SPC BENCHMARK 1™

FULL DISCLOSURE REPORT

HUAWEI TECHNOLOGIES CO., LTD
HUAWEI OCEANSTOR™ DORADO5000 V3

SPC-1 V3.5.0

SUBMISSION IDENTIFIER: A31010

SUBMITTED FOR REVIEW: DECEMBER 5, 2017

Second Edition – February 2018

THE INFORMATION CONTAINED IN THIS DOCUMENT IS DISTRIBUTED ON AN AS IS BASIS WITHOUT ANY WARRANTY EITHER EXPRESS OR IMPLIED. The use of this information or the implementation of any of these techniques is the customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by Huawei for accuracy, in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

This publication was produced in the People's Republic of China. Huawei may not offer the products, services, or features discussed in this document in other countries, and the information is subject to change with notice. Consult your local Huawei representative for information on products and services available in your area.

Trademarks

SPC Benchmark 1, SPC-1, SPC-1 IOPS, SPC-1 LRT and SPC-1 Price-Performance are trademarks of the Storage Performance Council.

Huawei, the Huawei logo, OceanStor™ and Dorado™ are trademarks or registered trademarks of Huawei in the United States and other countries. All other brands, trademarks, and product names are the property of their respective owners.

Benchmark Specification and Glossary

The official SPC Benchmark 1™ (SPC-1™) specification is available on the website of the Storage Performance Council (SPC) at www.storageperformance.org.

The SPC-1™ specification contains a glossary of the SPC-1™ terms used in this publication.

Table of Contents

AUDIT CERTIFICATION

InfoSizing
The Right Metric For Sizing IT

Storage Performance Council
Certified Auditor

Zhong Xu
Huawei Technologies Co., Ltd.
Huawei Industrial Base, Bantian,
Longgang, Shenzhen city,
Guangdong province, China

December 5, 2017

I verified the SPC Benchmark 1™ (SPC-1™ Revision3.5.0) test execution and performance results of the following Tested Storage Product:

HUAWEI OCEANSTOR™ DORADO5000 V3

The results were:

SPC-1 IOPS™	**800,465**
SPC-1 Price-Performance™	**$0.48/SPC-1 IOPS™**
SPC-1 IOPS™ Response Time	0.507 ms
SPC-1 Overall Response Time	0.341 ms
SPC-1 ASU Capacity	27,058 GB
SPC-1 ASU Price	$13.95/GB
SPC-1 Total System Price	$377,376.62

In my opinion, these performance results were produced in compliance with the SPC requirements for the benchmark.

The testing was executed using the SPC-1 Toolkit Version 3.0 Build d34fb3c. The audit process was conducted in accordance with the SPC Policies and met the requirements for the benchmark.

A Letter of Good Faith was issued by the Test Sponsor, stating the accuracy and completeness of the documentation and testing data provided in support of the audit of this result.

A Full Disclosure Report for this result was prepared by InfoSizing, reviewed and approved by the Test Sponsor, and can be found at **www.storageperformance.org** under the Submission Identifier **A31010**.

20 Kreg Lane • Manitou Springs, CO 80829 • 719-473-7555 • www.sizing.com

The independent audit process conducted by InfoSizing included the verifications of the following items:

- The physical capacity of the data repository;
- The total capacity of the Application Storage Unit (ASU);
- The accuracy of the Benchmark Configuration diagram;
- The tuning parameters used to configure the Benchmark Configuration;
- The Workload Generator commands used to execute the testing;
- The validity and integrity of the test result files;
- The compliance of the results from each performance test;
- The compliance of the results from the persistence test;
- The compliance of the submitted pricing model; and
- The differences between the tested and the priced configuration, if any.

The Full Disclosure Report for this result was prepared in accordance with the disclosure requirements set forth in the specification for the benchmark.

The following benchmark requirements, if any, were waived according to the SPC Policies:

- None.

Respectfully Yours,

François Raab, Certified SPC Auditor

20 Kreg Lane • Manitou Springs, CO 80829 • 719-473-7555 • www.sizing.com

LETTER OF GOOD FAITH

HUAWEI

©Huawei Technologies Co., Ltd.
Huawei Industrial Base, Bantian, Longgang
Shenzhen city
Guangdong province
China
Tel: 0086-755-28780808
http://www.huawei.com/en/

Date: December 4, 2017

From: Huawei Technologies Co., Ltd.

To: Mr. Francois Raab, Certified SPC Auditor
InfoSizing
20 Kreg Lane
Manitou Springs, CO 80829

Subject: SPC-1 Letter of Good Faith for the Huawei OceanStor Dorado 5000 V3

Huawei Technologies Co., Ltd. is the SPC-1 Test Sponsor for the above listed product. To the best of our knowledge and belief, the required SPC-1 benchmark results and materials we have submitted for that product are complete, accurate, and in full compliance with V3.5 of the SPC-1 benchmark specification.
In addition, we have reported any items in the Benchmark Configuration and execution of the benchmark that affected the reported results even if the items are not explicitly required to be disclosed by the SPC-1 benchmark specification.

Signed: Date:

2017.12.04

Meng Guanglbin
President of Storage Product Line

SPC BENCHMARK 1™

EXECUTIVE SUMMARY

HUAWEI TECHNOLOGIES CO., LTD.
HUAWEI OCEANSTOR™ DORADO5000 V3

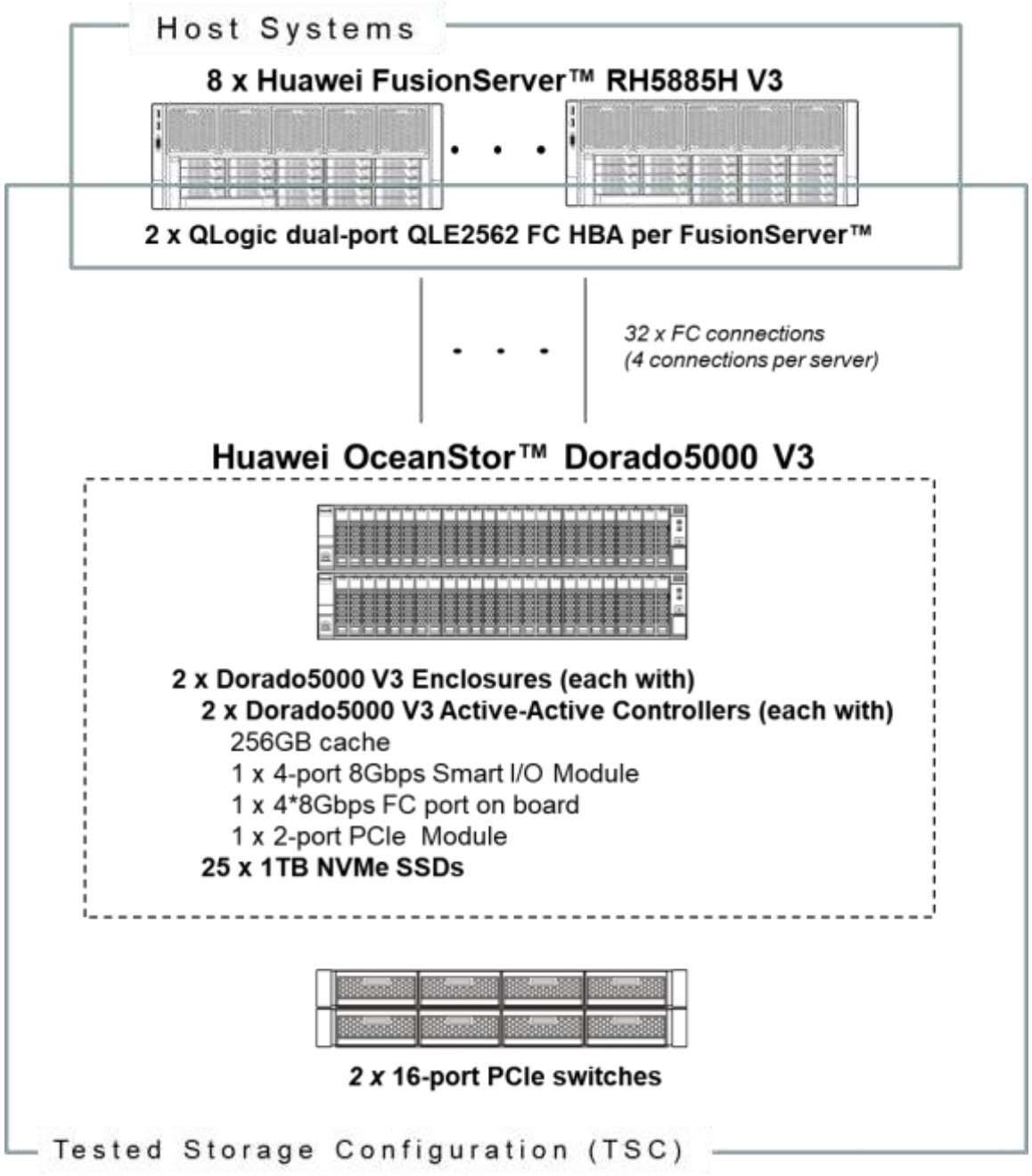
SPC-1 IOPS™	**800,465**
SPC-1 Price-Performance™	**$471.45/SPC-1 KIOPS™**
SPC-1 IOPS™ Response Time	0.507 ms
SPC-1 Overall Response Time	0.341 ms
SPC-1 ASU Capacity	27,058 GB
SPC-1 ASU Price	$13.95/GB
SPC-1 Total System Price	$377,376.62
Data Protection Level	Protected 2 (RAID 6 and full redundancy)
Physical Storage Capacity	50,791 GB
Pricing Currency / Target Country	U.S. Dollars / USA

SPC-1 V3.5.0

SUBMISSION IDENTIFIER: A31010

SUBMITTED FOR REVIEW: DECEMBER 5, 2017

Benchmark Configuration Diagram



Tested Storage Product Description

Huawei's OceanStor™ Dorado™ V3 all-flash storage is the ideal choice for enterprises' mission-critical business. It is the industry's first commercial use of NVMe all-flash storage, and it delivers high-performing, reliable, and efficient storage services.

The HyperMetro gateway-free active-active solution ensures 99.9999% availability, and 3:1 data reduction guarantee reduces TCO by 50%.

The Dorado satisfies the storage requirements of databases, Virtual Desktop Infrastructure (VDI) and Virtual Server Infrastructure (VSI), smoothing the way to the all-flash era.

For more details, visit:

http://e.huawei.com/en/products/cloud-computing-dc/storage/unified-storage/dorado-v3

Priced Storage Configuration Components

16 x QLogic dual-ported QLE2562 FC HBA
2 x Dorado5000 V3 Enclosures (each with) **2 x Dorado5000 V3 Active-Active Controllers (each with)** **256GB cache** **1 x 4-port 8Gbps Smart I/O Module** **1 x 4*8Gbps FC port on board** **1 x 2-port PCIe Module** **25 x 1TB NVMe SSDs**
2 x 16-port PCIe 2.0 switches

Storage Configuration Pricing

	Description	Qty	Unit Price	Ext. Price	Disc.	Disc. Price
Hardware & Software						
D5V3-512G-NAC-8	Dorado5000 V3(2U,Dual Ctrl,NVMe,AC\240HVDC,512GB Cache,SmartIO,16*8Gb FC,25*2.5",SPE61C0225)	2	123,590.82	247,181.64	45%	135,949.90
DV3-LPU5PCIE	8G PCIE Interface Board(with two NT Ports)	4	1,929.20	7,716.80	45%	4,244.24
D5V3-SSD-NVME-1T	1TB SSD NVMe Disk Unit(2.5")	50	4,971.40	248,570.00	45%	136,713.50
N8GHBA000	QLOGIC QLE2562 HBA Card,PCIE,8Gbps DualPort ,Fiber Channel Multimode LC Optic Interface,English Manual, No Drive CD	16	1,698.00	27,168.00	0%	27,168.00
SN2F01FCPC	Patch Cord,DLC/PC,DLC/PC,Multi-mode,3m,A1a.2,2mm,42mm DLC,OM3 bending insensitive	32	19.80	633.60	0%	633.60
DV3-PCIESWITCH3	PCIe 3.0 Switch(AC\240HVDC,8GB Cache,16 Port,SWE1600P08)	2	16,218.00	32,436.00	0%	32,436.00
AOC-QSFP41G	Quadwire 40 Gb/s Parallel AOC for PCIe 3.0	8	1,065.30	8,522.40	0%	8,522.40
D5V3-LBS	Basic Software License(Include DeviceManager,SmartThin,SmartMigration,SmartDedupe,SmartCompression,eService,SystemReporter,UltraPath)	1	7,196.00	7,196.00	44%	4,029.76
				Hardware & Software Subtotal		**349,697.40**
Support & Maintenance						
02351GSA-88134ULF-36	Dorado5000 V3(2U,Dual Ctrl,NVMe,AC\240HVDC,512GB Cache,SmartIO,8*8Gb FC,25*2.5",SPE61C0225)-Hi-Care Onsite Premier 24x7x4H Engineer Onsite Service-36Month(s)	2	4,446.00	8,892.00	10%	8,002.80
02351GTY-88134ULF-36	1TB SSD NVMe Disk Unit(2.5")-Hi-Care Onsite Premier 24x7x4H Engineer Onsite Service-36Month(s)	50	216.00	10,800.00	10%	9,720.00
88033NAM-88134UHK-36	Basic Software License(Include DeviceManager,SmartThin,SmartMigration,SmartDedupe,SmartCompression,eService,SystemReporter)-Hi-Care Application Software Upgrade Support Service,UltraPath Software License-36Month(s)	1	1,854.00	1,854.00	30%	1,297.80
8812102575	OceanStor Dorado5000 V3 Installation Service - Engineering	1	9,620.68	9,620.68	10%	8,658.61
				Support & Maintenance Subtotal		**27,679.21**
SPC-1 Total System Price						**377,376.61**
SPC-1 IOPS™						800,465
SPC-1 Price-Performance™ ($/SPC-1 KIOPS™)						**471.45**
SPC-1 ASU Capacity (GB)						27,058
SPC-1 ASU Price ($/GB)						**13.95**

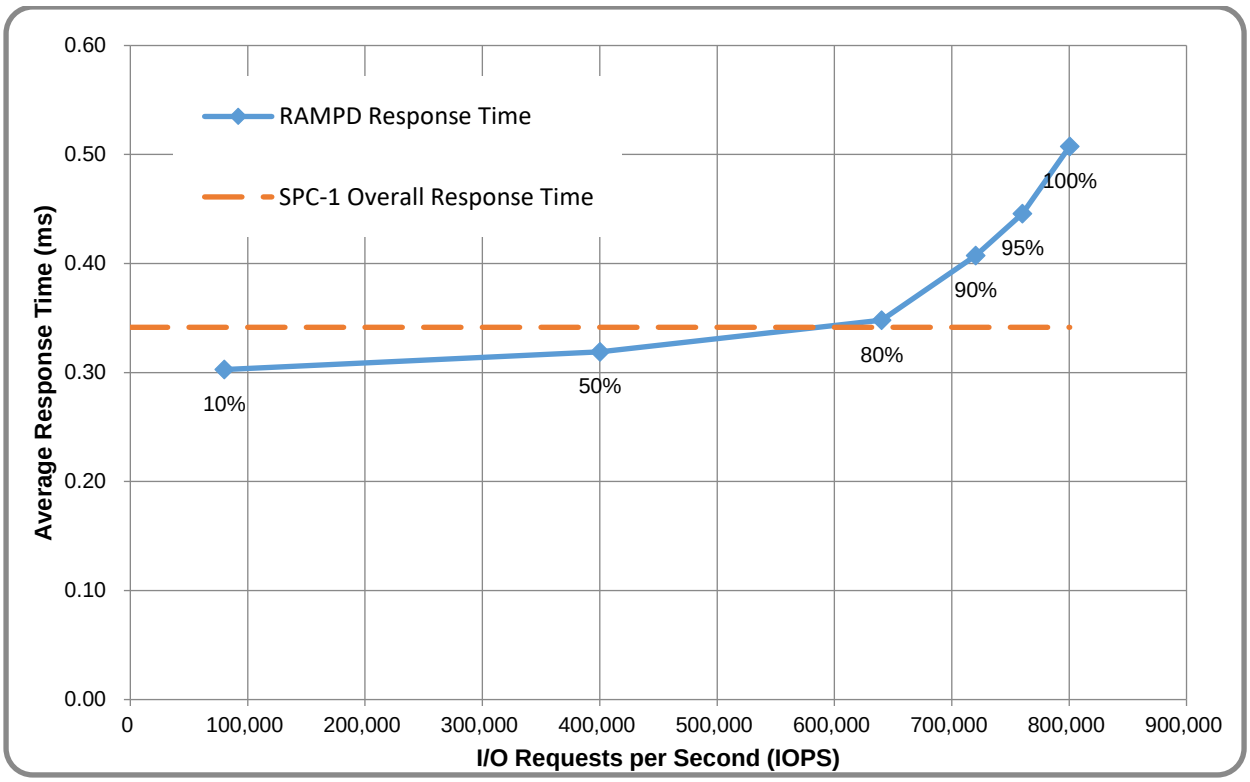
Third-Party Reseller: Huawei Technologies Co., Ltd. only sells its products to third-party resellers who, in turn, sell those products to U.S. customers. The above reflects the pricing quoted by one of those third-party resellers. See Appendix B of the Full Disclosure Report for a copy of the third-party reseller's quotation.

Discount Details: The discounts shown are based on the storage capacity purchased and are generally available.

Warranty: Hi-Care Premier On-Site Service include: 7x24 Technical Assistance Center Access. Access to all new software updates and Online Support. 24x7 with 4-hour On-site Hardware Replacement.

Availability Date: Currently available.

Response Time and Throughput Graph



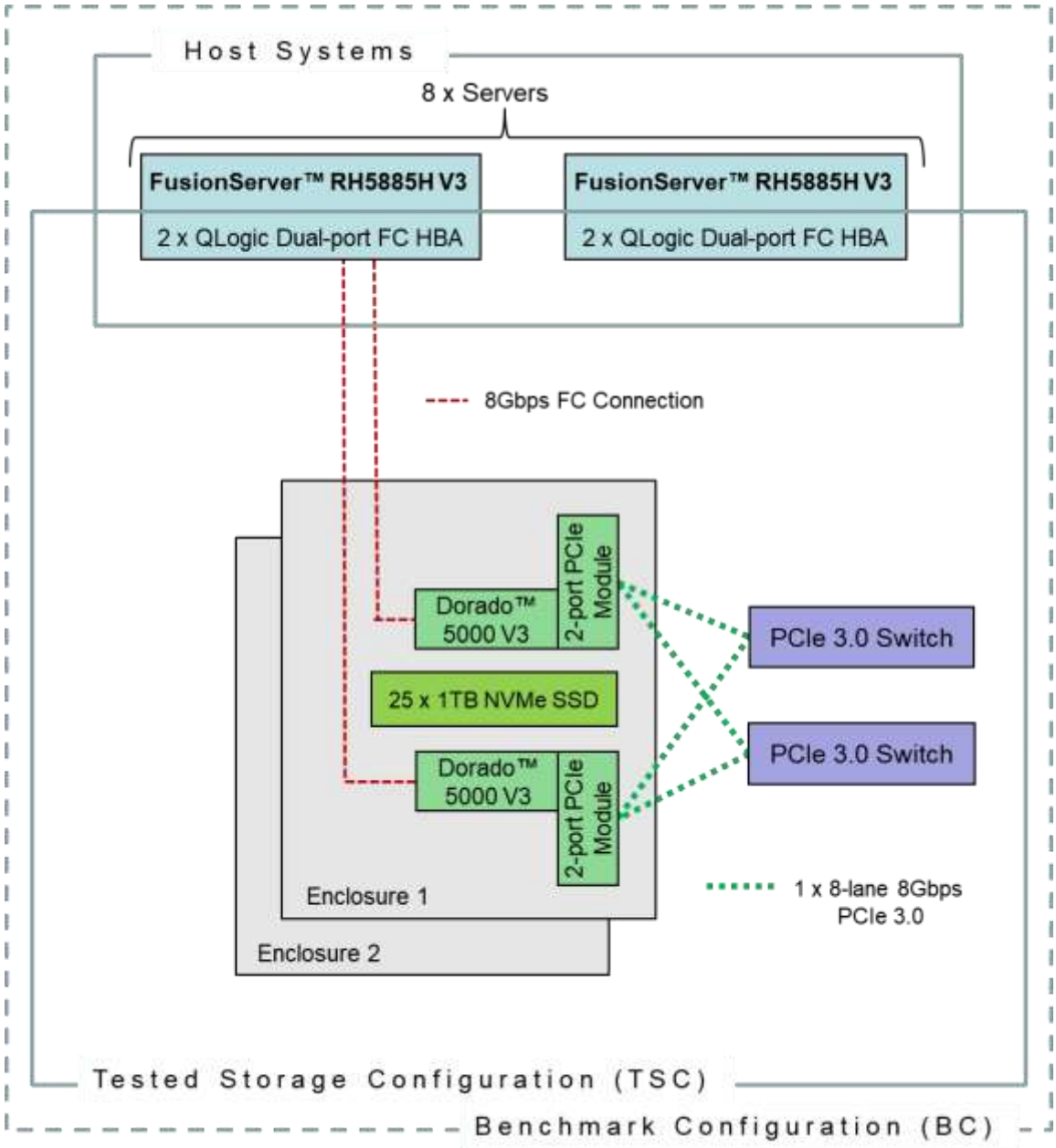
Contact Information	
Test Sponsor Primary Contact	Huawei Technologies Co., Ltd. – www.huawei.com Zhong Xu – xuzhong@huawei.com
SPC Auditor	InfoSizing – www.sizing.com Francois Raab – francois@sizing.com

Revision Information	
SPC Benchmark 1™ Revision	V3.5.0
SPC-1 Workload Generator Revision	V3.0 build d34fb3c
Publication Revision History	• First Edition: December 5, 2017 • Second Edition: February 15, 2018 - Updated SPC-1 Price-Performance™ metric based on SPC-1 v3.6.0 definition.

CONFIGURATION INFORMATION

Benchmark Configuration and Tested Storage Configuration

The following diagram illustrates the Benchmark Configuration (BC), including the Tested Storage Configuration (TSC) and the Host Systems.



Storage Network Configuration

The Tested Storage Configuration (TSC) involved an external storage subsystem made of four Huawei OceanStor™ Dorado5000 V3 Active-Active Storage Controllers, driven by 8 host systems (Huawei FusionServer™ RH5885H V3). Each FusionServer connected one-to-one to each Dorado controller. That connection was established via two dual-port Fibre Chanel HBAs on each FusionServer; and eight ports on each Dorado controller, using four on board ports and a 4-port Smart I/O Module. These Fibre Chanel paths operated at 8Gbps. The four Dorado controllers were interconnected using two 16-port PCIe 3.0 Switches. Each Controller had one 8Gbps 8-lane connection to each Switch.

Host System and Tested Storage Configuration Components

The following table lists the components of the Host System(s) and the Tested Storage Configuration (TSC).

Host Systems
8 x Huawei FusionServer™ RH5885H V3 2 x Intel Xeon E7-4820 V2 (2.0 GHz 8-Core 16MB L3) 256GB Main Memory Red Hat Enterprise Linux 7.1
Tested Storage Configuration
16 x QLogic dual-ported QLE2562 FC HBA
2 x Dorado5000 V3 Enclosures (each with) 2 x Dorado5000 V3 Active-Active Controllers (each with) 256GB cache 1 x 4-port 8Gbps Smart I/O Module 1 x 4*8Gbps FC port on board 1 x 2-port PCIe Module 25 x 1TB NVMe SSDs
2 x 16-port PCIe 2.0 switches

Differences Between Tested and Priced Storage Configurations

There were no differences between the Tested Storage Configuration and the Priced Storage Configuration.

Component Changes in Revised Full Disclosure Report

The following table outlines component changes that were made in revisions to this Full Disclosure Report.

Original Component	Revised Component	Description of Change
n/a	n/a	Initial submission

Benchmark Configuration Creation Process

Customer Tuning Parameters and Options

All the customer tuning parameters and options that have been altered from their default values for this benchmark are included in Appendix C and in the Supporting Files (see Appendix A).

Tested Storage Configuration Creation

A detailed description of how the logical representation of the TSC was created is included in Appendix D and in the Supporting Files (see Appendix A).

Tested Storage Configuration Inventory

An inventory of the components in the TSC, as seen by the Benchmark Configuration, is included in Appendix E and in the Supporting Files (see Appendix A).

Workload Generator Storage Configuration

The SPC-1 Workload Generator storage configuration commands and parameters used to invoke the execution of the tests are included in Appendix F and in the Supporting Files (see Appendix A).

Logical Volume Capacity and ASU Mapping

The following table details the capacity of each ASU and how they are mapped to logical volumes (LV).

	LV per ASU	LV Capacity	Used per LV	Total per ASU	% ASU Capacity
ASU-1	18	676.5	676.5	12,176.2	100.00%
ASU-2	18	676.5	676.5	12,176.2	100.00%
ASU-3	2	1,353.0	1,352.9	2,705.8	22.22%
			SPC-1 ASU Capacity	**27,058.3**	

Physical Storage Capacity and Utilization

The following table details the Physical Capacity of the storage devices and the Physical Capacity Utilization (percentage of Total Physical Capacity used) in support of hosting the ASUs.

Devices	Count	Physical Capacity	Total Capacity
1TB NVMe SSD	50	1015.8	50,791.7
		Total Physical Capacity	**50,791.7**
		Physical Capacity Utilization	**53.27%**

Data Protection

The data protection level used for all logical volumes was **Protected 2**, which was accomplished by configuring the 50 drives into two RAID 6 storage pools of 25 drives each, and by having redundant paths to the storage pools through redundant components. The controller caches were mirrored across controllers and protected against power failures through a battery-backed flushing mechanism.

BENCHMARK EXECUTION RESULTS

This portion of the Full Disclosure Report documents the results of the various SPC-1 Tests, Test Phases, and Test Runs.

Benchmark Execution Overview

Workload Generator Input Parameters

The SPC-1 Workload Generator commands and input parameters for the Test Phases are presented in the Supporting Files (see Appendix A).

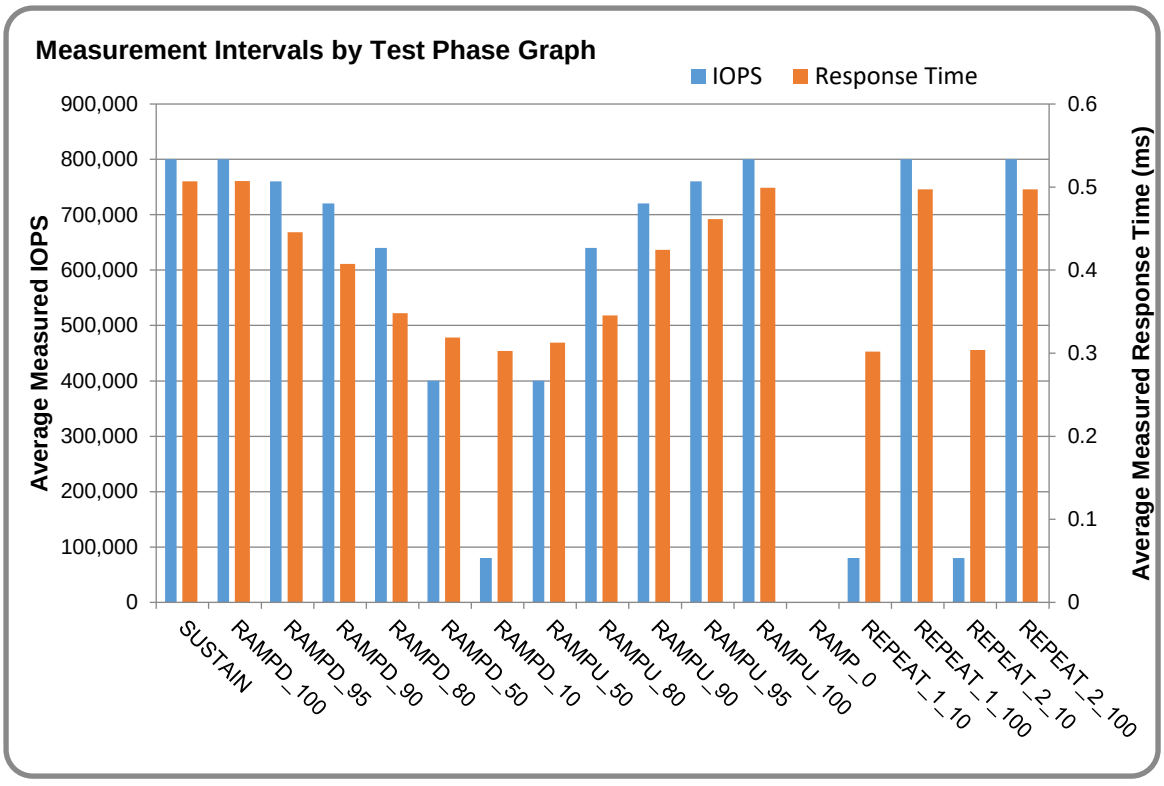
Primary Metrics Test Phases

The benchmark execution consists of the Primary Metrics Test Phases, including the Test Phases SUSTAIN, RAMPD_100 to RAMPD_10, RAMPU_50 to RAMPU_100, RAMP_0, REPEAT_1 and REPEAT_2.

Each Test Phase starts with a transition period followed by a Measurement Interval.

Measurement Intervals by Test Phase Graph

The following graph presents the average IOPS and the average Response Times measured over the Measurement Interval (MI) of each Test Phase.



Exception and Waiver

During the course of the benchmark audit, no exceptions were encountered and no benchmark requirements were waived.

SUSTAIN Test Phase

SUSTAIN – Results File

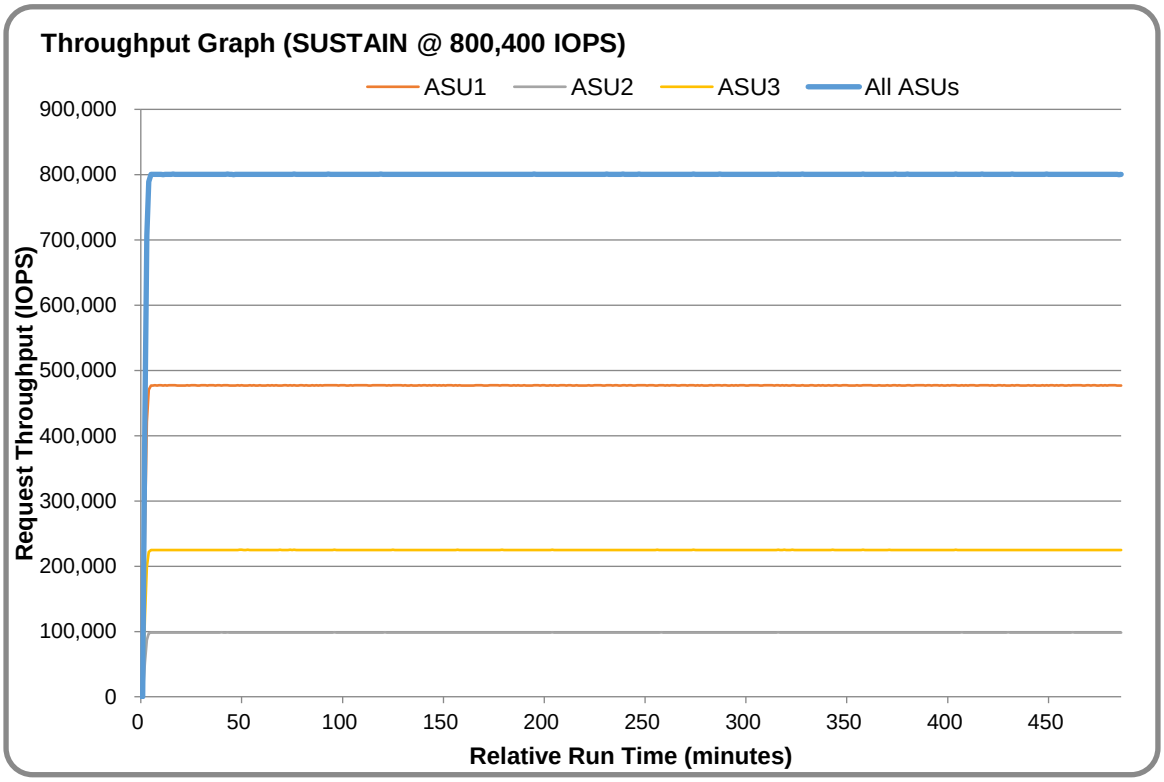
The results file generated during the execution of the SUSTAIN Test Phase is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

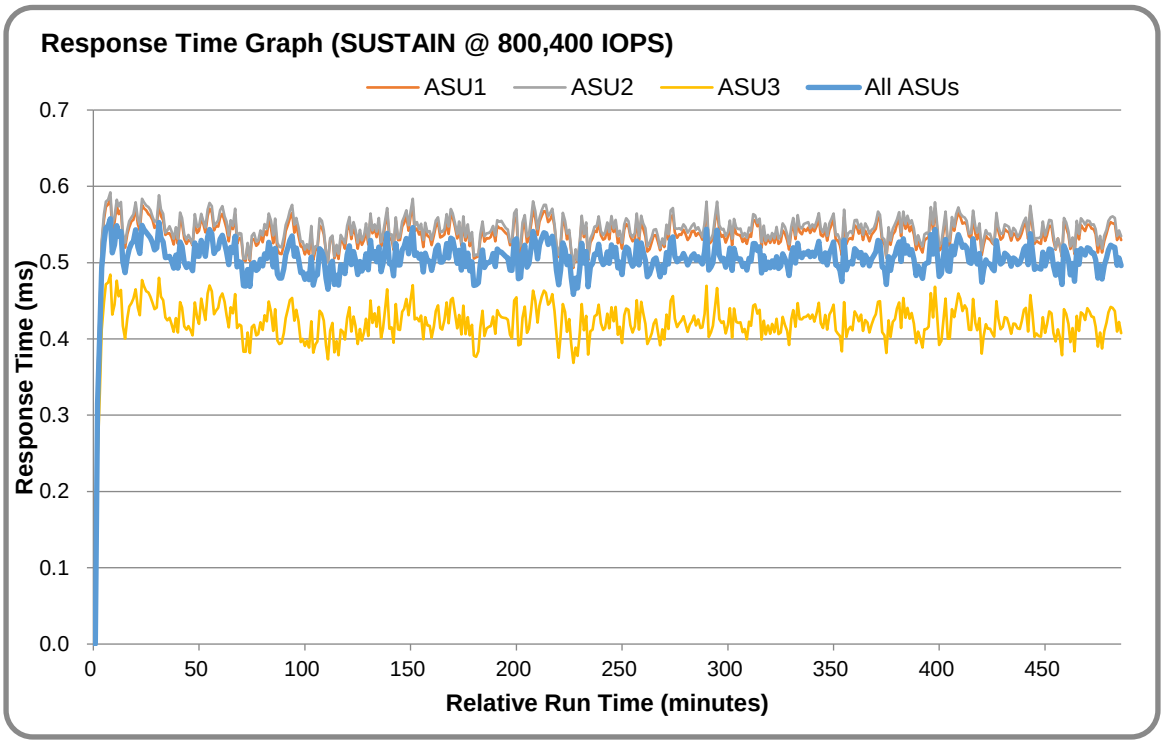
SUSTAIN – Execution Times

Interval	Start Date & Time	End Date &Time	Duration
Transition Period	22-Nov-17 19:13:58	22-Nov-17 19:18:56	0:04:58
Measurement Interval	22-Nov-17 19:18:56	23-Nov-17 03:18:57	8:00:01

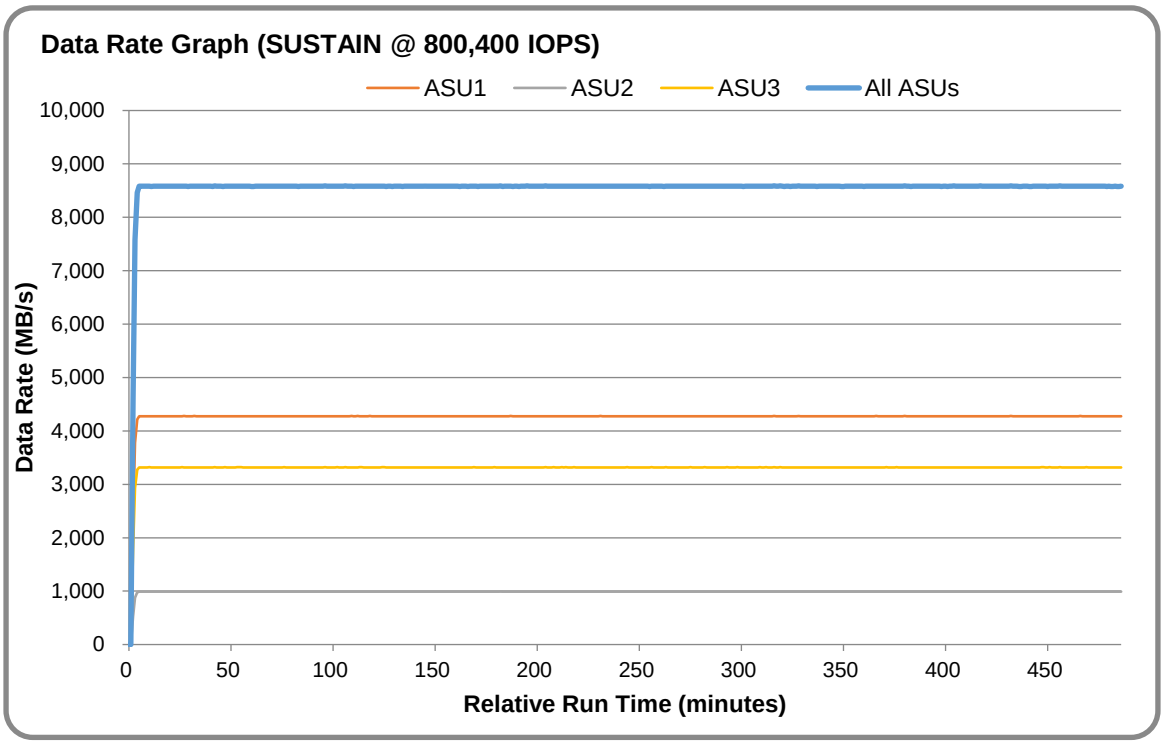
SUSTAIN – Throughput Graph



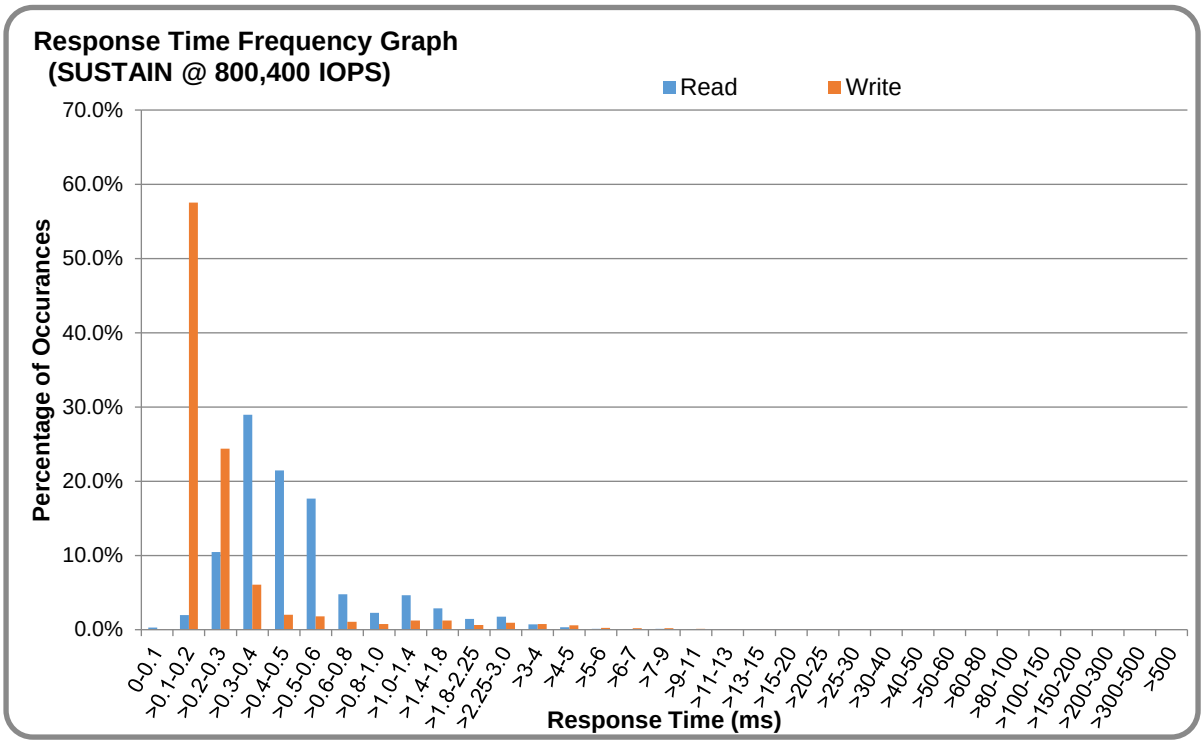
SUSTAIN – Response Time Graph



SUSTAIN – Data Rate Graph



SUSTAIN – Response Time Frequency Graph



SUSTAIN – Intensity Multiplier

The following table lists the targeted intensity multiplier (Defined), the measured intensity multiplier (Measured) for each I/O STREAM, its coefficient of variation (Variation) and the percentage of difference (Difference) between Target and Measured.

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
Defined	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Measured	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Variation	0.0008	0.0002	0.0005	0.0003	0.0011	0.0006	0.0007	0.0002
Difference	0.008%	0.003%	0.001%	0.000%	0.008%	0.004%	0.003%	0.000%

RAMPD_100 Test Phase

RAMPD_100 – Results File

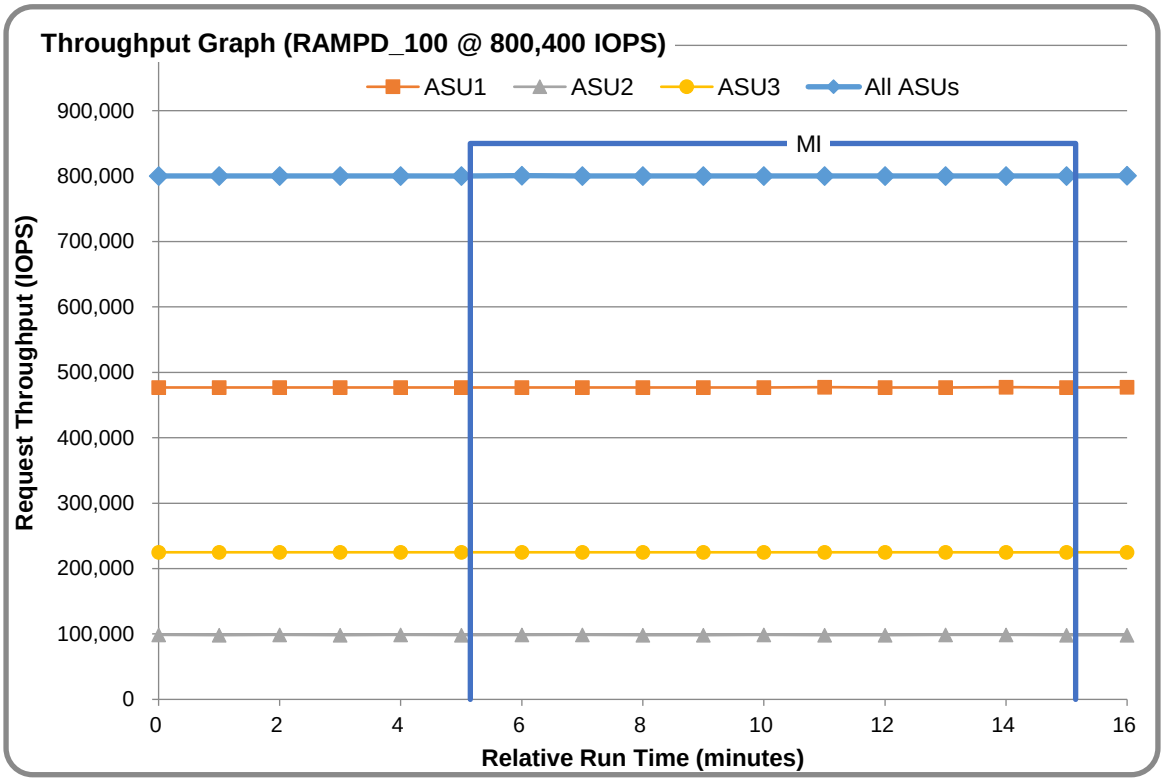
The results file generated during the execution of the RAMPD_100 Test Phase is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

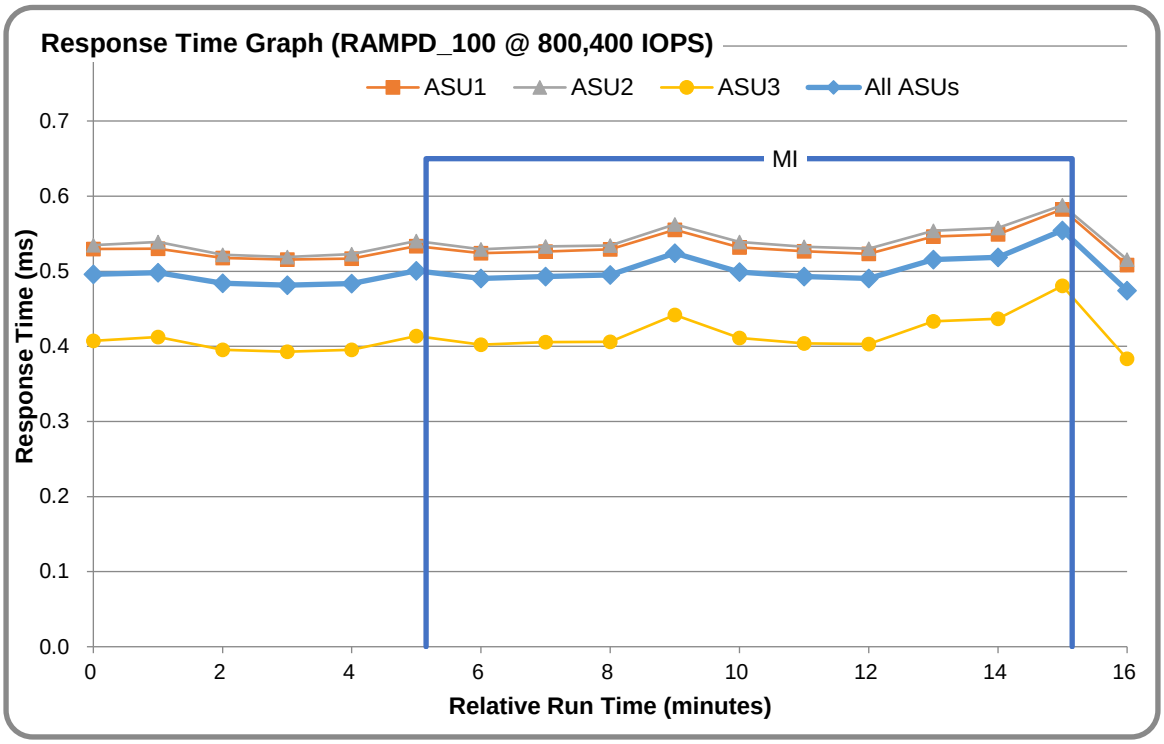
RAMPD_100 – Execution Times

Interval	Start Date & Time	End Date &Time	Duration
Transition Period	23-Nov-17 03:19:56	23-Nov-17 03:24:56	0:05:00
Measurement Interval	23-Nov-17 03:24:56	23-Nov-17 03:34:57	0:10:00

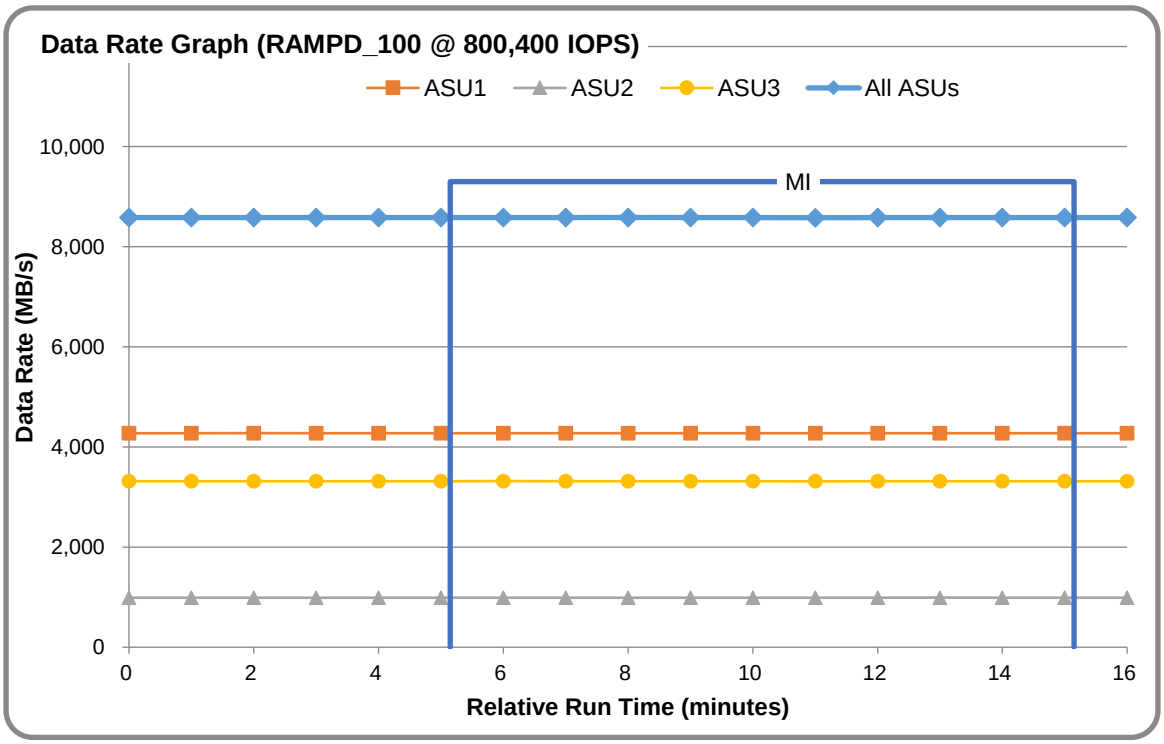
RAMPD_100 – Throughput Graph



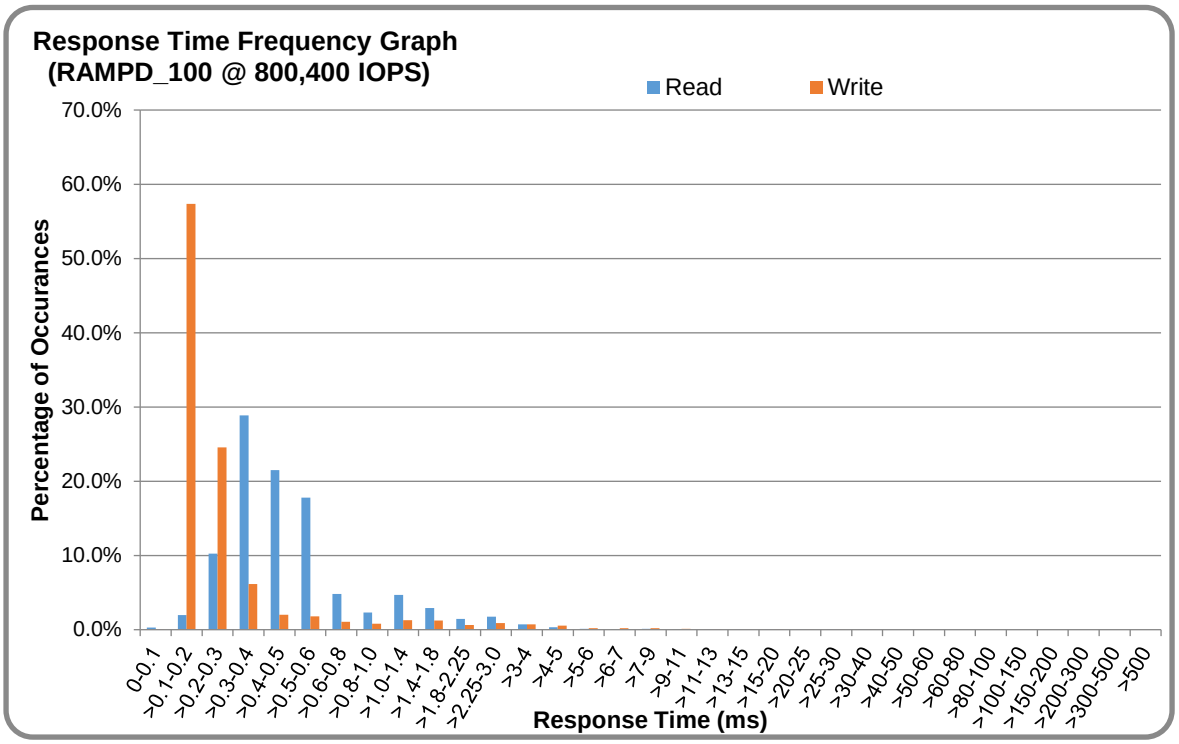
RAMPD_100 – Response Time Graph



RAMPD_100 – Data Rate Graph



RAMPD_100 – Response Time Frequency Graph



RAMPD_100 – Intensity Multiplier

The following table lists the targeted intensity multiplier (Defined), the measured intensity multiplier (Measured) for each I/O STREAM, its coefficient of variation (Variation) and the percentage of difference (Difference) between Target and Measured.

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
Defined	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Measured	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Variation	0.0005	0.0003	0.0006	0.0004	0.0005	0.0005	0.0009	0.0002
Difference	0.025%	0.009%	0.022%	0.013%	0.013%	0.004%	0.008%	0.006%

RAMPD_100 – I/O Request Summary

I/O Requests Completed in the Measurement Interval	480,278,751
I/O Requests Completed with Response Time <= 30 ms	480,277,565
I/O Requests Completed with Response Time > 30 ms	1,186

Response Time Ramp Test

Response Time Ramp Test – Results File

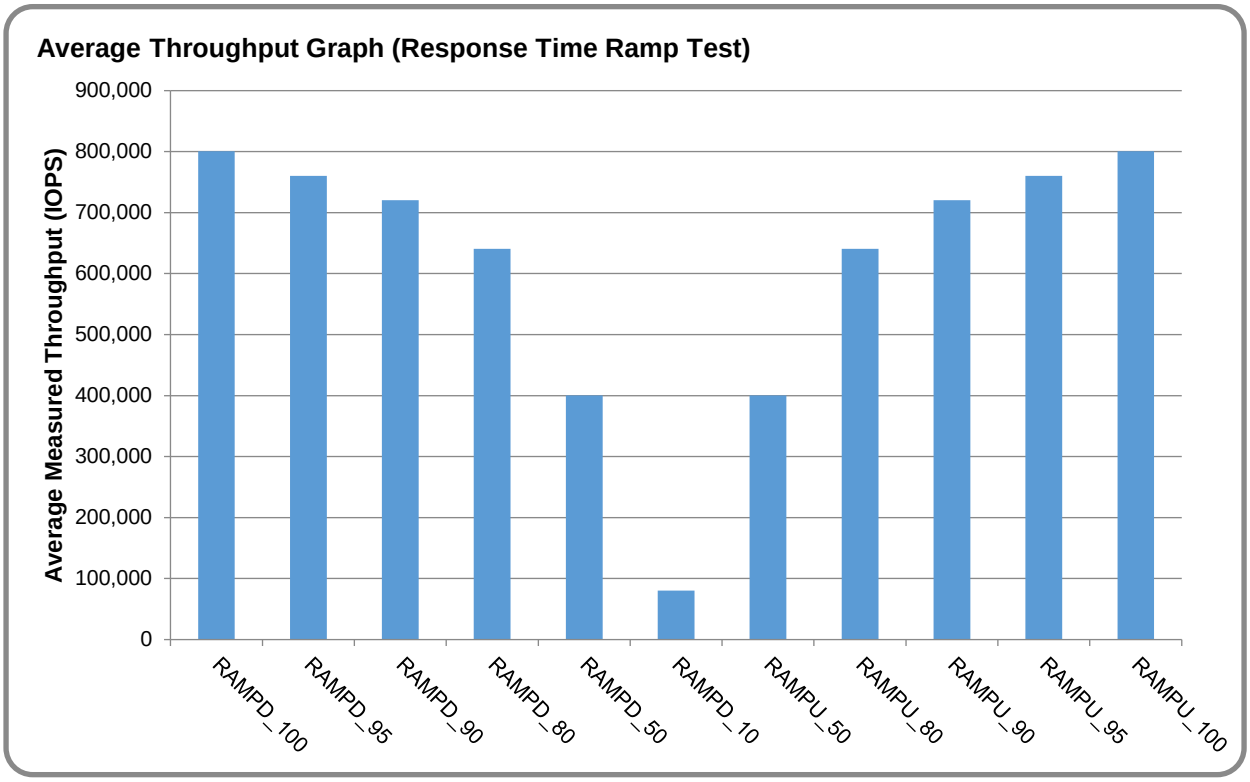
The results file generated during the execution of the Response Time Ramp Test is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

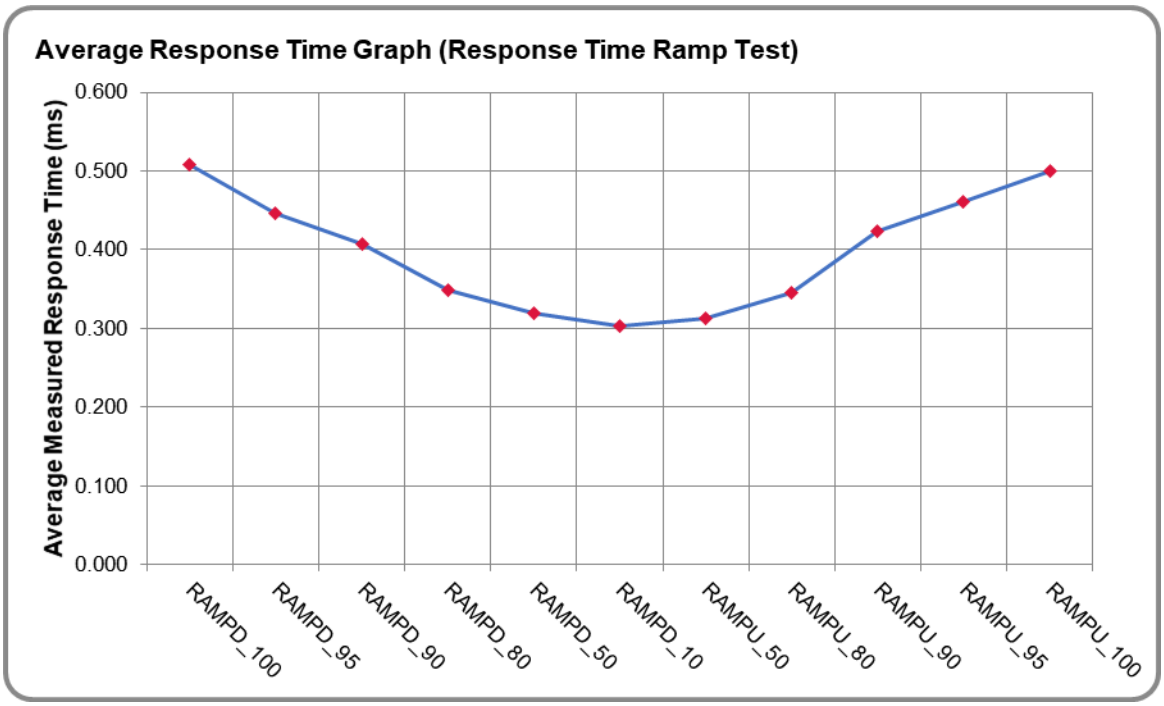
Response Time Ramp Test – Phases

The Response Time Ramp Test is comprised of 11 Test Phases, including six Ramp-Down Phases (executed at 100%, 95%, 90%, 80%, 50%, and 10% of the Business Scaling Unit) and five Ramp-Up Phases (executed at 50%, 80%, 90%, 95%, and 100% of the Business Scaling Unit).

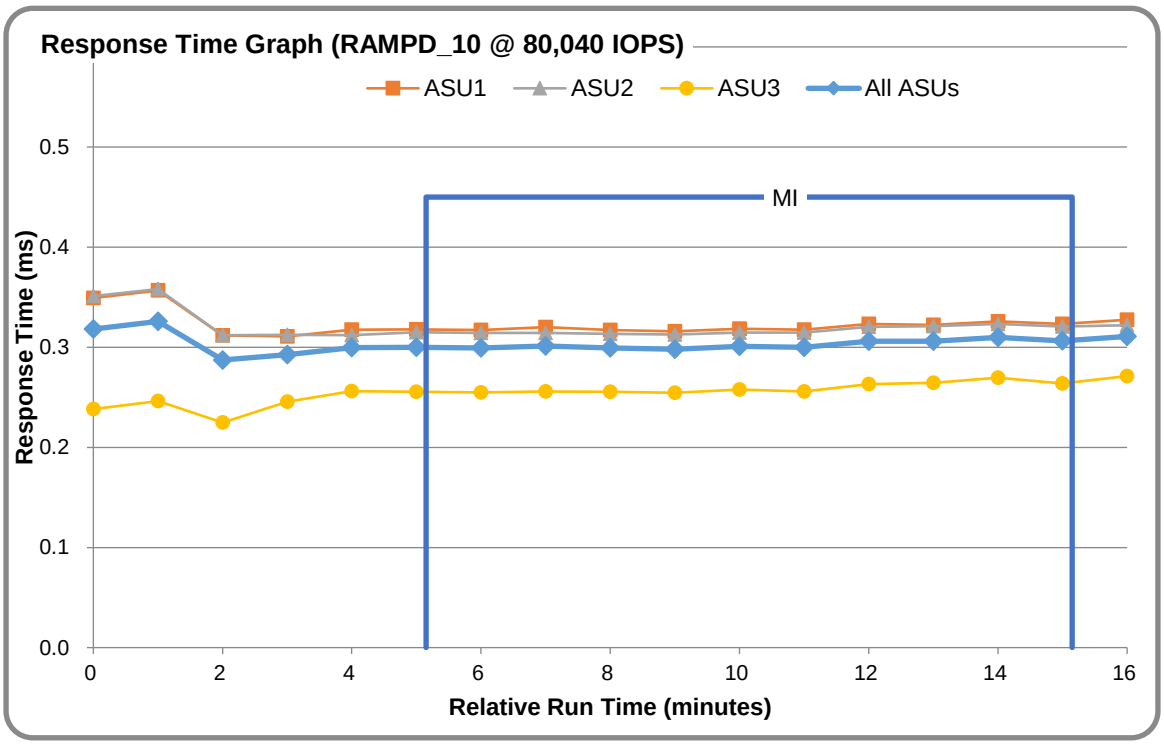
Response Time Ramp Test – Average Throughput Graph



Response Time Ramp Test – Average Response Time Graph



Response Time Ramp Test – RAMPD_10 Response Time Graph



Repeatability Test

Repeatability Test Results File

The results file generated during the execution of the Repeatability Test is included in the Supporting Files (see Appendix A) as follows:

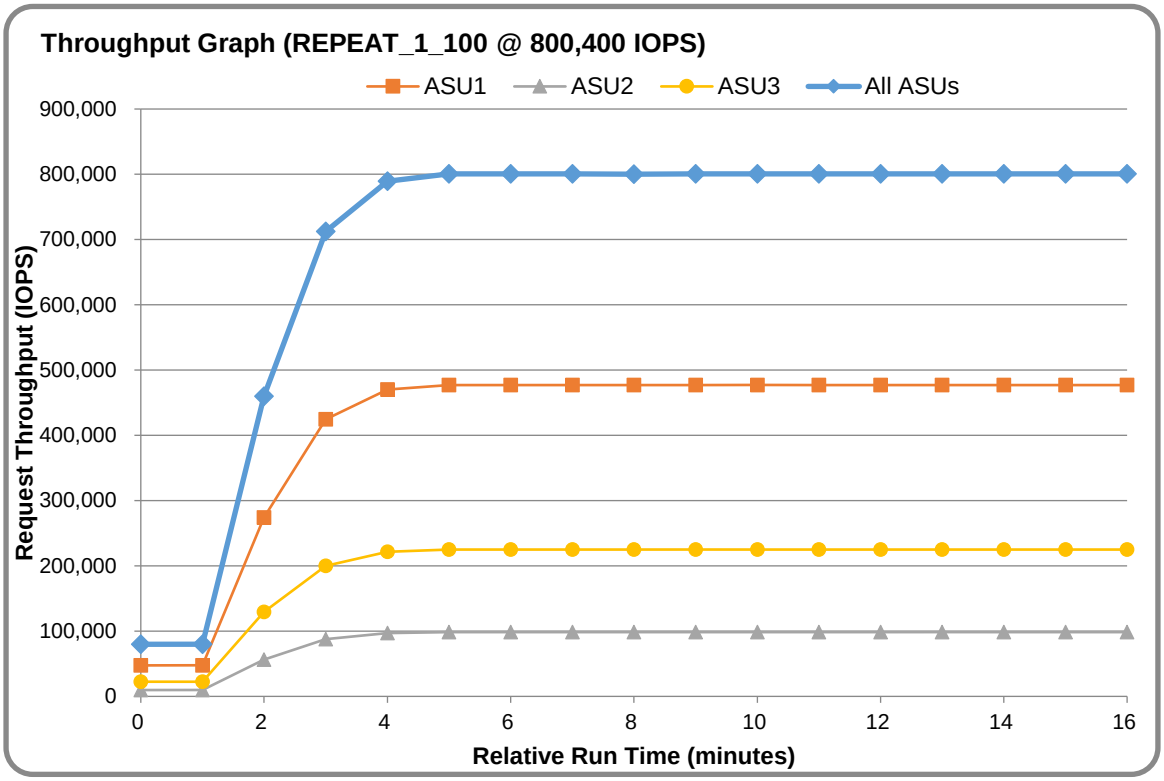
- **SPC1_METRICS_0_Raw_Results.xlsx**

Repeatability Test Results

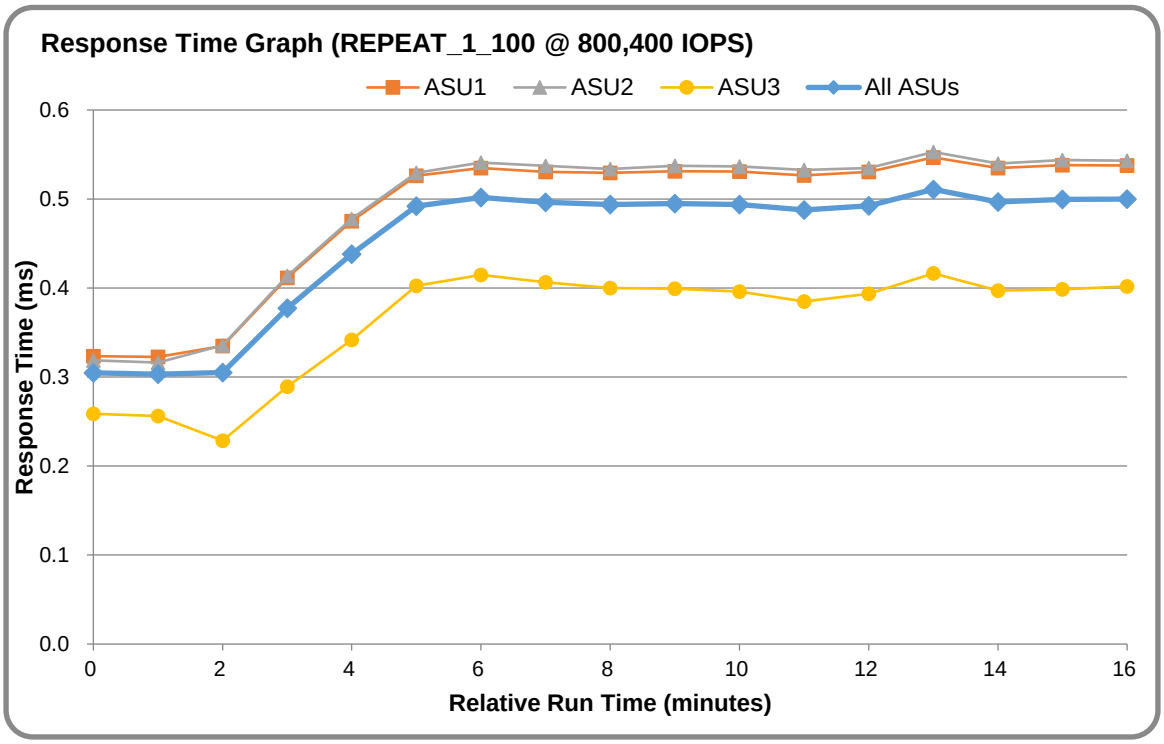
The throughput measurements for the Response Time Ramp Test (RAMPD) and the Repeatability Test Phases (REPEAT_1 and REPEAT_2) are listed in the tables below.

Test Phase	100% IOPS	10% IOPS
RAMPD	800,465.6	80,061.2
REPEAT_1	800,495.0	80,078.6
REPEAT_2	800,425.7	80,051.0

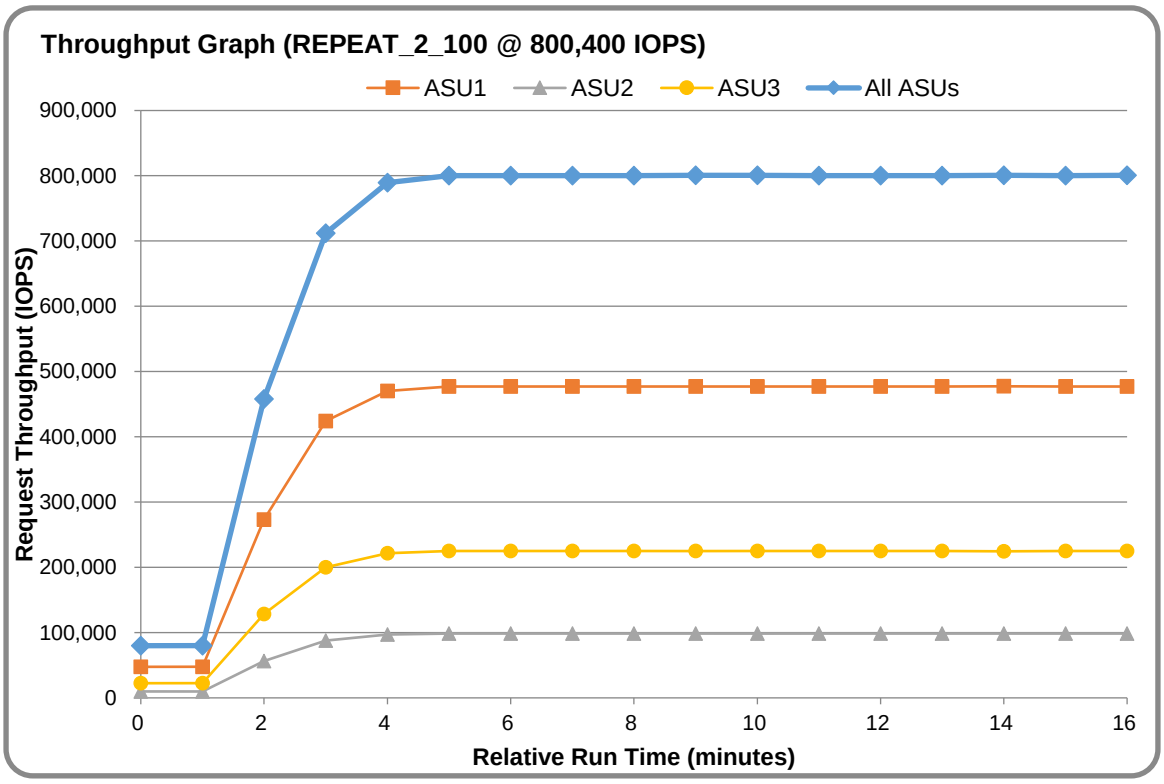
REPEAT_1_100 – Throughput Graph



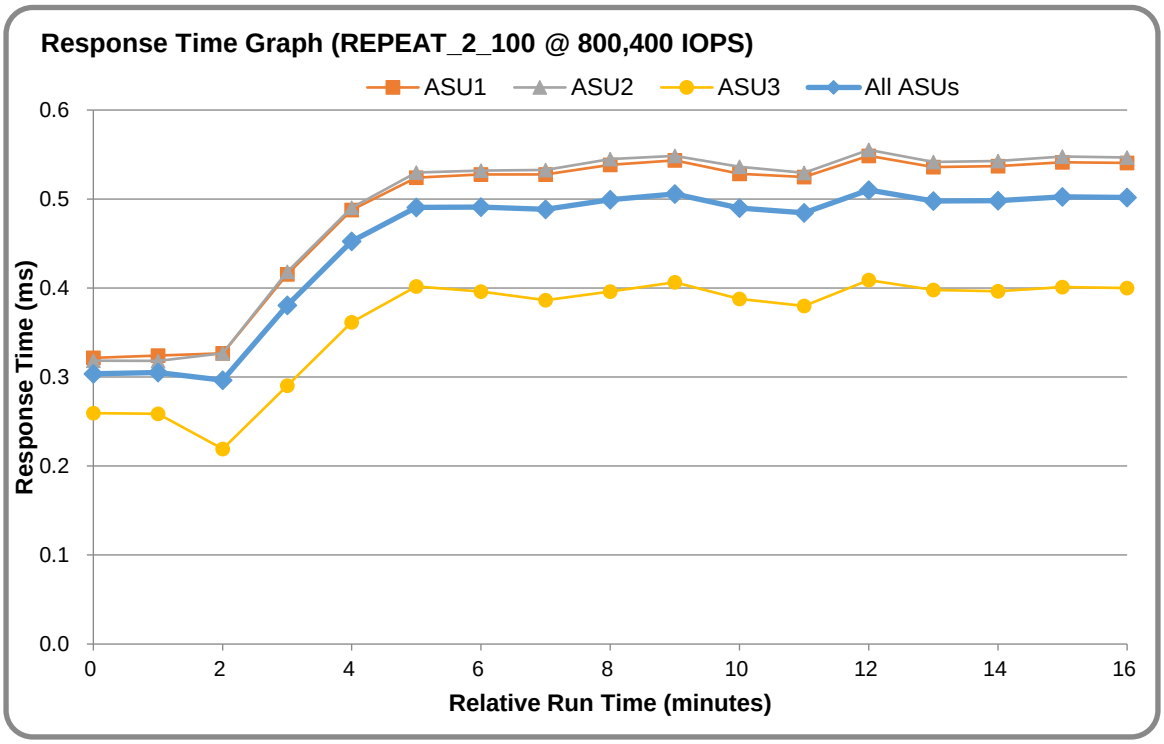
REPEAT_1_100 – Response Time Graph



REPEAT_2_100 – Throughput Graph



REPEAT_2_100 – Response Time Graph



Repeatability Test – Intensity Multiplier

The following tables lists the targeted intensity multiplier (Defined), the measured intensity multiplier (Measured) for each I/O STREAM, its coefficient of variation (Variation) and the percent of difference (Difference) between Target and Measured.

REPEAT_1_100 Test Phase

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
Defined	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Measured	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Variation	0.0005	0.0002	0.0005	0.0002	0.0008	0.0006	0.0007	0.0002
Difference	0.048%	0.010%	0.017%	0.013%	0.001%	0.014%	0.028%	0.005%

REPEAT_2_100 Test Phase

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
Defined	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Measured	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Variation	0.0008	0.0003	0.0007	0.0002	0.0009	0.0005	0.0009	0.0003
Difference	0.049%	0.008%	0.011%	0.021%	0.022%	0.026%	0.026%	0.010%

Data Persistence Test

Data Persistence Test Result files

The results files generated during the execution of the Data Persistence Test is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_PERSIST_1_0_Raw_Results.xlsx**
- **SPC1_PERSIST_2_0_Raw_Results.xlsx**

Data Persistence Test Execution

The Data Persistence Test was executed using the following sequence of steps:

- The PERSIST_1_0 Test Phase was executed to completion.
- The Benchmark Configuration was taken through an orderly shutdown process and powered off.
- The Benchmark Configuration was powered on and taken through an orderly startup process.
- The PERSIST_2_0 Test Phase was executed to completion.

Data Persistence Test Results

Data Persistence Test Phase: Persist1	
Total Number of Logical Blocks Written	98,951,000
Total Number of Logical Blocks Verified	52,120,082
Total Number of Logical Blocks Overwritten	46,830,918
Total Number of Logical Blocks that Failed Verification	0
Time Duration for Writing Test Logical Blocks (sec.)	301
Size in Bytes of each Logical Block	8,192
Number of Failed I/O Requests During the Test	0

Committed Data Persistence Implementation

The persistency of committed data is implemented at two levels. At the disk level, data loss is prevented through the use of RAID 6 arrays. At the controller level, all caches are mirrored across controllers, where write requests are only completed once the local cache has been successfully mirrored in another controller's cache. In addition, cache content is protected from a loss of power by flushing the cache content to permanent flash memory, as soon as a power loss is detected. The flushing action is powered by a battery backup located in each controller.

APPENDIX A: SUPPORTING FILES

The following table details the content of the Supporting Files provided as part of this Full Disclosure Report.

File Name	Description	Location
/SPC1_RESULTS	**Data reduction worksheets**	**root**
SPC1_INIT_0_Raw_Results.xlsx	Raw results for INIT Test Phase	/SPC1_RESULTS
SPC1_METRICS_0_Quick_Look.xlsx	Quick Look Test Run Overview	/SPC1_RESULTS
SPC1_METRICS_0_Raw_Results.xlsx	Raw results for Primary Metrics Test	/SPC1_RESULTS
SPC1_METRICS_0_Summary_Results.xlsx	Primary Metrics Summary	/SPC1_RESULTS
SPC1_PERSIST_1_0_Raw_Results.xlsx	Raw results for PERSIST1 Test Phase	/SPC1_RESULTS
SPC1_PERSIST_2_0_Raw_Results.xlsx	Raw results for PERSIST2 Test Phase	/SPC1_RESULTS
SPC1_Run_Set_Overview.xlsx	Run Set Overview Worksheet	/SPC1_RESULTS
SPC1_VERIFY_0_Raw_Results.xlsx	Raw results for first VERIFY Test Phase	/SPC1_RESULTS
SPC1_VERIFY_1_Raw_Results.xlsx	Raw results for second VERIFY Test Phase	/SPC1_RESULTS
/C_Tuning	**Tuning parameters and options**	**root**
aio-max-nr.sh	Set maximum asynchronous I/O	/C_Tuning
nr_requests.sh	Increase disk queue depth	/C_Tuning
scheduler.sh	Change the I/O scheduler	/C_Tuning
/D_Creation	**Storage configuration creation**	**root**
mklun.txt	Create the storage environment	/D_Creation
mkvolume.sh	Create the Logical Volumes	/D_Creation
/E_Inventory	**Configuration inventory**	**root**
shstorage.tcl	Captures profile of storage environment	/E_Inventory
profile1_volume.log	List of logical volumes before INIT	/E_Inventory
profile1_storage.log	List of storage devices before INIT	/E_Inventory
profile2_volume.log	List of logical volumes after restart	/E_Inventory
profile2_storage.log	List of storage devices after restart	/E_Inventory
/F_Generator	**Workload generator**	**root**
slave_asu.asu	Defining LUNs hosting the ASUs	/F_generator
8host.HST	Host configuration file	/F_generator
full_run.sh	Executing all test phases	/F_generator

APPENDIX B: THIRD PARTY QUOTATION

Reseller: Noviant

Address: 32 Broadway, Suite 401
New York, NY 10004
Tel: 212-809-6625
Email: sales@noviant.com

11/27/2017, Quote Valid:90 Days

No.	Model	Description	Qty.	Unit Price (USD)	Ext.Price (USD)	Disc. (off)	Disc. Price (USD)
1	Phase						
1.1	Location						
1.1.1	OceanStor Dorado5000 V3 Main Equipment						
1.1.1	Controller Enclosure						
	D5V3-512G-NAC-8	Dorado5000 V3(2U,Dual Ctrl,NVMe,AC\240HVDC,512GB Cache,SmartIO,16*8Gb FC,25*2.5",SPE61C0225)	2	123,590.82	247,181.64	45.00%	135,949.90
1.1.2	Expanding Interface Module						
	DV3-LPU5PCIE	8G PCIE Interface Board(with two NT Ports)	4	1,929.20	7,716.80	45.00%	4,244.24
1.1.2	Disk Components						
	D5V3-SSD-NVME-1T	1TB SSD NVMe Disk Unit(2.5")	50	4,971.40	248,570.00	45.00%	136,713.50
1.1.4	HBA						
	N8GHBA000	QLOGIC QLE2562 HBA Card,PCIE,8Gbps DualPort ,Fiber Channel Multimode LC Optic Interface,English Manual, No Drive CD	16	1,698.00	27,168.00	0.00%	27,168.00
1.1.5	Accessory						
	SN2F01FCPC	Patch Cord,DLC/PC,DLC/PC,Multi-mode,3m,A1a.2,2mm,42mm DLC,OM3 bending insensitive	32	19.80	633.60	0.00%	633.60
	DV3-PCIESWITCH3	PCIe 3.0 Switch(AC\240HVDC,8GB Cache,16 Port,SWE1600P08)	2	16,218.00	32,436.00	0.00%	32,436.00
	AOC-QSFP41G	Quadwire 40 Gb/s Parallel AOC for PCIe 3.0	8	1,065.30	8,522.40	0.00%	8,522.40
1.1.6	Storage Software						
	D5V3-LBS	Basic Software License(Include DeviceManager,SmartThin,SmartMigration,SmartDedupe,SmartCompression,eService,SystemReporter,UltraPath)	1	7,196.00	7,196.00	44.00%	4,029.76
Total of Product							349,697.40
1.1.7	Maintenance Support Service						
	02351GSA-88134ULF-36	Dorado5000 V3(2U,Dual Ctrl,NVMe,AC\240HVDC,512GB Cache,SmartIO,8*8Gb FC,25*2.5",SPE61C0225)-Hi-Care Onsite Premier 24x7x4H Engineer Onsite Service-36Month(s)	2	4,446.00	8,892.00	10.00%	8,002.80
	02351GTY-88134ULF-36	1TB SSD NVMe Disk Unit(2.5")-Hi-Care Onsite Premier 24x7x4H Engineer Onsite Service-36Month(s)	50	216.00	10,800.00	10.00%	9,720.00
	88033NAM-88134UHK-36	Basic Software License(Include DeviceManager,SmartThin,SmartMigration,SmartDedupe,SmartCompression,eService,SystemReporter)-Hi-Care Application Software Upgrade Support Service,UltraPath Software License-36Month(s)	1	1,854.00	1,854.00	30.00%	1,297.80
	8812102575	OceanStor Dorado5000 V3 Installation Service - Engineering	1	9,620.68	9,620.68	10.00%	8,658.61
Total of Service (3 years)							27,679.21
Total Price							377,376.61
Notes:Hi-Care Premier On-Site Service include: 7*24 Technical Assistance Center Access. Access to all new software updates and Online Support. 24*7*4 Hours Onsite Hardware Replacement.							

Payment Terms: Comments:
Noviant is an Authorized Value Added reseller (VAR) of networking products. Products sold by NF are factory new unless otherwise specified. All new products sold by NF carry its own Original Equipment Manufacturer's (OEM) Limited Warranty and software licenses. This Quote is valid for 90 days. Prices and availability is subject to change without notice. Installation and configuration costs are not included in the quoted pricing unless specified. A 20% Restocking Fee applies to all cancelled orders and/or returned products. Special Orders are non-returnable. Buyer is responsible for payment of all applicable taxes and freight charges. Issuance of customer PO against this Quote constitutes acceptance of Noviant Sales Terms conditions.
I agree to the these terms and conditions.
Authorized Acceptance:_____ Print Name:_____ Date:_____
Noviant:_____ Print Name:_____ Date:_____

APPENDIX C: TUNING PARAMETERS AND OPTIONS

The following scripts, listed below, were used to set tuning parameters and options:

- ***aio-max-nr.sh*** to set the maximum asynchronous I/O
- ***nr_requests.sh*** to change the I/O scheduler
- ***scheduler.sh*** to increase the disk queue depth

The scripts described above are included in the Supporting Files (see Appendix A) and listed below.

aio-max-nr.sh

```
echo 1048576 > /proc/sys/fs/aio-max-nr
```

nr_requests.sh

```
echo 1024 > /sys/block/sdb/queue/nr_requests
echo 1024 > /sys/block/sdc/queue/nr_requests
echo 1024 > /sys/block/sdd/queue/nr_requests
echo 1024 > /sys/block/sde/queue/nr_requests
echo 1024 > /sys/block/sdf/queue/nr_requests
echo 1024 > /sys/block/sdg/queue/nr_requests
echo 1024 > /sys/block/sdh/queue/nr_requests
echo 1024 > /sys/block/sdi/queue/nr_requests
echo 1024 > /sys/block/sdj/queue/nr_requests
echo 1024 > /sys/block/sdk/queue/nr_requests
echo 1024 > /sys/block/sdl/queue/nr_requests
echo 1024 > /sys/block/sdm/queue/nr_requests
echo 1024 > /sys/block/sdn/queue/nr_requests
echo 1024 > /sys/block/sdo/queue/nr_requests
echo 1024 > /sys/block/sdp/queue/nr_requests
echo 1024 > /sys/block/sdq/queue/nr_requests
echo 1024 > /sys/block/sdr/queue/nr_requests
echo 1024 > /sys/block/sds/queue/nr_requests
echo 1024 > /sys/block/sdt/queue/nr_requests
echo 1024 > /sys/block/sdu/queue/nr_requests
echo 1024 > /sys/block/sdv/queue/nr_requests
echo 1024 > /sys/block/sdw/queue/nr_requests
echo 1024 > /sys/block/sdx/queue/nr_requests
echo 1024 > /sys/block/sdy/queue/nr_requests
echo 1024 > /sys/block/sdz/queue/nr_requests
echo 1024 > /sys/block/sdaa/queue/nr_requests
echo 1024 > /sys/block/sdab/queue/nr_requests
echo 1024 > /sys/block/sdac/queue/nr_requests
echo 1024 > /sys/block/sdad/queue/nr_requests
echo 1024 > /sys/block/sdae/queue/nr_requests
echo 1024 > /sys/block/sdaf/queue/nr_requests
echo 1024 > /sys/block/sdag/queue/nr_requests
echo 1024 > /sys/block/sdah/queue/nr_requests
echo 1024 > /sys/block/sdai/queue/nr_requests
echo 1024 > /sys/block/sdaj/queue/nr_requests
echo 1024 > /sys/block/sdak/queue/nr_requests
echo 1024 > /sys/block/sdal/queue/nr_requests
```

```
echo 1024 > /sys/block/sdam/queue/nr_requests
echo 1024 > /sys/block/sdan/queue/nr_requests
echo 1024 > /sys/block/sdao/queue/nr_requests
echo 1024 > /sys/block/sdap/queue/nr_requests
echo 1024 > /sys/block/sdaq/queue/nr_requests
echo 1024 > /sys/block/sdar/queue/nr_requests
echo 1024 > /sys/block/sdas/queue/nr_requests
echo 1024 > /sys/block/sdat/queue/nr_requests
echo 1024 > /sys/block/sdau/queue/nr_requests
echo 1024 > /sys/block/sdav/queue/nr_requests
echo 1024 > /sys/block/sdaw/queue/nr_requests
echo 1024 > /sys/block/sdax/queue/nr_requests
echo 1024 > /sys/block/sday/queue/nr_requests
echo 1024 > /sys/block/sdaz/queue/nr_requests
echo 1024 > /sys/block/sdba/queue/nr_requests
echo 1024 > /sys/block/sdbb/queue/nr_requests
echo 1024 > /sys/block/sdbc/queue/nr_requests
echo 1024 > /sys/block/sdbd/queue/nr_requests
echo 1024 > /sys/block/sdbe/queue/nr_requests
echo 1024 > /sys/block/sdbf/queue/nr_requests
echo 1024 > /sys/block/sdbg/queue/nr_requests
echo 1024 > /sys/block/sdbh/queue/nr_requests
echo 1024 > /sys/block/sdbi/queue/nr_requests
echo 1024 > /sys/block/sdbj/queue/nr_requests
echo 1024 > /sys/block/sdbk/queue/nr_requests
echo 1024 > /sys/block/sdbl/queue/nr_requests
echo 1024 > /sys/block/sdbm/queue/nr_requests
echo 1024 > /sys/block/sdbn/queue/nr_requests
echo 1024 > /sys/block/sdbo/queue/nr_requests
echo 1024 > /sys/block/sdbp/queue/nr_requests
echo 1024 > /sys/block/sdbq/queue/nr_requests
echo 1024 > /sys/block/sdbr/queue/nr_requests
echo 1024 > /sys/block/sdbs/queue/nr_requests
echo 1024 > /sys/block/sdbt/queue/nr_requests
echo 1024 > /sys/block/sdbu/queue/nr_requests
echo 1024 > /sys/block/sdbv/queue/nr_requests
echo 1024 > /sys/block/sdbw/queue/nr_requests
echo 1024 > /sys/block/sdbx/queue/nr_requests
echo 1024 > /sys/block/sdby/queue/nr_requests
echo 1024 > /sys/block/sdbz/queue/nr_requests
echo 1024 > /sys/block/sdca/queue/nr_requests
echo 1024 > /sys/block/sdcb/queue/nr_requests
echo 1024 > /sys/block/sdcc/queue/nr_requests
echo 1024 > /sys/block/sdcd/queue/nr_requests
echo 1024 > /sys/block/sdce/queue/nr_requests
echo 1024 > /sys/block/sdcf/queue/nr_requests
echo 1024 > /sys/block/sdcg/queue/nr_requests
echo 1024 > /sys/block/sdch/queue/nr_requests
echo 1024 > /sys/block/sdci/queue/nr_requests
echo 1024 > /sys/block/sdcj/queue/nr_requests
echo 1024 > /sys/block/sdck/queue/nr_requests
echo 1024 > /sys/block/sdcl/queue/nr_requests
echo 1024 > /sys/block/sdcm/queue/nr_requests
echo 1024 > /sys/block/sdcn/queue/nr_requests
echo 1024 > /sys/block/sdco/queue/nr_requests
echo 1024 > /sys/block/sdcp/queue/nr_requests
echo 1024 > /sys/block/sdcq/queue/nr_requests
echo 1024 > /sys/block/sdcr/queue/nr_requests
echo 1024 > /sys/block/sdcs/queue/nr_requests
echo 1024 > /sys/block/sdct/queue/nr_requests
echo 1024 > /sys/block/sdcu/queue/nr_requests
```

```
echo 1024 > /sys/block/sdcv/queue/nr_requests
echo 1024 > /sys/block/sdcw/queue/nr_requests
echo 1024 > /sys/block/sdcx/queue/nr_requests
echo 1024 > /sys/block/sdcy/queue/nr_requests
echo 1024 > /sys/block/sdcz/queue/nr_requests
echo 1024 > /sys/block/sdda/queue/nr_requests
echo 1024 > /sys/block/sddb/queue/nr_requests
echo 1024 > /sys/block/sddc/queue/nr_requests
echo 1024 > /sys/block/sddd/queue/nr_requests
echo 1024 > /sys/block/sdde/queue/nr_requests
echo 1024 > /sys/block/sddf/queue/nr_requests
echo 1024 > /sys/block/sddg/queue/nr_requests
echo 1024 > /sys/block/sddh/queue/nr_requests
echo 1024 > /sys/block/sddj/queue/nr_requests
echo 1024 > /sys/block/sddi/queue/nr_requests
echo 1024 > /sys/block/sddk/queue/nr_requests
echo 1024 > /sys/block/sddl/queue/nr_requests
echo 1024 > /sys/block/sddn/queue/nr_requests
echo 1024 > /sys/block/sddm/queue/nr_requests
echo 1024 > /sys/block/sddo/queue/nr_requests
echo 1024 > /sys/block/sddp/queue/nr_requests
echo 1024 > /sys/block/sddq/queue/nr_requests
echo 1024 > /sys/block/sddr/queue/nr_requests
echo 1024 > /sys/block/sdds/queue/nr_requests
echo 1024 > /sys/block/sddu/queue/nr_requests
echo 1024 > /sys/block/sddt/queue/nr_requests
echo 1024 > /sys/block/sddw/queue/nr_requests
echo 1024 > /sys/block/sddx/queue/nr_requests
echo 1024 > /sys/block/sddv/queue/nr_requests
echo 1024 > /sys/block/sddy/queue/nr_requests
```

- ***scheduler.sh***

```
#!/bin/bash

echo noop > /sys/block/sdb/queue/scheduler
echo noop > /sys/block/sdc/queue/scheduler
echo noop > /sys/block/sdd/queue/scheduler
echo noop > /sys/block/sde/queue/scheduler
echo noop > /sys/block/sdf/queue/scheduler
echo noop > /sys/block/sdg/queue/scheduler
echo noop > /sys/block/sdh/queue/scheduler
echo noop > /sys/block/sdi/queue/scheduler
echo noop > /sys/block/sdj/queue/scheduler
echo noop > /sys/block/sdk/queue/scheduler
echo noop > /sys/block/sdl/queue/scheduler
echo noop > /sys/block/sdm/queue/scheduler
echo noop > /sys/block/sdn/queue/scheduler
echo noop > /sys/block/sdo/queue/scheduler
echo noop > /sys/block/sdp/queue/scheduler
echo noop > /sys/block/sdq/queue/scheduler
```

APPENDIX D: STORAGE CONFIGURATION CREATION

Environment

First, the CLI commands from the following command file are copied and pasted into the Dorado5000 V3 CLI window. These commands are executed on one of the Host Systems.

- ***mklun.txt***

Next, the following shell script is executed on one of the Host Systems.

- ***mkvolume.sh***

The command files described above are included in the Supporting Files (see Appendix A) and listed below.

Following are the detailed steps involved in creating the storage configuration:

Step 1 - Create Disk Domains, Storage Pools, LUNs

The ***mklun.txt*** command file, listed below, includes all the CLI commands to perform the following actions:

- Create 2 disk domains
- Create 2 storage pools
- Create 16 LUNs
- Create one LUN group
- Add the 16 LUNs to the LUN group

Step 2 - Create Mapping View, Host Group and Host

The ***mklun.txt*** command file, listed below, includes all the CLI commands to perform the following actions:

- Create 1 mapping view
- Create 1 host group
- Create 8 hosts
- Add the 8 hosts to the host group
- Add the host group and the LUN group to the mapping view
- Add the Fibre Chanel ports to the hosts

Step 3 - Create Volumes on the Master Host System

The ***mkvolume.sh*** script, listed below, is executed on the Master Host System to perform the following actions:

- Create 16 physical volumes
- Create 1 volume group
- Create 38 logical volumes (18 for ASU-1, 18 for ASU-2 and 2 for ASU-3)

mklun.txt

```
create disk_domain name=dd0 disk_list=all controller_enclosure_list=CTE0
   disk_domain_id=0
create disk_domain name=dd1 disk_list=all controller_enclosure_list=CTE1
   disk_domain_id=1
create storage_pool name=pool0 capacity=15554GB disk_domain_id=0 raid_level=RAID6
   pool_id=0
create storage_pool name=pool1 capacity=15554GB disk_domain_id=1 raid_level=RAID6
   pool_id=1
create lun name=lun1 pool_id=0 capacity=1944GB owner_controller=0A lun_id=1
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun2 pool_id=0 capacity=1944GB owner_controller=0B lun_id=2
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun3 pool_id=1 capacity=1944GB owner_controller=1A lun_id=3
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun4 pool_id=1 capacity=1944GB owner_controller=1B lun_id=4
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun5 pool_id=0 capacity=1944GB owner_controller=0A lun_id=5
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun6 pool_id=0 capacity=1944GB owner_controller=0B lun_id=6
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun7 pool_id=1 capacity=1944GB owner_controller=1A lun_id=7
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun8 pool_id=1 capacity=1944GB owner_controller=1B lun_id=8
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun9 pool_id=0 capacity=1944GB owner_controller=0A lun_id=9
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun10 pool_id=0 capacity=1944GB owner_controller=0B lun_id=10
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun11 pool_id=1 capacity=1944GB owner_controller=1A lun_id=11
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun12 pool_id=1 capacity=1944GB owner_controller=1B lun_id=12
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun13 pool_id=0 capacity=1944GB owner_controller=0A lun_id=13
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun14 pool_id=0 capacity=1944GB owner_controller=0B lun_id=14
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun15 pool_id=1 capacity=1944GB owner_controller=1A lun_id=15
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun name=lun16 pool_id=1 capacity=1944GB owner_controller=1B lun_id=16
   lun_type=thin compression_enabled=yes dedup_enabled=no
create lun_group name=lg lun_group_id=1
add lun_group lun lun_group_id=1 lun_id_list=1-16
create mapping_view name=mv mapping_view_id=1
create host_group name=hg host_group_id=1
create host name=h1 operating_system=Linux host_id=1
create host name=h2 operating_system=Linux host_id=2
create host name=h3 operating_system=Linux host_id=3
create host name=h4 operating_system=Linux host_id=4
create host name=h5 operating_system=Linux host_id=5
create host name=h6 operating_system=Linux host_id=6
create host name=h7 operating_system=Linux host_id=7
create host name=h8 operating_system=Linux host_id=8
add host_group host host_group_id=1 host_id_list=1,2,3,4,5,6,7,8
add mapping_view host_group mapping_view_id=1 host_group_id=1
add mapping_view lun_group mapping_view_id=1 lun_group_id=1
add host initiator host_id=1 initiator_type=FC wwn=21000024ff756e88
add host initiator host_id=1 initiator_type=FC wwn=21000024ff756e89
add host initiator host_id=1 initiator_type=FC wwn=21000024ff53330c
add host initiator host_id=1 initiator_type=FC wwn=21000024ff53330d
```

```
add host initiator host_id=2 initiator_type=FC wwn=21000024ff543b14
add host initiator host_id=2 initiator_type=FC wwn=21000024ff543b15
add host initiator host_id=2 initiator_type=FC wwn=21000024ff4380b6
add host initiator host_id=2 initiator_type=FC wwn=21000024ff4380b7
add host initiator host_id=3 initiator_type=FC wwn=21000024ff4c3001
add host initiator host_id=3 initiator_type=FC wwn=21000024ff5332b4
add host initiator host_id=3 initiator_type=FC wwn=21000024ff5332b5
add host initiator host_id=3 initiator_type=FC wwn=21000024ff4c3000
add host initiator host_id=4 initiator_type=FC wwn=21000024ff533872
add host initiator host_id=4 initiator_type=FC wwn=21000024ff533873
add host initiator host_id=4 initiator_type=FC wwn=21000024ff54527c
add host initiator host_id=4 initiator_type=FC wwn=21000024ff54527d
add host initiator host_id=5 initiator_type=FC wwn=21000024ff53b4f2
add host initiator host_id=5 initiator_type=FC wwn=21000024ff53b4f3
add host initiator host_id=5 initiator_type=FC wwn=21000024ff5332d4
add host initiator host_id=5 initiator_type=FC wwn=21000024ff5332d5
add host initiator host_id=6 initiator_type=FC wwn=21000024ff4b826a
add host initiator host_id=6 initiator_type=FC wwn=21000024ff4b826b
add host initiator host_id=6 initiator_type=FC wwn=21000024ff4c300c
add host initiator host_id=6 initiator_type=FC wwn=21000024ff4c300d
add host initiator host_id=7 initiator_type=FC wwn=21000024ff3cb6b8
add host initiator host_id=7 initiator_type=FC wwn=21000024ff3cb6b9
add host initiator host_id=7 initiator_type=FC wwn=21000024ff540bba
add host initiator host_id=7 initiator_type=FC wwn=21000024ff540bbb
add host initiator host_id=8 initiator_type=FC wwn=21000024ff53b652
add host initiator host_id=8 initiator_type=FC wwn=21000024ff53b653
add host initiator host_id=8 initiator_type=FC wwn=21000024ff5333c6
add host initiator host_id=8 initiator_type=FC wwn=21000024ff5333c7
```

mkvolume.sh

```
pvcreate /dev/sdb
pvcreate /dev/sdc
pvcreate /dev/sdd
pvcreate /dev/sde
pvcreate /dev/sdf
pvcreate /dev/sdg
pvcreate /dev/sdh
pvcreate /dev/sdi
pvcreate /dev/sdj
pvcreate /dev/sdk
pvcreate /dev/sdl
pvcreate /dev/sdm
pvcreate /dev/sdn
pvcreate /dev/sdo
pvcreate /dev/sdp
pvcreate /dev/sdq

vgcreate vg1 /dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg /dev/sdh
   /dev/sdi /dev/sdj /dev/sdk /dev/sdl /dev/sdm /dev/sdn /dev/sdo /dev/sdp
   /dev/sdq

lvcreate -n asu101 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu102 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu103 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu104 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu105 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu106 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu107 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu108 -i 16 -I 512 -C y -L 630g vg1
```

```
lvcreate -n asu109 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu110 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu111 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu112 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu113 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu114 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu115 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu116 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu117 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu118 -i 16 -I 512 -C y -L 630g vg1

lvcreate -n asu201 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu202 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu203 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu204 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu205 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu206 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu207 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu208 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu209 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu210 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu211 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu212 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu213 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu214 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu215 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu216 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu217 -i 16 -I 512 -C y -L 630g vg1
lvcreate -n asu218 -i 16 -I 512 -C y -L 630g vg1

lvcreate -n asu301 -i 16 -I 512 -C y -L 1260g vg1
lvcreate -n asu302 -i 16 -I 512 -C y -L 1260g vg1
```

APPENDIX E: CONFIGURATION INVENTORY

An inventory of the Tested Storage Configuration was collected before and after the test execution. The test execution script invokes ***shstorage.tcl*** to collect the inventory profile of the storage configuration. The following log file are generated and are included in the Supporting Files (see Appendix A):

- ***profile1_volume.log*** List of configured volumes before the INIT Phase.
- ***profile1_storage.log*** List of configured storage before the INIT Phase.
- ***Profile2_volume.log*** List of configured volumes after TSC restart.
- ***Profile2_storage.log*** List of configured storage after TSC restart.

The above script is included in the Supporting Files (see Appendix A) and listed below.

shstorage.tcl

```
#!/usr/bin/tclsh

package require Expect
set stor 8.46.177.230
set stor_user admin
set stor_pswd Admin@storage1

proc cmd {str} {
    append str "\r"
         send "\r"
         expect ">"
         send $str
    expect {
           -re "More" {
              send " ";
                         exp_continue
        }
           -re ">" {
                         send "\r"
        }
    }
         set expect_out(buffer) " "
}

spawn ssh $stor_user@$stor
set timeout 60
expect {
    "assword" {
       send "$stor_pswd\r"
    }
    "yes/no" {
       send "yes\r";
       expect "assword";
       send "$stor_pswd\r"
    }
}
expect ">"

set ddcount 2
```

```
set poolcount 2
set luncount 16
set mapcount 1

cmd "change user_mode current_mode user_mode=developer"
cmd "show system general"
cmd "show controller general"
sleep 3
send "\003"
cmd "show fan"
cmd "show power_supply"
cmd "change cli capacity_mode=precise"

cmd "show disk_domain general"
for {set i 0} { $i < $ddcount } { incr i } {
   cmd "show disk_domain general disk_domain_id=$i"
        sleep 1
        send " "
        send "\r"
        send "\003"
}

cmd "show storage_pool general"
sleep 1
send " "
send "\r"
send "\003"
for {set i 0} { $i < $poolcount } { incr i } {
   cmd "show storage_pool general pool_id=$i"
}
sleep 1
send " "
send "\r"
send "\003"
cmd "show lun general"
sleep 1
send " "
send "\r"
send "\003"
for {set i 1} { $i <= $luncount } { incr i } {
   cmd "show lun general lun_id=$i"
        sleep 1
        send " "
        send "\r"
        send "\003"
}
sleep 1
send " "
send "\r"
send "\003"
cmd "show disk general"
send "G"
sleep 4
send "\r"
send "\003"
set expect_out(buffer) " "

for {set a 0} { $a <= 24 } { incr a } {
   cmd "show disk general disk_id=CTE0.$a"
        send " "
        send "\r"
```

```
        send "\003"
}
for {set a 0} { $a <= 24 } { incr a } {
   cmd "show disk general disk_id=CTE1.$a"
        send " "
        send "\r"
        send "\003"
}

cmd "change cli capacity_mode=automatic"
cmd "show mapping_view general"
sleep 1
send " "
send "\r"
send "\003"
cmd "show mapping_view general mapping_view_id=1"
sleep 1
send " "
send "\r"
send "\003"
cmd "show mapping_view lun_group mapping_view_id=1"
sleep 1
send " "
send "\r"
send "\003"
cmd "show mapping_view host_group mapping_view_id=1"
sleep 1
send " "
send "\r"
send "\003"
cmd "show lun_group lun lun_group_id=1"
cmd "show host_group host host_group_id=1"
sleep 1
send " "
send "\r"
send "\003"
cmd "show host general host_id=1"
sleep 1
send " "
send "\r"
send "\003"
cmd "show host general host_id=2"
sleep 1
send " "
send "\r"
send "\003"
cmd "show host general host_id=3"
sleep 1
send " "
send "\r"
send "\003"
cmd "show host general host_id=4"
sleep 1
send " "
send "\r"
send "\003"
cmd "show host general host_id=5"
sleep 1
send " "
send "\r"
send "\003"
```

```
cmd "show host general host_id=6"
sleep 1
send " "
send "\r"
send "\003"
cmd "show host general host_id=7"
sleep 1
send " "
send "\r"
send "\003"
cmd "show host general host_id=8"
sleep 1
send " "
send "\r"
send "\003"
cmd "show initiator host_id=1"
sleep 2
send "\r"
cmd "show initiator host_id=2"
sleep 2
send "\r"
cmd "show initiator host_id=3"
sleep 2
send "\r"
cmd "show initiator host_id=4"
sleep 2
send "\r"
cmd "show initiator host_id=5"
sleep 2
send "\r"
cmd "show initiator host_id=6"
sleep 2
send "\r"
cmd "show initiator host_id=7"
sleep 2
send "\r"
cmd "show initiator host_id=8"
sleep 2
send "\r"
cmd "show enclosure"
sleep 2
send "\003"
cmd "show port general physical_type=FC"
sleep 2
send "\003"
cmd "show port general physical_type=SAS"
sleep 2
send "\003"
cmd "show port general"
send "G"
sleep 5
send "\r"
set expect_out(buffer) " "
sleep 5
cmd "show system general"
send "\003"
cmd "show system general"
send "exit\r"
expect ">"
send "exit\r"
expect "(y/n):"
```

```
send "y\r"
sleep 5
send "exit\r"
expect "closed"
```

APPENDIX F: WORKLOAD GENERATOR

The host parameters for the SPC-1 workload generator were defined using the script ***8host.HST***.

The ASUs accessed by the SPC-1 workload generator are defined using the script ***slave_asu.asu***.

The phases of the benchmark are executed using the script ***full_run.sh***. The script pauses at the end of the PERSIST_1 test phase. Once the TSC has been restarted, the PERSIST_2 test phase is executed by pressing ENTER from the console where the script has been invoked.

The above script is included in the Supporting Files (see Appendix A) and listed below.

8host.HST

```
PORT=1962
LOGIN=root
CONFIG=/root/SPCv302
WEIGHT=1
STORAGE=slave_asu.asu
EXEC=spc1
-- Host Entries
HOST=8.46.177.224
HOST=8.46.177.223
HOST=8.46.177.222
HOST=8.46.177.221
HOST=8.46.177.121
HOST=8.46.177.120
HOST=8.46.177.126
HOST=8.46.177.220
```

slave_asu.asu

```
ASU=1
OFFSET=0
SIZE=0
DEVICE=/dev/vg1/asu101
DEVICE=/dev/vg1/asu102
DEVICE=/dev/vg1/asu103
DEVICE=/dev/vg1/asu104
DEVICE=/dev/vg1/asu105
DEVICE=/dev/vg1/asu106
DEVICE=/dev/vg1/asu107
DEVICE=/dev/vg1/asu108
DEVICE=/dev/vg1/asu109
DEVICE=/dev/vg1/asu110
DEVICE=/dev/vg1/asu111
DEVICE=/dev/vg1/asu112
DEVICE=/dev/vg1/asu113
DEVICE=/dev/vg1/asu114
DEVICE=/dev/vg1/asu115
DEVICE=/dev/vg1/asu116
```

```
DEVICE=/dev/vg1/asu117
DEVICE=/dev/vg1/asu118
--
ASU=2
OFFSET=0
SIZE=0
DEVICE=/dev/vg1/asu201
DEVICE=/dev/vg1/asu202
DEVICE=/dev/vg1/asu203
DEVICE=/dev/vg1/asu204
DEVICE=/dev/vg1/asu205
DEVICE=/dev/vg1/asu206
DEVICE=/dev/vg1/asu207
DEVICE=/dev/vg1/asu208
DEVICE=/dev/vg1/asu209
DEVICE=/dev/vg1/asu210
DEVICE=/dev/vg1/asu211
DEVICE=/dev/vg1/asu212
DEVICE=/dev/vg1/asu213
DEVICE=/dev/vg1/asu214
DEVICE=/dev/vg1/asu215
DEVICE=/dev/vg1/asu216
DEVICE=/dev/vg1/asu217
DEVICE=/dev/vg1/asu218
--
ASU=3
OFFSET=0
SIZE=0
DEVICE=/dev/vg1/asu301
DEVICE=/dev/vg1/asu302
```

full_run.sh

```
#!/bin/sh
expect shstorage.tcl > profile1_storage.log
date > profile1_volume.log
lvdisplay >> profile1_volume.log
date >> profile1_volume.log

spc1    -run    SPC1_INIT    -iops    45000    -storage    slave_asu.asu    -output
   ~/newtool/spc1_INIT_45k_iops -master 8host.HST
spc1    -run    SPC1_VERIFY    -iops    100    -storage    slave_asu.asu    -output
   ~/newtool/spc1_VERIFY1_100_iops
spc1   -run   SPC1_METRICS   -iops   800000   -storage   slave_asu.asu   -output
   ~/newtool/spc1_METRICS_800k_iops -master 8host.HST
spc1    -run    SPC1_VERIFY    -iops    100    -storage    slave_asu.asu    -output
   ~/newtool/spc1_VERIFY2_100_iops
spc1   -run   SPC1_PERSIST_1   -iops   200000   -storage   slave_asu.asu   -output
   ~/newtool/spc1_PERSIST_200k_iops -master 8host.HST
echo "Power cycle TSC, then Enter to continue"
read

expect shstorage.tcl > profile2_storage.log
date > profile2_volume.log
lvdisplay >> profile2_volume.log
date >> profile2_volume.log
spc1   -run   SPC1_PERSIST_2   -iops   200000   -storage   slave_asu.asu   -output
   ~/newtool/spc1_PERSIST_200k_iops -master 8host.HST
```