SPC BENCHMARK 1™

FULL DISCLOSURE REPORT

HUAWEI TECHNOLOGIES CO., LTD.
HUAWEI OCEANSTOR DORADO18000 V3

SPC-1 V3.8

SUBMISSION IDENTIFIER: A31017

SUBMITTED FOR REVIEW: OCTOBER 30, 2018

REVISED: NOVEMBER 6, 2018

Second Edition – November 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 United States. 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, FusionServer™ and OceanStor 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.spcresults.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

November 6, 2018

I verified the SPC Benchmark 1™ (SPC-1™ Revision3.8) test execution and performance results of the following Tested Storage Product:

Huawei OceanStor Dorado18000 V3

The results were:

SPC-1 IOPS™	**7,000,565**
SPC-1 Price-Performance™	**$378.40/SPC-1 KIOPS™**
SPC-1 IOPS™ Response Time	0.814 ms
SPC-1 Overall Response Time	0.554 ms
SPC-1 ASU Capacity	211,316 GB
SPC-1 Space Effectiveness Ratio	0.73
SPC-1 ASU Price	$12.54/GB
SPC-1 Total System Price	$2,648,965.96

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 3.0.2-1-g823a. 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.spcresults.org** under the Submission Identifier **A31017**.

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

A31017 *Huawei OceanStor Dorado 18000 V3* *p.2*

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

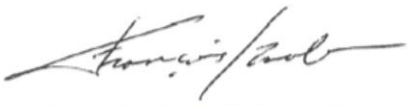
- The physical capacity of the data repository;
- The total capacity of the Application Storage Unit (ASU);
- The accuracy of the Benchmark Configuration diagram;
- The tuning parameters used to configure the Benchmark Configuration;
- The Workload Generator commands used to execute the testing;
- The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



François Raab, Certified SPC Auditor

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

LETTER OF GOOD FAITH

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

Date: October 27, 2018

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 Dorado18000 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 V3.6 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: 2018-10-27

Meng Guangbin
President of Storage Product Line

SPC BENCHMARK 1™

EXECUTIVE SUMMARY

HUAWEI TECHNOLOGIES CO., LTD.
HUAWEI OCEANSTOR DORADO18000 V3

SPC-1 IOPS™	**7,000,565**
SPC-1 Price-Performance™	**$378.40/SPC-1 KIOPS™**
SPC-1 IOPS™ Response Time	0.814 ms
SPC-1 Overall Response Time	0.554 ms
SPC-1 ASU Capacity	211,316 GB
SPC-1 Space Effectiveness Ratio	0.73
SPC-1 ASU Price	$12.54/GB
SPC-1 Total System Price	$2,648,965.96
Data Protection Level	Protect2 (RAID 6)
Physical Storage Capacity	445,440 GB
Pricing Currency / Target Country	U.S. Dollars / USA

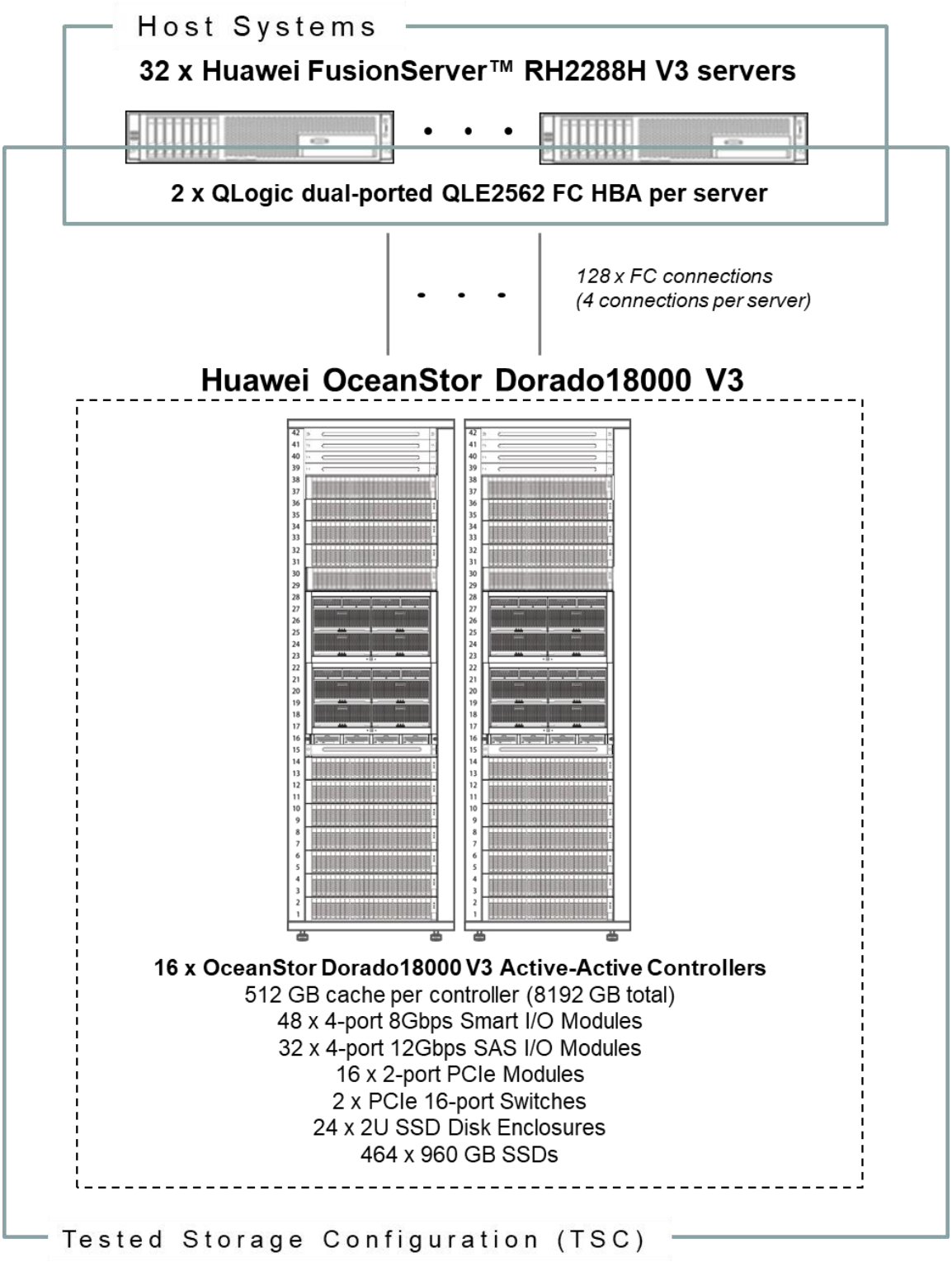
SPC-1 V3.8

SUBMISSION IDENTIFIER: A31017

SUBMITTED FOR REVIEW: OCTOBER 30, 2018

REVISED: NOVEMBER 6, 2018

Benchmark Configuration Diagram



Tested Storage Product Description

Huawei OceanStor Dorado V3 all-flash storage is purpose-built for enterprise-class mission-critical business. Powered by Huawei FlashLink™ technology, it is scalable to 16 controllers and supports up to 7 million IOPS, boosting application performance threefold with inline compression enabled. It employs gateway-free active-active technology to ensure RPO=0 and RTO≈0, and can be upgraded to the 3DC solution to achieve 99.9999% availability.

The OceanStor Dorado V3 is ideal for use with databases, virtual desktop infrastructure (VDI), virtual server infrastructure (VSI), and SAP HANA. It has been designed to facilitate the transition to all-flash for customers in the finance, manufacturing, and telecom sectors.

For more details, visit:

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

Priced Storage Configuration Components

64 x QLogic dual-ported QLE2562 FC HBA
16 x OceanStor™ Dorado 18000 V3 Active-Active Controllers, each with: **512 GB cache (8192 GB total)** **48 x 4-port 8Gbps Smart I/O Modules** **32 x 4-port 12Gbps SAS I/O Modules** **16 x 2-port PCIe Modules** **8 x 2U Disk Enclosures, each with:** **8 x 960 GB SSDs** **16 x 2U Disk Enclosures, each with:** **25 x 960 GB SSDs** **2 x PCIe 16-port Switches**

Storage Configuration Pricing

	Description	Qty	Unit Price	Ext. Price	Disc.	Disc. Price
Hardware & Software						
D18V3-4C2T-AC	Dorado18000 V3 Engine (6U, Four Controller, AC\240HVDC, 4*512GB Cache, 12*4-port 8G FC, 8*4-port 12G SAS,SPE73C0600)	4	411,608.00	1,646,432.00	50%	823,216.00
DV5-LPU5P2PCIE	2 port PCIe I/O module(with two NT Ports)	16	681.00	10,896.00	0%	10,896.00
D18V3-SSD-SAS-960G	960GB SSD SAS Disk Unit(2.5")	464	4,569.25	2,120,132.00	48%	1,102,468.64
D18V3-SDAE25U2-AC	Disk Enclosure(2U,AC\240HVDC,2.5",Expanding Module,25 Disk Slots,without Disk Unit,DAE52525U2)	24	4,454.00	106,896.00	48%	55,585.92
D18V3-RACK-SYS-AC	OceanStor Dorado18000 V3 Series AC System Cabinet	2	13,127.00	26,254.00	48%	13,652.08
N8GHBA000	QLOGIC QLE2562 HBA Card,PCIE,8Gbps DualPort ,Fiber Channel Multimode LC Optic Interface,English Manual, No Drive CD	64	1,698.00	108,672.00	0%	108,672.00
SWITCH-V5H2	PCIe 3.0 Switch(AC\240HVDC,8GB Cache,16 Port,SWE1600P08)	2	13,336.00	26,672.00	0%	26,672.00
SN2F01FCPC	Patch Cord,DLC/PC,DLC/PC,Multi-mode,3m,A1a.2,2mm,42mm DLC,OM3 bending insensitive	128	20.00	2,560.00	0%	2,560.00
D18V3-LBSB	Basic Software Suite Unlimited Capacity License (Including OceanStor OS,Device Manager,SmartThin,SmartCompression,SmartDedup,SmartQoS,SmartMigration,SmartVirtualization,HyperMetro,HyperClone,HyperSnap,HyperReplication, SystemReporter,eService, and UltraPath)	1	521,310.40	521,310.40	50%	260,655.20
				Hardware & Software Subtotal		**2,404,377.84**
Support & Maintenance						
02351WXJ-88134ULF-36	OceanStor Dorado18000 V3 Engine(6U,Four Controller,AC\240HVDC,4*512GB Cache,32*8G FC,32*port SAS,SPE73C0600)-Hi-Care Onsite Premier 24x7x4H Engineer Onsite Service-36Month(s)	4	17,202.60	68,810.40	0%	68,810.40
02352ANB-88134ULF-36	960GB SSD SAS Disk Unit(2.5")-Hi-Care Onsite Premier 24x7x4H Engineer Onsite Service-36Month(s)	464	122.40	56,793.60	0%	56,793.60
88034SDG-88134UHK-36	Basic Software Licenses (Including OceanStor OS,DeviceManager, SmartThin, SmartMigration, SmartDedupe, SmartCompression,SmartQoS and SystemReporter)-Hi-Care Application Software Upgrade Support Service-36Month(s)	1	71,235.00	71,235.00	0%	71,235.00
8812175865	OceanStor Dorado18000 V3 Installation Service - Engineering	1	47,749.12	47,749.12	0%	47,749.12
				Support & Maintenance Subtotal		**244,588.12**
SPC-1 Total System Price						**2,648,965.96**
SPC-1 IOPS™						7,000,565
SPC-1 Price-Performance™ ($/SPC-1 KIOPS™)						**378.40**
SPC-1 ASU Capacity (GB)						211,316
SPC-1 ASU Price ($/GB)						**12.54**

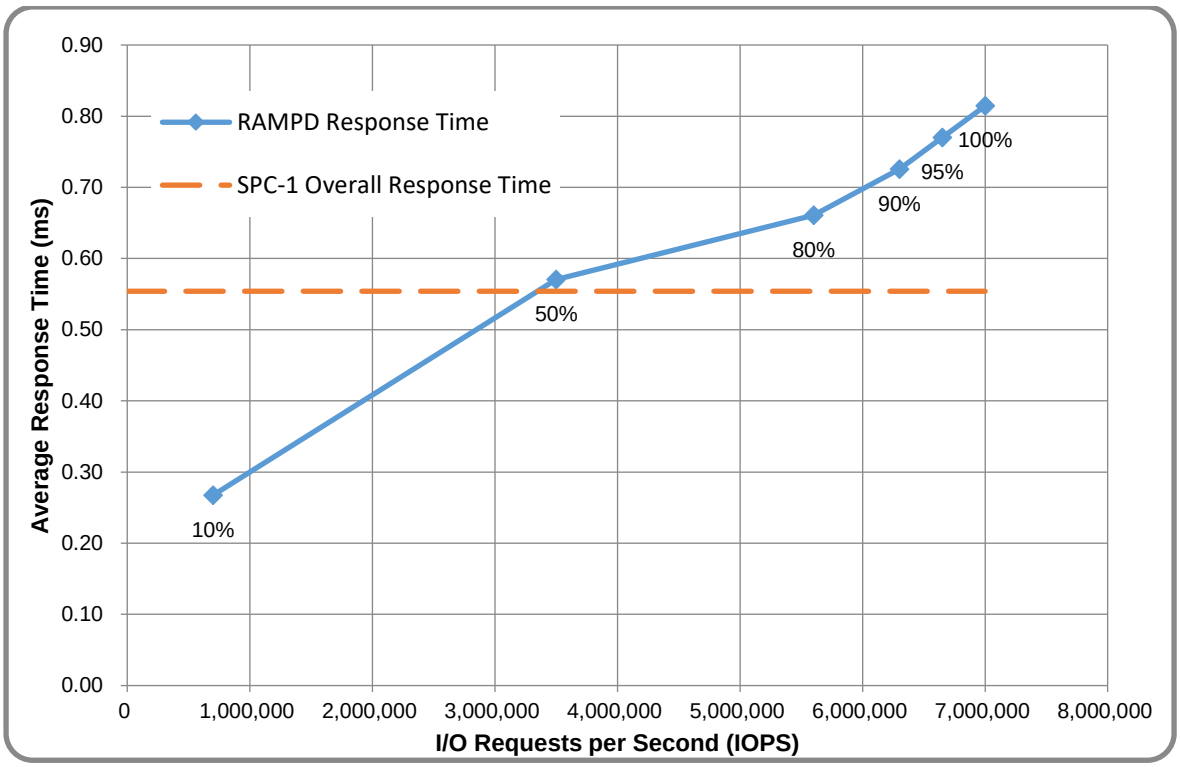
Third-Party Reseller: Huawei Technologies Co., Ltd. only sells its products to third-party resellers who, in turn, sell those products to U.S. customers. The above reflects the pricing quoted by one of those third-party resellers. See Appendix B of the Full Disclosure Report for a copy of the third-party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



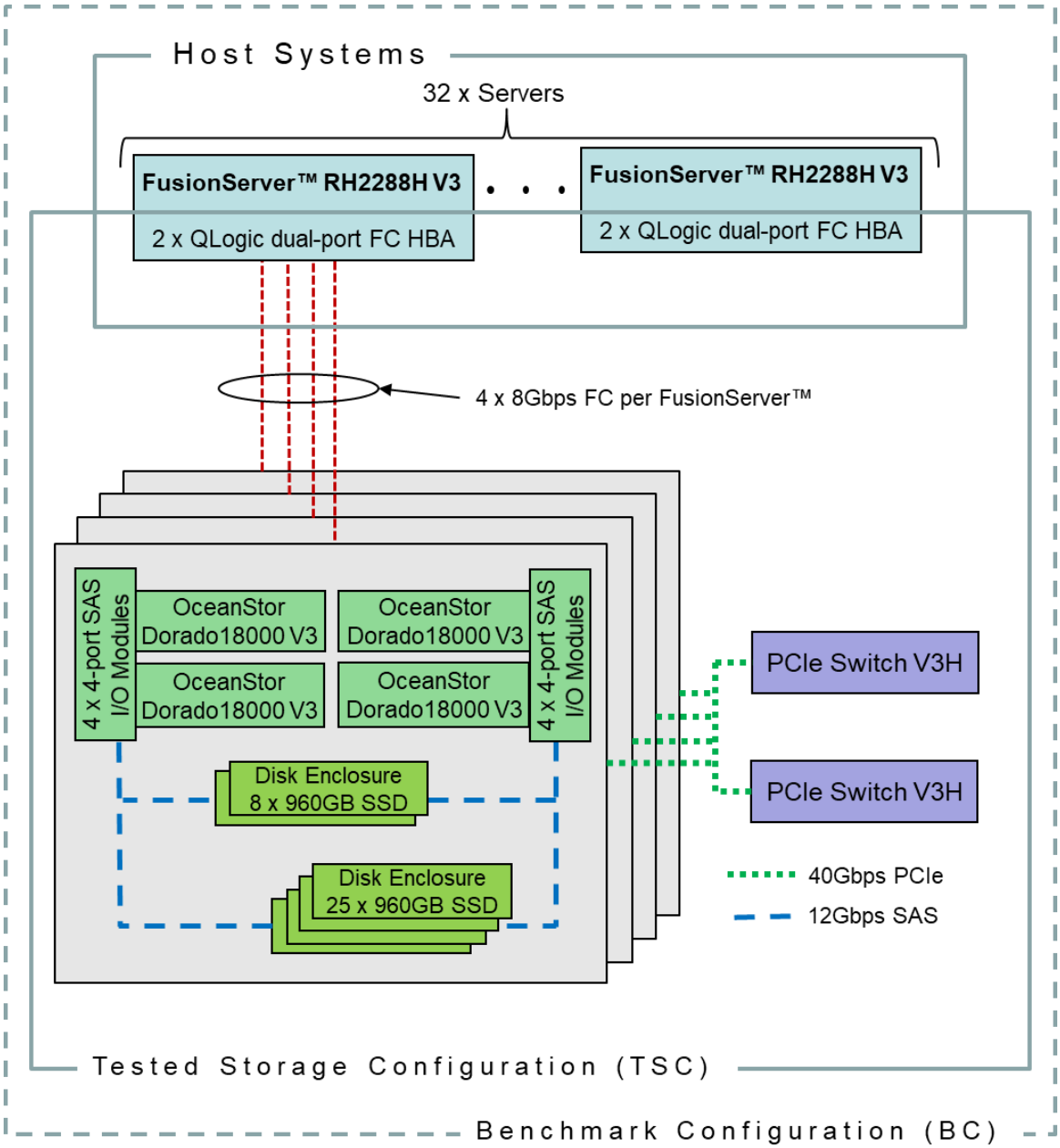
Contact Information	
Test Sponsor Primary Contact	Huawei Technologies Co., Ltd. – www.huawei.com Zhong Xu – xuzhong@huawei.com
SPC Auditor	InfoSizing – www.sizing.com Francois Raab – francois@sizing.com

Revision Information	
SPC Benchmark 1™ Revision	V3.8.0
SPC-1 Workload Generator Revision	3.0.2-1-g823a
Publication Revision History	First Edition: October 30, 2018 Second Edition: November 6, 2018

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 16 Huawei OceanStor Dorado18000 V3, driven by 32 host systems (Huawei

FusionServer RH2288H V3). The OceanStor controllers were grouped in sets of four, forming four OceanStor Engines. Each FusionServer host system connected one-to-one to each OceanStor Engine. That connection was established via a port from one of the two dual-port Fibre Chanel HBAs on the FusionServer; and a port from one of the twelve 4-port Smart I/O Modules in each of the OceanStor Engine. These Fibre Chanel paths operated 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
32 x Huawei FusionServer™ RH2288H V3 2 x Intel® Xeon® E5-2667 v4 (3.2 GHz, 8 Cores, 25 MB L3) 128 GB Main Memory Red Hat Enterprise Linux 7.1 x86-64
Priced Storage Configuration
64 x QLogic dual-ported QLE2562 FC HBA
16 x OceanStor™ Dorado 18000 V3 Active-Active Controllers, each with: 512 GB cache (8192 GB total) 48 x 4-port 8Gbps Smart I/O Modules 32 x 4-port 12Gbps SAS I/O Modules 16 x 2-port PCIe Modules 8 x 2U Disk Enclosures, each with: 8 x 960 GB SSDs 16 x 2U Disk Enclosures, each with: 25 x 960 GB SSDs 2 x PCIe 16-port 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
32 x 4-port 8Gbps Smart I/O Modules	48 x 4-port 8Gbps Smart I/O Modules	The number of configured FC I/O Modules in each Engine was incorrectly reported at 8 (32 total) instead of 12 (48 total).

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	5,282.9	5,282.9	95,092.2	45.00%
ASU-2	18	5,282.9	5,282.9	95,092.2	45.00%
ASU-3	2	10,565.7	10,565.7	21,131.4	10.00%
			SPC-1 ASU Capacity	**211,315.8**	

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
960GB SSD	464	960.0	445,440.0
		Total Physical Storage Capacity	**445,440.0**
		Physical Capacity Utilization	**47.43%**

Data Protection

The data protection level used for all logical volumes was **Protected 2**, which was accomplished by configuring 8 pools of 58 drives into 16 RAID 6 arrays.

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).

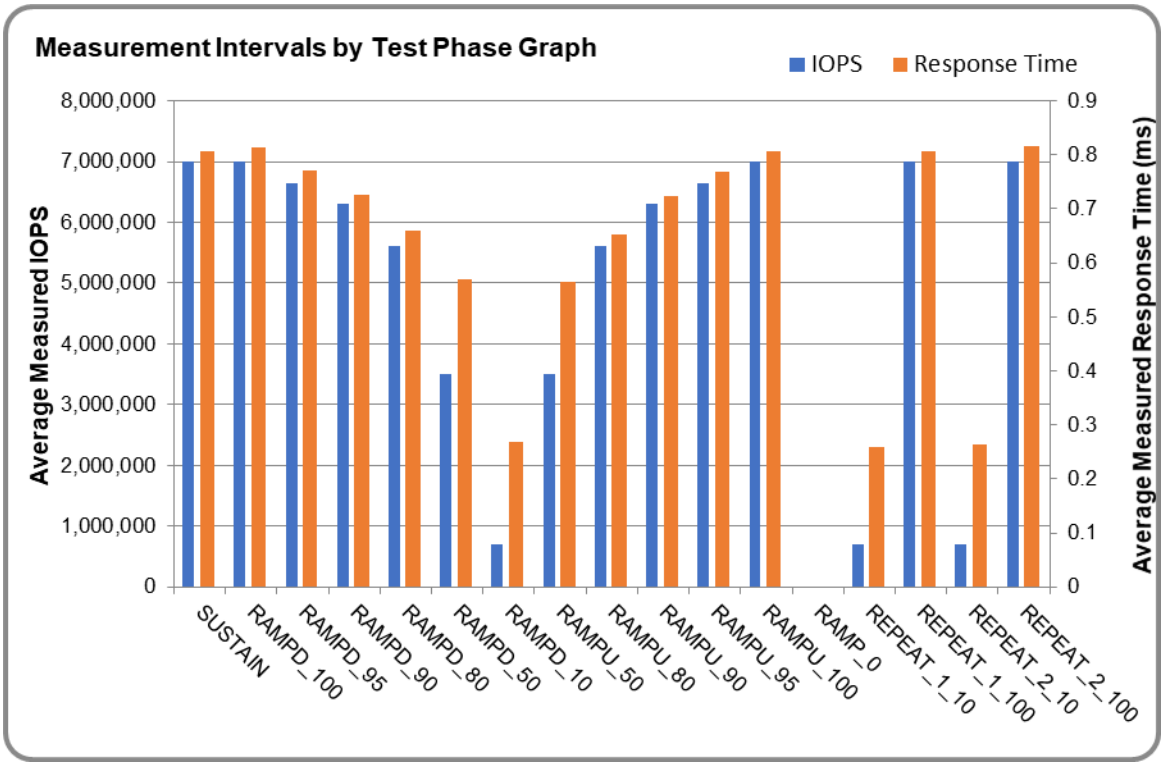
Primary Metrics Test Phases

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

None.

SUSTAIN Test Phase

SUSTAIN – Results File

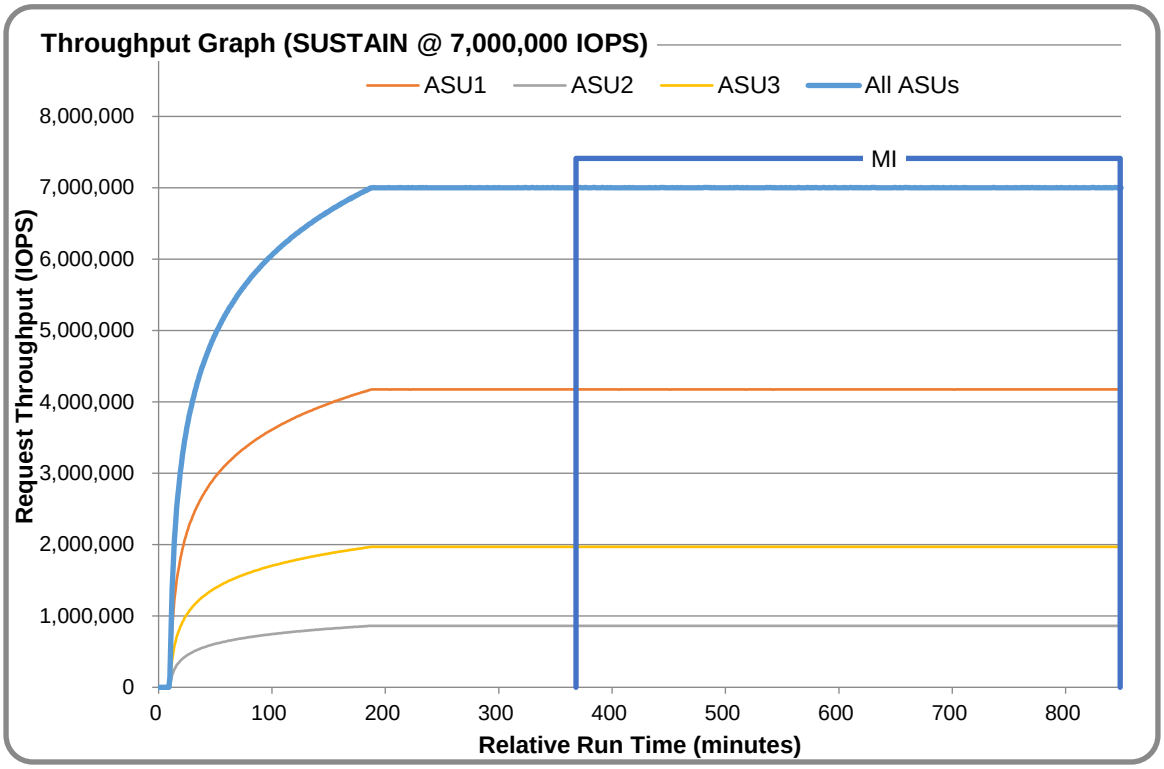
The results file generated during the execution of the SUSTAIN Test Phase is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

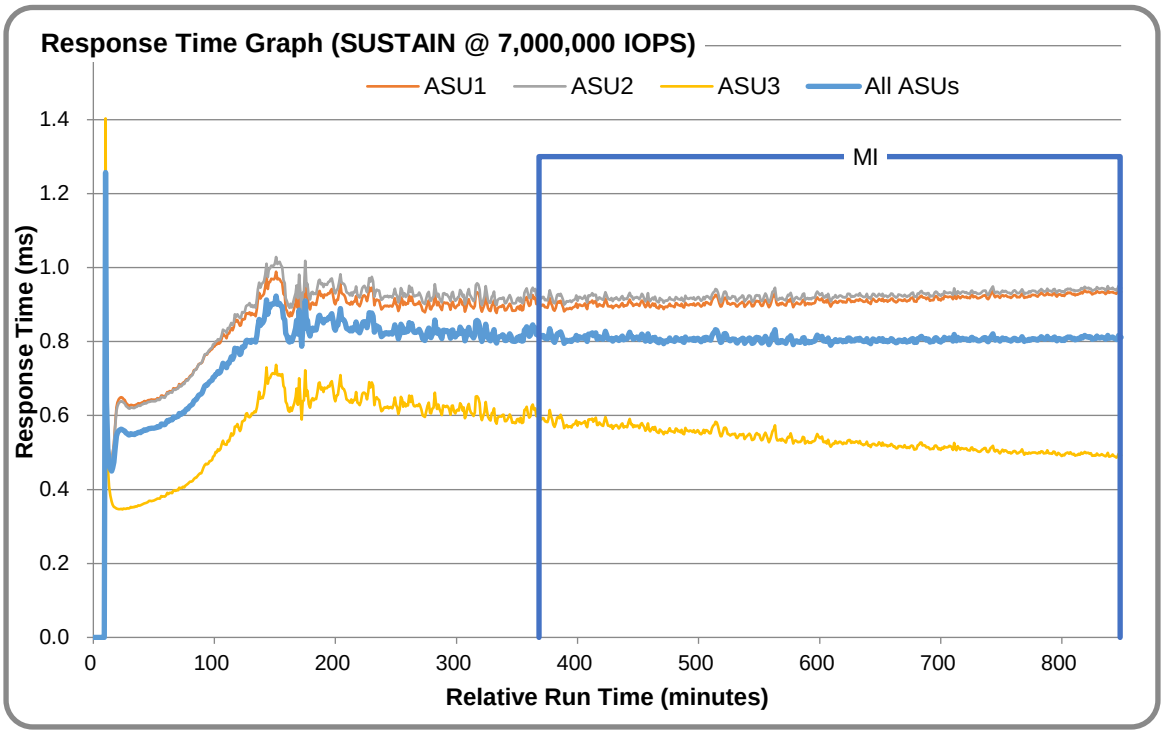
SUSTAIN – Execution Times

Interval	Start Time	End Time	Duration
Transition Period	19-Oct-18 22:13:46	20-Oct-18 04:13:46	6:00:00
Measurement Interval	20-Oct-18 04:13:46	20-Oct-18 12:13:47	8:00:01

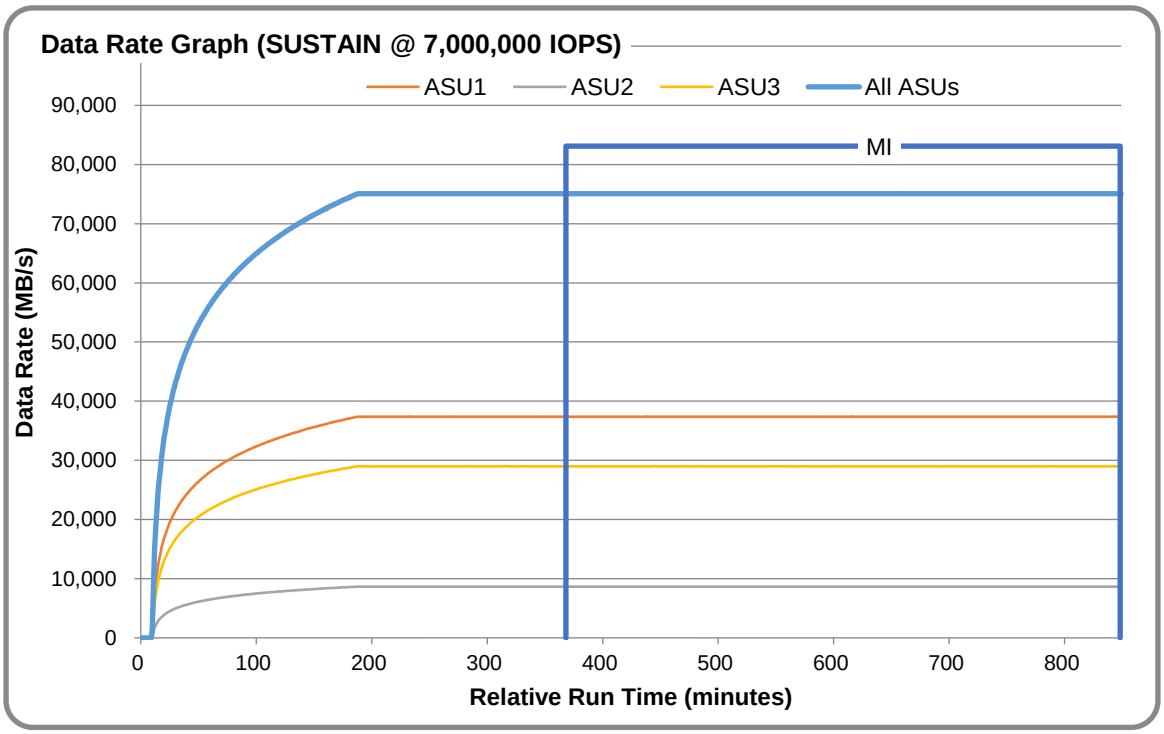
SUSTAIN – Throughput Graph



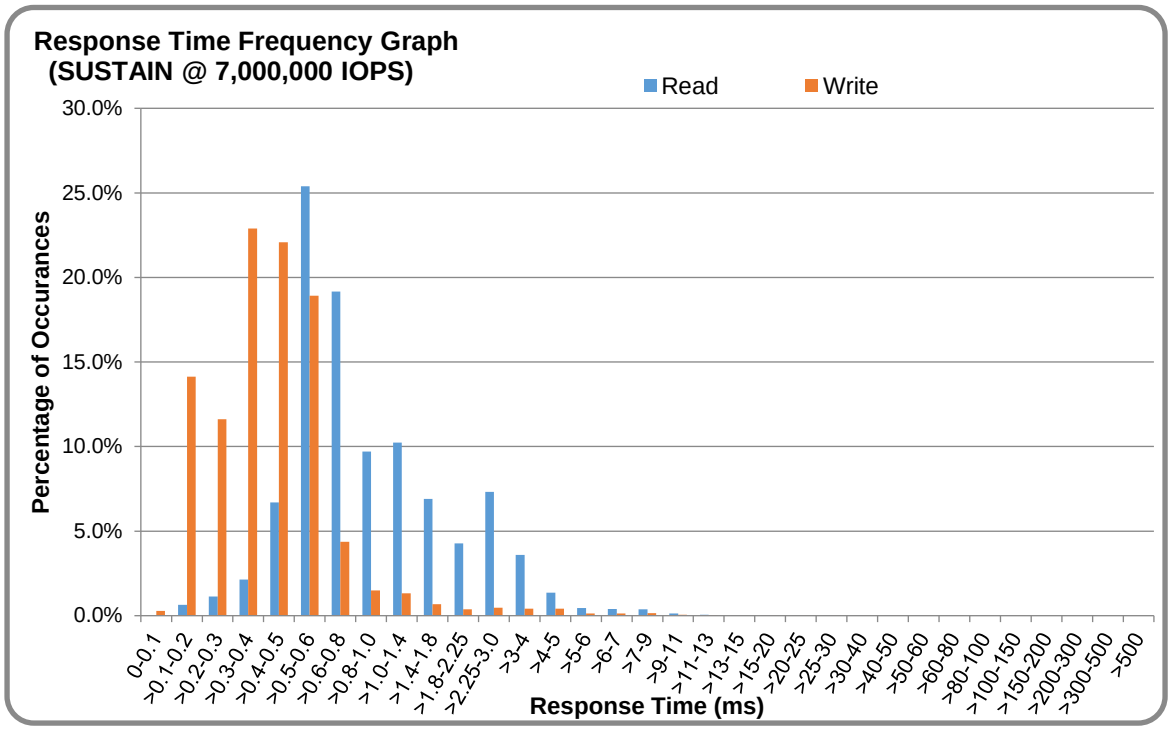
SUSTAIN – Response Time Graph



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	0.0002	0.0001	0.0002	0.0001	0.0004	0.0002	0.0003	0.0001
Difference	0.005%	0.001%	0.003%	0.000%	0.005%	0.005%	0.005%	0.002%

RAMPD_100 Test Phase

RAMPD_100 – Results File

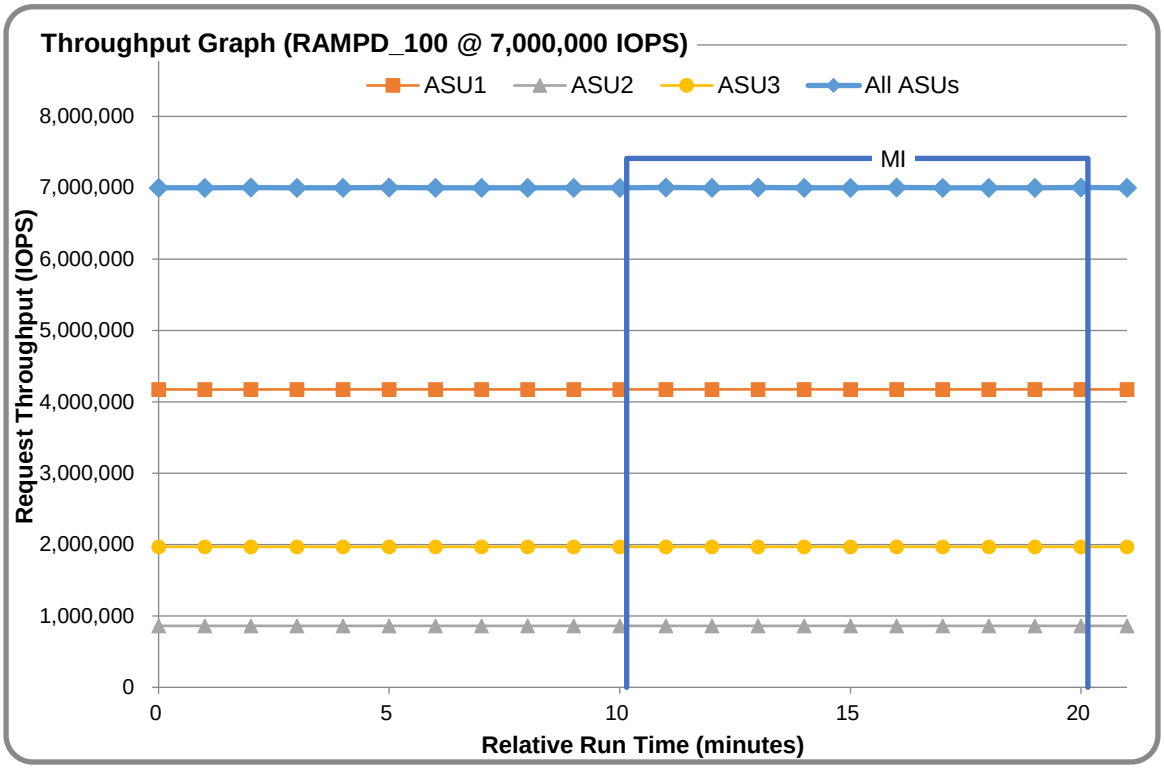
The results file generated during the execution of the RAMPD_100 Test Phase is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

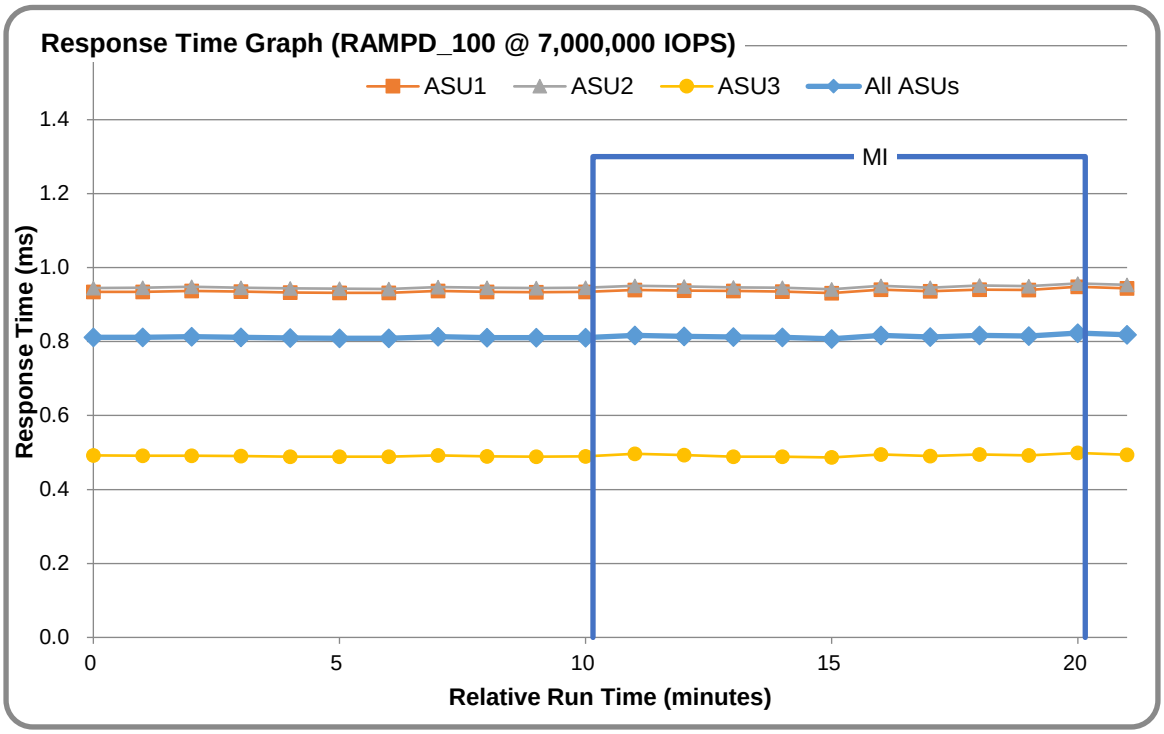
RAMPD_100 – Execution Times

Interval	Start Time	End Time	Duration
Transition Period	20-Oct-18 12:14:46	20-Oct-18 12:24:46	0:10:00
Measurement Interval	20-Oct-18 12:24:46	20-Oct-18 12:34:47	0:10:01

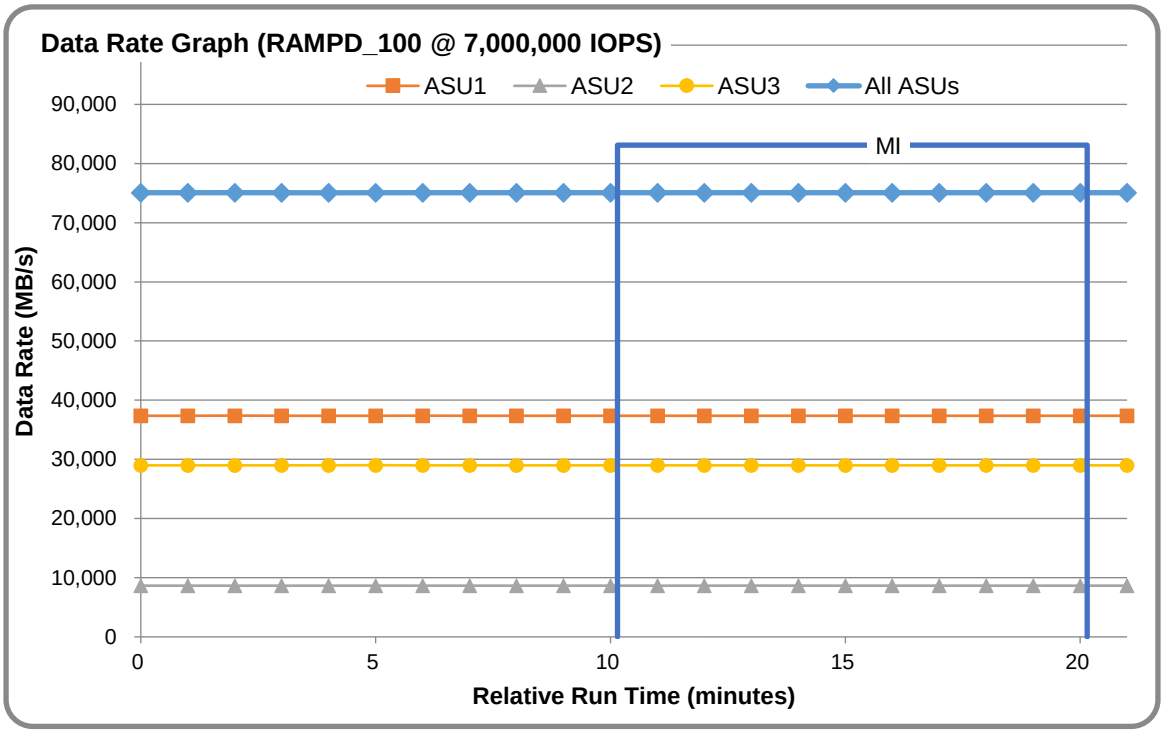
RAMPD_100 – Throughput Graph



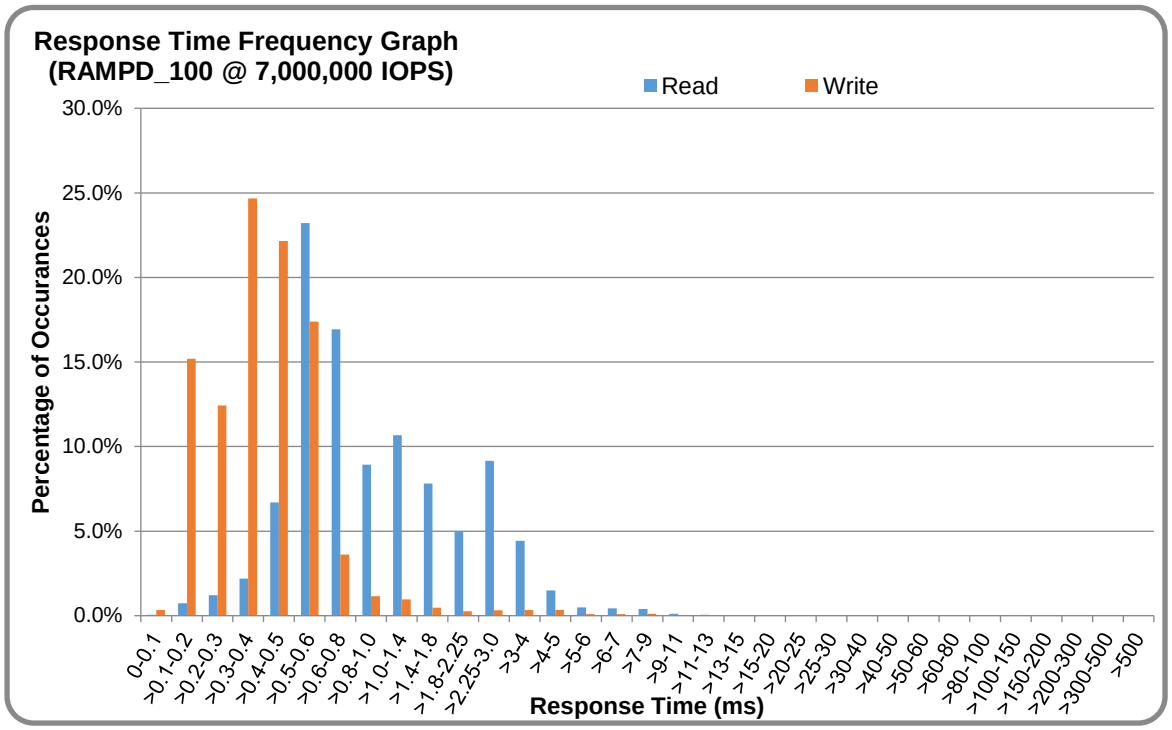
RAMPD_100 – Response Time Graph



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	0.0002	0.0001	0.0001	0.0001	0.0003	0.0002	0.0002	0.0001
Difference	0.002%	0.002%	0.008%	0.000%	0.016%	0.008%	0.007%	0.001%

RAMPD_100 – I/O Request Summary

I/O Requests Completed in the Measurement Interval	4,200,332,535
I/O Requests Completed with Response Time <= 30 ms	4,200,299,786
I/O Requests Completed with Response Time > 30 ms	32,749

Response Time Ramp Test

Response Time Ramp Test – Results File

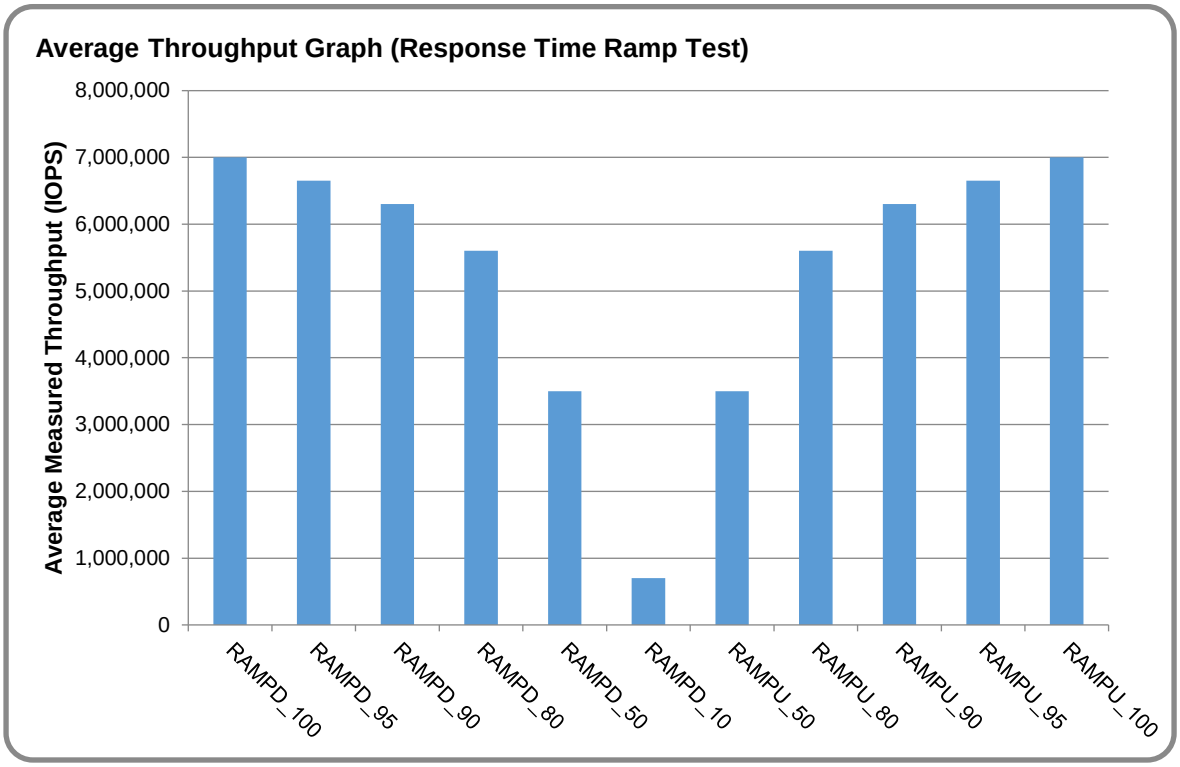
The results file generated during the execution of the Response Time Ramp Test is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

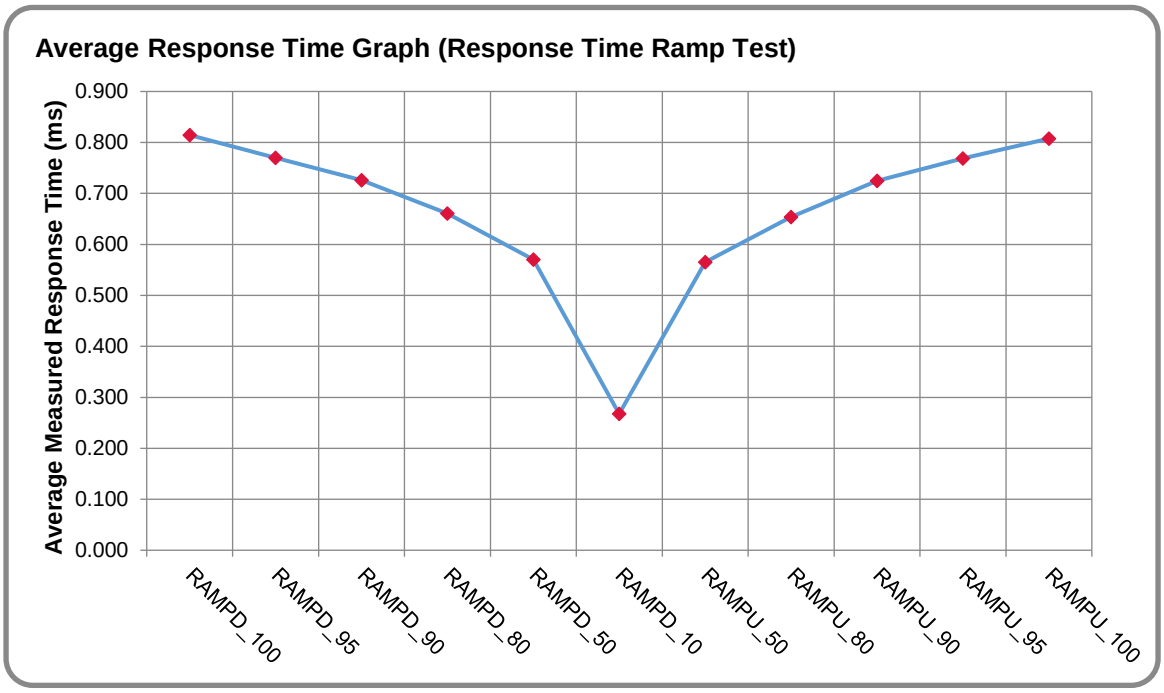
Response Time Ramp Test – Phases

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).

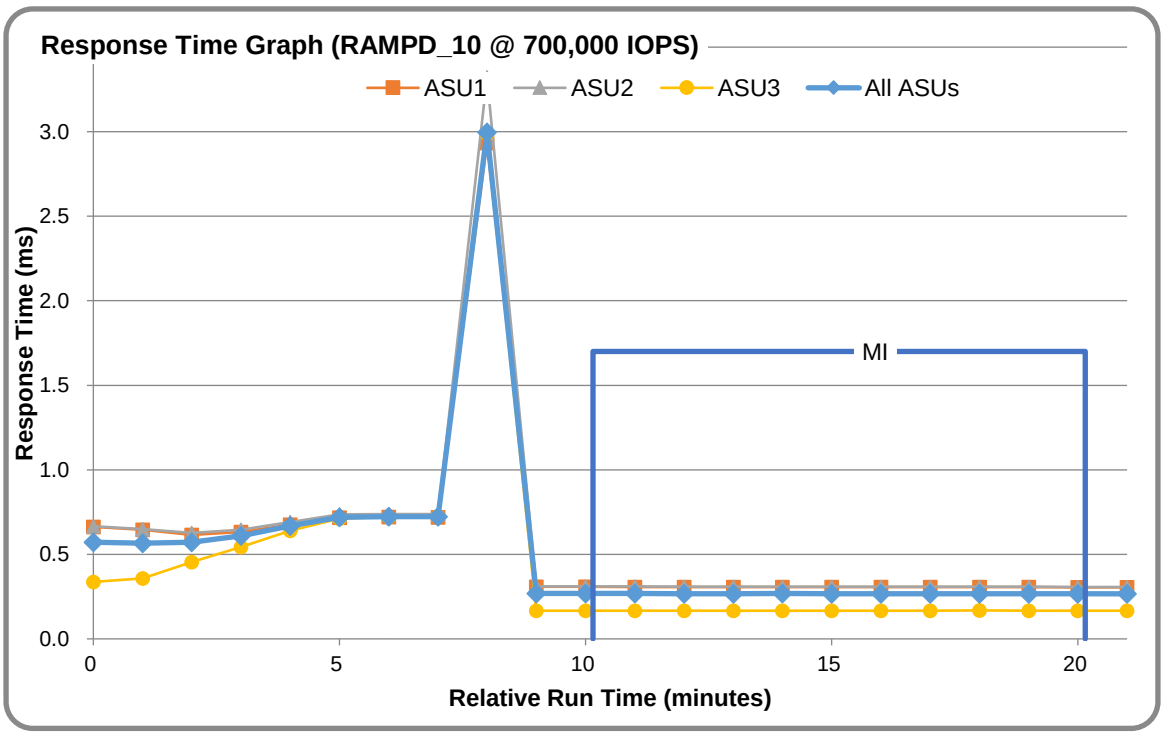
Response Time Ramp Test – Average Throughput Graph



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:

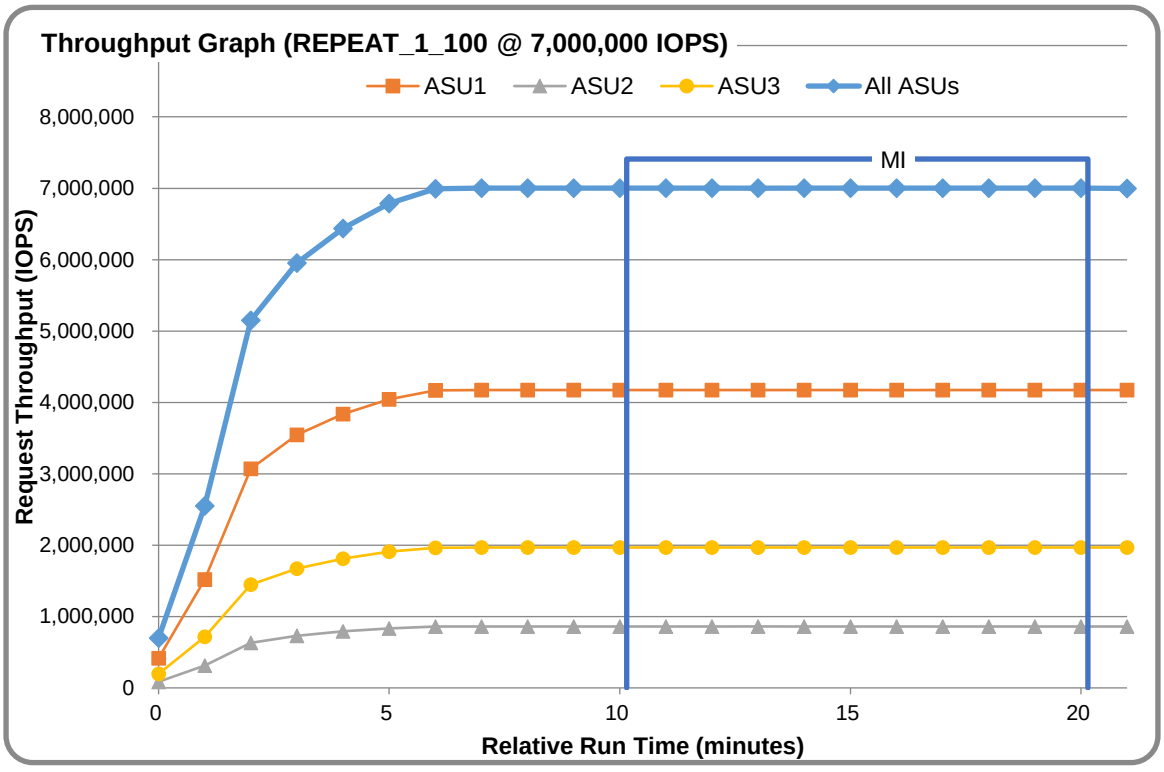
- **SPC1_METRICS_0_Raw_Results.xlsx**

Repeatability Test Results

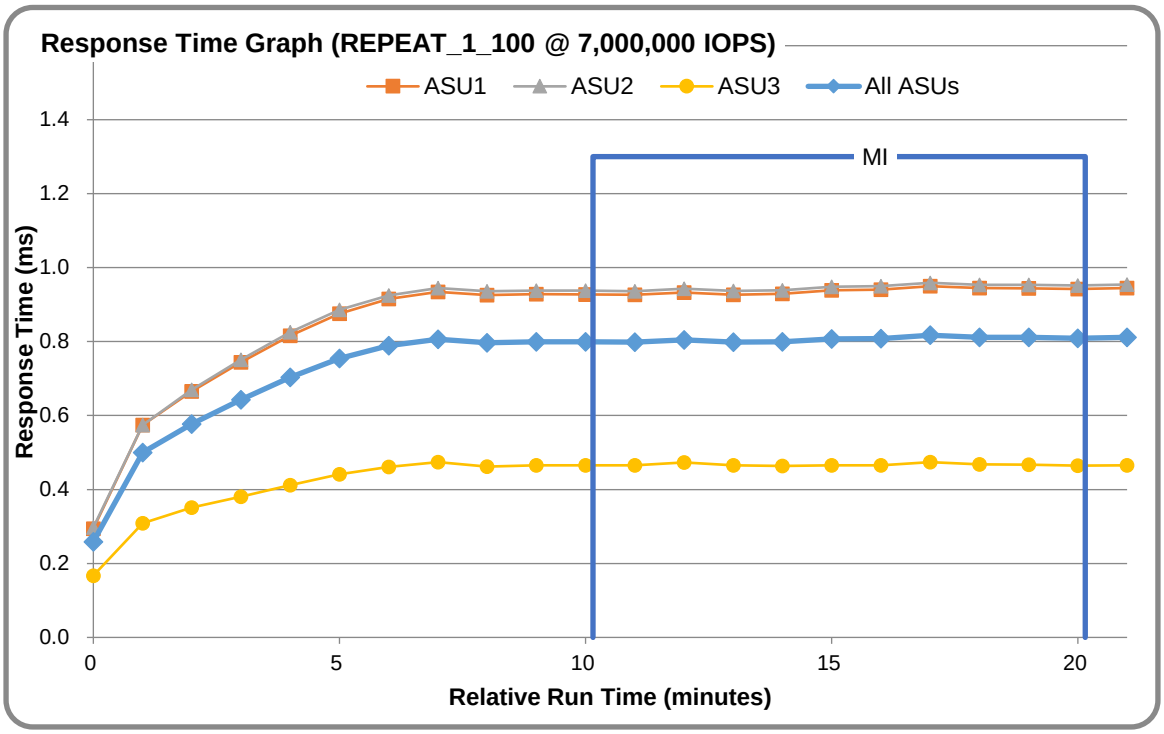
The throughput measurements for the Response Time Ramp Test (RAMPD) and the Repeatability Test Phases (REPEAT_1 and REPEAT_2) are listed in the tables below.

Test Phase	100% IOPS	10% IOPS
RAMPD	7,000,565.7	700,057.6
REPEAT_1	7,000,627.7	700,027.6
REPEAT_2	7,000,385.2	700,040.4

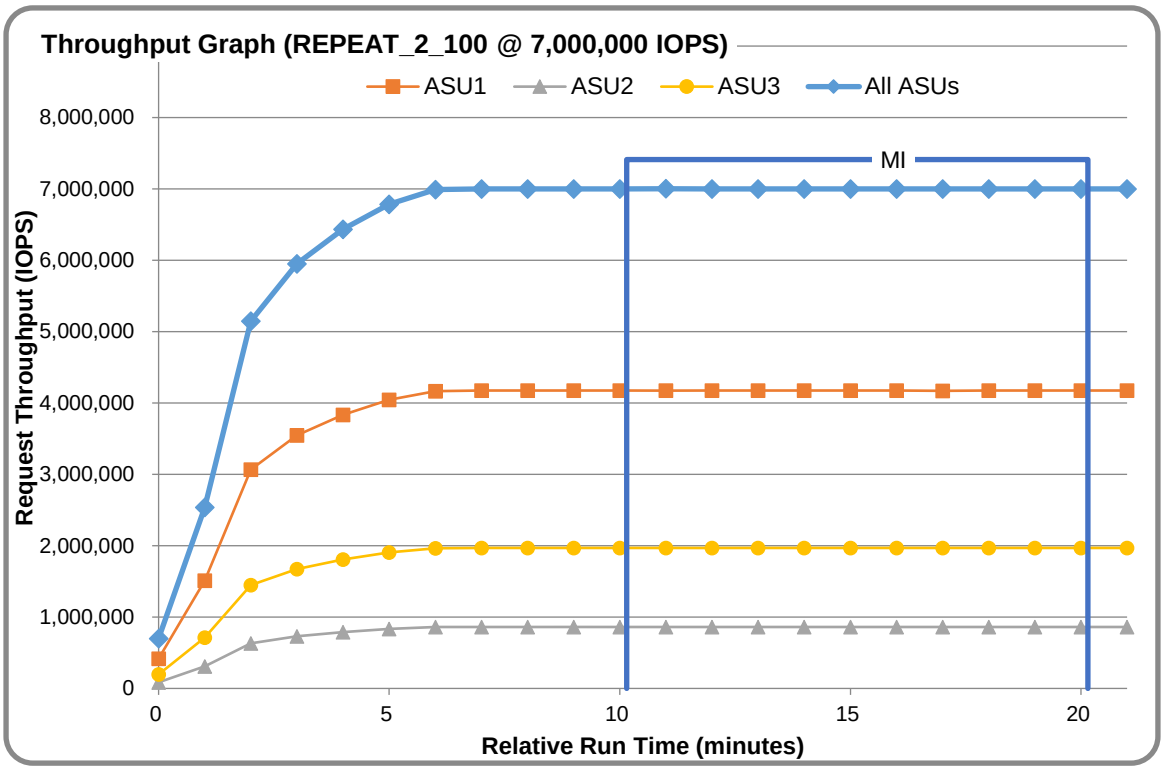
REPEAT_1_100 – Throughput Graph



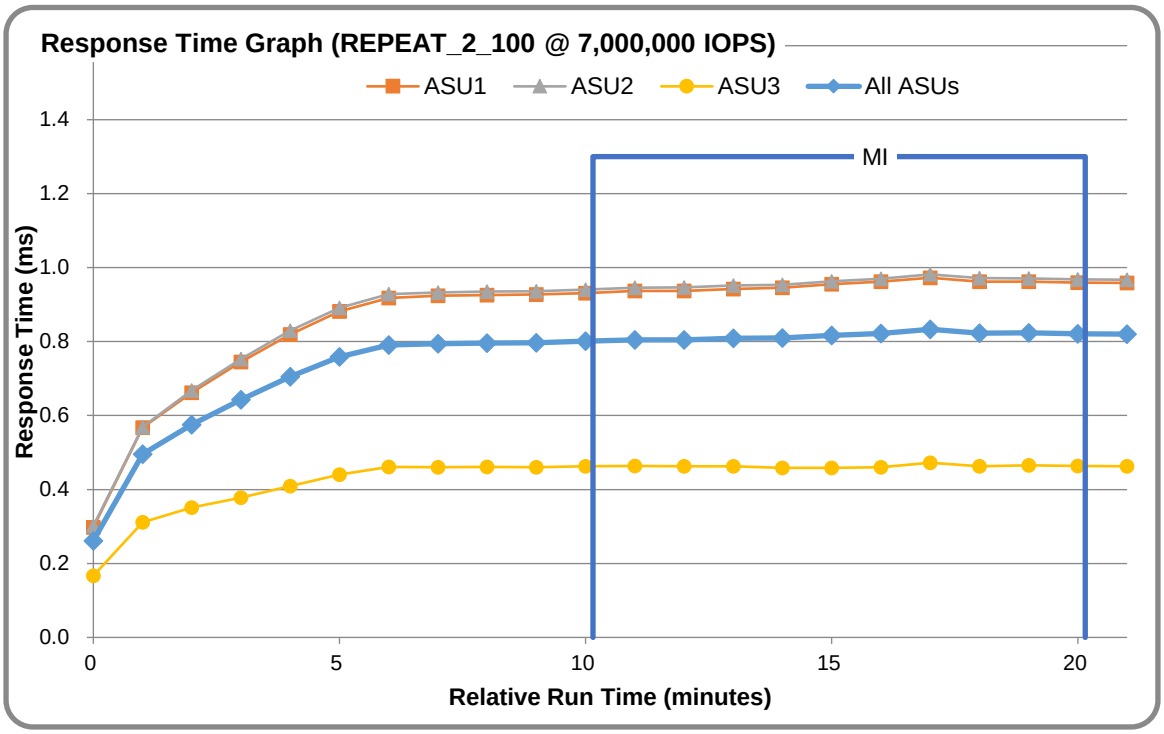
REPEAT_1_100 – Response Time Graph



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	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
Variation	0.0002	0.0001	0.0002	0.0001	0.0006	0.0001	0.0003	0.0001
Difference	0.011%	0.001%	0.002%	0.001%	0.007%	0.001%	0.022%	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	0.0002	0.0001	0.0002	0.0001	0.0004	0.0002	0.0003	0.0001
Difference	0.003%	0.001%	0.001%	0.001%	0.007%	0.004%	0.003%	0.002%

Space Optimization Reporting

Description of Utilized Techniques

SmartCompression compresses data online before writing data to flash media. In addition, compression is performed after deduplication, ensuring that no duplicate data is compressed and improving compression efficiency. SmartCompression reduces the amount of data written to SSDs and minimizes write amplification, improving the longevity of flash arrays.

The compression algorithm is a compute-intensive program. Inline compression consumes significant CPU resources, affecting end-to-end performance of the system. Open-source compression algorithms that feature high performance and low compression ratio are commonly used in the industry, for example, LZ4, LZO, and Snappy. OceanStor Dorado V3 uses the Fast LZ4 algorithm, which is an improvement of the open-source LZ4 compression algorithm and doubles the compression efficiency without decreasing the compression ratio.

The size of data blocks to be compressed can be 4 KB, 8 KB, 16 KB, and 32 KB. The unit for storing compressed data is 1 KB, which improves the compression efficiency and reduces the storage space needed for compressed data.

The compression ratio of OceanStor Dorado V3 varies based on user data. SmartCompression can be enabled or disabled for each specific LUN. In applications that require high performance, SmartCompression can be disabled.

Physical Free Space Measurements

The following table lists the Physical Free Space as measured at each of the required points during test execution. If space optimization techniques were not used, "NA" is reported.

Physical Free Space Measurement	Free Space (GB)
After Logical Volume Creation	211,312.5
After ASU Pre-Fill	74,718.2
After Repeatability Test Phase	74,718.2

Space Optimization Metrics

The following table lists the required space optimization metrics. If space optimization techniques were not used, "NA" is reported.

Space Optimization Metric	Value
SPC-1 Space Optimization Ratio	1.54
SPC-1 Space Effectiveness Ratio	0.73

Data Persistence Test

Data Persistence Test Results file

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	575,373,285
Total Number of Logical Blocks Verified	293,193,436
Total Number of Logical Blocks Overwritten	282,179,849
Total Number of Logical Blocks that Failed Verification	0
Time Duration for Writing Test Logical Blocks (sec.)	601
Size in bytes of each Logical Block	8,192
Number of Failed I/O Requests in the process of the Test	0

Committed Data Persistence Implementation

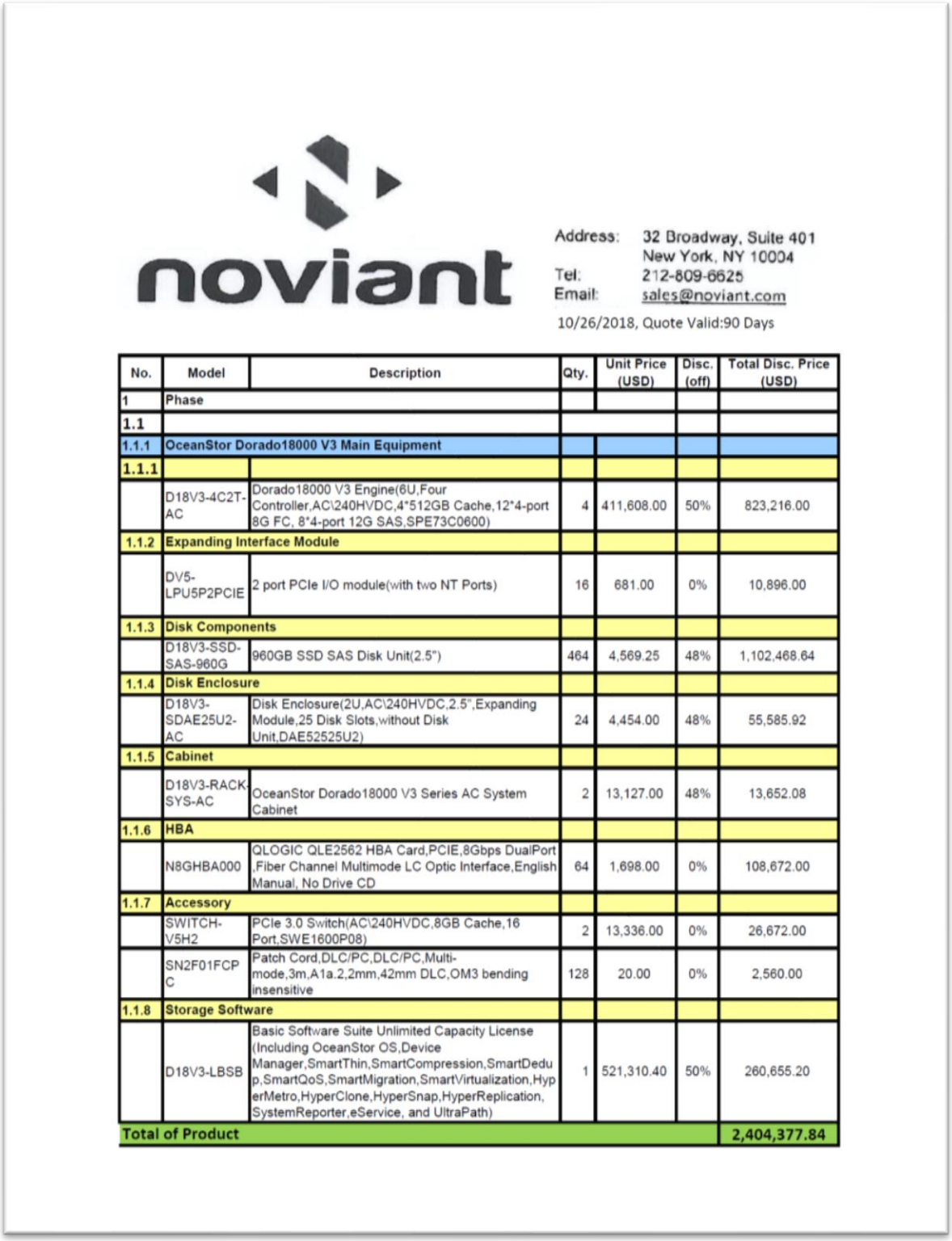
The persistency of committed data is implemented at two levels. At the disk level, data loss is prevented through the use of RAID 6 arrays. At the controller level, all caches are mirrored across controllers, where write requests are only completed once the local cache has been successfully mirrored 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**
profile1_volume.log	List of logical volumes before INIT	/E_Inventory
profile2_volume.log	List of logical volumes after restart	/E_Inventory
step1.log	Storage profile before INIT	/E_Inventory
step2.log	Storage profile after INIT	/E_Inventory
step3.log	Storage profile after METRICS	/E_Inventory
step4.log	Storage profile after PERSIST_1	/E_Inventory
/F_Generator	**Workload generator**	**root**
slave_asu.asu	Defining LUNs hosting the ASUs	/F_generator
28host.HST	Host configuration file	/F_generator
full_run.sh	Executing all test phases	/F_generator

APPENDIX B: THIRD PARTY QUOTATION



noviant

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

10/26/2018, Quote Valid:90 Days

No.	Model	Description	Qty.	Unit Price (USD)	Disc. (off)	Total Disc. Price (USD)
1	Phase					
1.1						
1.1.1	OceanStor Dorado18000 V3 Main Equipment					
1.1.1						
	D18V3-4C2T-AC	Dorado18000 V3 Engine(6U,Four Controller,AC\240HVDC,4*512GB Cache,12*4-port 8G FC, 8*4-port 12G SAS,SPE73C0600)	4	411,608.00	50%	823,216.00
1.1.2	Expanding Interface Module					
	DV5-LPU5P2PCIE	2 port PCIe I/O module(with two NT Ports)	16	681.00	0%	10,896.00
1.1.3	Disk Components					
	D18V3-SSD-SAS-960G	960GB SSD SAS Disk Unit(2.5")	464	4,569.25	48%	1,102,468.64
1.1.4	Disk Enclosure					
	D18V3-SDAE25U2-AC	Disk Enclosure(2U,AC\240HVDC,2.5",Expanding Module,25 Disk Slots,without Disk Unit,DAE52525U2)	24	4,454.00	48%	55,585.92
1.1.5	Cabinet					
	D18V3-RACK-SYS-AC	OceanStor Dorado18000 V3 Series AC System Cabinet	2	13,127.00	48%	13,652.08
1.1.6	HBA					
	N8GHBA000	QLOGIC QLE2562 HBA Card,PCIE,8Gbps DualPort ,Fiber Channel Multimode LC Optic Interface,English Manual, No Drive CD	64	1,698.00	0%	108,672.00
1.1.7	Accessory					
	SWITCH-V5H2	PCIe 3.0 Switch(AC\240HVDC,8GB Cache,16 Port,SWE1600P08)	2	13,336.00	0%	26,672.00
	SN2F01FCPC	Patch Cord,DLC/PC,DLC/PC,Multi-mode,3m,A1a.2,2mm,42mm DLC,OM3 bending insensitive	128	20.00	0%	2,560.00
1.1.8	Storage Software					
	D18V3-LBSB	Basic Software Suite Unlimited Capacity License (Including OceanStor OS,Device Manager,SmartThin,SmartCompression,SmartDedup,SmartQoS,SmartMigration,SmartVirtualization,HyperMetro,HyperClone,HyperSnap,HyperReplication, SystemReporter,eService, and UltraPath)	1	521,310.40	50%	260,655.20
Total of Product						2,404,377.84

1.1.9	Maintenance Support Service					
	02351WXJ-88134ULF-36	OceanStor Dorado18000 V3 Engine(6U,Four Controller,AC\240HVDC,4*512GB Cache,12*4-port 8G FC, 8*4-port 12G SAS,SPE73C0600)-Hi-Care Onsite Premier 24x7x4H Engineer Onsite Service-36Month(s)	4	17,202.60	0%	68,810.40
	02352ANB-88134ULF-36	960GB SSD SAS Disk Unit(2.5")-Hi-Care Onsite Premier 24x7x4H Engineer Onsite Service-36Month(s)	464	122.40	0%	56,793.60
	88034SDG-88134UHK-36	Basic Software Licenses (Including OceanStor OS,DeviceManager, SmartThin, SmartMigration, SmartDedupe, SmartCompression,SmartQoS and SystemReporter)-Hi-Care Application Software Upgrade Support Service-36Month(s)	1	71,235.00	0%	71,235.00
	8812175865	OceanStor Dorado18000 V3 Installation Service - Engineering	1	47,749.12	0%	47,749.12
Total of Service (3 years)						244,588.12
Total Price						2,648,965.96

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:

- ***scheduler.sh*** to change the I/O scheduler from cfq to noop on each Host System, which will result in all incoming I/O requests inserted into a simple, unordered FIFO queue
- ***nr_requests.sh*** to change nr_requests from 128 to 1024 on each Host System for each device
- ***aio-max-nr.sh*** to change the maximum number of AIO operations to 1048576

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

- ***scheduler.sh***

```
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
echo noop >/sys/block/sdr/queue/scheduler
echo noop >/sys/block/sds/queue/scheduler
echo noop >/sys/block/sdt/queue/scheduler
echo noop >/sys/block/sdu/queue/scheduler
echo noop >/sys/block/sdv/queue/scheduler
echo noop >/sys/block/sdw/queue/scheduler
echo noop >/sys/block/sdx/queue/scheduler
echo noop >/sys/block/sdy/queue/scheduler
echo noop >/sys/block/sdz/queue/scheduler
echo noop >/sys/block/sdaa/queue/scheduler
echo noop >/sys/block/sdab/queue/scheduler
echo noop >/sys/block/sdac/queue/scheduler
echo noop >/sys/block/sdad/queue/scheduler
echo noop >/sys/block/sdae/queue/scheduler
echo noop >/sys/block/sdaf/queue/scheduler
echo noop >/sys/block/sdag/queue/scheduler
echo noop >/sys/block/sdah/queue/scheduler
echo noop >/sys/block/sdai/queue/scheduler
echo noop >/sys/block/sdaj/queue/scheduler
echo noop >/sys/block/sdak/queue/scheduler
echo noop >/sys/block/sdal/queue/scheduler
echo noop >/sys/block/sdam/queue/scheduler
echo noop >/sys/block/sdan/queue/scheduler
```

```
echo noop >/sys/block/sdao/queue/scheduler
echo noop >/sys/block/sdap/queue/scheduler
echo noop >/sys/block/sdaq/queue/scheduler
echo noop >/sys/block/sdar/queue/scheduler
echo noop >/sys/block/sdas/queue/scheduler
echo noop >/sys/block/sdat/queue/scheduler
echo noop >/sys/block/sdau/queue/scheduler
echo noop >/sys/block/sdav/queue/scheduler
echo noop >/sys/block/sdaw/queue/scheduler
echo noop >/sys/block/sdax/queue/scheduler
echo noop >/sys/block/sday/queue/scheduler
echo noop >/sys/block/sdaz/queue/scheduler
echo noop >/sys/block/sdba/queue/scheduler
echo noop >/sys/block/sdbb/queue/scheduler
echo noop >/sys/block/sdbc/queue/scheduler
echo noop >/sys/block/sdbd/queue/scheduler
echo noop >/sys/block/sdbe/queue/scheduler
echo noop >/sys/block/sdbf/queue/scheduler
echo noop >/sys/block/sdbg/queue/scheduler
echo noop >/sys/block/sdbh/queue/scheduler
echo noop >/sys/block/sdbi/queue/scheduler
echo noop >/sys/block/sdbj/queue/scheduler
echo noop >/sys/block/sdbk/queue/scheduler
echo noop >/sys/block/sdbl/queue/scheduler
echo noop >/sys/block/sdbm/queue/scheduler
```

nr_requests.sh

```
echo 2048 >/sys/block/sdb/queue/nr_requests
echo 2048 >/sys/block/sdc/queue/nr_requests
echo 2048 >/sys/block/sdd/queue/nr_requests
echo 2048 >/sys/block/sde/queue/nr_requests
echo 2048 >/sys/block/sdf/queue/nr_requests
echo 2048 >/sys/block/sdg/queue/nr_requests
echo 2048 >/sys/block/sdh/queue/nr_requests
echo 2048 >/sys/block/sdi/queue/nr_requests
echo 2048 >/sys/block/sdj/queue/nr_requests
echo 2048 >/sys/block/sdk/queue/nr_requests
echo 2048 >/sys/block/sdl/queue/nr_requests
echo 2048 >/sys/block/sdm/queue/nr_requests
echo 2048 >/sys/block/sdn/queue/nr_requests
echo 2048 >/sys/block/sdo/queue/nr_requests
echo 2048 >/sys/block/sdp/queue/nr_requests
echo 2048 >/sys/block/sdq/queue/nr_requests
echo 2048 >/sys/block/sdr/queue/nr_requests
echo 2048 >/sys/block/sds/queue/nr_requests
echo 2048 >/sys/block/sdt/queue/nr_requests
echo 2048 >/sys/block/sdu/queue/nr_requests
echo 2048 >/sys/block/sdv/queue/nr_requests
echo 2048 >/sys/block/sdw/queue/nr_requests
echo 2048 >/sys/block/sdx/queue/nr_requests
echo 2048 >/sys/block/sdy/queue/nr_requests
echo 2048 >/sys/block/sdz/queue/nr_requests
echo 2048 >/sys/block/sdaa/queue/nr_requests
echo 2048 >/sys/block/sdab/queue/nr_requests
echo 2048 >/sys/block/sdac/queue/nr_requests
echo 2048 >/sys/block/sdad/queue/nr_requests
echo 2048 >/sys/block/sdae/queue/nr_requests
echo 2048 >/sys/block/sdaf/queue/nr_requests
echo 2048 >/sys/block/sdag/queue/nr_requests
echo 2048 >/sys/block/sdah/queue/nr_requests
```

```
echo 2048 >/sys/block/sdai/queue/nr_requests
echo 2048 >/sys/block/sdaj/queue/nr_requests
echo 2048 >/sys/block/sdak/queue/nr_requests
echo 2048 >/sys/block/sdal/queue/nr_requests
echo 2048 >/sys/block/sdam/queue/nr_requests
echo 2048 >/sys/block/sdan/queue/nr_requests
echo 2048 >/sys/block/sdao/queue/nr_requests
echo 2048 >/sys/block/sdap/queue/nr_requests
echo 2048 >/sys/block/sdaq/queue/nr_requests
echo 2048 >/sys/block/sdar/queue/nr_requests
echo 2048 >/sys/block/sdas/queue/nr_requests
echo 2048 >/sys/block/sdat/queue/nr_requests
echo 2048 >/sys/block/sdau/queue/nr_requests
echo 2048 >/sys/block/sdav/queue/nr_requests
echo 2048 >/sys/block/sdaw/queue/nr_requests
echo 2048 >/sys/block/sdax/queue/nr_requests
echo 2048 >/sys/block/sday/queue/nr_requests
echo 2048 >/sys/block/sdaz/queue/nr_requests
echo 2048 >/sys/block/sdba/queue/nr_requests
echo 2048 >/sys/block/sdbb/queue/nr_requests
echo 2048 >/sys/block/sdbc/queue/nr_requests
echo 2048 >/sys/block/sdbd/queue/nr_requests
echo 2048 >/sys/block/sdbe/queue/nr_requests
echo 2048 >/sys/block/sdbf/queue/nr_requests
echo 2048 >/sys/block/sdbg/queue/nr_requests
echo 2048 >/sys/block/sdbh/queue/nr_requests
echo 2048 >/sys/block/sdbi/queue/nr_requests
echo 2048 >/sys/block/sdbj/queue/nr_requests
echo 2048 >/sys/block/sdbk/queue/nr_requests
echo 2048 >/sys/block/sdbl/queue/nr_requests
echo 2048 >/sys/block/sdbm/queue/nr_requests
echo 2048 >/sys/block/sdai/queue/nr_requests
echo 2048 >/sys/block/sdaj/queue/nr_requests
echo 2048 >/sys/block/sdak/queue/nr_requests
echo 2048 >/sys/block/sdal/queue/nr_requests
echo 2048 >/sys/block/sdam/queue/nr_requests
echo 2048 >/sys/block/sdan/queue/nr_requests
echo 2048 >/sys/block/sdao/queue/nr_requests
echo 2048 >/sys/block/sdap/queue/nr_requests
echo 2048 >/sys/block/sdaq/queue/nr_requests
echo 2048 >/sys/block/sdar/queue/nr_requests
echo 2048 >/sys/block/sdas/queue/nr_requests
echo 2048 >/sys/block/sdat/queue/nr_requests
echo 2048 >/sys/block/sdau/queue/nr_requests
echo 2048 >/sys/block/sdav/queue/nr_requests
echo 2048 >/sys/block/sdaw/queue/nr_requests
echo 2048 >/sys/block/sdax/queue/nr_requests
echo 2048 >/sys/block/sday/queue/nr_requests
echo 2048 >/sys/block/sdaz/queue/nr_requests
echo 2048 >/sys/block/sdba/queue/nr_requests
echo 2048 >/sys/block/sdbb/queue/nr_requests
echo 2048 >/sys/block/sdbc/queue/nr_requests
echo 2048 >/sys/block/sdbd/queue/nr_requests
echo 2048 >/sys/block/sdbe/queue/nr_requests
echo 2048 >/sys/block/sdbf/queue/nr_requests
echo 2048 >/sys/block/sdbg/queue/nr_requests
echo 2048 >/sys/block/sdbh/queue/nr_requests
echo 2048 >/sys/block/sdbi/queue/nr_requests
echo 2048 >/sys/block/sdbj/queue/nr_requests
echo 2048 >/sys/block/sdbk/queue/nr_requests
echo 2048 >/sys/block/sdbl/queue/nr_requests
echo 2048 >/sys/block/sdbm/queue/nr_requests
```

aio-max-nr.sh

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

APPENDIX D: STORAGE CONFIGURATION CREATION

Environment

First, the CLI commands from the following command file are copied and pasted into the OceanStor Dorado18000 V3 CLI window. These commands are executed on one of the Host Systems.

- ***mklun.sh***

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

- ***mkvolume.sh***

Step 1 - Create Disk Domains, Storage Pools, LUNs

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

- Create 8 disk domains
- Create 8 storage pools
- Create 64 LUNs
- Create one LUN group
- Add the 64 LUNs to the LUN group

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

mklun.sh

```
#!/usr/bin/tclsh

package require Expect

proc create_storage {} {
#host config
set host_name root
set host_pswd huawei@123
set host(0) 8.46.203.84
set host(1) 8.46.203.83
set host(2) 8.46.203.85
set host(3) 8.46.203.86
set host(4) 8.46.203.87
set host(5) 8.46.203.89
set host(6) 8.46.203.90
set host(7) 8.46.203.91
set host(8) 8.46.203.92
set host(9) 8.46.203.93
set host(10) 8.46.203.94
set host(11) 8.46.203.95
set host(12) 8.46.203.96
set host(13) 8.46.203.120
set host(14) 8.46.203.121
set host(15) 8.46.203.128
set host(16) 8.46.203.129
set host(17) 8.46.203.175
```

```
set host(18) 8.46.203.180
set host(19) 8.46.203.181
set host(20) 8.46.203.182
set host(21) 8.46.203.185
set host(22) 8.46.203.198
set host(23) 8.46.203.199
set host(24) 8.46.203.200
set host(25) 8.46.203.201
set host(26) 8.46.203.202
set host(27) 8.46.203.203
set host(28) 8.46.203.214
set host(29) 8.46.203.215
set host(30) 8.46.203.216
set host(31) 8.46.203.217

set host_num 32
set initiator_num_perhost 4

set shell_path "/home/config"

### 8.46.203.83
set   initiator(4) 21000024ff1b8ee1
set   initiator(5) 21000024ff1bdd8c
set   initiator(6) 21000024ff1bdd8d
set   initiator(7) 21000024ff1b8ee0
### 8.46.203.84
set   initiator(0) 21000024ff438099
set   initiator(1) 21000024ff1bdd62
set   initiator(2) 21000024ff1bdd63
set   initiator(3) 21000024ff438098
### 8.46.203.85
set   initiator(8) 21000024ff368169
set   initiator(9) 21000024ff3c02c2
set   initiator(10) 21000024ff3c02c3
set   initiator(11) 21000024ff368168
### 8.46.203.86
set   initiator(12) 21000024ff35696f
set   initiator(13) 21000024ff3fafba
set   initiator(14) 21000024ff3fafbb
set   initiator(15) 21000024ff35696e
### 8.46.203.87
set   initiator(16) 21000024ff37203d
set   initiator(17) 21000024ff7f8aba
set   initiator(18) 21000024ff7f8abb
set   initiator(19) 21000024ff37203c
### 8.46.203.89
set   initiator(20) 21000024ff4b825c
set   initiator(21) 21000024ff4b825d
set   initiator(22) 21000024ff75a43c
set   initiator(23) 21000024ff75a43d
### 8.46.203.90
set   initiator(24) 21000024ff751f6f
set   initiator(25) 21000024ff4a53fe
set   initiator(26) 21000024ff4a53ff
set   initiator(27) 21000024ff751f6e
### 8.46.203.91
set   initiator(28) 21000024ff7fb903
set   initiator(29) 21000024ff7fb716
set   initiator(30) 21000024ff7fb717
set   initiator(31) 21000024ff7fb902
### 8.46.203.93
set   initiator(32) 21000024ff17dff5
```

```
set   initiator(33) 21000024ff17df38
set   initiator(34) 21000024ff17df39
set   initiator(35) 21000024ff17dff4
### 8.46.203.94
set   initiator(36) 21000024ff7f431a
set   initiator(37) 21000024ff7f431b
set   initiator(38) 21000024ff7f78fe
set   initiator(39) 21000024ff7f78ff
### 8.46.203.95
set   initiator(40) 21000024ff7ea0fb
set   initiator(41) 21000024ff17e0bc
set   initiator(42) 21000024ff17e0bd
set   initiator(43) 21000024ff7ea0fa
### 8.46.203.96
set   initiator(44) 21000024ff3cc4cb
set   initiator(45) 21000024ff4b81fc
set   initiator(46) 21000024ff4b81fd
set   initiator(47) 21000024ff3cc4ca
### 8.46.203.120
set   initiator(48) 21000024ff175ff7
set   initiator(49) 21000024ff553dc0
set   initiator(50) 21000024ff553dc1
set   initiator(51) 21000024ff175ff6
### 8.46.203.121
set   initiator(52) 21000024ff1bdfe3
set   initiator(53) 21000024ff3fb12a
set   initiator(54) 21000024ff3fb12b
set   initiator(55) 21000024ff1bdfe2
### 8.46.203.128
set   initiator(56) 21000024ff5439d6
set   initiator(57) 21000024ff5439d7
set   initiator(58) 21000024ff4b82ea
set   initiator(59) 21000024ff4b82eb
### 8.46.203.129
set   initiator(60) 21000024ff41c4a9
set   initiator(61) 21000024ff41c4a8
set   initiator(62) 21000024ff89be00
set   initiator(63) 21000024ff89be01
### 8.46.203.175
set   initiator(64) 21000024ff49997b
set   initiator(65) 21000024ff7e881c
set   initiator(66) 21000024ff7e881d
set   initiator(67) 21000024ff49997a
### 8.46.203.180
set   initiator(68) 21000024ff543be3
set   initiator(69) 21000024ff7f3fd6
set   initiator(70) 21000024ff7f3fd7
set   initiator(71) 21000024ff543be2
### 8.46.203.181
set   initiator(72) 21000024ff7e884e
set   initiator(73) 21000024ff7e884f
set   initiator(74) 21000024ff536a84
set   initiator(75) 21000024ff536a85
### 8.46.203.182
set   initiator(76) 21000024ff752081
set   initiator(77) 21000024ff5fbbd2
set   initiator(78) 21000024ff5fbbd3
set   initiator(79) 21000024ff752080
### 8.46.203.92
set   initiator(80) 21000024ff17e0bb
set   initiator(81) 21000024ff28ea5c
set   initiator(82) 21000024ff28ea5d
```

```
set   initiator(83) 21000024ff17e0ba
### 8.46.203.185
set   initiator(84) 21000024ff8f27a8
set   initiator(85) 21000024ff8f27a9
set   initiator(86) 21000024ff7f889a
set   initiator(87) 21000024ff7f889b
### 8.46.203.198
set   initiator(88) 21000024ff543a15
set   initiator(89) 21000024ff7e889c
set   initiator(90) 21000024ff7e889d
set   initiator(91) 21000024ff543a14
### 8.46.203.199
set   initiator(92) 21000024ff547009
set   initiator(93) 21000024ff3cafe8
set   initiator(94) 21000024ff3cafe9
set   initiator(95) 21000024ff54702b
### 8.46.203.200
set   initiator(96) 21000024ff91e4c5
set   initiator(97) 21000024ff208834
set   initiator(98) 21000024ff208835
set   initiator(99) 21000024ff91e4c4
### 8.46.203.201
set   initiator(100) 21000024ff5f8c1f
set   initiator(101) 21000024ff536a8e
set   initiator(102) 21000024ff536a8f
set   initiator(103) 21000024ff5f8c1e
### 8.46.203.202
set   initiator(104) 21000024ff7f8a03
set   initiator(105) 21000024ff7ea3c0
set   initiator(106) 21000024ff7ea3c1
set   initiator(107) 21000024ff7f8a02
### 8.46.203.203
set   initiator(108) 21000024ff756f04
set   initiator(109) 21000024ff756f05
set   initiator(110) 21000024ff5fbc12
set   initiator(111) 21000024ff5fbc13
### 8.46.203.214
set   initiator(112) 21000024ff4a5403
set   initiator(113) 21000024ff4bc1fc
set   initiator(114) 21000024ff4bc1fd
set   initiator(115) 21000024ff4a5402
### 8.46.203.215
set   initiator(116) 21000024ff4b8217
set   initiator(117) 21000024ff90188c
set   initiator(118) 21000024ff90188d
set   initiator(119) 21000024ff4b8216
### 8.46.203.216
set   initiator(120) 21000024ff553e1f
set   initiator(121) 21000024ff2b0e74
set   initiator(122) 21000024ff2b0e75
set   initiator(123) 21000024ff553e1e
### 8.46.203.217
set   initiator(124) 21000024ff36cedf
set   initiator(125) 21000024ff7e8660
set   initiator(126) 21000024ff7e8661
set   initiator(127) 21000024ff36cede

#storage config
set stor_user ibc_os_hs
set stor_pswd Storage@21st
set stor(0) 8.46.203.150
set stor(1) 8.46.203.142
```

```
set stor(2) 8.46.203.135
set stor(3) 8.46.203.130
set engine_num 4
set stor_num 4

#disk_domain config
set dd_num 8

#lun_workload config
set grain_size 8KB

#lun config
set lun_num 64
set lun_size 3100GB

spawn ssh $stor_user@$stor(0)
set timeout 60
expect {
   -re "assword" { send "$stor_pswd\n" }
   -re "yes/no" { send "yes\n"; exp_continue }
}

expect {
   "#" { send "/ISM/cli/start.sh -u admin\r" }
   ">" { send "\r" }
   "(y/n):" { send "y\r"; exp_continue }
}
expect ">"

#create disk_domain
for {set i 0} {$i < $dd_num} {incr i} {
   send "create disk_domain name=spc1_dd$i disk_list=all disk_domain_id=$i\r"
   expect ">"
}
sleep 10
#create storage_pool
for {set i 0} {$i < $dd_num} {incr i} {
   send "create storage_pool name=spc1_p$i capacity=remain disk_domain_id=$i
   pool_id=$i raid_level=RAID6\r"
   expect ">"
}
sleep 10
#create lun_workload
send   "create   lun_workload_type   general   name=spc1   io_size=$grain_size
   dedup_enabled=no compression_enabled=yes id=16\r"
expect ">"
sleep 1
#create lun
################################################################
send "create lun name=spc1_0A_0 pool_id=0 capacity=$lun_size workload_type_id=16
   lun_id=31 owner_controller=0A\r"
expect ">"
send "create lun name=spc1_0A_1 pool_id=0 capacity=$lun_size workload_type_id=16
   lun_id=56 owner_controller=0A\r"
expect ">"
send "create lun name=spc1_0A_2 pool_id=0 capacity=$lun_size workload_type_id=16
   lun_id=15 owner_controller=0A\r"
expect ">"
send "create lun name=spc1_0A_3 pool_id=0 capacity=$lun_size workload_type_id=16
   lun_id=41 owner_controller=0A\r"
expect ">"
```

```
send "create lun name=spc1_0B_0 pool_id=0 capacity=$lun_size workload_type_id=16
   lun_id=62 owner_controller=0B\r"
expect ">"
send "create lun name=spc1_0B_1 pool_id=0 capacity=$lun_size workload_type_id=16
   lun_id=8 owner_controller=0B\r"
expect ">"
send "create lun name=spc1_0B_2 pool_id=0 capacity=$lun_size workload_type_id=16
   lun_id=24 owner_controller=0B\r"
expect ">"
send "create lun name=spc1_0B_3 pool_id=0 capacity=$lun_size workload_type_id=16
   lun_id=44 owner_controller=0B\r"
expect ">"
send "create lun name=spc1_0C_0 pool_id=1 capacity=$lun_size workload_type_id=16
   lun_id=48 owner_controller=0C\r"
expect ">"
send "create lun name=spc1_0C_1 pool_id=1 capacity=$lun_size workload_type_id=16
   lun_id=11 owner_controller=0C\r"
expect ">"
send "create lun name=spc1_0C_2 pool_id=1 capacity=$lun_size workload_type_id=16
   lun_id=19 owner_controller=0C\r"
expect ">"
send "create lun name=spc1_0C_3 pool_id=1 capacity=$lun_size workload_type_id=16
   lun_id=45 owner_controller=0C\r"
expect ">"
send "create lun name=spc1_0D_0 pool_id=1 capacity=$lun_size workload_type_id=16
   lun_id=59 owner_controller=0D\r"
expect ">"
send "create lun name=spc1_0D_1 pool_id=1 capacity=$lun_size workload_type_id=16
   lun_id=3 owner_controller=0D\r"
expect ">"
send "create lun name=spc1_0D_2 pool_id=1 capacity=$lun_size workload_type_id=16
   lun_id=18 owner_controller=0D\r"
expect ">"
send "create lun name=spc1_0D_3 pool_id=1 capacity=$lun_size workload_type_id=16
   lun_id=37 owner_controller=0D\r"
expect ">"
send "create lun name=spc1_1A_0 pool_id=2 capacity=$lun_size workload_type_id=16
   lun_id=53 owner_controller=1A\r"
expect ">"
send "create lun name=spc1_1A_1 pool_id=2 capacity=$lun_size workload_type_id=16
   lun_id=10 owner_controller=1A\r"
expect ">"
send "create lun name=spc1_1A_2 pool_id=2 capacity=$lun_size workload_type_id=16
   lun_id=20 owner_controller=1A\r"
expect ">"
send "create lun name=spc1_1A_3 pool_id=2 capacity=$lun_size workload_type_id=16
   lun_id=40 owner_controller=1A\r"
expect ">"
send "create lun name=spc1_1B_0 pool_id=2 capacity=$lun_size workload_type_id=16
   lun_id=55 owner_controller=1B\r"
expect ">"
send "create lun name=spc1_1B_1 pool_id=2 capacity=$lun_size workload_type_id=16
   lun_id=5 owner_controller=1B\r"
expect ">"
send "create lun name=spc1_1B_2 pool_id=2 capacity=$lun_size workload_type_id=16
   lun_id=28 owner_controller=1B\r"
expect ">"
send "create lun name=spc1_1B_3 pool_id=2 capacity=$lun_size workload_type_id=16
   lun_id=38 owner_controller=1B\r"
expect ">"
send "create lun name=spc1_1C_0 pool_id=3 capacity=$lun_size workload_type_id=16
   lun_id=49 owner_controller=1C\r"
```

```
expect ">"
send "create lun name=spc1_1C_1 pool_id=3 capacity=$lun_size workload_type_id=16
   lun_id=6 owner_controller=1C\r"
expect ">"
send "create lun name=spc1_1C_2 pool_id=3 capacity=$lun_size workload_type_id=16
   lun_id=17 owner_controller=1C\r"
expect ">"
send "create lun name=spc1_1C_3 pool_id=3 capacity=$lun_size workload_type_id=16
   lun_id=32 owner_controller=1C\r"
expect ">"
send "create lun name=spc1_1D_0 pool_id=3 capacity=$lun_size workload_type_id=16
   lun_id=60 owner_controller=1D\r"
expect ">"
send "create lun name=spc1_1D_1 pool_id=3 capacity=$lun_size workload_type_id=16
   lun_id=4 owner_controller=1D\r"
expect ">"
send "create lun name=spc1_1D_2 pool_id=3 capacity=$lun_size workload_type_id=16
   lun_id=30 owner_controller=1D\r"
expect ">"
send "create lun name=spc1_1D_3 pool_id=3 capacity=$lun_size workload_type_id=16
   lun_id=46 owner_controller=1D\r"
expect ">"
send "create lun name=spc1_2A_0 pool_id=4 capacity=$lun_size workload_type_id=16
   lun_id=58 owner_controller=2A\r"
expect ">"
send "create lun name=spc1_2A_1 pool_id=4 capacity=$lun_size workload_type_id=16
   lun_id=2 owner_controller=2A\r"
expect ">"
send "create lun name=spc1_2A_2 pool_id=4 capacity=$lun_size workload_type_id=16
   lun_id=26 owner_controller=2A\r"
expect ">"
send "create lun name=spc1_2A_3 pool_id=4 capacity=$lun_size workload_type_id=16
   lun_id=35 owner_controller=2A\r"
expect ">"
send "create lun name=spc1_2B_0 pool_id=4 capacity=$lun_size workload_type_id=16
   lun_id=52 owner_controller=2B\r"
expect ">"
send "create lun name=spc1_2B_1 pool_id=4 capacity=$lun_size workload_type_id=16
   lun_id=9 owner_controller=2B\r"
expect ">"
send "create lun name=spc1_2B_2 pool_id=4 capacity=$lun_size workload_type_id=16
   lun_id=21 owner_controller=2B\r"
expect ">"
send "create lun name=spc1_2B_3 pool_id=4 capacity=$lun_size workload_type_id=16
   lun_id=34 owner_controller=2B\r"
expect ">"
send "create lun name=spc1_2C_0 pool_id=5 capacity=$lun_size workload_type_id=16
   lun_id=50 owner_controller=2C\r"
expect ">"
send "create lun name=spc1_2C_1 pool_id=5 capacity=$lun_size workload_type_id=16
   lun_id=14 owner_controller=2C\r"
expect ">"
send "create lun name=spc1_2C_2 pool_id=5 capacity=$lun_size workload_type_id=16
   lun_id=29 owner_controller=2C\r"
expect ">"
send "create lun name=spc1_2C_3 pool_id=5 capacity=$lun_size workload_type_id=16
   lun_id=43 owner_controller=2C\r"
expect ">"
send "create lun name=spc1_2D_0 pool_id=5 capacity=$lun_size workload_type_id=16
   lun_id=54 owner_controller=2D\r"
expect ">"
```

```
send "create lun name=spc1_2D_1 pool_id=5 capacity=$lun_size workload_type_id=16
    lun_id=12 owner_controller=2D\r"
expect ">"
send "create lun name=spc1_2D_2 pool_id=5 capacity=$lun_size workload_type_id=16
    lun_id=25 owner_controller=2D\r"
expect ">"
send "create lun name=spc1_2D_3 pool_id=5 capacity=$lun_size workload_type_id=16
    lun_id=39 owner_controller=2D\r"
expect ">"
send "create lun name=spc1_3A_0 pool_id=6 capacity=$lun_size workload_type_id=16
    lun_id=61 owner_controller=3A\r"
expect ">"
send "create lun name=spc1_3A_1 pool_id=6 capacity=$lun_size workload_type_id=16
    lun_id=1 owner_controller=3A\r"
expect ">"
send "create lun name=spc1_3A_2 pool_id=6 capacity=$lun_size workload_type_id=16
    lun_id=22 owner_controller=3A\r"
expect ">"
send "create lun name=spc1_3A_3 pool_id=6 capacity=$lun_size workload_type_id=16
    lun_id=47 owner_controller=3A\r"
expect ">"
send "create lun name=spc1_3B_0 pool_id=6 capacity=$lun_size workload_type_id=16
    lun_id=51 owner_controller=3B\r"
expect ">"
send "create lun name=spc1_3B_1 pool_id=6 capacity=$lun_size workload_type_id=16
    lun_id=13 owner_controller=3B\r"
expect ">"
send "create lun name=spc1_3B_2 pool_id=6 capacity=$lun_size workload_type_id=16
    lun_id=27 owner_controller=3B\r"
expect ">"
send "create lun name=spc1_3B_3 pool_id=6 capacity=$lun_size workload_type_id=16
    lun_id=42 owner_controller=3B\r"
expect ">"
send "create lun name=spc1_3C_0 pool_id=7 capacity=$lun_size workload_type_id=16
    lun_id=57 owner_controller=3C\r"
expect ">"
send "create lun name=spc1_3C_1 pool_id=7 capacity=$lun_size workload_type_id=16
    lun_id=7 owner_controller=3C\r"
expect ">"
send "create lun name=spc1_3C_2 pool_id=7 capacity=$lun_size workload_type_id=16
    lun_id=23 owner_controller=3C\r"
expect ">"
send "create lun name=spc1_3C_3 pool_id=7 capacity=$lun_size workload_type_id=16
    lun_id=33 owner_controller=3C\r"
expect ">"
send "create lun name=spc1_3D_0 pool_id=7 capacity=$lun_size workload_type_id=16
    lun_id=63 owner_controller=3D\r"
expect ">"
send "create lun name=spc1_3D_1 pool_id=7 capacity=$lun_size workload_type_id=16
    lun_id=0 owner_controller=3D\r"
expect ">"
send "create lun name=spc1_3D_2 pool_id=7 capacity=$lun_size workload_type_id=16
    lun_id=36 owner_controller=3D\r"
expect ">"
send "create lun name=spc1_3D_3 pool_id=7 capacity=$lun_size workload_type_id=16
    lun_id=16 owner_controller=3D\r"
expect ">"
sleep 1
#create lun_group
send "create lun_group name=spc1_lg0 lun_group_id=0 lun_id_list=0-[expr \$lun_num
    - 1]\r"
expect ">"
```

Step 2 - Create Mapping View, Host Group and Host

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

- Create 32 hosts
- Create a host group
- Add the 32 hosts to the host group
- Add the FC port's WWN to the 4 hosts
- Create a mapping view
- Add the host group and the LUN group to the mapping view

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

mklun.sh

```
sleep 1
#create host
for {set i 0} {$i < $host_num} {incr i} {
        send "create host name=spc1_host$i operating_system=Linux host_id=$i
   ip=$host($i)\r"
        expect "(y/n)"
        send "y\r"
        expect ">"
}
sleep 1
#create host_group
send "create host_group name=spc1_hg0 host_group_id=0 host_id_list=0-[expr
   \$host_num - 1]\r"
expect ">"
sleep 1
#add initiator
for {set i 0} {$i < $host_num} {incr i} {
        for {set j 0} {$j < $initiator_num_perhost} {incr j} {
                set tmp_index [ expr \$i * \$initiator_num_perhost + \$j ]
                send "add host initiator host_id=$i initiator_type=FC
   wwn=$initiator($tmp_index)\r"
                expect {
               -re ">" { send "\r"; }
               -re "y/n" { send "y\n"; exp_continue }
                }
                expect ">"
        }
}
sleep 1
#create mapping_view
send "create mapping_view name=map_view0 lun_group_id=0 host_group_id=0
   mapping_view_id=0\r"
expect {
   -re ">" { send "\r"; }
   -re "y/n" { send "y\n"; exp_continue }
}
expect ">"
puts "set storage success!"
}
```

Step 3 - Create Volumes on the Host Systems

The ***mkvolume.sh*** shell script, listed below, is invoked on the master Host Systems to perform the following actions:

- Create 64 physical volumes
- Create a volume group for the 64 physical volumes
- Create 18 Logical Volumes for ASU-1
- Create 18 Logical Volumes for ASU-2
- Create 2 Logical Volumes for ASU-3

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

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
pvcreate /dev/sdr
pvcreate /dev/sds
pvcreate /dev/sdt
pvcreate /dev/sdu
pvcreate /dev/sdv
pvcreate /dev/sdw
pvcreate /dev/sdx
pvcreate /dev/sdy
pvcreate /dev/sdz
pvcreate /dev/sdaa
pvcreate /dev/sdab
pvcreate /dev/sdac
pvcreate /dev/sdad
pvcreate /dev/sdae
pvcreate /dev/sdaf
pvcreate /dev/sdag
pvcreate /dev/sdah
pvcreate /dev/sdai
pvcreate /dev/sdaj
pvcreate /dev/sdak
pvcreate /dev/sdal
pvcreate /dev/sdam
pvcreate /dev/sdan
pvcreate /dev/sdao
pvcreate /dev/sdap
pvcreate /dev/sdaq
```

```
pvcreate /dev/sdar
pvcreate /dev/sdas
pvcreate /dev/sdat
pvcreate /dev/sdau
pvcreate /dev/sdav
pvcreate /dev/sdaw
pvcreate /dev/sdax
pvcreate /dev/sday
pvcreate /dev/sdaz
pvcreate /dev/sdba
pvcreate /dev/sdbb
pvcreate /dev/sdbc
pvcreate /dev/sdbd
pvcreate /dev/sdbe
pvcreate /dev/sdbf
pvcreate /dev/sdbg
pvcreate /dev/sdbh
pvcreate /dev/sdbi
pvcreate /dev/sdbj
pvcreate /dev/sdbk
pvcreate /dev/sdbl
pvcreate /dev/sdbm

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 /dev/sdr /dev/sds /dev/sdt /dev/sdu /dev/sdv /dev/sdw /dev/sdx
   /dev/sdy /dev/sdz /dev/sdaa /dev/sdab /dev/sdac /dev/sdad /dev/sdae /dev/sdaf
   /dev/sdag /dev/sdah /dev/sdai /dev/sdaj /dev/sdak /dev/sdal /dev/sdam
   /dev/sdan /dev/sdao /dev/sdap /dev/sdaq /dev/sdar /dev/sdas /dev/sdat
   /dev/sdau /dev/sdav /dev/sdaw /dev/sdax /dev/sday /dev/sdaz /dev/sdba
   /dev/sdbb /dev/sdbc /dev/sdbd /dev/sdbe /dev/sdbf /dev/sdbg /dev/sdbh
   /dev/sdbi /dev/sdbj /dev/sdbk /dev/sdbl /dev/sdbm

lvcreate -n asu101 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu102 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu103 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu104 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu105 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu106 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu107 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu108 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu109 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu110 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu111 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu112 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu113 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu114 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu115 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu116 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu117 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu118 -i 16 -I 512 -C y -L 4920g vg1

lvcreate -n asu201 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu202 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu203 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu204 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu205 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu206 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu207 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu208 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu209 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu210 -i 16 -I 512 -C y -L 4920g vg1
```

```
lvcreate -n asu211 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu212 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu213 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu214 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu215 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu216 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu217 -i 16 -I 512 -C y -L 4920g vg1
lvcreate -n asu218 -i 16 -I 512 -C y -L 4920g vg1

lvcreate -n asu301 -i 16 -I 512 -C y -L 9840g vg1
lvcreate -n asu302 -i 16 -I 512 -C y -L 9840g vg1
```

APPENDIX E: CONFIGURATION INVENTORY

An inventory of the Tested Storage Configuration was collected during the execution the script *full_run.sh*. It generated the following log file:

- ***profile1_volume.log*** List of configured volumes before the INIT Phase.
- ***Profile2_volume.log*** List of configured volumes after TSC restart.
- ***Step1.log*** List of configured storage after TSC restart.
- ***Step2.log*** List of configured storage after TSC restart.
- ***Step3.log*** List of configured storage after TSC restart.
- ***Step4.log*** List of configured storage after TSC restart.

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

APPENDIX F: WORKLOAD GENERATOR

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 script ***collectinfo.sh*** was used to capture the profile of the storage subsystem and was invoked by ***full_run.sh*** as follows:

- Before the INIT phase, generating ***step1.log***.
- After the INIT phase, generating ***step2.log***.
- After the METRICS phase, generating ***step3.log***.
- After the PERSIST_1 phase, generating ***step4.log***.

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

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/bash
./collectinfo.sh > step1.log
date >> profile1_volume.log
lvdisplay >> profile1_volume.log
date >> profile1_volume.log
echo "Start spc-1 init..."
/root/SPCv302_2017504/spc1 -run SPC1_INIT -iops 60000 -storage slave_asu.asu -
   output ./newtool/spc1_INIT_60k_iops -master host.HST
./collectinfo.sh > step2.log
echo "Start spc-1 verify1..."
/root/SPCv302_2017504/spc1 -run SPC1_VERIFY -iops 100 -storage slave_asu.asu -
   output ./newtool/spc1_VERIFY1_100_iops
echo "Start spc-1 metrics"
/root/SPCv302_2017504/spc1 -run SPC1_METRICS -iops 7000000 -storage slave_asu.asu
   -output ./newtool/spc1_METRICS_7000k_iops -master host.HST
./collectinfo.sh > step3.log
echo "Start spc-1 verify2..."
/root/SPCv302_2017504/spc1 -run SPC1_VERIFY -iops 100 -storage slave_asu.asu -
   output ./newtool/spc1_VERIFY2_100_iops
echo "Start spc-1 persist1..."
/root/SPCv302_2017504/spc1   -run   SPC1_PERSIST_1   -iops   700000   -storage
   slave_asu.asu -output ./newtool/spc1_PERSIST_700k_iops -master host.HST
echo "Power cycle TSC, then Enter to continue"
read
./collectinfo.sh > step4.log
date >> profile2_volume.log
lvdisplay >> profile2_volume.log
date >> profile2_volume.log
echo "Start spc-1 persist2..."
/root/SPCv302_2017504/spc1   -run   SPC1_PERSIST_2   -iops   700000   -storage
   slave_asu.asu -output ./newtool/spc1_PERSIST_700k_iops -master host.HST
echo "spc-1 test end!"
```

collectinfo.sh

```
#!/usr/bin/tclsh

package require Expect
```

```
proc collectstorinfo {} {
#host config


#storage config
set stor_user ibc_os_hs
set stor_pswd Storage@21st
set stor(0) 8.46.203.150
set stor(1) 8.46.203.142
set stor(2) 8.46.203.134
set stor(3) 8.46.203.130
set engine_num 4
set stor_num 4
set host_num 36

#disk_domain config
set dd_num 8
set pool_num 8
#lun_workload config
set grain_size 8KB

#lun config
set lun_num 64
#set lun_num 32
set lun_size 2742GB
#set lun_size 5484GB

spawn ssh $stor_user@$stor(0)
set timeout 60
expect {
    -re "assword" { send "$stor_pswd\n" }
    -re "yes/no" { send "yes\n"; exp_continue }
}

expect {
    "#" { send "/ISM/cli/start.sh -u admin\r" }
    ">" { send "\r" }
    "(y/n):" { send "y\r"; exp_continue }
}
expect ">"
send "change user_mode current_mode user_mode=developer\r"
expect {
        -re ">" { send "\r"; }
        -re "y/n" { send "y\n"; exp_continue }
}
expect ">"
send "show system general\r"
expect ">"
send "show controller general\r"
expect {
        -re ">" { send "\r" }
        -re "--More--" { send "G"; exp_continue }
}
expect ">"
send "show fan\r"
expect {
        -re ">" { send "\r" }
        -re "--More--" { send "G"; exp_continue }
}
expect ">"
send "show power_supply\r"
```

```
expect {
        -re ">" { send "\r" }
        -re "--More--" { send "G"; exp_continue }
}
#expect ">"
#send "change cli capacity_mode=precise\r"
expect ">"
send "show disk_domain general\r"
expect ">"
send "show storage_pool general\r"
expect ">"
#for {set i 0} { $i < $pool_num } { incr i } {
#       send "show storage_pool general pool_id=$i\r"
#  expect ">"
#}
send "show lun general\r"
expect {
                -re ">" { send "\r" }
                -re "--More--" { send "G"; exp_continue }
        }
expect ">"
for {set i 0} { $i < $lun_num } { incr i } {
        send "show lun general lun_id=$i\r"
   expect {
                    -re ">" { send "\r" }
              -re "--More--" { send "G"; exp_continue }
   }
   expect ">"
}
send "show disk general\r"
expect {
        -re ">" { send "\r" }
        -re "--More--" { send "G"; exp_continue }
}
#expect ">"
#send "change cli capacity_mode=automatic\r"
expect ">"
send "show mapping_view general\r"
expect ">"
send "show mapping_view general mapping_view_id=0\r"
expect ">"
send "show mapping_view lun_group mapping_view_id=0\r"
expect ">"
send "show mapping_view host_group mapping_view_id=0\r"
expect ">"
send "show lun_group lun lun_group_id=0\r"
expect {
                -re ">" { send "\r" }
                -re "--More--" { send "G"; exp_continue }
        }
expect ">"
send "show host_group host host_group_id=0\r"
expect ">"
for {set i 0} { $i < $host_num } { incr i } {
        send "show host general host_id=$i\r"
        expect ">"
}
for {set i 0} { $i < $host_num } { incr i } {
        send "show initiator host_id=$i\r"
        expect ">"
}
send "show enclosure\r"
```

```
expect ">"
send "show port general physical_type=FC\r"
expect ">"
send "show port general physical_type=SAS\r"
expect ">"
send "show port general\r"
expect {
        -re ">" { send "\r" }
        -re "--More--" { send "G"; exp_continue }
}
expect ">"
send "exit\r"
expect ">"
send "exit\r"
expect {
   -re "#" { send "\r"; }
        -re ">" { send "\r"; }
        -re "y/n" { send "y\n"; exp_continue }
}
expect "#"
puts "collect storage info success!"
}


##########################
collectstorinfo
puts "collect storage info success!"
```