    0    0    0    0    0     30    1    0    1    0    0    0    0    0
2:    0 0               0    0    0    0    0    0    0      0    0    0    0    0    0    0    0
0
3:    0 0               0    0    0    0    0    0    0      0    0    0    0    0    0    0    0
0
```

Test Results

PASS. I/O completes successfully for all LUNs. ExpressLane-enabled LUN performance is improved, and I/O to the LUN is prioritized throughout the FC fabric.

2.6 Storage Device—FC Protocol Jammer Test Suite

Test Objective

Perform FC Jammer tests including areas such as CRC corruption, packet corruption, missing frames, host error recovery, and target error recovery.

Test Configuration

Test Execution

1. Insert the Jammer device in the I/O path on the storage end.
2. Execute the following Jammer scenarios:
   - Delete one frame.
   - Delete R_RDY.
   - Replace CRC of data frame.
   - Replace EOF of data frame.
   - Replace "good status" with "check condition".
   - Replace IDLE with LR.
   - Truncate frame.
   - Create S_ID/D_ID error of data frame.

Result Validation

1. Check the host and storage logs for errors.
2. Check the switch logs and interface stats for errors.

```
# errdumpall

# porterrshow
```

```
0    0    0    0    0    0    0    946    1    0    1    0    0    0    0    0
0    0    0    0    0    0    0    30     1    0    1    0    0    0    0    0
0    0    0    0    0    0    0     0     0    0    0    0    0    0    0    0
0    0    0    0    0    0    0     0     0    0    0    0    0    0    0    0
```

3. Verify Jammer operations and recovery with Analyzer.

Test Results

PASS. Host and target recover from errors and continue I/O operations.

2.7 Clear Link Diagnostics (D_Port) Test

Test Objective

Execute the Clear Link Diagnostics feature on all supported HBAs, and check for any port and link problems.

Tested with FOS 7.4.0.

Test Configuration

Test Execution

1. The Clear Link Diagnostic test is performed with Emulex LPe16202, QLogic 2672, and QLogic 1860 HBAs.
2. The Brocade FC switch can be configured in static, dynamic, and on-demand D_Port modes. The switch supports dynamic D_Port by default at the chassis level.

```
root> configure
Configure...
  Fabric parameters (yes, y, no, n): [no]
  D-Port Parameters (yes, y, no, n): [no] y
   Dynamic D-Port (on, off): [on]
On Demand D-Port (on, off): [off]
```

3. With the Emulex LPe16202 adapter, the switch port is in dynamic mode and the test is manually initiated on the HBA port from the Emulex OneCommand Manager.

```
root> switchshow
0   0   030000   id    N16      In_Sync     FC  D-Port 10:00:00:90:fa:61:92:3b Dynamic
```

4. With the QLogic 2672 adapter, the HBA port is in dynamic D_Port mode and the test is initiated by configuring the switch port as a static D_Port.

```
root> portdisable 19
root> portcfgdport --enable 19

Caution: D_Port functionality is only available on 16Gb-capable platforms with 16Gb FC SFPs, 10Gb FC
SFPs, 8Gb LWL/ELWL FC SFPs, QSFPs or QSFP+.
```

5. With the QLogic 1860 adapter, the HBA port can be in static or dynamic D_Port mode.

```
port disabled

# bcu diag --dportenable 2/0
D-port mode for port 2/0 enabled.
```

6. Perform the D_Port test, and verify that all tests pass and that no port and link problems are reported.

Result Validation

Check the D_Port test results on the switch ports and host HBA diagnostic utilities.

Mode = Manual -> Static D_Port

Mode = Automatic -> Dynamic D_Port

```
root> portdporttest --show X
```

Test Results

PASS. D_Port tests passed with all tested HBAs.

3. Stress and Error Recovery with Device Multipath

3.1 Storage Device Fabric I/O Integrity—Congested Fabric

Test Objective

1. From all available initiators, start a mixture of read/write/verify traffic with random data patterns continuously to all their targets overnight.
2. Verify that no host application failover or unexpected change in I/O throughput occurs.
3. Configure the fabric and devices for maximum link and device saturation.

Test Configuration

Test Execution

1. Start FC I/O to the storage array from multiple hosts.
2. Set up a mix of read/write traffic.

Result Validation

1. Check the host and storage logs for errors.
2. Verify the link congestion, and check the switch logs for errors.

```
# errdumpall

# portperfshow

# porterrshow

0    0    0    0    0    0    0    946    1    0    1    0    0    0    0    0
0    0    0    0    0    0    0     30    1    0    1    0    0    0    0    0
0    0    0    0    0    0    0      0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0      0    0    0    0    0    0    0    0    0
```

3. Check the I/O generator tool logs to verify that I/O runs without errors.

Test Results

PASS. All I/O completes without errors. All validation checks pass.

3.2 Storage Device Integrity—Device Recovery from Port Toggle

Test Objective

1. With I/O running, perform a quick port toggle on every storage device and adapter port.
2. Verify that host I/O recovers.
3. Perform the test sequentially for each storage device and adapter port.

Test Configuration

Test Execution

1. Set up multipath on the host, and start I/O.
2. Perform multiple iterations of sequential port toggles across the initiator and target switch ports.

Result Validation

1. Check the switch port status after toggling, and check for errors in the switch logs.

```
# errdumpall

# portstatsshow X

# portshow X
```

2. Check the multipath status on hosts to verify that the toggled path recovers.

```
-    Windows: mpclaim -s -d
-    Linux: multipath -ll
-    VMware: Check the paths at Configuration > Storage > Devices > Manage Paths
```

3. Check host and storage error logs, and verify that I/O continues without errors.

Test Results

PASS. I/O fails over and recovers successfully. All validation checks pass.

3.3 Storage Device Integrity—Device Recovery from Device Relocation

Test Objective

1. With I/O running, manually disconnect a port and reconnect it to a different switch in the same fabric.
2. Verify that host I/O fails over to an alternate path and that the toggled path recovers.
3. Perform the test sequentially for each storage device and adapter port.
4. Repeat the test for all switch types.

Test Configuration

Test Execution

1. Set up multipath on the host, and start I/O.
2. Move storage target ports to different switch ports in the fabric.

Result Validation

1. Check for errors in the switch logs and the status of the new switch port.

```
# errdumpall

# portstatsshow X

# portshow X
```

2. Check the multipath status on hosts to verify that the toggled path recovers.

- Windows: **mpclaim -s -d**
- Linux: **multipath -ll**
- VMware: Check the paths at **Configuration** > **Storage** > **Devices** > **Manage Paths**

3. Check host and storage error logs and verify that I/O continues without errors.

Test Results

PASS. I/O fails over and recovers successfully. All validation checks pass.

3.4 Storage Device Stress—Device Recovery from Device Port Toggle—Extended Run

Test Objective

1. Sequentially toggle each initiator and target port in the fabric.
2. Verify that host I/O recovers to an alternate path and that the toggled path recovers.
3. Run the test for 24 hours.

Test Configuration

Test Execution

1. Set up multipath on the host, and start I/O.
2. Perform multiple iterations of sequential port toggles across the initiator and target switch ports.

Result Validation

1. Check the switch port status after toggling, and check for errors in the switch logs.

```
# errdumpall

# portstatsshow X

# portshow X
```

2. Check the multipath properties for iSCSI hosts to verify that the toggled path recovers.

```
On Windows: mpclaim -s -d
On Linux: multipath -ll
On VMware: Check the paths at Configuration > Storage > Devices > Manage Paths
```

3. Check host and storage error logs, and verify that I/O continues without errors.

Test Results

PASS. I/O fails over and recovers successfully. All validation checks pass.

3.5 Storage Device Recovery—ISL Port Toggle (Sequential)

Test Objective

1. Sequentially toggle each ISL path on all switches. Host I/O may pause but should recover.
2. Verify fabric ISL path redundancy between hosts and storage devices.
3. Verify host I/O throughout the test.

Test Configuration

Test Execution

1. Set up host multipath with links on different switches in the FC fabric, and start I/O.
2. Ensure ISL redundancy by provisioning multiple ISLs connected to different switches to provide multiple paths through the fabric.

```
# islshow
```

3. Perform multiple iterations of sequential ISL toggles across the fabric.

Result Validation

1. Check the FC fabric status after ISL toggles. Verify that all nodes are online.

```
# fabricshow
```

2. Check the switch logs for errors, and verify that I/O fails over to an alternate ISL path in the fabric.

```
# errdumpall

# portperfshow

# porterrshow
```

3. Check host and storage error logs, and verify that I/O continues without errors.

Test Results

PASS. I/O re-routes to available paths in the fabric and recovers when the link is restored. All validations checks pass.

3.6 Storage Device Recovery—ISL Port Toggle (Entire Switch)

Test Objective

1. Sequentially, and for all switches, disable all ISLs on the switch under test.
2. Verify fabric switch path redundancy between hosts and storage devices.
3. Verify that the switch can merge back in to the fabric.
4. Verify the host I/O path throughout the test.

Test Configuration

Test Execution

1. Set up host multipath with links on different switches in the FC fabric, and start I/O.
2. Ensure ISL redundancy by provisioning multiple ISLs connected to different switches to provide multiple paths through the fabric.

```
# islshow
```

3. Perform multiple iterations of sequentially disabling all ISLs on a switch in the fabric.

Result Validation

1. Check the FC fabric status after ISL toggling. Verify that all nodes are online.

```
# fabricshow
```

2. Check the switch logs for errors, and verify that I/O fails over to an alternate ISL path in the fabric.

```
# errdumpall
# portperfshow
# porterrshow
```

3. Check host and storage error logs, and verify that I/O continues without errors.

Test Results

PASS. I/O fails over to an alternate path and recovers once the switch merges back in the fabric. All validations checks pass.

3.7 Storage Device Recovery—Director Blade Maintenance

Test Objective

1. Toggle each blade on the director in sequential order.
2. Include blade enable/disable, power on/off, and reboot testing.

Tested with FOS 7.4.0.

Test Configuration

Test Execution

1. Uplink edge switch ISLs to different blades on the directors.
2. Set up host multipath with links on different switches in the FC fabric, and start I/O.
3. Perform multiple iterations of sequential disable/enable, power on/off, and reboot of all blades on the 8510 directors.

Result Validation

1. Check the FC fabric status after toggling the blades. Verify that all nodes are present in the fabric.

```
# fabricshow
```

2. Check the switch logs for errors, and verify that I/O fails over to an alternate ISL path in the fabric.

```
# errdumpall

# portperfshow

# porterrshow

0     0     0     0     0     0     0     946     1     0     1     0     0     0     0     0
0     0     0     0     0     0     0      30     1     0     1     0     0     0     0     0
0     0     0     0     0     0     0       0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0       0     0     0     0     0     0     0     0     0
```

3. Check host and storage error logs, and verify that I/O continues without errors.

Test Results

PASS. I/O fails over to an alternate path and recovers once the blade recovers from the disruption.

3.8 Storage Device Recovery—Switch Offline

Test Objective

1. Toggle each switch in sequential order.
2. Include switch enable/disable, power on/off, and reboot testing.

Test Configuration

Test Execution

1. Set up host multipath with links on different switches in the FC fabric, and start I/O.
2. Perform multiple iterations of sequential disable/enable, power on/off, and reboot of all switches in the fabric.

Result Validation

1. Check the FC fabric status after toggling the switches. Verify that all nodes are present in the fabric.

```
# fabricshow
```

2. Check the switch logs for errors, and verify that the toggled switch recovers.

```
# errdumpall
# switchshow
```

3. Check host and storage error logs, and verify that I/O continues without errors.

Test Results

PASS. I/O fails over to an alternate path and recovers once the switch merges back in the fabric.

3.9 Storage Device Recovery—Switch Firmware Download

Test Objective

1. Sequentially perform the firmware maintenance procedure on all device-connected switches under test.
2. Verify that host I/O continues (with minimal disruption) through the "firmware download" and that device pathing remains consistent.

Test Configuration

Test Execution

1. Set up host multipath with links on different switches in the FC fabric, and start I/O.
2. Sequentially perform firmware upgrades on all switches in the fabric.

Result Validation

1. Verify that the firmware upgrade completes successfully on each switch node and that they merge back in the FC fabric.

```
# version
# fabricshow
```

2. Check I/O generator tool logs to verify that I/O runs without errors throughout the firmware upgrade.
3. Check the switch logs for errors, and verify that I/O resumes on the node after the firmware upgrade is complete.

```
# errdumpall
# portperfshow
```

Test Results

PASS. I/O operations complete without errors. I/O fails over to an alternate path during the switch reload after the firmware upgrade, and it resumes after the switch is online. All validation checks pass.

4. Storage Device Fibre Channel Routing (FCR) Internetworking Tests

(Covered in Fabric OS 8.1.0 and previous releases.)

4.1 Storage Device Internetworking Validation with the FC Host

Test Objective

1. Configure two FC fabrics with FCR.
2. Verify that edge devices are imported into adjacent name servers and that hosts have access to their routed targets after FC routers are configured.

Test Configuration

Test Execution

1. Set up FCR in an Edge-Backbone-Edge configuration.
2. Set up LSAN zoning, verify host access to target LUNs, and start I/O.

Result Validation

1. Verify the name server and FCR fabric state.

```
# fcrfabricshow

# fcrproxydevshow
```

2. Verify that I/O runs successfully without error.

Test Results

PASS. I/O completes successfully. Both edge fabrics have the corresponding proxy name server entries for the host and target ports.

4.2 Storage Device Internetworking Validation with FCoE Using VDX FlexPort

Test Objective

1. Configure an FC fabric with FCR connected to an FCoE fabric.
2. Verify that edge devices are imported into adjacent name servers and that hosts have access to their routed targets after FC routers are configured.

Tested with FOS 7.4.0.

Test Configuration

Test Execution

1. Add an FCoE VCS fabric to the FCR setup.
2. Set up LSAN zoning, verify host access to target LUNs, and start I/O.

Result Validation

1. Verify the name server and FCR fabric state.

```
# fcrfabricshow
# fcrproxydevshow
```

2. Verify that I/O runs successfully without errors.

Test Results

PASS. I/O completes successfully. Both edge fabrics have the corresponding proxy name server entries for the host and target ports.

4.3 Storage Device Edge Recovery After FCR Disruptions

Test Objective

1. Configure FCR in an edge-backbone-edge configuration.
2. With I/O running, validate device access and pathing.
3. Perform reboots, switch disables, and port toggles on edge connections to disrupt device pathing and I/O.
4. Verify path and I/O recovery once switches and ports recover.

Test Configuration

Test Execution

1. Set up FCR in an edge-backbone-edge configuration.
2. Set up LSAN zoning, verify host access to target LUNs, and start I/O.
3. Perform sequential reboots, switch disables, and ISL port toggles on the switches in the edge fabric.

Result Validation

1. Verify the FCR fabric state throughout the disruptions.

```
# fcrfabricshow
# fcrproxydevshow
```

2. Check the switch logs for errors.

```
# errdumpall
# portperfshow
```

3. Check host and storage logs, and verify that I/O runs without errors.

Test Results

PASS. I/O fails over to an available switch path and recovers when the disrupted switch is restored.

4.4 Storage Device Backbone Recovery After FCR Disruptions

Test Objective

1. Configure FCR in a backbone-edge configuration.
2. With I/O running, validate device access and pathing.
3. Perform reboots, switch disables, and port toggles on backbone connections to disrupt device pathing and I/O.
4. Verify path and I/O recovery once switches and ports recover.

Test Configuration

Test Execution

1. Connect array target ports to the backbone fabric in an edge-backbone configuration.
2. Set up LSAN zoning, verify host access to target LUNs, and start I/O.
3. Perform sequential reboots, switch disables, and ISL port toggles on the switches in the backbone fabric.

Result Validation

1. Verify the FCR fabric state throughout the disruptions.

```
# fcrfabricshow
# fcrproxydevshow
```

2. Check the switch logs for errors.

```
# errdumpall
# portperfshow
```

3. Check host and storage logs, and verify that I/O runs without errors.

Test Results

PASS. I/O fails over to an available switch path and recovers when the disrupted switch is restored.

5. Optional/Additional Tests

5.1 Nondisruptive Firmware Upgrade on Storage Device

Test Objective

1. Perform the firmware maintenance procedure on the storage device.

2. Verify that host I/O continues (with minimal disruption) through the “firmware download” and that device pathing remains consistent.

Test Configuration

Test Execution

1. Set up host multipath with links on different switches in the FC fabric, and start I/O.
2. Perform the firmware update on all nodes of the storage array.

Result Validation

1. Check the I/O generator tools logs to verify that I/O completes without errors.
2. Check the host and storage logs for errors throughout the I/O operations.
3. Check the switch logs and port stats for errors or I/O drops.

```
# errdumpall

# porterrshow

0    0    0    0    0    0    0    946    1    0    1    0    0    0    0    0
0    0    0    0    0    0    0     30    1    0    1    0    0    0    0    0
0    0    0    0    0    0    0      0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0      0    0    0    0    0    0    0    0    0
```

Test Results

PASS. I/O completes successfully throughout the firmware upgrade process.

5.2 Workload Simulation Test Suite—Medusa

Test Objective

1. Validate storage/fabric behavior while running a workload simulation test suite.
2. Areas of focus may include random and sequential data patterns of various block sizes and database simulation.

Test Configuration

Test Execution

1. Set up four standalone hosts with two multipathed initiator ports for I/O generation.
2. Use the Medusa I/O tool for generating I/O and simulating workloads.
3. Run random and sequential I/O in a loop at block transfer sizes of 512, 4k, 8k, 16k, 32k, 64k, 128k, 256k, 512k, and 1m. Include a nested loop of 100-percent read, 100-percent write, and 50-percent read/write.
4. Run the Medusa Application I/O workload suite, which includes OLTP, Decision Support System (DSS), Exchange Email, File Servers, Media Streaming, OS Drive, OS Paging, SQL, Video on Demand, VDI, and Web Server profiles.

Result Validation

1. Check the I/O generator tool logs to verify that I/O completes without errors.
2. Check the host and storage logs for errors throughout the I/O operations.

3. Check the switch logs and port stats for errors or I/O drops.

```
# errdumpall

# porterrshow

0     0     0     0     0     0     0     946     1     0     1     0     0     0     0     0
0     0     0     0     0     0     0      30     1     0     1     0     0     0     0     0
0     0     0     0     0     0     0       0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0       0     0     0     0     0     0     0     0     0
```

Test Results

PASS. All workload runs are monitored at the host, storage, and fabric and complete without I/O errors or faults.

5.3 Workload Simulation Test Suite—VMware

Test Objective

1. Validate storage/fabric behavior while running a virtual workload simulation test suite.
2. Areas of focus include VM environments running de-duplication/compression data patterns and database simulation.

Test Configuration

Test Execution

1. Set up an ESX cluster of two hosts with four worker VMs per host.
2. Use the VMware I/O Analyzer tool to generate I/O and simulate workloads.
   a. Run random and sequential I/O at large and small block transfer sizes.
   b. Run the SQL Server simulation workload.
   c. Run the OLTP simulation workload.
   d. Run the Web Server simulation workload.
   e. Run the Video on Demand simulation workload.
   f. Run the Workstation simulation workload.
   g. Run the Exchange Server simulation workload.
3. Set up VMs with the Medusa I/O tool, and run random and sequential I/O in a loop at block transfer sizes of 512, 4k, 8k, 16k, 32k, 64k, 128k, 256k, 512k, and 1m. Include a nested loop of 100-percent read, 100-percent write, and 50-percent read/write.
4. Run the Medusa Application I/O workload suite, which includes OLTP, Decision Support System (DSS), Exchange Email, File Servers, Media Streaming, OS Drive, OS Paging, SQL, Video on Demand, VDI, and Web Server profiles.

Result Validation

1. Check the I/O generator tool logs to verify that I/O completes without errors.
2. Check the host and storage logs for errors throughout the I/O operations.
3. Check the switch logs and port stats for any errors or I/O drops.

```
# errdumpall
```

```
# porterrshow

0     0     0     0     0     0     0     946     1     0     1     0     0     0     0     0
0     0     0     0     0     0     0      30     1     0     1     0     0     0     0     0
0     0     0     0     0     0     0       0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0       0     0     0     0     0     0     0     0     0
```

Test Results

PASS. All workload runs are monitored at the host, storage, and fabric and complete without I/O errors or faults.

Test Conclusions

1. Achieved a 100% pass rate on all test cases in the SSR qualification test plan. The network and the storage were able to handle the various stress and error recovery scenarios without issue.
2. Different I/O workload scenarios were simulated using Medusa and VMware IO Analyzer tools, and sustained performance levels were achieved across all workload types. The Infinidat array and the Brocade FC fabric handled both the low-latency and high-throughput I/O workloads with equal efficiency without I/O errors or packet drops.
3. The results confirm that the Infinidat InfiniBox array interoperates seamlessly with Brocade FC fabrics, and they demonstrate high availability and sustained performance.
4. The Brocade Gen 5 (16-Gb) and Gen 6 (32-Gb) FC switches were able to handle the sustained throughput and latency performance requirements efficiently with fewer ISL trunks. Multiple ISLs to different switches in the fabric should be set up for providing path redundancy through the fabric.
5. We recommend that you enable the Monitoring and Alerting Policy Suite (MAPS) heath monitor on all switches in the FC fabric to report fabric-wide events and traffic performance metrics. The MAPS feature of Fabric Performance Impact monitoring is enabled by default and should be used to detect fabric bottlenecks in the form of timeouts and latency.
6. We recommend that you implement the IO Insight feature with MAPS alerting to closely monitor the critical flows in the fabric at the SCSI level.
7. Utilizing Peer Zoning helps reduce the zone database size and the zoning complexity, while providing the RSCN and hardware resource efficiencies of Single-Initiator Zoning.
8. QoS Zoning should be used to classify host-target traffic into high, medium, or low priority zones to provide traffic prioritization through the FC fabric for the desired host-target pair by allocating more resources to the traffic in the higher-priority zone.
9. Enabling Emulex ExpressLane on a LUN provides prioritized queuing on the HBA for traffic to that LUN and also sets the CS_CTL tag on the frame, which allows the traffic to be prioritized through the FC fabric based on the value of the CS_CTL tag and the corresponding priority level.
10. Host multipath should be configured for optimal availability and performance. Multipath configuration details for the hosts are provided in the host setup section of the “Configure DUT and Test Equipment” section.
11. We recommend that you install the Infinidat Host Power Tools application on all supported hosts. This application allows storage volume provisioning from the host and other recommended host-side configurations for multipathing and performance tuning to be done with ease.