BROCADE
A Broadcom Inc. Company

VALIDATION TEST REPORT

Brocade Fabric Technology with the Violin Memory FSP 7300 Flash Array

Supporting Fabric OS 8.0.1, 8.2.0, and 8.2.1

53-1005034-03
12 November 2018

Contents

Preface

Document History

Date	Part Number	Description
June 12, 2017	53-1005034-01	Initial release supporting FOS 8.0.1.
March 16, 2018	53-1005034-02	Supporting FOS 8.2.0.
November 12, 2018	53-1005034-03	Supporting FOS 8.2.1.

Overview

The Storage Fabric Ready (SFR) program is a comprehensive testing and configuration initiative to provide Fibre Channel SAN interoperability with flash storage. This program provides testing of multiple fabrics, heterogeneous servers, and HBAs in large-port-count Brocade environments. The SFR qualification program helps verify seamless interoperability and optimum performance with solid-state storage, software-defined storage, and hyper-converged systems in Brocade storage fabrics.

Purpose of This Document

This document provides the validation of Brocade fabric technology with the Violin FSP 7300 all-flash storage array, using multiple switch platforms, HBAs, and server operating systems. This validation shows that the Violin FSP 7300 interoperates properly within a Brocade Fibre Channel fabric, while supporting the performance and low latency associated with solid-state storage.

Audience

This document is written for a technical audience, including solution architects, system engineers, and technical development representatives.

Objectives

- Test the Violin FSP 7300 array with Brocade FC fabrics, in single and routed configurations for different stress and error recovery scenarios to validate the interoperability and integration of the array with Brocade FC fabrics.
- Validate the performance of the 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 Flow Vision Configuration Guide*
- *Brocade Fabric OS Command Reference Manual*
- *Brocade SAN Design and Best Practices*
- *Brocade SAN Fabric Resiliency and Administration Best Practices Guide*
- Emulex ExpressLane Configuration
- *Emulex OneCommand Manager User Manual*
- *QLogic QLE2672 Adapter User's Guide*
- *QLogic BR-1860 Adapter Administrator's Guide*

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 Violin Memory

Business in a Flash. Violin Memory transforms the speed of business with high-performance, always available, low-cost management of critical business information and applications.

Violin's All-Flash optimized solutions accelerate breakthrough CAPEX and OPEX savings for building the next generation data center. Violin's Flash Fabric Architecture (FFA) speeds data delivery with chip-to-chassis performance optimization that achieves lower consistent latency and cost per transaction for cloud, enterprise, and virtualized mission-critical applications. Violin's all-flash arrays and appliances and enterprise data management software solutions enhance agility and mobility while revolutionizing data center economics.

Founded in 2005, Violin Memory is headquartered in Santa Clara, California.

Configure DUT and Test Equipment

Task 1. Brocade FC Fabric Configuration

1. Configure MAPS on the switches (requires a Fabric Vision license). This will enable reporting of latency and congestion alerts on each switch, as well as a number of switch and fabric health metrics. For more information on configuring and using MAPS, see the *Brocade Monitoring and Alerting Policy Suite Configuration Guide*.

   a) Enable the desired MAPS policy using any of the available default policies, or create a custom policy. For this test, the default Aggressive Policy is used. This provides the most sensitive threshold levels for detection of latency and congestion.

   ```
   > mapsconfig --enablemaps -policy dflt_aggressive_policy
   ```

   b) Define the reporting actions that MAPS will take. In this test, errors discovered by MAPS will generate a RASLog entry and send email to the configured recipients.

   ```
   >  mapsconfig --actions raslog,email
   ```

   The Fabric Performance Impact (FPI) category within MAPS monitors the current condition of the latency seen on F_Ports over different time windows, and it uses that information to determine the performance impact to the fabric and network.

   c) To examine the effective MAPS configuration, use the following command:

   ```
   > mapsconfig –show
   Configured Notifications:       RASLOG,EMAIL
   Mail Recipient:                 testuser1@domain.com,testuser2@domain.com
   FPI Monitoring:                 Enabled
   Paused members :
   ===============
   PORT :
   CIRCUIT :
   SFP :
   ```

   d) Use the following command to view the summary of events or rules triggered and the objects on which the rules were triggered over a specified period of time.

   ```
   > mapsdb –show
   1 Dashboard Information:
   ========================
   DB start time:                  Wed Dec 17 20:59:09 2014
   Active policy:                  dflt_aggressive_policy
   Configured Notifications:       RASLOG,EMAIL
   Fenced Ports :                  None
   Decommissioned Ports :          None
   ```

2. Configure flow monitoring with the IO Insight feature on the Brocade G620 Switch in the fabric.

   The IO Insight feature supported on the Gen 6 hardware 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.

   NOTE
   Requires Fabric Vision and IO Insight licenses.

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

```
root> flow --create ios_violin_1 -fea mon -dstdev df1400  -egrport 20
Monitor feature(s) have been activated.
root> flow --create ios_violin_2 -fea mon -dstdev df0400 -egrport 4
Monitor feature(s) have been activated.
root> flow --show ios_violin_1
================================================================================
Name      : ios_violin_1  Features: mon(Activated)        noConfig: Off
Definition: EgrPort(20),DstDev(0xdf1400)
```

   b) Import the created flows into MAPS.

```
root> mapsconfig --import ios_violin_1
root> mapsconfig --import ios_violin_2
root> logicalgroup -show
-------------------------------------------------------------------------------------------
Group Name                      |Predefined |Type              |Member Count |Members
-------------------------------------------------------------------------------------------
ios_violin_1                    |No         |Flow              |1            |Monitored Flow
ios_violin_2                    |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.

```
root> mapspolicy --clone dflt_moderate_policy -name ios_mod_policy

root> mapsrule --create ios_violin_4k_rd_status -group ios_violin_1 -monitor
RD_STATUS_TIME_LT_8K -timebase min -op ge -value 1 -action email,raslog -policy ios_mod_policy
root> mapspolicy --show ios_mod_policy
Policy Name: ios_mod_policy
Rule Name                  |Condition                                  |Actions      |
ios_violin_4k_rd_status     |ios_violin_1(RD_STATUS_TIME_LT_8K/min>=1)  |email,raslog |
```

   d) Enable the MAPS policy.

```
root> mapspolicy --enable ios_mod_policy
root>
root> mapsdb --show

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

DB start time:                  Thu May 19 09:27:24 2016
Active policy:                  ios_mod_policy
Configured Notifications:       RASLOG,EMAIL,SW_CRITICAL,SW_MARGINAL
Fenced Ports :                  None
Decommissioned Ports :          None
Fenced circuits :               N/A
Quarantined Ports :             None
Top Zoned PIDs <pid(it-flows)>: 0xdf2000(30) 0xdf2100(30) 0xdf0400(28) 0xdf1400(28) 0xdf1b00(22)
```

3. Configure Fibre Channel Routing (requires an Integrated Routing license). Detailed information on FCR setup can be found in the *Brocade Fabric OS Administration Guide*.

   An example FCR configuration is shown below:

   The prefix 'lsan' is used when configuring zones for use in Fibre Channel Routing. The following is an example of a zone prefixed with 'lsan':

```
> zoneshow lsan_hb067166_violin
zone:  lsan_hb067166_violin
               10:00:8c:7c:ff:23:b7:00; 10:00:8c:7c:ff:23:b7:01;
               52:4a:93:7d:f3:5f:61:00; 52:4a:93:7d:f3:5f:61:01
               52:4a:93:7d:f3:5f:61:10; 52:4a:93:7d:f3:5f:61:11
```

   a) 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
```

4. Zoning is configured per host, and the array ports are placed in a zone alias for convenience. The array target ports to use for zoning are the NPIV ports presented by the storage and not the physical port WWPNs.

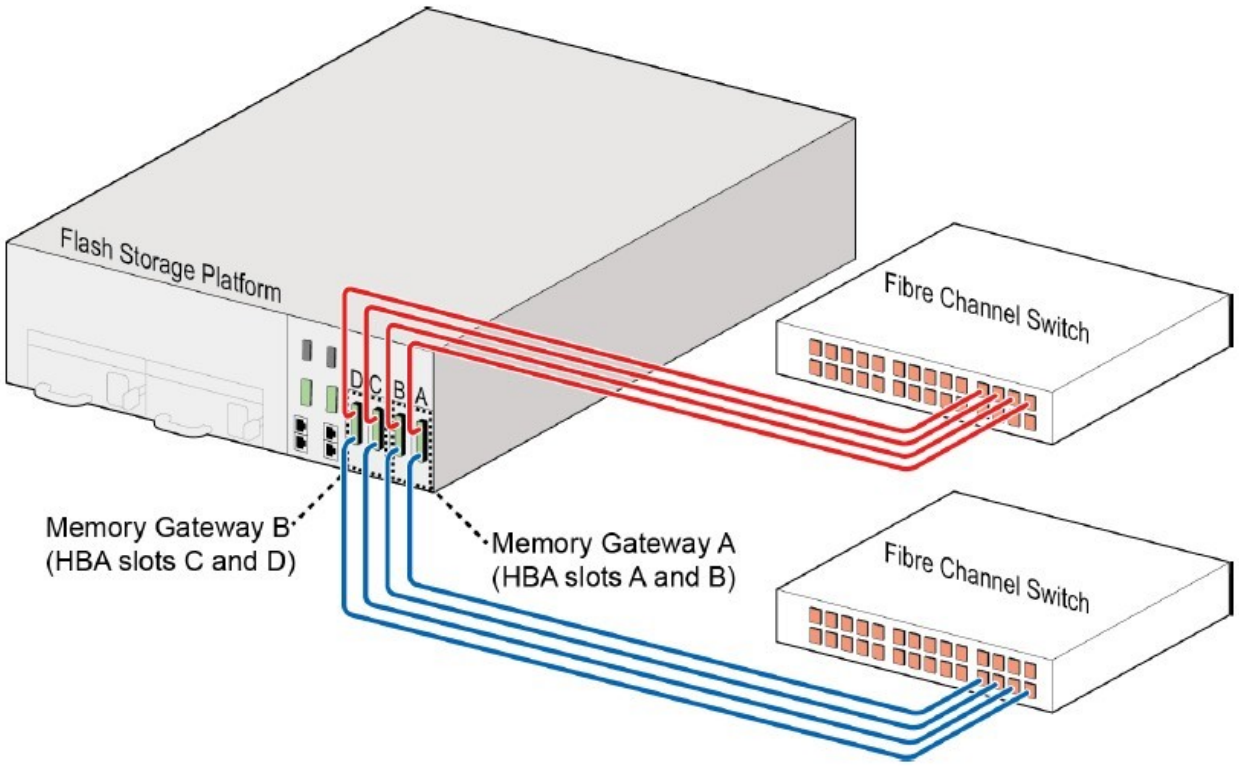
```
> alicreate violin_ali, "21:00:00:1b:97:28:d9:61; 21:00:00:1b:97:28:d9:60; 21:00:00:1b:97:28:d8:41;
21:00:00:1b:97:28:d8:40; 21:00:00:1b:97:28:d9:11; 21:00:00:1b:97:28:d9:10; 21:00:00:1b:97:28:da:f1;
21:00:00:1b:97:28:da:f0"
> alishow violin_alialias: violin_ali21:00:00:1b:97:28:d9:61;
21:00:00:1b:97:28:d9:60;21:00:00:1b:97:28:d8:41; 21:00:00:1b:97:28:d8:40;21:00:00:1b:97:28:d9:11;
21:00:00:1b:97:28:d9:10;21:00:00:1b:97:28:da:f1; 21:00:00:1b:97:28:da:f0
> zonecreate ssd067172_violin, "10:00:8c:7c:ff:1f:7b:00; 10:00:8c:7c:ff:1f:7b:01; violin_ali"
> zoneshow ssd067172_violinzone:ssd067172_violin10:00:8c:7c:ff:1f:7b:00; 10:00:8c:7c:ff:1f:7b:01;
violin_ali
```

Task 2. Violin Memory Array Configuration

Configuration steps for the Violin Memory array are covered here.

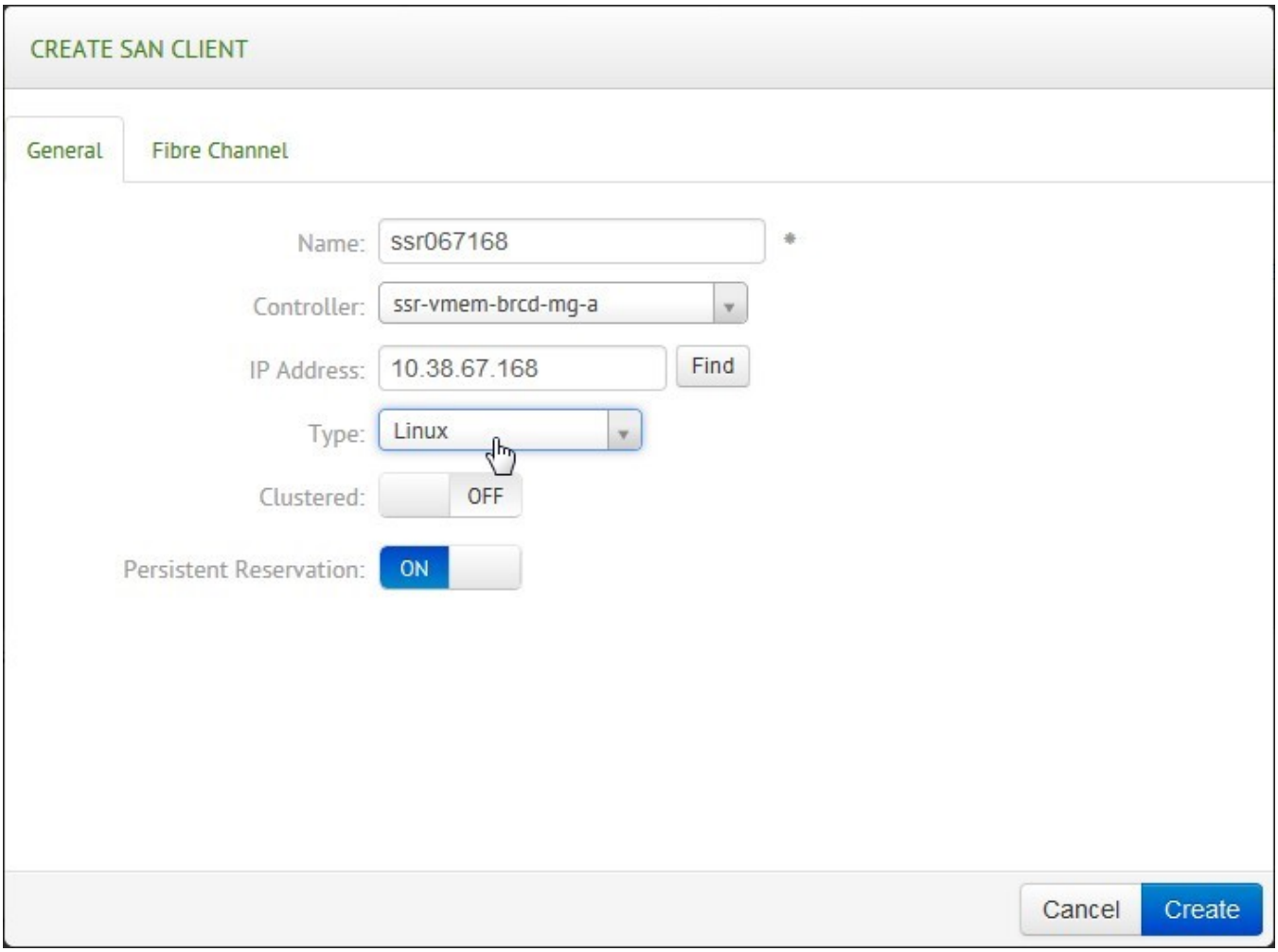
1. Provision eight array target ports so that the two controllers are distributed across two switches, as detailed in the following diagram. This provides the best configuration for high availability. See the *7300 Flash Storage Platform Installation Guide* for complete details.

FIGURE 1 Recommended Cabling for High-Availability Configuration



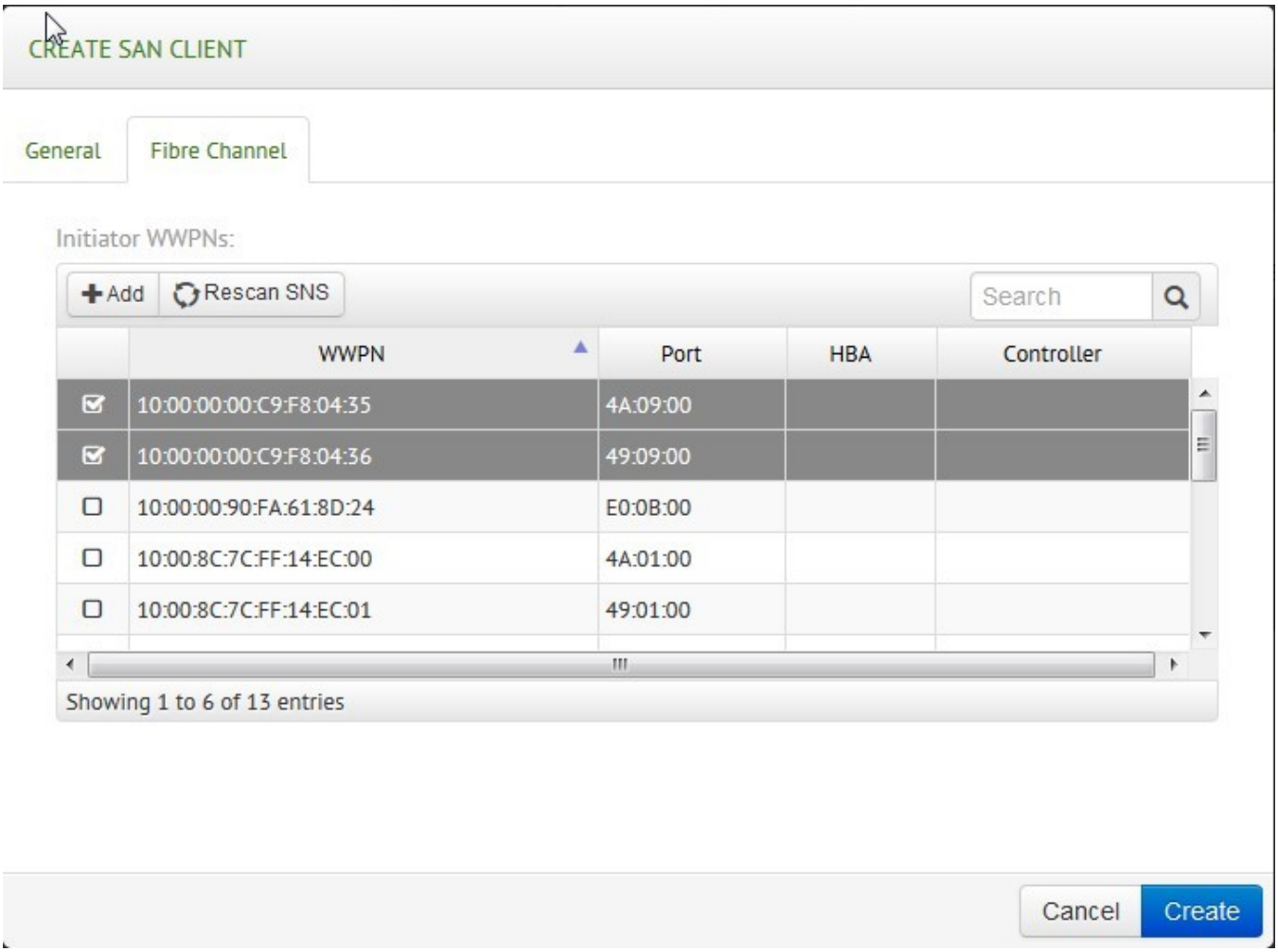
2. Create a new SAN client:

FIGURE 2 SAN Client Creation in the Symphony Web Interface



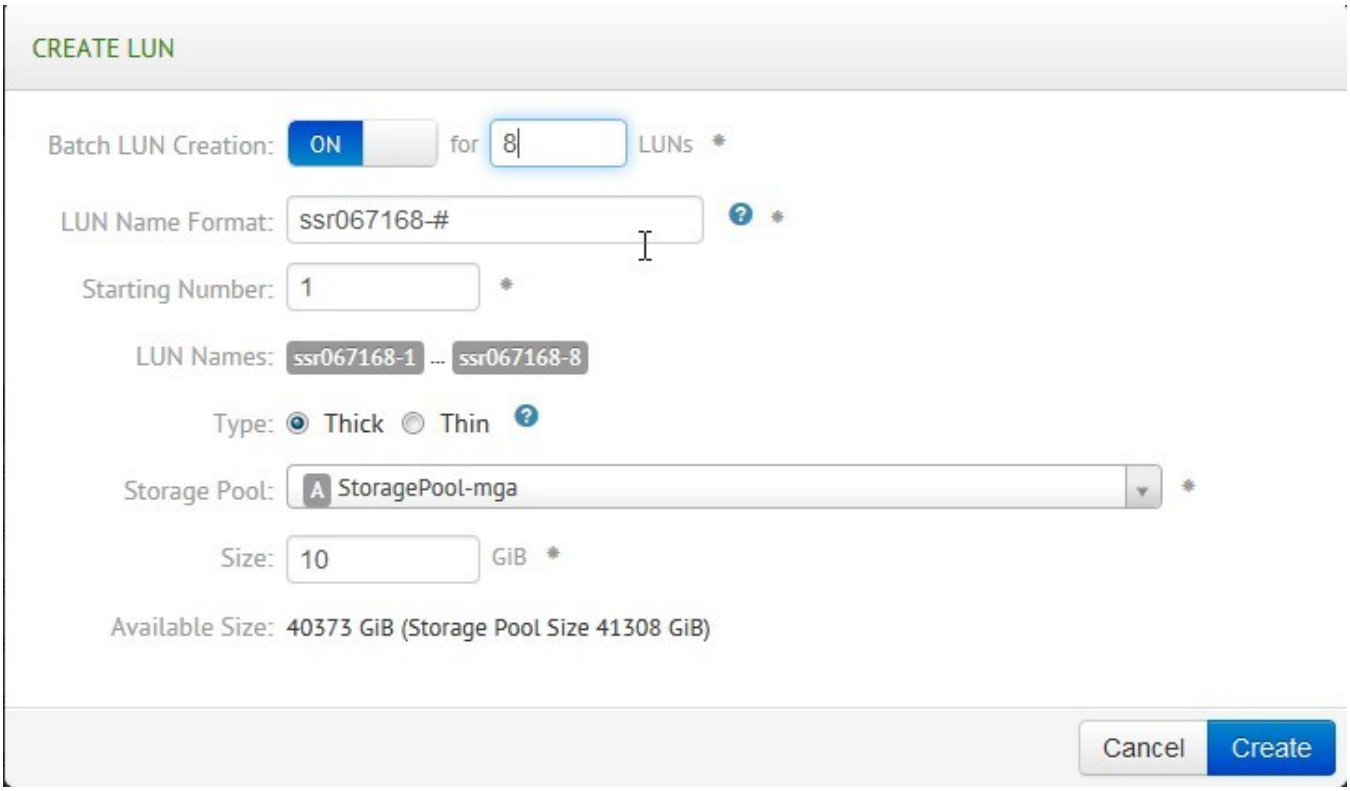
3. Assign host WWPNs to the new SAN client:

FIGURE 3 Adding WWPNs to the SAN Client in the Symphony Web Interface



4. Create new LUNs. For this test configuration, 8 x 10GB LUNs are created:

FIGURE 4 Creating New LUNs in the Symphony Web Interface



5. Assign LUNs to the SAN client:

FIGURE 5 Assigning LUNs to the SAN Client in the Symphony Web Interface

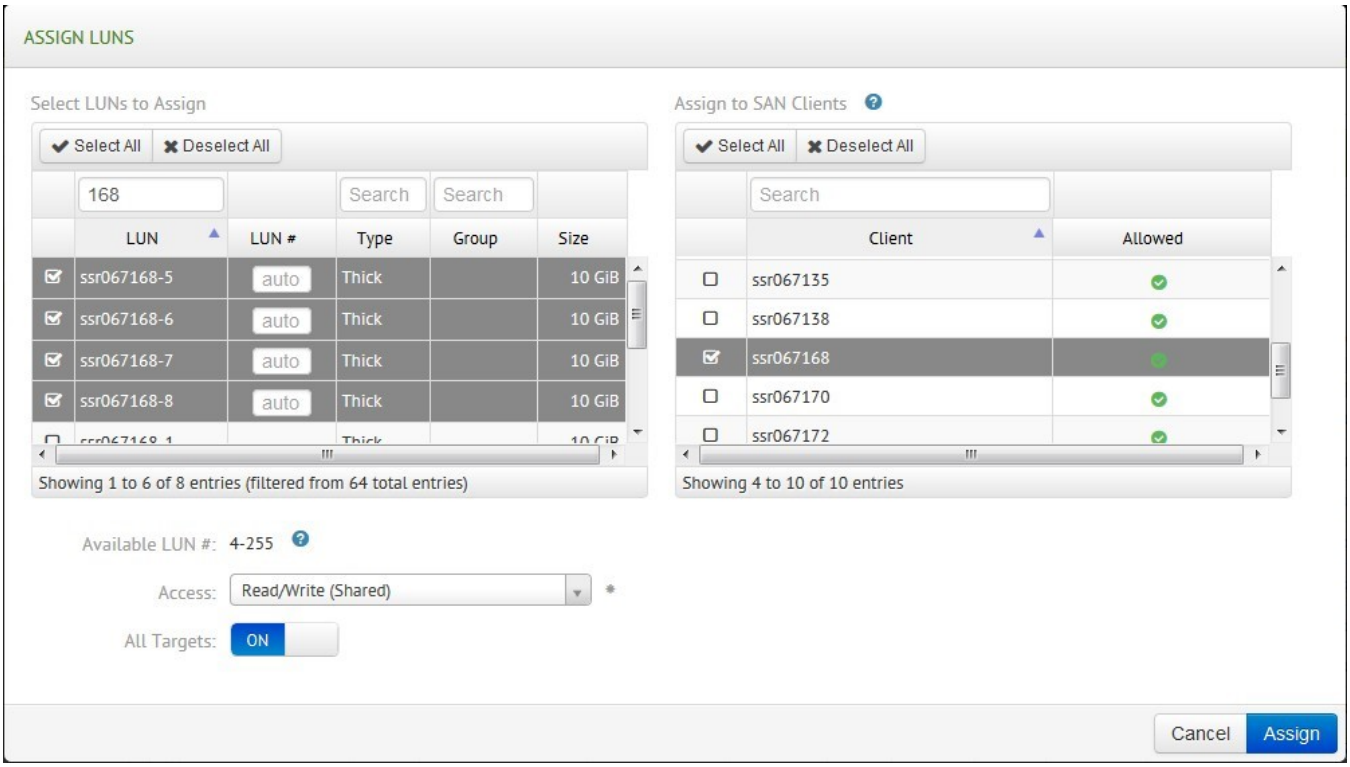
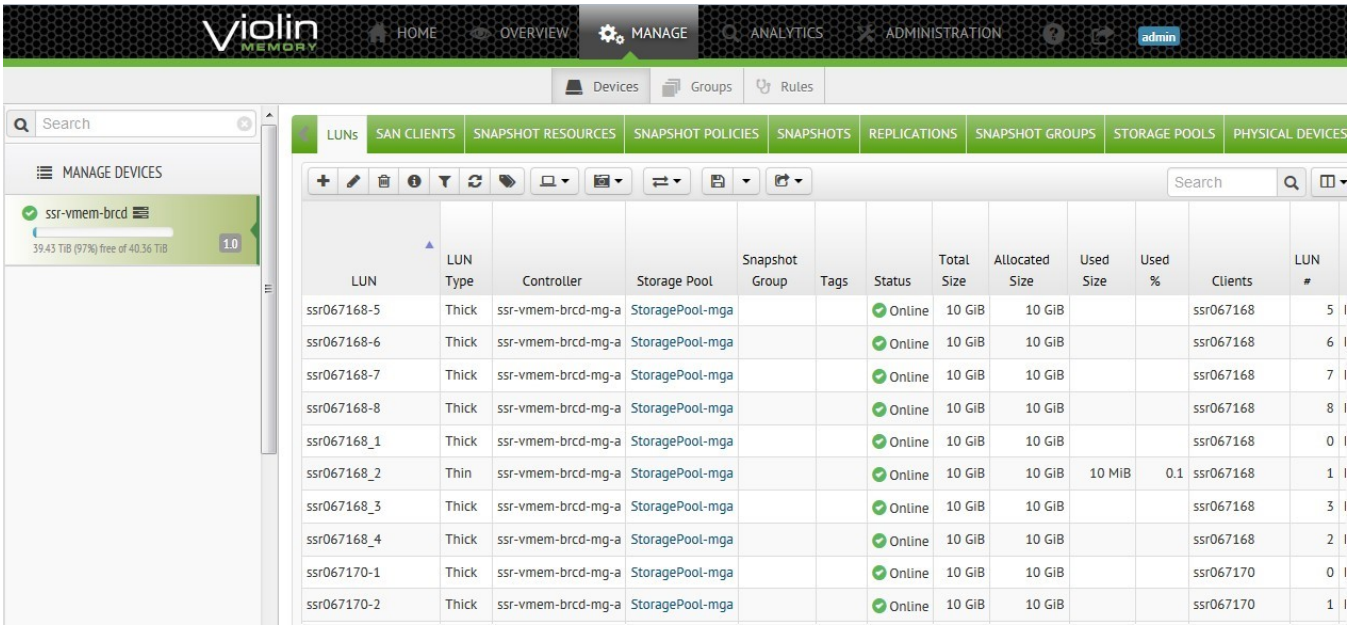


FIGURE 6 Completed LUN Presentation in the Symphony Web Interface



Task 3. Host Setup

Configuration settings of servers in the test bed are covered here.

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. Multipath configuration on Linux hosts.

   An example /etc/multipath.conf configuration on an RHEL 7.2 host is shown below. For more information. see the *Violin Memory Interoperability Best Practices Guide*.

```
device {
   vendor "VIOLIN"
   product "CONCERTO ARRAY"
   path_selector "round-robin 0"
   path_grouping_policy multibus
   prio aluapath_checker tur
   rr_min_io 100
   rr_weight priorities
   failback immediate
   features "1 queue_if_no_path"no_path_retry 300
}
```

   Example multipath configuration on Linux:

```
# multipath -ll
mpathbp (36001b970b716b179b716b1795996a912) dm-13 VIOLIN  ,CONCERTO ARRAY
size=10.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:5:3  sdbp 68:48  active ready running
  |- 13:0:6:3  sdbx 68:176 active ready running
  |- 14:0:5:3  sdcj 69:112 active ready running
  |- 13:0:4:3  sdcg 69:64  active ready running
  |- 13:0:7:3  sdcw 70:64  active ready running
  |- 14:0:6:3  sdcz 70:112 active ready running
  |- 14:0:7:3  sddl 71:48  active ready running
  `- 14:0:4:3  sddt 71:176 active ready running
mpathbo (36001b970b716b179b716b1794df3d665) dm-15 VIOLIN  ,CONCERTO ARRAY
size=10.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:5:8  sdbu 68:128 active ready running
  |- 13:0:6:8  sdcc 69:0   active ready running
  |- 14:0:5:8  sdcs 70:0   active ready running
  |- 13:0:4:8  sdcp 69:208 active ready running
  |- 14:0:6:8  sddi 71:0   active ready running
  |- 13:0:7:8  sddh 70:240 active ready running
  |- 14:0:7:8  sddq 71:128 active ready running
  `- 14:0:4:8  sddy 128:0  active ready running
mpathbn (36001b970b716b179b716b179453c26a3) dm-12 VIOLIN  ,CONCERTO ARRAY
size=10.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:5:1  sdbn 68:16  active ready running
  |- 13:0:6:1  sdbv 68:144 active ready running
  |- 13:0:4:1  sdcd 69:16  active ready running
  |- 14:0:5:1  sdcf 69:48  active ready running
  |- 14:0:6:1  sdcv 70:48  active ready running
  |- 13:0:7:1  sdct 70:16  active ready running
  |- 14:0:7:1  sddj 71:16  active ready running
  `- 14:0:4:1  sddr 71:144 active ready running
mpathbm (36001b970b716b179b716b17975c58392) dm-11 VIOLIN  ,CONCERTO ARRAY
size=10.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:5:4  sdbq 68:64  active ready running
  |- 13:0:6:4  sdby 68:192 active ready running
  |- 14:0:5:4  sdcl 69:144 active ready running
  |- 13:0:4:4  sdci 69:96  active ready running
  |- 14:0:6:4  sddb 70:144 active ready running
  |- 13:0:7:4  sdcy 70:96  active ready running
  |- 14:0:7:4  sddm 71:64  active ready running
  `- 14:0:4:4  sddu 71:192 active ready running
mpathbl (36001b970b716b179b716b1798ebafd67) dm-18 VIOLIN  ,CONCERTO ARRAY
size=10.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:5:7  sdbt 68:112 active ready running
  |- 13:0:6:7  sdcb 68:240 active ready running
  |- 13:0:4:7  sdcn 69:176 active ready running
```

```
  |- 14:0:5:7  sdcr 69:240 active ready running
  |- 14:0:6:7  sddg 70:224 active ready running
  |- 13:0:7:7  sdde 70:192 active ready running
  |- 14:0:7:7  sddp 71:112 active ready running
  `- 14:0:4:7  sddx 71:240 active ready running
mpathbs (36001b970b716b179b716b17909052d48) dm-16 VIOLIN  ,CONCERTO ARRAY
size=10.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:5:5  sdbr 68:80  active ready running
  |- 13:0:6:5  sdbz 68:208 active ready running
  |- 13:0:4:5  sdck 69:128 active ready running
  |- 14:0:5:5  sdco 69:192 active ready running
  |- 14:0:6:5  sddd 70:176 active ready running
  |- 13:0:7:5  sdda 70:128 active ready running
  |- 14:0:7:5  sddn 71:80  active ready running
  `- 14:0:4:5  sddv 71:208 active ready running
mpathbr (36001b970b716b179b716b179c367a45c) dm-17 VIOLIN  ,CONCERTO ARRAY
size=10.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:5:6  sdbs 68:96  active ready running
  |- 13:0:6:6  sdca 68:224 active ready running
  |- 14:0:5:6  sdcq 69:224 active ready running
  |- 13:0:4:6  sdcm 69:160 active ready running
  |- 13:0:7:6  sddc 70:160 active ready running
  |- 14:0:6:6  sddf 70:208 active ready running
  |- 14:0:7:6  sddo 71:96  active ready running
  `- 14:0:4:6  sddw 71:224 active ready running
mpathbq (36001b970b716b179b716b179072e9a2a) dm-14 VIOLIN  ,CONCERTO ARRAY
size=10.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:5:2  sdbo 68:32  active ready running
  |- 13:0:6:2  sdbw 68:160 active ready running
  |- 13:0:4:2  sdce 69:32  active ready running
  |- 14:0:5:2  sdch 69:80  active ready running
  |- 14:0:6:2  sdcx 70:80  active ready running
  |- 13:0:7:2  sdcu 70:32  active ready running
  |- 14:0:7:2  sddk 71:32  active ready running
  `- 14:0:4:2  sdds 71:160 active ready running
```

3. Configure multipathing on Windows hosts.

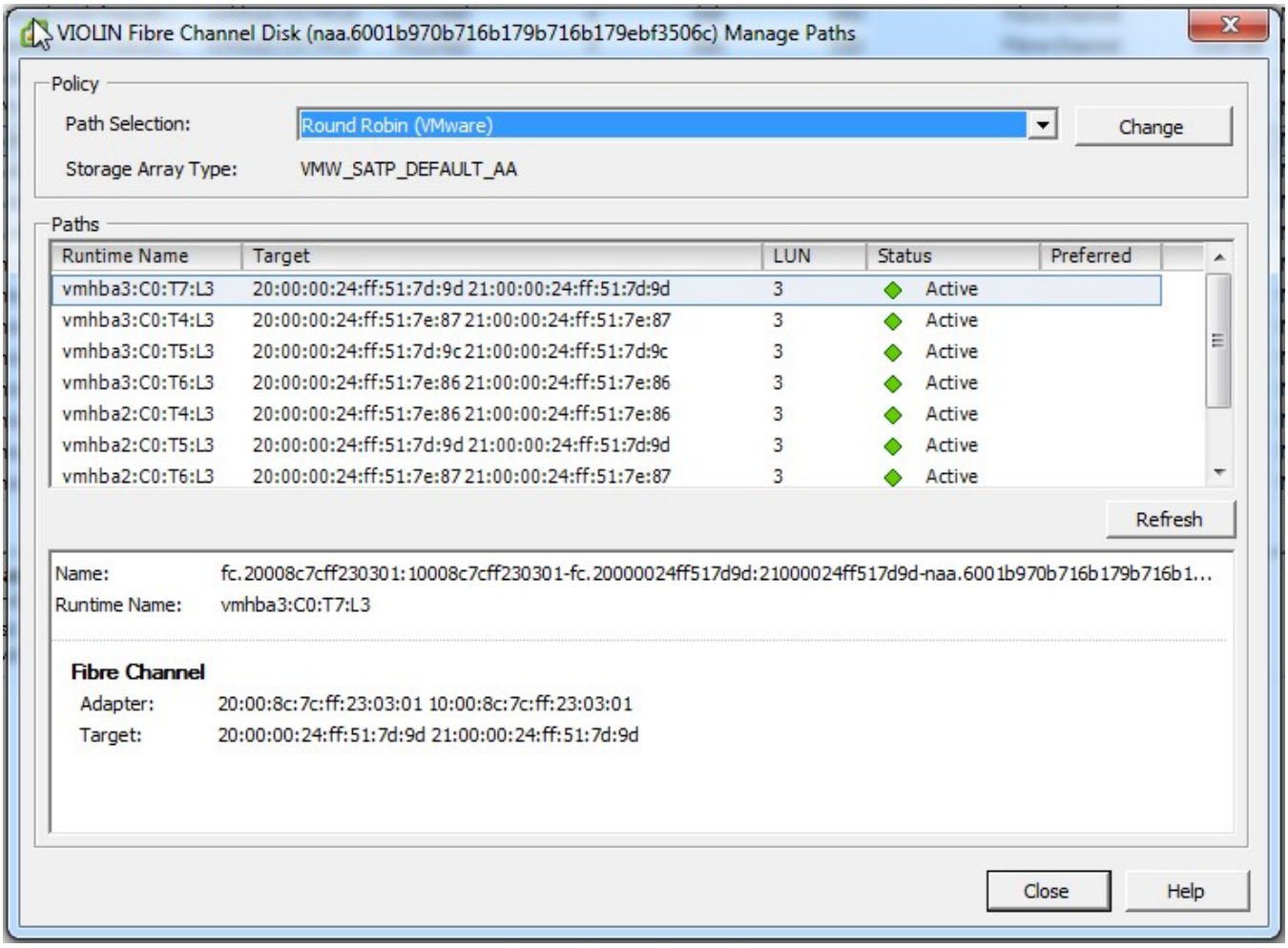
   Use the Windows **MPIO Properties** dialog to discover and manage multipath entries for the Violin SAN array.

   FIGURE 7 Windows MPIO Multipathing with the Violin Array

4. Multipath configuration on VMware hosts.

   Change the path selection to Round Robin. This provides superior performance to the Most Recently Used setting, which uses only a single active path per LUN.

   FIGURE 8 Round Robin Multipath Configuration on VMware



5. Apply any additional host tuning (for Linux systems). Settings shown below help to support high IOPS operation of the Violin Memory flash array with Linux hosts.

   Create a file named "/etc/udev/rules.d/99-violin-storage.rules" and add the following lines:

```
# Use noop scheduler for high-performance solid-state storage
ACTION=="add|change", KERNEL=="sd*[!0-9]", SUBSYSTEM=="block", nimbENV{ID_VENDOR}=="VIOLIN",
ATTR{queue/scheduler}="noop"
# Reduce CPU overhead due to entropy collection
ACTION=="add|change", KERNEL=="sd*[!0-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="VIOLIN", ATTR{queue/
add_random}="0"
# Spread CPU load by redirecting completions to originating CPU
ACTION=="add|change", KERNEL=="sd*[!0-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="VIOLIN", ATTR{queue/
rq_affinity}="2"
```

6. Set up workload generators.

   On Windows and Linux systems, Medusa Labs Test Tools is installed. On VMware systems, VMware's IO Analyzer is installed.

Test Report

What's New in This Report

- The Brocade Fabric OS (FOS) version under test is 8.2.1.
- Emulex and QLogic adapters have updated firmware and drivers.

For detailed information, see the tables in the "DUT Descriptions" section.

Test History

Storage Model	Storage Firmware	Brocade FOS Version	Date
Violin FSP 7300	A7.1.2.1 / 7.6.1	FOS 8.0.1	June 2017
Violin FSP 7300	A7.1.2.1 / 7.6.1.3	FOS 8.2.0	January 2018
Violin FSP 7300	A7.1.2.1 / 7.6.1.3	FOS 8.2.1	November 2018

Test Plan Overview

The storage array is connected to two SAN fabrics and multiple server hosts to drive I/O in a multipath configuration. Error injection is introduced, and failover and recovery behaviors are observed. I/O performance is observed across different workload configurations.

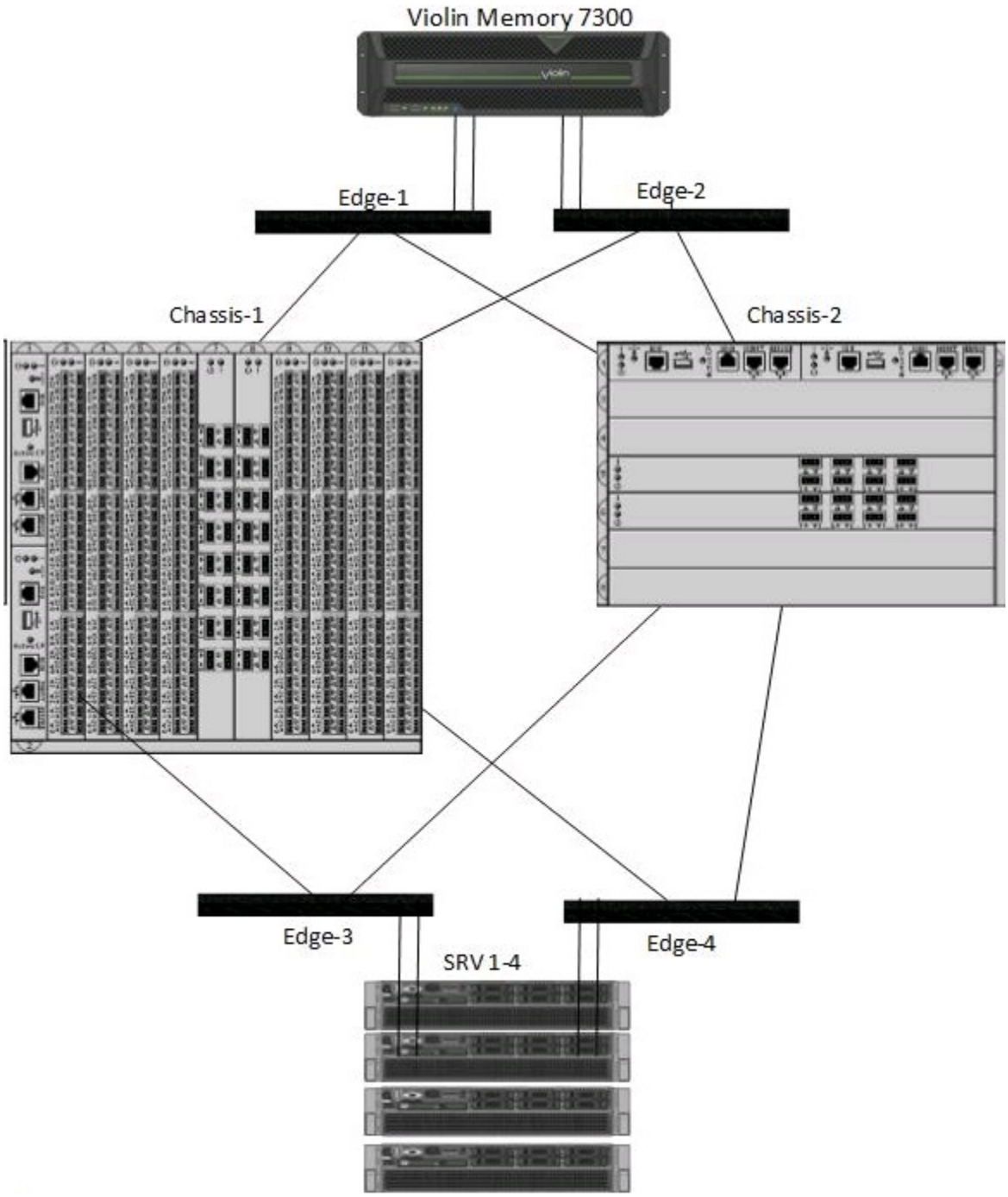
Scope

Testing is performed with a mix of GA and development versions of Brocade's Fabric OS (FOS) in a heterogeneous environment. Test beds include Brocade directors and switches configured in routed and non-routed fabric configurations.

Testing is centered on interoperability and optimal configuration. 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 the "Test Cases" section. Standard test-bed setup includes IBM/HP/Dell chassis server hosts with Brocade/QLogic/Emulex HBAs with two uplinks from every host to a Brocade FC fabric. I/O generators included Medusa Labs Test Tools and VMware IO 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
Violin FSP 7300	Violin Flash Storage Platform 7300	Violin Memory	The Violin Memory Flash Storage Platform 7300 under test has 64 x 1T MLC Flash VIMMs (Violin Intelligent Memory Modules). All eight 16-Gb FC connections are connected to the fabric in a high-availability configuration.

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

1.0	**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
2.0	**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
3.0	**STRESS AND ERROR RECOVERY WITH DEVICE MULTIPATH**
3.1	Storage Device Fabric I/O Integrity—Congested Fabric
3.2	Storage Device Name-Server Integrity—Device Recovery with Port Toggle
3.3	Storage Device Name-Server Integrity—Device Recovery with Device Relocation
3.4	Storage Device Name-Server Stress—Device Recovery with Device Port Toggle
3.5	Storage Device Recovery—ISL Port Toggle (Sequential)
3.6	Storage Device Recovery—ISL Port Toggle (Entire Switch)
3.7	Storage Device Recovery—Switch Offline
3.8	Storage Device Recovery—Switch Firmware Download
4.0	**STORAGE DEVICE—FIBRE CHANNEL ROUTING (FCR) INTERNETWORKING TESTS**
4.1	Storage Device Internetworking Validation with FC Host
4.2	Storage Device Edge Recovery After FCR Disruptions
4.3	Storage Device Backbone Recovery After FCR Disruptions
5.0	**OPTIONAL/ADDITIONAL TESTS**
5.1	Non-disruptive Firmware Upgrade on the Array
5.2	Synthetic I/O Workload Loop with Varying Block Sizes
5.3	Storage Device Failover Tests

1.0 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 Execution

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

Result Validation

1. Validate link states on the array, and verify speed negotiation and device login at different speeds. Use the **portshow** command on the switch to check the link state and speed, and the **nscamshow** command to verify device login to the fabric.
2. Check the switch port status, and verify the "actual" and "configured" link speed. Check the name server for device login.

```
root> portshow 19
portIndex:  19
portName: port19
portHealth: HEALTHY

portState: 1    Online
Protocol: FC
portWwn of device(s) connected:
        10:00:8c:7c:ff:22:f7:81
Distance:  normal
portSpeed: N16Gbps
```

Test Results

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

1.2 Zoning and LUN Mapping

Test Objective

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

Test Execution

1. Create an FC zone on the fabric with the initiator and target WWNs.
2. Create Host Groups and LUNs on the array with access to initiator WWN.

Result Validation

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

- Linux: Check the output of **lsscsi**
- Windows: Check the output of **Computer Management > Storage > Disk Management**
- VMware: Check the 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 write/read/verify testing.

Test Execution

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

Result Validation

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

Test Results

PASS. I/O resumes without error.

1.4 Storage Device Multipath Configuration—Path Integrity

Test Objective

Verify that multipath configures successfully, and verify the integrity of each path.

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.
3. Start I/O.
4. Isolate paths individually, and run I/O on the separate paths.

Result Validation

1. Check the 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.
4. Check the I/O logs, and verify that I/O continues without error.

Test Results

PASS. Each individual path is valid, and the paths combine successfully in a multipath configuration.

2.0 Fabric—Advanced Functionality

2.1 Bottleneck Detection Using MAPS FPI—With Congested Host

Test Objective

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

Test Execution

1. Configure MAPS and FPI on all switches. (See the "Task 1: Brocade FC Fabric Configuration" section.)
2. Start I/O from a single host initiator to multiple targets.
3. Monitor switch logs for IO_PERF_IMPACT/ IO_FRAME_LOSS warnings.

Result Validation

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

```
root> errdumpall | grep IO_
2014/12/17-11:56:00:672622, [MAPS-1003], 117148/115018, FID 128, WARNING, B6510_066_088, Port 16,
Condition=ALL_F_PORTS(DEV_LATENCY_IMPACT==IO_PERF_IMPACT), Current Value:
[DEV_LATENCY_IMPACT,IO_PERF_IMPACT, 30.0% in 10 secs], RuleName=defALL_F_PORTS_IO_PERF_IMPACT, Dashboard
Category=Fabric Performance Impact., actionHndlr.c, line: 755, comp:md, ltime:2014/12/17-11:56:00:671909

root> mapsdb --show

1 Dashboard Information:
========================
Active policy:                  dflt_aggressive_policy
……….

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

3.1 Summary Report:
```

```
==================
Category                |Today                    |Last 7 days              |
-------------------------------------------------------------------------------
Port Health             |No Errors                |Out of operating range   |
Fru Health              |In operating range       |In operating range       |
Security Violations     |No Errors                |In operating range       |
Fabric State Changes    |No Errors                |In operating range       |
Switch Resource         |In operating range       |In operating range       |
Traffic Performance     |In operating range       |In operating range       |
FCIP Health             |Not applicable           |Not applicable           |
Fabric Performance Impact|Out of operating range  |Out of operating range   |
```

Test Results

PASS. With MAPS reporting configured, performance warnings are reported as expected. The bottlenecked ports are displayed on the MAPS dashboard, and a RASLog warning is also created.

2.2 Bottleneck Detection Using MAPS FPI—With Congested Fabric

Test Objective

Validate bottleneck detection with congested fabric.

Test Execution

1. Configure MAPS on all switches. (More info in "Task 1. Brocade FC Fabric Configuration" section.)
2. Isolate a single ISL in the fabric.
3. Start I/O from multiple host initiators to multiple targets.
4. Monitor switch logs for Traffic Performance warnings.

Result Validation

Check switch error logs and MAPS dashboard for bottleneck warnings.

```
2014/12/18-08:39:03:497512, [MAPS-1003], 9637/5496, FID 128, WARNING, B6510_066_082, Port 33,
Condition=ALL_TARGET_PORTS(RX/hour>60.00), Current Value:[RX,65.16 %], RuleName=defALL_TARGET_PORTSRX_60,
Dashboard Category=Traffic Performance., actionHndlr.c, line: 755, comp:md, ltime:2014/12/18-08:39:03:496754

root> mapsdb --show

1 Dashboard Information:
========================
Active policy:                  dflt_aggressive_policy
……….

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

3.1 Summary Report:
===================
Category                |Today                    |Last 7 days              |
-------------------------------------------------------------------------------
Port Health             |No Errors                |Out of operating range   |
Fru Health              |In operating range       |In operating range       |
Security Violations     |No Errors                |In operating range       |
Fabric State Changes    |No Errors                |In operating range       |
Switch Resource         |In operating range       |In operating range       |
Traffic Performance     |In operating range       |In operating range       |
```

```
    FCIP Health               |Not applicable           |Not applicable           |
    Fabric Performance Impact |Out of operating range   |Out of operating range   |
```

Test Results

PASS. With MAPS reporting configured, performance warnings are reported as expected.

2.3 Flow Monitoring with IO Insight and MAPS

Test Objective

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

Test Execution

1. Baseline a target LUN's latency running 4k reads from a workload generator. Here we are seeing about a 2- to 3-ms range.

```
Avg Completion Time   0.002531
```

2. Configure a flow to monitor the array target port traffic.

```
root> flow --create ios_violin_1 -fea mon -dstdev df1400  -egrport 20
Monitor feature(s) have been activated.
root> flow --create ios_violin_2 -fea mon -dstdev df0400 -egrport 4
Monitor feature(s) have been activated.
Import the flow into MAPS.
root>
root>
root> mapsconfig --import ios_violin_1
root> mapsconfig --import ios_violin_2

--------------------------------------------------------------------------------
Group Name                      |Predefined |Type              |Member Count |Members
--------------------------------------------------------------------------------
```

3. Create a custom policy, and add MAPS rules for the I/O statistics to be monitored. The threshold is set at a higher level than the baseline. The units for the command are in microseconds, so in this example, we set the threshold to 1 ms.

```
root> mapspolicy --clone dflt_moderate_policy -name ios_mod_policy
root> mapsrule --create ios_violin_4k_rd_status -group ios_violin_2 -monitor RD_STATUS_TIME_LT_8K -timebase min
-op ge -value 1 -action email,raslog -policy ios_mod_policy
Confirm the flow is active

root> flow --show ios_violin_2
================================================================================
Name      : ios_violin_2  Features: mon(Activated)        noConfig: Off
Definition: EgrPort(4),DstDev(0xdf0400)
```

4. 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

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

```
root> flow --show ios_violin_2
================================================================================================================
```

```
Name      : ios_violin_2  Features: mon(Activated)        noConfig: Off
Definition: EgrPort(4),DstDev(0xdf0400)


Flow Monitor (Activated):
Monitor time:  | Thu May 19 07:02:29 MDT 2016 |
-----------------------------------------------------
```

Tx Frames Count	Tx Frames per Sec.	Tx Bytes Count	Tx Throughput(Bps)	Avg Tx Frm Sz(Bytes)
4.88M	171.52k	8.94G	321.18M	1964

I/O Performance:

Metric	IO Size	I/O Count	Max(IOPS)	Avg(IOPS)	
		All	All	6 sec	/ All
RD IO Count	<8K	0	0	0	/ 0
	8K - <64K	0	0	0	/ 0
	64K - <512K	133.45k	5.19k	5.19k	/ 4.45k
	>=512K	0	0	0	/ 0
	ALL	133.45k	5.19k	5.19k	/ 4.45k
WR IO Count	<8K	0	0	0	/ 0
	8K - <64K	0	0	0	/ 0
	64K - <512K	131.06k	5.02k	4.98k	/ 4.37k
	>=512K	0	0	0	/ 0
	ALL	131.06k	5.02k	4.98k	/ 4.37k

I/O Latency Metrics:

Metric	IO Size	Max		AVG	
		6 sec	/ All	6 sec	/ All
RD CMD -> Status Time	<8K		/		/
	8K - <64K		/		/
	64K - <512K	5.07m	/ 6.07m	382u	/ 384u
	>=512K		/		/
	ALL	5.07m	/ 6.07m	382u	/ 384u
WR CMD -> Status Time	<8K		/		/
	8K - <64K		/		/
	64K - <512K	91.83m	/ 139.40m	37.31m	/ 37.40m
	>=512K		/		/
	ALL	91.83m	/ 139.40m	37.31m	/ 37.40m
RD CMD -> 1st Data Time	<8K		/		/
	8K - <64K		/		/
	64K - <512K	4.97m	/ 5.61m	223u	/ 224u
	>=512K		/		/
	ALL	4.97m	/ 5.61m	223u	/ 224u
WR CMD -> 1st XFER_RDY Time	<8K		/		/
	8K - <64K		/		/
	64K - <512K	26.52m	/ 28.85m	18.09m	/ 18.19m
	>=512K		/		/
	ALL	26.52m	/ 28.85m	18.09m	/ 18.19m
RD Pending IOs	<8K	0	/ 0	0	/ 0
	8K - <64K	0	/ 0	0	/ 0
	64K - <512K	13	/ 13	2	/ 5
	>=512K	0	/ 0	0	/ 0
WR Pending IOs	<8K	0	/ 0	0	/ 0
	8K - <64K	0	/ 0	0	/ 0
	64K - <512K	57	/ 217	26	/ 80
	>=512K	0	/ 0	0	/ 0

```
-----------------------------------------------------------------------------------------------------
=====================================================================================================

2016/05/19-09:09:29, [MAPS-1003], 101769, FID 128, WARNING, G620_066_223, Flow (ios_violin_1),
Condition=ios_violin_1(RD_STATUS_TIME_LT_8K/min>=1),
Current Value:[ RD_STATUS_TIME_LT_8K,837 Microseconds], RuleName=ios_violin_4k_rd_status, Dashboard
Category=Traffic Performance.
```

MAPS Dashboard:

```
root> mapsdb --show

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

DB start time:                  Thu May 19 08:59:07 2016
Active policy:                  ios_mod_policy
Configured Notifications:       RASLOG,EMAIL,SW_CRITICAL,SW_MARGINAL
Fenced Ports :                  None
Decommissioned Ports :          None
Fenced circuits :               N/A
Quarantined Ports :             None
Top Zoned PIDs <pid(it-flows)>: 0xdf2000(30) 0xdf2100(30) 0xdf0400(28) 0xdf1400(28) 0xdf1b00(22)

2 Switch Health Report:
=======================

Current Switch Policy Status: HEALTHY


3.1 Summary Report:
===================

Category                 |Today                     |Last 7 days               |
-------------------------------------------------------------------------------
Port Health              |No Errors                 |In operating range        |
BE Port Health           |No Errors                 |No Errors                 |
GE Port Health           |No Errors                 |No Errors                 |
Fru Health               |In operating range        |In operating range        |
Security Violations      |No Errors                 |No Errors                 |
Fabric State Changes     |No Errors                 |In operating range        |
Switch Resource          |In operating range        |In operating range        |
Traffic Performance      |Out of operating range    |In operating range        |
FCIP Health              |Not applicable            |Not applicable            |
Fabric Performance Impact|Out of operating range    |In operating range        |


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

Category(Rule Count)|RepeatCount|Rule Name                  |Execution Time   |Object             |Triggered Value(Units)|
-----------------------------------------------------------------------------------------------------------------------------
Traffic Performance |1          |ios_violin_4k_rd_status    |06/18/16 11:00:29|Flow (ios_violin_1)|867 Microseconds      |
(4)                 |           |                           |                 |                   |                      |
                    |1          |ios_violin_4k_rd_status    |06/18/16 12:59:29|Flow (ios_violin_1)|829 Microseconds      |
Fabric Performance  |1          |defALL_LOCAL_PIDSIT_FLOW_16|06/18/16 08:59:59|Pid 0xdf2400       |21 IT-Flow(s)         |
Impact(1)           |           |                           |                 |                   |                      |
                    |           |                           |                 |Pid 0xdf2500       |21 IT-Flow(s)         |
                    |           |                           |                 |Pid 0xdf1400       |28 IT-Flow(s)         |
                    |           |                           |                 |Pid 0xdf0400       |28 IT-Flow(s)         |
                    |           |                           |                 |Pid 0xdf2100       |30 IT-Flow(s)         |
```

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

Validate QoS functionality.

Test Execution

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

2. Start I/O across all pairs, and validate traffic priority.

Result Validation

1. Check I/O logs and verify that I/O continues without error.
2. Check switch error logs and switch port status for errors.
3. Verify traffic prioritization by examining performance and VC utilization.

Test Results

PASS. I/O completes without error in low, medium, and high priority zones.

2.5 QoS Integrity with CS_CTL Tag Based Frame Prioritization

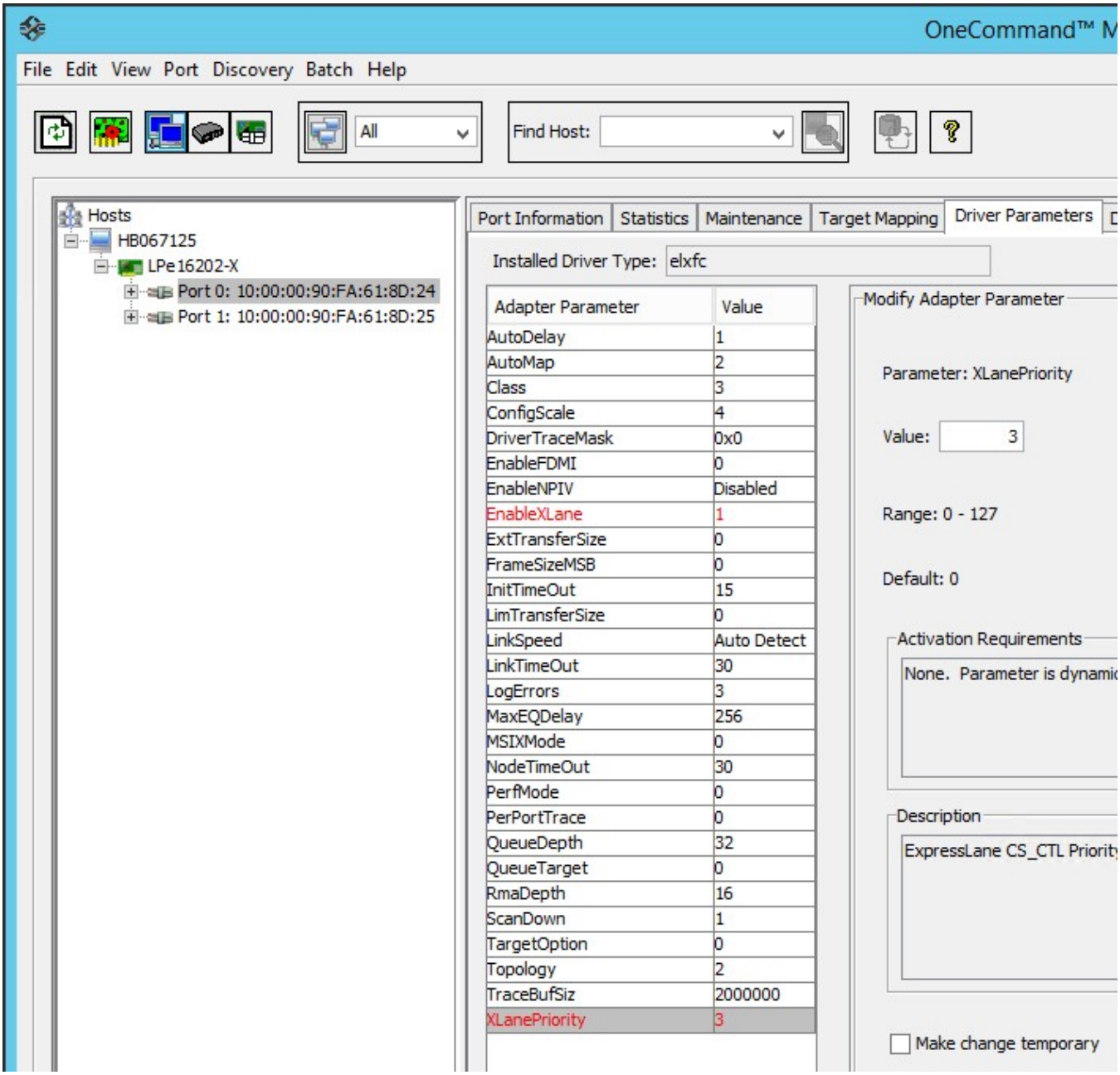
Test Objective

Validate CS_CTL QoS functionality.

Test Execution

1. Enable ExpressLane on the Emulex adapter driver by setting the value to 1 and setting the priority to 3 for high.

FIGURE 9 Configuring Driver Parameters in Emulex OneCommand Interface



2. Enable CS_CTL QoS Auto-Mode at the chassis level on all switches in the Brocade fabric.

```
> 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
```

To verify:

```
B6510_066_073:root> configshow -all | grep csctl
fos.csctlMode:1
```

3. Enable CS_CTL QoS on initiator and target ports.

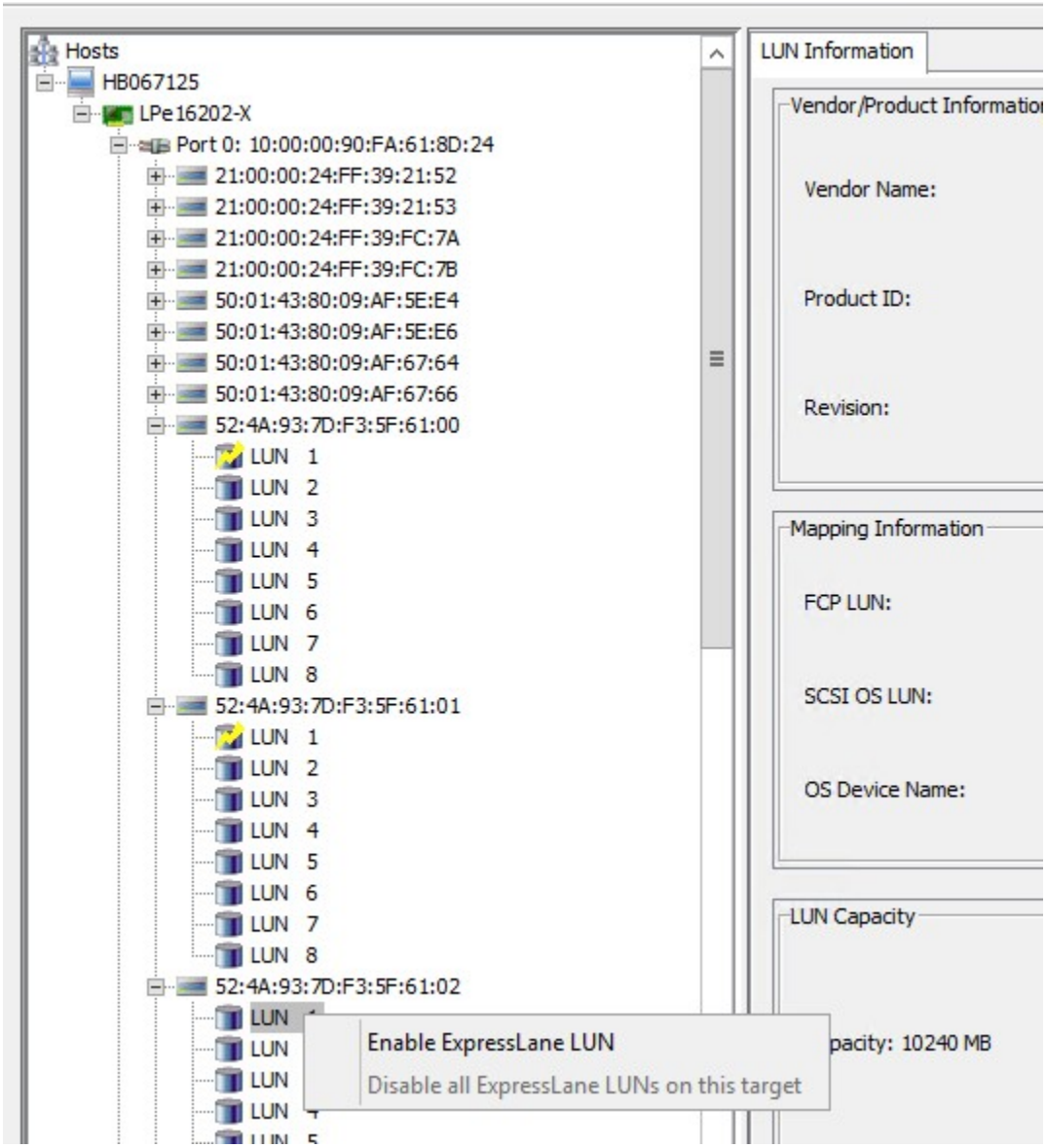
```
B6510_066_073:root> portcfgqos --enable 7 csctl_mode
Enabling CSCTL mode flows causes QoS zone flows to lose priority on such ports.

Do you want to proceed?(y/n):y
```

To verify:

```
B6510_066_073:root> portcfgshow 7 | grep -i csctl
CSCTL mode:              ON
```

4. Enable ExpressLane QoS on selected LUNs. For multipath LUNs, enable ExpressLane on each device path:

FIGURE 10 Enabling LUNs for ExpressLane in Emulex OneCommand Interface



5. Start high throughput I/O to all LUNs.

Result Validation

- Check I/O logs and verify that I/O continues without error for all LUNs.
- Verify that I/O performance is improved on ExpressLane-enabled LUNs.
- 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
```

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

Test Results

PASS. Prioritized LUNs display higher performance, the traffic receives prioritization through the fabric, and I/O completes without error.

2.6 Storage Device—FC Protocol Jammer Test Suite

Test Objective

Perform FC Jammer Tests including areas such as: CRC corruption, packet corruption, missing frame, host error recovery, target error recovery.

Test Execution

1. Insert the Jammer device in the I/O path on the storage link.
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
3. Verify Jammer operations and recovery with Analyzer.

Result Validation

Check the Medusa log and verify that I/O recovers and completes without error.

Test Results

PASS. Introduced packet anomalies and verified proper recovery.

3.0 Stress and Error Recovery with Device Multipath

3.1 Storage Device Fabric I/O Integrity—Congested Fabric

Test Objective

Validate I/O integrity in a congested fabric environment.

Test Execution

From all initiators start a mixture of READ, READ/WRITE, and WRITE traffic continuously to all their targets for a 24-hour period.

Result Validation

1. Check the host and storage logs for errors.
2. Verify the link congestion and check the switch logs for errors.
3. Check I/O generator tool logs to verify that I/O runs without error.

Test Results

PASS. In a congested fabric, I/O runs successfully without error or failover.

3.2 Storage Device Name-Server Integrity—Device Recovery with Port Toggle

Test Objective

Validate path recovery behavior on storage and host ports.

Test Execution

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

Result Validation

1. Check switch port status after toggle and for errors in the switch error logs.

```
root> portshow 19
portHealth: HEALTHY   <==
……
portState: 1    Online   <==
……
portWwn of device(s) connected:
10:00:8c:7c:ff:22:f7:81   <==
Distance:  normal
portSpeed: N16Gbps
```

2. Check host multipath status on hosts to verify 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 I/O continues without errors.

Test Results

PASS. Failover between 8 logical paths (2 host x 4 storage) tested successfully. Paths recover, and I/O completes without errors.

3.3 Storage Device Name-Server Integrity—Device Recovery with Device Relocation

Test Objective

Validate storage device path recovery after physical port relocation.

Test Execution

1. Perform the test sequentially for each storage device port.
2. Disconnect and reconnect a port to a different switch port in the same fabric.

Result Validation

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

```
root> portshow 8
portHealth: HEALTHY   <==
……
portState: 1    Online   <==
……
portWwn of device(s) connected:
52:4a:93:7d:f3:5f:61:11   <==
Distance:  normal
portSpeed: N8Gbps
```

2. Check the host 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. Confirm that there are no errors in the Medusa log.

Test Results

PASS. The physical move of the storage port shows successful recovery. Paths recover, and I/O completes without error.

3.4 Storage Device Name-Server Stress—Device Recovery with Device Port Toggle

Test Objective

Validate path recovery behavior on storage and host ports for an extended duration run.

Test Execution

1. Set up multipath on the host, and start I/O.
2. Sequentially toggle each initiator and target port in the fabric (multiple iterations).
3. Run for an extended period.

Result Validation

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

```
root> portshow 19
portHealth: HEALTHY   <==
.............
portState: 1    Online   <==
.............
portWwn of device(s) connected:
10:00:8c:7c:ff:22:f7:81    <==
Distance:  normal
portSpeed: N16Gbps
```

2. Check the host 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 the host and storage error logs, and verify that I/O continues without error.

Test Results

PASS. 48-hour run; paths recover, and I/O completes without error.

3.5 Storage Device Recovery—ISL Port Toggle (Sequential)

Test Objective

Validate path recovery and I/O integrity when ISL links are disabled.

Test Execution

1. Set up multipath on the host, and start I/O.
2. Sequentially toggle each ISL path (one at a time) on all switches.

Result Validation

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

```
root> fabricshow
Switch ID   Worldwide Name            Enet IP Addr    FC IP Addr    Name
-----------------------------------------------------------------------------
  1: fffc01 50:00:53:35:b1:d3:df:1b   0.0.0.0         0.0.0.0       "fcr_xd_1_40"
  3: fffc03 10:00:00:05:33:13:95:9a   10.38.66.73     0.0.0.0       "B6510_066_073"
 19: fffc13 10:00:00:05:33:a5:bf:86   10.38.66.74     0.0.0.0      >"B6510_066_074"
 82: fffc52 10:00:00:05:33:13:96:5a   10.38.66.82     0.0.0.0       "B6510_066_082"
 83: fffc53 10:00:00:05:33:5b:1d:1d   10.38.66.83     0.0.0.0       "B6510_066_083"

The Fabric has 5 switches   <==
Fabric Name: SSR
```

2. Check the switch logs for errors, and verify that I/O failed over to alternate ISL path in the fabric.
3. Check the host and storage error logs, and verify that I/O continues without error.

Test Results

PASS. Paths recover, and I/O completes without error.

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

Test Objective

Validate path recovery and I/O integrity when all ISL links on a switch are disabled.

Test Execution

1. Ensure ISL redundancy by provisioning multiple ISLs connected to different switches to provide multiple paths through the fabric.
2. Set up multipath on the host, and start I/O.
3. On each switch, disable all ISL links at once.

Result Validation

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

```
root> fabricshow
Switch ID   Worldwide Name           Enet IP Addr    FC IP Addr      Name
-------------------------------------------------------------------------------
  1: fffc01 50:00:53:35:b1:d3:df:1b 0.0.0.0         0.0.0.0         "fcr_xd_1_40"
  3: fffc03 10:00:00:05:33:13:95:9a 10.38.66.73     0.0.0.0         "B6510_066_073"
 19: fffc13 10:00:00:05:33:a5:bf:86 10.38.66.74     0.0.0.0        >"B6510_066_074"
 82: fffc52 10:00:00:05:33:13:96:5a 10.38.66.82     0.0.0.0         "B6510_066_082"
 83: fffc53 10:00:00:05:33:5b:1d:1d 10.38.66.83     0.0.0.0         "B6510_066_083"

The Fabric has 5 switches     <==
Fabric Name: SSR
```

2. Check the switch logs for errors, and verify that I/O failed over to an alternate ISL path in the fabric.
3. Check the host and storage error logs, and verify that I/O continues without error.

Test Results

PASS. Paths recover, and I/O completes without error.

3.7 Storage Device Recovery—Director Blade Maintenance

Test Objective

Validate path recovery and I/O integrity during director blade maintenance.

Test Execution

1. Uplink edge switch ISLs to different blades on the directors.
2. Sequentially toggle each director blade.
3. Include blade disable/enable and blade power on/off.

   Syntax:

   slotpoweroff/slotpoweron, **bladedisable/bladeenable**, **slotshow**

Result Validation

1. Check FC fabric status after the blade toggle. Verify that all nodes are present in the fabric.

```
root> fabricshow
Switch ID   Worldwide Name          Enet IP Addr    FC IP Addr      Name
-------------------------------------------------------------------------------
  1: fffc01 50:00:53:39:47:bd:6e:0a 0.0.0.0         0.0.0.0         "fcr_fd_1"
  2: fffc02 10:00:00:05:1e:09:15:3e 10.38.51.5      0.0.0.0         "sw5300sus1005"
  3: fffc03 10:00:c4:f5:7c:07:c2:0c 10.38.51.3      0.0.0.0         "sw300sus1003"
  4: fffc04 10:00:c4:f5:7c:7b:86:1f 10.38.51.4      0.0.0.0         "sw6510sus1004"
  5: fffc05 10:00:00:05:33:13:80:ef 10.38.51.24     0.0.0.0         "swBESsus1024"
 13: fffc0d 10:00:00:05:1e:4c:ff:00 10.38.51.13     0.0.0.0         "sw8510sus1013"
 16: fffc10 10:00:00:05:1e:44:02:00 10.38.51.16     0.0.0.0        >"sw8518sus1016"
 22: fffc16 10:00:00:05:1e:07:7d:99 10.38.51.22     0.0.0.0         "sw5100sus1022"
 23: fffc17 10:00:00:27:f8:2b:9f:aa 10.38.51.23     0.0.0.0         "sw6520sus1023"
 25: fffc19 10:00:00:05:1e:54:8b:d4 10.38.51.25     0.0.0.0         "swBESsus1025"
 26: fffc1a 10:00:00:05:33:8f:b6:77 10.38.51.26     0.0.0.0         "sw6510sus1026"
 29: fffc1d 10:00:00:05:33:83:a4:00 10.38.51.29     0.0.0.0         "swDCX4sus1029"
 32: fffc20 10:00:00:05:1e:b8:c1:00 10.38.51.32     0.0.0.0         "swDCXsus1032"
160: fffca0 10:00:00:05:1e:a8:35:9d 10.38.51.21     0.0.0.0         "swVA40FCsus1021"
171: fffcab 10:00:00:05:1e:d7:1a:04 10.38.51.171    172.172.172.26  "sw5480sus1171"
                                    2620:100:4:fa00:205:1eff:fed7:1a04
183: fffcb7 10:00:00:05:33:00:b2:e2 10.38.51.183    0.0.0.0         "swM5424sus1183"
195: fffcc3 10:00:00:05:33:8d:df:d1 10.38.51.195    0.0.0.0         "sw5480sus1195"
199: fffcc7 10:00:00:27:f8:44:33:61 10.38.51.199    0.0.0.0         "sw6547sus1199"
                                    2620:100:4:fa00:227:f8ff:fe44:3362
220: fffcdc 10:00:00:05:1e:86:1b:1f 10.38.51.220    0.0.0.0         "sw5450sus1220"

The Fabric has 19 switches
```

2. Check the switch logs for errors, and verify that I/O failed over to an alternate ISL path in the fabric.
3. Check host and storage error logs, and verify that I/O continues without error.

Test Results

PASS. Paths recover, and I/O completes without error.

3.8 Storage Device Recovery—Switch Offline

Test Objective

Validate path recovery and I/O integrity during switch offline events.

Test Execution

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

Result Validation

1. Check FC fabric status after the switch toggle. Verify all nodes are present in the fabric.

```
root> fabricshow
Switch ID   Worldwide Name          Enet IP Addr    FC IP Addr      Name
-------------------------------------------------------------------------------
  1: fffc01 50:00:53:35:b1:d3:df:1b 0.0.0.0         0.0.0.0         "fcr_xd_1_40"
  3: fffc03 10:00:00:05:33:13:95:9a 10.38.66.73     0.0.0.0         "B6510_066_073"
 19: fffc13 10:00:00:05:33:a5:bf:86 10.38.66.74     0.0.0.0        >"B6510_066_074"
 82: fffc52 10:00:00:05:33:13:96:5a 10.38.66.82     0.0.0.0         "B6510_066_082"
 83: fffc53 10:00:00:05:33:5b:1d:1d 10.38.66.83     0.0.0.0         "B6510_066_083"

The Fabric has 5 switches <==
Fabric Name: SSR
```

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

```
root> switchshow
switchName:     B6510_066_088
switchType:     109.1
switchState:    Online    <==
switchMode:     Native
switchRole:     Subordinate
switchDomain:   88
switchId:       fffc58
switchWwn:      10:00:00:27:f8:06:23:28
zoning:         ON (SSR)
switchBeacon:   OFF
FC Router:      ON
FC Router BB Fabric ID: 100
Address Mode:   0
Fabric Name:    SSR_2
```

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

Test Results

PASS. Paths fail over and recover, and I/O completes without error.

3.9 Storage Device Recovery—Switch Firmware Download

Test Objective

Verify I/O will continue with minimal disruption throughout the switch firmware upgrade process.

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 the nodes merge back in the FC fabric.

```
root> version
Kernel:     2.6.14.2
Fabric OS:  v7.3.1
Made on:    Thu Dec 11 14:30:38 2014
Flash:      Wed Dec 17 09:53:59 2014
BootProm:   1.0.11

root> fabricshow
Switch ID   Worldwide Name          Enet IP Addr    FC IP Addr      Name
-------------------------------------------------------------------------------
  1: fffc01 50:00:53:35:b1:d3:df:1b 0.0.0.0         0.0.0.0         "fcr_xd_1_40"
  3: fffc03 10:00:00:05:33:13:95:9a 10.38.66.73     0.0.0.0         "B6510_066_073"
 19: fffc13 10:00:00:05:33:a5:bf:86 10.38.66.74     0.0.0.0        >"B6510_066_074"
 82: fffc52 10:00:00:05:33:13:96:5a 10.38.66.82     0.0.0.0         "B6510_066_082"
 83: fffc53 10:00:00:05:33:5b:1d:1d 10.38.66.83     0.0.0.0         "B6510_066_083"

The Fabric has 5 switches   <==
Fabric Name: SSR
```

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

Test Results

PASS. I/O failover and recovery is successful during firmware download. Paths recover, and I/O completes without error.

4.0 Storage Device—Fibre Channel Routing (FCR) Internetworking Tests

(Testing covered in FOS 8.1.0 and previous releases.)

4.1 Storage Device Internetworking Validation with FC Host

Test Objective

Validate storage targets are imported successfully in a routed Fibre Channel environment.

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.

```
root> fcrfabricshow
FC Router WWN: 10:00:00:05:33:13:96:5a, Dom ID:  82,
Info: 10.38.66.82, "B6510_066_082"
   EX_Port       FID    Neighbor Switch Info (enet IP, WWN, name)
   ------------------------------------------------------------------------------
      40         40     10.38.66.88      10:00:00:27:f8:06:23:28   "B6510_066_088"
      41         40     10.38.66.92      10:00:00:27:f8:66:f3:81   "B6520_066_92"

FC Router WWN: 10:00:00:05:33:5b:1d:1d, Dom ID:  83,
Info: 10.38.66.83, "B6510_066_083"
   EX_Port       FID    Neighbor Switch Info (enet IP, WWN, name)
   ------------------------------------------------------------------------------
      40         40     10.38.66.92      10:00:00:27:f8:66:f3:81   "B6520_066_92"
      41         40     10.38.66.88      10:00:00:27:f8:06:23:28   "B6510_066_088"

root> fcrproxydevshow
  Proxy           WWN             Proxy      Device   Physical    State
 Created                           PID       Exists     PID
in Fabric                                   in Fabric
-------------------------------------------------------------------------
    40   10:00:00:05:1e:60:b4:6b  02ff02      100      132400   Imported
    40   10:00:00:05:1e:60:b4:6c  02fe02      100      032400   Imported
……….
   100   50:05:07:60:5e:80:76:52  01fe02       40      612800   Imported
   100   50:05:07:60:5e:80:76:71  01ff02       40      642300   Imported
Total devices displayed: 20
```

2. Verify I/O runs successfully without errors.

Test Results

PASS. All devices are available via routed fabric. Running I/O confirms successful routing.

4.2 Storage Device Edge Recovery After FCR Disruptions

Test Objective

Validate that storage target paths recover successfully from disruptions in a routed Fibre Channel environment. Edge-Backbone-Edge configuration.

Test Execution

1. Set up FCR in an Edge-Backbone-Edge configuration.
2. Set up LSAN zoning.
3. With I/O running, perform sequential reboots, switch disables, and ISL port toggles on the switches in the backbone fabric.

Result Validation

1. Verify FCR fabric state throughout the disruptions.

```
root> fcrfabricshow
FC Router WWN: 10:00:00:05:33:13:96:5a, Dom ID:  82,
Info: 10.38.66.82, "B6510_066_082"
   EX_Port       FID    Neighbor Switch Info (enet IP, WWN, name)
```

```
   ---------------------------------------------------------------------------
      40         40     10.38.66.88      10:00:00:27:f8:06:23:28   "B6510_066_088"
      41         40     10.38.66.92      10:00:00:27:f8:66:f3:81   "B6520_066_92"

FC Router WWN: 10:00:00:05:33:5b:1d:1d, Dom ID:  83,
Info: 10.38.66.83, "B6510_066_083"
   EX_Port       FID    Neighbor Switch Info (enet IP, WWN, name)
   ---------------------------------------------------------------------------
      40         40     10.38.66.92      10:00:00:27:f8:66:f3:81   "B6520_066_92"
      41         40     10.38.66.88      10:00:00:27:f8:06:23:28   "B6510_066_088"

root> fcrproxydevshow
  Proxy           WWN             Proxy      Device   Physical    State
 Created                           PID       Exists     PID
in Fabric                                   in Fabric
-------------------------------------------------------------------
    40   10:00:00:05:1e:60:b4:6b  02ff02      100      132400   Imported
    40   10:00:00:05:1e:60:b4:6c  02fe02      100      032400   Imported
……….
   100   50:05:07:60:5e:80:76:52  01fe02       40      612800   Imported
   100   50:05:07:60:5e:80:76:71  01ff02       40      642300   Imported
Total devices displayed: 20
```

2. Check the switch logs for errors.
3. Check host and storage logs, and verify I/O runs without error.

Test Results

PASS. Paths fail over and recover with FCR disruptions, and I/O completes without error.

4.3 Storage Device Backbone Recovery After FCR Disruption

Test Objective

Validate that storage target paths recover successfully from disruptions in a routed Fibre Channel environment. Edge-Backbone configuration.

Test Execution

1. Set up FCR in an Edge-Backbone configuration.
2. Set up LSAN zoning.
3. With I/O running, 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.

```
root> fcrfabricshow
FC Router WWN: 10:00:00:05:33:13:96:5a, Dom ID:  82,
Info: 10.38.66.82, "B6510_066_082"
   EX_Port       FID    Neighbor Switch Info (enet IP, WWN, name)
   ---------------------------------------------------------------------------
      40         40     10.38.66.88      10:00:00:27:f8:06:23:28   "B6510_066_088"
      41         40     10.38.66.92      10:00:00:27:f8:66:f3:81   "B6520_066_92"

FC Router WWN: 10:00:00:05:33:5b:1d:1d, Dom ID:  83,
Info: 10.38.66.83, "B6510_066_083"
   EX_Port       FID    Neighbor Switch Info (enet IP, WWN, name)
```

```
   ---------------------------------------------------------------------------
      40         40     10.38.66.92      10:00:00:27:f8:66:f3:81   "B6520_066_92"
      41         40     10.38.66.88      10:00:00:27:f8:06:23:28   "B6510_066_088"

root> fcrproxydevshow
  Proxy           WWN             Proxy      Device   Physical    State
 Created                           PID       Exists     PID
in Fabric                                   in Fabric
-------------------------------------------------------------------------
    40   10:00:00:05:1e:60:b4:6b  02ff02      100      132400   Imported
    40   10:00:00:05:1e:60:b4:6c  02fe02      100      032400   Imported
……….
   100   50:05:07:60:5e:80:76:52  01fe02       40      612800   Imported
   100   50:05:07:60:5e:80:76:71  01ff02       40      642300   Imported
Total devices displayed: 20
```

2. Check the switch logs for errors.
3. Check host and storage logs, and verify that I/O runs without error.

Test Results

PASS. Paths fail over successfully, and I/O completes without error.

5.0 Optional/Additional Tests

5.1 Non-disruptive Firmware Upgrade on the Array

Test Objective

Run I/O workload generation with varying dimensions (block size, read/write/mix), and verify that performance characteristics are as expected.

Test Execution

1. Run an I/O loop at block transfer sizes of 512, 1k, 2k, 4k, 8k, 16k, 32k, 64k, 128k, 256k, 512k, and 1m.
2. Include a nested loop of 100% read, 100% write, and 50% read/write.

   Repeat the test for the following configurations:

   - 1 host port to 1 target port (single path)
   - 2 host ports to multiple target ports (multipath)
   - Multiple hosts to multiple target ports (multihost multipath)

Result Validation

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

Test Results

PASS. All workload runs are monitored at the host, storage, and fabric and complete without I/O error or fault. Performance behavior is as expected.

5.2 Synthetic I/O Workload Loop with Varying Block Sizes

Test Objective

Run workload test suite including varying application workloads generated from multiple VMs.

Test Execution

1. Configure a 2-host VMware cluster with multipath on 2 initiator ports per host, and 4 target ports.
2. Configure workload generation from 8 worker VMs using VMware IO Analyzer.
3. Run a variety of application IO workload patterns from the IO Analyzer suite. For this round of testing, the following workloads were used:
   - Workstation
   - Webserver
   - Video on Demand
   - Max Write Throughput
   - Max Write IOPS
   - Max IOPS
   - Max Throughput
   - SQL Server 16k
   - 4k Read
   - Exchange 2007
   - OLTP 4k

Result Validation

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

Test Results

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

5.3 Storage Device Failover Tests

Test Objective

Validate array controller failover and failback behavior.

Test Execution

1. Run continuous I/O to the array.

2. Create a controller failure by disabling all FC links connected to the active controller.

```
> portdisable 26-27 <== Run this on both switches.
```

3. Confirm that the array is in the failover state from the array CLI.

```
[root@ssr-vmem-brcd-mg-a ~]# isscli getfailoverstatus
Failover Servers=ssr-vmem-brcd-mg-a / ssr-vmem-brcd-mg-b
Configuration Type=Mutual Failover
Failover State=Failed Over to Secondary Server ssr-vmem-brcd-mg-b.
Failover Suspended=No
Stretched Cluster Enabled=No
FailOverStatus=3(DOWN)
```

4. Re-enable the switch ports.

```
> portenable 26-27 <== Run this on both switches.
```

5. Execute a manual failback from the array CLI interface on the current active controller.

```
[root@ssr-vmem-brcd-mg-b ~]# isscli stoptakeover
Command: stoptakeover executed successfully.
```

Result Validation

1. Confirm that the array NPIV ports of the offline controller are automatically migrated to the standby controller. The following is an inspection of MG-B ports after failing MG-A:

```
> portshow 24 | grep -i -A3 wwn
portWwn:20:18:c4:f5:7c:41:1b:14
portWwn of device(s) connected:
21:00:00:1b:97:28:d8:41 <== Servicing NPIV port from the failed controller
21:00:00:1b:97:28:da:f1 <== NPIV target port for this controller
21:00:00:0e:1e:28:da:f1 <== The physical port
```

2. Confirm that I/O continues uninterrupted.

3. Check the host and storage logs for errors throughout the failover operations.

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

Test Results

PASS. The target NPIV ports are successfully migrated to the standby controller upon failure of all links to the active controller. Manual failback is successful and returns the array ports to their initial state. I/O is uninterrupted and continues without error.

Test Conclusions

1. Achieved a 100% pass rate on all test cases in the SFR qualification test plan. The network and the storage were able 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 demonstrated across all workload types.
3. The results confirm that the Violin FSP 7300 array interoperates seamlessly with Brocade Fibre Channel fabrics, and together they demonstrate high availability, performance, and low latency.
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 to provide path redundancy through the fabric.
5. We recommend enabling 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 additional MAPS feature of Fabric Performance Impact monitoring should also be enabled to detect bottlenecks in the form of timeouts and latency. Using MAPS is recommended to maximize the benefit of high-performance low-latency storage.
6. Implementing the IO Insight feature to monitor the critical flows at the SCSI level can provide valuable data and insight into traffic performance across the fabric. Combining IO Insight with MAPS custom rules and alerts can provide proactive monitoring to help preserve the investment in flash storage performance.
7. QoS Zoning can be used to classify host-target traffic into high, medium, or low priority zones and 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.
8. 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.
9. For optimal availability and performance, consideration should be given to multipath configuration on the host side. While Windows 2008 and 2012 will provide Round-Robin behavior by default, Linux systems will benefit from adding a custom entry to /etc/multipath.conf, and VMware hosts systems should be changed from the default Most Recently Used (VMware) setting to Round-Robin (VMware). Actively using all available paths provides a significant improvement in performance throughput.
10. Once an array failover occurs, a manual failback operation is required. For optimal high availability, it is recommended to observe best practices when attaching array ports to the SAN fabric as outlined above in the "Violin Memory Array Configuration" section or in the Violin document *7300 Flash Storage Platform Installation Guide.*