SPC BENCHMARK 1™
FULL DISCLOSURE REPORT

HUAWEI TECHNOLOGIES CO., LTD.
HUAWEI OCEANSTOR™ 18800 V3

SPC-1 V1.14

Submitted for Review: November 14, 2015
Submission Identifier: A00163

First Edition – November 2015

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

Table of Contents

Audit Certification vii
Audit Certification *(cont.)* viii
Letter of Good Faith ix
Executive Summary 10
Test Sponsor and Contact Information 10
Revision Information and Key Dates 10
Tested Storage Product (TSP) Description 10
Summary of Results 11
Storage Capacities, Relationships, and Utilization 12
Response Time – Throughput Curve 15
Response Time – Throughput Data 15
Priced Storage Configuration Pricing 16
Priced Storage Configuration Pricing *(continued)* 17
Differences between the Tested Storage Configuration (TSC) and Priced Storage Configuration 18
Priced Storage Configuration Diagram 19
Priced Storage Configuration Components 20
Configuration Information 21
Benchmark Configuration (BC)/Tested Storage Configuration (TSC) Diagram 21
Storage Network Configuration 21
Host System(s) and Tested Storage Configuration (TSC) Table of Components 21
Benchmark Configuration/Tested Storage Configuration Diagram 22
Host System and Tested Storage Configuration Components 23
Customer Tunable Parameters and Options 24
Tested Storage Configuration (TSC) Description 24
SPC-1 Workload Generator Storage Configuration 24
ASU Pre-Fill 25
SPC-1 Data Repository 26
Storage Capacities and Relationships 26
SPC-1 Storage Capacities 26
SPC-1 Storage Hierarchy Ratios 27
SPC-1 Storage Capacity Charts 27
Storage Capacity Utilization 29
Logical Volume Capacity and ASU Mapping 30
SPC-1 Benchmark Execution Results 31

SPC-1 Tests, Test Phases, and Test Runs **31**
"Ramp-Up" Test Runs **32**
Primary Metrics Test – Sustainability Test Phase **32**
SPC-1 Workload Generator Input Parameters 33
Sustainability Test Results File 33
Sustainability – Data Rate Distribution Data (MB/second) 33
Sustainability – Data Rate Distribution Graph 33
Sustainability – I/O Request Throughput Distribution Data 34
Sustainability – I/O Request Throughput Distribution Graph 34
Sustainability – Average Response Time (ms) Distribution Data 35
Sustainability – Average Response Time (ms) Distribution Graph 35
Sustainability – Response Time Frequency Distribution Data 36
Sustainability – Response Time Frequency Distribution Graph 36
Sustainability – Measured Intensity Multiplier and Coefficient of Variation 37
Primary Metrics Test – IOPS Test Phase **38**
SPC-1 Workload Generator Input Parameters 38
IOPS Test Results File 38
IOPS Test Run – I/O Request Throughput Distribution Data 39
IOPS Test Run – I/O Request Throughput Distribution Graph 39
IOPS Test Run – Average Response Time (ms) Distribution Data 40
IOPS Test Run – Average Response Time (ms) Distribution Graph 40
IOPS Test Run –Response Time Frequency Distribution Data 41
IOPS Test Run –Response Time Frequency Distribution Graph 41
IOPS Test Run – I/O Request Information 42
IOPS Test Run – Measured Intensity Multiplier and Coefficient of Variation 42
Primary Metrics Test – Response Time Ramp Test Phase **43**
SPC-1 Workload Generator Input Parameters 43
Response Time Ramp Test Results File 43
Response Time Ramp Distribution (IOPS) Data 44
Response Time Ramp Distribution (IOPS) Data *(continued)* 45
Response Time Ramp Distribution (IOPS) Graph 45
SPC-1 LRT™ Average Response Time (ms) Distribution Data 46
SPC-1 LRT™ Average Response Time (ms) Distribution Graph 46
SPC-1 LRT™ (10%) – Measured Intensity Multiplier and Coefficient of Variation 47
Repeatability Test **48**
SPC-1 Workload Generator Input Parameters 48
Repeatability Test Results File 49
Repeatability 1 LRT – I/O Request Throughput Distribution Data 50
Repeatability 1 LRT – I/O Request Throughput Distribution Graph 50

Repeatability 1 LRT –Average Response Time (ms) Distribution Data 51
Repeatability 1 LRT –Average Response Time (ms) Distribution Graph 51
Repeatability 1 IOPS – I/O Request Throughput Distribution Data 52
Repeatability 1 IOPS – I/O Request Throughput Distribution Graph 52
Repeatability 1 IOPS –Average Response Time (ms) Distribution Data 53
Repeatability 1 IOPS –Average Response Time (ms) Distribution Graph 53
Repeatability 2 LRT – I/O Request Throughput Distribution Data 54
Repeatability 2 LRT – I/O Request Throughput Distribution Graph 54
Repeatability 2 LRT –Average Response Time (ms) Distribution Data 55
Repeatability 2 LRT –Average Response Time (ms) Distribution Graph 55
Repeatability 2 IOPS – I/O Request Throughput Distribution Data 56
Repeatability 2 IOPS – I/O Request Throughput Distribution Graph 56
Repeatability 2 IOPS –Average Response Time (ms) Distribution Data 57
Repeatability 2 IOPS –Average Response Time (ms) Distribution Graph 57
Repeatability 1 (LRT) Measured Intensity Multiplier and Coefficient of Variation 58
Repeatability 1 (IOPS) Measured Intensity Multiplier and Coefficient of Variation 58
Repeatability 2 (LRT) Measured Intensity Multiplier and Coefficient of Variation 58
Repeatability 2 (IOPS) Measured Intensity Multiplier and Coefficient of Variation 59
Data Persistence Test 60
SPC-1 Workload Generator Input Parameters 60
Data Persistence Test Results File 60
Data Persistence Test Results 61
Priced Storage Configuration Availability Date 62
Pricing Information 62
Tested Storage Configuration (TSC) and Priced Storage Configuration Differences 62
Anomalies or Irregularities 62
Appendix A: SPC-1 Glossary 63
"Decimal" *(powers of ten)* Measurement Units 63
"Binary" *(powers of two)* Measurement Units 63
SPC-1 Data Repository Definitions 63
SPC-1 Data Protection Levels 64
SPC-1 Test Execution Definitions 64
I/O Completion Types 66
SPC-1 Test Run Components 66
Appendix B: Customer Tunable Parameters and Options 67
Red Hat Enterprise Linux 7.0 (64-bit) 67

Appendix C: Tested Storage Configuration (TSC) Creation 68
Step 1: Create Mapping View, LUN Group, Host Group and Host 68
Step 2: Create Disk Domains, Storage Pools, LUNs 72
Step 3: Create Volumes on the Master Host System 72
Step 4: Change the Scheduler on each Host System 73
Referenced Scripts 73
mklun.sh 73
mkvolume.sh 75
scheduler.sh 78
Appendix D: SPC-1 Workload Generator Storage Commands and Parameters 81
ASU Pre-Fill 81
Common Command Lines – Primary Metrics and Repeatability Tests 82
Primary Metrics and Repeatability Tests 84
Primary Metrics Test: Sustainability Test Phase/Test Run 84
Primary Metrics Test: IOPS Test Phase (100% Test Run) 84
Primary Metrics Test: Response Time Ramp Test Phase *(95% Test Run)* 84
Primary Metrics Test: Response Time Ramp Test Phase *(90% Test Run)* 84
Primary Metrics Test: Response Time Ramp Test Phase *(80% Test Run)* 84
Primary Metrics Test: Response Time Ramp Test Phase *(50% Test Run)* 84
Repeatability Test: Repeatability Test Phase 1 *(10% Test Run)* 84
Repeatability Test: Repeatability Test Phase 1 *(100% Test Run)* 84
Repeatability Test: Repeatability Test Phase 2 *(10% Test Run)* 84
Repeatability Test: Repeatability Test Phase 2 *(100% Test Run)* 84
SPC-1 Persistence Test Run 1 85
SPC-2 Persistence Test 86
Common Command Lines – SPC-2 Persistence Test 86
SPC-2 Persistence Test Run 1 (write phase) 93
SPC-2 Persistence Test Run 2 (read phase) 93
Slave JVMs 93
Appendix E: SPC-1 Workload Generator Input Parameters 95
run.sh 95
slave.sh 96
killjava.sh 98
Appendix F: Third-Party Quotation 99
Priced Storage Configuration 99
Priced Storage Configuration *(continued)* 100

AUDIT CERTIFICATION

Xu Zhong
Huawei Technologies Co., Ltd.
Huawei Chengdu Base
No. 1899, Xiyuan Avenue
Chengdu, 611731 P.R. China

November 13, 2015

The SPC Benchmark 1™ Reported Data listed below for the **Huawei OceanStor™ 18800 V3** was produced in compliance with the SPC Benchmark 1™ v1.14 Remote Audit requirements.

SPC Benchmark 1™ v1.14 Reported Data	
Tested Storage Product (TSP) Name: Huawei OceanStor™ 18800 V3	
Metric	**Reported Result**
SPC-1 IOPS™	3,010,077.37
SPC-1 Price-Performance	$0.79/SPC-1 IOPS™
Total ASU Capacity	68,934.225 GB
Data Protection Level	Protected 1 *(Mirroring)*
Total Price (including three-year maintenance)	$2,370,763.89
Currency Used	U.S. Dollars
Target Country for availability, sales and support	U.S.A.

The following SPC Benchmark 1™ Remote Audit requirements were reviewed and found compliant with 1.14 of the SPC Benchmark 1™ specification:

- A Letter of Good Faith, signed by a senior executive.
- The following Data Repository storage items were verified by visual inspection and information supplied by Huawei Technologies Co., Ltd.:
  - ✓ Physical Storage Capacity and requirements.
  - ✓ Configured Storage Capacity and requirements.
  - ✓ Addressable Storage Capacity and requirements.
  - ✓ Capacity of each Logical Volume and requirements.
  - ✓ Capacity of each Application Storage Unit (ASU) and requirements.

Storage Performance Council
643 Bair Island Road, Suite 103
Redwood City, CA 94062
AuditService@storageperformance.org
650.556.9384

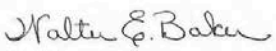
AUDIT CERTIFICATION *(CONT.)*

- The total Application Storage Unit (ASU) Capacity was filled with random data, using an auditor approved tool, prior to execution of the SPC-1 Tests.
- An appropriate diagram of the Benchmark Configuration (BC)/Tested Storage Configuration (TSC).
- Listings and commands to configure the Benchmark Configuration/Tested Storage Configuration, including any customer tunable parameters that were changed from default values.
- SPC-1 Workload Generator commands and parameters used for the audited SPC Test Runs.
- The following Host System requirements were verified by visual inspection and information supplied by Huawei Technologies Co., Ltd.:
  - ✓ The type of Host Systems including the number of processors and main memory.
  - ✓ The presence and version number of the SPC-1 Workload Generator on each Host System.
  - ✓ The TSC boundary within each Host System.
- The execution of each Test, Test Phase, and Test Run was observed and found compliant with all of the requirements and constraints of Clauses 4, 5, and 11 of the SPC-1 Benchmark Specification.
- The Test Results Files and resultant Summary Results Files received from Huawei Technologies Co., Ltd. for each of following were authentic, accurate, and compliant with all of the requirements and constraints of Clauses 4 and 5 of the SPC-1 Benchmark Specification:
  - ✓ Data Persistence Test
  - ✓ Sustainability Test Phase
  - ✓ IOPS Test Phase
  - ✓ Response Time Ramp Test Phase
  - ✓ Repeatability Test
- There were no differences between the Tested Storage Configuration (TSC) and Priced Storage Configuration.
- The submitted pricing information met all of the requirements and constraints of Clause 8 of the SPC-1 Benchmark Specification.
- The Full Disclosure Report (FDR) met all of the requirements in Clause 9 of the SPC-1 Benchmark Specification.
- This successfully audited SPC measurement is not subject to an SPC Confidential Review.

Audit Notes:

There were no audit notes or exceptions.

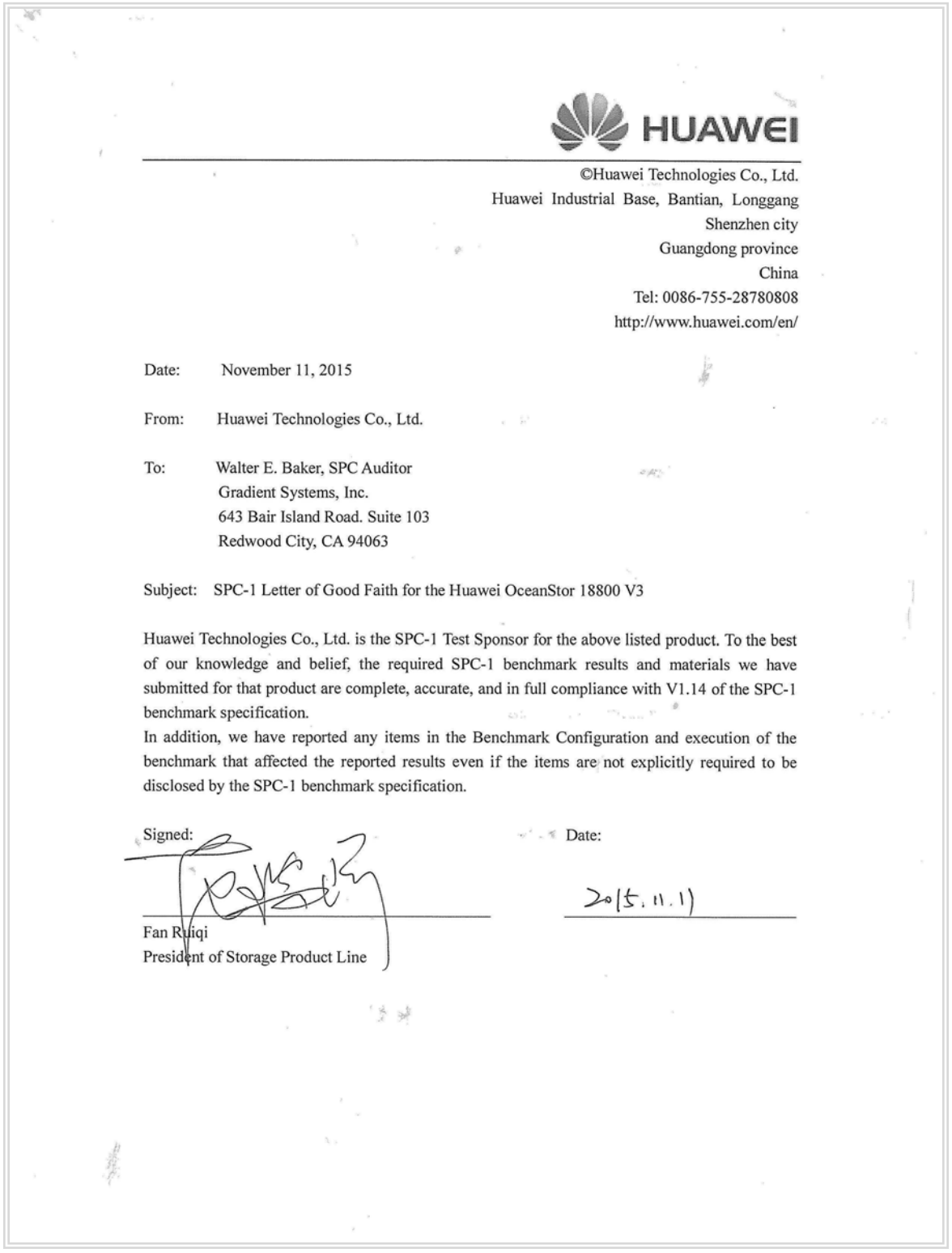
Respectfully,



Walter E. Baker
SPC Auditor

Storage Performance Council
643 Bair Island Road, Suite 103
Redwood City, CA 94062
AuditService@storageperformance.org
650.556.9384

LETTER OF GOOD FAITH



HUAWEI

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

Date: November 11, 2015

From: Huawei Technologies Co., Ltd.

To: Walter E. Baker, SPC Auditor
Gradient Systems, Inc.
643 Bair Island Road. Suite 103
Redwood City, CA 94063

Subject: SPC-1 Letter of Good Faith for the Huawei OceanStor 18800 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 V1.14 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: 2015.11.1

Fan Ruiqi
President of Storage Product Line

EXECUTIVE SUMMARY

Test Sponsor and Contact Information

Test Sponsor and Contact Information	
Test Sponsor Primary Contact	Huawei Technologies Co., Ltd. – http://www.huawei.com/en/ Xu Zhong – xuzhong@huawei.com Huawei Chengdu Base No. 1899, Xiyuan Avenue Chengdu, 611731 P.R. China Phone: 86 28 65281927 FAX: 86 28 62282516
Test Sponsor Alternate Contact	Huawei Technologies Co., Ltd. – http://www.huawei.com/en/ Li Huan – tomas.l@huawei.com Huawei Chengdu Base No. 1899, Xiyuan Avenue Chengdu, 611731 P.R. China Phone: 86 28 65281927 FAX: 86 28 62282516
Auditor	Storage Performance Council – http://www.storageperformance.org Walter E. Baker – AuditService@StoragePerformance.org 643 Bair Island Road, Suite 103 Redwood City, CA 94063 Phone: (650) 556-9384 FAX: (650) 556-9385

Revision Information and Key Dates

Revision Information and Key Dates	
SPC-1 Specification revision number	V1.14
SPC-1 Workload Generator revision number	V2.3.0
Date Results were first used publicly	November 14, 2015
Date the FDR was submitted to the SPC	November 14, 2015
Date the Priced Storage Configuration is available for shipment to customers	currently available
Date the TSC completed audit certification	November 13, 2015

Tested Storage Product (TSP) Description

The OceanStor 18000 V3 series storage systems are the new benchmark for high-end storage, providing the best data services for enterprises' mission-critical businesses.

Scalable to 16 controllers, 16 TB of cache, and 27.6 PB total capacity, the 18000 V3 system handles large-scale OLTP and OLAP database applications with ease. The system's 3 million IOPS and extremely low latency give life to cloud computing applications.

Available in 2 models for large-scale, data-intensive applications in government, finance, telecommunications, power utilities, transportation, and manufacturing industries.

Summary of Results

SPC-1 Reported Data	
Tested Storage Product (TSP) Name: Huawei OceanStor™ 18800 V3	
Metric	**Reported Result**
SPC-1 IOPS™	3,010,007.37
SPC-1 Price-Performance™	$0.79/SPC-1 IOPS™
Total ASU Capacity	68,934.225 GB
Data Protection Level	Protected 1 *(Mirroring)*
Total Price	$2,370,763.89
Currency Used	U.S. Dollars
Target Country for availability, sales and support	USA

SPC-1 IOPS™ represents the maximum I/O Request Throughput at the 100% load point.

SPC-1 Price-Performance™ is the ratio of **Total Price** to **SPC-1 IOPS™**.

Total ASU (Application Storage Unit) **Capacity** represents the total storage capacity available to be read and written in the course of executing the SPC-1 benchmark.

A **Data Protection Level** of **Protected 1** using ***Mirroring*** configures two or more identical copies of user data.

> ***Protected 1:*** *The single point of failure of any* ***storage device*** *in the configuration will not result in permanent loss of access to or integrity of the SPC-1 Data Repository.*

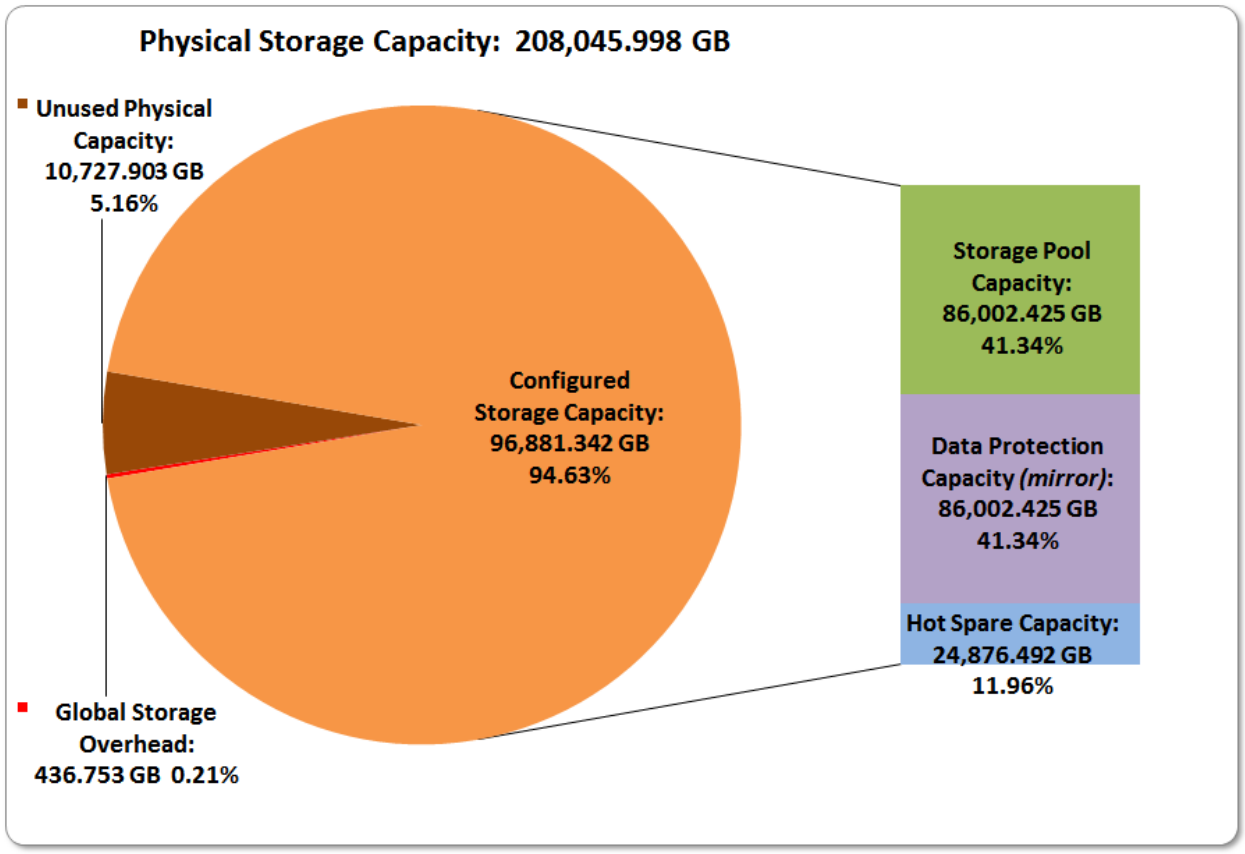
Total Price includes the cost of the Priced Storage Configuration plus three years of hardware maintenance and software support as detailed on page 16.

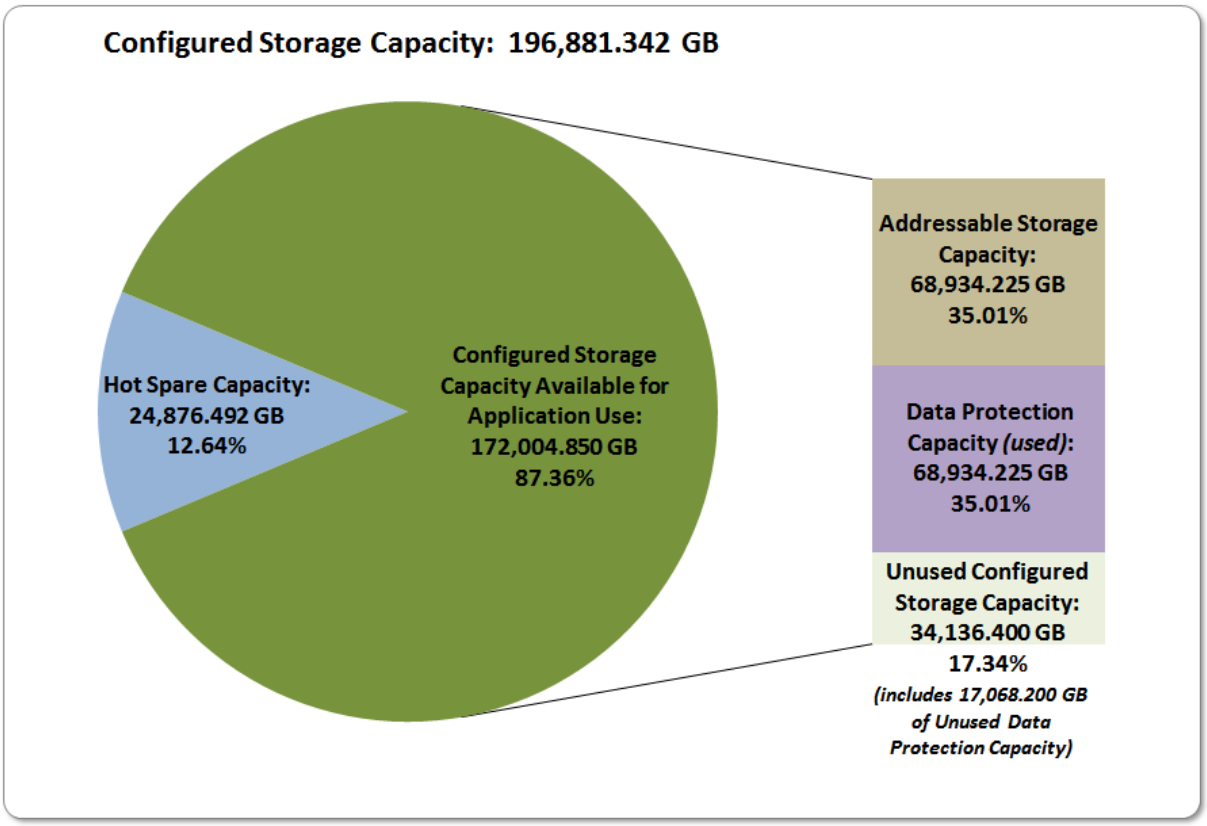
Currency Used is formal name for the currency used in calculating the **Total Price** and **SPC-1 Price-Performance™**. That currency may be the local currency of the **Target Country** or the currency of a difference country *(non-local currency)*.

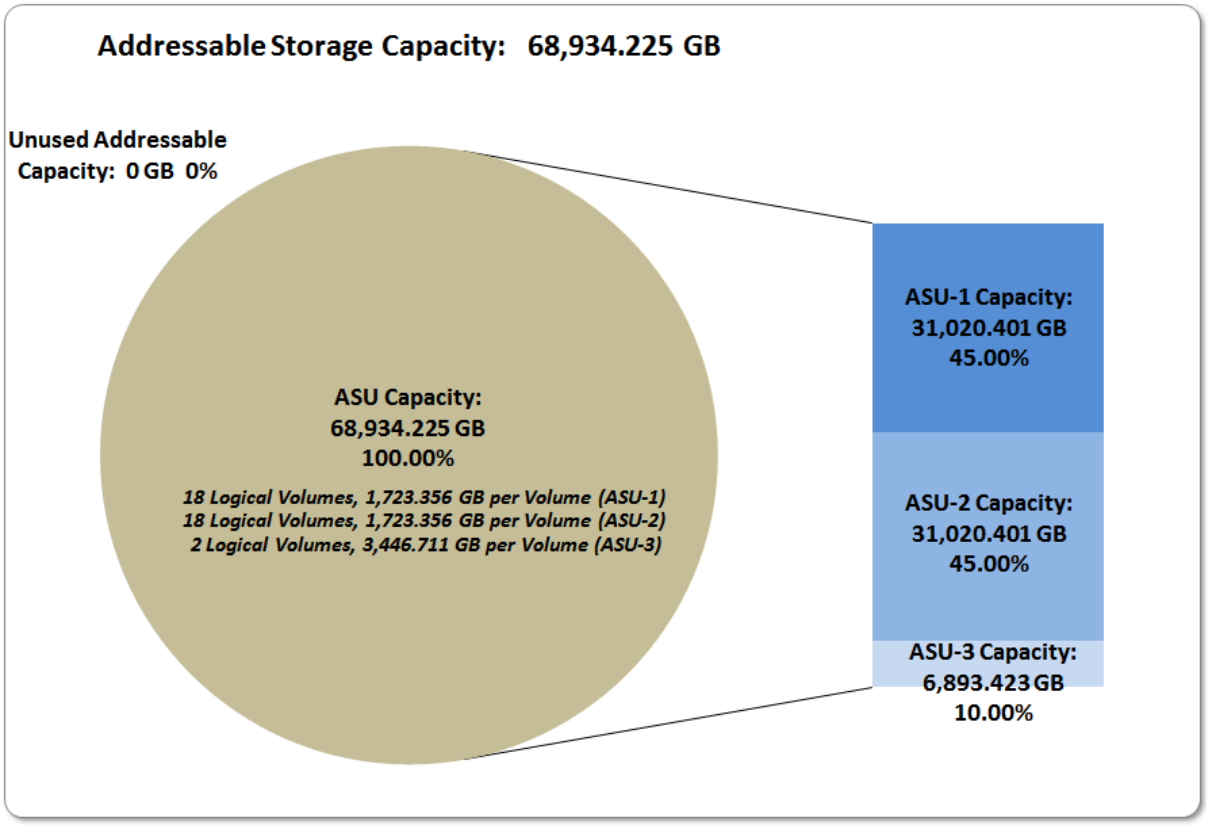
The **Target Country** is the country in which the Priced Storage Configuration is available for sale and in which the required hardware maintenance and software support is provided either directly from the Test Sponsor or indirectly via a third-party supplier.

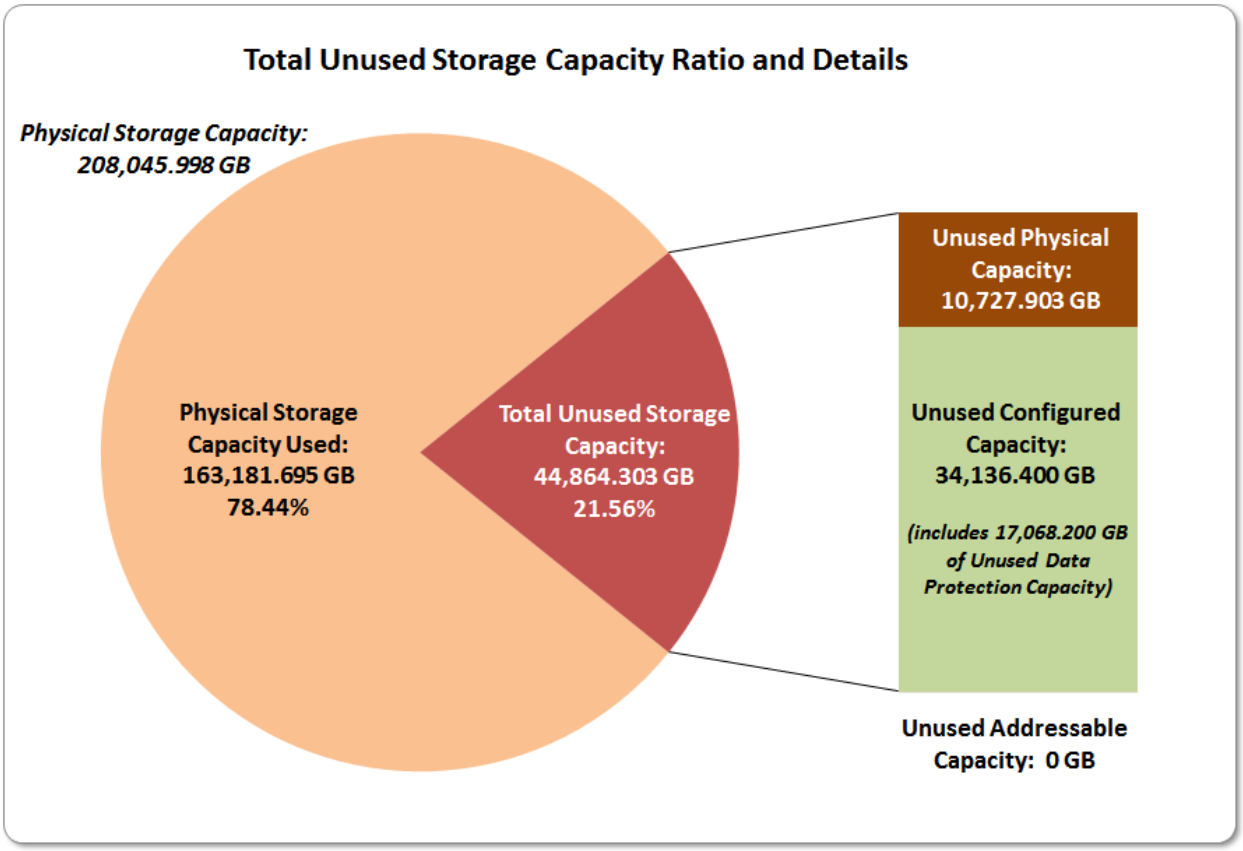
Storage Capacities, Relationships, and Utilization

The following four charts and table document the various storage capacities, used in this benchmark, and their relationships, as well as the storage utilization values required to be reported.




Configured Storage Capacity: 196,881.342 GB
Hot Spare Capacity:
24,876.492 GB
12.64%
Configured Storage Capacity Available for Application Use:
172,004.850 GB
87.36%
Addressable Storage Capacity:
68,934.225 GB
35.01%
Data Protection Capacity (used):
68,934.225 GB
35.01%
Unused Configured Storage Capacity:
34,136.400 GB
17.34%
(includes 17,068.200 GB of Unused Data Protection Capacity)


Addressable Storage Capacity: 68,934.225 GB
Unused Addressable Capacity: 0 GB 0%
ASU Capacity:
68,934.225 GB
100.00%
18 Logical Volumes, 1,723.356 GB per Volume (ASU-1)
18 Logical Volumes, 1,723.356 GB per Volume (ASU-2)
2 Logical Volumes, 3,446.711 GB per Volume (ASU-3)
ASU-1 Capacity:
31,020.401 GB
45.00%
ASU-2 Capacity:
31,020.401 GB
45.00%
ASU-3 Capacity:
6,893.423 GB
10.00%



SPC-1 Storage Capacity Utilization	
Application Utilization	33.13%
Protected Application Utilization	66.27%
Unused Storage Ratio	21.56%

Application Utilization: Total ASU Capacity *(68,934.225 GB)* divided by Physical Storage Capacity *(208,045.998 GB)*.

Protected Application Utilization: (Total ASU Capacity *(68,934.225 GB)* plus total Data Protection Capacity *(86,002.425 GB)* minus unused Data Protection Capacity *(17,068.200 GB)*) divided by Physical Storage Capacity *(208,045.998 GB)*.

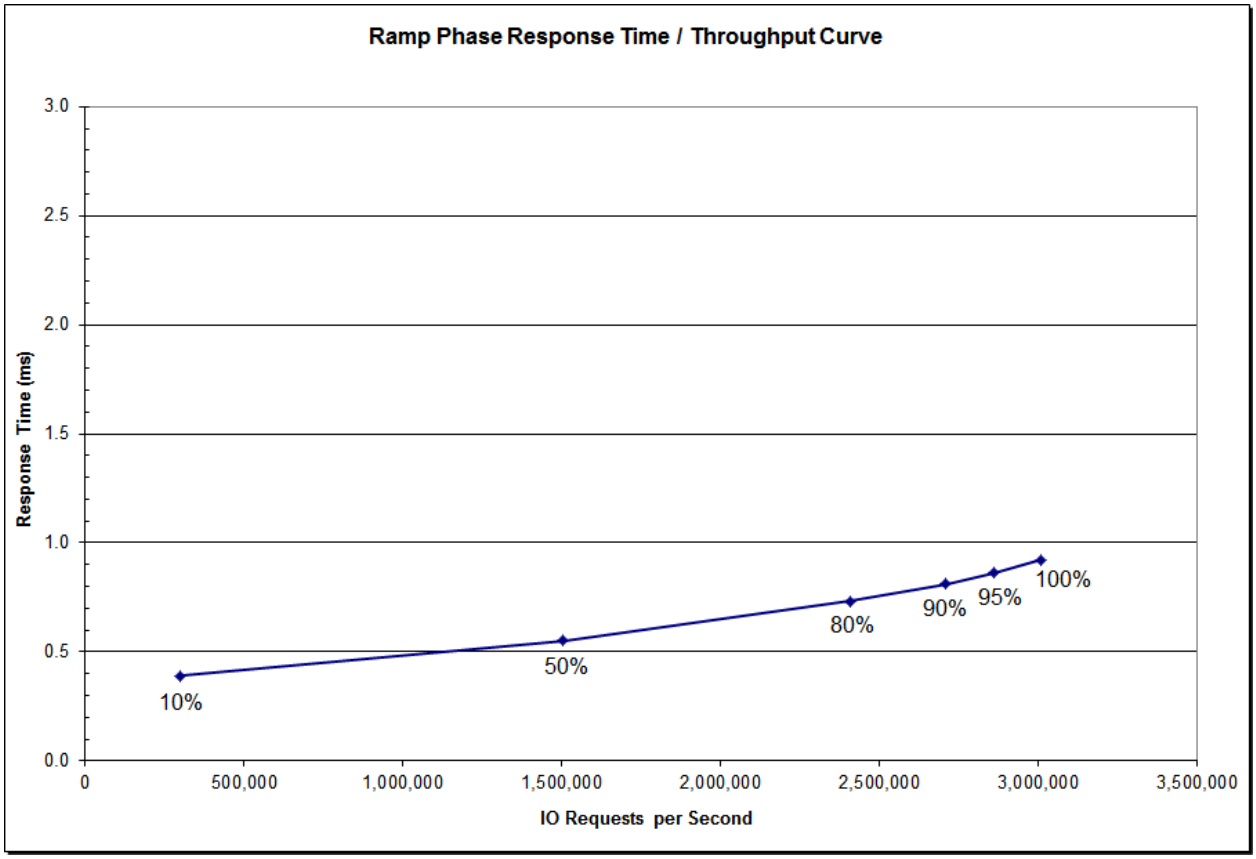
Unused Storage Ratio: Total Unused Capacity *(44,864.303 GB)* divided by Physical Storage Capacity *(208,045.998 GB)* and may not exceed 45%.

Detailed information for the various storage capacities and utilizations is available on pages 26-27.

Response Time – Throughput Curve

The Response Time-Throughput Curve illustrates the Average Response Time (milliseconds) and I/O Request Throughput at 100%, 95%, 90%, 80%, 50%, and 10% of the workload level used to generate the SPC-1 IOPS™ metric.

The Average Response Time measured at any of the above load points cannot exceed 30 milliseconds or the benchmark measurement is invalid.



Response Time – Throughput Data

	10% Load	50% Load	80% Load	90% Load	95% Load	100% Load
I/O Request Throughput	300,985.93	1,505,067.60	2,407,867.14	2,708,970.22	2,859,478.87	3,010,007.37
Average Response Time (ms):						
All ASUs	0.39	0.55	0.73	0.81	0.86	0.92
ASU-1	0.38	0.58	0.77	0.86	0.91	0.97
ASU-2	0.42	0.63	0.85	0.95	1.01	1.07
ASU-3	0.39	0.46	0.59	0.65	0.70	0.75
Reads	0.39	0.71	0.97	1.08	1.15	1.22
Writes	0.39	0.45	0.58	0.64	0.68	0.73

Priced Storage Configuration Pricing

No.	Model	Description	Qty.	Unit Price (USD)	Total Price (USD)
1	**Phase**				
1.1	**Location**				
1.1.1	**OceanStor 18800 V3 Storage System**				
1.1.1	**Engine**				
	88V3-4C1T-AC	OceanStor 18800 V3 Engine(Four Controller,AC\240HVDC,1TB Cache,SPE72C0600)	4	67,081.80	268,327.20
1.1.2	**Expand Interface Module**				
	LPU5PCIEV3H	2 port PCIe I/O module	16	811.06	12,976.96
	SMARTIO8FCV3H	4 port SmartIO I/O module(SFP+,8Gb FC)	48	972.17	46,664.16
	LPU12S12V3H	12 port 12Gb SAS Entire Sharing I/O module(MiniSAS HD)	16	3,962.48	63,399.68
1.1.3	**Disk Components**				
	SSD400-2-H	400GB SSD eMLC SAS Disk Unit(2.5")	512	3,174.65	1,625,420.80
1.1.4	**Disk Enclosure**				
	DAE22525U2-H-AC	Disk Enclosure(2U,2.5",AC\240HVDC,DAE22525U2)	24	1,757.80	42,187.20
1.1.5	**Cabinet**				
	RACK-SYS-H-AC	OceanStor 18000 V3 Series System Cabinet	2	6,422.33	12,844.66
1.1.6	**Option Class Item**				
	SVP4-V3H	Service Processor (1U, AC\240HDVC,8GB Cache,Including Windows OS Software and Security software)	1	3,760.39	3,760.39
	WM1P0CIKVM02	KVM,KVM 4 in 1 Control Module,1U, 17" LED, 8 KVM ports, With Power Cable,8 USB Straight signal cables/With mounting Accessories,English doc,110V/240V AC,Black,Compliant	1	585.00	585.00
	SWITCH-V3H	PCIe Switch(AC\240HVDC,2GB Cache,16 Port,SWE1603P05)	2	1,587.47	3,174.94
	OQSFPOM00	Quadwire 40 Gb/s Parallel AOC	32	1,494.00	47,808.00
	PDU2000-V3-H	AC Power Distribution Unit	8	127.00	1,016.00
	HS-SAS-1-01	High Speed Cable,External MiniSAS HD Cable,1m,(SFF 8644 Plug),(28AWG*4P*2B(S)),(SFF 8644 Plug),Indoor use	32	55.00	1,760.00
	HS-SAS-3-01	High Speed Cable,Mini SAS HD Cable,3m,(SFF 8644 Plug),(28AWG*4P*2B(S)),(SFF 8644 Plug),Indoor use	16	96.00	1,536.00
1.1.7	**HBA**				
	N8GHBA000	QLOGIC QLE2562 HBA Card,PCIE,8Gbps DualPort ,Fiber Channel Multimode LC Optic Interface,English Manual, No Drive CD	80	1,000.00	80,000.00
1.1.8	**Accessory**				
	VADMSMR02	Software Service,Trend Micro,05280169,Original Manufacturer Service, 1 Year,7*24 Standard Service,10 users	2	314.00	628.00
	SN2F01FCPC	Patch Cord,DLC/PC,DLC/PC,Multi-mode,3m,A1a.2,2mm,OM3 bending insensitive	160	11.00	1,760.00
	P-16mm^2-Olivine-LSZH	Power Cable,450V/750V,H07Z-K UL3386,16mm^2,Yellow/Green,107A,LSZH Cable,VDE,UL (Unit:meter)	10	3.80	38.00
	C3006BK01	Power Cable,600V/1000V,ZA-RVV,3x6mm^2,Black(3Cores:Brown,Blue,Yellow/Green), 46A,Outdoor Cable,CE (Unit:meter)	80	6.90	552.00

Priced Storage Configuration Pricing *(continued)*

1.1.9	**Storage Software**				
	88V3-LBASIC-N	Basic Software Suite License(OceanStor OS,DeviceManager,SmartThin,SmartMotion,SmartQos,SmartPartition,SmartCache,SmartMigration,SmartErase,SmartMulti-tenant,System Reporter,Cloud Service)	1	2,815.25	2,815.25
	88V3-LBASIC200	Basic Software Suite Capacity License(101-200TB)	200	160.87	32,174.00
	88V3-LBASICU	Basic Software Suite Unlimited Capacity License	1	0.01	0.01
	88V3-LULTRAPATH	OceanStor UltraPath Software License	1	596.37	596.37
Total of Product					**2,250,024.62**
1.1.10	**Maintenance Support Service**				
	88125ESH	OceanStor 18800 V3 Installation Service - Engineering	1	107,346.67	107,346.67
	88032XVE-88134UHK-3	Basic Software Suite Capacity License(101-200TB)-Hi-Care Application Software Upgrade Support Service-3Year(s)	200	62.74	12,548.00
	88032XVG-88134UHK-3	Basic Software Suite Unlimited Capacity License-Hi-Care Application Software Upgrade Support Service-3Year(s)	1	0.01	0.01
	88032YCT-88134UHK-3	OceanStor UltraPath Software License-Hi-Care Application Software Upgrade Support Service-3Year(s)	1	0.01	0.01
	88033JKR-88134UHK-3	Basic Software Suite License(OceanStor OS,DeviceManager,SmartThin,SmartMotion,SmartQos,SmartPartition,SmartCache,SmartMigration,SmartErase,SmartMulti-tenant,System Reporter,Cloud Service)-Hi-Care Application Software Upgrade Support Service-3Year(s)	1	844.58	844.58
Total of Service (3 years)					**120,739.27**
Total Price					**2,370,763.89**

The above pricing includes hardware maintenance and software support for three years, 7 days per week, 24 hours per day. The hardware maintenance and software support provides the following:

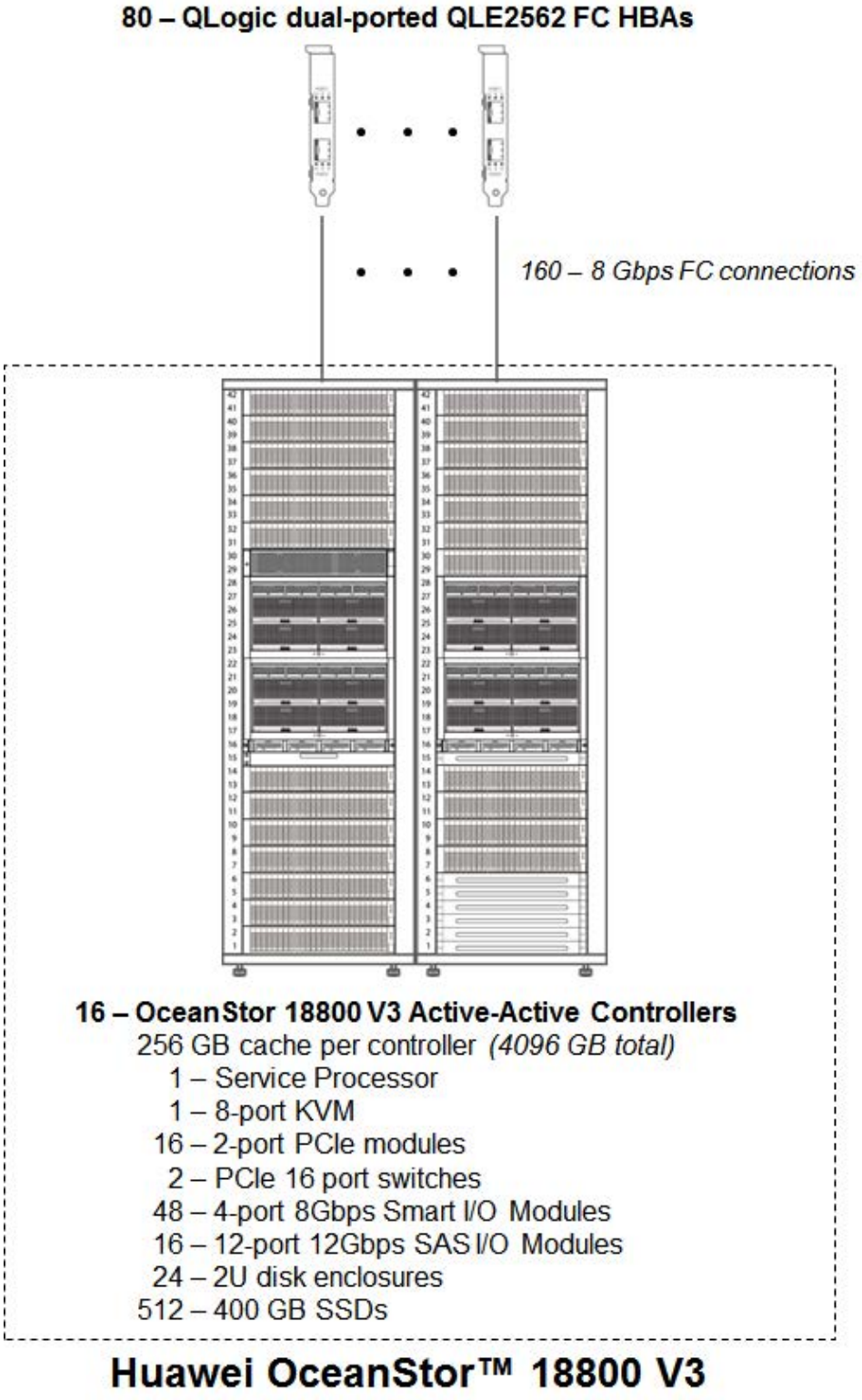
- Acknowledgement of new and existing problems within four (4) hours.
- Onsite presence of a qualified maintenance engineer or provision of a customer replaceable part within four (4) hours of the above acknowledgement for any hardware failure that results in an inoperative Price Storage Configuration that can be remedied by the repair or replacement of a Priced Storage Configuration component.

Huawei Technologies Co., Ltd. only sells its products to third-party resellers, who in turn, sell those products to U.S. customers. The above pricing, which also includes the required three-year maintenance and support, was obtained from one of those third-party resellers. See page 99 (*Appendix F: Third-Party Quotation*) for a copy of the third-party reseller quotation.

Differences between the Tested Storage Configuration (TSC) and Priced Storage Configuration

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

Priced Storage Configuration Diagram



Priced Storage Configuration Components

Priced Storage Configuration
OceanStor UltraPath
80 – QLogic QLE2562 dual-port, 8 Gbps, FC HBAs
Huawei OceanStor™ 18800 V3 16 – Active-Active Controllers each controller includes: 256 GB cache *(4,096 GB total)* 3 – 4-port 8Gbps Smart I/O modules *(48 modules total, 12 ports per controller, 192 ports total)* *(10 ports used per controller, 160 ports total used)* 1 – 12-port 12 Gbps SAS I/O module *(16 modules total, 192 ports total)* *(3 ports used per controller, 48 ports total used)* 1 – Service Processor 1 – 8-port KVM
16 – 2-port PCIe modules *(for inter-controller connectivity)*
2 – PCIe 16 port switches *(for inter-controller connectivity)*
24 – Disk Enclosures *(2U, 2.5")*
512 – 400 GB SSDs *24 SSDs in 16 enclosures* *16 SSDs in 8 enclosures*
2 – OceanStor 18000 V3 Series System Cabinet
8 – AC Power Distribution Units

In each of the following sections of this document, the appropriate Full Disclosure Report requirement, from the SPC-1 benchmark specification, is stated in italics followed by the information to fulfill the stated requirement.

CONFIGURATION INFORMATION

Benchmark Configuration (BC)/Tested Storage Configuration (TSC) Diagram

Clause 9.4.3.4.1

A one page Benchmark Configuration (BC)/Tested Storage Configuration (TSC) diagram shall be included in the FDR…

The Benchmark Configuration (BC)/Tested Storage Configuration (TSC) is illustrated on page 22 (*Benchmark Configuration/Tested Storage Configuration Diagram*).

Storage Network Configuration

Clause 9.4.3.4.1

…

5. *If the TSC contains network storage, the diagram will include the network configuration. If a single diagram is not sufficient to illustrate both the Benchmark Configuration and network configuration in sufficient detail, the Benchmark Configuration diagram will include a high-level network illustration as shown in Figure 9-8. In that case, a separate, detailed network configuration diagram will also be included as described in Clause 9.4.3.4.2.*

Clause 9.4.3.4.2

If a storage network was configured as a part of the Tested Storage Configuration and the Benchmark Configuration diagram described in Clause 9.4.3.4.1 contains a high-level illustration of the network configuration, the Executive Summary will contain a one page topology diagram of the storage network as illustrated in Figure 9-9.

The Tested Storage Configuration (TSC) was configured with direct-attached storage.

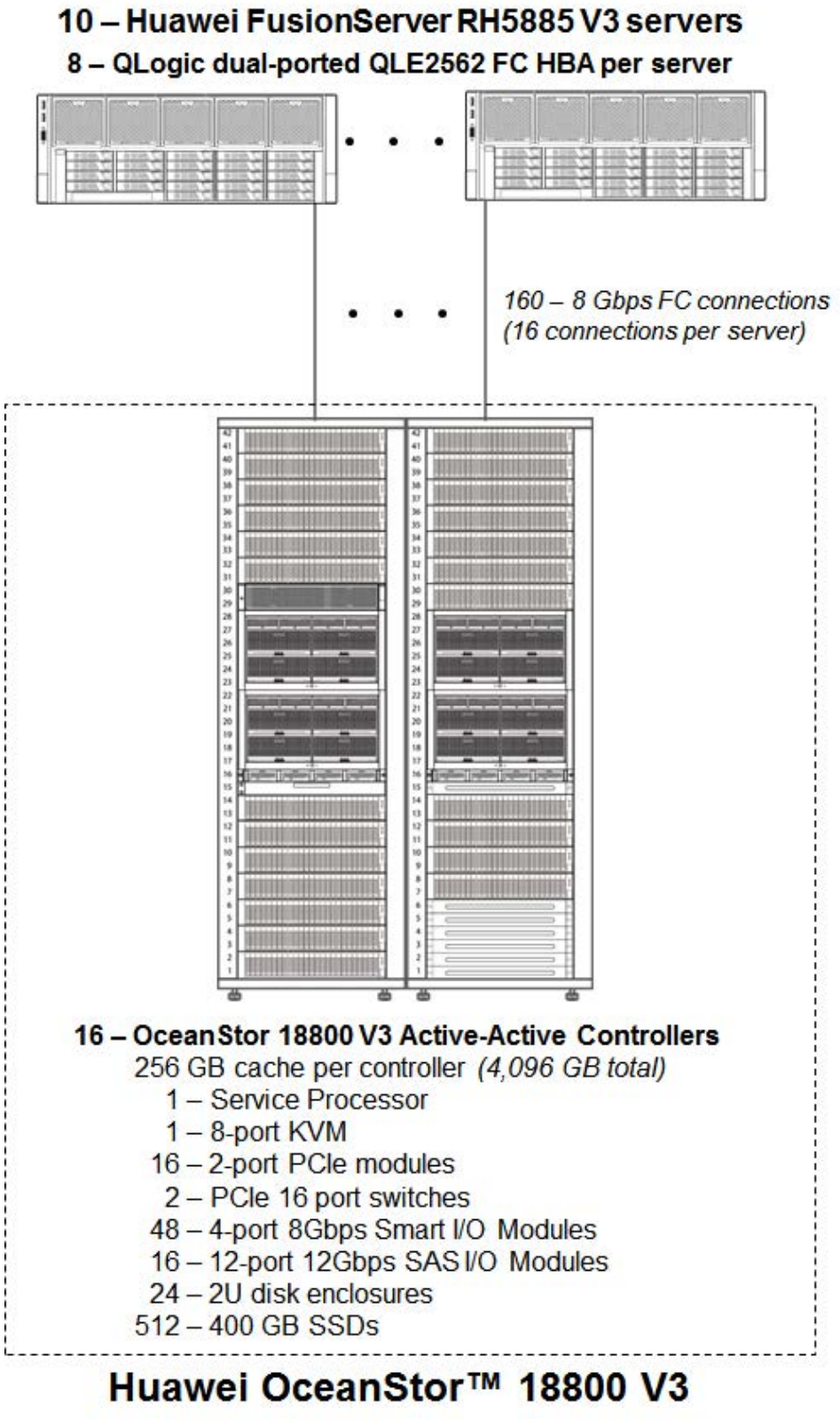
Host System(s) and Tested Storage Configuration (TSC) Table of Components

Clause 9.4.3.4.3

The FDR will contain a table that lists the major components of each Host System and the Tested Storage Configuration (TSC).

The Host System(s) and TSC table of components may be found on page 23 (*Host System and Tested Storage Configuration Components*).

Benchmark Configuration/Tested Storage Configuration Diagram



Host System and Tested Storage Configuration Components

Host Systems
16 – Huawei FusionServer RH5885 V3 servers, each with: 4 – Intel® Xeon® 2.00 GHz processor E7-4820 V3 each with 6 cores, 16 MB cache 512 GB main memory Red Hat Enterprise Linux Server release 7.0 x86_64 PCIe
Priced Storage Configuration
OceanStor UltraPath
80 – QLogic QLE2562 dual-port, 8 Gbps, FC HBAs
Huawei OceanStor™ 18800 V3 16 – Active-Active Controllers each controller includes: 256 GB cache *(4,096 GB total)* 3 – 4-port 8Gbps Smart I/O modules *(48 modules total, 12 ports per controller, 192 ports total)* *(10 ports used per controller, 160 ports total used)* 1 – 12-port 12 Gbps SAS I/O module *(16 modules total, 192 ports total)* *(3 ports used per controller, 48 ports total used)* 1 – Service Processor 1 – 8-port KVM
16 – 2-port PCIe modules *(for inter-controller connectivity)*
2 – PCIe 16 port switches *(for inter-controller connectivity)*
24 – Disk Enclosures *(2U, 2.5")*
512 – 400 GB SSDs *24 SSDs in 16 enclosures* *16 SSDs in 8 enclosures*
2 – OceanStor 18000 V3 Series System Cabinet
8 – AC Power Distribution Units

Customer Tunable Parameters and Options

Clause 9.4.3.5.1

All Benchmark Configuration (BC) components with customer tunable parameter and options that have been altered from their default values must be listed in the FDR. The FDR entry for each of those components must include both the name of the component and the altered value of the parameter or option. If the parameter name is not self-explanatory to a knowledgeable practitioner, a brief description of the parameter's use must also be included in the FDR entry.

Appendix B: Customer Tunable Parameters and Options on page 67 contains the customer tunable parameters and options that have been altered from their default values for this benchmark.

Tested Storage Configuration (TSC) Description

Clause 9.4.3.5.2

The FDR must include sufficient information to recreate the logical representation of the TSC. In addition to customer tunable parameters and options (Clause 4.2.4.5.3), that information must include, at a minimum:

- *A diagram and/or description of the following:*
  - *All physical components that comprise the TSC. Those components are also illustrated in the BC Configuration Diagram in Clause 9.2.4.4.1 and/or the Storage Network Configuration Diagram in Clause 9.2.4.4.2.*
  - *The logical representation of the TSC, configured from the above components that will be presented to the Workload Generator.*
- *Listings of scripts used to create the logical representation of the TSC.*
- *If scripts were not used, a description of the process used with sufficient detail to recreate the logical representation of the TSC.*

Appendix C: Tested Storage Configuration (TSC) Creation on page 68 contains the detailed information that describes how to create and configure the logical TSC.

SPC-1 Workload Generator Storage Configuration

Clause 9.4.3.5.3

The FDR must include all SPC-1 Workload Generator storage configuration commands and parameters.

The SPC-1 Workload Generator storage configuration commands and parameters for this measurement appear in Appendix D: SPC-1 Workload Generator Storage Commands and Parameters on page 81.

ASU Pre-Fill

Clause 5.3.3

Each of the three SPC-1 ASUs (ASU-1, ASU-2 and ASU-3) is required to be completely filled with specified content prior to the execution of audited SPC-1 Tests. The content is required to consist of random data pattern such as that produced by an SPC recommended tool.

The configuration file used to complete the required ASU pre-fill appears in Appendix D: SPC-1 Workload Generator Storage Commands and Parameters on page 81.

SPC-1 DATA REPOSITORY

This portion of the Full Disclosure Report presents the detailed information that fully documents the various SPC-1 storage capacities and mappings used in the Tested Storage Configuration. SPC-1 Data Repository Definitions on page 63 contains definitions of terms specific to the SPC-1 Data Repository.

Storage Capacities and Relationships

Clause 9.4.3.6.1

Two tables and four charts documenting the storage capacities and relationships of the SPC-1 Storage Hierarchy (Clause 2.1) shall be included in the FDR. … The capacity value in each chart may be listed as an integer value, for readability, rather than the decimal value listed in the table below.

SPC-1 Storage Capacities

The Physical Storage Capacity consisted of 208,045.998 GB distributed over 512 solid state devices (SSDs), each with a formatted capacity of 406.340 GB per SSD. There was 10,727.902 GB (5.16%) of Unused Storage within the Physical Storage Capacity. Global Storage Overhead consisted of 436,753 GB (0.21%) of the Physical Storage Capacity. There was 34,136.400 GB (17.34%) of Unused Storage within the Configured Storage Capacity. The Total ASU Capacity utilized 100% of the Addressable Storage Capacity resulting in 0.00 GB (0.00%) of Unused Storage within the Addressable Storage Capacity. The Data Protection *(Mirroring)* capacity was 86,002.425 GB of which 68,934.225 GB was utilized. The total Unused Storage capacity was 44,864.303 GB.

Note: The configured Storage Devices may include additional storage capacity reserved for system overhead, which is not accessible for application use. That storage capacity may not be included in the value presented for Physical Storage Capacity.

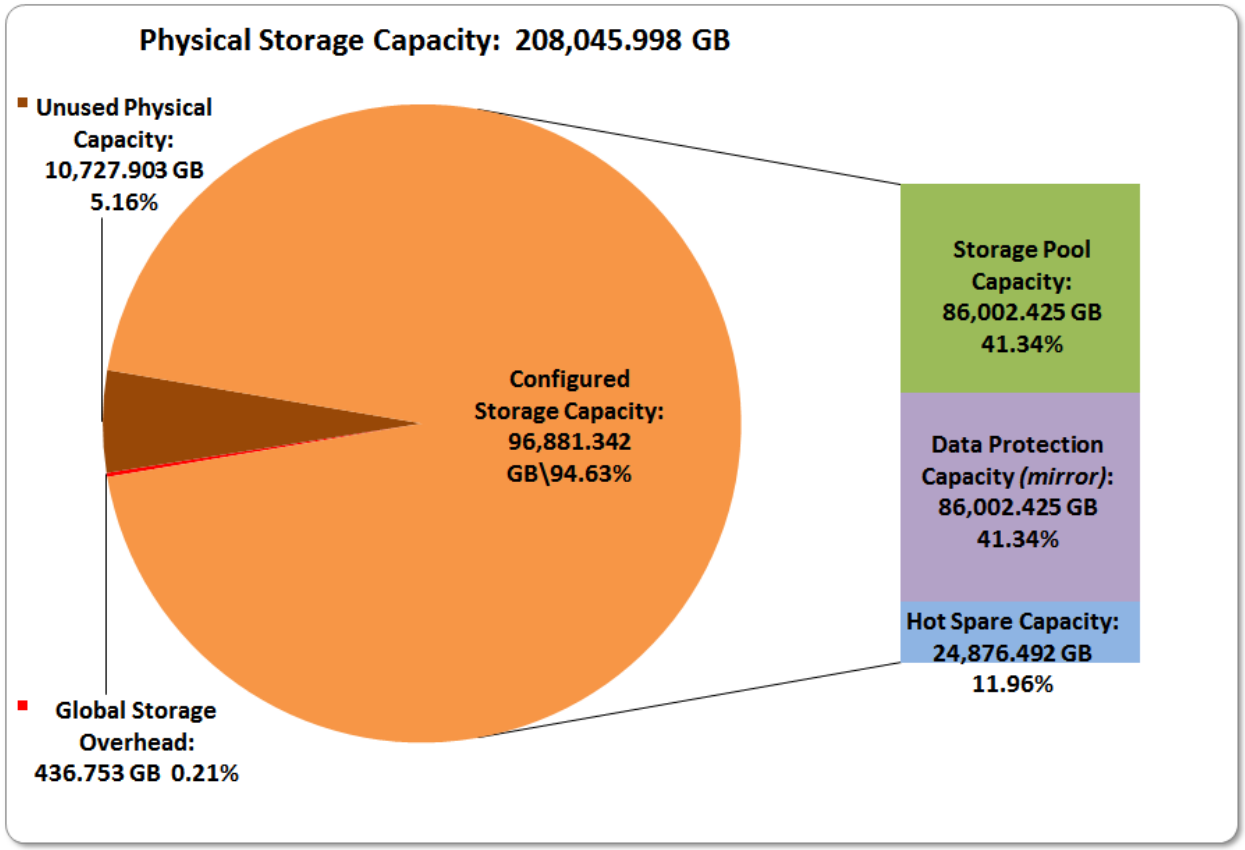
SPC-1 Storage Capacities

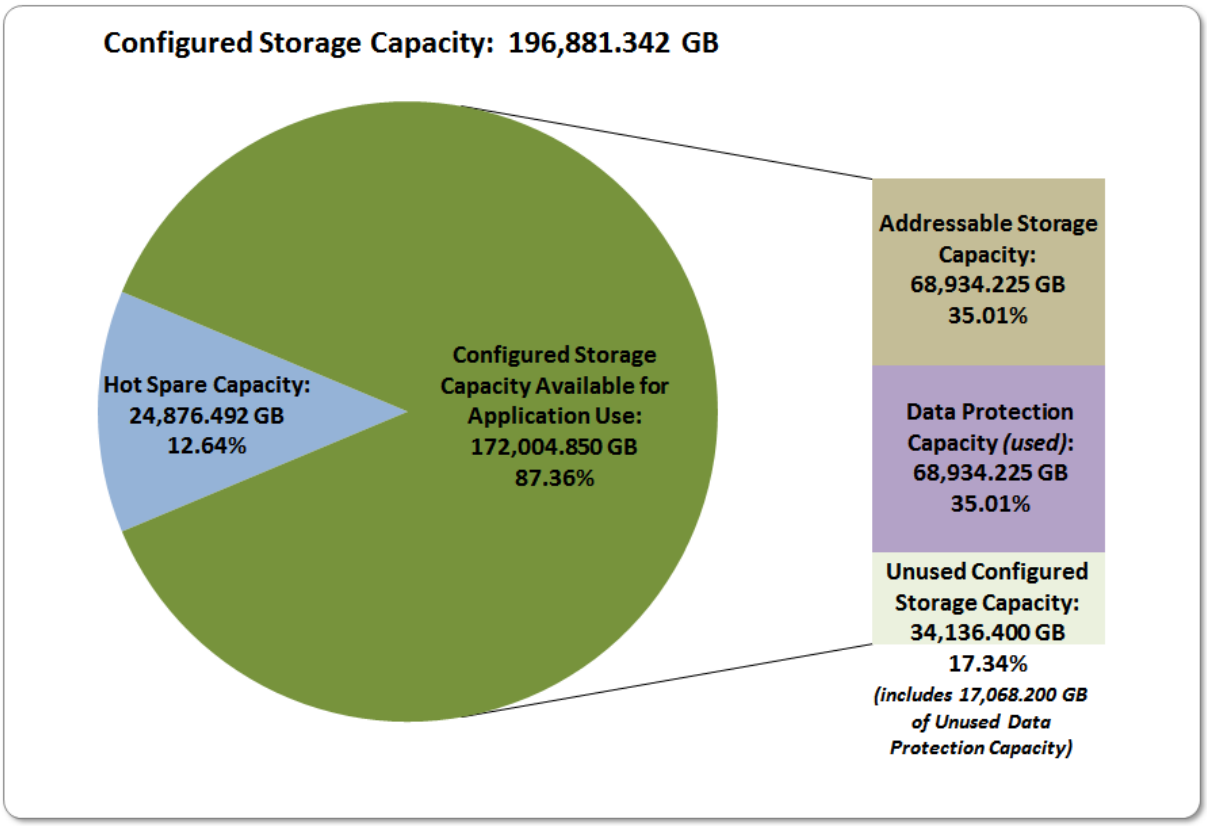
Storage Hierarchy Component	Units	Capacity
Total ASU Capacity	Gigabytes (GB)	68,934.225
Addressable Storage Capacity	Gigabytes (GB)	68,934.225
Configured Storage Capacity	Gigabytes (GB)	196,881.342
Physical Storage Capacity	Gigabytes (GB)	208,045.998
Data Protection *(Mirroring)*	Gigabytes (GB)	86,002.425
Required Storage *(sparing)*	Gigabytes (GB)	24,876.492
Global Storage Overhead	Gigabytes (GB)	436.753
Total Unused Storage	Gigabytes (GB)	44,864.303

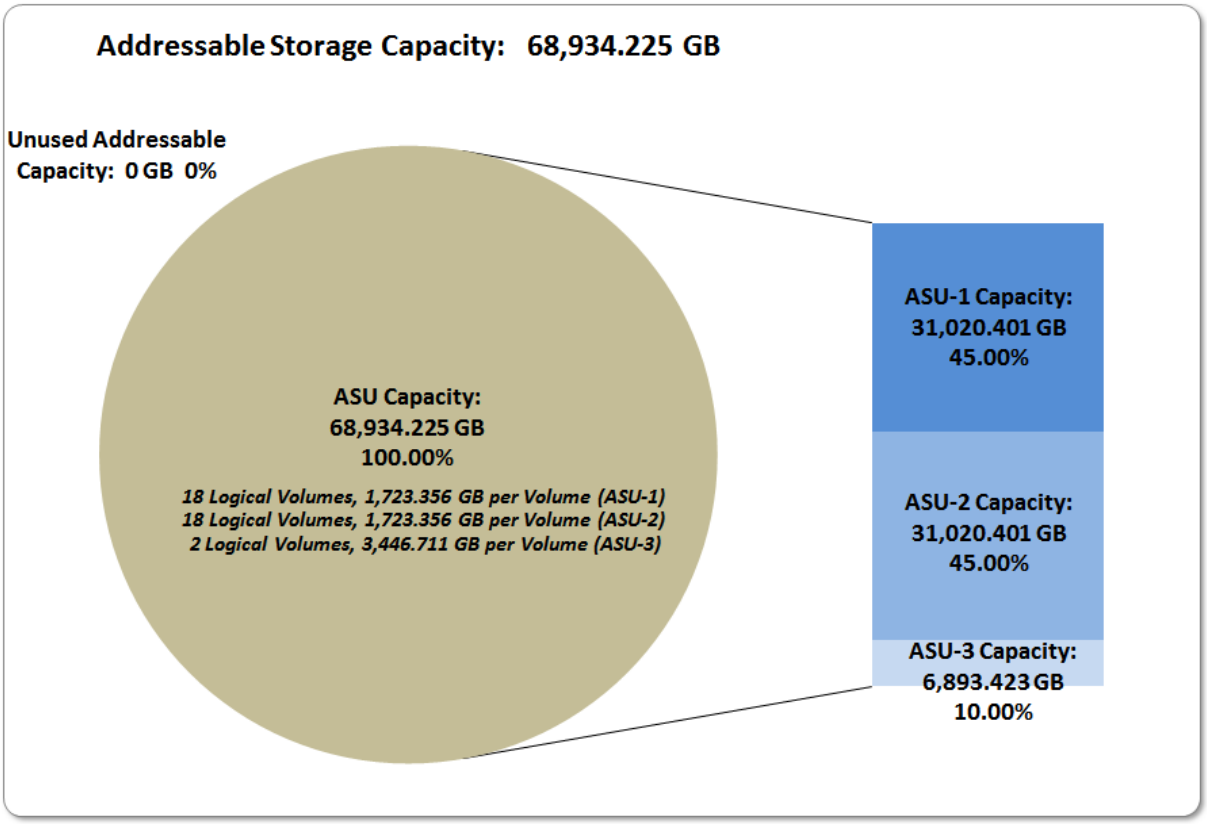
SPC-1 Storage Hierarchy Ratios

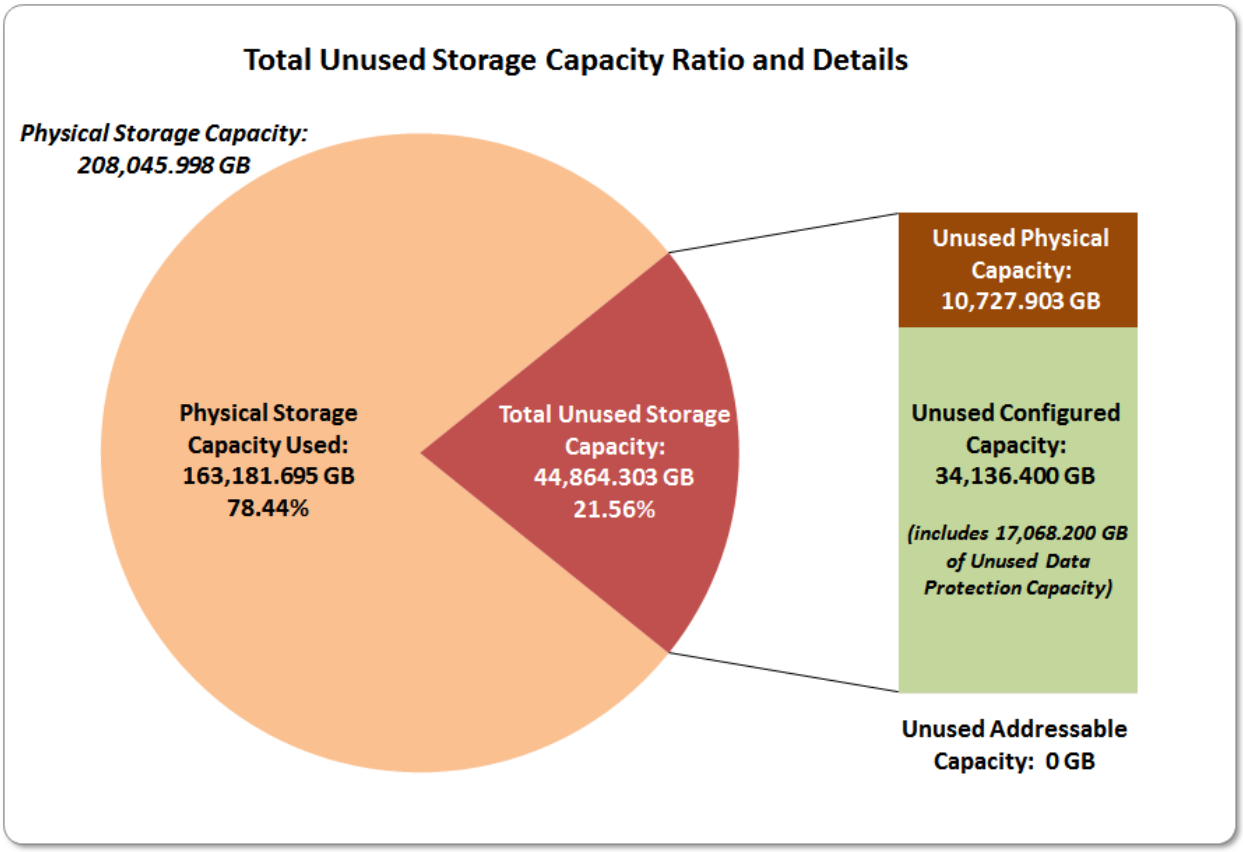
	Addressable Storage Capacity	Configured Storage Capacity	Physical Storage Capacity
Total ASU Capacity	100.00%	35.01%	33.13%
Required for Data Protection *(Mirroring)*		43.68%	41.34%
Addressable Storage Capacity		35.01%	33.13%
Required Storage *(sparing)*		12.64%	11.96%
Configured Storage Capacity			94.63%
Global Storage Overhead			0.21%
Unused Storage:			
Addressable	0.00%		
Configured		17.34%	
Physical			5.16%

SPC-1 Storage Capacity Charts




Configured Storage Capacity: 196,881.342 GB
Hot Spare Capacity:
24,876.492 GB
12.64%
Configured Storage Capacity Available for Application Use:
172,004.850 GB
87.36%
Addressable Storage Capacity:
68,934.225 GB
35.01%
Data Protection Capacity (used):
68,934.225 GB
35.01%
Unused Configured Storage Capacity:
34,136.400 GB
17.34%
(includes 17,068.200 GB of Unused Data Protection Capacity)


Addressable Storage Capacity: 68,934.225 GB
Unused Addressable Capacity: 0 GB 0%
ASU Capacity:
68,934.225 GB
100.00%
18 Logical Volumes, 1,723.356 GB per Volume (ASU-1)
18 Logical Volumes, 1,723.356 GB per Volume (ASU-2)
2 Logical Volumes, 3,446.711 GB per Volume (ASU-3)
ASU-1 Capacity:
31,020.401 GB
45.00%
ASU-2 Capacity:
31,020.401 GB
45.00%
ASU-3 Capacity:
6,893.423 GB
10.00%



Storage Capacity Utilization

Clause 9.4.3.6.2

The FDR will include a table illustrating the storage capacity utilization values defined for Application Utilization (Clause 2.8.1), Protected Application Utilization (Clause 2.8.2), and Unused Storage Ratio (Clause 2.8.3).

Clause 2.8.1

Application Utilization is defined as Total ASU Capacity divided by Physical Storage Capacity.

Clause 2.8.2

Protected Application Utilization is defined as (Total ASU Capacity plus total Data Protection Capacity minus unused Data Protection Capacity) divided by Physical Storage Capacity.

Clause 2.8.3

Unused Storage Ratio is defined as Total Unused Capacity divided by Physical Storage Capacity and may not exceed 45%.

SPC-1 Storage Capacity Utilization	
Application Utilization	33.13%
Protected Application Utilization	66.27%
Unused Storage Ratio	21.56%

Logical Volume Capacity and ASU Mapping

Clause 9.4.3.6.3

A table illustrating the capacity of each ASU and the mapping of Logical Volumes to ASUs shall be provided in the FDR. … Logical Volumes shall be sequenced in the table from top to bottom per its position in the contiguous address space of each ASU. The capacity of each Logical Volume shall be stated. … In conjunction with this table, the Test Sponsor shall provide a complete description of the type of data protection (see Clause 2.4.5) used on each Logical Volume.

Logical Volume Capacity and Mapping
ASU-1 (31,020.401 GB)
18 Logical Volumes 1,723.356 GB per Logical Volume (1,723.356 GB used per Logical Volume)
ASU-2 (31,020.401 GB)
18 Logical Volumes 1,723.356 GB per Logical Volume (1,723.356 GB used per Logical Volume)
ASU-3 (6,893.423 GB)
2 Logical Volumes 3,446.711 GB per Logical Volume (3,446.711 GB used per Logical Volume)

The Data Protection Level used for all Logical Volumes was **Protected 1** using ***Mirroring*** as described on page 11. See "ASU Configuration" in the **IOPS Test Results File** for more detailed configuration information.

SPC-1 BENCHMARK EXECUTION RESULTS

This portion of the Full Disclosure Report documents the results of the various SPC-1 Tests, Test Phases, and Test Runs. An SPC-1 glossary on page 63 contains definitions of terms specific to the SPC-1 Tests, Test Phases, and Test Runs.

Clause 5.4.3

The Tests must be executed in the following sequence: Primary Metrics, Repeatability, and Data Persistence. That required sequence must be uninterrupted from the start of Primary Metrics to the completion of Persistence Test Run 1. Uninterrupted means the Benchmark Configuration shall not be power cycled, restarted, disturbed, altered, or adjusted during the above measurement sequence. If the required sequence is interrupted other than for the Host System/TSC power cycle between the two Persistence Test Runs, the measurement is invalid.

SPC-1 Tests, Test Phases, and Test Runs

The SPC-1 benchmark consists of the following Tests, Test Phases, and Test Runs:

- **Primary Metrics Test**
  - Sustainability Test Phase and Test Run
  - IOPS Test Phase and Test Run
  - Response Time Ramp Test Phase
    - 95% of IOPS Test Run
    - 90% of IOPS Test Run
    - 80% of IOPS Test Run
    - 50% of IOPS Test Run
    - 10% of IOPS Test Run (LRT)
- **Repeatability Test**
  - Repeatability Test Phase 1
    - 10% of IOPS Test Run (LRT)
    - IOPS Test Run
  - Repeatability Test Phase 2
    - 10% of IOPS Test Run (LRT)
    - IOPS Test Run
- **Data Persistence Test**
  - Data Persistence Test Run 1
  - Data Persistence Test Run 2

Each Test is an atomic unit that must be executed from start to finish before any other Test, Test Phase, or Test Run may be executed.

The results from each Test, Test Phase, and Test Run are listed below along with a more detailed explanation of each component.

"Ramp-Up" Test Runs

Clause 5.3.13

In order to warm-up caches or perform the initial ASU data migration in a multi-tier configuration, a Test Sponsor may perform a series of "Ramp-Up" Test Runs as a substitute for an initial, gradual Ramp-Up.

Clause 5.3.13.3

The "Ramp-Up" Test Runs will immediately precede the Primary Metrics Test as part of the uninterrupted SPC-1 measurement sequence.

Clause 9.4.3.7.1

If a series of "Ramp-Up" Test Runs were included in the SPC-1 measurement sequence, the FDR shall report the duration (ramp-up and measurement interval), BSU level, SPC-1 IOPS and average response time for each "Ramp-Up" Test Run in an appropriate table.

There were no "Ramp-Up" Test Runs executed.

Primary Metrics Test – Sustainability Test Phase

Clause 5.4.4.1.1

The Sustainability Test Phase has exactly one Test Run and shall demonstrate the maximum sustainable I/O Request Throughput within at least a continuous eight (8) hour Measurement Interval. This Test Phase also serves to insure that the TSC has reached Steady State prior to reporting the final maximum I/O Request Throughput result (SPC-1 IOPS™).

Clause 5.4.4.1.2

The computed I/O Request Throughput of the Sustainability Test must be within 5% of the reported SPC-1 IOPS™ result.

Clause 5.4.4.1.4

The Average Response Time, as defined in Clause 5.1.1, will be computed and reported for the Sustainability Test Run and cannot exceed 30 milliseconds. If the Average Response time exceeds that 30-milliseconds constraint, the measurement is invalid.

Clause 9.4.3.7.2

For the Sustainability Test Phase the FDR shall contain:

1. *A Data Rate Distribution graph and data table.*
2. *I/O Request Throughput Distribution graph and data table.*
3. *A Response Time Frequency Distribution graph and table.*
4. *An Average Response Time Distribution graph and table.*
5. *The human readable Test Run Results File produced by the Workload Generator (may be included in an appendix).*
6. *A listing or screen image of all input parameters supplied to the Workload Generator (may be included in an appendix).*
7. *The Measured Intensity Multiplier for each I/O stream.*
8. *The variability of the Measured Intensity Multiplier, as defined in Clause 5.3.13.3.*

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 95.

Sustainability Test Results File

A link to the test results file generated from the Sustainability Test Run is listed below.

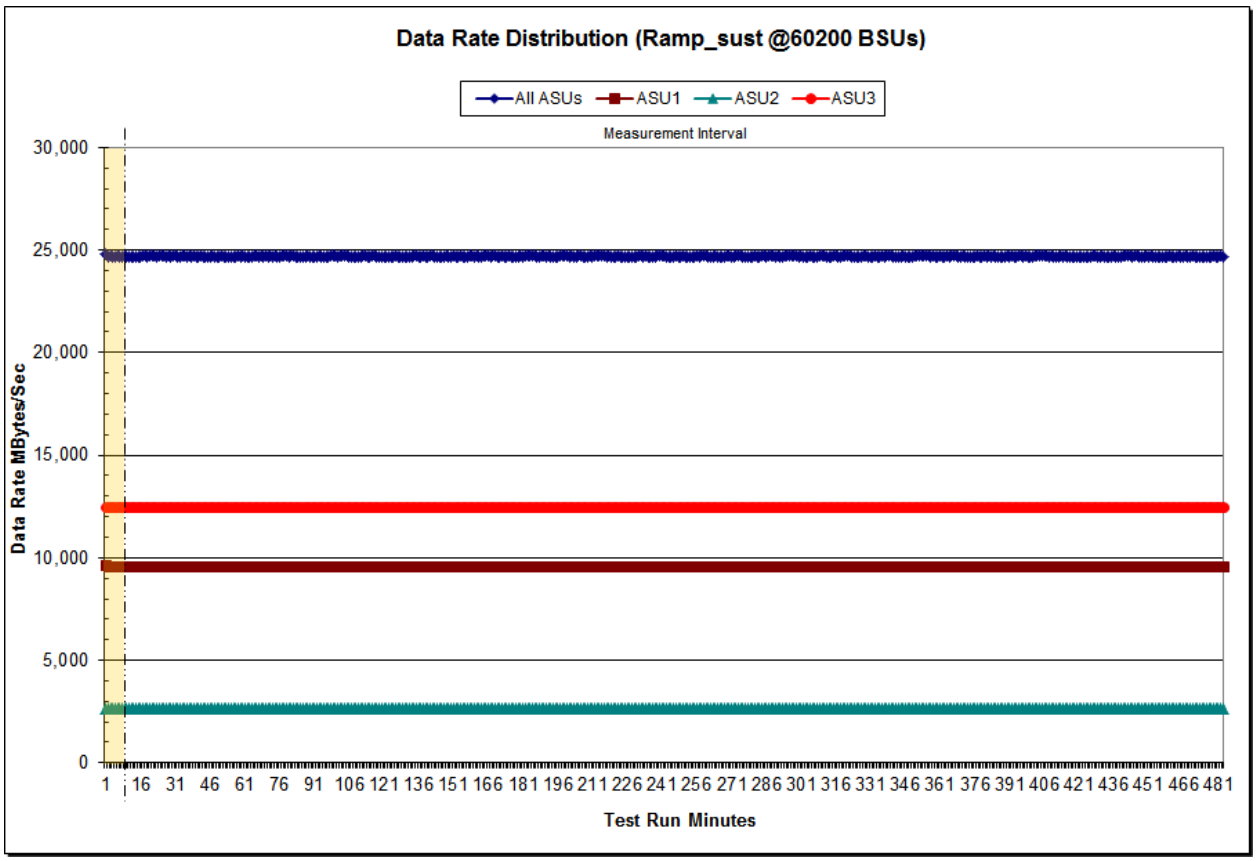
Sustainability Test Results File

Sustainability – Data Rate Distribution Data (MB/second)

The Sustainability Data Rate table of data is not embedded in this document due to its size. The table is available via the following URL:

Sustainability Data Rate Table

Sustainability – Data Rate Distribution Graph

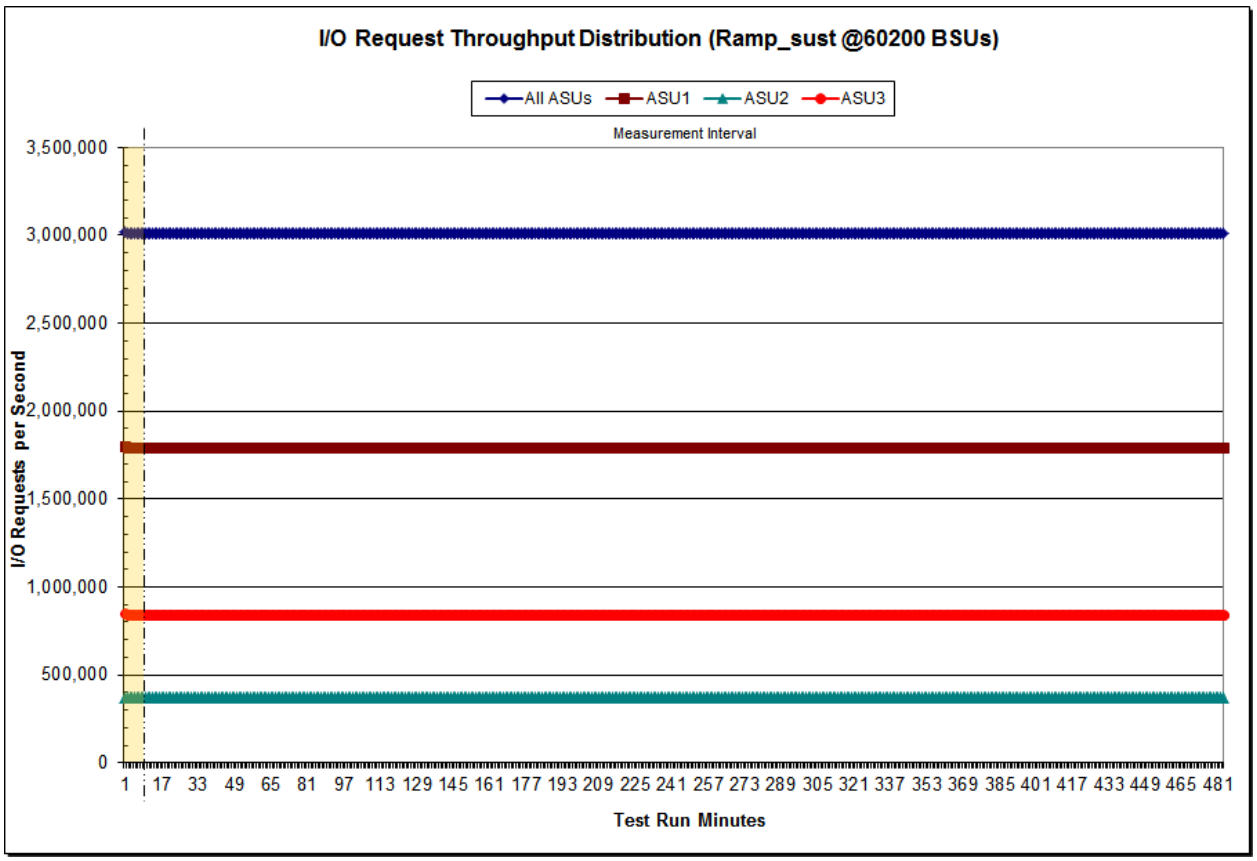


Sustainability – I/O Request Throughput Distribution Data

The Sustainability I/O Request Throughput table of data is not embedded in this document due to its size. The table is available via the following URL:

Sustainability I/O Request Throughput Table

Sustainability – I/O Request Throughput Distribution Graph

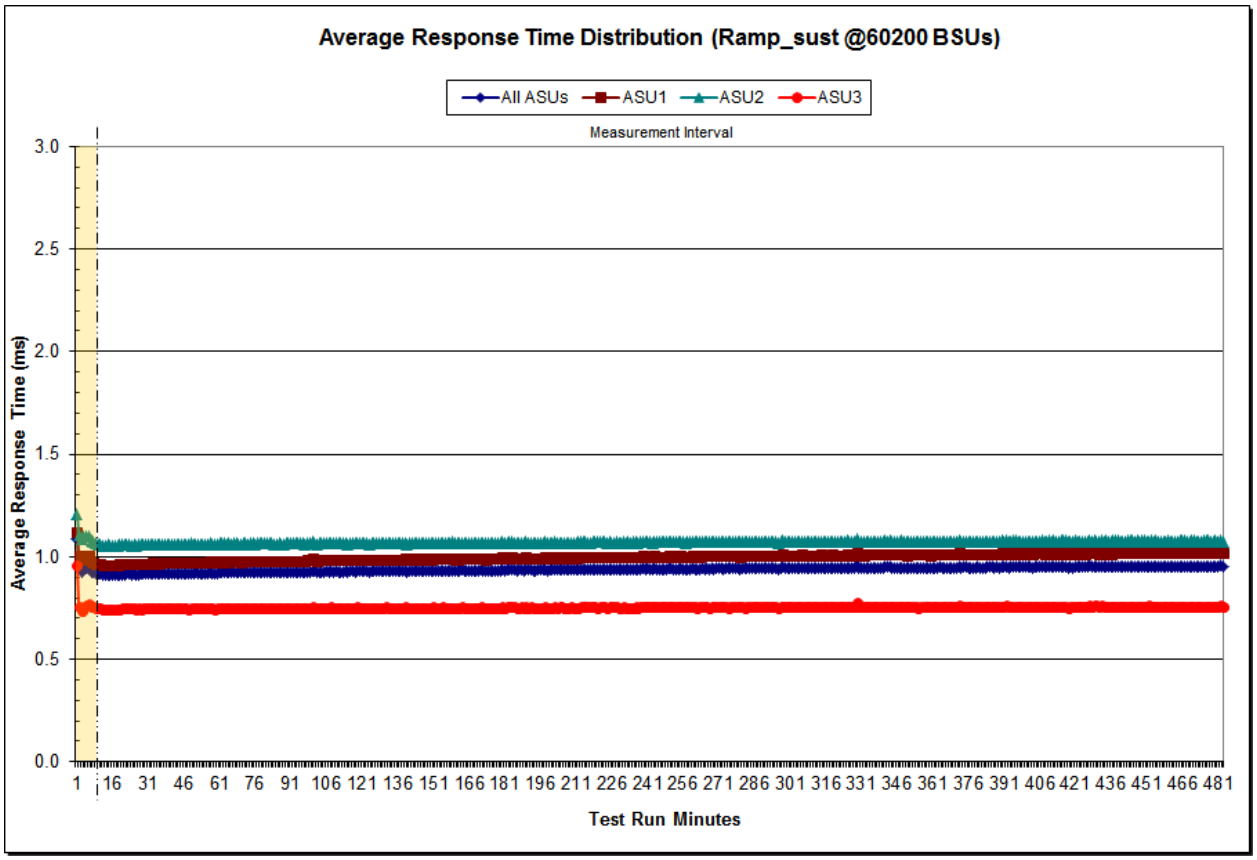


Sustainability – Average Response Time (ms) Distribution Data

The Sustainability Average Response Time table of data is not embedded in this document due to its size. The table is available via the following URL:

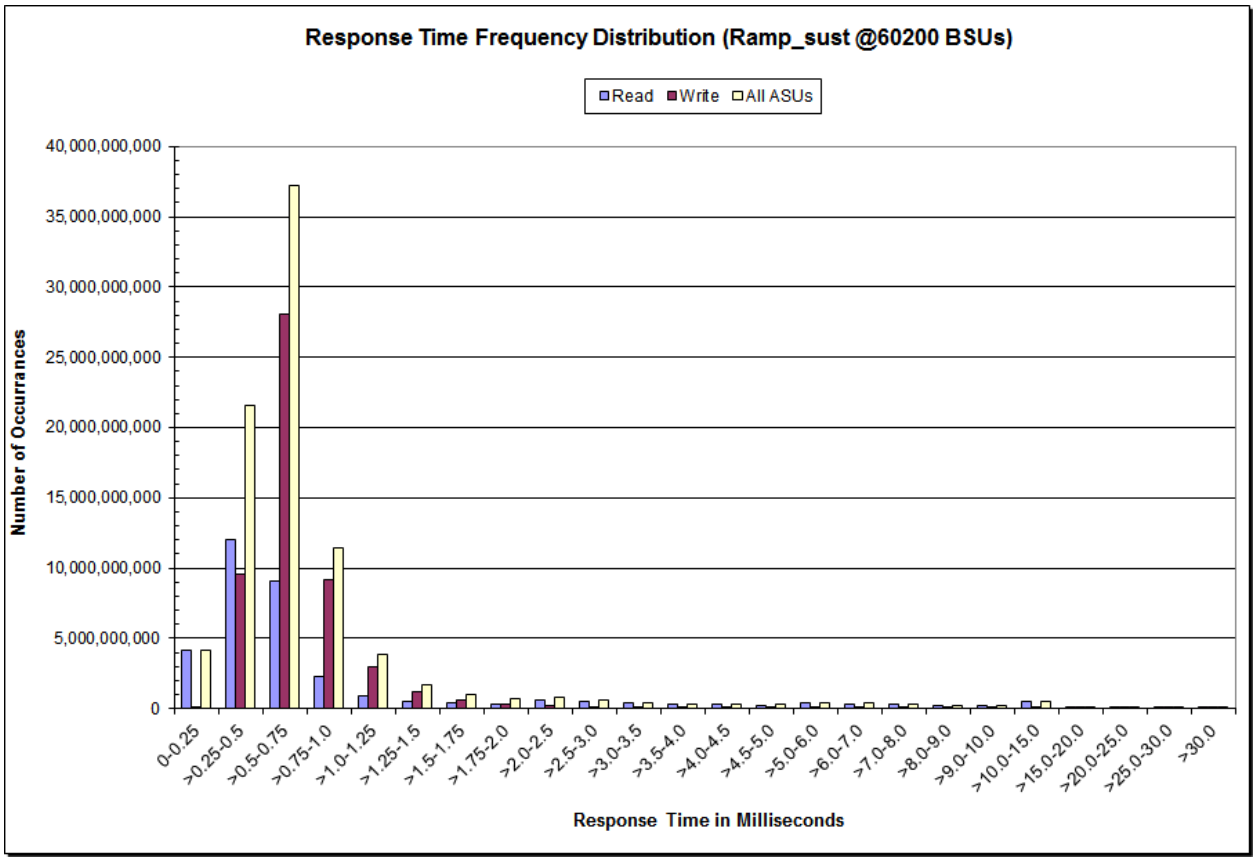
Sustainability Average Response Time Table

Sustainability – Average Response Time (ms) Distribution Graph



Sustainability – Response Time Frequency Distribution Data

Sustainability – Response Time Frequency Distribution Graph



Sustainability – Measured Intensity Multiplier and Coefficient of Variation

Clause 3.4.3

IM – Intensity Multiplier: *The ratio of I/Os for each I/O stream relative to the total I/Os for all I/O streams (ASU1-1 – ASU3-1) as required by the benchmark specification.*

Clauses 5.1.10 and 5.3.15.2

MIM – Measured Intensity Multiplier: *The Measured Intensity Multiplier represents the ratio of measured I/Os for each I/O stream relative to the total I/Os measured for all I/O streams (ASU1-1 – ASU3-1). This value may differ from the corresponding Expected Intensity Multiplier by no more than 5%.*

Clause 5.3.15.3

COV – Coefficient of Variation: *This measure of variation for the Measured Intensity Multiplier cannot exceed 0.2.*

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
COV	0.000	0.000	0.000	0..000	0.001	0.000	0.000	0.000

Primary Metrics Test – IOPS Test Phase

Clause 5.4.4.2

The IOPS Test Phase consists of one Test Run at the 100% load point with a Measurement Interval of ten (10) minutes. The IOPS Test Phase immediately follows the Sustainability Test Phase without any interruption or manual intervention.

The IOPS Test Run generates the SPC-1 IOPS™ primary metric, which is computed as the I/O Request Throughput for the Measurement Interval of the IOPS Test Run.

The Average Response Time is computed for the IOPS Test Run and cannot exceed 30 milliseconds. If the Average Response Time exceeds the 30 millisecond constraint, the measurement is invalid.

Clause 9.4.3.7.3

For the IOPS Test Phase the FDR shall contain:

1. *I/O Request Throughput Distribution (data and graph).*
2. *A Response Time Frequency Distribution.*
3. *An Average Response Time Distribution.*
4. *The human readable Test Run Results File produced by the Workload Generator.*
5. *A listing or screen image of all input parameters supplied to the Workload Generator.*
6. *The total number of I/O Requests completed in the Measurement Interval as well as the number of I/O Requests with a Response Time less than or equal to 30 milliseconds and the number of I/O Requests with a Response Time greater than 30 milliseconds.*

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 95.

IOPS Test Results File

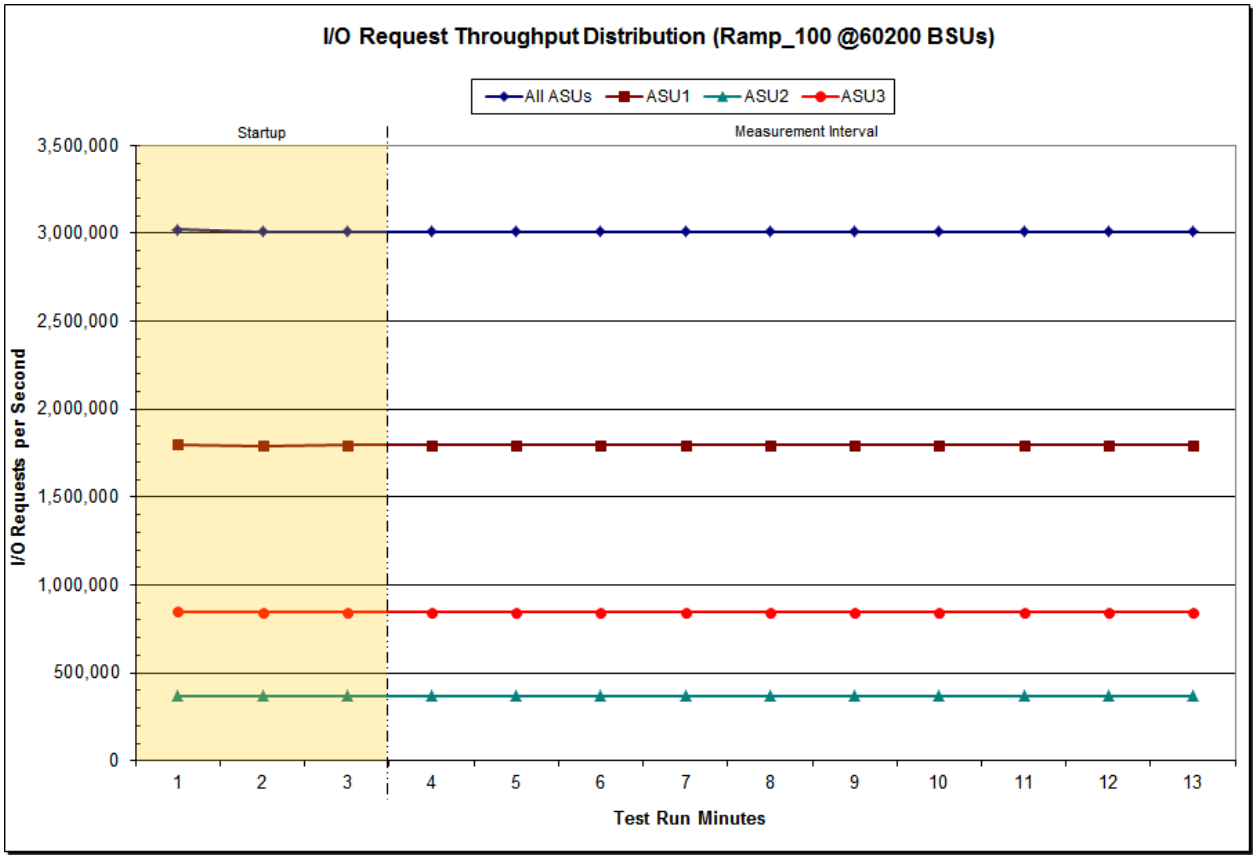
A link to the test results file generated from the IOPS Test Run is listed below.

IOPS Test Results File

IOPS Test Run – I/O Request Throughput Distribution Data

60,200 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	9:06:19	9:09:19	0-2	0:03:00
Measurement Interval	9:09:19	9:19:22	3-12	0:10:03
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	3,019,989.20	1,800,137.65	371,366.90	848,484.65
1	3,009,769.78	1,793,736.82	370,178.05	845,854.92
2	3,009,930.82	1,793,795.12	370,192.90	845,942.80
3	3,010,165.53	1,794,032.48	370,262.05	845,871.00
4	3,010,074.13	1,794,061.33	370,314.37	845,698.43
5	3,010,013.53	1,794,093.63	370,128.87	845,791.03
6	3,009,985.25	1,793,995.27	370,231.27	845,758.72
7	3,009,889.22	1,793,890.40	370,200.77	845,798.05
8	3,009,971.47	1,794,023.97	370,322.55	845,624.95
9	3,009,796.63	1,793,864.53	370,185.55	845,746.55
10	3,010,175.92	1,794,202.87	370,251.43	845,721.62
11	3,010,139.05	1,794,143.43	370,277.15	845,718.47
12	3,009,862.93	1,793,892.58	370,153.45	845,816.90
Average	***3,010,007.37***	***1,794,020.05***	***370,232.75***	***845,754.57***

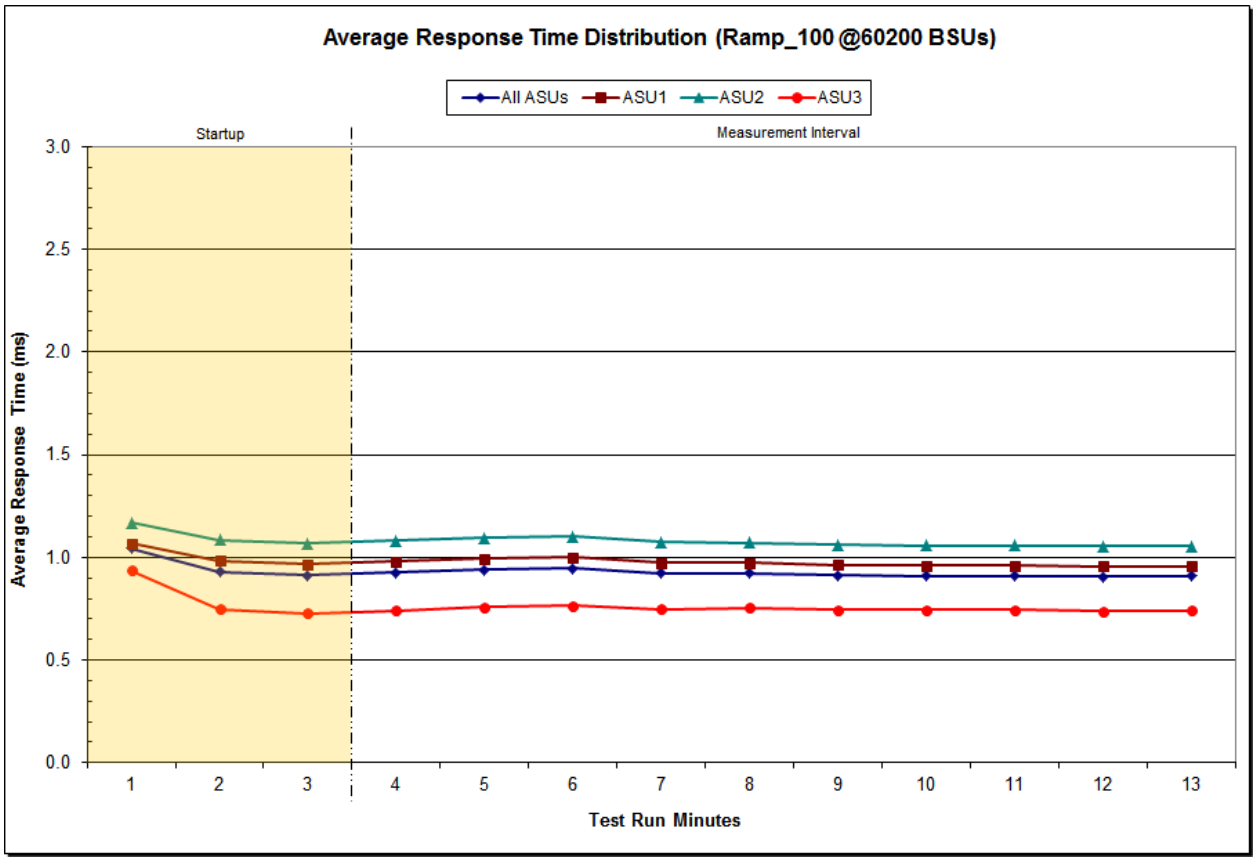
IOPS Test Run – I/O Request Throughput Distribution Graph



IOPS Test Run – Average Response Time (ms) Distribution Data

60,200 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	9:06:19	9:09:19	0-2	0:03:00
Measurement Interval	9:09:19	9:19:22	3-12	0:10:03
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	1.04	1.07	1.17	0.94
1	0.93	0.98	1.09	0.75
2	0.91	0.97	1.07	0.73
3	0.93	0.98	1.08	0.74
4	0.94	0.99	1.10	0.76
5	0.95	1.00	1.10	0.77
6	0.92	0.97	1.07	0.75
7	0.92	0.97	1.07	0.75
8	0.91	0.96	1.06	0.74
9	0.91	0.96	1.06	0.74
10	0.91	0.96	1.06	0.74
11	0.91	0.96	1.05	0.74
12	0.91	0.96	1.05	0.74
Average	***0.92***	***0.97***	***1.07***	***0.75***

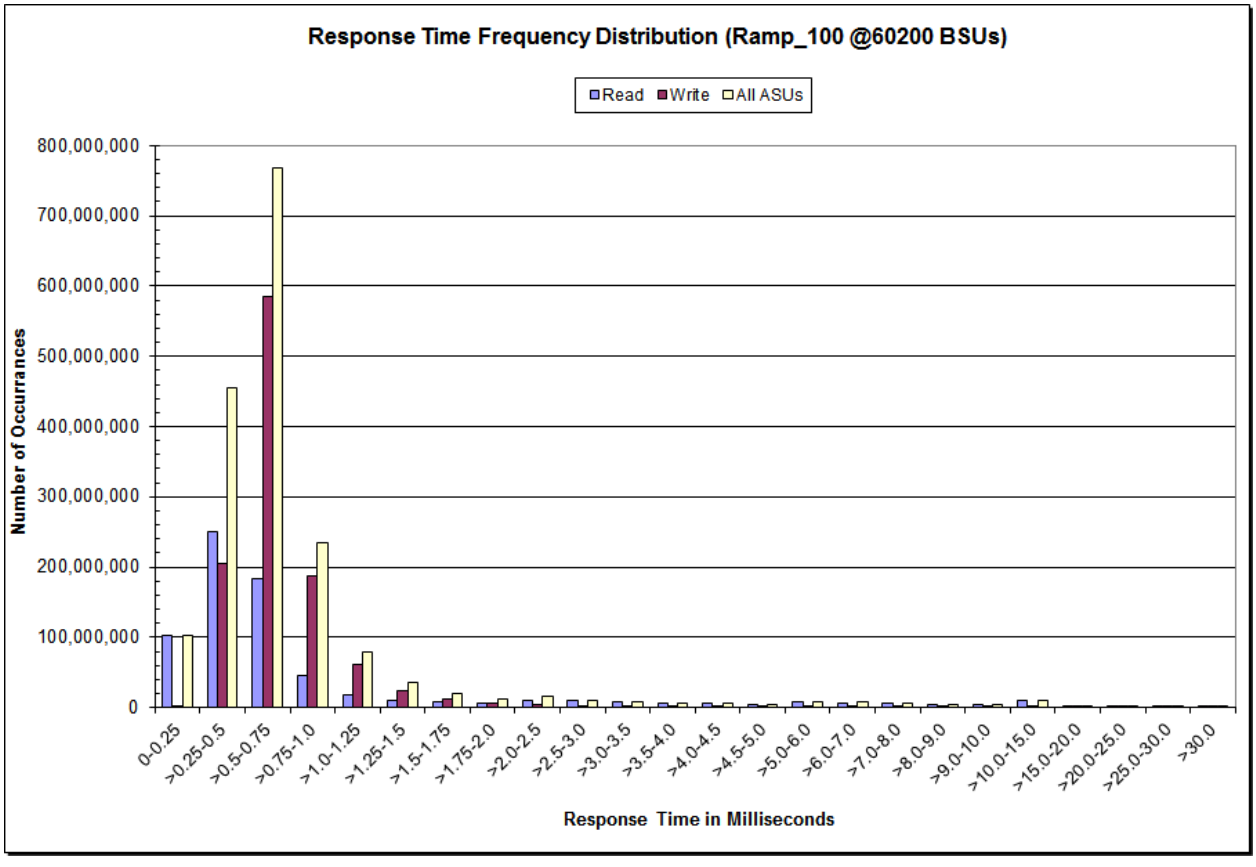
IOPS Test Run – Average Response Time (ms) Distribution Graph



IOPS Test Run –Response Time Frequency Distribution Data

Response Time (ms)	0-0.25	>0.25-0.5	>0.5-0.75	>0.75-1.0	>1.0-1.25	>1.25-1.5	>1.5-1.75	>1.75-2.0
Read	103,466,494	249,814,377	182,606,964	45,456,415	18,041,934	10,855,932	8,070,495	6,622,873
Write	1,484	205,391,857	584,710,216	188,337,143	60,930,177	24,199,191	12,196,363	6,447,445
All ASUs	103,467,978	455,206,234	767,317,180	233,793,558	78,972,111	35,055,123	20,266,858	13,070,318
ASU1	97,501,966	309,671,504	401,191,385	114,290,062	39,817,043	18,882,976	11,673,893	8,077,201
ASU2	5,965,293	57,034,726	95,064,329	27,992,587	9,545,561	4,452,139	2,721,124	1,874,886
ASU3	719	88,500,004	271,061,466	91,510,909	29,609,507	11,720,008	5,871,841	3,118,231
Response Time (ms)	**>2.0-2.5**	**>2.5-3.0**	**>3.0-3.5**	**>3.5-4.0**	**>4.0-4.5**	**>4.5-5.0**	**>5.0-6.0**	**>6.0-7.0**
Read	11,011,379	9,449,851	7,364,920	5,954,148	5,354,182	4,898,725	8,543,863	7,061,656
Write	5,023,653	1,642,048	732,922	474,068	354,716	278,068	432,154	348,976
All ASUs	16,035,032	11,091,899	8,097,842	6,428,216	5,708,898	5,176,793	8,976,017	7,410,632
ASU1	11,036,139	8,333,371	6,262,877	5,038,839	4,503,969	4,097,349	7,114,082	5,866,568
ASU2	2,557,631	1,953,892	1,469,462	1,149,051	1,021,702	932,809	1,627,369	1,348,057
ASU3	2,441,262	804,636	365,503	240,326	183,227	146,635	234,566	196,007
Response Time (ms)	**>7.0-8.0**	**>8.0-9.0**	**>9.0-10.0**	**>10.0-15.0**	**>15.0-20.0**	**>20.0-25.0**	**>25.0-30.0**	**>30.0**
Read	5,778,618	4,697,374	3,775,481	9,898,617	2,697,867	698,857	162,012	46,984
Write	298,323	271,533	237,065	900,369	289,709	114,987	36,168	23,043
All ASUs	6,076,941	4,968,907	4,012,546	10,798,986	2,987,576	813,844	198,180	70,027
ASU1	4,793,585	3,900,810	3,133,193	8,260,268	2,211,602	573,970	133,827	43,785
ASU2	1,111,209	908,063	734,593	1,956,834	534,989	139,893	32,788	10,291
ASU3	172,147	160,034	144,760	581,884	240,985	99,981	31,565	15,951

IOPS Test Run –Response Time Frequency Distribution Graph



IOPS Test Run – I/O Request Information

I/O Requests Completed in the Measurement Interval	I/O Requests Completed with Response Time = or < 30 ms	I/O Requests Completed with Response Time > 30 ms
1,806,001,696	1,805,931,669	70,027

IOPS Test Run – Measured Intensity Multiplier and Coefficient of Variation

Clause 3.4.3

IM – Intensity Multiplier: *The ratio of I/Os for each I/O stream relative to the total I/Os for all I/O streams (ASU1-1 – ASU3-1) as required by the benchmark specification.*

Clauses 5.1.10 and 5.3.15.2

MIM – Measured Intensity Multiplier: *The Measured Intensity Multiplier represents the ratio of measured I/Os for each I/O stream relative to the total I/Os measured for all I/O streams (ASU1-1 – ASU3-1). This value may differ from the corresponding Expected Intensity Multiplier by no more than 5%.*

Clause 5.3.15.3

COV – Coefficient of Variation: *This measure of variation for the Measured Intensity Multiplier cannot exceed 0.2.*

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
COV	0.001	0.000	0.000	0.000	0.001	0.000	0.001	0.000

Primary Metrics Test – Response Time Ramp Test Phase

Clause 5.4.4.3

The Response Time Ramp Test Phase consists of five Test Runs, one each at 95%, 90%, 80%, 50%, and 10% of the load point (100%) used to generate the SPC-1 IOPS™ primary metric. Each of the five Test Runs has a Measurement Interval of ten (10) minutes. The Response Time Ramp Test Phase immediately follows the IOPS Test Phase without any interruption or manual intervention.

The five Response Time Ramp Test Runs, in conjunction with the IOPS Test Run (100%), demonstrate the relationship between Average Response Time and I/O Request Throughput for the Tested Storage Configuration (TSC) as illustrated in the response time/throughput curve on page 15.

In addition, the Average Response Time measured during the 10% Test Run is the value for the SPC-1 LRT™ metric. That value represents the Average Response Time of a lightly loaded TSC.

Clause 9.4.3.7.4

The following content shall appear in the FDR for the Response Time Ramp Phase:

1. *A Response Time Ramp Distribution.*
2. *The human readable Test Run Results File produced by the Workload Generator for each Test Run within the Response Time Ramp Test Phase.*
3. *For the 10% Load Level Test Run (SPC-1 LRT™ metric) an Average Response Time Distribution.*
4. *A listing or screen image of all input parameters supplied to the Workload Generator.*

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 95.

Response Time Ramp Test Results File

A link to each test result file generated from each Response Time Ramp Test Run list listed below.

95% Load Level

90% Load Level

80% Load Level

50% Load Level

10% Load Level

Response Time Ramp Distribution (IOPS) Data

The five Test Runs that comprise the Response Time Ramp Phase are executed at 95%, 90%, 80%, 50%, and 10% of the Business Scaling Unit (BSU) load level used to produce the SPC-1 IOPS™ primary metric. The 100% BSU load level is included in the following Response Time Ramp data table and graph for completeness.

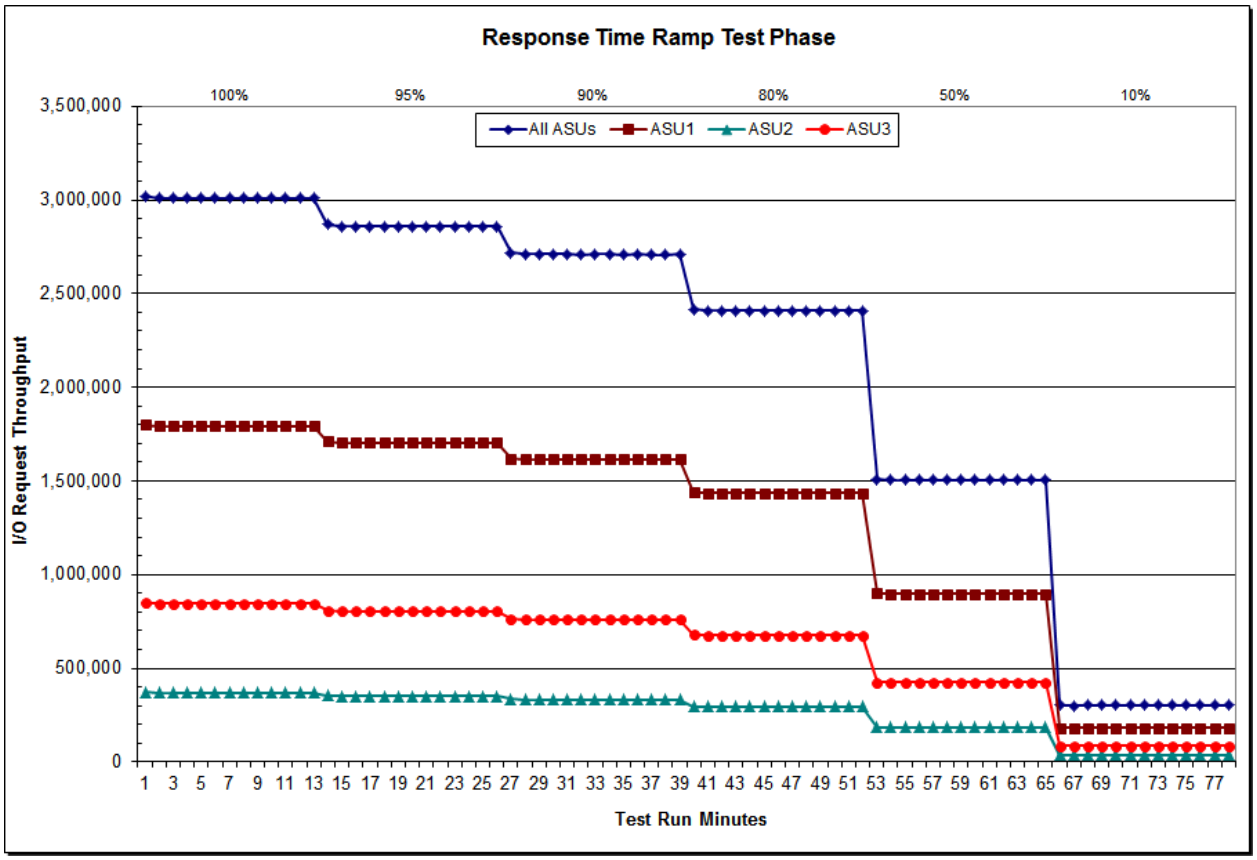
100% Load Level: 60,200- BSUs	Start	Stop	Interval	Duration	95% Load Level: 57,190 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	9:06:19	9:09:19	0-3	0:03:00	Start-Up/Ramp-Up	10:19:38	10:22:38	0-3	0:03:00
Measurement Interval	9:09:19	9:19:22	3-12	0:10:03	Measurement Interval	10:22:38	10:32:40	3-12	0:10:02
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3	*(60 second intervals)*	All ASUs	ASU-1	ASU-2	ASU-3
0	3,019,989.20	1,800,137.65	371,366.90	848,484.65	0	2,869,289.00	1,710,024.13	353,063.58	806,201.28
1	3,009,769.78	1,793,736.82	370,178.05	845,854.92	1	2,859,987.85	1,704,556.33	351,749.27	803,682.25
2	3,009,930.82	1,793,795.12	370,192.90	845,942.80	2	2,859,470.25	1,704,155.53	351,764.12	803,550.60
3	3,010,165.53	1,794,032.48	370,262.05	845,871.00	3	2,859,672.52	1,704,341.37	351,820.00	803,511.15
4	3,010,074.13	1,794,061.33	370,314.37	845,698.43	4	2,859,620.92	1,704,086.08	351,828.98	803,705.85
5	3,010,013.53	1,794,093.63	370,128.87	845,791.03	5	2,859,590.07	1,704,452.15	351,701.30	803,436.62
6	3,009,985.25	1,793,995.27	370,231.27	845,758.72	6	2,859,355.87	1,704,076.52	351,606.00	803,673.35
7	3,009,889.22	1,793,890.40	370,200.77	845,798.05	7	2,859,386.80	1,704,159.97	351,739.00	803,487.83
8	3,009,971.47	1,794,023.97	370,322.55	845,624.95	8	2,859,294.98	1,704,050.35	351,735.32	803,509.32
9	3,009,796.63	1,793,864.53	370,185.55	845,746.55	9	2,859,369.65	1,704,330.52	351,651.75	803,387.38
10	3,010,175.92	1,794,202.87	370,251.43	845,721.62	10	2,859,584.55	1,704,183.65	351,840.57	803,560.33
11	3,010,139.05	1,794,143.43	370,277.15	845,718.47	11	2,859,345.33	1,704,122.57	351,708.38	803,514.38
12	3,009,862.93	1,793,892.58	370,153.45	845,816.90	12	2,859,567.97	1,704,345.72	351,789.23	803,433.02
Average	***3,010,007.37***	***1,794,020.05***	***370,232.75***	***845,754.57***	*Average*	***2,859,478.87***	***1,704,214.89***	***351,742.05***	***803,521.92***

90% Load Level: 54,180 BSUs	Start	Stop	Interval	Duration	80% Load Level: 48,160 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	11:31:20	11:34:20	0-3	0:03:00	Start-Up/Ramp-Up	12:38:05	12:41:05	0-3	0:03:00
Measurement Interval	11:34:20	11:44:22	3-12	0:10:02	Measurement Interval	12:41:05	12:51:07	3-12	0:10:02
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3	*(60 second intervals)*	All ASUs	ASU-1	ASU-2	ASU-3
0	2,718,030.98	1,620,141.28	334,337.87	763,551.83	0	2,415,945.43	1,440,037.27	297,129.62	678,778.55
1	2,708,992.33	1,614,706.62	333,089.80	761,195.92	1	2,407,926.37	1,435,211.32	296,112.97	676,602.08
2	2,708,985.23	1,614,644.32	333,248.55	761,092.37	2	2,407,745.62	1,435,116.63	296,078.78	676,550.20
3	2,709,060.40	1,614,711.10	333,133.33	761,215.97	3	2,407,774.98	1,435,170.55	296,108.48	676,495.95
4	2,709,103.78	1,614,855.05	333,137.93	761,110.80	4	2,407,904.00	1,435,007.35	296,222.10	676,674.55
5	2,708,813.27	1,614,369.13	333,107.95	761,336.18	5	2,408,001.98	1,435,097.03	296,346.38	676,558.57
6	2,709,020.35	1,614,569.85	333,295.63	761,154.87	6	2,407,868.62	1,434,778.05	296,258.95	676,831.62
7	2,709,366.55	1,615,028.37	333,082.47	761,255.72	7	2,407,809.95	1,435,031.15	296,219.68	676,559.12
8	2,708,669.80	1,614,318.87	333,198.57	761,152.37	8	2,407,673.58	1,435,068.37	295,941.45	676,663.77
9	2,709,047.28	1,614,678.00	333,200.75	761,168.53	9	2,407,878.28	1,435,182.05	296,069.33	676,626.90
10	2,708,566.92	1,614,068.07	333,239.23	761,259.62	10	2,407,813.08	1,435,047.10	296,142.67	676,623.32
11	2,708,902.02	1,614,484.38	333,179.62	761,238.02	11	2,407,930.63	1,435,049.17	296,202.52	676,678.95
12	2,709,151.82	1,614,740.43	333,247.33	761,164.05	12	2,408,016.25	1,435,140.77	296,240.52	676,634.97
Average	***2,708,970.22***	***1,614,582.33***	***333,182.28***	***761,205.61***	*Average*	***2,407,867.14***	***1,435,057.16***	***296,175.21***	***676,634.77***

Response Time Ramp Distribution (IOPS) Data *(continued)*

50% Load Level: 30,100 BSUs	Start	Stop	Interval	Duration	10% Load Level: 6,020 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	13:33:29	13:36:29	0-3	0:03:00	Start-Up/Ramp-Up	14:14:26	14:17:26	0-3	0:03:00
Measurement Interval	13:36:29	13:46:31	3-12	0:10:02	Measurement Interval	14:17:26	14:27:28	3-12	0:10:02
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3	*(60 second intervals)*	All ASUs	ASU-1	ASU-2	ASU-3
0	1,509,547.87	899,673.82	185,632.73	424,241.32	0	301,623.27	179,741.58	37,142.02	84,739.67
1	1,504,984.53	896,970.05	185,036.08	422,978.40	1	300,890.85	179,343.07	36,964.10	84,583.68
2	1,505,239.98	897,114.10	185,093.07	423,032.82	2	300,960.75	179,424.48	36,979.30	84,556.97
3	1,505,188.97	897,039.40	185,197.37	422,952.20	3	301,006.33	179,401.28	37,006.50	84,598.55
4	1,504,783.47	896,816.78	185,078.27	422,888.42	4	300,965.95	179,416.77	37,028.70	84,520.48
5	1,505,039.32	896,965.63	185,152.20	422,921.48	5	301,037.72	179,400.67	37,032.02	84,605.03
6	1,505,299.77	897,095.85	185,178.32	423,025.60	6	300,966.12	179,380.58	37,018.32	84,567.22
7	1,504,959.70	897,037.10	185,076.63	422,845.97	7	301,001.78	179,390.92	37,010.90	84,599.97
8	1,505,102.50	897,010.87	185,193.02	422,898.62	8	300,984.28	179,384.83	36,994.75	84,604.70
9	1,505,275.65	897,105.47	185,114.55	423,055.63	9	300,962.33	179,426.22	37,030.45	84,505.67
10	1,505,054.68	897,029.48	185,099.12	422,926.08	10	301,028.75	179,408.63	37,017.03	84,603.08
11	1,505,038.82	897,059.12	185,108.03	422,871.67	11	300,953.82	179,370.67	36,983.58	84,599.57
12	1,504,933.08	896,981.05	185,116.32	422,835.72	12	300,952.18	179,349.65	37,026.38	84,576.15
Average	***1,505,067.60***	***897,014.08***	***185,131.38***	***422,922.14***	*Average*	***300,985.93***	***179,393.02***	***37,014.86***	***84,578.04***

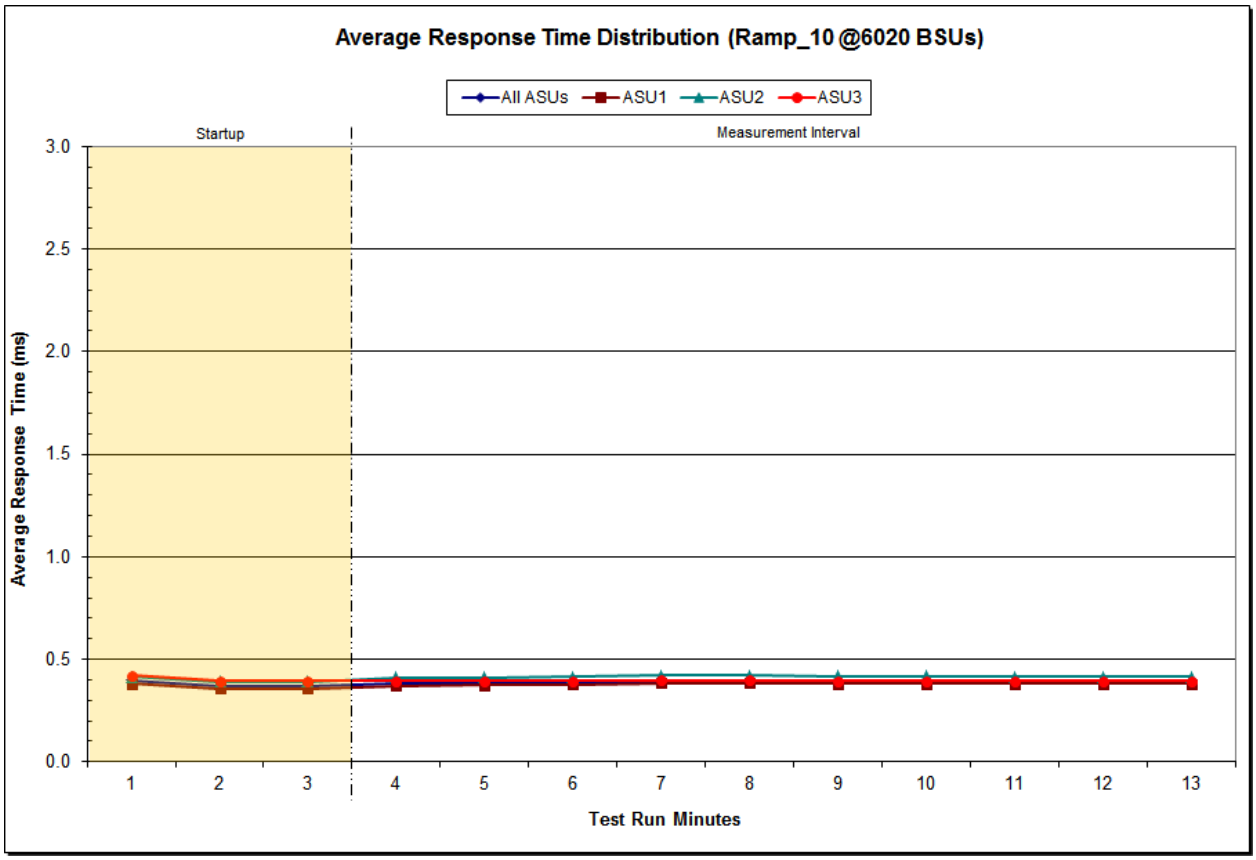
Response Time Ramp Distribution (IOPS) Graph



SPC-1 LRT™ Average Response Time (ms) Distribution Data

6,020 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	14:14:26	14:17:26	0-2	0:03:00
Measurement Interval	14:17:26	14:27:28	3-12	0:10:02
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	0.40	0.38	0.42	0.42
1	0.37	0.36	0.39	0.39
2	0.37	0.36	0.39	0.39
3	0.38	0.37	0.41	0.40
4	0.38	0.37	0.41	0.39
5	0.39	0.38	0.42	0.39
6	0.39	0.38	0.42	0.40
7	0.39	0.38	0.42	0.40
8	0.39	0.38	0.42	0.39
9	0.39	0.38	0.42	0.40
10	0.39	0.38	0.42	0.39
11	0.39	0.38	0.42	0.39
12	0.39	0.38	0.42	0.39
Average	***0.39***	***0.38***	***0.42***	***0.39***

SPC-1 LRT™ Average Response Time (ms) Distribution Graph



SPC-1 LRT™ (10%) – Measured Intensity Multiplier and Coefficient of Variation

Clause 3.4.3

IM – Intensity Multiplier: *The ratio of I/Os for each I/O stream relative to the total I/Os for all I/O streams (ASU1-1 – ASU3-1) as required by the benchmark specification.*

Clauses 5.1.10 and 5.3.15.2

MIM – Measured Intensity Multiplier: *The Measured Intensity Multiplier represents the ratio of measured I/Os for each I/O stream relative to the total I/Os measured for all I/O streams (ASU1-1 – ASU3-1). This value may differ from the corresponding Expected Intensity Multiplier by no more than 5%.*

Clause 5.3.15.3

COV – Coefficient of Variation: *This measure of variation for the Measured Intensity Multiplier cannot exceed 0.2.*

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2809	0.0700	0.2101	0.0180	0.0700	0.0350	0.2810
COV	0.002	0.000	0.001	0.000	0.001	0.001	0.001	0.000

Repeatability Test

Clause 5.4.5

The Repeatability Test demonstrates the repeatability and reproducibility of the SPC-1 IOPS™ primary metric and the SPC-1 LRT™ metric generated in earlier Test Runs.

There are two identical Repeatability Test Phases. Each Test Phase contains two Test Runs. Each of the Test Runs will have a Measurement Interval of no less than ten (10) minutes. The two Test Runs in each Test Phase will be executed without interruption or any type of manual intervention.

The first Test Run in each Test Phase is executed at the 10% load point. The Average Response Time from each of the Test Runs is compared to the SPC-1 LRT™ metric. Each Average Response Time value must be less than the SPC-1 LRT™ metric plus 5% or less than the SPC-1 LRT™ metric plus one (1) millisecond (ms).

The second Test Run in each Test Phase is executed at the 100% load point. The I/O Request Throughput from the Test Runs is compared to the SPC-1 IOPS™ primary metric. Each I/O Request Throughput value must be greater than the SPC-1 IOPS™ primary metric minus 5%. In addition, the Average Response Time for each Test Run cannot exceed 30 milliseconds.

If any of the above constraints are not met, the benchmark measurement is invalid.

Clause 9.4.3.7.5

The following content shall appear in the FDR for each Test Run in the two Repeatability Test Phases:

1. *A table containing the results of the Repeatability Test.*
2. *An I/O Request Throughput Distribution graph and table.*
3. *An Average Response Time Distribution graph and table.*
4. *The human readable Test Run Results File produced by the Workload Generator.*
5. *A listing or screen image of all input parameters supplied to the Workload Generator.*

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 95.

Repeatability Test Results File

The values for the SPC-1 IOPS™, SPC-1 LRT™, and the Repeatability Test measurements are listed in the tables below.

	SPC-1 IOPS™
Primary Metrics	***3,010,007.37***
Repeatability Test Phase 1	3,010,075.52
Repeatability Test Phase 2	3,010,089.76

The SPC-1 IOPS™ values in the above table were generated using 100% of the specified Business Scaling Unit (BSU) load level. Each of the Repeatability Test Phase values for SPC-1 IOPS™ must greater than 95% of the reported SPC-1 IOPS™ Primary Metric.

	SPC-1 LRT™
Primary Metrics	***0.39***
Repeatability Test Phase 1	0.39
Repeatability Test Phase 2	0.39

The average response time values in the SPC-1 LRT™ column were generated using 10% of the specified Business Scaling Unit (BSU) load level. Each of the Repeatability Test Phase values for SPC-1 LRT™ must be less than 105% of the reported SPC-1 LRT™ Primary Metric or less than the reported SPC-1 LRT™ Primary Metric plus one (1) millisecond (ms).

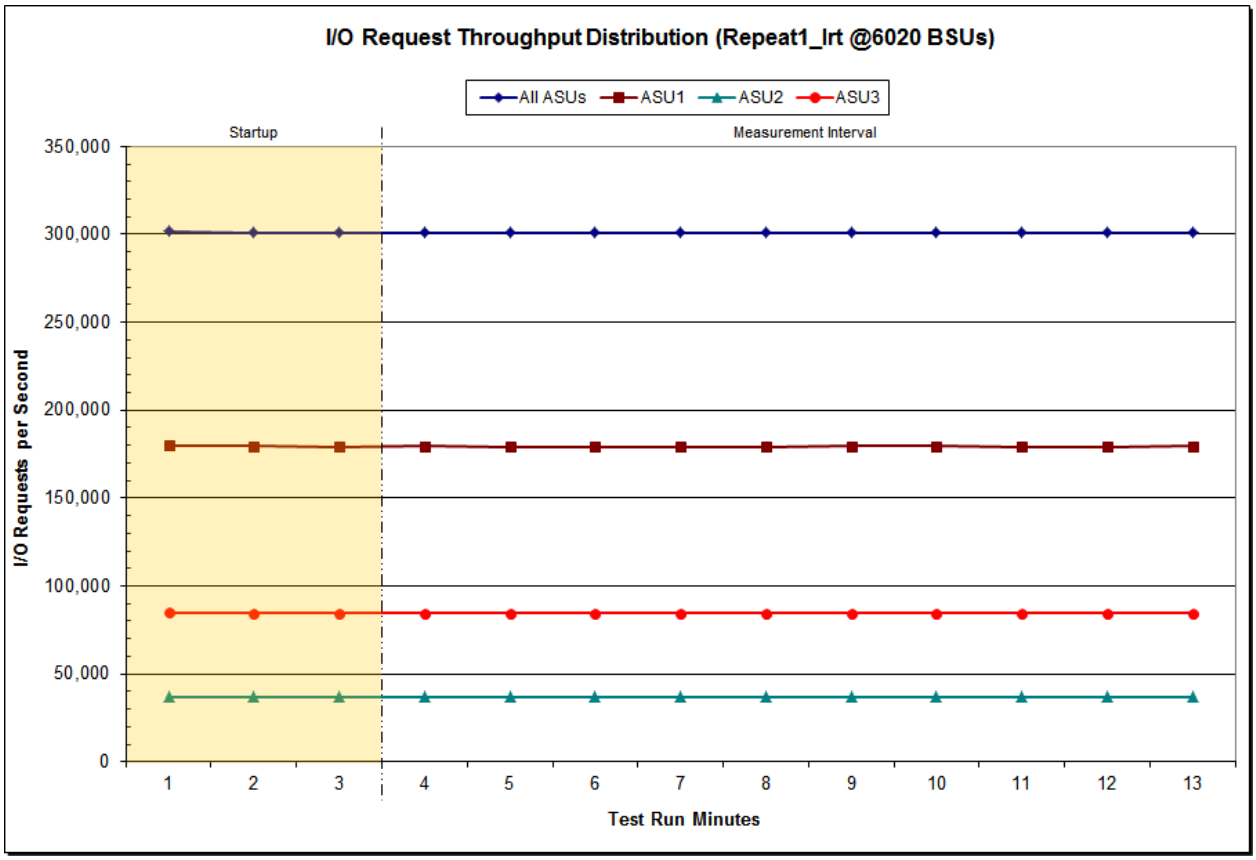
A link to the test result file generated from each Repeatability Test Run is listed below.

Repeatability Test Phase 1, Test Run 1 (LRT)
Repeatability Test Phase 1, Test Run 2 (IOPS)
Repeatability Test Phase 2, Test Run 1 (LRT)
Repeatability Test Phase 2, Test Run 2 (IOPS)

Repeatability 1 LRT – I/O Request Throughput Distribution Data

6,020 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	14:55:27	14:58:27	0-2	0:03:00
Measurement Interval	14:58:27	15:08:29	3-12	0:10:02
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	301,639.05	179,798.10	37,092.60	84,748.35
1	301,108.53	179,496.35	37,011.48	84,600.70
2	300,902.27	179,335.80	36,999.07	84,567.40
3	300,991.78	179,426.42	36,989.75	84,575.62
4	301,002.90	179,372.62	37,010.93	84,619.35
5	300,924.15	179,323.97	37,017.08	84,583.10
6	300,917.92	179,344.05	37,050.30	84,523.57
7	300,969.62	179,369.55	37,039.52	84,560.55
8	301,060.63	179,423.10	37,040.15	84,597.38
9	301,061.32	179,409.65	37,040.95	84,610.72
10	300,908.53	179,286.77	37,003.80	84,617.97
11	300,932.27	179,309.75	36,994.83	84,627.68
12	301,016.47	179,419.10	37,021.72	84,575.65
Average	***300,978.56***	***179,368.50***	***37,020.90***	***84,589.16***

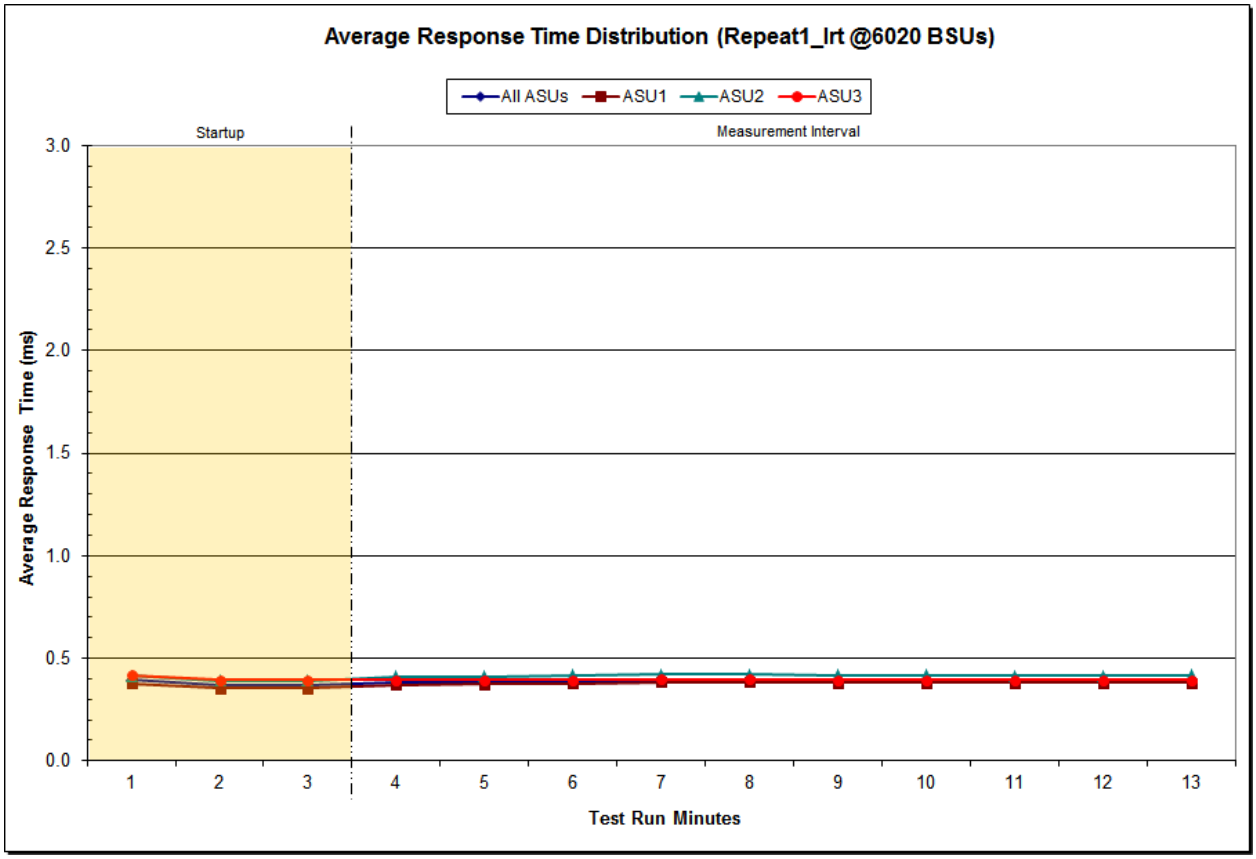
Repeatability 1 LRT – I/O Request Throughput Distribution Graph



Repeatability 1 LRT –Average Response Time (ms) Distribution Data

6,020 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	14:55:27	14:58:27	0-2	0:03:00
Measurement Interval	14:58:27	15:08:29	3-12	0:10:02
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	0.39	0.38	0.41	0.42
1	0.37	0.35	0.39	0.39
2	0.37	0.36	0.39	0.40
3	0.38	0.37	0.41	0.40
4	0.38	0.37	0.41	0.39
5	0.39	0.38	0.42	0.39
6	0.39	0.38	0.42	0.40
7	0.39	0.39	0.42	0.40
8	0.39	0.38	0.42	0.39
9	0.39	0.38	0.42	0.39
10	0.39	0.38	0.42	0.39
11	0.39	0.38	0.42	0.39
12	0.39	0.38	0.42	0.39
Average	***0.39***	***0.38***	***0.42***	***0.39***

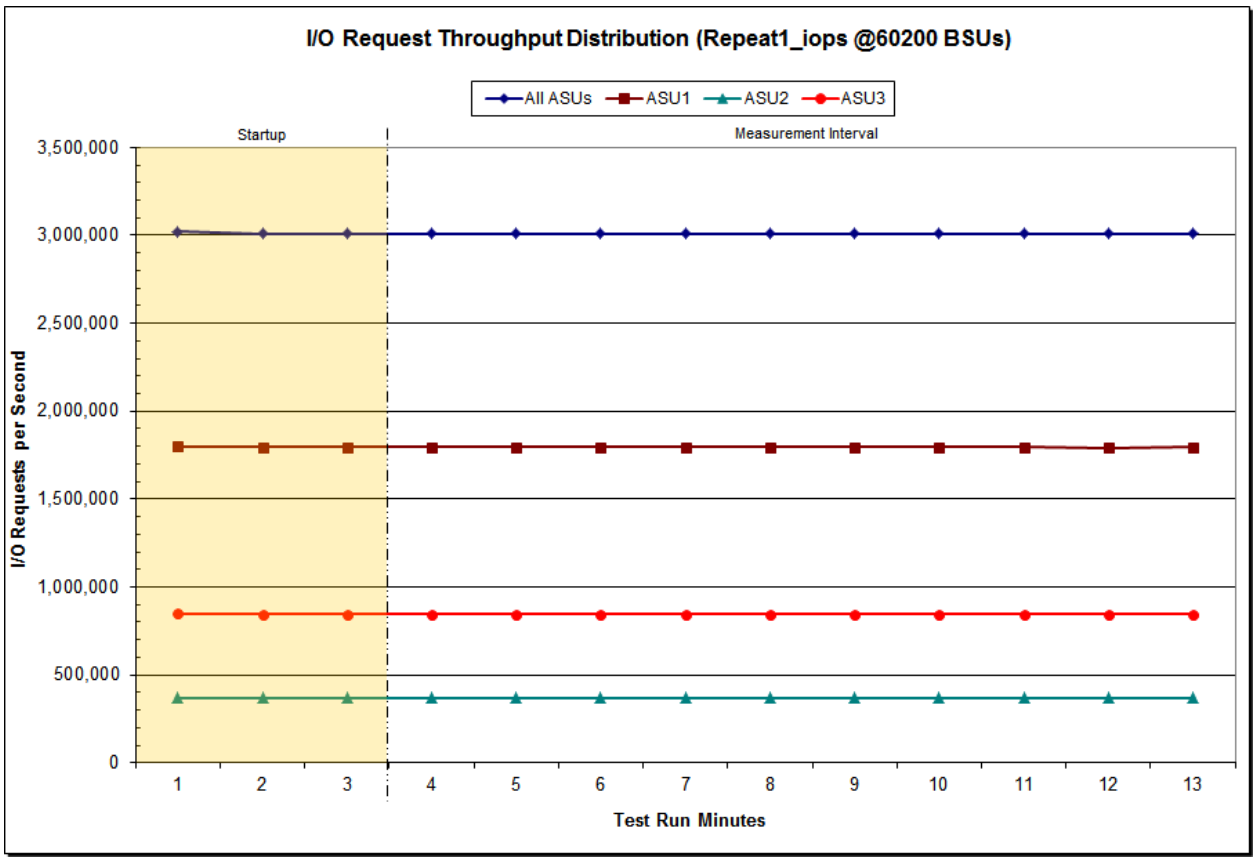
Repeatability 1 LRT –Average Response Time (ms) Distribution Graph



Repeatability 1 IOPS – I/O Request Throughput Distribution Data

60,200 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	16:11:46	16:14:46	0-2	0:03:00
Measurement Interval	16:14:46	16:24:48	3-12	0:10:02
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	3,019,795.55	1,799,937.85	371,473.20	848,384.50
1	3,010,308.47	1,794,297.92	370,223.90	845,786.65
2	3,009,866.23	1,794,065.90	370,149.18	845,651.15
3	3,010,333.52	1,794,180.72	370,356.08	845,796.72
4	3,009,614.48	1,793,764.18	370,197.75	845,652.55
5	3,010,172.98	1,793,965.83	370,248.33	845,958.82
6	3,010,126.87	1,794,005.20	370,200.12	845,921.55
7	3,010,099.98	1,794,011.95	370,202.02	845,886.02
8	3,010,163.00	1,794,005.13	370,337.60	845,820.27
9	3,010,046.03	1,793,987.62	370,131.10	845,927.32
10	3,010,157.38	1,793,981.52	370,290.58	845,885.28
11	3,009,896.22	1,793,757.37	370,305.97	845,832.88
12	3,010,144.77	1,793,909.37	370,318.10	845,917.30
Average	***3,010,075.52***	***1,793,956.89***	***370,258.77***	***845,859.87***

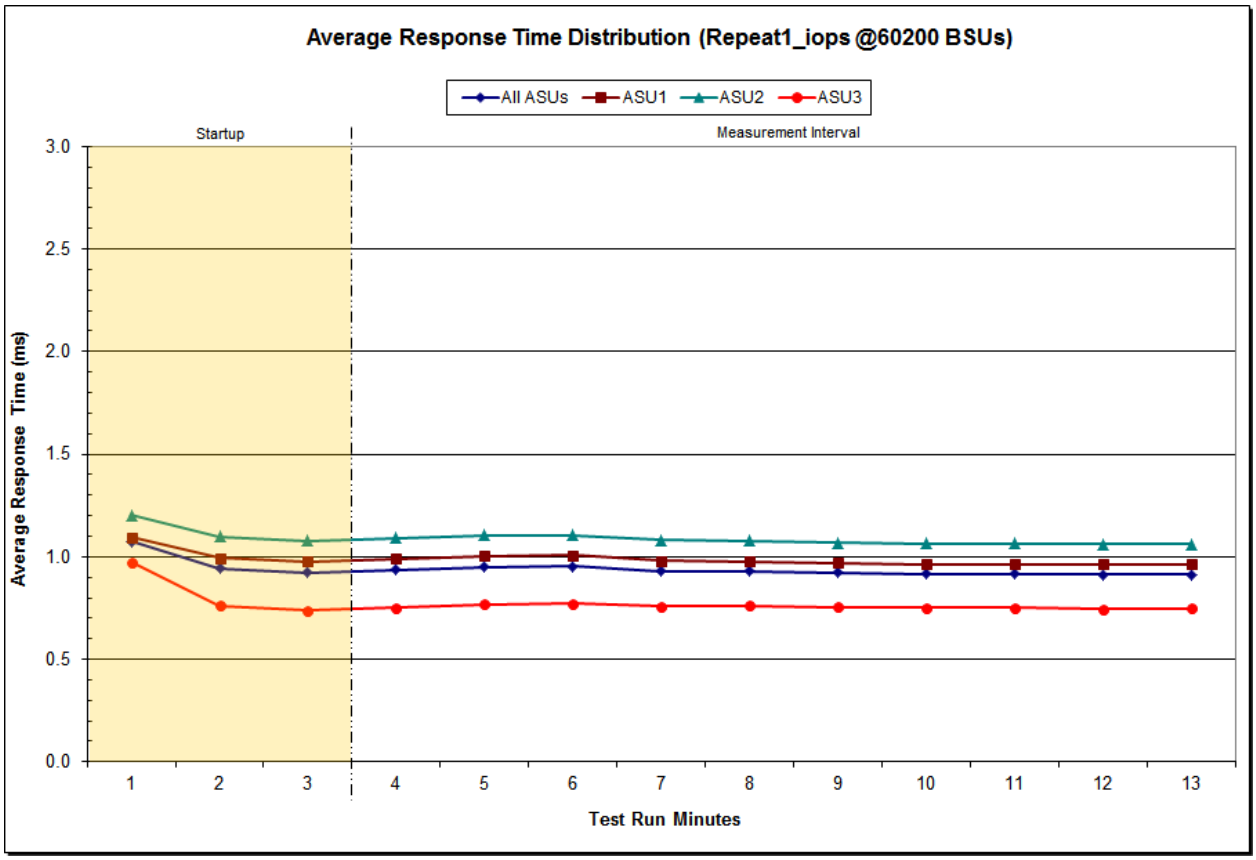
Repeatability 1 IOPS – I/O Request Throughput Distribution Graph



Repeatability 1 IOPS –Average Response Time (ms) Distribution Data

60,200 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	16:11:46	16:14:46	0-2	0:03:00
Measurement Interval	16:14:46	16:24:48	3-12	0:10:02
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	1.07	1.10	1.20	0.97
1	0.94	0.99	1.10	0.76
2	0.92	0.98	1.08	0.74
3	0.94	0.99	1.09	0.75
4	0.95	1.00	1.10	0.77
5	0.95	1.01	1.11	0.77
6	0.93	0.98	1.08	0.76
7	0.93	0.98	1.08	0.76
8	0.92	0.97	1.07	0.76
9	0.92	0.96	1.06	0.75
10	0.92	0.96	1.06	0.75
11	0.91	0.96	1.06	0.74
12	0.91	0.96	1.06	0.75
Average	***0.93***	***0.98***	***1.08***	***0.76***

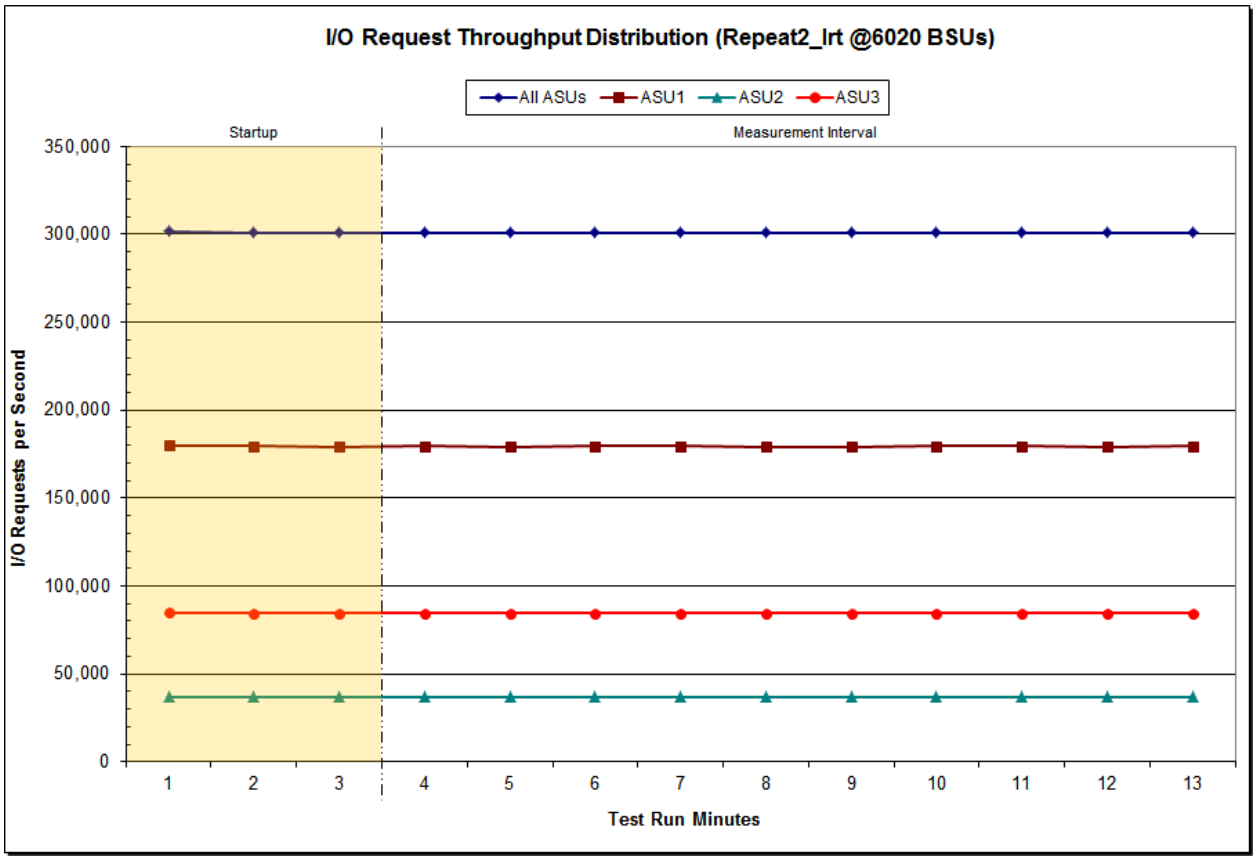
Repeatability 1 IOPS –Average Response Time (ms) Distribution Graph



Repeatability 2 LRT – I/O Request Throughput Distribution Data

6,020 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	16:52:49	16:55:49	0-2	0:03:00
Measurement Interval	16:55:49	17:05:51	3-12	0:10:02
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	301,773.15	179,837.75	37,125.52	84,809.88
1	301,008.17	179,438.73	37,014.80	84,554.63
2	300,913.70	179,372.72	36,962.30	84,578.68
3	300,975.88	179,423.52	37,055.28	84,497.08
4	300,914.27	179,333.78	36,988.12	84,592.37
5	300,985.33	179,419.32	36,979.90	84,586.12
6	300,977.82	179,379.70	37,000.35	84,597.77
7	300,835.58	179,334.13	36,981.82	84,519.63
8	300,956.32	179,348.30	37,039.28	84,568.73
9	301,042.07	179,450.52	37,067.15	84,524.40
10	301,058.25	179,409.85	37,052.90	84,595.50
11	301,001.78	179,367.97	37,024.60	84,609.22
12	300,989.35	179,410.35	37,036.47	84,542.53
Average	***300,973.67***	***179,387.74***	***37,022.59***	***84,563.34***

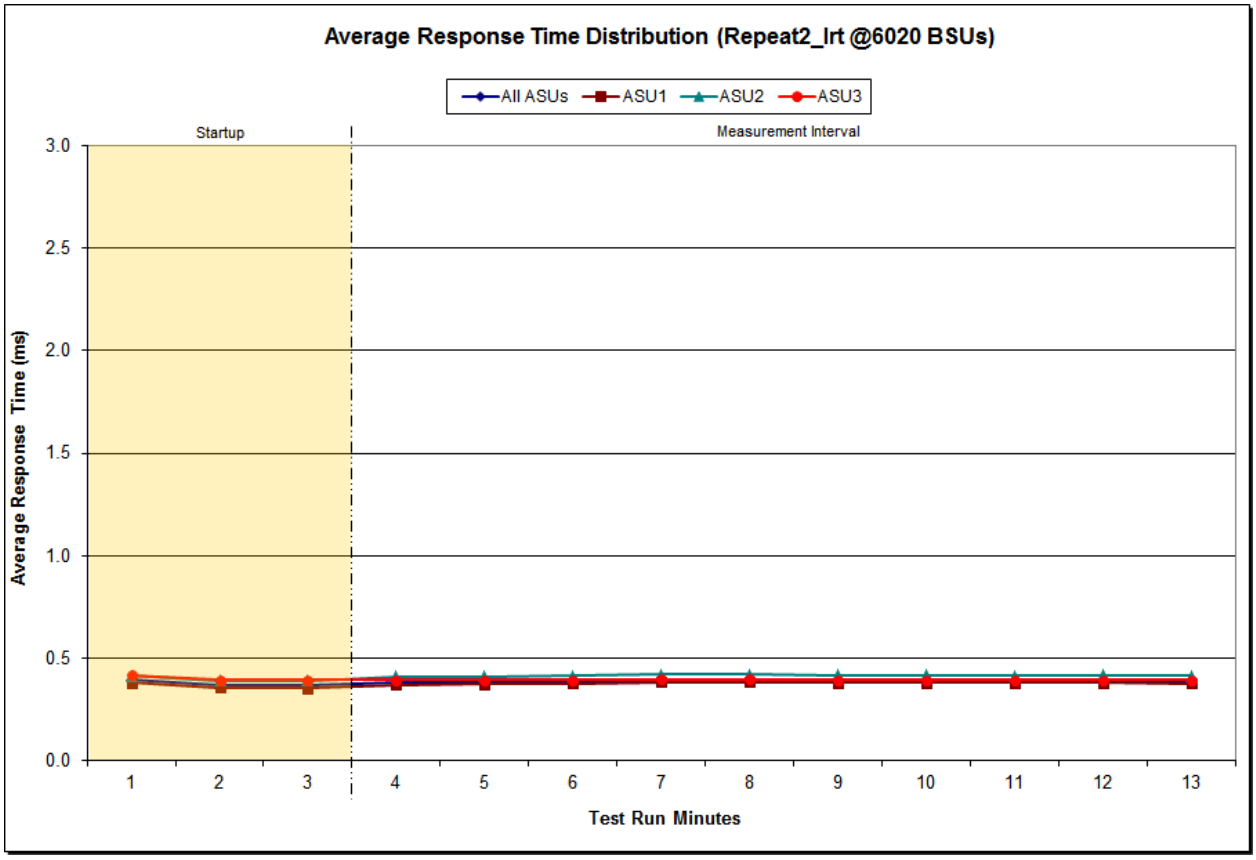
Repeatability 2 LRT – I/O Request Throughput Distribution Graph



Repeatability 2 LRT –Average Response Time (ms) Distribution Data

6,020 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	16:52:49	16:55:49	0-2	0:03:00
Measurement Interval	16:55:49	17:05:51	3-12	0:10:02
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	0.39	0.38	0.41	0.42
1	0.37	0.36	0.39	0.40
2	0.37	0.36	0.39	0.39
3	0.38	0.37	0.41	0.40
4	0.38	0.37	0.41	0.39
5	0.39	0.38	0.42	0.39
6	0.39	0.38	0.42	0.40
7	0.39	0.39	0.42	0.40
8	0.39	0.38	0.42	0.39
9	0.39	0.38	0.42	0.39
10	0.39	0.38	0.42	0.39
11	0.39	0.38	0.42	0.39
12	0.39	0.38	0.41	0.39
Average	***0.39***	***0.38***	***0.42***	***0.39***

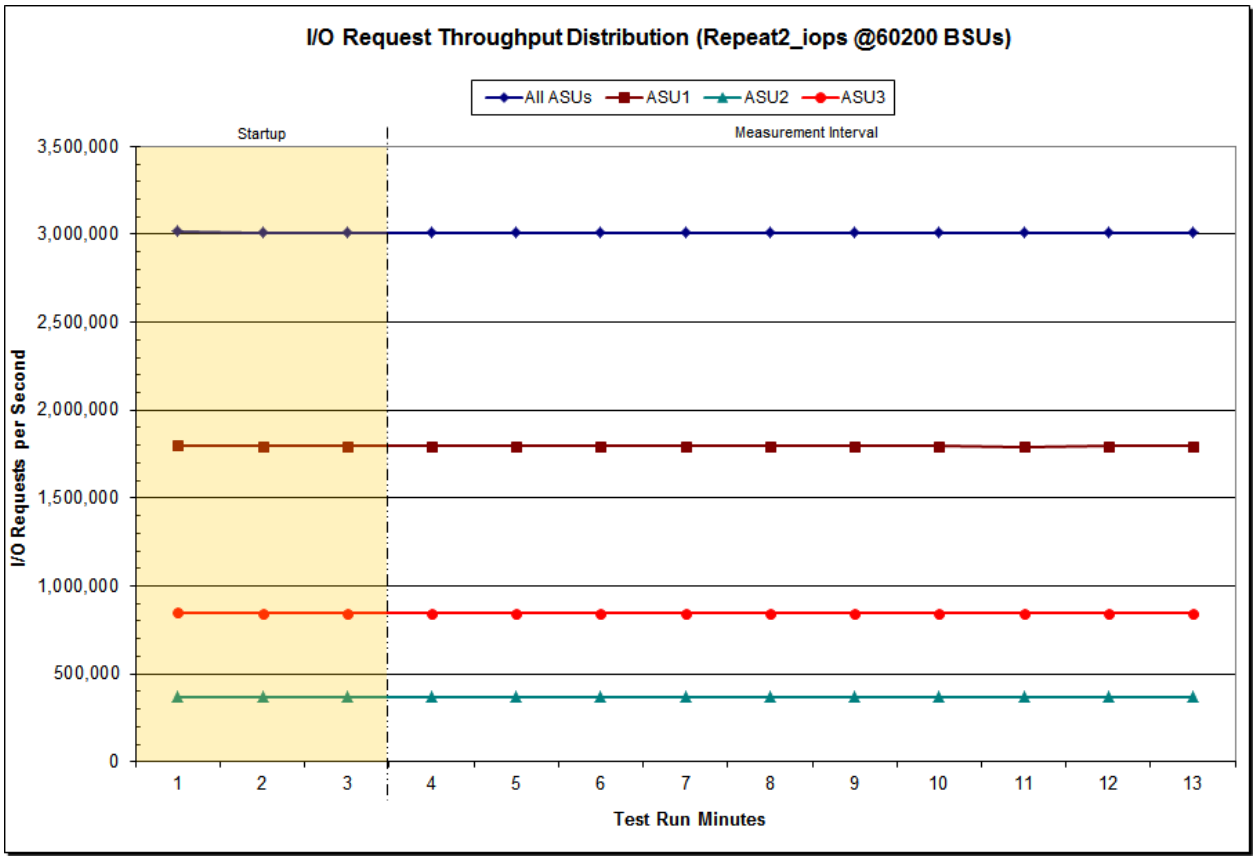
Repeatability 2 LRT –Average Response Time (ms) Distribution Graph



Repeatability 2 IOPS – I/O Request Throughput Distribution Data

60,200 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	18:08:02	18:11:02	0-2	0:03:00
Measurement Interval	18:11:02	18:21:05	3-12	0:10:03
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	3,019,390.70	1,799,417.00	371,544.13	848,429.57
1	3,009,739.83	1,793,843.53	370,155.93	845,740.37
2	3,010,184.03	1,794,122.32	370,333.80	845,727.92
3	3,010,273.40	1,794,178.63	370,230.70	845,864.07
4	3,010,090.70	1,794,102.55	370,197.82	845,790.33
5	3,009,903.67	1,793,805.72	370,199.37	845,898.58
6	3,009,799.68	1,793,929.32	370,215.03	845,655.33
7	3,010,485.20	1,794,145.70	370,297.38	846,042.12
8	3,009,772.95	1,793,812.43	370,313.78	845,646.73
9	3,009,791.38	1,793,914.75	370,121.20	845,755.43
10	3,010,034.40	1,793,593.50	370,354.35	846,086.55
11	3,010,414.47	1,794,275.65	370,324.78	845,814.03
12	3,010,331.75	1,794,090.87	370,137.33	846,103.55
Average	***3,010,089.76***	***1,793,984.91***	***370,239.18***	***845,865.67***

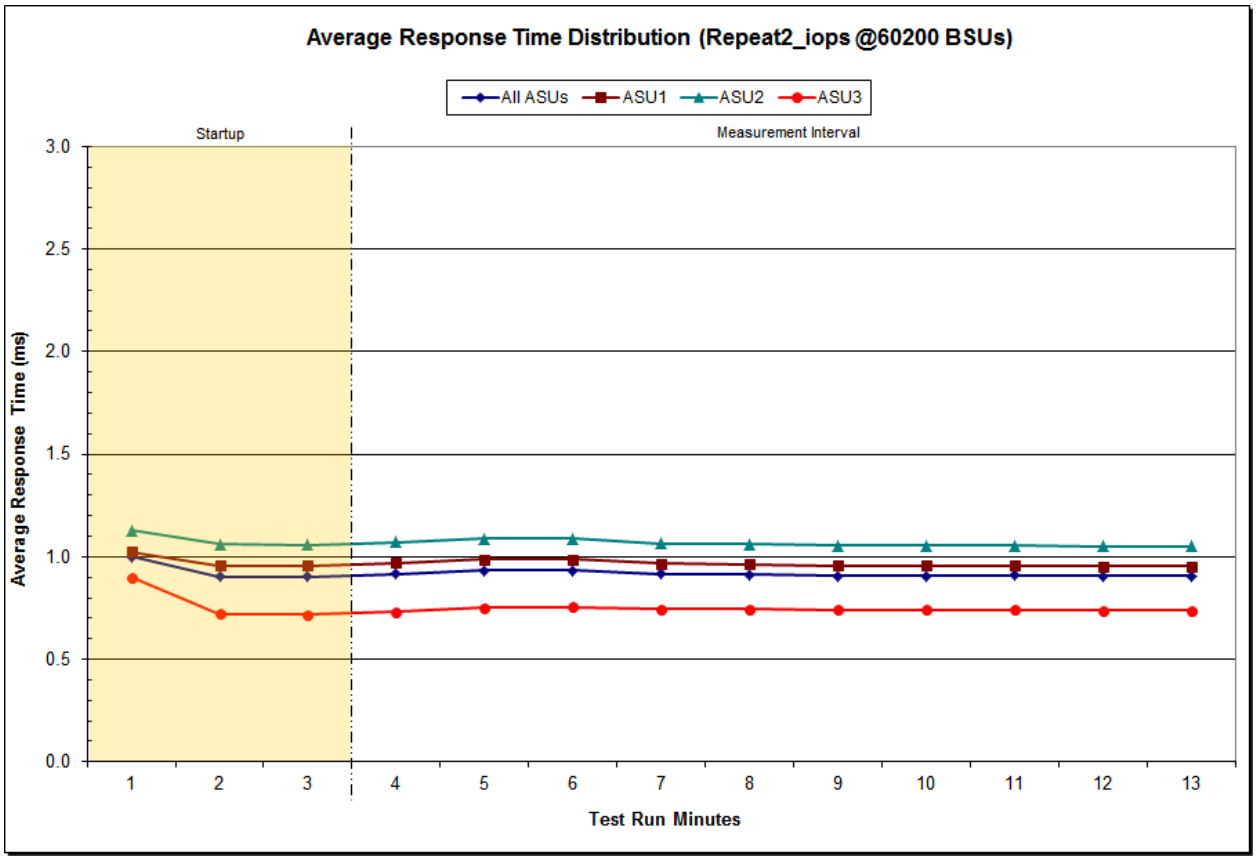
Repeatability 2 IOPS – I/O Request Throughput Distribution Graph



Repeatability 2 IOPS –Average Response Time (ms) Distribution Data

60,200 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	18:08:02	18:11:02	0-2	0:03:00
Measurement Interval	18:11:02	18:21:05	3-12	0:10:03
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	1.00	1.02	1.13	0.90
1	0.90	0.96	1.06	0.72
2	0.90	0.96	1.06	0.72
3	0.92	0.97	1.07	0.73
4	0.93	0.99	1.09	0.75
5	0.93	0.99	1.09	0.76
6	0.92	0.97	1.07	0.74
7	0.91	0.96	1.06	0.75
8	0.91	0.96	1.05	0.74
9	0.91	0.96	1.05	0.74
10	0.91	0.96	1.06	0.74
11	0.91	0.95	1.05	0.74
12	0.91	0.95	1.05	0.74
Average	***0.91***	***0.97***	***1.06***	***0.74***

Repeatability 2 IOPS –Average Response Time (ms) Distribution Graph



Repeatability 1 (LRT)
Measured Intensity Multiplier and Coefficient of Variation

Clause 3.4.3

IM – Intensity Multiplier: *The ratio of I/Os for each I/O stream relative to the total I/Os for all I/O streams (ASU1-1 – ASU3-1) as required by the benchmark specification.*

Clauses5.1.10 and 5.3.15.2

MIM – Measured Intensity Multiplier: *The Measured Intensity Multiplier represents the ratio of measured I/Os for each I/O stream relative to the total I/Os measured for all I/O streams (ASU1-1 – ASU3-1). This value may differ from the corresponding Expected Intensity Multiplier by no more than 5%.*

Clause 5.3.15.3

COV – Coefficient of Variation: *This measure of variation for the Measured Intensity Multiplier cannot exceed 0.2.*

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
COV	0.001	0.000	0.001	0.000	0.002	0.001	0.001	0.000

Repeatability 1 (IOPS)
Measured Intensity Multiplier and Coefficient of Variation

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
COV	0.000	0.000	0.000	0.000	0.000	0.000	0.0350	0.2810

Repeatability 2 (LRT)
Measured Intensity Multiplier and Coefficient of Variation

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2101	0.0180	0.0700	0.0350	0.2810
COV	0.001	0.000	0.001	0.001	0.001	0.001	0.001	0.000

Repeatability 2 (IOPS)
Measured Intensity Multiplier and Coefficient of Variation

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0,0350	0.2810
COV	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000

Data Persistence Test

Clause 6

The Data Persistence Test demonstrates the Tested Storage Configuration (TSC):

- *Is capable of maintain data integrity across a power cycle.*
- *Ensures the transfer of data between Logical Volumes and host systems occurs without corruption or loss.*

The SPC-1 Workload Generator will write 16 block I/O requests at random over the total Addressable Storage Capacity of the TSC for ten (10) minutes at a minimum of 25% of the load used to generate the SPC-1 IOPS™ primary metric. The bit pattern selected to be written to each block as well as the address of the block will be retained in a log file.

The Tested Storage Configuration (TSC) will be shutdown and restarted using a power off/power on cycle at the end of the above sequence of write operations. In addition, any caches employing battery backup must be flushed/emptied.

The SPC-1 Workload Generator will then use the above log file to verify each block written contains the correct bit pattern.

Clause 9.4.3.8

The following content shall appear in this section of the FDR:

1. *A listing or screen image of all input parameters supplied to the Workload Generator.*
2. *For the successful Data Persistence Test Run, a table illustrating key results. The content, appearance, and format of this table are specified in Table 9-12. Information displayed in this table shall be obtained from the Test Run Results File referenced below in #3.*
3. *For the successful Data Persistence Test Run, the human readable Test Run Results file produced by the Workload Generator (may be contained in an appendix).*

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 95.

Data Persistence Test Results File

A link to each test result file generated from each Data Persistence Test is listed below.

Persistence 1 Test Results File
Persistence 2 Test Results File

Data Persistence Test Results

Data Persistence Test Results	
Data Persistence Test Run Number: 1	
Total Number of Logical Blocks Written	24,415,829
Total Number of Logical Blocks Verified	20,134,308
Total Number of Logical Blocks that Failed Verification	0
Time Duration for Writing Test Logical Blocks	5 minutes
Size in bytes of each Logical Block	1024
Number of Failed I/O Requests in the process of the Test	0

If approved by the SPC Auditor, the SPC-2 Persistence Test may be used to meet the SPC-1 persistence requirements. Both the SPC-1 and SPC-2 Persistence Tests provide the same level of functionality and verification of data integrity. The SPC-2 Persistence Test may be easily configured to address an SPC-1 storage configuration. The SPC-2 Persistence Test extends the size of storage configurations that may be tested and significantly reduces the test duration of such configurations.

The SPC-2 Persistence Test was approved for use in this set of audited measurements.

In some cases the same address was the target of multiple writes, which resulted in more Logical Blocks Written than Logical Blocks Verified. In the case of multiple writes to the same address, the pattern written and verified must be associated with the last write to that address.

PRICED STORAGE CONFIGURATION AVAILABILITY DATE

Clause 9.4.3.9

The committed delivery data for general availability (Availability Date) of all products that comprise the Priced Storage Configuration must be reported. When the Priced Storage Configuration includes products or components with different availability dates, the reported Availability Date for the Priced Storage Configuration must be the date at which all components are committed to be available.

The Huawei OceanStor™ 18800 V3 as documented in this Full Disclosure Report is currently available for customer purchase and shipment.

PRICING INFORMATION

Clause 9.4.3.3.6

The Executive Summary shall contain a pricing spreadsheet as documented in Clause 8.3.1.

Pricing information may be found in the Priced Storage Configuration Pricing section on page 16.

TESTED STORAGE CONFIGURATION (TSC) AND PRICED STORAGE CONFIGURATION DIFFERENCES

Clause 9.4.3.3.8

The Executive Summary shall contain a list of all differences between the Tested Storage Configuration (TSC) and the Priced Storage Configuration.

A list of all differences between the Tested Storage Configuration (TSC) and Priced Storage Configuration may be found in the Executive Summary portion of this document on page 16.

ANOMALIES OR IRREGULARITIES

Clause 9.4.3.10

The FDR shall include a clear and complete description of any anomalies or irregularities encountered in the course of executing the SPC-1 benchmark that may in any way call into question the accuracy, verifiability, or authenticity of information published in this FDR.

There were no anomalies or irregularities encountered during the SPC-1 Remote Audit of the Huawei OceanStor™ 18800 V3.

APPENDIX A: SPC-1 GLOSSARY

"Decimal" *(powers of ten)* Measurement Units

In the storage industry, the terms "kilo", "mega", "giga", "tera", "peta", and "exa" are commonly used prefixes for computing performance and capacity. For the purposes of the SPC workload definitions, all of the following terms are defined in "powers of ten" measurement units.

A kilobyte (KB) is equal to 1,000 (10^3) bytes.

A megabyte (MB) is equal to 1,000,000 (10^6) bytes.

A gigabyte (GB) is equal to 1,000,000,000 (10^9) bytes.

A terabyte (TB) is equal to 1,000,000,000,000 (10^{12}) bytes.

A petabyte (PB) is equal to 1,000,000,000,000,000 (10^{15}) bytes

An exabyte (EB) is equal to 1,000,000,000,000,000,000 (10^{18}) bytes

"Binary" *(powers of two)* Measurement Units

The sizes reported by many operating system components use "powers of two" measurement units rather than "power of ten" units. The following standardized definitions and terms are also valid and may be used in this document.

A kibibyte (KiB) is equal to 1,024 (2^{10}) bytes.

A mebibyte (MiB) is equal to 1,048,576 (2^{20}) bytes.

A gigibyte (GiB) is equal to 1,073,741,824 (2^{30}) bytes.

A tebibyte (TiB) is equal to 1,099,511,627,776 (2^{40}) bytes.

A pebibyte (PiB) is equal to 1,125,899,906,842,624 (2^{50}) bytes.

An exbibyte (EiB) is equal to 1,152,921,504,606,846,967 (2^{60}) bytes.

SPC-1 Data Repository Definitions

Total ASU Capacity: The total storage capacity read and written in the course of executing the SPC-1 benchmark.

Application Storage Unit (ASU): The logical interface between the storage and SPC-1 Workload Generator. The three ASUs (Data, User, and Log) are typically implemented on one or more Logical Volume.

Logical Volume: The division of Addressable Storage Capacity into individually addressable logical units of storage used in the SPC-1 benchmark. Each Logical Volume is implemented as a single, contiguous address space.

Addressable Storage Capacity: The total storage (sum of Logical Volumes) that can be read and written by application programs such as the SPC-1 Workload Generator.

Configured Storage Capacity: This capacity includes the Addressable Storage Capacity and any other storage (parity disks, hot spares, etc.) necessary to implement the Addressable Storage Capacity.

Physical Storage Capacity: The formatted capacity of all storage devices physically present in the Tested Storage Configuration (TSC).

Data Protection Overhead: The storage capacity required to implement the selected level of data protection.

Required Storage: The amount of Configured Storage Capacity required to implement the Addressable Storage Configuration, excluding the storage required for the three ASUs.

Global Storage Overhead: The amount of Physical Storage Capacity that is required for storage subsystem use and unavailable for use by application programs.

Total Unused Storage: The amount of storage capacity available for use by application programs but not included in the Total ASU Capacity.

SPC-1 Data Protection Levels

Protected 1: The single point of failure of any ***storage device*** in the configuration will not result in permanent loss of access to or integrity of the SPC-1 Data Repository.

Protected 2: The single point of failure of any ***component*** in the configuration will not result in permanent loss of access to or integrity of the SPC-1 Data Repository.

SPC-1 Test Execution Definitions

Average Response Time: The sum of the Response Times for all Measured I/O Requests divided by the total number of Measured I/O Requests.

Completed I/O Request: An I/O Request with a Start Time and a Completion Time (see "I/O Completion Types" below).

Completion Time: The time recorded by the Workload Generator when an I/O Request is satisfied by the TSC as signaled by System Software.

Data Rate: The data transferred in all Measured I/O Requests in an SPC-1 Test Run divided by the length of the Test Run in seconds.

Expected I/O Count: For any given I/O Stream and Test Phase, the product of 50 times the BSU level, the duration of the Test Phase in seconds, and the Intensity Multiplier for that I/O Stream.

Failed I/O Request: Any I/O Request issued by the Workload Generator that could not be completed or was signaled as failed by System Software. A Failed I/O Request has no Completion Time (see "I/O Completion Types" below).

I/O Request Throughput: The total number of Measured I/O requests in an SPC-1 Test Run divided by the duration of the Measurement Interval in seconds.

In-Flight I/O Request: An I/O Request issued by the I/O Command Generator to the TSC that has a recorded Start Time, but does not complete within the Measurement Interval (see "I/O Completion Types" below).

Measured I/O Request: A Completed I/O Request with a Completion Time occurring within the Measurement Interval (see "I/O Completion Types" below).

Measured Intensity Multiplier: The percentage of all Measured I/O Requests that were issued by a given I/O Stream.

Measurement Interval: The finite and contiguous time period, after the TSC has reached Steady State, when data is collected by a Test Sponsor to generate an SPC-1 test result or support an SPC-1 test result.

Ramp-Up: The time required for the Benchmark Configuration (BC) to produce Steady State throughput after the Workload Generator begins submitting I/O Requests to the TSC for execution.

Ramp-Down: The time required for the BC to complete all I/O Requests issued by the Workload Generator. The Ramp-Down period begins when the Workload Generator ceases to issue new I/O Requests to the TSC.

Response Time: The Response Time of a Measured I/O Request is its Completion Time minus its Start Time.

Start Time: The time recorded by the Workload Generator when an I/O Request is submitted, by the Workload Generator, to the System Software for execution on the Tested Storage Configuration (TSC).

Start-Up: The period that begins after the Workload Generator starts to submit I/O requests to the TSC and ends at the beginning of the Measurement Interval.

Shut-Down: The period between the end of the Measurement Interval and the time when all I/O Requests issued by the Workload Generator have completed or failed.

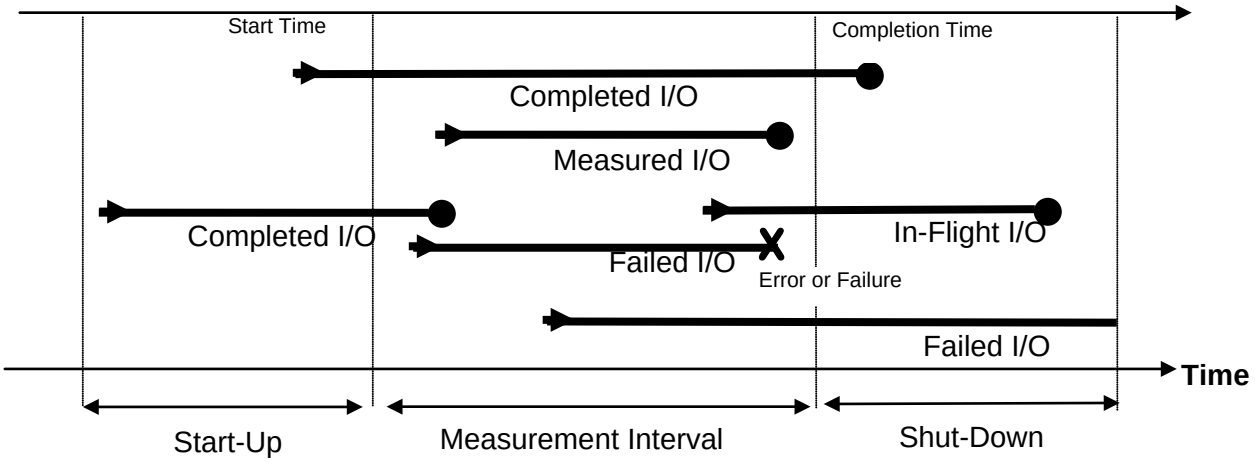
Steady State: The consistent and sustainable throughput of the TSC. During this period the load presented to the TSC by the Workload Generator is constant.

Test: A collection of Test Phases and or Test Runs sharing a common objective.

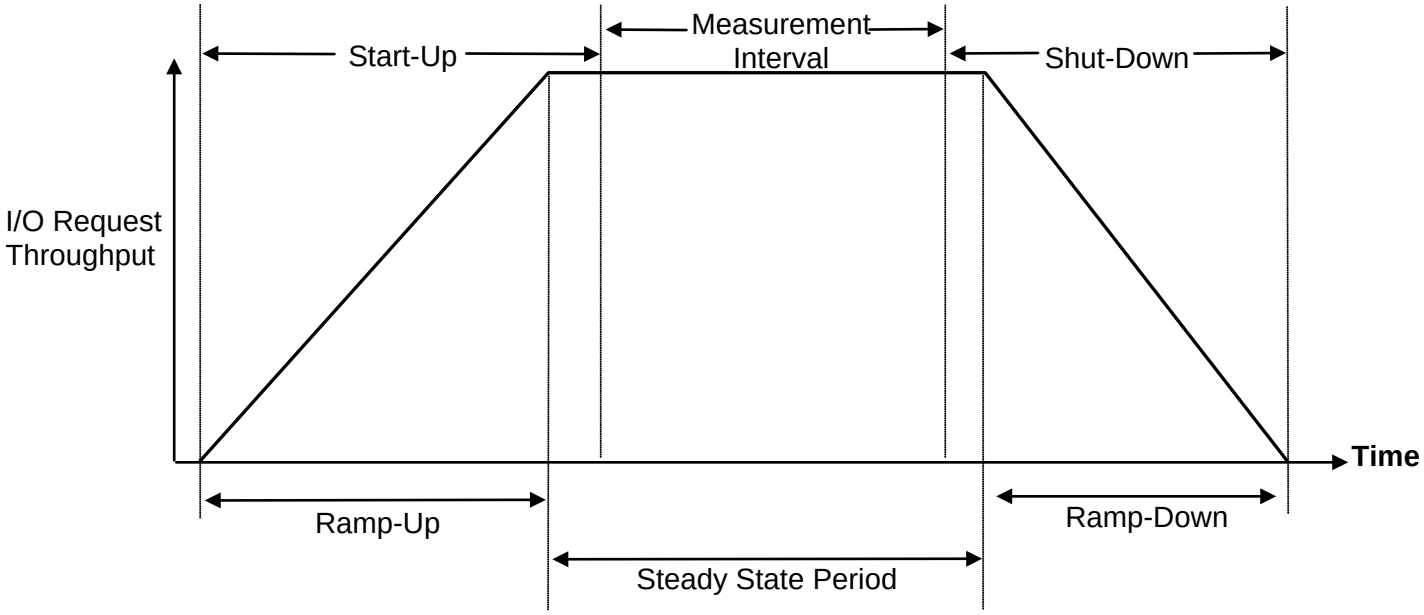
Test Run: The execution of SPC-1 for the purpose of producing or supporting an SPC-1 test result. SPC-1 Test Runs may have a finite and measured Ramp-Up period, Start-Up period, Shut-Down period, and Ramp-Down period as illustrated in the "SPC-1 Test Run Components" below. All SPC-1 Test Runs shall have a Steady State period and a Measurement Interval.

Test Phase: A collection of one or more SPC-1 Test Runs sharing a common objective and intended to be run in a specific sequence.

I/O Completion Types



SPC-1 Test Run Components



APPENDIX B: CUSTOMER TUNABLE PARAMETERS AND OPTIONS

Red Hat Enterprise Linux 7.0 (64-bit)

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. This change was done by the execution of the **scheduler.sh** script as documented in *Appendix C: Tested Storage Configuration (TSC) Creation*.

APPENDIX C: TESTED STORAGE CONFIGURATION (TSC) CREATION

The scripts referenced in Steps 2 and 3 appear in the section, Referenced Scripts.

Step 1: Create Mapping View, LUN Group, Host Group and Host

Execute the following commands using the OceanStor 18800 V3 CLI from one of the Host Systems to complete the following:

- Create one ***mapping_view*** (***map1***)
- Create one ***lun_group*** (***lg1***)
- Create one ***host_group*** (***hg1***)
- Create five ***hosts*** (***host1, host2, host3, host4, host5***)
- Add ***host1, host2, host3, host4, host5*** to ***hg1***
- Add ***hg1*** and ***lg1*** to ***map1***
- Add the FC ports' WWN to ***host1, host2, host3, host4, host5***

```
create mapping_view name=map1 mapping_view_id=1
create lun_group name=lg1 lun_group_id=1
create host_group name=hg1 host_group_id=1
create host name=host1 operating_system=Linux host_id=1
create host name=host2 operating_system=Linux host_id=2
create host name=host3 operating_system=Linux host_id=3
create host name=host4 operating_system=Linux host_id=4
create host name=host4 operating_system=Linux host_id=5

add host_group host host_group_id=1 host_id_list=1,2,3,4,5
add mapping_view host_group mapping_view_id=1 host_group_id=1
add mapping_view lun_group mapping_view_id=1 lun_group_id=1

add host initiator host_id=1 initiator_type=FC wwn=2001000e1e09f9ca
add host initiator host_id=1 initiator_type=FC wwn=2001000e1e09f9cb
add host initiator host_id=1 initiator_type=FC wwn=2001000e1ec22c60
add host initiator host_id=1 initiator_type=FC wwn=2001000e1ec22c61
add host initiator host_id=1 initiator_type=FC wwn=2001000e1ec22c66
add host initiator host_id=1 initiator_type=FC wwn=2001000e1ec22c67
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e13b7e0
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e13b7e1
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e13c520
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e13c521
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a4780
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a4781
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a4850
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a4851
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a4bd0
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a4bd1
```

```
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a4f50
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a4f51
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a5460
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a5461
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a55b0
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a55b1
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a5d20
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a5d21
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a5e40
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a5e41
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a89e0
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a89e1
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a89f0
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a89f1
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a8a40
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a8a41
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a8ac0
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a8ac1
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a8b20
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a8b21
add host initiator host_id=1 initiator_type=FC wwn=2100000e1e1a8b60
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1a8b61
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1a9c80
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1a9c81
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1a9ca0
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1a9ca1
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1aa260
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1aa261
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1c2140
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1c2141
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1c2200
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1c2201
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1c2230
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1c2231
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1e0080
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1e0081
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1e0110
add host initiator host_id=2 initiator_type=FC wwn=2100000e1e1e0111
add host initiator host_id=2 initiator_type=FC wwn=2100001b320b463e
add host initiator host_id=2 initiator_type=FC wwn=21000024ff28e93e
add host initiator host_id=2 initiator_type=FC wwn=21000024ff28e93f
add host initiator host_id=2 initiator_type=FC wwn=21000024ff28fdce
add host initiator host_id=2 initiator_type=FC wwn=21000024ff28fdcf
add host initiator host_id=2 initiator_type=FC wwn=21000024ff29aff6
add host initiator host_id=2 initiator_type=FC wwn=21000024ff29aff7
add host initiator host_id=2 initiator_type=FC wwn=21000024ff2c953e
add host initiator host_id=2 initiator_type=FC wwn=21000024ff2c953f
add host initiator host_id=2 initiator_type=FC wwn=21000024ff2c95cc
add host initiator host_id=3 initiator_type=FC wwn=21000024ff2c95cd
```

```
add host initiator host_id=3 initiator_type=FC wwn=21000024ff349bd6
add host initiator host_id=3 initiator_type=FC wwn=21000024ff349bd7
add host initiator host_id=3 initiator_type=FC wwn=21000024ff35e744
add host initiator host_id=3 initiator_type=FC wwn=21000024ff35e745
add host initiator host_id=3 initiator_type=FC wwn=21000024ff36e6e2
add host initiator host_id=3 initiator_type=FC wwn=21000024ff36e6e3
add host initiator host_id=3 initiator_type=FC wwn=21000024ff36f1fa
add host initiator host_id=3 initiator_type=FC wwn=21000024ff36f1fb
add host initiator host_id=3 initiator_type=FC wwn=21000024ff371eec
add host initiator host_id=3 initiator_type=FC wwn=21000024ff371eed
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3721e2
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3721e3
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3722cc
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3722cd
add host initiator host_id=3 initiator_type=FC wwn=21000024ff37290c
add host initiator host_id=3 initiator_type=FC wwn=21000024ff37290d
add host initiator host_id=3 initiator_type=FC wwn=21000024ff37555c
add host initiator host_id=3 initiator_type=FC wwn=21000024ff37555d
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cafe8
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cafe9
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cb158
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cb159
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cb6c0
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cb6c1
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cc406
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cc407
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cc43e
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cc43f
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cc440
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cc441
add host initiator host_id=3 initiator_type=FC wwn=21000024ff3cc450
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3cc451
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3cc486
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3cc487
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3cc528
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3cc529
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3e091a
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3e091b
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3f719e
add host initiator host_id=4 initiator_type=FC wwn=21000024ff3f719f
add host initiator host_id=4 initiator_type=FC wwn=21000024ff403916
add host initiator host_id=4 initiator_type=FC wwn=21000024ff403917
add host initiator host_id=4 initiator_type=FC wwn=21000024ff403952
add host initiator host_id=4 initiator_type=FC wwn=21000024ff403953
add host initiator host_id=4 initiator_type=FC wwn=21000024ff40398a
add host initiator host_id=4 initiator_type=FC wwn=21000024ff40398b
add host initiator host_id=4 initiator_type=FC wwn=21000024ff40aba0
add host initiator host_id=4 initiator_type=FC wwn=21000024ff40aba1
add host initiator host_id=4 initiator_type=FC wwn=21000024ff455e92
```

```
add host initiator host_id=4 initiator_type=FC wwn=21000024ff455e93
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49582e
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49582f
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49992c
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49992d
add host initiator host_id=4 initiator_type=FC wwn=21000024ff499aa2
add host initiator host_id=4 initiator_type=FC wwn=21000024ff499aa3
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49af3a
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49af3b
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49af78
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49af79
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49af7a
add host initiator host_id=4 initiator_type=FC wwn=21000024ff49af7b
add host initiator host_id=4 initiator_type=FC wwn=21000024ff4b8194
add host initiator host_id=5 initiator_type=FC wwn=21000024ff4b8195
add host initiator host_id=5 initiator_type=FC wwn=21000024ff4b829c
add host initiator host_id=5 initiator_type=FC wwn=21000024ff4b829d
add host initiator host_id=5 initiator_type=FC wwn=21000024ff4b900c
add host initiator host_id=5 initiator_type=FC wwn=21000024ff4b900d
add host initiator host_id=5 initiator_type=FC wwn=21000024ff5342fa
add host initiator host_id=5 initiator_type=FC wwn=21000024ff5342fb
add host initiator host_id=5 initiator_type=FC wwn=21000024ff536a1c
add host initiator host_id=5 initiator_type=FC wwn=21000024ff536a1d
add host initiator host_id=5 initiator_type=FC wwn=21000024ff536ab4
add host initiator host_id=5 initiator_type=FC wwn=21000024ff536ab5
add host initiator host_id=5 initiator_type=FC wwn=21000024ff536aea
add host initiator host_id=5 initiator_type=FC wwn=21000024ff536aeb
add host initiator host_id=5 initiator_type=FC wwn=21000024ff53b496
add host initiator host_id=5 initiator_type=FC wwn=21000024ff53b497
add host initiator host_id=5 initiator_type=FC wwn=21000024ff53b4d6
add host initiator host_id=5 initiator_type=FC wwn=21000024ff53b4d7
add host initiator host_id=5 initiator_type=FC wwn=21000024ff543a1c
add host initiator host_id=5 initiator_type=FC wwn=21000024ff543a1d
add host initiator host_id=5 initiator_type=FC wwn=21000024ff547009
add host initiator host_id=5 initiator_type=FC wwn=21000024ff54702b
add host initiator host_id=5 initiator_type=FC wwn=21000024ff55c724
add host initiator host_id=5 initiator_type=FC wwn=21000024ff55c725
add host initiator host_id=5 initiator_type=FC wwn=21000024ff55c79a
add host initiator host_id=5 initiator_type=FC wwn=21000024ff55c79b
add host initiator host_id=5 initiator_type=FC wwn=21000024ff5bca1a
add host initiator host_id=5 initiator_type=FC wwn=21000024ff5bca1b
add host initiator host_id=5 initiator_type=FC wwn=21000024ff5c29ce
add host initiator host_id=5 initiator_type=FC wwn=21000024ff5c29cf
add host initiator host_id=5 initiator_type=FC wwn=2101001b322b463e
add host initiator host_id=5 initiator_type=FC wwn=50014380186b22fc
add host initiator host_id=5 initiator_type=FC wwn=50014380186b22fe
```

Step 2: Create Disk Domains, Storage Pools, LUNs

Execute the **mklun.sh** script on one of the Host Systems, which has **expect** installed to complete the following:

- Create 64 disk domains
- Create 64 storage pools
  (one storage pool per disk domain using all available capacity)
- Create 128 LUNs
  (two LUNs per storage pool using all available capacity)
- Add the 128 LUNs to ***lun_group***, ***lg1***

*Note: **Expect** is a Unix automation and testing tool, written by Don Libes as an extension to the Tcl scripting language, for interactive applications such as telnet, ftp, passwd, fsck, rlogin, tip, ssh, and others. It uses Unix pseudo terminals to wrap up subprocesses transparently, allowing the automation of arbitrary applications that are accessed over a terminal. Expect is an open source tool can be downloaded at the following location: http://www.nist.gov/el/msid/expect.cfm*

Step 3: Create Volumes on the Master Host System

Execute the **mkvolume.sh** script on the Master Host System to create 38 logical volumes as follows:

1. **Create Physical Volume**

   Create 128 physical volumes using the **pvcreate** command.

2. **Create Volumes Groups**

   Create one volume group (**vg1**) using the **vgcreate** command and the following 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 /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 /dev/sdbn /dev/sdbo /dev/sdbp /dev/sdbq /dev/sdbr /dev/sdbs /dev/sdbt /dev/sdbu /dev/sdbv /dev/sdbw /dev/sdbx /dev/sdby /dev/sdbz /dev/sdca /dev/sdcb /dev/sdcc /dev/sdcd /dev/sdce /dev/sdcf /dev/sdcg /dev/sdch /dev/sdci /dev/sdcj /dev/sdck /dev/sdcl /dev/sdcm /dev/sdcn /dev/sdco /dev/sdcp /dev/sdcq /dev/sdcr /dev/sdcs /dev/sdct /dev/sdcu /dev/sdcv /dev/sdcw /dev/sdcx /dev/sdcy /dev/sdcz /dev/sdda /dev/sddb /dev/sddc /dev/sddd /dev/sdde /dev/sddf /dev/sddg /dev/sddh /dev/sddi /dev/sddj /dev/sddk /dev/sddl /dev/sddm /dev/sddn /dev/sddo /dev/sddp /dev/sddq /dev/sddr /dev/sdds /dev/sddt /dev/sddu /dev/sddv /dev/sddw /dev/sddx /dev/sddy

3. **Create Logical Volumes**
   - Create 18 logical volumes, each with a capacity of 1,605 GiB, on **vg1** for ASU-1.
   - Create 18 logical volumes, each with a capacity of 1,605 GiB, on **vg1** for ASU-2.
   - Create 2 logical volumes, each with a capacity of 3,210 GiB, on **vg1** for ASU-3.

Step 4: Change the Scheduler on each Host System

Execute the **scheduler.sh** script on each Host System to change the scheduler of each block device from ***cfq*** to ***noop***.

Referenced Scripts

mklun.sh

```
#!/bin/bash

stor=100.148.51.101
stor_user=admin
stor_pswd=Admin@storage1

export LANG=C

echo "creating LUN ..."

expect <<__END_CREATE_LUN
     spawn ssh $stor_user@$stor
     expect {
            -re "assword" { send "$stor_pswd\r" }
            -re "yes/no" { send "yes\r"; exp_continue }
     }
     expect ">"
     foreach EngineId {0 1 2 3} {
             # ------create disk_domain--------
             send "create disk_domain name=ASU\${EngineId}00
disk_list=DAE\${EngineId}00.0-7 disk_domain_id=[expr \$EngineId * 16 + 0]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}01
disk_list=DAE\${EngineId}00.8-15 disk_domain_id=[expr \$EngineId * 16 + 1]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}02
disk_list=DAE\${EngineId}00.16-23 disk_domain_id=[expr \$EngineId * 16 + 2]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}03
disk_list=DAE\${EngineId}10.0-7 disk_domain_id=[expr \$EngineId * 16 + 3]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}04
disk_list=DAE\${EngineId}10.8-15 disk_domain_id=[expr \$EngineId * 16 + 4]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}05
disk_list=DAE\${EngineId}10.16-23 disk_domain_id=[expr \$EngineId * 16 + 5]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}06
disk_list=DAE\${EngineId}24.0-7 disk_domain_id=[expr \$EngineId * 16 + 6]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}07
disk_list=DAE\${EngineId}24.8-15 disk_domain_id=[expr \$EngineId * 16 + 7]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}08
disk_list=DAE\${EngineId}04.0-7 disk_domain_id=[expr \$EngineId * 16 + 8]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}09
disk_list=DAE\${EngineId}04.8-15 disk_domain_id=[expr \$EngineId * 16 + 9]\r"
             expect ">"
             send "create disk_domain name=ASU\${EngineId}10
disk_list=DAE\${EngineId}04.16-23 disk_domain_id=[expr \$EngineId * 16 + 10]\r"
             expect ">"
```

```
              send "create disk_domain name=ASU\${EngineId}11
 disk_list=DAE\${EngineId}14.0-7 disk_domain_id=[expr \$EngineId * 16 + 11]\r"
              expect ">"
              send "create disk_domain name=ASU\${EngineId}12
 disk_list=DAE\${EngineId}14.8-15 disk_domain_id=[expr \$EngineId * 16 + 12]\r"
              expect ">"
              send "create disk_domain name=ASU\${EngineId}13
 disk_list=DAE\${EngineId}14.16-23 disk_domain_id=[expr \$EngineId * 16 + 13]\r"
              expect ">"
              send "create disk_domain name=ASU\${EngineId}14
 disk_list=DAE\${EngineId}20.0-7 disk_domain_id=[expr \$EngineId * 16 + 14]\r"
              expect ">"
              send "create disk_domain name=ASU\${EngineId}15
 disk_list=DAE\${EngineId}20.8-15 disk_domain_id=[expr \$EngineId * 16 + 15]\r"
              expect ">"

              # ------create storage_pool--------
              send "create storage_pool name=ASU\${EngineId}00 disk_type=SSD
 capacity=1229GB pool_id=[expr \$EngineId * 16 + 0] disk_domain_id=[expr \$EngineId *
 16 + 0] raid_level=RAID10\r"
                 expect ">"
                 for { set poolid 1 } { \$poolid <= 9 } { incr poolid } {
                    send "create storage_pool name=ASU\${EngineId}0\${poolid}
 disk_type=SSD capacity=1253GB pool_id=[expr \$EngineId * 16 + \$poolid]
 disk_domain_id=[expr \$EngineId * 16 + \$poolid] raid_level=RAID10\r"
                    expect ">"
              }
              for { set poolid 10 } { \$poolid <= 15 } { incr poolid } {
                    send "create storage_pool name=ASU\${EngineId}\${poolid}
 disk_type=SSD capacity=1253GB pool_id=[expr \$EngineId * 16 + \$poolid]
 disk_domain_id=[expr \$EngineId * 16 + \$poolid] raid_level=RAID10\r"
                    expect ">"
              }

              # ------create lun --------
              send "create lun name=ASU\${EngineId}00 pool_id=[expr \$EngineId * 16 +
 0] capacity=614GB owner_controller=\${EngineId}A\r"
             expect ">"
             send "create lun name=ASU\${EngineId}01 pool_id=[expr \$EngineId * 16 +
 0] capacity=614GB owner_controller=\${EngineId}B\r"
             expect ">"

              for { set lunid 1 } { \$lunid <= 4 } { incr lunid } {
                    send "create lun name=ASU\${EngineId}0[expr \$lunid * 2 + 0]
 pool_id=[expr \$EngineId * 16 + \$lunid] capacity=626GB
 owner_controller=\${EngineId}A\r"
                    expect ">"
                    send "create lun name=ASU\${EngineId}0[expr \$lunid * 2 + 1]
 pool_id=[expr \$EngineId * 16 + \$lunid] capacity=626GB
 owner_controller=\${EngineId}B\r"
                    expect ">"
             }
             for { set lunid 5 } { \$lunid <= 7 } { incr lunid } {
                    send "create lun name=ASU\${EngineId}[expr \$lunid * 2 + 0]
 pool_id=[expr \$EngineId * 16 + \$lunid] capacity=626GB
 owner_controller=\${EngineId}A\r"
                    expect ">"
                    send "create lun name=ASU\${EngineId}[expr \$lunid * 2 + 1]
 pool_id=[expr \$EngineId * 16 + \$lunid] capacity=626GB
 owner_controller=\${EngineId}B\r"
                    expect ">"
             }
             for { set lunid 8 } { \$lunid <= 15 } { incr lunid } {
```

```
                    send "create lun name=ASU\${EngineId}[expr \$lunid * 2 + 0]
pool_id=[expr \$EngineId * 16 + \$lunid] capacity=626GB
owner_controller=\${EngineId}C\r"
                    expect ">"
                    send "create lun name=ASU\${EngineId}[expr \$lunid * 2 + 1]
pool_id=[expr \$EngineId * 16 + \$lunid] capacity=626GB
owner_controller=\${EngineId}D\r"
                    expect ">"
            }

            }
             #add lun to lun_group
             send "add lun_group lun lun_group_id=1
lun_id_list=0,1,16,17,32,33,48,49,64,65,80,81,96,97,112,113,2,3,18,19,34,35,50,51,66
,67,82,83,98,99,114,115\r"
             expect ">"
             send "add lun_group lun lun_group_id=1
lun_id_list=4,5,20,21,36,37,52,53,68,69,84,85,100,101,116,117,6,7,22,23,38,39,54,55,
70,71,86,87,102,103,118,119\r"
             expect ">"
             send "add lun_group lun lun_group_id=1
lun_id_list=8,9,24,25,40,41,56,57,72,73,88,89,104,105,120,121,10,11,26,27,42,43,58,5
9,74,75,90,91,106,107,122,123\r"
             expect ">"
             send "add lun_group lun lun_group_id=1
lun_id_list=12,13,28,29,44,45,60,61,76,77,92,93,108,109,124,125,14,15,30,31,46,47,62
,63,78,79,94,95,110,111,126,127\r"
             expect ">"


     send "exit\r"
     expect "(y/n):"
     send "y\r"
     sleep 5
     send "exit\r"
     expect EOF
__END_CREATE_LUN
```

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
pvcreate /dev/sdbn
pvcreate /dev/sdbo
pvcreate /dev/sdbp
pvcreate /dev/sdbq
pvcreate /dev/sdbr
pvcreate /dev/sdbs
pvcreate /dev/sdbt
pvcreate /dev/sdbu
pvcreate /dev/sdbv
pvcreate /dev/sdbw
pvcreate /dev/sdbx
pvcreate /dev/sdby
pvcreate /dev/sdbz
pvcreate /dev/sdca
pvcreate /dev/sdcb
pvcreate /dev/sdcc
pvcreate /dev/sdcd
pvcreate /dev/sdce
pvcreate /dev/sdcf
pvcreate /dev/sdcg
```

```
pvcreate /dev/sdch
pvcreate /dev/sdci
pvcreate /dev/sdcj
pvcreate /dev/sdck
pvcreate /dev/sdcl
pvcreate /dev/sdcm
pvcreate /dev/sdcn
pvcreate /dev/sdco
pvcreate /dev/sdcp
pvcreate /dev/sdcq
pvcreate /dev/sdcr
pvcreate /dev/sdcs
pvcreate /dev/sdct
pvcreate /dev/sdcu
pvcreate /dev/sdcv
pvcreate /dev/sdcw
pvcreate /dev/sdcx
pvcreate /dev/sdcy
pvcreate /dev/sdcz
pvcreate /dev/sdda
pvcreate /dev/sddb
pvcreate /dev/sddc
pvcreate /dev/sddd
pvcreate /dev/sdde
pvcreate /dev/sddf
pvcreate /dev/sddg
pvcreate /dev/sddh
pvcreate /dev/sddi
pvcreate /dev/sddj
pvcreate /dev/sddk
pvcreate /dev/sddl
pvcreate /dev/sddm
pvcreate /dev/sddn
pvcreate /dev/sddo
pvcreate /dev/sddp
pvcreate /dev/sddq
pvcreate /dev/sddr
pvcreate /dev/sdds
pvcreate /dev/sddt
pvcreate /dev/sddu
pvcreate /dev/sddv
pvcreate /dev/sddw
pvcreate /dev/sddx
pvcreate /dev/sddy


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 /dev/sdbn /dev/sdbo
/dev/sdbp /dev/sdbq /dev/sdbr /dev/sdbs /dev/sdbt /dev/sdbu /dev/sdbv /dev/sdbw
/dev/sdbx /dev/sdby /dev/sdbz /dev/sdca /dev/sdcb /dev/sdcc /dev/sdcd /dev/sdce
/dev/sdcf /dev/sdcg /dev/sdch /dev/sdci /dev/sdcj /dev/sdck /dev/sdcl /dev/sdcm
/dev/sdcn /dev/sdco /dev/sdcp /dev/sdcq /dev/sdcr /dev/sdcs /dev/sdct /dev/sdcu
/dev/sdcv /dev/sdcw /dev/sdcx /dev/sdcy /dev/sdcz /dev/sdda /dev/sddb /dev/sddc
/dev/sddd /dev/sdde /dev/sddf /dev/sddg /dev/sddh /dev/sddi /dev/sddj /dev/sddk
/dev/sddl /dev/sddm /dev/sddn /dev/sddo /dev/sddp /dev/sddq /dev/sddr /dev/sdds
/dev/sddt /dev/sddu /dev/sddv /dev/sddw /dev/sddx /dev/sddy
```

```
lvcreate -n asu101 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu102 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu103 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu104 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu105 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu106 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu107 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu108 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu109 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu110 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu111 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu112 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu113 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu114 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu115 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu116 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu117 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu118 -i 128 -I 512 -C y -L 1605g vg1

lvcreate -n asu201 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu202 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu203 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu204 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu205 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu206 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu207 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu208 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu209 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu210 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu211 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu212 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu213 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu214 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu215 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu216 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu217 -i 128 -I 512 -C y -L 1605g vg1
lvcreate -n asu218 -i 128 -I 512 -C y -L 1605g vg1

lvcreate -n asu301 -i 128 -I 512 -C y -L 3210g vg1
lvcreate -n asu302 -i 128 -I 512 -C y -L 3210g vg1
```

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
echo noop > /sys/block/sdbn/queue/scheduler
echo noop > /sys/block/sdbo/queue/scheduler
echo noop > /sys/block/sdbp/queue/scheduler
echo noop > /sys/block/sdbq/queue/scheduler
echo noop > /sys/block/sdbr/queue/scheduler
echo noop > /sys/block/sdbs/queue/scheduler
echo noop > /sys/block/sdbt/queue/scheduler
echo noop > /sys/block/sdbu/queue/scheduler
echo noop > /sys/block/sdbv/queue/scheduler
echo noop > /sys/block/sdbw/queue/scheduler
echo noop > /sys/block/sdbx/queue/scheduler
echo noop > /sys/block/sdby/queue/scheduler
echo noop > /sys/block/sdbz/queue/scheduler
echo noop > /sys/block/sdca/queue/scheduler
echo noop > /sys/block/sdcb/queue/scheduler
echo noop > /sys/block/sdcc/queue/scheduler
echo noop > /sys/block/sdcd/queue/scheduler
echo noop > /sys/block/sdce/queue/scheduler
```

```
echo noop > /sys/block/sdcf/queue/scheduler
echo noop > /sys/block/sdcg/queue/scheduler
echo noop > /sys/block/sdch/queue/scheduler
echo noop > /sys/block/sdci/queue/scheduler
echo noop > /sys/block/sdcj/queue/scheduler
echo noop > /sys/block/sdck/queue/scheduler
echo noop > /sys/block/sdcl/queue/scheduler
echo noop > /sys/block/sdcm/queue/scheduler
echo noop > /sys/block/sdcn/queue/scheduler
echo noop > /sys/block/sdco/queue/scheduler
echo noop > /sys/block/sdcp/queue/scheduler
echo noop > /sys/block/sdcq/queue/scheduler
echo noop > /sys/block/sdcr/queue/scheduler
echo noop > /sys/block/sdcs/queue/scheduler
echo noop > /sys/block/sdct/queue/scheduler
echo noop > /sys/block/sdcu/queue/scheduler
echo noop > /sys/block/sdcv/queue/scheduler
echo noop > /sys/block/sdcw/queue/scheduler
echo noop > /sys/block/sdcx/queue/scheduler
echo noop > /sys/block/sdcy/queue/scheduler
echo noop > /sys/block/sdcz/queue/scheduler
echo noop > /sys/block/sdda/queue/scheduler
echo noop > /sys/block/sddb/queue/scheduler
echo noop > /sys/block/sddc/queue/scheduler
echo noop > /sys/block/sddd/queue/scheduler
echo noop > /sys/block/sdde/queue/scheduler
echo noop > /sys/block/sddf/queue/scheduler
echo noop > /sys/block/sddg/queue/scheduler
echo noop > /sys/block/sddh/queue/scheduler
echo noop > /sys/block/sddj/queue/scheduler
echo noop > /sys/block/sddi/queue/scheduler
echo noop > /sys/block/sddk/queue/scheduler
echo noop > /sys/block/sddl/queue/scheduler
echo noop > /sys/block/sddn/queue/scheduler
echo noop > /sys/block/sddm/queue/scheduler
echo noop > /sys/block/sddo/queue/scheduler
echo noop > /sys/block/sddp/queue/scheduler
echo noop > /sys/block/sddq/queue/scheduler
echo noop > /sys/block/sddr/queue/scheduler
echo noop > /sys/block/sdds/queue/scheduler
echo noop > /sys/block/sddu/queue/scheduler
echo noop > /sys/block/sddt/queue/scheduler
echo noop > /sys/block/sddw/queue/scheduler
echo noop > /sys/block/sddx/queue/scheduler
echo noop > /sys/block/sddv/queue/scheduler
echo noop > /sys/block/sddy/queue/scheduler
```

APPENDIX D: SPC-1 WORKLOAD GENERATOR STORAGE COMMANDS AND PARAMETERS

ASU Pre-Fill

The content of command and parameter file, used in this benchmark to execute the required ASU pre-fill, is listed below.

```
hd=default,vdbench=/root/vdbench,user=root,shell=ssh

hd=hd1,system=host1
hd=hd2,system=host2
hd=hd3,system=host3
hd=hd4,system=host5

sd=default,openflags=o_direct,threads=8

sd=sd1,host=hd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,host=hd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,host=hd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,host=hd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,host=hd1,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,host=hd2,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,host=hd3,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,host=hd4,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,host=hd1,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,host=hd2,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,host=hd3,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,host=hd4,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,host=hd1,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,host=hd2,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,host=hd3,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,host=hd4,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,host=hd1,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,host=hd2,lun=/dev/vg1/asu118,size=1723355627520

sd=sd19,host=hd3,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,host=hd4,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,host=hd1,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,host=hd2,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,host=hd3,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,host=hd4,lun=/dev/vg1/asu206,size=1723355627520
sd=sd25,host=hd1,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,host=hd2,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,host=hd3,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,host=hd4,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,host=hd1,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,host=hd2,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,host=hd3,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,host=hd4,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,host=hd1,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,host=hd2,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,host=hd3,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,host=hd4,lun=/dev/vg1/asu218,size=1723355627520

sd=sd37,host=hd1,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,host=hd2,lun=/dev/vg1/asu302,size=3446711255040

wd=wd1,sd=sd*,rdpct=0,seekpct=-1,xfersize=1024K
rd=PREPASU1,wd=wd1,iorate=max,elapsed=3600000,interval=10
```

Common Command Lines – Primary Metrics and Repeatability Tests

The following command lines appear at the beginning of each command and parameter file for the Primary Metrics and Repeatability Tests. The command lines are only listed below to eliminate redundancy.

```
host=master
slaves=(slave1,slave2,slave3,slave4,slave5,slave6,slave7,slave8,slave9,slave10,slave
11,slave12,slave13,slave14,slave15,slave16,slave17,slave18,slave19,slave20,slave21,s
lave22,slave23,slave24,slave25,slave26,slave27,slave28,slave29,slave30,slave31,slave
32,slave33,slave34,slave35,slave36,slave37,slave38,slave39,slave40,slave41,slave42,s
lave43,slave44,slave45,slave46,slave47,slave48,slave49,slave50,slave51,slave52,slave
53,slave54,slave55,slave56,slave57,slave58,slave59,slave60,slave61,slave62,slave63,s
lave64,slave65,slave66,slave67,slave68,slave69,slave70,slave71,slave72,slave73,slave
74,slave75,slave76,slave77,slave78,slave79,slave80,slave81,slave82,slave83,slave84,s
lave85,slave86,slave87,slave88,slave89,slave90,slave91,slave92,slave93,slave94,slave
95,slave96,slave97,slave98,slave99,slave100,slave101,slave102,slave103,slave104,slav
e105,slave106,slave107,slave108,slave109,slave110,slave111,slave112,slave113,slave11
4,slave115,slave116,slave117,slave118,slave119,slave120,slave121,slave122,slave123,s
lave124,slave125,slave126,slave127,slave128,slave129,slave130,slave131,slave132,slav
e133,slave134,slave135,slave136,slave137,slave138,slave139,slave140,slave141,slave14
2,slave143,slave144,slave145,slave146,slave147,slave148,slave149,slave150,slave151,s
lave152,slave153,slave154,slave155,slave156,slave157,slave158,slave159,slave160,slav
e161,slave162,slave163,slave164,slave165,slave166,slave167,slave168,slave169,slave17
0,slave171,slave172,slave173,slave174,slave175,slave176,slave177,slave178,slave179,s
lave180,slave181,slave182,slave183,slave184,slave185,slave186,slave187,slave188,slav
e189,slave190,slave191,slave192,slave193,slave194,slave195,slave196,slave197,slave19
8,slave199,slave200,slave201,slave202,slave203,slave204,slave205,slave206,slave207,s
lave208,slave209,slave210,slave211,slave212,slave213,slave214,slave215,slave216,slav
e217,slave218,slave219,slave220,slave221,slave222,slave223,slave224,slave225,slave22
6,slave227,slave228,slave229,slave230,slave231,slave232,slave233,slave234,slave235,s
lave236,slave237,slave238,slave239,slave240,slave241,slave242,slave243,slave244,slav
e245,slave246,slave247,slave248,slave249,slave250,slave251,slave252,slave253,slave25
4,slave255,slave256,slave257,slave258,slave259,slave260,slave261,slave262,slave263,s
lave264,slave265,slave266,slave267,slave268,slave269,slave270,slave271,slave272,slav
e273,slave274,slave275,slave276,slave277,slave278,slave279,slave280,slave281,slave28
2,slave283,slave284,slave285,slave286,slave287,slave288,slave289,slave290,slave291,s
lave292,slave293,slave294,slave295,slave296,slave297,slave298,slave299,slave300,slav
e301,slave302,slave303,slave304,slave305,slave306,slave307,slave308,slave309,slave31
0,slave311,slave312,slave313,slave314,slave315,slave316,slave317,slave318,slave319,s
lave320,slave321,slave322,slave323,slave324,slave325,slave326,slave327,slave328,slav
e329,slave330,slave331,slave332,slave333,slave334,slave335,slave336,slave337,slave33
8,slave339,slave340,slave341,slave342,slave343,slave344,slave345,slave346,slave347,s
lave348,slave349,slave350,slave351,slave352,slave353,slave354,slave355,slave356,slav
e357,slave358,slave359,slave360,slave361,slave362,slave363,slave364,slave365,slave36
6,slave367,slave368,slave369,slave370,slave371,slave372,slave373,slave374,slave375,s
lave376,slave377,slave378,slave379,slave380,slave381,slave382,slave383,slave384,slav
e385,slave386,slave387,slave388,slave389,slave390,slave391,slave392,slave393,slave39
4,slave395,slave396,slave397,slave398,slave399,slave400,slave401,slave402,slave403,s
lave404,slave405,slave406,slave407,slave408,slave409,slave410,slave411,slave412,slav
e413,slave414,slave415,slave416,slave417,slave418,slave419,slave420,slave421,slave42
2,slave423,slave424,slave425,slave426,slave427,slave428,slave429,slave430,slave431,s
lave432,slave433,slave434,slave435,slave436,slave437,slave438,slave439,slave440,slav
e441,slave442,slave443,slave444,slave445,slave446,slave447,slave448,slave449,slave45
0,slave451,slave452,slave453,slave454,slave455,slave456,slave457,slave458,slave459,s
lave460,slave461,slave462,slave463,slave464,slave465,slave466,slave467,slave468,slav
e469,slave470,slave471,slave472,slave473,slave474,slave475,slave476,slave477,slave47
8,slave479,slave480,slave481,slave482,slave483,slave484,slave485,slave486,slave487,s
lave488,slave489,slave490,slave491,slave492,slave493,slave494,slave495,slave496,slav
e497,slave498,slave499,slave500,slave501,slave502,slave503,slave504,slave505,slave50
6,slave507,slave508,slave509,slave510,slave511,slave512,slave513,slave514,slave515,s
lave516,slave517,slave518,slave519,slave520,slave521,slave522,slave523,slave524,slav
```

```
e525,slave526,slave527,slave528,slave529,slave530,slave531,slave532,slave533,slave53
4,slave535,slave536,slave537,slave538,slave539,slave540,slave541,slave542,slave543,s
lave544,slave545,slave546,slave547,slave548,slave549,slave550,slave551,slave552,slav
e553,slave554,slave555,slave556,slave557,slave558,slave559,slave560,slave561,slave56
2,slave563,slave564,slave565,slave566,slave567,slave568,slave569,slave570,slave571,s
lave572,slave573,slave574,slave575,slave576,slave577,slave578,slave579,slave580,slav
e581,slave582,slave583,slave584,slave585,slave586,slave587,slave588,slave589,slave59
0,slave591,slave592,slave593,slave594,slave595,slave596,slave597,slave598,slave599,s
lave600,slave601,slave602,slave603,slave604,slave605,slave606,slave607,slave608,slav
e609,slave610,slave611,slave612,slave613,slave614,slave615,slave616,slave617,slave61
8,slave619,slave620,slave621,slave622,slave623,slave624,slave625,slave626,slave627,s
lave628,slave629,slave630,slave631,slave632,slave633,slave634,slave635,slave636,slav
e637,slave638,slave639,slave640,slave641,slave642,slave643,slave644,slave645,slave64
6,slave647,slave648,slave649,slave650)

sd=asu1_1,lun=/dev/vg1/asu101,size=1723355627520
sd=asu1_2,lun=/dev/vg1/asu102,size=1723355627520
sd=asu1_3,lun=/dev/vg1/asu103,size=1723355627520
sd=asu1_4,lun=/dev/vg1/asu104,size=1723355627520
sd=asu1_5,lun=/dev/vg1/asu105,size=1723355627520
sd=asu1_6,lun=/dev/vg1/asu106,size=1723355627520
sd=asu1_7,lun=/dev/vg1/asu107,size=1723355627520
sd=asu1_8,lun=/dev/vg1/asu108,size=1723355627520
sd=asu1_9,lun=/dev/vg1/asu109,size=1723355627520
sd=asu1_10,lun=/dev/vg1/asu110,size=1723355627520
sd=asu1_11,lun=/dev/vg1/asu111,size=1723355627520
sd=asu1_12,lun=/dev/vg1/asu112,size=1723355627520
sd=asu1_13,lun=/dev/vg1/asu113,size=1723355627520
sd=asu1_14,lun=/dev/vg1/asu114,size=1723355627520
sd=asu1_15,lun=/dev/vg1/asu115,size=1723355627520
sd=asu1_16,lun=/dev/vg1/asu116,size=1723355627520
sd=asu1_17,lun=/dev/vg1/asu117,size=1723355627520
sd=asu1_18,lun=/dev/vg1/asu118,size=1723355627520

sd=asu2_1,lun=/dev/vg1/asu201,size=1723355627520
sd=asu2_2,lun=/dev/vg1/asu202,size=1723355627520
sd=asu2_3,lun=/dev/vg1/asu203,size=1723355627520
sd=asu2_4,lun=/dev/vg1/asu204,size=1723355627520
sd=asu2_5,lun=/dev/vg1/asu205,size=1723355627520
sd=asu2_6,lun=/dev/vg1/asu206,size=1723355627520
sd=asu2_7,lun=/dev/vg1/asu207,size=1723355627520
sd=asu2_8,lun=/dev/vg1/asu208,size=1723355627520
sd=asu2_9,lun=/dev/vg1/asu209,size=1723355627520
sd=asu2_10,lun=/dev/vg1/asu210,size=1723355627520
sd=asu2_11,lun=/dev/vg1/asu211,size=1723355627520
sd=asu2_12,lun=/dev/vg1/asu212,size=1723355627520
sd=asu2_13,lun=/dev/vg1/asu213,size=1723355627520
sd=asu2_14,lun=/dev/vg1/asu214,size=1723355627520
sd=asu2_15,lun=/dev/vg1/asu215,size=1723355627520
sd=asu2_16,lun=/dev/vg1/asu216,size=1723355627520
sd=asu2_17,lun=/dev/vg1/asu217,size=1723355627520
sd=asu2_18,lun=/dev/vg1/asu218,size=1723355627520

sd=asu3_1,lun=/dev/vg1/asu301,size=3446711255040
sd=asu3_2,lun=/dev/vg1/asu302,size=3446711255040
```

Primary Metrics and Repeatability Tests

The content of SPC-1 Workload Generator command and parameter files used in this benchmark to execute the Primary Metrics *(Sustainability Test Phase, IOPS Test Phase, and Response Time Ramp Test Phase)* and Repeatability *(Repeatability Test Phase 1 and Repeatability Test Phase 2)* Tests are listed below.

Primary Metrics Test: Sustainability Test Phase/Test Run

```
common commands_1
rd=ramp_sust,bsus=60200,startup=180,elapsed=28800,interval=60
```

Primary Metrics Test: IOPS Test Phase (100% Test Run)

```
common commands_1
rd=ramp_100,bsus=60200,startup=180,elapsed=600,interval=60
```

Primary Metrics Test: Response Time Ramp Test Phase *(95% Test Run)*

```
common commands_1
rd=ramp_95,bsus=57190,startup=180,elapsed=600,interval=60
```

Primary Metrics Test: Response Time Ramp Test Phase *(90% Test Run)*

```
common commands_1
rd=ramp_90,bsus=54180,startup=180,elapsed=600,interval=60
```

Primary Metrics Test: Response Time Ramp Test Phase *(80% Test Run)*

```
common commands_1
rd=ramp_80,bsus=48160,startup=180,elapsed=600,interval=60
```

Primary Metrics Test: Response Time Ramp Test Phase *(50% Test Run)*

```
common commands_1
rd=ramp_50,bsus=30100,startup=180,elapsed=600,interval=60
```

Repeatability Test: Repeatability Test Phase 1 *(10% Test Run)*

```
common commands_1
rd=repeat1_lrt,bsus=6020,startup=180,elapsed=600,interval=60
```

Repeatability Test: Repeatability Test Phase 1 *(100% Test Run)*

```
common commands_1
rd=repeat1_iops,bsus=60200,startup=180,elapsed=600,interval=60
```

Repeatability Test: Repeatability Test Phase 2 *(10% Test Run)*

```
common commands_1
rd=repeat2_lrt,bsus=6020,startup=180,elapsed=600,interval=60
```

Repeatability Test: Repeatability Test Phase 2 *(100% Test Run)*

```
common commands_1
rd=repeat2_iops,bsus=60200,startup=180,elapsed=600,interval=60
```

SPC-1 Persistence Test Run 1

The content of SPC-1 Workload Generator command and parameter file, used in this benchmark to execute a reduced level SPC-1 Persistence Test Run 1, is listed below.

```
sd=asu1_1,lun=/dev/vg1/asu101,size=1723355627520
sd=asu1_2,lun=/dev/vg1/asu102,size=1723355627520
sd=asu1_3,lun=/dev/vg1/asu103,size=1723355627520
sd=asu1_4,lun=/dev/vg1/asu104,size=1723355627520
sd=asu1_5,lun=/dev/vg1/asu105,size=1723355627520
sd=asu1_6,lun=/dev/vg1/asu106,size=1723355627520
sd=asu1_7,lun=/dev/vg1/asu107,size=1723355627520
sd=asu1_8,lun=/dev/vg1/asu108,size=1723355627520
sd=asu1_9,lun=/dev/vg1/asu109,size=1723355627520
sd=asu1_10,lun=/dev/vg1/asu110,size=1723355627520
sd=asu1_11,lun=/dev/vg1/asu111,size=1723355627520
sd=asu1_12,lun=/dev/vg1/asu112,size=1723355627520
sd=asu1_13,lun=/dev/vg1/asu113,size=1723355627520
sd=asu1_14,lun=/dev/vg1/asu114,size=1723355627520
sd=asu1_15,lun=/dev/vg1/asu115,size=1723355627520
sd=asu1_16,lun=/dev/vg1/asu116,size=1723355627520
sd=asu1_17,lun=/dev/vg1/asu117,size=1723355627520
sd=asu1_18,lun=/dev/vg1/asu118,size=1723355627520

sd=asu2_1,lun=/dev/vg1/asu201,size=1723355627520
sd=asu2_2,lun=/dev/vg1/asu202,size=1723355627520
sd=asu2_3,lun=/dev/vg1/asu203,size=1723355627520
sd=asu2_4,lun=/dev/vg1/asu204,size=1723355627520
sd=asu2_5,lun=/dev/vg1/asu205,size=1723355627520
sd=asu2_6,lun=/dev/vg1/asu206,size=1723355627520
sd=asu2_7,lun=/dev/vg1/asu207,size=1723355627520
sd=asu2_8,lun=/dev/vg1/asu208,size=1723355627520
sd=asu2_9,lun=/dev/vg1/asu209,size=1723355627520
sd=asu2_10,lun=/dev/vg1/asu210,size=1723355627520
sd=asu2_11,lun=/dev/vg1/asu211,size=1723355627520
sd=asu2_12,lun=/dev/vg1/asu212,size=1723355627520
sd=asu2_13,lun=/dev/vg1/asu213,size=1723355627520
sd=asu2_14,lun=/dev/vg1/asu214,size=1723355627520
sd=asu2_15,lun=/dev/vg1/asu215,size=1723355627520
sd=asu2_16,lun=/dev/vg1/asu216,size=1723355627520
sd=asu2_17,lun=/dev/vg1/asu217,size=1723355627520
sd=asu2_18,lun=/dev/vg1/asu218,size=1723355627520

sd=asu3_1,lun=/dev/vg1/asu301,size=3446711255040
sd=asu3_2,lun=/dev/vg1/asu302,size=3446711255040
```

SPC-2 Persistence Test

Use of the SPC-2 Persistence Test to meet the SPC-1 persistence requirements was approved by the SPC Auditor. Both the SPC-1 and SPC-2 Persistence Tests provide the same level of functionality and verification of data integrity.

Common Command Lines – SPC-2 Persistence Test

The following command lines appear at the beginning of each command and parameter file for the two SPC-2 Persistence Test Runs. The command lines are only listed below to eliminate redundancy.

```
host=localhost,java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -Xms4096m -
Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.2,coltrane1),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.3,coltrane2),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.4,coltrane3),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.5,coltrane4),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.6,coltrane5),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.8,coltrane6),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.9,coltrane7),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.10,coltrane8),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3
host=(10.1.1.11,coltrane9),java=("/usr/java/jre1.6.0_45/bin/java","-d64 -Xmx4096m -
Xms4096m -Xmn1024m -Xss192k -Xincgc"),spc2="/root/spc2",shell=spc2,jvms=3

sd=default,host=localhost
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520

sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
```

```
sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040



sd=default,host=coltrane1
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520

sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040



sd=default,host=coltrane2
```

```
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520

sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040



sd=default,host=coltrane3
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520
```

```
sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,lun=/dev/vg1/asu218,size=1723355627520
sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040

sd=default,host=coltrane4
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520

sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040
```

```
sd=default,host=coltrane5
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520

sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040


sd=default,host=coltrane6
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520
```

```
  sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
  sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
  sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
  sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
  sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
  sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
  sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
  sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
  sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
  sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
  sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
  sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
  sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
  sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
  sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
  sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
  sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
  sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

  sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
  sd=sd38,lun=/dev/vg1/asu302,size=3446711255040


  sd=default,host=coltrane7
  sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
  sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
  sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
  sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
  sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
  sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
  sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
  sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
  sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
  sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
  sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
  sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
  sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
  sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
  sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
  sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
  sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
  sd=sd18,lun=/dev/vg1/asu118,size=1723355627520

  sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
  sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
  sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
  sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
  sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
  sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
  sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
  sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
  sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
  sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
  sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
  sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
  sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
  sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
  sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
  sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
  sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
  sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

  sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
```

```
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040


sd=default,host=coltrane8
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520

sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040



sd=default,host=coltrane9
sd=sd1,lun=/dev/vg1/asu101,size=1723355627520
sd=sd2,lun=/dev/vg1/asu102,size=1723355627520
sd=sd3,lun=/dev/vg1/asu103,size=1723355627520
sd=sd4,lun=/dev/vg1/asu104,size=1723355627520
sd=sd5,lun=/dev/vg1/asu105,size=1723355627520
sd=sd6,lun=/dev/vg1/asu106,size=1723355627520
sd=sd7,lun=/dev/vg1/asu107,size=1723355627520
sd=sd8,lun=/dev/vg1/asu108,size=1723355627520
sd=sd9,lun=/dev/vg1/asu109,size=1723355627520
sd=sd10,lun=/dev/vg1/asu110,size=1723355627520
sd=sd11,lun=/dev/vg1/asu111,size=1723355627520
sd=sd12,lun=/dev/vg1/asu112,size=1723355627520
sd=sd13,lun=/dev/vg1/asu113,size=1723355627520
sd=sd14,lun=/dev/vg1/asu114,size=1723355627520
sd=sd15,lun=/dev/vg1/asu115,size=1723355627520
```

```
sd=sd16,lun=/dev/vg1/asu116,size=1723355627520
sd=sd17,lun=/dev/vg1/asu117,size=1723355627520
sd=sd18,lun=/dev/vg1/asu118,size=1723355627520

sd=sd19,lun=/dev/vg1/asu201,size=1723355627520
sd=sd20,lun=/dev/vg1/asu202,size=1723355627520
sd=sd21,lun=/dev/vg1/asu203,size=1723355627520
sd=sd22,lun=/dev/vg1/asu204,size=1723355627520
sd=sd23,lun=/dev/vg1/asu205,size=1723355627520
sd=sd24,lun=/dev/vg1/asu206,size=1723355627520
sd=sd25,lun=/dev/vg1/asu207,size=1723355627520
sd=sd26,lun=/dev/vg1/asu208,size=1723355627520
sd=sd27,lun=/dev/vg1/asu209,size=1723355627520
sd=sd28,lun=/dev/vg1/asu210,size=1723355627520
sd=sd29,lun=/dev/vg1/asu211,size=1723355627520
sd=sd30,lun=/dev/vg1/asu212,size=1723355627520
sd=sd31,lun=/dev/vg1/asu213,size=1723355627520
sd=sd32,lun=/dev/vg1/asu214,size=1723355627520
sd=sd33,lun=/dev/vg1/asu215,size=1723355627520
sd=sd34,lun=/dev/vg1/asu216,size=1723355627520
sd=sd35,lun=/dev/vg1/asu217,size=1723355627520
sd=sd36,lun=/dev/vg1/asu218,size=1723355627520

sd=sd37,lun=/dev/vg1/asu301,size=3446711255040
sd=sd38,lun=/dev/vg1/asu302,size=3446711255040


maxlatestart=1
reportinginterval=5
segmentlength=512m
```

SPC-2 Persistence Test Run 1 (write phase)

```
common commands_2
rd=default,rampup=360,periods=180,measurement=300,runout=0,rampdown=0,buffers=1

rd=default,rdpct=0,xfersize=1024k
rd=TR1-101s_SPC-2-persist-w,streams=2007
```

SPC-2 Persistence Test Run 2 (read phase)

```
common commands_2
maxpersistenceerrors=10

rd=default,buffers=1,rdpct=100,xfersize=1024k
rd=TR1-5s_SPC-2-persist-r
```

Slave JVMs

Each Slave JVM was invoked with a command and parameter file similar to the example listed below. The only difference in each file was **host** parameter value, which was unique to each Slave JVM, e.g. **slave1**…**slave650**.

```
master=host1
host=slave1
sd=asu1_1,lun=/dev/vg1/asu101,size=1723355627520
```

```
sd=asu1_2,lun=/dev/vg1/asu102,size=1723355627520
sd=asu1_3,lun=/dev/vg1/asu103,size=1723355627520
sd=asu1_4,lun=/dev/vg1/asu104,size=1723355627520
sd=asu1_5,lun=/dev/vg1/asu105,size=1723355627520
sd=asu1_6,lun=/dev/vg1/asu106,size=1723355627520
sd=asu1_7,lun=/dev/vg1/asu107,size=1723355627520
sd=asu1_8,lun=/dev/vg1/asu108,size=1723355627520
sd=asu1_9,lun=/dev/vg1/asu109,size=1723355627520
sd=asu1_10,lun=/dev/vg1/asu110,size=1723355627520
sd=asu1_11,lun=/dev/vg1/asu111,size=1723355627520
sd=asu1_12,lun=/dev/vg1/asu112,size=1723355627520
sd=asu1_13,lun=/dev/vg1/asu113,size=1723355627520
sd=asu1_14,lun=/dev/vg1/asu114,size=1723355627520
sd=asu1_15,lun=/dev/vg1/asu115,size=1723355627520
sd=asu1_16,lun=/dev/vg1/asu116,size=1723355627520
sd=asu1_17,lun=/dev/vg1/asu117,size=1723355627520
sd=asu1_18,lun=/dev/vg1/asu118,size=1723355627520
sd=asu2_1,lun=/dev/vg1/asu201,size=1723355627520
sd=asu2_2,lun=/dev/vg1/asu202,size=1723355627520
sd=asu2_3,lun=/dev/vg1/asu203,size=1723355627520
sd=asu2_4,lun=/dev/vg1/asu204,size=1723355627520
sd=asu2_5,lun=/dev/vg1/asu205,size=1723355627520
sd=asu2_6,lun=/dev/vg1/asu206,size=1723355627520
sd=asu2_7,lun=/dev/vg1/asu207,size=1723355627520
sd=asu2_8,lun=/dev/vg1/asu208,size=1723355627520
sd=asu2_9,lun=/dev/vg1/asu209,size=1723355627520
sd=asu2_10,lun=/dev/vg1/asu210,size=1723355627520
sd=asu2_11,lun=/dev/vg1/asu211,size=1723355627520
sd=asu2_12,lun=/dev/vg1/asu212,size=1723355627520
sd=asu2_13,lun=/dev/vg1/asu213,size=1723355627520
sd=asu2_14,lun=/dev/vg1/asu214,size=1723355627520
sd=asu2_15,lun=/dev/vg1/asu215,size=1723355627520
sd=asu2_16,lun=/dev/vg1/asu216,size=1723355627520
sd=asu2_17,lun=/dev/vg1/asu217,size=1723355627520
sd=asu2_18,lun=/dev/vg1/asu218,size=1723355627520
sd=asu3_1,lun=/dev/vg1/asu301,size=3446711255040
sd=asu3_2,lun=/dev/vg1/asu302,size=3446711255040
```

APPENDIX E: SPC-1 WORKLOAD GENERATOR INPUT PARAMETERS

The following script, **run.sh**, was invoked to execute the required ASU pre-fill, start the required number of Slave JVMs, execute the Primary Metrics Test *(Sustainability Test Phase, IOPS Test Phase, and Response Time Ramp Test Phase)*, Repeatability Test *(Repeatability Test Phase 1 and Repeatability Test Phase 2)*, a reduced BSU level SPC-1 Persistence Test Run 1 *(write phase)* and SPC-2 Persistence Test Run 1 *(write phase)* in an uninterrupted sequence.

After the above test sequence completed, the script pauses until the required TSC power off/power on cycle is completed then executes SPC-2 Persistence Test Run 2 *(read phase)*.

The **run.sh** script also included the appropriate commands to capture the detailed TSC profile listings required for a Remote Audit.

run.sh

```
#!/bin/sh

#JAVA="/usr/java/jre1.6.0_45/bin/java -d64 -Xms7168m -Xmx7168m -Xmn1792m -Xss192k -
Xincgc"
JAVA="/usr/java/jre1.6.0_45/bin/java -Xmx7168m -Xincgc"
EXEDIR=/root/300w

expect shstorage.tcl > profile1_storage.log
date > profile1_volume.log
lvdisplay >> profile1_volume.log
date >> profile1_volume.log

echo "ASU prefill started......"
../vdbench/vdbench -f /root/300w/prefilling.cfg -o /root/300w/PreFill
echo "ASU prefill complete....."

for test in ramp_sust ramp_100 ramp_95 ramp_90 ramp_80 ramp_50 ramp_10 repeat1_lrt
repeat1_iops repeat2_lrt repeat2_iops
do
     sh slave.sh
     $JAVA -cp ../spc1 spc1 -w SPC1 -f $test.cfg -o $test SPCOut
     for host in host1 host2 host3 host4 host5 host6 host11 host8 host9 host10
     do
            ssh $host mv $EXEDIR/output $EXEDIR/output_$test
            ssh $host mv $EXEDIR/config $EXEDIR/config_$test
     done
     sh killjava.sh
done

rm -rf spc1.cfg
cp persist.cfg spc1.cfg

$JAVA -cp ../spc1 persist1 -b 6020
sh killjava.sh

for host in host2 host3 host4 host5 host6 host8 host9 host10 host11
do
    ssh $host $JAVA -cp $EXEDIR/../spc2 RemoteStart &
done

sh /root/spc2/run-spc2-persist1.sh
```

```
echo "Power cycle TSC, then Enter to continue"
read

expect shstorage.tcl > profile2_storage.log
date > profile2_volume.log
lvdisplay >> profile2_volume.log
date >> profile2_volume.log

sh /root/spc2/run-spc2-persist2.sh
```

slave.sh

This script was invoked from **run.sh** to start the Slave JVMs prior to each Test Run.

```
#!/bin/sh

#JAVA="/usr/java/jre1.6.0_45/bin/java -d64 -Xms7168m -Xmx7168m -Xmn1792m -Xss192k -
Xincgc"
JAVA="/usr/java/jre1.6.0_45/bin/java -Xmx7168m -Xincgc"
EXEDIR=/root/300w

N=1
for host in host1 host2 host3 host4 host5 host6 host11 host8 host9 host10
do
  ssh $host rm -rf $EXEDIR/output
  ssh $host rm -rf $EXEDIR/config
  ssh $host mkdir $EXEDIR/output
  ssh $host mkdir $EXEDIR/config
  for((i=1;i<=65;i++))
  do
    echo "start slave$N on $host"
    echo "master=host1" > $EXEDIR/config/slave$N.cfg
    echo "host=slave$N" >> $EXEDIR/config/slave$N.cfg

echo "sd=asu1_1,lun=/dev/vg1/asu101,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_2,lun=/dev/vg1/asu102,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_3,lun=/dev/vg1/asu103,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_4,lun=/dev/vg1/asu104,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_5,lun=/dev/vg1/asu105,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_6,lun=/dev/vg1/asu106,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_7,lun=/dev/vg1/asu107,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_8,lun=/dev/vg1/asu108,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_9,lun=/dev/vg1/asu109,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_10,lun=/dev/vg1/asu110,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_11,lun=/dev/vg1/asu111,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_12,lun=/dev/vg1/asu112,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
echo "sd=asu1_13,lun=/dev/vg1/asu113,size=1723355627520" >>
$EXEDIR/config/slave$N.cfg
```

```
  echo "sd=asu1_14,lun=/dev/vg1/asu114,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu1_15,lun=/dev/vg1/asu115,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu1_16,lun=/dev/vg1/asu116,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu1_17,lun=/dev/vg1/asu117,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu1_18,lun=/dev/vg1/asu118,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg

  echo "sd=asu2_1,lun=/dev/vg1/asu201,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_2,lun=/dev/vg1/asu202,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_3,lun=/dev/vg1/asu203,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_4,lun=/dev/vg1/asu204,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_5,lun=/dev/vg1/asu205,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_6,lun=/dev/vg1/asu206,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_7,lun=/dev/vg1/asu207,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_8,lun=/dev/vg1/asu208,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_9,lun=/dev/vg1/asu209,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_10,lun=/dev/vg1/asu210,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_11,lun=/dev/vg1/asu211,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_12,lun=/dev/vg1/asu212,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_13,lun=/dev/vg1/asu213,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_14,lun=/dev/vg1/asu214,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_15,lun=/dev/vg1/asu215,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_16,lun=/dev/vg1/asu216,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_17,lun=/dev/vg1/asu217,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu2_18,lun=/dev/vg1/asu218,size=1723355627520" >>
  $EXEDIR/config/slave$N.cfg

  echo "sd=asu3_1,lun=/dev/vg1/asu301,size=3446711255040" >>
  $EXEDIR/config/slave$N.cfg
  echo "sd=asu3_2,lun=/dev/vg1/asu302,size=3446711255040" >>
  $EXEDIR/config/slave$N.cfg

  scp $EXEDIR/config/slave$N.cfg $host:$EXEDIR/config/slave$N.cfg
      ssh $host "$JAVA -cp $EXEDIR/../spc1 spc1 -f $EXEDIR/config/slave$N.cfg -o
  $EXEDIR/output/slave$N" > /dev/null &
      N=$[N+1]
    done
  done
```

killjava.sh

This script wsa invoked from **run.sh** to terminate all Slave JVMs at the end of each Test Run.

```
!/bin/sh

for host in host1 host2 host3 host4 host5 host6 host11 host8 host9 host10
do
  ssh $host pkill -9 java
done

sleep 60
```

APPENDIX F: THIRD-PARTY QUOTATION

Priced Storage Configuration

Netfast Technology Solutions, Inc.
989, Avenues of America, Fl 12
New York, NY 10018, USA
Voice: (212) 792-5200 , Fax: (212) 213-1152

11/13/2015, Quote Valid:90 Days

No.	Model	Description	Qty.	Unit Price (USD)	Total Price (USD)
1	Phase				
1.1	Location				
1.1.1	OceanStor 18800 V3 Storage System				
1.1.1	Engine				
	88V3-4C1T-AC	OceanStor 18800 V3 Engine(Four Controller,AC\240HVDC,1TB Cache,SPE72C0600)	4	67,081.80	268,327.20
1.1.2	Expand Interface Module				
	LPU5PCIEV3H	2 port PCIe I/O module	16	811.06	12,976.96
	SMARTIO8FCV3H	4 port SmartIO I/O module(SFP+,8Gb FC)	48	972.17	46,664.16
	LPU12S12V3H	12 port 12Gb SAS Entire Sharing I/O module(MiniSAS HD)	16	3,962.48	63,399.68
1.1.3	Disk Components				
	SSD400-2-H	400GB SSD eMLC SAS Disk Unit(2.5")	512	3,174.65	1,625,420.80
1.1.4	Disk Enclosure				
	DAE22525U2-H-AC	Disk Enclosure(2U,2.5",AC\240HVDC,DAE22525U2)	24	1,757.80	42,187.20
1.1.5	Cabinet				
	RACK-SYS-H-AC	OceanStor 18000 V3 Series System Cabinet	2	6,422.33	12,844.66
1.1.6	Option Class Item				
	SVP4-V3H	Service Processor (1U, AC\240HDVC,8GB Cache,Including Windows OS Software and Security software)	1	3,760.39	3,760.39
	WM1P0CIKVM02	KVM,KVM 4 in 1 Control Module,1U, 17" LED, 8 KVM ports, With Power Cable,8 USB Straight signal cables/With mounting Accessories,English doc,110V/240V AC,Black,Compliant	1	585.00	585.00
	SWITCH-V3H	PCIe Switch(AC\240HVDC,2GB Cache,16 Port,SWE1603P05)	2	1,587.47	3,174.94
	OQSFPOM00	Quadwire 40 Gb/s Parallel AOC	32	1,494.00	47,808.00
	PDU2000-V3-H	AC Power Distribution Unit	8	127.00	1,016.00
	HS-SAS-1-01	High Speed Cable,External MiniSAS HD Cable,1m,(SFF 8644 Plug),(28AWG*4P*2B(S)),(SFF 8644 Plug),Indoor use	32	55.00	1,760.00
	HS-SAS-3-01	High Speed Cable,Mini SAS HD Cable,3m,(SFF 8644 Plug),(28AWG*4P*2B(S)),(SFF 8644 Plug),Indoor use	16	96.00	1,536.00
1.1.7	HBA				
	N8GHBA000	QLOGIC QLE2562 HBA Card,PCIE,8Gbps DualPort ,Fiber Channel Multimode LC Optic Interface,English Manual, No Drive CD	80	1,000.00	80,000.00

Priced Storage Configuration *(continued)*

Netfast Technology Solutions, Inc.
989, Avenues of America, Fl 12
New York, NY 10018, USA
Voice: (212) 792-5200 , Fax: (212) 213-1152

11/13/2015, Quote Valid:90 Days

No.	Model	Description	Qty.	Unit Price (USD)	Total Price (USD)
1.1.8	Accessory				
	VADMSMR02	Software Service,Trend Micro,05280169,Original Manufacturer Service, 1 Year,7*24 Standard Service,10 users	2	314.00	628.00
	SN2F01FCPC	Patch Cord,DLC/PC,DLC/PC,Multi-mode,3m,A1a.2,2mm,OM3 bending insensitive	160	11.00	1,760.00
	P-16mm^2-Olivine-LSZH	Power Cable,450V/750V,H07Z-K UL3386,16mm^2,Yellow/Green,107A,LSZH Cable,VDE,UL (Unit:meter)	10	3.80	38.00
	C3006BK01	Power Cable,600V/1000V,ZA-RVV,3x6mm^2,Black(3Cores:Brown,Blue,Yellow/Green),46A, Outdoor Cable,CE (Unit:meter)	80	6.90	552.00
1.1.9	Storage Software				
	88V3-LBASIC-N	Basic Software Suite License(OceanStor OS,DeviceManager,SmartThin,SmartMotion,SmartQos,SmartPartition,SmartCache,SmartMigration,SmartErase,SmartMulti-tenant,SystemReporter,Cloud Service)	1	2,815.25	2,815.25
	88V3-LBASIC200	Basic Software Suite Capacity License(101-200TB)	200	160.87	32,174.00
	88V3-LBASICU	Basic Software Suite Unlimited Capacity License	1	0.01	0.01
	88V3-LULTRAPATH	OceanStor UltraPath Software License	1	596.37	596.37
Total of Product					2,250,024.62
1.1.10	Maintenance Support Service				
	88125ESH	OceanStor 18800 V3 Installation Service - Engineering	1	107,346.67	107,346.67
	88032XVE-88134UHK-3	Basic Software Suite Capacity License(101-200TB)-Hi-Care Application Software Upgrade Support Service-3Year(s)	200	62.74	12,548.00
	88032XVG-88134UHK-3	Basic Software Suite Unlimited Capacity License-Hi-Care Application Software Upgrade Support Service-3Year(s)	1	0.01	0.01
	88032YCT-88134UHK-3	OceanStor UltraPath Software License-Hi-Care Application Software Upgrade Support Service-3Year(s)	1	0.01	0.01
	88033JKR-88134UHK-3	Basic Software Suite License(OceanStor OS,DeviceManager,SmartThin,SmartMotion,SmartQos,SmartPartition,SmartCache,SmartMigration,SmartErase,SmartMulti-tenant,SystemReporter,Cloud Service)-Hi-Care Application Software Upgrade Support Service-3Year(s)	1	844.58	844.58
Total of Service (3 years)					120,739.27
Total Price					2,370,763.89

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.