



SPC BENCHMARK 1™

FULL DISCLOSURE REPORT

**MACROSAN TECHNOLOGIES CO., LTD
MACROSAN MS2500G2-25E**

SPC-1 V3.6.0

SUBMISSION IDENTIFIER: A31011

SUBMITTED FOR REVIEW: FEBRUARY 12, 2018

First 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 MacroSAN 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. MacroSAN 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 MacroSAN representative for information on products and services available in your area.

© Copyright MacroSAN 2018. All rights reserved.

Permission is hereby granted to publicly disclose and reproduce this document, in whole or in part, provided the copyright notice as printed above is set forth in full text on the title page of each item reproduced.

Trademarks

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

MacroSAN, the MacroSAN logo and MS2500G2 are trademarks or registered trademarks of MacroSAN in the People's Republic of China 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..... 4

Letter Of Good Faith 6

Executive Summary 7

Configuration Information 12

Benchmark Configuration and Tested Storage Configuration..... 12

Benchmark Configuration Creation Process 14

Benchmark Execution Results..... 16

Benchmark Execution Overview 16

SUSTAIN Test Phase..... 18

RAMPD_100 Test Phase..... 21

Response Time Ramp Test..... 24

Repeatability Test 26

Data Persistence Test 29

Appendix A: Supporting Files 30

Appendix B: Third Party Quotation 31

Appendix C: Tuning Parameters and Options 32

Appendix D: Storage Configuration Creation 33

Appendix E: Configuration Inventory..... 41

Appendix F: Workload Generator 42

AUDIT CERTIFICATION



MacroSan Technologies Co., Ltd.
 11F-12F Building A, No.482
 Qianmo Road, BinJiang District
 Hangzhou, P.R.China

February 9, 2018

I verified the SPC Benchmark 1™ (SPC-1™ Revision 3.6.0) test execution and performance results of the following Tested Storage Product:

MACROSAN MS2500G2-25E

The results were:

SPC-1 IOPS™	300,028
SPC-1 Price-Performance™	\$379.50/SPC-1 KIOPS™
SPC-1 IOPS™ Response Time	0.480 ms
SPC-1 Overall Response Time	0.388 ms
SPC-1 ASU Capacity	10,640.0 GB
SPC-1 ASU Price	\$10.71/GB
SPC-1 Total System Price	\$113,858.31

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 3.0.2 build 1-g823a. The audit process was conducted remotely, 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 **A31011**.

A31011

MACROSAN MS2500G2-25E

p.2

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

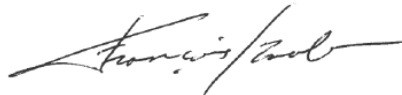
- The physical capacity of the data repository;
- The total capacity of the Application Storage Unit (ASU);
- The accuracy of the Benchmark Configuration diagram;
- The tuning parameters used to configure the Benchmark Configuration;
- The Workload Generator commands used to execute the testing;
- The 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,



Francois Raab, Certified SPC Auditor

LETTER OF GOOD FAITH



Date: February 9, 2018

From: MacroSAN Technologies Co., Ltd.
10F-12F, Building A, No. 482
Qianmo Road, Binjiang District,
Hangzhou, P.R. China 310053
<http://www.macrosan.com/en/>

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

Subject: SPC-1 Letter of Good Faith for the MacroSAN MS2500G2-25E

MacroSAN 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 V3.5.0 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:

A handwritten signature in blue ink, appearing to read 'Li Zhi'.

Date:

February 9, 2018

Li Zhi
President
MacroSAN Technologies Co., Ltd.



SPC BENCHMARK 1™

EXECUTIVE SUMMARY

MACROSAN TECHNOLOGIES CO., LTD MACROSAN MS2500G2-25E

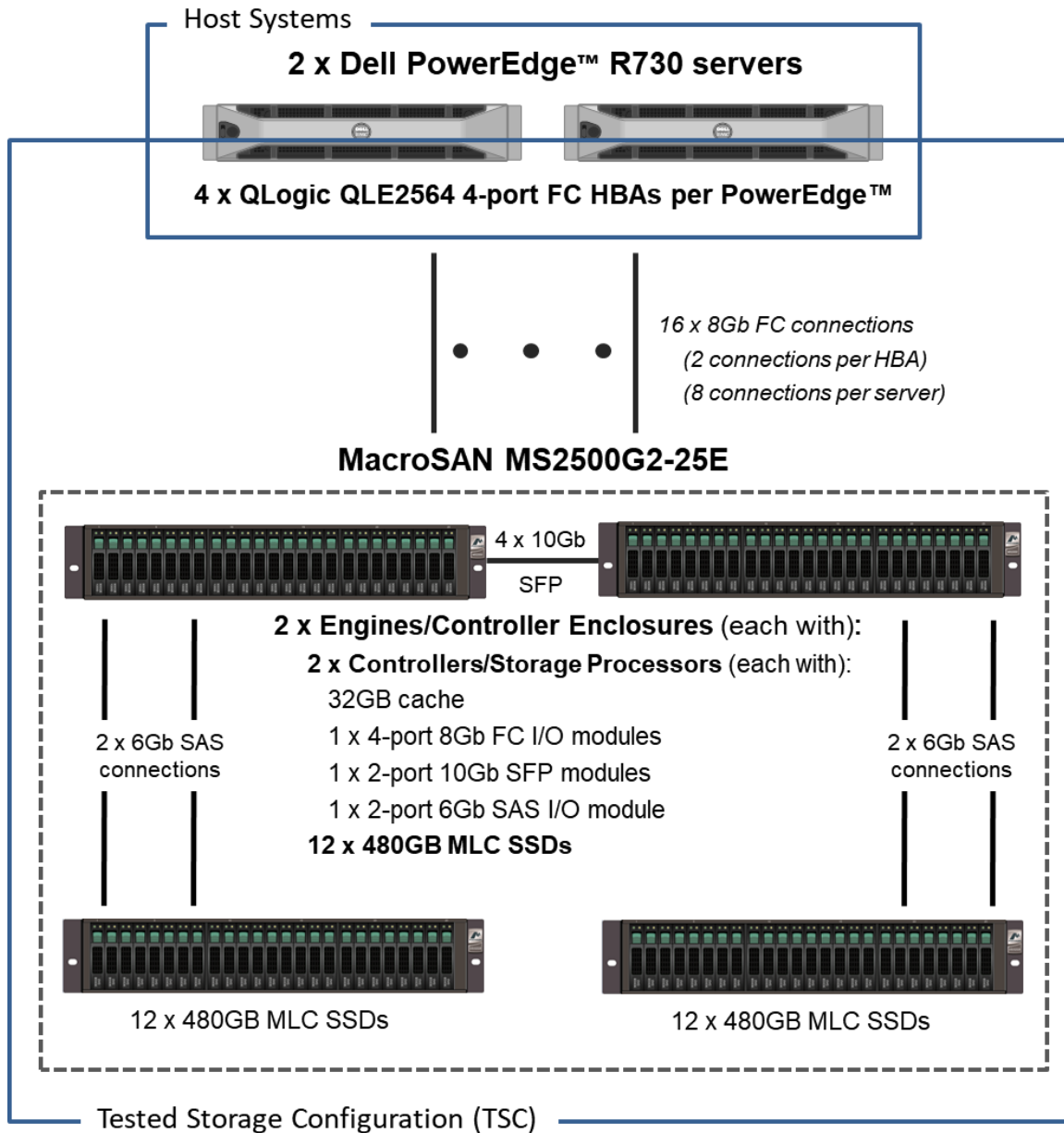
SPC-1 IOPS™	300,028
SPC-1 Price-Performance™	\$379.50/SPC-1 KIOPS™
SPC-1 IOPS™ Response Time	0.480 ms
SPC-1 Overall Response Time	0.388 ms
SPC-1 ASU Capacity	10,640.0 GB
SPC-1 ASU Price	\$10.71/GB
SPC-1 Total System Price	\$113,858.31
Data Protection Level	Protected 2 (RAID 10 and full redundancy)
Physical Storage Capacity	21,408 GB
Pricing Currency / Target Country	U.S. Dollars / People's Republic of China

SPC-1 V3.6.0

SUBMISSION IDENTIFIER: A31011

SUBMITTED FOR REVIEW: FEBRUARY 12, 2018

Benchmark Configuration Diagram



Tested Storage Product Description

The MacroSAN MS2500G2-25E is a new product in the MacroSAN MS2500G2 series, which is the new generation of MacroSAN entry-level data center storage products, oriented toward small size data centers and branches. Based on its leading hardware architecture and rich software function, the MS2500G2-25E can provide more storage solutions for the small business to guarantee data safety and service continuity, including database storage solution, centralized online storage, backup storage solution, data disaster recovery solution and more.

The MS2500G2-25E provides an optimal balance between performance and cost, with better reliability and expansion capability, to support customer's rapid business growth.

For more details, visit:

http://www.macrosan.com/en/product_detail.aspx?k1=2&k2=10&id=73

Priced Storage Configuration Components

8 x QLogic QLE2564 4-port FC HBAs
2 x MacroSAN MS2500G2 Engines, each with:
2 x Controllers/Storage Processors, each with:
32GB cache
1 x 4-port 8Gb FC I/O Modules
1 x 2-port 10Gb SFP Modules
1 x 2-port 6Gb SAS 2.0 I/O Modules
12 x 480GB MLC SSDs
2 x External storage enclosures, each with:
12 x 480GB MLC SSDs

Storage Configuration Pricing

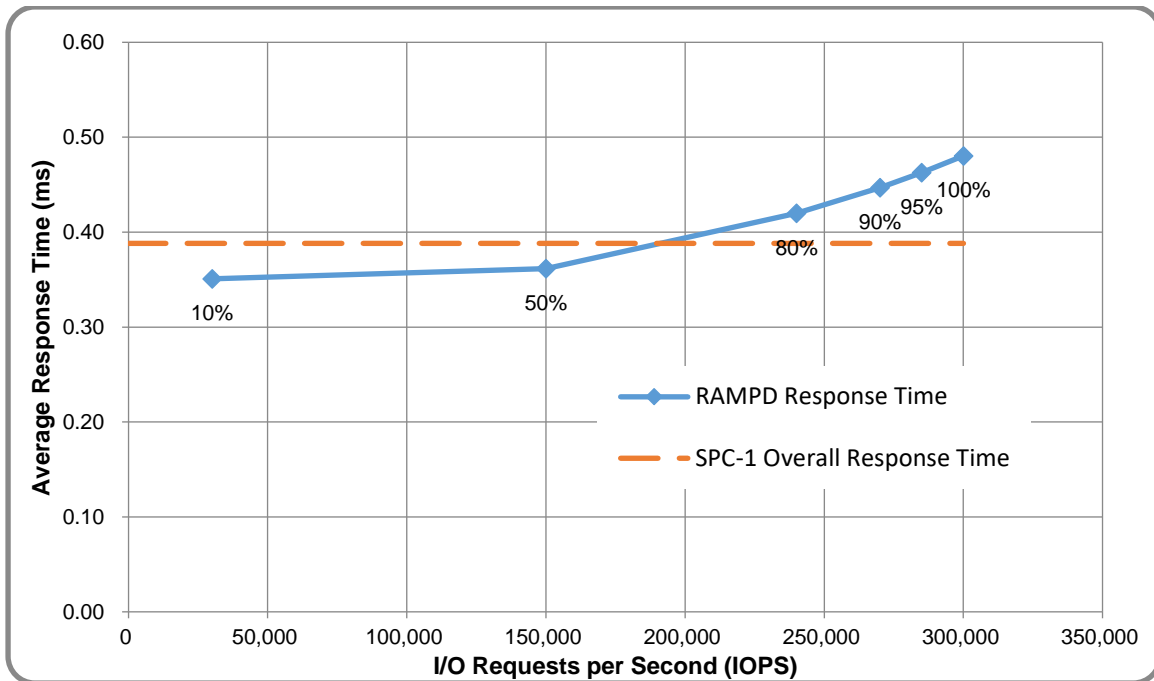
	Description	Qty	Unit Price	Ext. Price	Disc.	Disc. Price
Hardware & Software						
MS2500G2-25E-64GB	MS2500G2-25E Storage Controllers Unit (Dual Controllers, 2U, 25*2.5 disks, 64GB cache, 4*GE ports+4*10GE port-no SFPs, 3Y 7x9xND Basic Svc&Warranty)	2	18,595.59	37,191.18	50%	18,595.59
DSU2625	DSU2625 SAS/SATA Disk Shelf Unit (support 25 2.5inch disks, SAS 3.0)	2	6,223.53	12,447.06	50%	6,223.53
MS2M1SD480G0M2AEA	480GB MLC SSD Drive(2.5") for MS2500G2/DSU2625	48	2,367.65	113,647.20	50%	56,823.60
IOA1014A	4*8Gb FC IO Module	8	2,029.41	16,235.29	50%	8,117.65
MS1MMF010G	10Gb MultiMode Datacom SFP+ Optical Transceiver	8	321.32	2,570.56	50%	1,285.28
LIS_MS2000_BASE_STD	MacroSAN ODSP for Multi control - MS2500G2 Storage Management Software(Basic Storage Management, CRAID, support FC&SCSI, System monitoring and Warning) -English version	1	4,102.94	4,102.94	50%	2,051.47
LIS_MS2000_2C_EXPAND	MacroSAN ODSP-MS2500G2 storage system management software -2 Controllers-EXPAND	1	1,992.49	1,992.49	50%	996.25
LC05-LC05-M12-L5	5M fiber (multimode, LC-LC)	20	8.00	160.00		160.00
MS4.150.0001MX	RACK KIT (For 2U devices)	4	61.12	244.48		244.48
5020012	SAS cable (1M, miniSASHD-miniSASHD)	8	0.00	0.00		0.00
2090044	Qlogic QLE2564 HBA Card,PCIe,8Gbps 4-Ports,Fiber Channel Multimode LC Optic Interface	8	1,500.00	12,000.00		12,000.00
Hardware & Software Subtotal						106,497.84
Support & Maintenance						
MS-2000-SV-BS-SPU	MS2500G2 Installation Service - Per SPU	2	982.41	1,964.82	20%	1,571.86
MS-2001-SV-BS-DSU	Disk Enclosure Installation Base Service per Set	2	588.24	1,176.48	20%	941.18
MS-2000-SV-GS-1Y-SPU	MS2500G2 Svc&Warranty Upgrade - 7x24x4 Premium - Per Year Per SPU	2	1,711.89	3,423.78	20%	2,739.02
MS-2000-SV-GS-1Y-DSU	MS2500G2 Svc&Warranty Upgrade - 7x24x4 Premium - Per Year Per DSU	2	1,317.75	2,635.50	20%	2,108.40
Support & Maintenance Subtotal						7,360.46
SPC-1 Total System Price						113,858.31
SPC-1 IOPS™						300,028
SPC-1 Price-Performance™ (\$/SPC-1 KIOPS™)						379.50
SPC-1 ASU Capacity (GB)						10,640
SPC-1 ASU Price (\$/GB)						10.71

Discount Details: The discounts shown are generally available, and based on the capacity and total price of the storage configuration purchased.

Warranty: Pricing includes Gold-Level Service with: 24x7 online support, unlimited software upgrades and bug fixes, and on-site presence of a qualified maintenance engineer within 4 hours of a problem acknowledgement, inside the Target Market.

Availability Date: Currently available.

Response Time and Throughput Graph



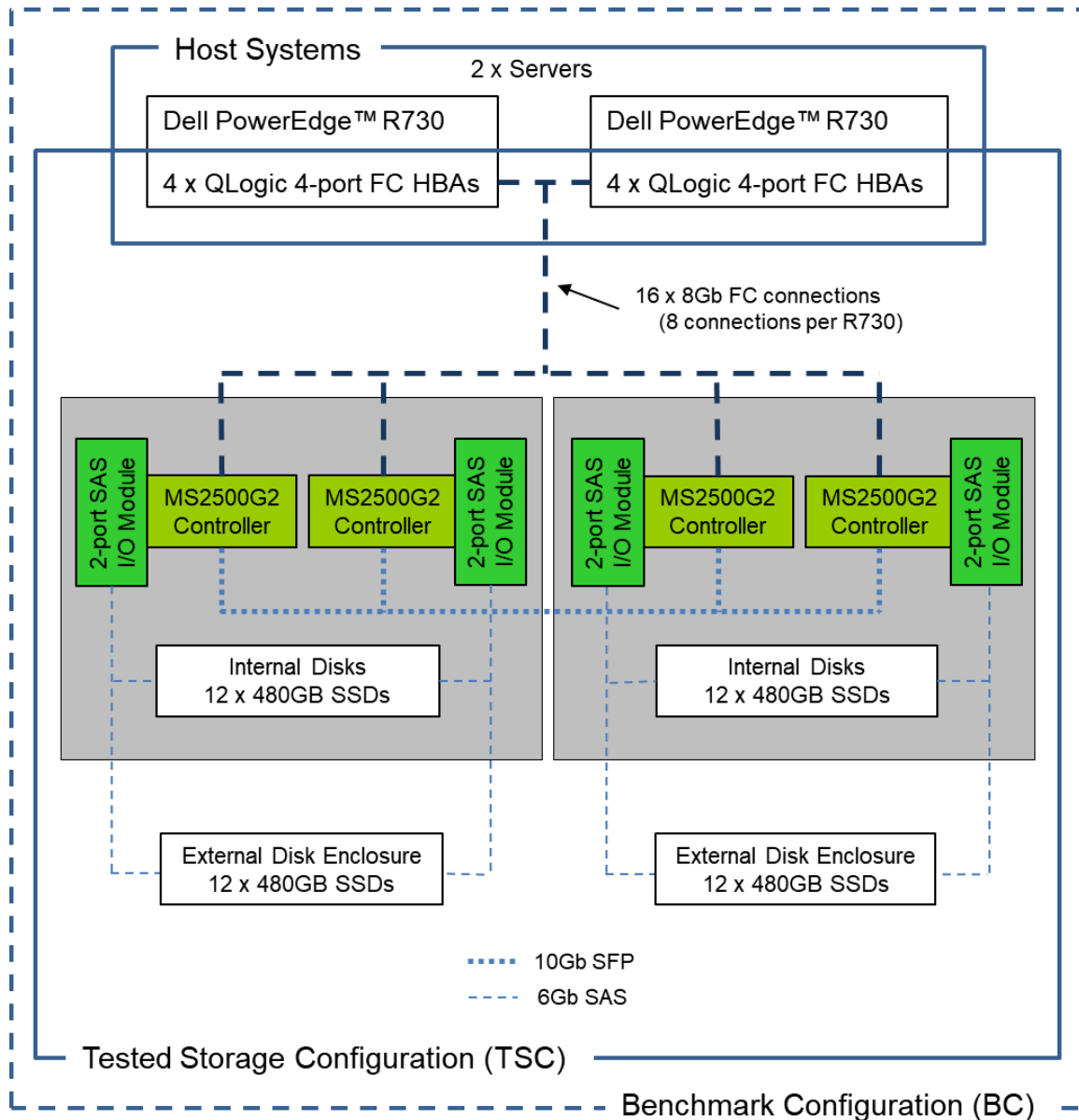
Contact Information	
Test Sponsor Primary Contact	MacroSAN Technologies Co., Ltd.– http://www.macrosan.com.cn Yi Shen – shenyi@macrosan.com
SPC Auditor	InfoSizing – www.sizing.com Francois Raab – francois@sizing.com

Revision Information	
SPC Benchmark 1™ Revision	V3.6.0
SPC-1 Workload Generator Revision	V3.0.2-1-g823a
Publication Revision History	Initial Publication

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 System(s).



Storage Network Configuration

The Tested Storage Configuration (TSC) involved an external storage subsystem made of 2 MacroSAN MS2500G2 Engines, driven by 2 host systems (Dell PowerEdge R730). Each R730 connected one-to-one to each MS2500G2. Each one-to-one connection was established using two ports from each of the four 4-port HBAs on the R730; and all ports from each of the two 4-port I/O Modules on each of the two Controllers/Storage Processors in each MS2500G2. These paths used Fibre Chanel operating at 8Gbps.

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
2 x Dell PowerEdge R730 servers, each with: 2 x Intel Xeon E5-2620 V4 (2.1 GHz, 8-Core, 20MB L3) 64GB Main Memory Red Hat Enterprise Linux 7.1
Priced Storage Configuration
8 x QLogic QLE2564 4-port FC HBAs 2 x MacroSAN MS2500G2 Engines, each with: 2 x Controllers/Storage Processors, each with: 32GB cache 1 x 4-port 8Gb FC I/O Modules 1 x 2-port 10Gb SFP Modules 1 x 2-port 6Gb SAS 2.0 I/O Modules 12 x 480GB MLC SSDs 2 x External storage enclosures, each with: 12 x 480GB MLC SSDs

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	266.0	266.0	4,788.0	45%
ASU-2	18	266.0	266.0	4,788.0	45%
ASU-3	2	532.0	532.0	1,064.0	10%
SPC-1 ASU Capacity				10,640.0	

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
480GB SSD	48	446.0	21,408.0
Total Physical Capacity			21,408.0
Physical Capacity Utilization			49.70%

Data Protection

The data protection level used for all logical volumes was **Protected 2**, which was accomplished by configuring 2 pools of 24 drives (one pool for each Engine), into 8 RAID 10 arrays. All components and access paths from the Host Systems to the Storage Devices were redundant.

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).

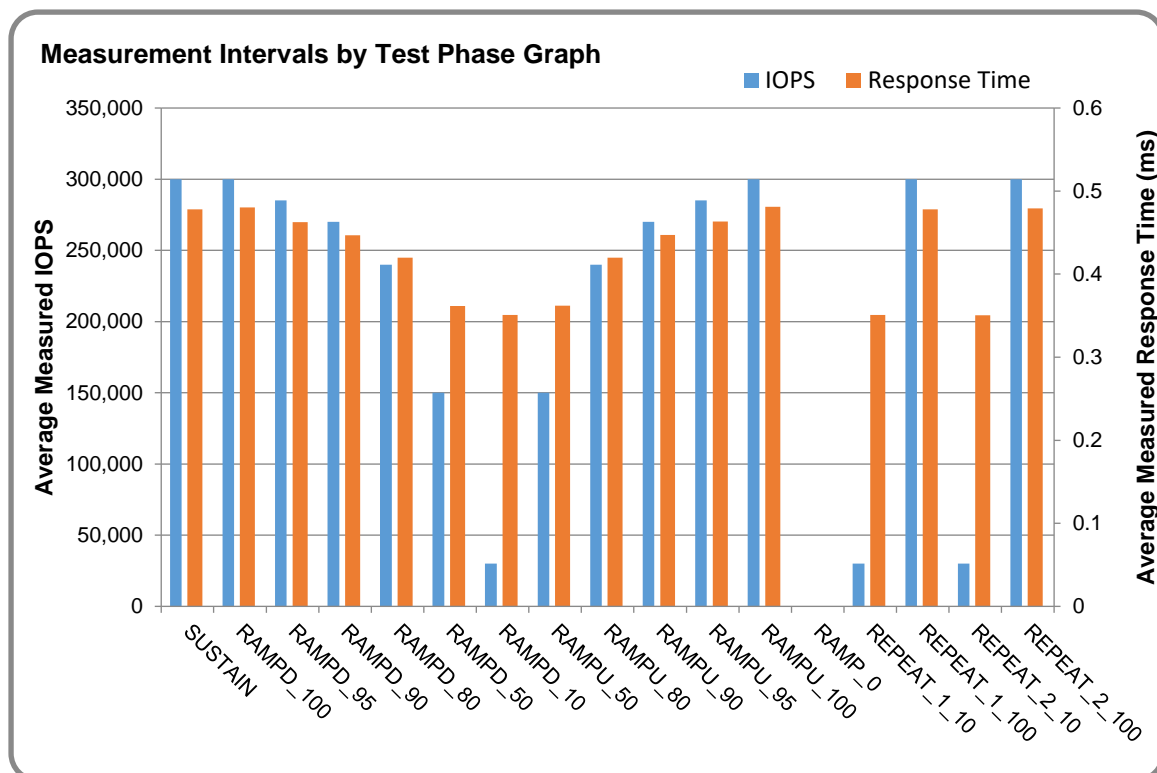
Primary Metrics Test Phases

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

The audit of this SPC-1 result was conducted on-site. During the course of the audit, no exceptions were encountered and no benchmark requirements were waived.

SUSTAIN Test Phase

SUSTAIN – Results File

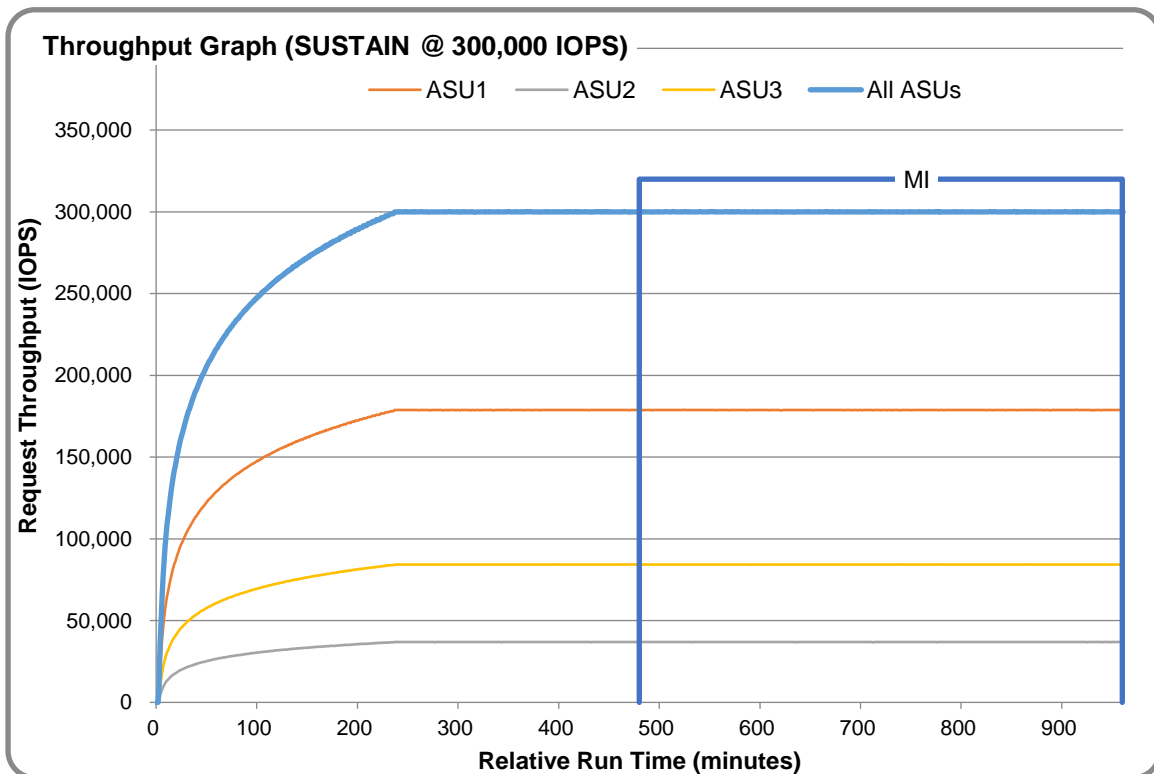
The results file generated during the execution of the SUSTAIN Test Phase is included in the Supporting Files (see Appendix A) as follows:

- SPC1_METRICS_0_Raw_Results.xlsx

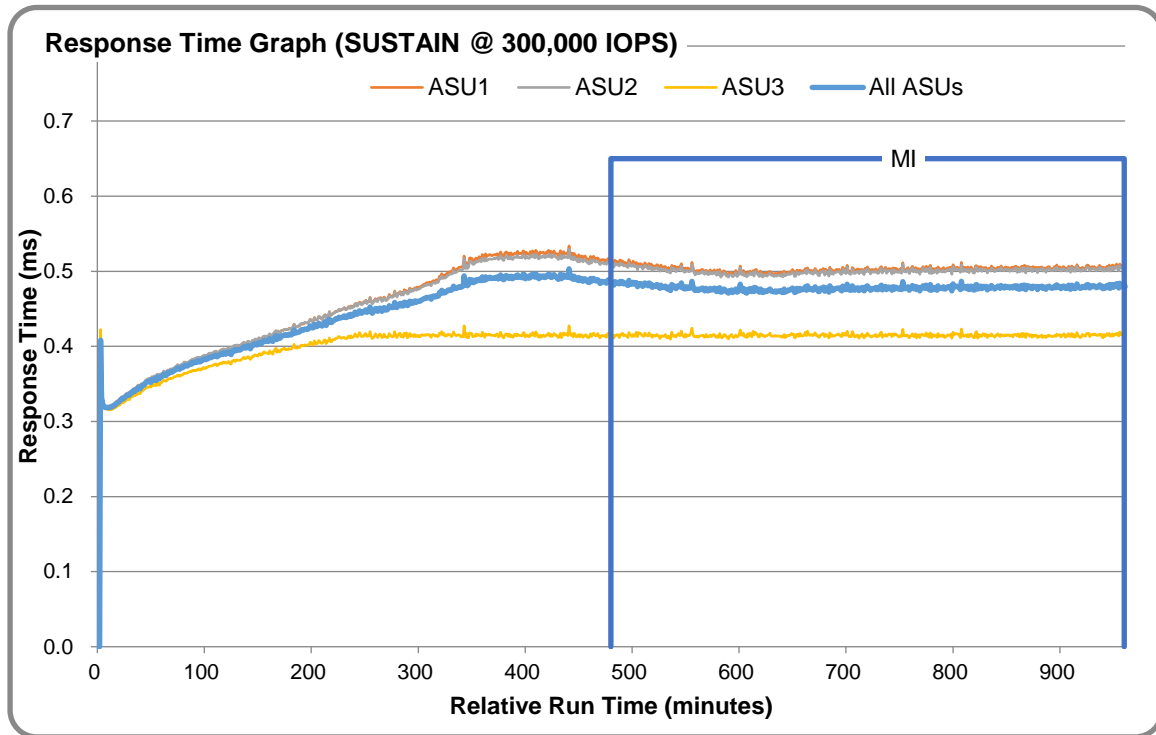
SUSTAIN – Execution Times

Interval	Start Date & Time	End Date & Time	Duration
Transition Period	5-Feb-18 12:13:48	5-Feb-18 20:13:44	7:59:56
Measurement Interval	5-Feb-18 20:13:44	6-Feb-18 04:13:45	8:00:01

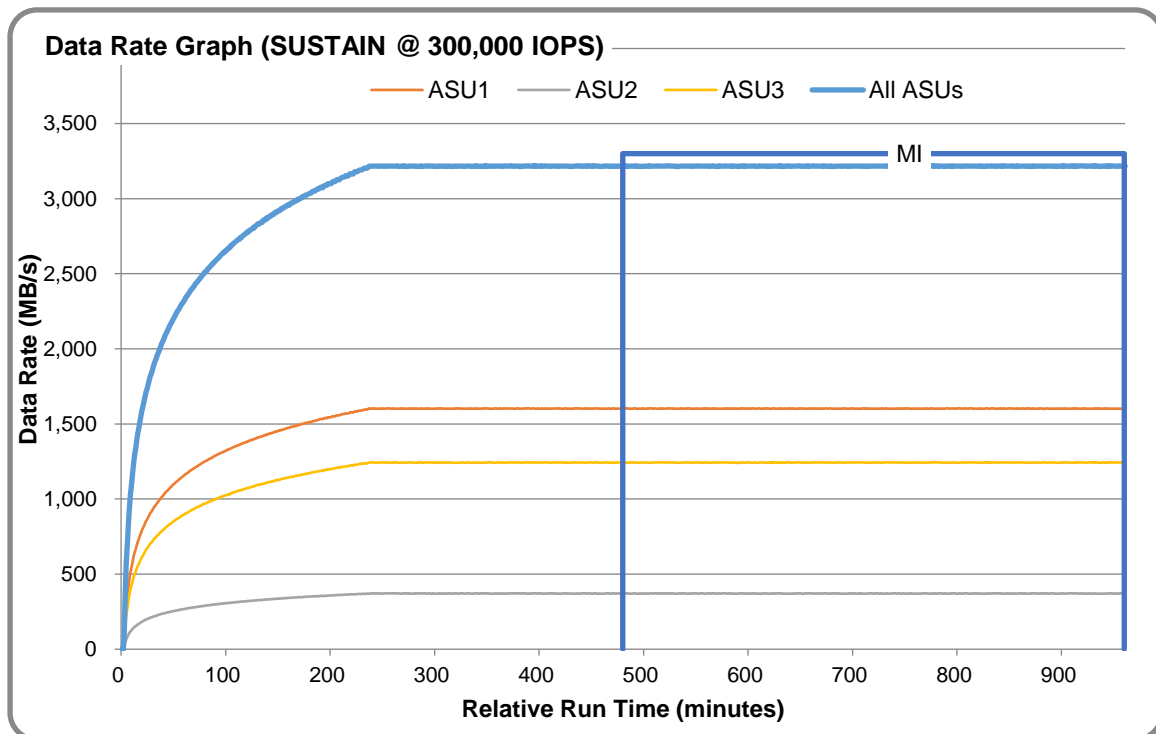
SUSTAIN – Throughput Graph



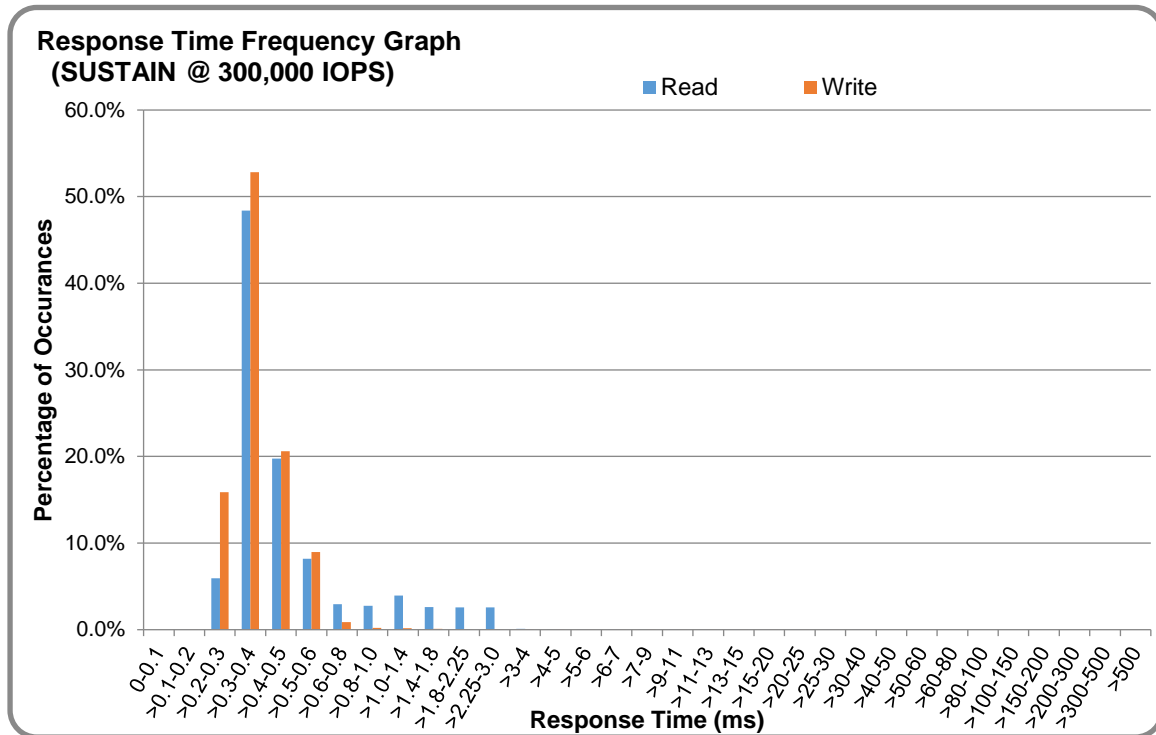
SUSTAIN – Response Time Graph



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	0.0012	0.0004	0.0008	0.0005	0.0017	0.0009	0.0012	0.0004
Difference	0.002%	0.004%	0.008%	0.001%	0.017%	0.011%	0.011%	0.005%

RAMPD_100 Test Phase

RAMPD 100 – Results File

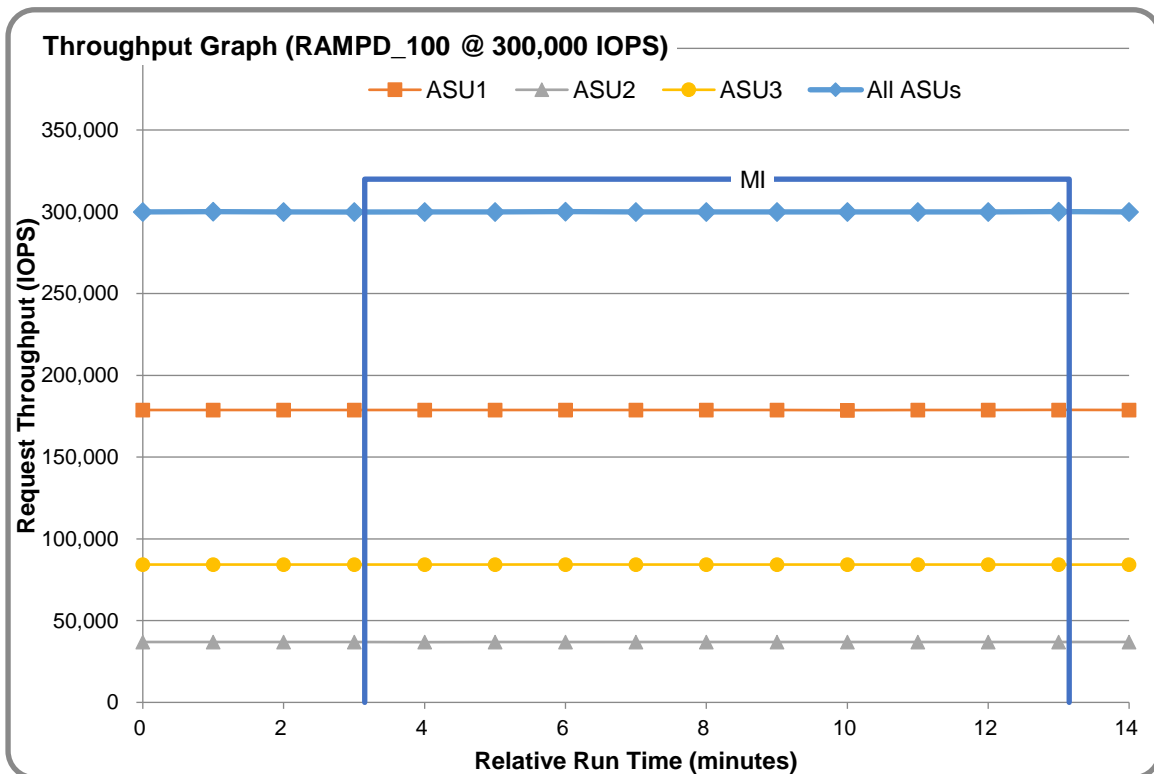
The results file generated during the execution of the RAMPD_100 Test Phase is included in the Supporting Files (see Appendix A) as follows:

- SPC1_METRICS_0_Raw_Results.xlsx

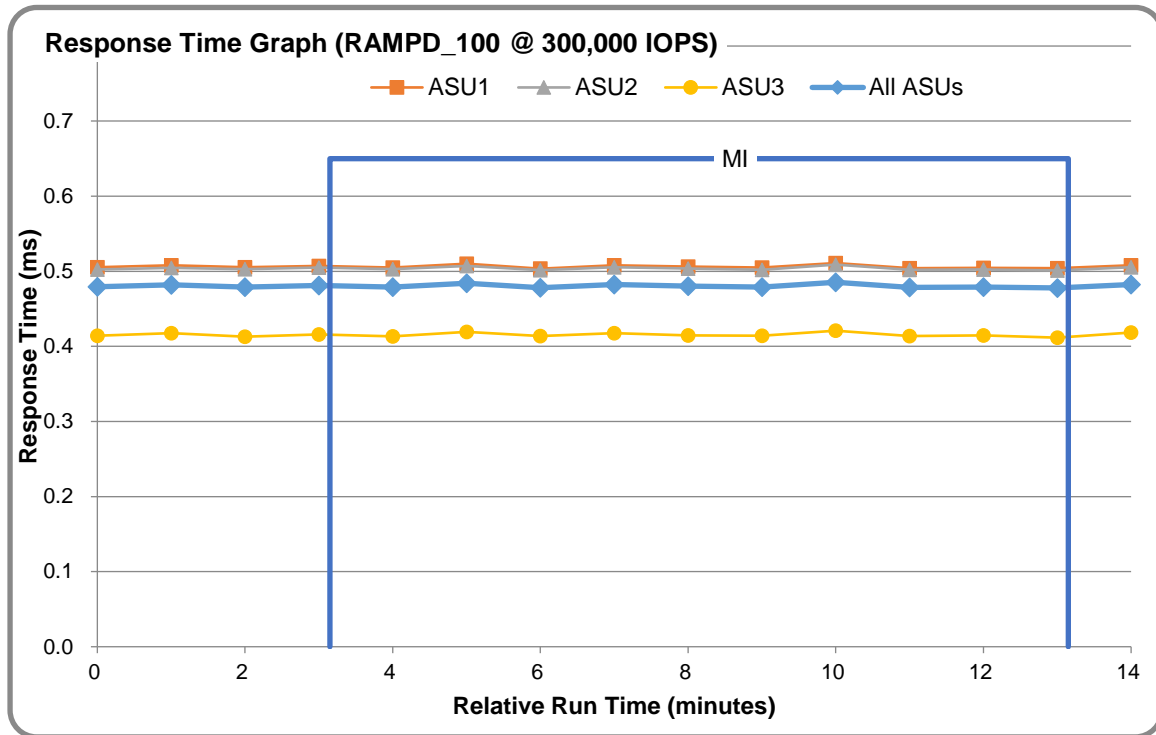
RAMPD 100 – Execution Times

Interval	Start Date & Time	End Date & Time	Duration
Transition Period	6-Feb-18 04:14:44	6-Feb-18 04:17:44	0:03:00
Measurement Interval	6-Feb-18 04:17:44	6-Feb-18 04:27:45	0:10:01

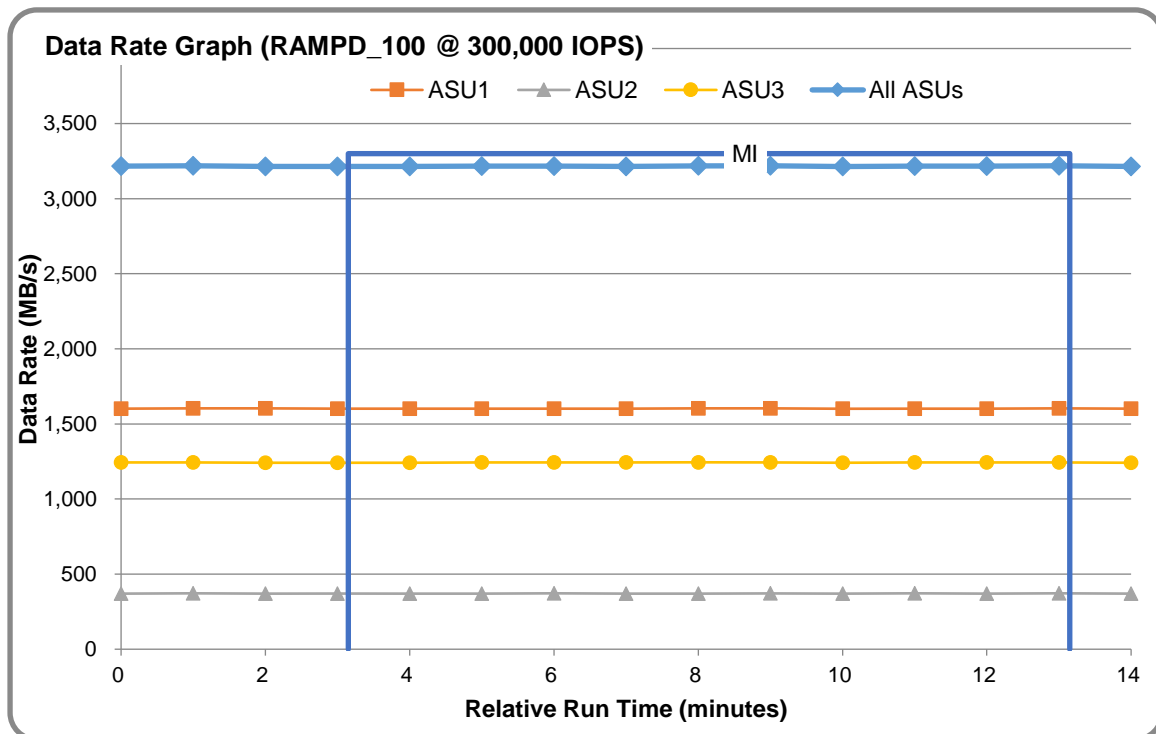
RAMPD 100 – Throughput Graph



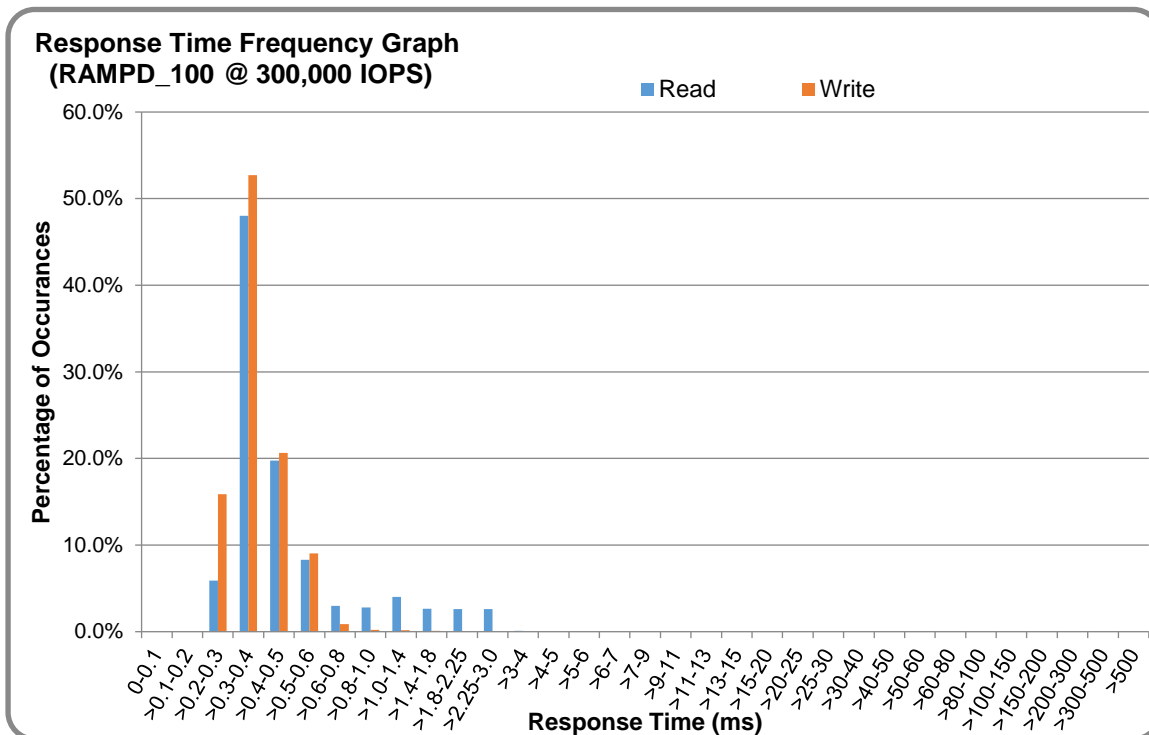
RAMPD 100 – Response Time Graph



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	0.0014	0.0004	0.0009	0.0003	0.0017	0.0009	0.0010	0.0004
Difference	0.027%	0.000%	0.002%	0.017%	0.052%	0.043%	0.019%	0.006%

RAMPD 100 – I/O Request Summary

I/O Requests Completed in the Measurement Interval	180,018,884
I/O Requests Completed with Response Time <= 30 ms	180,018,647
I/O Requests Completed with Response Time > 30 ms	237

Response Time Ramp Test

Response Time Ramp Test – Results File

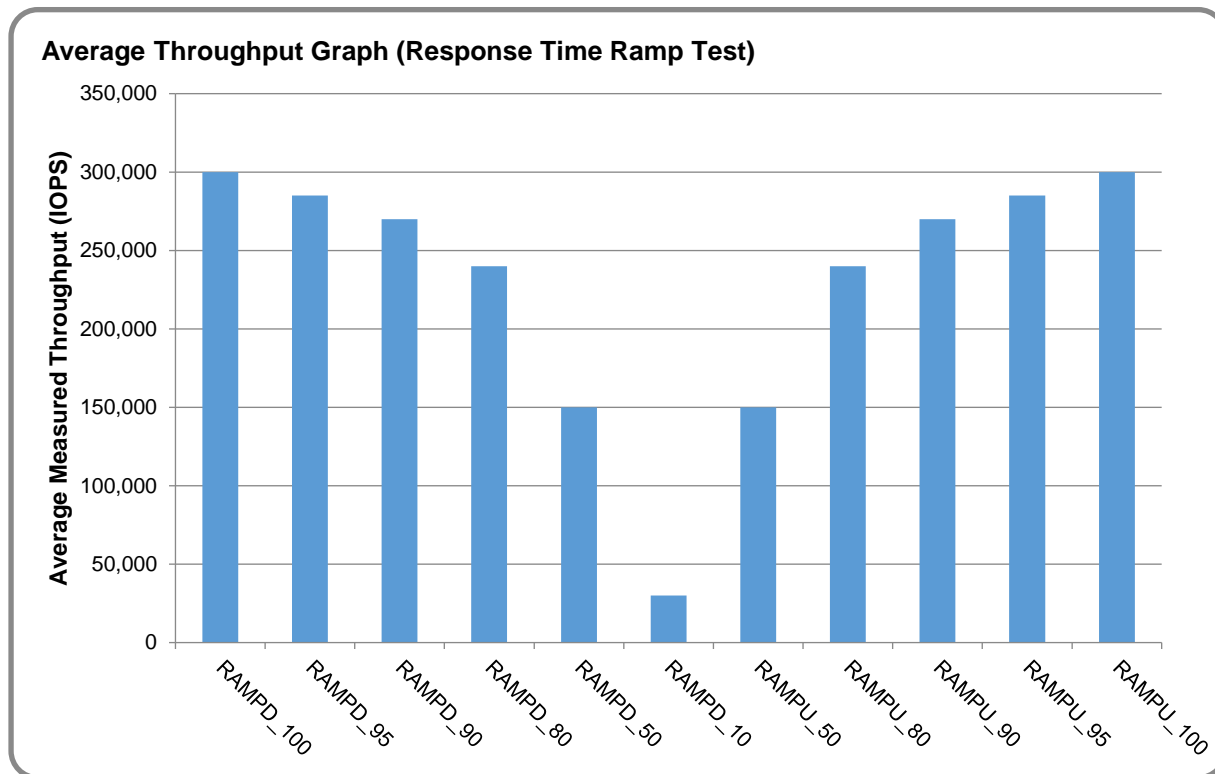
The results file generated during the execution of the Response Time Ramp Test is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

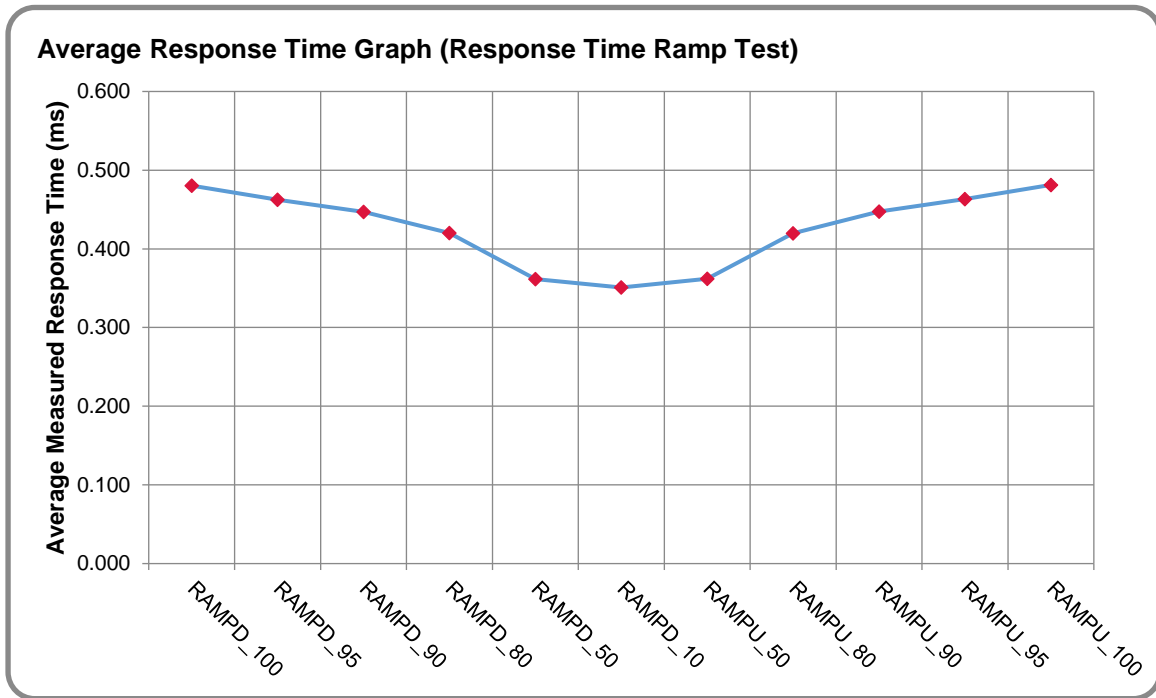
Response Time Ramp Test – Phases

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).

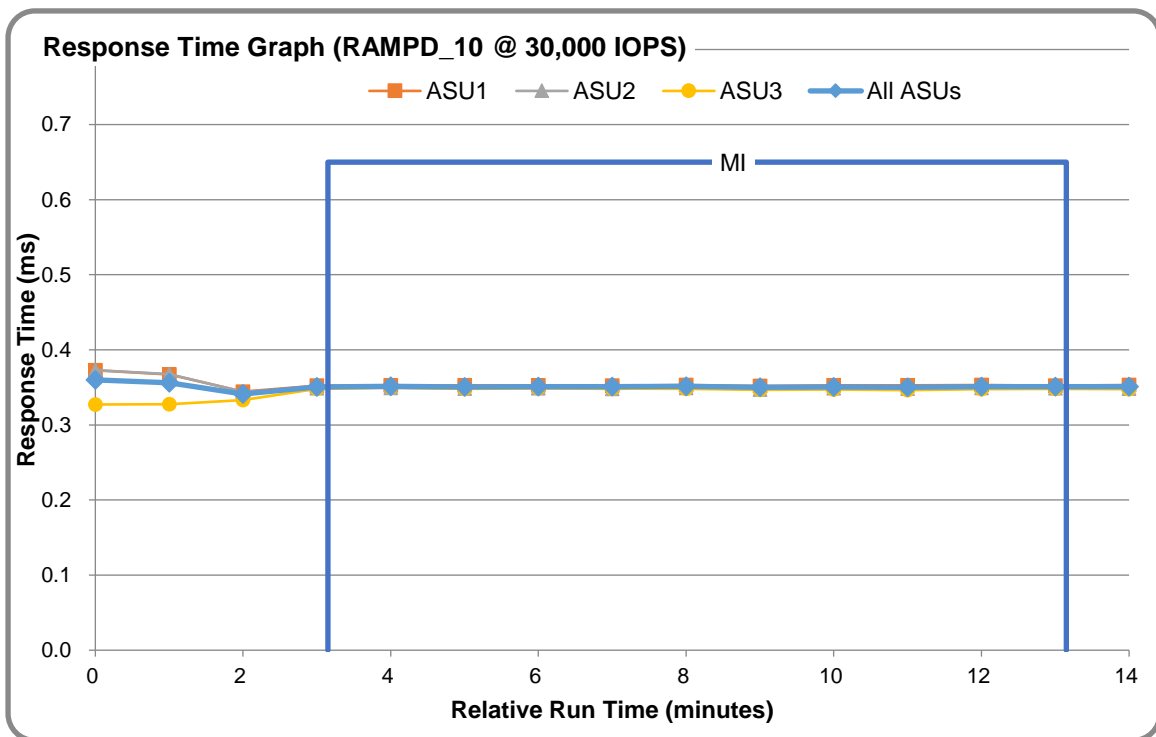
Response Time Ramp Test – Average Throughput Graph



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:

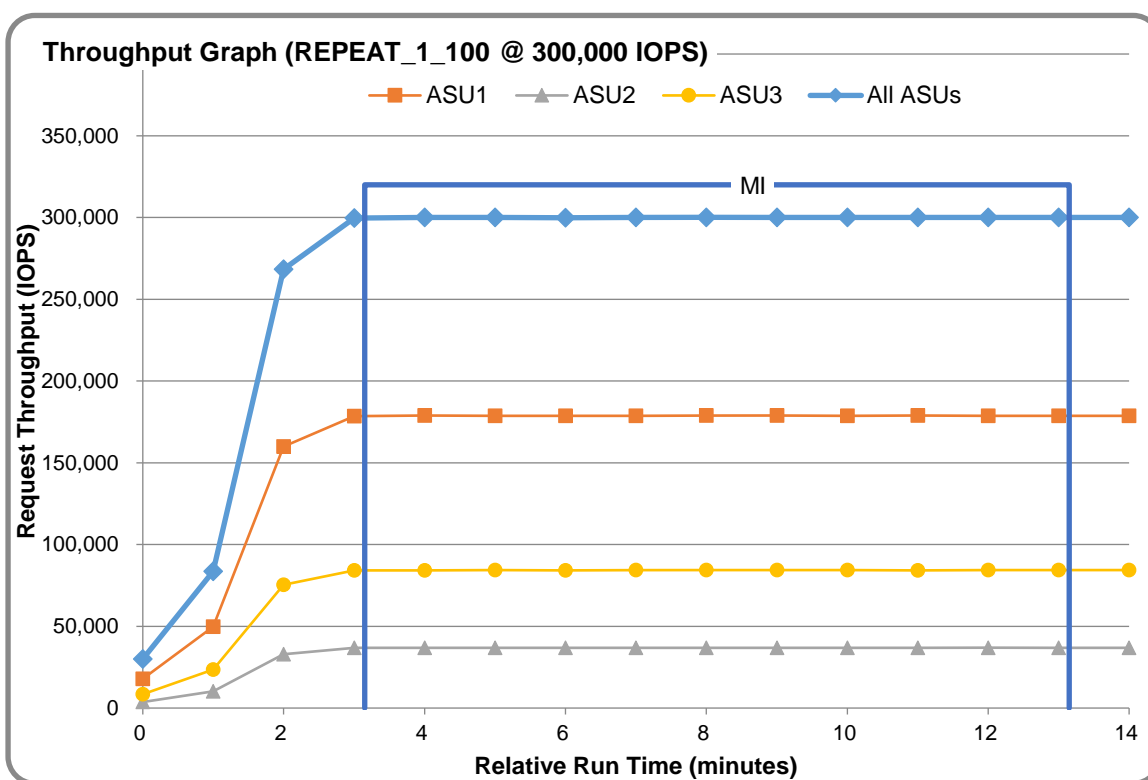
- SPC1_METRICS_0_Raw_Results.xlsx

Repeatability Test Results

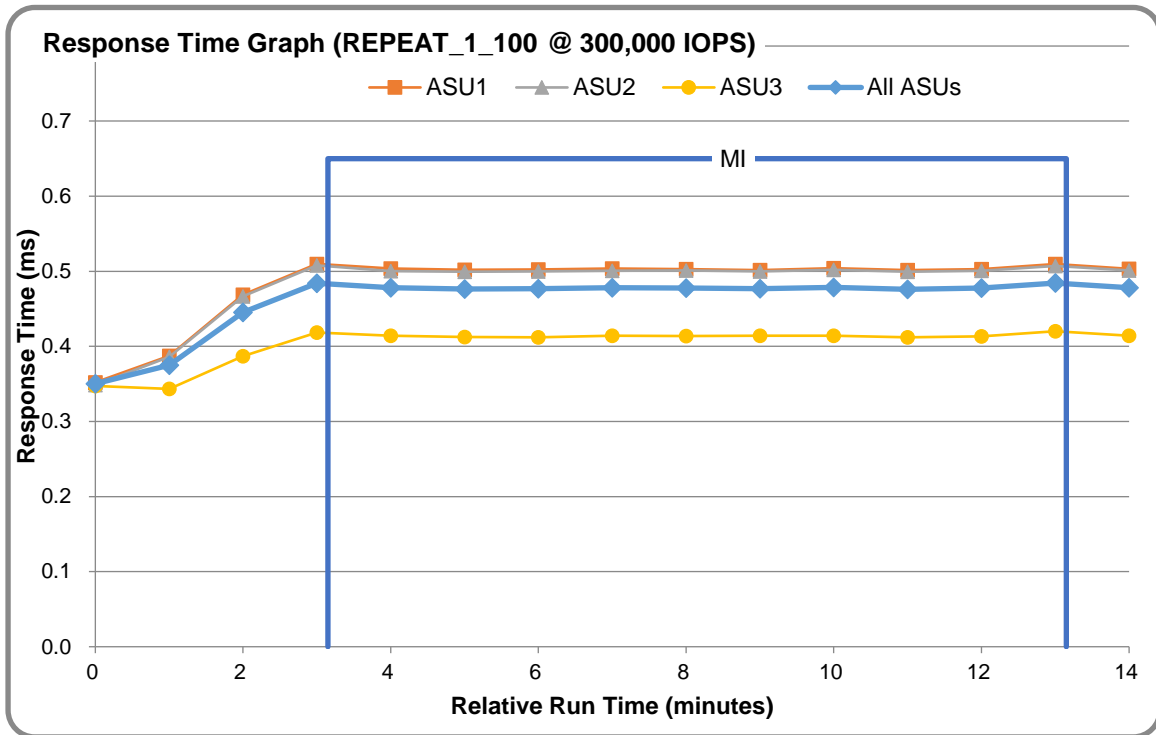
The throughput measurements for the Response Time Ramp Test (RAMPD) and the Repeatability Test Phases (REPEAT_1 and REPEAT_2) are listed in the tables below.

Test Phase	100% IOPS	10% IOPS
RAMPD	300,028.5	30,015.9
REPEAT_1	300,031.7	30,011.0
REPEAT_2	299,978.7	30,007.4

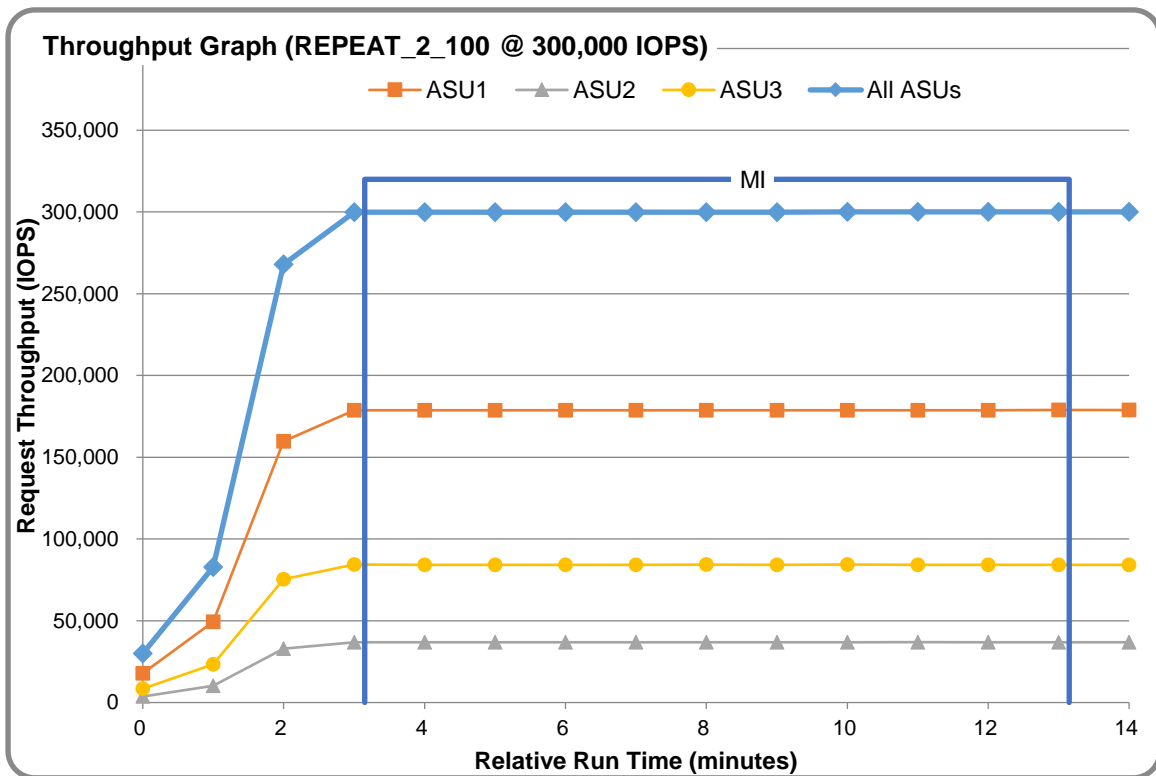
REPEAT 1 100 – Throughput Graph



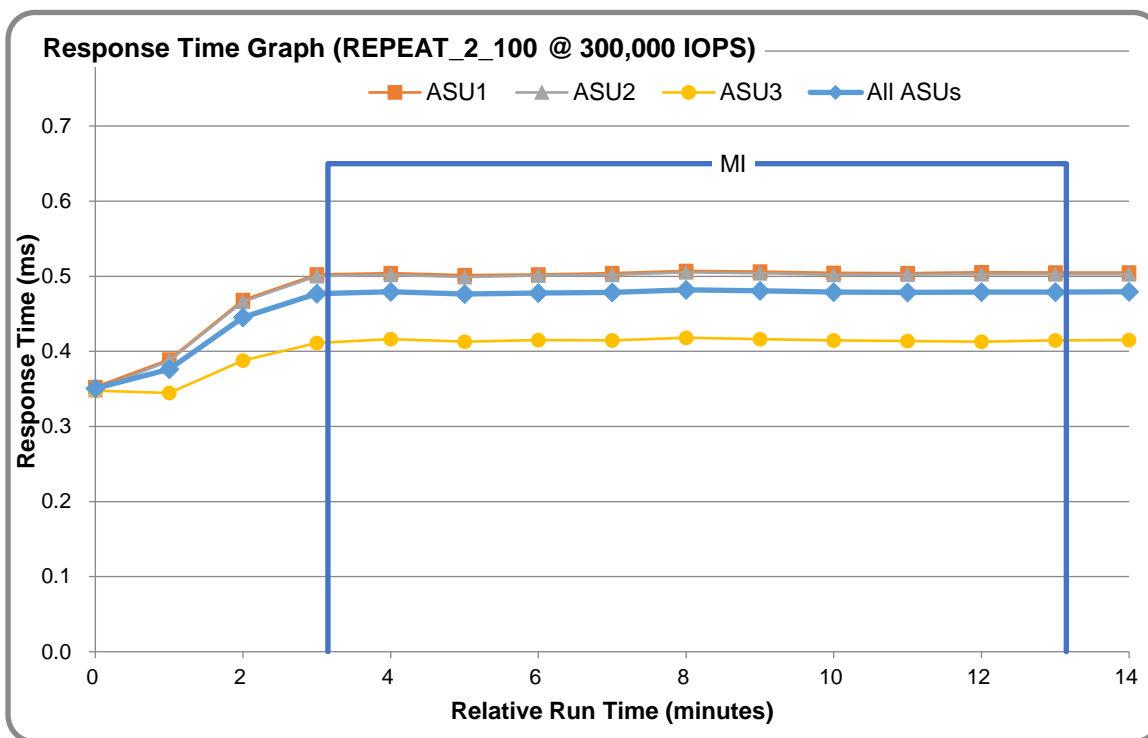
REPEAT 1 100 – Response Time Graph



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	0.2809	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Variation	0.0010	0.0005	0.0008	0.0005	0.0024	0.0005	0.0015	0.0004
Difference	0.053%	0.018%	0.034%	0.022%	0.075%	0.004%	0.031%	0.002%

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	0.0010	0.0003	0.0006	0.0005	0.0016	0.0011	0.0013	0.0004
Difference	0.008%	0.006%	0.002%	0.002%	0.079%	0.006%	0.042%	0.004%

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	37,124,920
Total Number of Logical Blocks Verified	20,082,674
Total Number of Logical Blocks Overwritten	17,042,246
Total Number of Logical Blocks that Failed Verification	0
Time Duration for Writing Test Logical Blocks (sec.)	300
Size in Bytes of each Logical Block	8,192
Number of Failed I/O Requests During the Test	0

Committed Data Persistence Implementation

The TSC uses cache power-loss protection technology. Each MS2500G2 Engine has two batteries and a built-in SSD. When an accidental power loss occurs, the controllers continue to be powered by the built-in batteries and are able to flush the data in their cache to permanent storage on built-in SSD. When power is restored and the Engine is restarted, the data on the built-in SSD is used for automatic recovery.

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_request.sh	Increase disk queue depth	/C_Tuning
scheduler.sh	Change the I/O scheduler	/C_Tuning
/D_Creation	Storage configuration creation	root
init_ms2500g2.sh	Create the storage environment	/D_Creation
mklvm_ms2500g2.sh	Create the Logical Volumes	/D_Creation
lv_scan.sh	Scan the Logical Volumes	/D_Creation
/E_Inventory	Configuration inventory	root
volume_list.sh	Captures list of Logical Volumes	/E_Inventory
profile_ms2500g2.sh	Captures storage devices profiles	/E_Inventory
volume_listing_start.txt	List of logical volumes before INIT	/E_Inventory
volume_listing_end.txt	List of logical volumes after restart	/E_Inventory
profile_start_ms2500g2.txt	List of storage devices before INIT	/E_Inventory
profile_end_ms2500g2.txt	List of storage devices after restart	/E_Inventory
/F_Generator	Workload generator	root
slave_asu.asu	Defining LUNs hosting the ASUs	/F_generator
2host.HST	Host configuration file	/F_generator
ms2500g2_test_phase1.sh	Executing test phases up to VERIFY_1	/F_generator
ms2500g2_test_persist1.sh	Executing test phase PERSIST_1	/F_generator
ms2500g2_test_phase2.sh	Executing test phase PERSIST_2	/F_generator

APPENDIX B: THIRD PARTY QUOTATION

All components are directly available through the Test Sponsor.

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_request.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
cat /proc/sys/fs/aio-max-nr
```

nr_request.sh

```
#!/bin/bash

#default:
for i in sdc sdd sdb sde sdg sdf sdh sdj sdi sdk sdm sdl sdn sdp sdo sdq sdr sds
    sdt sdu sdv sdw sdx sdy sdz
do
    echo 1024 > /sys/block/$i/queue/nr_requests
done

#show
cat -n /sys/block/sd[b-z]/queue/nr_requests
```

scheduler.sh

```
#!/bin/bash

#default: noop anticipatory deadline [cfq]
for i in sdc sdd sdb sde sdg sdf sdh sdj sdi sdk sdm sdl sdn sdp sdo sdq sdr sds
    sdt sdu sdv sdw sdx sdy sdz
do
    echo noop > /sys/block/$i/queue/scheduler
done

#show
cat -n /sys/block/sd[b-z]/queue/scheduler
```


APPENDIX D: STORAGE CONFIGURATION CREATION

Step 1 - Create Storage Pool, RAIDs, LUNs, Clients, Mapping

The *init_ms2500g2.sh* script, listed below, is executed on either of the Storage Controllers via a CLI session to perform the following actions:

1. Create a storage pool for each Engine: Pool-A-1 and Pool-B-1
2. Create 8 RAID10
3. Create 16 LUNs (2 LUNs per RAID10, 667GB per LUN)
4. Create 2 Clients in each Engine
5. Add Host System FC Initiators for the Clients (4 Initiator per Client)
6. Create Storage Targets for the Clients (4 Targets per Client and each Target is associated with 1 FC port)
7. Add LUNs to the Targets (2 LUN per Target)

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

init_ms2500g2.sh

```
#!/bin/bash

#Create 2*Pool:Pool-A-1/Pool-B-1
echo "Waiting for create 2*Pool"
ms-cli EngineA "pool mgt create -n Pool-A-1 -t t"
ms-cli EngineB "pool mgt create -n Pool-B-1 -t t"

#Create RAID: 8 * RAID10
echo "Waiting for create EngineA RAID"
ms-cli EngineA "raid mgt create -n RAID-0001 -p Pool-A-1 -l RAID10 -d
1:1:1:1,1:1:1:2,1:1:1:3,1:1:1:4,1:1:1:5,1:1:1:6"
ms-cli EngineA "raid mgt create -n RAID-0002 -p Pool-A-1 -l RAID10 -d
1:1:1:7,1:1:1:8,1:1:1:9,1:1:1:10,1:1:1:11,1:1:1:12"
ms-cli EngineA "raid mgt create -n RAID-0003 -p Pool-A-1 -l RAID10 -d
1:2:1:1,1:2:1:2,1:2:1:3,1:2:1:4,1:2:1:5,1:2:1:6"
ms-cli EngineA "raid mgt create -n RAID-0004 -p Pool-A-1 -l RAID10 -d
1:2:1:7,1:2:1:8,1:2:1:9,1:2:1:10,1:2:1:11,1:2:1:12"

echo "Waiting for create EngineB RAID"
ms-cli EngineB "raid mgt create -n RAID-0001 -p Pool-B-1 -l RAID10 -d
1:1:1:1,1:1:1:2,1:1:1:3,1:1:1:4,1:1:1:5,1:1:1:6"
ms-cli EngineB "raid mgt create -n RAID-0002 -p Pool-B-1 -l RAID10 -d
1:1:1:7,1:1:1:8,1:1:1:9,1:1:1:10,1:1:1:11,1:1:1:12"
ms-cli EngineB "raid mgt create -n RAID-0003 -p Pool-B-1 -l RAID10 -d
1:2:1:1,1:2:1:2,1:2:1:3,1:2:1:4,1:2:1:5,1:2:1:6"
ms-cli EngineB "raid mgt create -n RAID-0004 -p Pool-B-1 -l RAID10 -d
1:2:1:7,1:2:1:8,1:2:1:9,1:2:1:10,1:2:1:11,1:2:1:12"

#Create LUN: 16
echo "Waiting for create LUN"
ms-cli EngineA "lun mgt createthicklun -n LUN-A-SP1-0001 -o SP1 -p Pool-A-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0001"
```

```
ms-cli EngineA "lun mgt createthicklun -n LUN-A-SP2-0001 -o SP2 -p Pool-A-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0001"
ms-cli EngineA "lun mgt createthicklun -n LUN-A-SP1-0002 -o SP1 -p Pool-A-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0002"
ms-cli EngineA "lun mgt createthicklun -n LUN-A-SP2-0002 -o SP2 -p Pool-A-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0002"
ms-cli EngineA "lun mgt createthicklun -n LUN-A-SP1-0003 -o SP1 -p Pool-A-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0003"
ms-cli EngineA "lun mgt createthicklun -n LUN-A-SP2-0003 -o SP2 -p Pool-A-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0003"
ms-cli EngineA "lun mgt createthicklun -n LUN-A-SP1-0004 -o SP1 -p Pool-A-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0004"
ms-cli EngineA "lun mgt createthicklun -n LUN-A-SP2-0004 -o SP2 -p Pool-A-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0004"

ms-cli EngineB "lun mgt createthicklun -n LUN-B-SP1-0001 -o SP1 -p Pool-B-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0001"
ms-cli EngineB "lun mgt createthicklun -n LUN-B-SP2-0001 -o SP2 -p Pool-B-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0001"
ms-cli EngineB "lun mgt createthicklun -n LUN-B-SP1-0002 -o SP1 -p Pool-B-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0002"
ms-cli EngineB "lun mgt createthicklun -n LUN-B-SP2-0002 -o SP2 -p Pool-B-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0002"
ms-cli EngineB "lun mgt createthicklun -n LUN-B-SP1-0003 -o SP1 -p Pool-B-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0003"
ms-cli EngineB "lun mgt createthicklun -n LUN-B-SP2-0003 -o SP2 -p Pool-B-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0003"
ms-cli EngineB "lun mgt createthicklun -n LUN-B-SP1-0004 -o SP1 -p Pool-B-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0004"
ms-cli EngineB "lun mgt createthicklun -n LUN-B-SP2-0004 -o SP2 -p Pool-B-1 -r
disable -w disable -m no-sync -s 667 -R RAID-0004"

#Create client:4
echo "Waiting for create client"
ms-cli EngineA "client mgt create -n Client-77"
ms-cli EngineA "client mgt create -n Client-78"

ms-cli EngineB "client mgt create -n Client-77"
ms-cli EngineB "client mgt create -n Client-78"

#Create initiator:16
echo "Waiting for create initiator"
ms-cli EngineA "client initiator create -t fc -w 21:00:00:24:ff:67:37:07 -a FC-
1:1:1 -o Linux"
ms-cli EngineA "client initiator create -t fc -w 21:00:00:24:ff:67:37:06 -a FC-
1:1:2 -o Linux"
ms-cli EngineA "client initiator create -t fc -w 21:00:00:24:ff:66:f2:3b -a FC-
1:1:3 -o Linux"
ms-cli EngineA "client initiator create -t fc -w 21:00:00:24:ff:66:f2:3a -a FC-
1:1:4 -o Linux"
ms-cli EngineA "client initiator create -t fc -w 21:00:00:24:ff:69:d1:cb -a FC-
2:1:1 -o Linux"
ms-cli EngineA "client initiator create -t fc -w 21:00:00:24:ff:69:d1:ca -a FC-
2:1:2 -o Linux"
ms-cli EngineA "client initiator create -t fc -w 21:00:00:24:ff:69:b6:7f -a FC-
2:1:3 -o Linux"
ms-cli EngineA "client initiator create -t fc -w 21:00:00:24:ff:69:b6:7e -a FC-
2:1:4 -o Linux"

ms-cli EngineB "client initiator create -t fc -w 21:00:00:24:ff:71:64:e7 -a FC-
1:1:1 -o Linux"
```

```
ms-cli EngineB "client initiator create -t fc -w 21:00:00:24:ff:71:64:e6 -a FC-1:1:2 -o Linux"
ms-cli EngineB "client initiator create -t fc -w 31:61:00:24:ff:af:04:07 -a FC-1:1:3 -o Linux"
ms-cli EngineB "client initiator create -t fc -w 21:61:00:24:ff:af:04:07 -a FC-1:1:4 -o Linux"
ms-cli EngineB "client initiator create -t fc -w 31:c7:00:24:ff:7b:d3:3f -a FC-2:1:1 -o Linux"
ms-cli EngineB "client initiator create -t fc -w 21:c7:00:24:ff:7b:d3:3f -a FC-2:1:2 -o Linux"
ms-cli EngineB "client initiator create -t fc -w 31:c9:00:24:ff:0a:96:0f -a FC-2:1:3 -o Linux"
ms-cli EngineB "client initiator create -t fc -w 21:c9:00:24:ff:0a:96:0f -a FC-2:1:4 -o Linux"

#Add Target: 16
echo "Waiting for add Target"
ms-cli EngineA "client target create -t fc -p FC-1:1:1"
ms-cli EngineA "client target create -t fc -p FC-1:1:2"
ms-cli EngineA "client target create -t fc -p FC-1:1:3"
ms-cli EngineA "client target create -t fc -p FC-1:1:4"
ms-cli EngineA "client target create -t fc -p FC-2:1:1"
ms-cli EngineA "client target create -t fc -p FC-2:1:2"
ms-cli EngineA "client target create -t fc -p FC-2:1:3"
ms-cli EngineA "client target create -t fc -p FC-2:1:4"

ms-cli EngineB "client target create -t fc -p FC-1:1:1"
ms-cli EngineB "client target create -t fc -p FC-1:1:2"
ms-cli EngineB "client target create -t fc -p FC-1:1:3"
ms-cli EngineB "client target create -t fc -p FC-1:1:4"
ms-cli EngineB "client target create -t fc -p FC-2:1:1"
ms-cli EngineB "client target create -t fc -p FC-2:1:2"
ms-cli EngineB "client target create -t fc -p FC-2:1:3"
ms-cli EngineB "client target create -t fc -p FC-2:1:4"

#map-client
echo "Waiting for map-client"
ms-cli EngineA "client mgt mapinitiator -c Client-77 -i 21:00:00:24:ff:67:37:07"
ms-cli EngineA "client mgt mapinitiator -c Client-77 -i 21:00:00:24:ff:67:37:06"
ms-cli EngineA "client mgt mapinitiator -c Client-77 -i 21:00:00:24:ff:69:d1:cb"
ms-cli EngineA "client mgt mapinitiator -c Client-77 -i 21:00:00:24:ff:69:d1:ca"
ms-cli EngineA "client mgt mapinitiator -c Client-78 -i 21:00:00:24:ff:66:f2:3b"
ms-cli EngineA "client mgt mapinitiator -c Client-78 -i 21:00:00:24:ff:66:f2:3a"
ms-cli EngineA "client mgt mapinitiator -c Client-78 -i 21:00:00:24:ff:69:b6:7f"
ms-cli EngineA "client mgt mapinitiator -c Client-78 -i 21:00:00:24:ff:69:b6:7e"

ms-cli EngineB "client mgt mapinitiator -c Client-77 -i 21:00:00:24:ff:71:64:e7"
ms-cli EngineB "client mgt mapinitiator -c Client-77 -i 21:00:00:24:ff:71:64:e6"
ms-cli EngineB "client mgt mapinitiator -c Client-77 -i 31:c7:00:24:ff:7b:d3:3f"
ms-cli EngineB "client mgt mapinitiator -c Client-77 -i 21:c7:00:24:ff:7b:d3:3f"
ms-cli EngineB "client mgt mapinitiator -c Client-78 -i 31:61:00:24:ff:af:04:07"
ms-cli EngineB "client mgt mapinitiator -c Client-78 -i 21:61:00:24:ff:af:04:07"
ms-cli EngineB "client mgt mapinitiator -c Client-78 -i 31:c9:00:24:ff:0a:96:0f"
ms-cli EngineB "client mgt mapinitiator -c Client-78 -i 21:c9:00:24:ff:0a:96:0f"

#map-target
echo "Waiting for map-target"
ms-cli EngineA "client itl maptarget -i 21:00:00:24:ff:67:37:07 -t FC-Target-1:1:1"
ms-cli EngineA "client itl maptarget -i 21:00:00:24:ff:67:37:06 -t FC-Target-1:1:2"
```

```
ms-cli EngineA "client itl maptarget -i 21:00:00:24:ff:66:f2:3b -t FC-Target-1:1:3"
ms-cli EngineA "client itl maptarget -i 21:00:00:24:ff:66:f2:3a -t FC-Target-1:1:4"
ms-cli EngineA "client itl maptarget -i 21:00:00:24:ff:69:d1:cb -t FC-Target-2:1:1"
ms-cli EngineA "client itl maptarget -i 21:00:00:24:ff:69:d1:ca -t FC-Target-2:1:2"
ms-cli EngineA "client itl maptarget -i 21:00:00:24:ff:69:b6:7f -t FC-Target-2:1:3"
ms-cli EngineA "client itl maptarget -i 21:00:00:24:ff:69:b6:7e -t FC-Target-2:1:4"

ms-cli EngineB "client itl maptarget -i 21:00:00:24:ff:71:64:e7 -t FC-Target-1:1:1"
ms-cli EngineB "client itl maptarget -i 21:00:00:24:ff:71:64:e6 -t FC-Target-1:1:2"
ms-cli EngineB "client itl maptarget -i 31:61:00:24:ff:af:04:07 -t FC-Target-1:1:3"
ms-cli EngineB "client itl maptarget -i 21:61:00:24:ff:af:04:07 -t FC-Target-1:1:4"
ms-cli EngineB "client itl maptarget -i 31:c7:00:24:ff:7b:d3:3f -t FC-Target-2:1:1"
ms-cli EngineB "client itl maptarget -i 21:c7:00:24:ff:7b:d3:3f -t FC-Target-2:1:2"
ms-cli EngineB "client itl maptarget -i 31:c9:00:24:ff:0a:96:0f -t FC-Target-2:1:3"
ms-cli EngineB "client itl maptarget -i 21:c9:00:24:ff:0a:96:0f -t FC-Target-2:1:4"

#map LUN 32
echo "Waiting for map LUN"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:67:37:07 -t FC-Target-1:1:1 -l LUN-A-SP1-0001 -I 0 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:67:37:07 -t FC-Target-1:1:1 -l LUN-A-SP1-0002 -I 1 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:67:37:06 -t FC-Target-1:1:2 -l LUN-A-SP1-0003 -I 0 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:67:37:06 -t FC-Target-1:1:2 -l LUN-A-SP1-0004 -I 1 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:69:d1:cb -t FC-Target-2:1:1 -l LUN-A-SP2-0001 -I 0 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:69:d1:cb -t FC-Target-2:1:1 -l LUN-A-SP2-0002 -I 1 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:69:d1:ca -t FC-Target-2:1:2 -l LUN-A-SP2-0003 -I 0 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:69:d1:ca -t FC-Target-2:1:2 -l LUN-A-SP2-0004 -I 1 -a rw"

ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:66:f2:3b -t FC-Target-1:1:3 -l LUN-A-SP1-0001 -I 0 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:66:f2:3b -t FC-Target-1:1:3 -l LUN-A-SP1-0002 -I 1 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:66:f2:3a -t FC-Target-1:1:4 -l LUN-A-SP1-0003 -I 0 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:66:f2:3a -t FC-Target-1:1:4 -l LUN-A-SP1-0004 -I 1 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:69:b6:7f -t FC-Target-2:1:3 -l LUN-A-SP2-0001 -I 0 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:69:b6:7f -t FC-Target-2:1:3 -l LUN-A-SP2-0002 -I 1 -a rw"
```

```

ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:69:b6:7e -t FC-Target-2:1:4
-l LUN-A-SP2-0003 -I 0 -a rw"
ms-cli EngineA "client itl maplun -i 21:00:00:24:ff:69:b6:7e -t FC-Target-2:1:4
-l LUN-A-SP2-0004 -I 1 -a rw"

ms-cli EngineB "client itl maplun -i 21:00:00:24:ff:71:64:e7 -t FC-Target-1:1:1
-l LUN-B-SP1-0001 -I 0 -a rw"
ms-cli EngineB "client itl maplun -i 21:00:00:24:ff:71:64:e7 -t FC-Target-1:1:1
-l LUN-B-SP1-0002 -I 1 -a rw"
ms-cli EngineB "client itl maplun -i 21:00:00:24:ff:71:64:e6 -t FC-Target-1:1:2
-l LUN-B-SP1-0003 -I 0 -a rw"
ms-cli EngineB "client itl maplun -i 21:00:00:24:ff:71:64:e6 -t FC-Target-1:1:2
-l LUN-B-SP1-0004 -I 1 -a rw"
ms-cli EngineB "client itl maplun -i 31:c7:00:24:ff:7b:d3:3f -t FC-Target-2:1:1
-l LUN-B-SP2-0001 -I 0 -a rw"
ms-cli EngineB "client itl maplun -i 31:c7:00:24:ff:7b:d3:3f -t FC-Target-2:1:1
-l LUN-B-SP2-0002 -I 1 -a rw"
ms-cli EngineB "client itl maplun -i 21:c7:00:24:ff:7b:d3:3f -t FC-Target-2:1:2
-l LUN-B-SP2-0003 -I 0 -a rw"
ms-cli EngineB "client itl maplun -i 21:c7:00:24:ff:7b:d3:3f -t FC-Target-2:1:2
-l LUN-B-SP2-0004 -I 1 -a rw"

ms-cli EngineB "client itl maplun -i 31:61:00:24:ff:af:04:07 -t FC-Target-1:1:3
-l LUN-B-SP1-0001 -I 0 -a rw"
ms-cli EngineB "client itl maplun -i 31:61:00:24:ff:af:04:07 -t FC-Target-1:1:3
-l LUN-B-SP1-0002 -I 1 -a rw"
ms-cli EngineB "client itl maplun -i 21:61:00:24:ff:af:04:07 -t FC-Target-1:1:4
-l LUN-B-SP1-0003 -I 0 -a rw"
ms-cli EngineB "client itl maplun -i 21:61:00:24:ff:af:04:07 -t FC-Target-1:1:4
-l LUN-B-SP1-0004 -I 1 -a rw"
ms-cli EngineB "client itl maplun -i 31:c9:00:24:ff:0a:96:0f -t FC-Target-2:1:3
-l LUN-B-SP2-0001 -I 0 -a rw"
ms-cli EngineB "client itl maplun -i 31:c9:00:24:ff:0a:96:0f -t FC-Target-2:1:3
-l LUN-B-SP2-0002 -I 1 -a rw"
ms-cli EngineB "client itl maplun -i 21:c9:00:24:ff:0a:96:0f -t FC-Target-2:1:4
-l LUN-B-SP2-0003 -I 0 -a rw"
ms-cli EngineB "client itl maplun -i 21:c9:00:24:ff:0a:96:0f -t FC-Target-2:1:4
-l LUN-B-SP2-0004 -I 1 -a rw"

echo "complete "
exit 0

```

Step 2 - Create Volumes on the Master Host System

The script *mk1vm_ms2500g2.sh* is executed on the Master Host System to create 38 logical volumes as follows:

1. Create Physical Volume:

Create 16 physical volumes using the pvcreate command.

2. Create Volumes Groups:

Create 1 volume group (vg11) using the vgcreate command over the 16 physical volumes: /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.

3. Create Logical Volumes:

Create 18 logical volumes with a volume capacity of 266GB, on vg11 for ASU-1.

Create 18 logical volumes with a volume capacity of 266GB, on vg11 for ASU-2.

Create 2 logical volumes with a volume capacity of 532GB, on vg11 for ASU-3.

4. Scan Physical Volumes, Volume Group, and Logical Volumes; and activate each Logical Volume:

Execute the *lv_scan.sh* on the Slave Host Systems to scan the physical volumes, volume group and logical volumes; and make each logical volume available (activate).

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

mk1vm_ms2500g2.sh

```
#!/bin/bash
#create_pv
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

#create_vg
vgcreate vg11 /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

#create_lv
lvcreate -n asu11 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu12 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu13 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu14 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu15 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu16 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu17 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu18 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu19 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu110 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu111 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu112 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu113 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu114 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu115 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu116 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu117 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu118 -i 16 -I 1024 -C y -L 266g vg11
```

```
lvcreate -n asu21 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu22 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu23 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu24 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu25 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu26 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu27 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu28 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu29 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu210 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu211 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu212 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu213 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu214 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu215 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu216 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu217 -i 16 -I 1024 -C y -L 266g vg11
lvcreate -n asu218 -i 16 -I 1024 -C y -L 266g vg11

lvcreate -n asu31 -i 16 -I 1024 -C y -L 532g vg11
lvcreate -n asu32 -i 16 -I 1024 -C y -L 532g vg11
```

lv_scan.sh

```
#!/bin/bash

echo '-----pvscan-----'
pvscan
echo '-----vgscan-----'
vgscan
echo '-----lvscan-----'
lvscan

lvchange -ay /dev/vg11/asu11
lvchange -ay /dev/vg11/asu12
lvchange -ay /dev/vg11/asu13
lvchange -ay /dev/vg11/asu14
lvchange -ay /dev/vg11/asu15
lvchange -ay /dev/vg11/asu16
lvchange -ay /dev/vg11/asu17
lvchange -ay /dev/vg11/asu18
lvchange -ay /dev/vg11/asu19
lvchange -ay /dev/vg11/asu110
lvchange -ay /dev/vg11/asu111
lvchange -ay /dev/vg11/asu112
lvchange -ay /dev/vg11/asu113
lvchange -ay /dev/vg11/asu114
lvchange -ay /dev/vg11/asu115
lvchange -ay /dev/vg11/asu116
lvchange -ay /dev/vg11/asu117
lvchange -ay /dev/vg11/asu118

lvchange -ay /dev/vg11/asu21
lvchange -ay /dev/vg11/asu22
lvchange -ay /dev/vg11/asu23
lvchange -ay /dev/vg11/asu24
lvchange -ay /dev/vg11/asu25
lvchange -ay /dev/vg11/asu26
lvchange -ay /dev/vg11/asu27
lvchange -ay /dev/vg11/asu28
```

```
lvchange -ay /dev/vg11/asu29
lvchange -ay /dev/vg11/asu210
lvchange -ay /dev/vg11/asu211
lvchange -ay /dev/vg11/asu212
lvchange -ay /dev/vg11/asu213
lvchange -ay /dev/vg11/asu214
lvchange -ay /dev/vg11/asu215
lvchange -ay /dev/vg11/asu216
lvchange -ay /dev/vg11/asu217
lvchange -ay /dev/vg11/asu218

lvchange -ay /dev/vg11/asu31
lvchange -ay /dev/vg11/asu32
```


APPENDIX E: CONFIGURATION INVENTORY

An inventory of the Tested Storage Configuration was collected during the execution of the scripts *ms2500g2_test_phase1.sh* and *ms2500g2_test_phase2.sh*. The following log files were generated:

- *volume_listing_start.txt* List of configured storage before the INIT Phase.
- *profile_start_ms2500g2.txt* List of configured volumes before the INIT Phase.
- *volume_listing_end.txt* List of configured storage after TSC restart.
- *profile_end_ms2500g2.txt* List of configured volumes after TSC restart.

The above log files are included in the Supporting Files (see Appendix A).

APPENDIX F: WORKLOAD GENERATOR

The host parameters for the SPC-1 workload generator were defined using the script ***2host.HST***.

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

The initial test phases of the benchmark are executed using the scripts ***ms2500g2_test_phase1.sh*** and ***ms2500g2_test_persist1.sh*** until the Persistence Test shutdown. Once the TSC has been restarted, the PERSIST_2 test phase is executed using the script ***ms2500g2_test_phase2.sh***.

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

2host.HST

```
-- SMALL_HOST definition
LOGIN=root
PASSWORD=passwd
CONFIG=/spc-1_V3_2500G2/v302
EXEC=Supported/RHEL/7_1/spc1_v3.0.2
OUTPUT=/spc-1_V3_2500G2/v302/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost

LOGIN=root
```

```
PASSWORD=passwd
CONFIG=/spc-1_v3_2500G2/v302
EXEC=Supported/RHEL/7_1/spc1_v3.0.2
OUTPUT=/spc-1_v3_2500G2/v302/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=172.0.65.78
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=172.0.65.78
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=172.0.65.78
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=172.0.65.78
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=172.0.65.78
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=172.0.65.78
```

slave_asu.asu

```
ASU=1
OFFSET=0
SIZE=0
DEVICE=/dev/vg11/asu11
DEVICE=/dev/vg11/asu12
DEVICE=/dev/vg11/asu13
DEVICE=/dev/vg11/asu14
DEVICE=/dev/vg11/asu15
DEVICE=/dev/vg11/asu16
DEVICE=/dev/vg11/asu17
DEVICE=/dev/vg11/asu18
DEVICE=/dev/vg11/asu19
DEVICE=/dev/vg11/asu110
DEVICE=/dev/vg11/asu111
DEVICE=/dev/vg11/asu112
DEVICE=/dev/vg11/asu113
DEVICE=/dev/vg11/asu114
DEVICE=/dev/vg11/asu115
DEVICE=/dev/vg11/asu116
DEVICE=/dev/vg11/asu117
DEVICE=/dev/vg11/asu118
--
ASU=2
```

```
OFFSET=0
SIZE=0
DEVICE=/dev/vg11/asu21
DEVICE=/dev/vg11/asu22
DEVICE=/dev/vg11/asu23
DEVICE=/dev/vg11/asu24
DEVICE=/dev/vg11/asu25
DEVICE=/dev/vg11/asu26
DEVICE=/dev/vg11/asu27
DEVICE=/dev/vg11/asu28
DEVICE=/dev/vg11/asu29
DEVICE=/dev/vg11/asu210
DEVICE=/dev/vg11/asu211
DEVICE=/dev/vg11/asu212
DEVICE=/dev/vg11/asu213
DEVICE=/dev/vg11/asu214
DEVICE=/dev/vg11/asu215
DEVICE=/dev/vg11/asu216
DEVICE=/dev/vg11/asu217
DEVICE=/dev/vg11/asu218
--
ASU=3
OFFSET=0
SIZE=0
DEVICE=/dev/vg11/asu31
DEVICE=/dev/vg11/asu32
```

ms2500g2_test_phase1.sh

```
#!/bin/bash

date

echo "Collect_MS2500G2_info"
ssh root@172.0.64.85 >/dev/null 2>&1 <<eeooff
/root/profile_ms2500g2.sh start
exit
eeooff
echo "-----"

echo "Collect_volume_info"
/v302/volume_list.sh start
echo "-----"

echo "Init tests"
/spc-1_V3_2500G2/v302/Supported/RHEL/7_1/spc1_v3.0.2 -run SPC1_INIT -iops 1600
-storage slave_asu.asu -output /spc-1_V3_2500G2/v302/Output -master 2host.HST
echo "-----"

echo "Verify"
/spc-1_V3_2500G2/v302/Supported/RHEL/7_1/spc1_v3.0.2 -run SPC1_VERIFY -iops 100
-storage slave_asu.asu -output /spc-1_V3_2500G2/v302/Output
echo "-----"

echo "Start metrics test"
/spc-1_V3_2500G2/v302/Supported/RHEL/7_1/spc1_v3.0.2 -run SPC1_METRICS -iops
300000 -storage slave_asu.asu -output /spc-1_V3_2500G2/v302/Output -master
2host.HST
echo "metrics test over"
```

```
echo "-----"

echo "Start test verify"
/spc-1_V3_2500G2/v302/Supported/RHEL/7_1/spc1_v3.0.2 -run SPC1_VERIFY -iops 100
-storage slave_asu.asu -output /spc-1_V3_2500G2/v302/Output
echo "-----"

echo "test_phase1 test over"

date
```

ms2500g2_test_persist1.sh

```
#!/bin/bash

date

echo "Start test persist1"
/spc-1_V3_2500G2/v302/Supported/RHEL/7_1/spc1_v3.0.2 -run SPC1_PERSIST_1 -iops
75000 -storage slave_asu.asu -output /spc-1_V3_2500G2/v302/Output -master
2host.HST
echo "-----"
echo "test_persist1 test over"

date
```

ms2500g2_test_phase2.sh

```
#!/bin/bash

date

echo "Collect_MS2500G2_info"
ssh root@172.0.64.85 >/dev/null 2>&1 <<eeooff
/root/profile_ms2500g2.sh end
exit
eeooff
echo "-----"

echo "Collect_volume_info"
/v302/volume_list.sh end
echo "-----"

echo "start test persist2"
/spc-1_V3_2500G2/v302/Supported/RHEL/7_1/spc1_v3.0.2 -run SPC1_PERSIST_2 -iops
75000 -storage slave_asu.asu -output /spc-1_V3_2500G2/v302/Output -master
2host.HST
echo "-----"
echo "test_phase2 test over"

date
```