ORACLE

SPC BENCHMARK1™
FULL DISCLOSURE REPORT

ORACLE CORPORATION
ORACLE SUN ZFS STORAGE 7420C APPLIANCE

SPC-1 V1.12

Submitted for Review: October 1, 2011
Submission Identifier: A00108
Revised: November 30, 2011

First Edition – October 2011

Trademarks

Table of Contents

AUDIT CERTIFICATION

Steven A. Johnson
Oracle Corporation
500 Eldorado Blvd.
Broomfield, CO 80021

October 1, 2011

The SPC Benchmark 1™ Reported Data listed below for the Oracle Sun ZFS Storage 7420c Appliance were produced in compliance with the SPC Benchmark 1™ v1.12 Onsite Audit requirements.

SPC Benchmark 1™ v1.12 Reported Data	
Tested Storage Product (TSP) Name: Oracle Sun ZFS Storage 7420c Appliance	
Metric	**Reported Result**
SPC-1 IOPS™	137,066.20
SPC-1 Price-Performance	$2.99/SPC-1 IOPS™
Total ASU Capacity	23,703.035 GB
Data Protection Level	Protected *(Mirroring)*
Total TSC Price (including three-year maintenance)	$409,933

The following SPC Benchmark 1™ Onsite Audit requirements were reviewed and found compliant with 1.12 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 physical inspection and information supplied by Oracle Corporation:
  - ✓ 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.
- An appropriate diagram of the Benchmark Configuration *(BC)*/Tested Storage Configuration *(TSC)*.
- Physical verification of the components to match the above diagram.
- Listings and commands to configure the Benchmark Configuration/Tested Storage Configuration, including customer tunable parameters that were changed from default values.

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

AUDIT CERTIFICATION *(CONT.)*

- SPC-1 Workload Generator commands and parameters used for the audited SPC Test Runs.
- The following Host System requirements were verified by physical inspection and information supplied by Oracle Corporation:
  - ✓ The type of Host System including the number of processors and main memory.
  - ✓ The presence and version number of the SPC-1 Workload Generator on the Host System.
  - ✓ The TSC boundary within the 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 Oracle Corporation 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
- The documented differences between the Tested Storage Configuration and Priced Storage Configuration had no influence on the reported performance of the Tested 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.

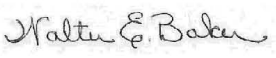
Audit Notes:

The requirement for identical Start-Up durations for each Test Run in the Primary Metrics was waived to provide a more appropriate selection of Start-Up durations.

The 95%, 90%, 80% and 50% BSU Response Time Ramp Test Runs each were each specified with one BSU more than required.

Neither of the above two items had any impact on the reported performance of the Tested Storage Configuration.

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

ORACLE

Oracle America, Inc. 500 Eldorado Boulevard Broomfield Colorado 80021 phone +1.303.464.4000 oracle.com

September 13, 2011
From:
Oracle Corporation
Steven A. Johnson
500 Eldorado Blvd.
Broomfield, CO 80021

To: Walter Baker
Gradient System
643 Blair Island road, Suite 103
Redwood City, CA 94063

Subject: SPC-1 Letter of Good Faith for the **Oracle's Sun ZFS Storage 7420c, Mirror**
Test Sponsor Name 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 **V1.13** 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.

Sincerely, Date:

Brian Wong Director Software Development — 13 Sept 2011 Date of Signature

EXECUTIVE SUMMARY

Test Sponsor and Contact Information

Test Sponsor and Contact Information	
Test Sponsor Primary Contact	Oracle Corporation – http://www.oracle.com Steven Johnson – Steven.A.Johnson@oracle.com 500 Eldorado Blvd. Broomfield, CO 80021 Phone: (303) 272-9476
Test Sponsor Alternate Contact	Oracle Corporation – http://www.oracle.com Jason Schaffer – Jason.Schaffer@oracle.com 500 Eldorado Blvd. Broomfield, CO 80021 Phone: (303) 272-4743 FAX: (303) 272-9704
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	V1.12
SPC-1 Workload Generator revision number	V2.1.0
Date Results were first used publicly	October 1, 2011
Date the FDR was submitted to the SPC	October 1, 2011
Date revised FDR was submitted to the SPC Appendix C was revised to more clearly document the TSC creation and configuration process.	November 30, 2011
Date the Priced Storage Configuration is available for shipment to customers	January 1, 2012
Date the TSC completed audit certification	October 1, 2011

Tested Storage Product (TSP) Description

Oracle's Sun ZFS Storage Appliances are Oracle's leading solution for NAS environments. Sun ZFS Storage Appliances provides high performance and enterprise class stability for business critical cloud computing, virtualization, storage consolidation, data protection and fixed media serving applications, with unique advantages for environments built on Oracle databases, middleware and applications. Sun ZFS Storage appliances leverage a deeply integrated performance-based architecture, rich data services and powerful and user friendly management and analytics tools to enable extreme performance of the storage system, easily adapt to changing business conditions and radically simplify storage management. Sun ZFS Storage Appliances deliver additional economic value bundled data services such as file and block-level protocols including connectivity over InfiniBand, Compression, Deduplication, Thin provisioning, DTrace Analytics, Virus Scan, Snapshots, Triple Mirror, Triple Parity RAID, Phone-home, NDMP, Clustering, etc. As a unified platform, ZFS Storage Appliances can simultaneously optimize NAS and SAN workloads with a broad set of protocol and connectivity options to support consolidation initiatives.

Summary of Results

SPC-1 Reported Data	
Tested Storage Product (TSP) Name: Oracle Sun ZFS Storage 7420c Appliance	
Metric	**Reported Result**
SPC-1 IOPS™	137,066.20
SPC-1 Price-Performance™	$2.99/SPC-1 IOPS™
Total ASU Capacity	23,703.035 GB
Data Protection Level	Protected *(Mirroring)*
Total TSC Price (including three-year maintenance)	$409,933

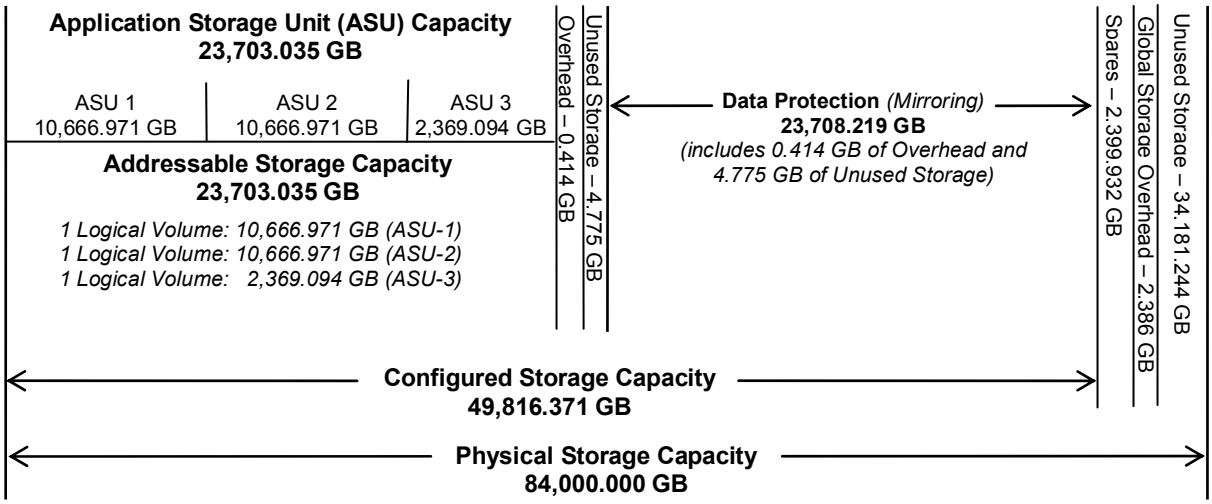
SPC-1 IOPS™ represents the maximum I/O Request Throughput at the 100% load point.

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

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

Storage Capacities, Relationships, and Utilization

The following diagram 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.



SPC-1 Storage Capacity Utilization	
Application Utilization	28.22%
Protected Application Utilization	56.44%
Unused Storage Ratio	40.70%

Application Utilization: Total ASU Capacity *(23,703.035 GB)* divided by Physical Storage Capacity *(84,000.000 GB)*

Protected Application Utilization: (Total ASU Capacity *(23,703.035 GB)* plus total Data Protection Capacity *(23,708.219 GB)* minus unused Data Protection Capacity *(4.744 GB)* divided by Physical Storage Capacity *(84,000.000 GB)*

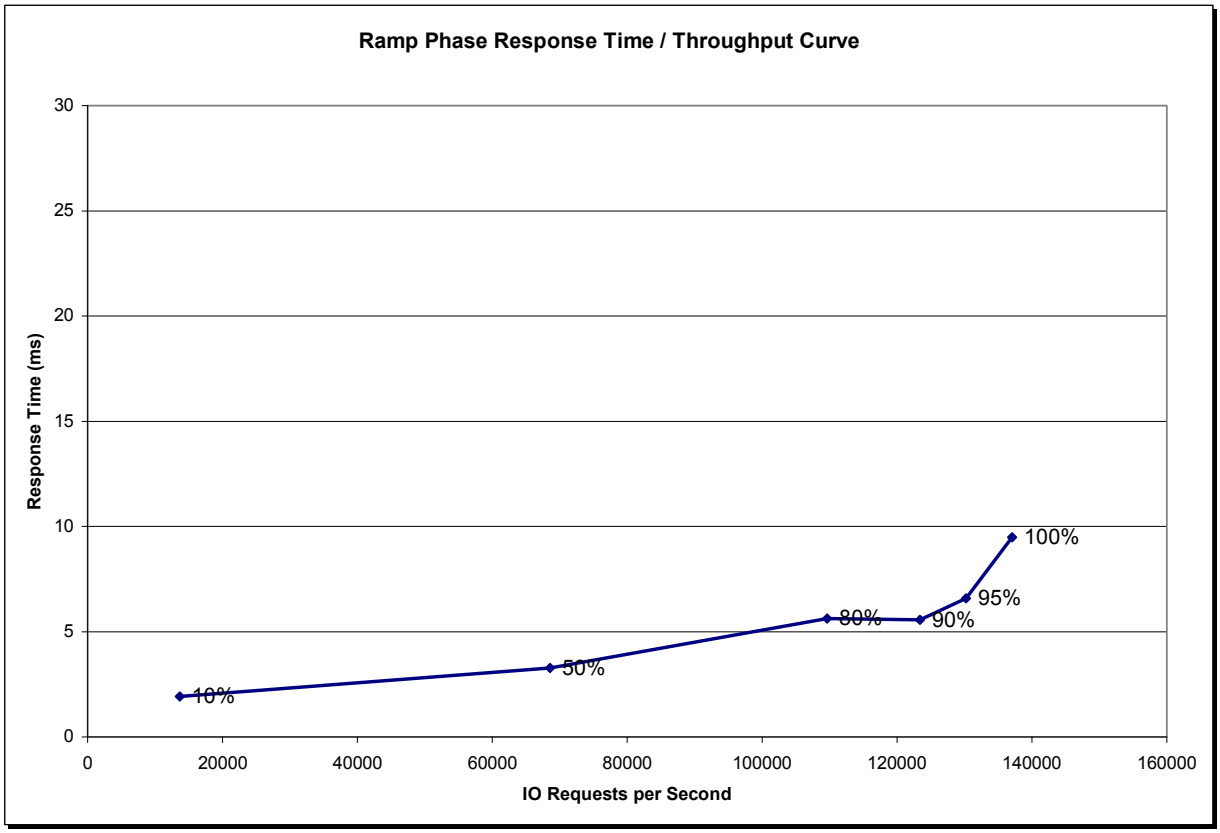
Unused Storage Ratio: Total Unused Capacity *(34,191.732 GB)* divided by Physical Storage Capacity *(84,000.000 GB)* and may not exceed 45%.

Detailed information for the various storage capacities and utilizations is available on pages 21-22.

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	13,702.49	68,542.59	109,651.91	123,355.14	130,194.51	137,066.20
Average Response Time (ms):						
All ASUs	1.91	3.26	5.62	5.57	6.60	9.48
ASU-1	2.17	3.69	5.98	6.07	6.98	10.15
ASU-2	2.86	5.24	8.52	9.05	10.54	15.28
ASU-3	0.95	1.49	3.58	2.98	4.06	5.53
Reads	2.17	4.05	6.31	6.95	8.13	12.14
Writes	1.75	2.75	5.17	4.66	5.59	7.75

Priced Storage Configuration Pricing

Part Number	Description	Quantity	US List	Total	Discount	Net Price
TA7420-28A-HA	Sun ZFS Storage 7420: controller for cluster configuration with 2 Intel(R) Xeon(R) X7550 8-core 2.0 GHz processors and 2 SAS- 2 HBAs (for factory installation)	2	$24,493	$48,986	30%	$34,290
8505A	Sun Fire X4470 server: 16 GB memory kit with two 8 GB 1066 MHz DDR3 DIMMs (for factory installation)	64	$799	$51,136	30%	$35,795
TA7000-READZ512	one 512 GB SATA SSD 2.5-inch read-flash accelerator with silver marlin bracket (for factory installation)	8	$3,732	$29,856	30%	$20,899
SG-PCIE2FC-QF8-Z	StorageTek 8 Gb Fibre Channel PCIe HBA dual port QLogic (for factory Installation - includes SFPs)	12	$2,399	$28,788	30%	$20,152
SG-PCIESAS-GEN2-Z	SAS PCIE 6Gbs 8 port	4	$679	$2,716	30%	$1,901
TA-2.0M-SAS	2m, Mini, shielded, SAS cable	8	$150	$1,200	30%	$840
2350A	Two 8-DIMM riser card (for factory installation)	4	$499	$1,996	30%	$1,397
2342A	2x Intel® Xeon® X7550 8-Core 2.00 GHz CPUs (for factory installation)	2	$10,699	$21,398	30%	$14,979
333A-25-15-NEMA	Power cord: North America and Asia, 2.5 meters, 5-15P plug, C13 connector, 15 A (for factory installation)	4	$13	$52	30%	$36
DS2-0BASE	Sun disk shelf: base chassis with 2 SAS-2 I/O modules, 2 AC PSUs and 2 cooling fans (for factory installation) Includes two 0.5M, Mini, shielded SAS cables	12	$4,905	$58,860	30%	$41,202
DS2-4URK-19U	Sun disk shelf: universal rail kit for 19-inch depth racks (for factory installation)	12	$230	$2,760	30%	$1,932
7101274	300GB 15K RPM disk	280	$411	$115,080	30%	$80,556
7101197	one SLC SAS-2 SSD 3.5-inch write-flash accelerator with stingray bracket (for factory installation)	8	$6,536	$52,288	30%	$36,602
333A-25-15-NEMA	Power cord: North America and Asia, 2.5 meters, 5-15P plug, C13 connector, 15 A (for factory installation)	24	$13	$312	30%	$218
SG-XPCIE2FC-QF8-N	Sun StorageTek 8 Gb FC PCIe Host Bus Adapter, Dual Port Includes Standard and Low Profile Brackets, Low Profile Form Factor, QLogic, RoHS-6 Compliant (includes SFPs)	6	$2,399	$14,394	30%	$10,076
X9732A-Z-N	2M LC to LC FC Optical Cable RoHS-6 compliant	12	65.00	$780	30%	$546
	Oracle Premium Support for Systems: 1-Year 7/24, 2 hour response time.	3		**$155,017**	30%	$108,512
				$585,619		**$409,933**

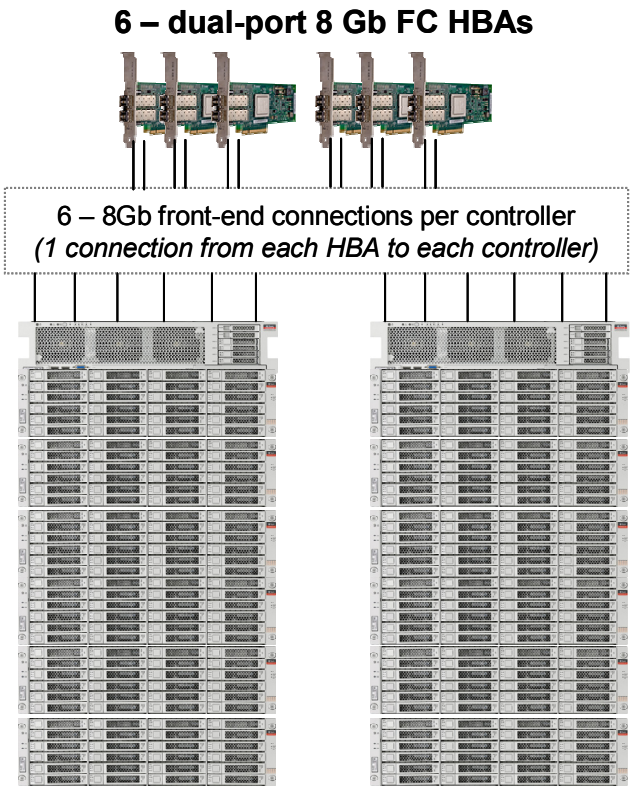
The above pricing includes hardware maintenance and software support for three years, 7 days per week, 24 hours per day. The hardware maintenance and software support provides the following:

- 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 Priced Storage Configuration that can be remedied by the repair or replacement of a Priced Storage Configuration component.

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

The Tested Storage Configuration included two (2) power distribution units, with associated power cables, and racking for the controllers and disk shelves. Those components were not included in the Priced Storage Configuration. Those components had no influence on the reported performance of the Tested Storage Configuration.

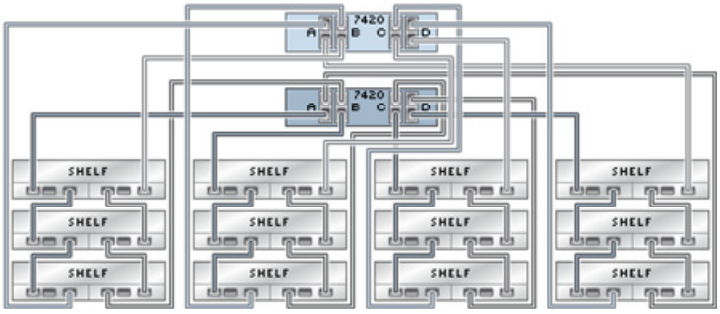
Priced Storage Configuration Diagram



Rear View - Connections

Controller A: 8 – SAS-2 backend connections

Controller B: 8 – SAS-2 backend connections



Priced Storage Configuration Components

Priced Storage Configuration Components:
6 – Sun StorageTek 8Gb Fibre Channel PCIe HBAs *(includes SFPs)*
Oracle Sun ZFS Storage 7420c Appliance 2 – Sun ZFS 7420 controllers *(cluster configuration)* 512 GB cache/memory per controller *(1024 GB total)* 12 – Sun StorageTek 8Gb Fibre Channel PCIe HBAs *(includes SFPs)* 8 – dual-ported SAS-2 HBAs 8 – 512 GB Solid State Devices *(read cache SSDs, 4096 GB total)* 8 – 73 GB Solid State Devices *(write cache SSDs, 584 GB total)* 24 – 8 Gb FC front-end connections *(12 used)* 16 – SAS-2 backend connections *(16 used)*
12 – 2m LC to LC FC Optical Cables RoHS-6 compliant
8 – 2m, Mini, shielded, SAS cables
12 – Sun disk shelf: base chassis each with 2 SAS-2 IO modules, 2 AC PSUs and 2 cooling fans
280 – 300 GB 15K RPM SAS-2 disk drives

Note: All HBAs include the required SFPs.

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 18 *(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 did not include network storage.

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

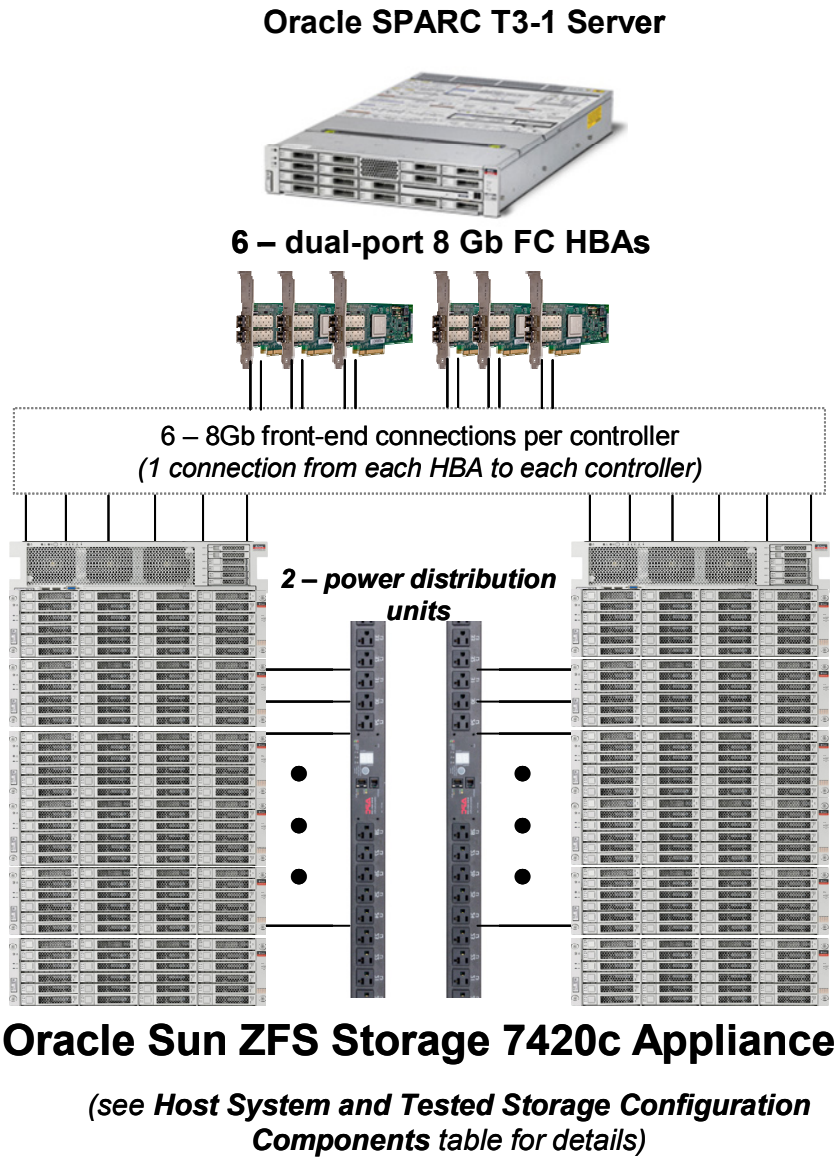
Clause 9.4.3.4.3

The FDR will contain a table that lists the major components of each Host System and the Tested Storage Configuration (TSC). Table 9-10 specifies the content, format, and appearance of the table.

The Host System and TSC table of components may be found on page 19 *(Benchmark Configuration/Tested Storage Configuration Diagram (cont.)*

Host System and Tested Storage Configuration Components*).*

Benchmark Configuration/Tested Storage Configuration Diagram



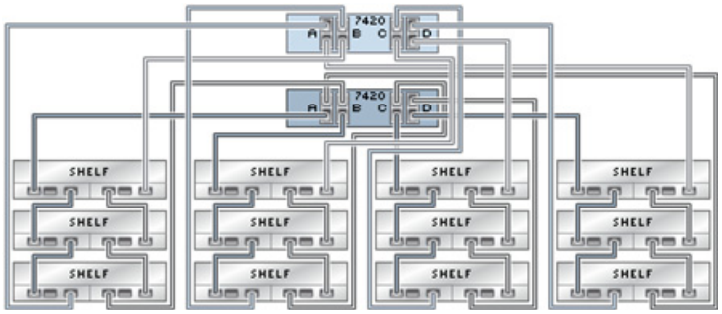
*(see **Host System and Tested Storage Configuration Components** table for details)*

Benchmark Configuration/Tested Storage Configuration Diagram *(cont.)*

Rear View - Connections

Controller A: 8 – SAS-2 backend connections

Controller B: 8 – SAS-2 backend connections



Host System and Tested Storage Configuration Components

Host System:	Tested Storage Configuration (TSC):
Oracle SPARC T3-1 Server with: 1 – 16-core 1.65 GHz SPARC T3 processor SPARC v9 architecture, ECC protected 6 MB integrated L2 cache 128 GB of main memory	6 – Sun StorageTek 8Gb Fibre Channel PCIe HBAs
Solaris 10 Update9	**Oracle Sun ZFS Storage 7420c Appliance** 2 – Sun ZFS 7420 controllers *(cluster configuration)* 512 GB cache/memory per controller *(1024 GB total)* 12 – Sun StorageTek 8Gb Fibre Channel PCIe HBAs 8 – dual-ported SAS-2 HBAs 8 – 512 GB Solid State Devices *(read cache SSDs, 4096 GB total)* 8 – 73 GB Solid State Devices *(write cache SSDs, 584 GB total)* 24 – 8 Gb FC front-end connections *(12 used)* 16 – SAS-2 backend connections *(16 used)*
PCIe Gen2	12 – 2m LC to LC FC Optical Cables RoHS-6 compliant
	8 – 2m, Mini, shielded, SAS cables
	2 – power distribution units and associated power cables
	12 – Sun disk shelf: base chassis each with 2 SAS-2 IO modules, 2 AC PSUs and 2 cooling fans
	280 – 300 GB 15K RPM SAS-2 disk drives

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 65 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 66 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 104.

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 61 contains definitions of terms specific to the SPC-1 Data Repository.

Storage Capacities and Relationships

Clause 9.4.3.6.1

Two tables and an illustration documenting the storage capacities and relationships of the SPC-1 Storage Hierarchy (Clause 2.1) shall be included in the FDR.

SPC-1 Storage Capacities

SPC-1 Storage Capacities		
Storage Hierarchy Component	**Units**	**Capacity**
Total ASU Capacity	Gigabytes (GB)	23,703.035
Addressable Storage Capacity	Gigabytes (GB)	23,703.035
Configured Storage Capacity	Gigabytes (GB)	49,816.371
Physical Storage Capacity	Gigabytes (GB)	84,000.000
Data Protection *(Mirroring)*	Gigabytes (GB)	23,708.219
Required Storage/Spares	Gigabytes (GB)	2,399.932
Global Storage Overhead	Gigabytes (GB)	2.386
Total Unused Storage	Gigabytes (GB)	34,191.732

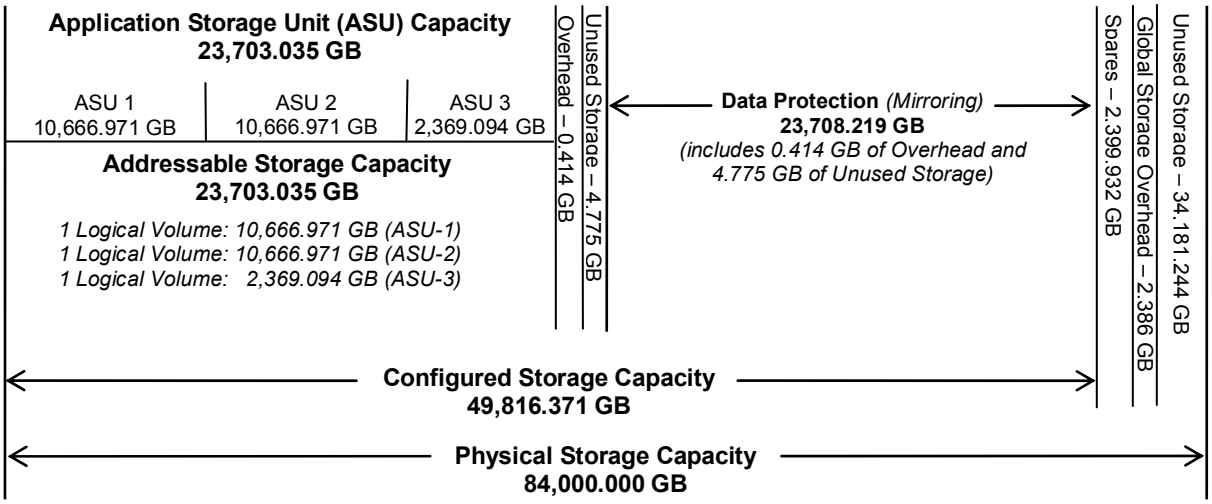
SPC-1 Storage Hierarchy Ratios

	Addressable Storage Capacity	**Configured Storage Capacity**	**Physical Storage Capacity**
Total ASU Capacity	100.00%	47.58%	28.22%
Required for Data Protection *(Mirrored)*		47.59%	28.22%
Addressable Storage Capacity		47.58%	28.22%
Required Storage		4.82%	2.86%
Configured Storage Capacity			59.31%
Global Storage Overhead			0.003%
Unused Storage:			
Addressable	0.00%		
Configured		0.02%	
Physical			40.69%

The Physical Storage Capacity consisted of 84,000.000 GB distributed over 280 disk drives, each with a formatted capacity of 300.000 GB. There was 34,181.244 GB (40.69%) of Unused Storage within the Physical Storage Capacity. Global Storage Overhead consisted of 2.386 GB (0.003%) of the Physical Storage Capacity. There was 9.550 GB (0.02%) of Unused Storage within the Configured Storage Capacity. The Total ASU Capacity utilized 100% of the Addressable Storage Capacity resulting in 0.000 GB (0.00%) of Unused Storage within the Addressable Storage Capacity. The Data Protection *(mirroring)* capacity was 23,708.291 GB of which 23.703.035 GB was utilized. The total Unused Storage was 34,190.794 GB.

SPC-1 Storage Capacities and Relationships Illustration

The various storage capacities configured in the benchmark result are illustrated below *(not to scale)*.



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 (10,666.971 GB)	**ASU-2 (10,666.971 GB)**	**ASU-3 (2,369.094 GB)**
1 Logical Volume 10,666.917 GB per Logical Volume (10,666.917 used per Logical Volume)	1 Logical Volume 10,666.917 GB per Logical Volume (10,666.917 used per Logical Volume)	1 Logical Volume 2,369.094 GB per Logical Volume (2,369.094 used per Logical Volume)

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

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	28.22%
Protected Application Utilization	56.44%
Unused Storage Ratio	40.70%

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. "SPC-1 Test Execution Definitions" on page 62 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.

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 three (3) 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 9.4.3.7.1

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 105.

Sustainability Test Results File

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

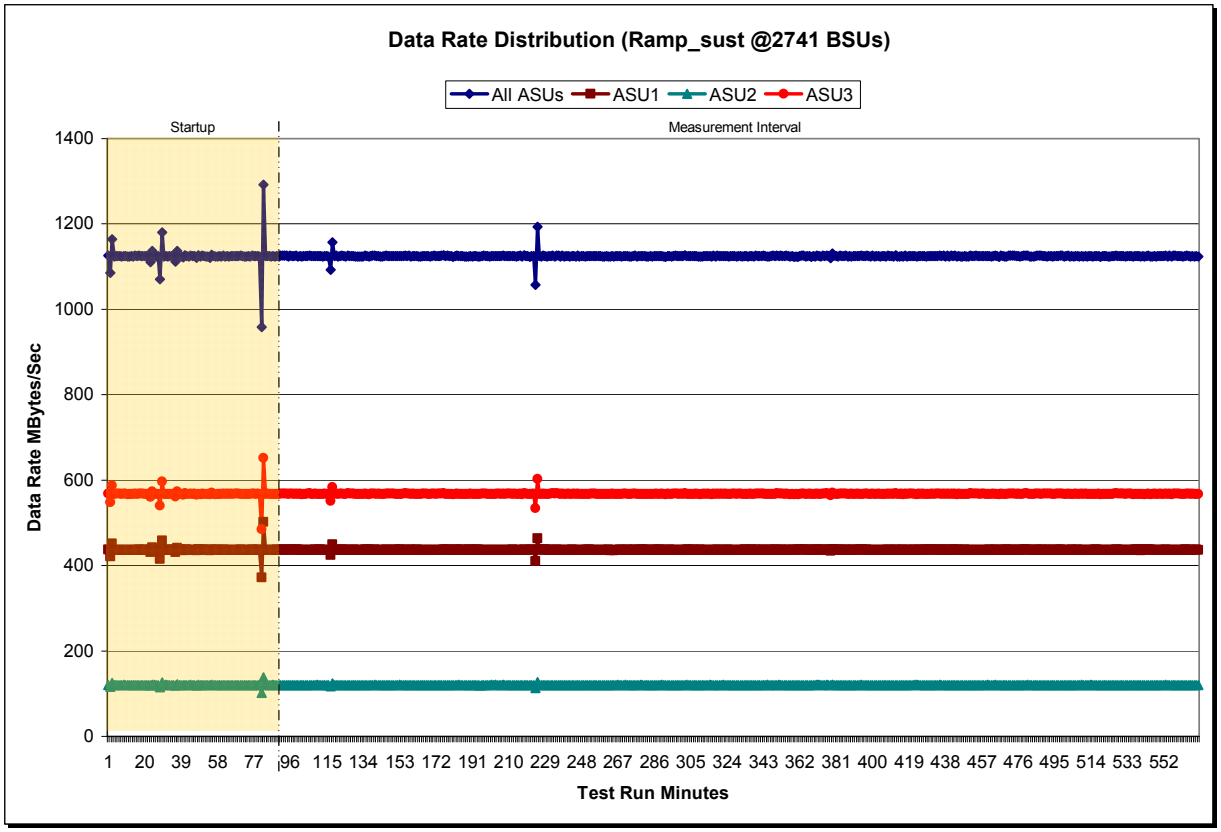
Sustainability Test Results File

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 Tables

Sustainability – Data Rate Distribution Graph

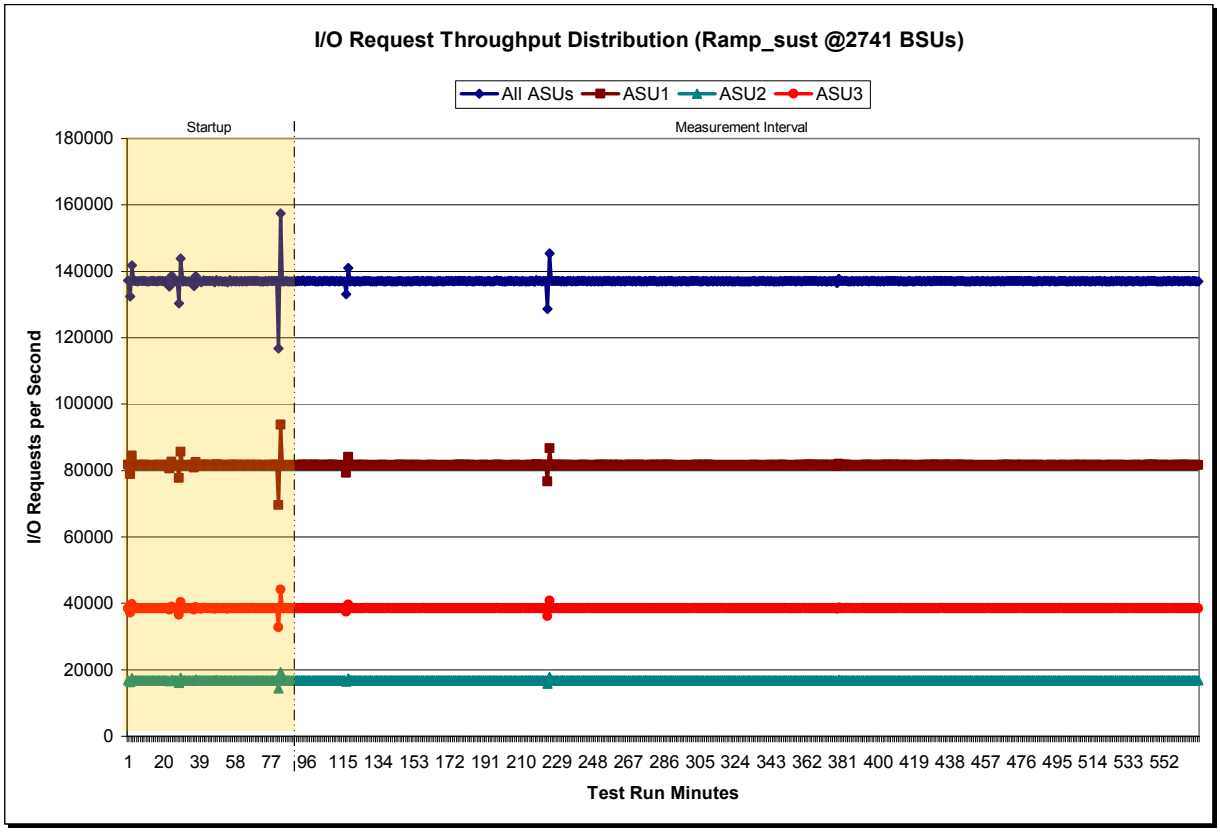


Sustainability – I/O Request Throughput Distribution Data

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 Data Tables

Sustainability – I/O Request Throughput Distribution Graph

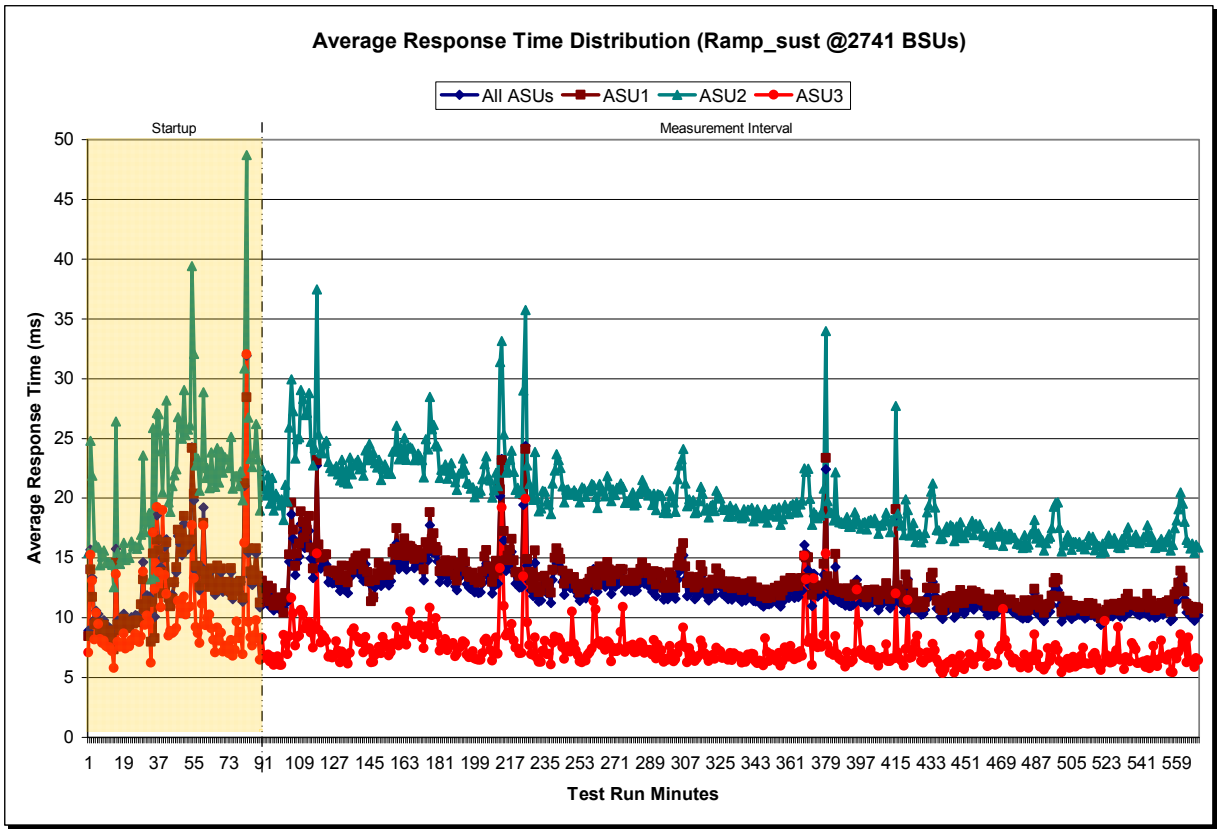


Sustainability – Average Response Time (ms) Distribution Data

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 Data Tables

Sustainability – Average Response Time (ms) Distribution Graph



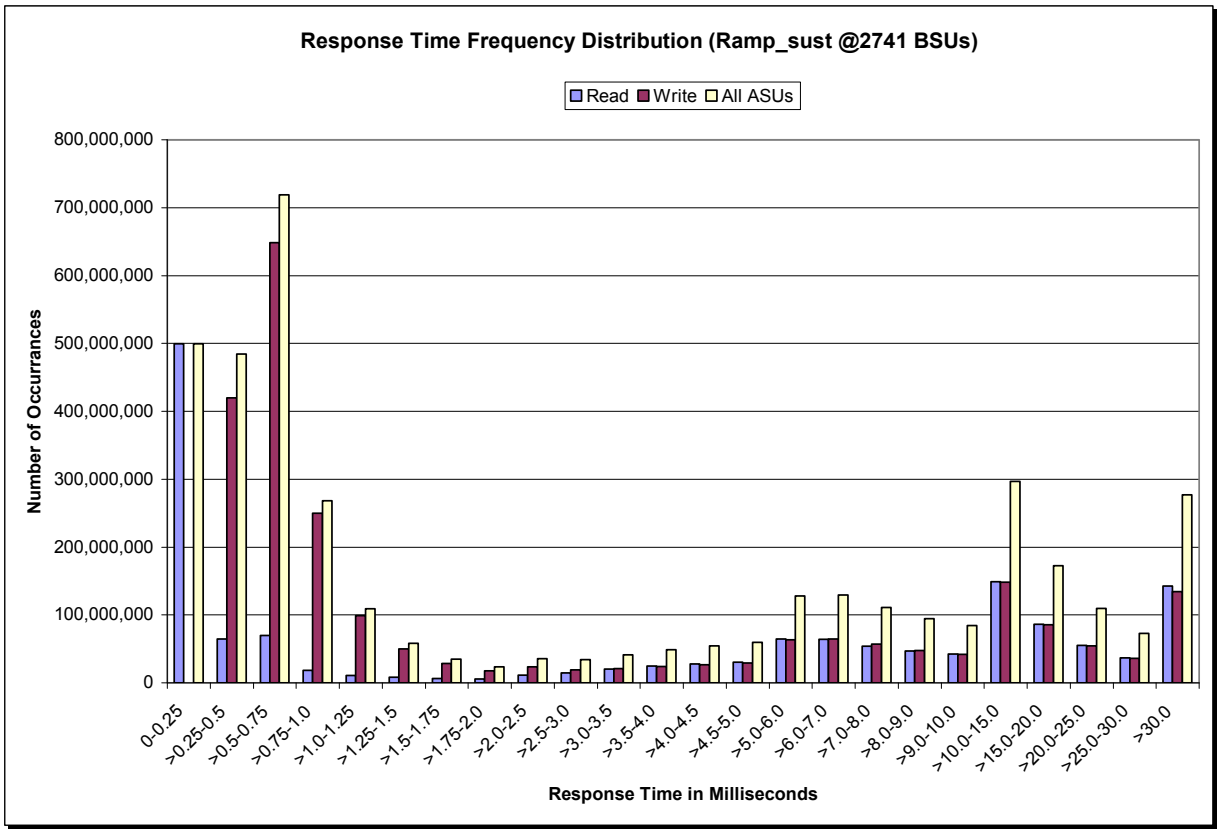
Sustainability – Response Time Frequency Distribution Data

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	499,647,436	64,758,735	70,026,766	18,204,575	10,520,260	7,994,238	6,563,717	5,840,237
Write	-	419,750,926	648,707,604	250,057,341	98,661,710	50,244,801	28,325,966	17,623,864
All ASUs	499,647,436	484,509,661	718,734,370	268,261,916	109,181,970	58,239,039	34,889,683	23,464,101
ASU1	429,252,815	235,688,996	297,734,188	83,876,836	38,612,974	22,913,824	14,232,570	10,783,635
ASU2	70,394,621	55,251,930	39,300,256	11,163,720	5,600,590	3,348,304	2,150,651	1,699,328
ASU3	-	193,568,735	381,699,926	173,221,360	64,968,406	31,976,911	18,506,462	10,981,138

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	11,706,465	14,841,640	20,510,203	24,694,672	28,056,104	30,179,727	64,732,783	64,257,056
Write	23,563,756	19,310,388	20,831,010	24,078,183	26,626,084	29,210,805	63,228,513	64,957,543
All ASUs	35,270,221	34,152,028	41,341,213	48,772,855	54,682,188	59,390,532	127,961,296	129,214,599
ASU1	19,003,502	21,404,626	28,595,670	35,110,599	39,993,708	43,444,243	93,288,323	93,866,897
ASU2	3,085,345	3,591,949	5,017,922	6,369,216	7,479,414	8,499,909	19,168,616	20,012,145
ASU3	13,181,374	9,155,453	7,727,621	7,293,040	7,209,066	7,446,380	15,504,357	15,335,557

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	53,985,472	46,957,913	42,641,338	148,908,526	86,502,276	55,339,881	37,020,517	142,706,234
Write	57,258,687	47,774,955	41,876,714	148,035,939	85,713,226	54,247,374	36,180,587	134,187,111
All ASUs	111,244,159	94,732,868	84,518,052	296,944,465	172,215,502	109,587,255	73,201,104	276,893,345
ASU1	79,500,610	67,567,535	60,882,062	212,037,172	121,166,996	76,066,316	50,102,758	177,339,169
ASU2	17,808,601	15,699,703	14,105,158	51,001,663	30,756,280	20,404,893	14,199,235	59,340,093
ASU3	13,934,948	11,465,630	9,530,832	33,905,630	20,292,226	13,116,046	8,899,111	40,214,083

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 5.3.13.2

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 5.3.13.3

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	0.002	0.001	0.001	0.001	0.003	0.001	0.002	0.001

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 9.4.3.7.2

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 105.

IOPS Test Results File

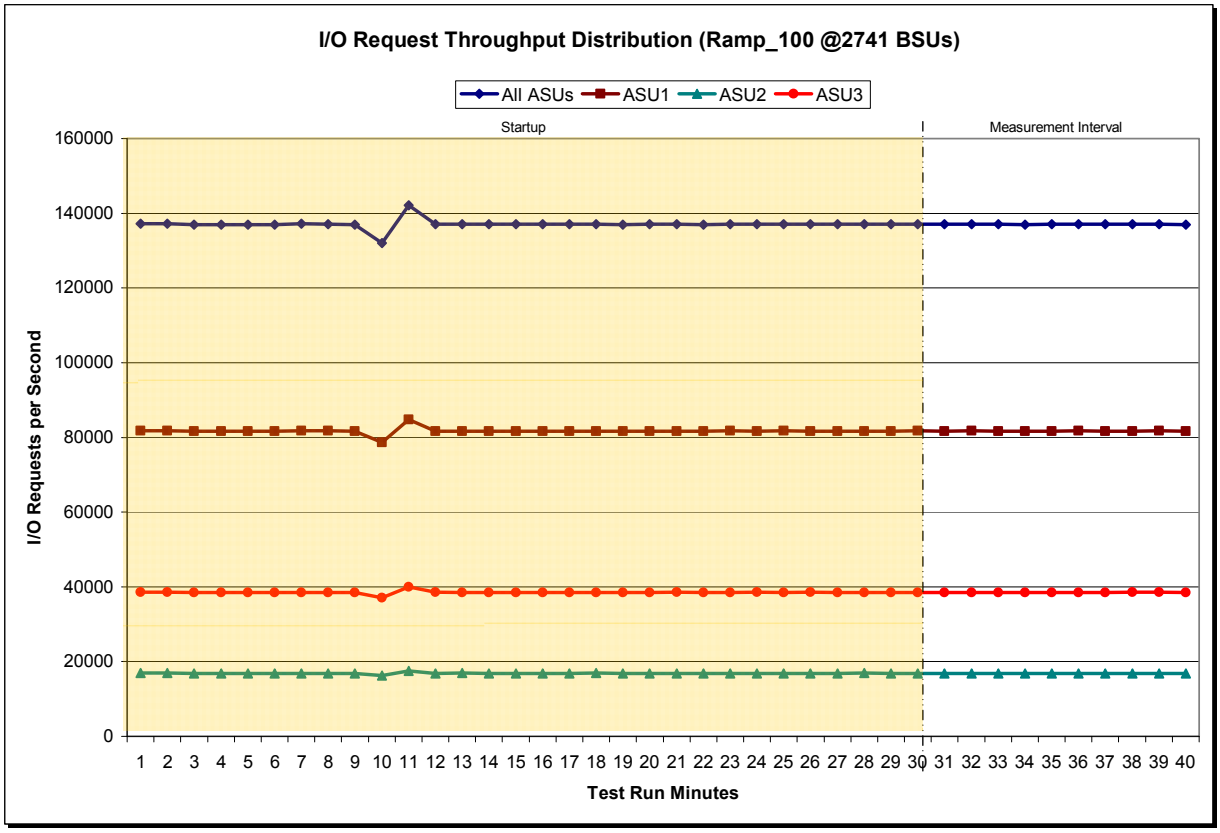
A link to the test results file generated from the IOPS Test Run is listed below.

IOPS Test Results File

IOPS Test Run – I/O Request Throughput Distribution Data

2741 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	0:09:25	0:39:25	0-29	0:30:00
Measurement Interval	0:39:25	0:49:25	30-39	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	137,183.78	81,745.88	16,883.57	38,554.33
1	137,154.40	81,730.97	16,876.17	38,547.27
2	136,990.02	81,661.72	16,870.63	38,457.67
3	136,967.83	81,634.65	16,841.42	38,491.77
4	136,994.12	81,631.67	16,849.38	38,513.07
5	137,011.37	81,633.20	16,857.77	38,520.40
6	137,143.82	81,764.85	16,853.90	38,525.07
7	137,101.92	81,731.25	16,864.70	38,505.97
8	137,007.62	81,673.02	16,840.30	38,494.30
9	132,035.47	78,704.58	16,241.42	37,089.47
10	142,111.48	84,731.52	17,461.18	39,918.78
11	137,070.70	81,667.70	16,864.07	38,538.93
12	137,070.85	81,683.08	16,885.77	38,502.00
13	137,065.42	81,688.95	16,850.02	38,526.45
14	137,023.05	81,704.43	16,849.33	38,469.28
15	137,022.53	81,643.92	16,858.93	38,519.68
16	137,086.92	81,698.43	16,860.23	38,528.25
17	137,132.95	81,725.85	16,885.93	38,521.17
18	137,012.65	81,655.53	16,861.63	38,495.48
19	137,076.67	81,714.30	16,852.70	38,509.67
20	137,097.92	81,693.02	16,872.15	38,532.75
21	137,015.35	81,656.43	16,862.25	38,496.67
22	137,098.62	81,730.18	16,869.88	38,498.55
23	137,051.35	81,664.28	16,843.78	38,543.28
24	137,094.38	81,758.60	16,857.48	38,478.30
25	137,065.78	81,674.83	16,851.73	38,539.22
26	137,032.92	81,668.93	16,866.87	38,497.12
27	137,067.57	81,676.20	16,888.33	38,503.03
28	137,022.10	81,690.85	16,833.00	38,498.25
29	137,110.22	81,738.72	16,874.38	38,497.12
30	137,045.48	81,705.40	16,850.37	38,489.72
31	137,061.12	81,731.80	16,848.58	38,480.73
32	137,069.53	81,696.08	16,852.75	38,520.70
33	137,004.48	81,630.30	16,863.45	38,510.73
34	137,066.12	81,682.33	16,859.38	38,524.40
35	137,127.40	81,741.73	16,873.42	38,512.25
36	137,042.28	81,688.97	16,833.85	38,519.47
37	137,104.93	81,692.17	16,860.15	38,552.62
38	137,122.00	81,738.75	16,845.38	38,537.87
39	137,018.62	81,672.60	16,873.92	38,472.10
Average	***137,066.20***	***81,698.01***	***16,856.13***	***38,512.06***

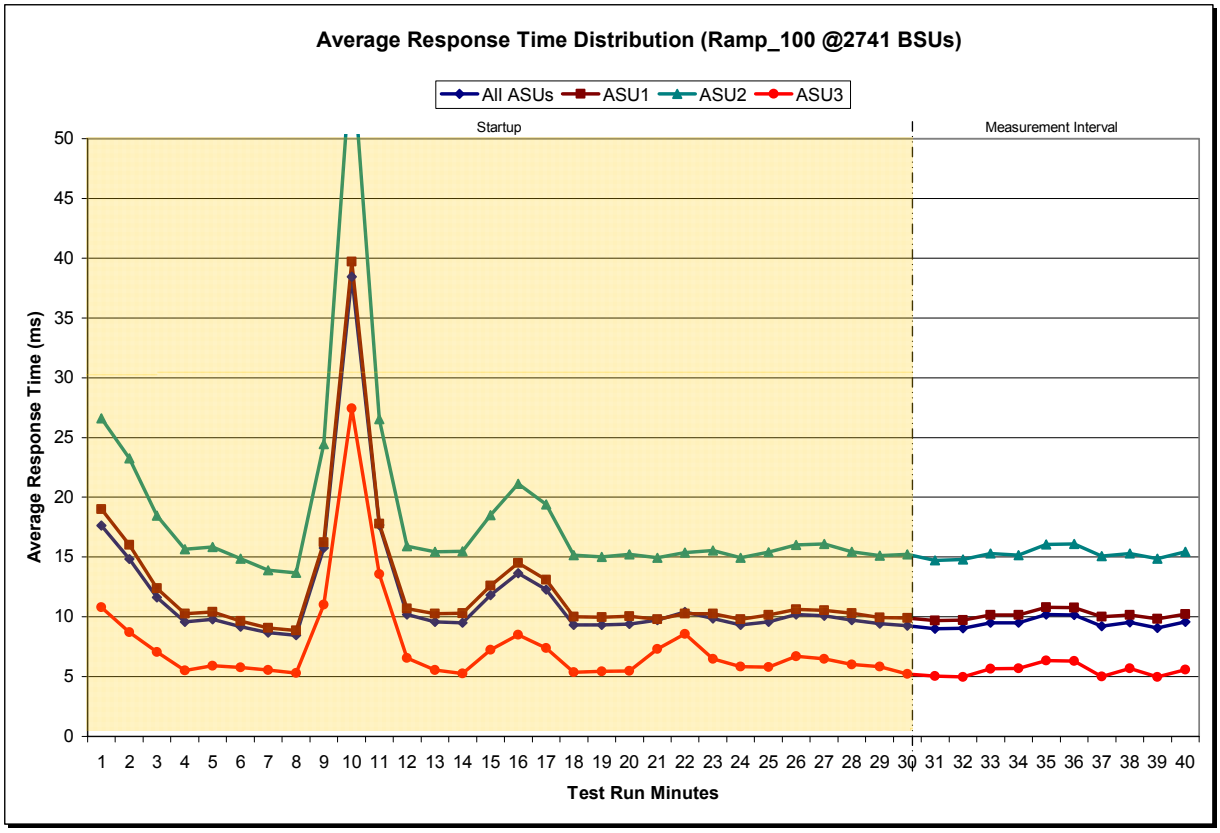
IOPS Test Run – I/O Request Throughput Distribution Graph



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

2741 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	0:09:25	0:39:25	0-29	0:30:00
Measurement Interval	0:39:25	0:49:25	30-39	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	17.64	19.02	26.61	10.78
1	14.84	16.00	23.26	8.70
2	11.63	12.37	18.47	7.04
3	9.59	10.26	15.67	5.52
4	9.80	10.40	15.85	5.90
5	9.19	9.64	14.88	5.74
6	8.67	9.07	13.90	5.53
7	8.45	8.86	13.69	5.29
8	15.78	16.24	24.45	11.00
9	38.43	39.69	57.47	27.44
10	17.67	17.77	26.53	13.58
11	10.18	10.70	15.91	6.57
12	9.57	10.25	15.44	5.55
13	9.51	10.28	15.48	5.27
14	11.81	12.58	18.51	7.23
15	13.64	14.51	21.14	8.50
16	12.27	13.09	19.40	7.39
17	9.34	10.00	15.17	5.38
18	9.32	9.97	15.00	5.45
19	9.41	10.06	15.22	5.47
20	9.73	9.80	14.95	7.30
21	10.42	10.27	15.36	8.58
22	9.85	10.26	15.56	6.47
23	9.32	9.80	14.95	5.83
24	9.57	10.15	15.39	5.78
25	10.19	10.64	16.01	6.68
26	10.09	10.55	16.08	6.49
27	9.72	10.28	15.45	6.02
28	9.42	9.93	15.13	5.85
29	9.24	9.90	15.21	5.21
30	8.99	9.67	14.71	5.05
31	9.02	9.73	14.79	4.99
32	9.51	10.14	15.29	5.64
33	9.51	10.15	15.14	5.70
34	10.20	10.80	16.06	6.34
35	10.16	10.75	16.10	6.29
36	9.22	10.00	15.09	5.00
37	9.53	10.16	15.31	5.69
38	9.07	9.81	14.85	4.97
39	9.58	10.24	15.44	5.60
Average	***9.48***	***10.15***	***15.28***	***5.53***

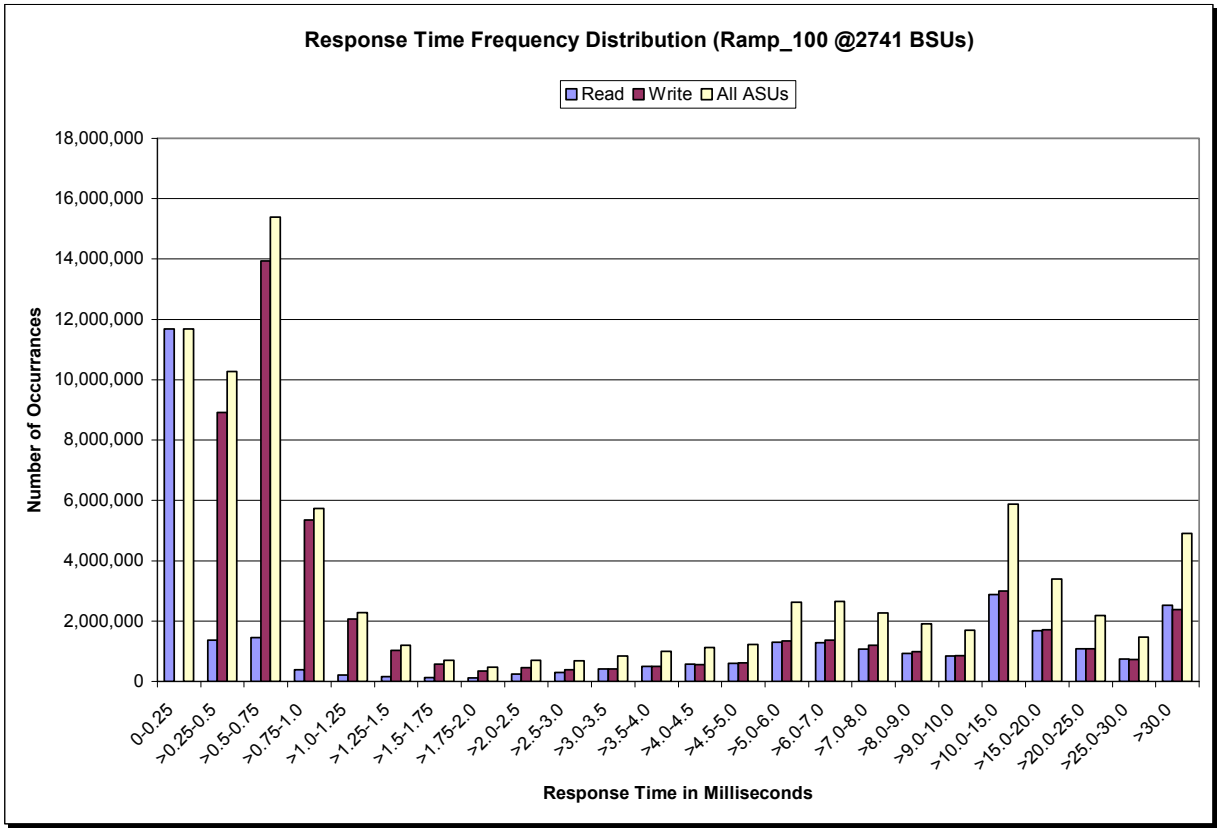
IOPS Test Run – Average Response Time (ms) Distribution Graph



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	11,676,390	1,364,655	1,458,442	378,825	216,248	162,575	132,801	118,381
Write	0	8,908,349	13,929,673	5,349,866	2,071,249	1,030,723	566,991	345,383
All ASUs	11,676,390	10,273,004	15,388,115	5,728,691	2,287,497	1,193,298	699,792	463,764
ASU1	10,180,991	5,008,170	6,369,688	1,784,571	805,905	469,501	285,866	213,743
ASU2	1,495,399	1,167,866	862,108	242,254	117,734	68,561	43,389	34,093
ASU3	1,125	437	120	567	375	750	-	589
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	236,546	298,992	414,897	497,309	565,887	605,919	1,293,778	1,280,061
Write	456,868	379,974	420,194	497,683	558,896	616,060	1,337,500	1,372,205
All ASUs	693,414	678,966	835,091	994,992	1,124,783	1,221,979	2,631,278	2,652,266
ASU1	376,578	427,222	577,548	710,375	813,468	882,428	1,890,396	1,897,614
ASU2	63,338	75,868	109,039	141,185	166,535	188,351	421,138	437,272
ASU3	253,498	175,876	148,504	143,432	144,780	151,200	319,744	317,380
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	1,065,976	922,130	834,827	2,875,607	1,676,348	1,087,653	737,316	2,530,992
Write	1,199,022	990,491	862,770	3,001,620	1,715,105	1,089,326	728,712	2,377,597
All ASUs	2,264,998	1,912,621	1,697,597	5,877,227	3,391,453	2,176,979	1,466,028	4,908,589
ASU1	1,593,341	1,343,783	1,204,585	4,141,341	2,368,539	1,505,306	1,002,121	3,165,151
ASU2	382,826	332,627	299,397	1,060,063	627,840	416,941	290,924	1,068,746
ASU3	288,831	236,211	193,615	675,823	395,074	254,732	172,983	674,692

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
82,238,812	77,330,223	4,908,589

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 5.3.13.2

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 5.3.13.3

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	0.001	0.002	0.001	0.002	0.001

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 13.

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 9.4.3.7.3

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 105.

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 tables and graphs for completeness.

100% Load Level - 2741 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	0:09:25	0:39:25	0-29	0:30:00
Measurement Interval	0:39:25	0:49:25	30-39	0:10:00

(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	137,183.78	81,745.88	16,883.57	38,554.33
1	137,154.40	81,730.97	16,876.17	38,547.27
2	136,990.02	81,661.72	16,870.63	38,457.67
3	136,967.83	81,634.65	16,841.42	38,491.77
4	136,994.12	81,631.67	16,849.38	38,513.07
5	137,011.37	81,633.20	16,857.77	38,520.40
6	137,143.82	81,764.85	16,853.90	38,525.07
7	137,101.92	81,731.25	16,864.70	38,505.97
8	137,007.62	81,673.02	16,840.30	38,494.30
9	132,035.47	78,704.58	16,241.42	37,089.47
10	142,111.48	84,731.52	17,461.18	39,918.78
11	137,070.70	81,667.70	16,864.07	38,538.93
12	137,070.85	81,683.08	16,885.77	38,502.00
13	137,065.42	81,688.95	16,850.02	38,526.45
14	137,023.05	81,704.43	16,849.33	38,469.28
15	137,022.53	81,643.92	16,858.93	38,519.68
16	137,086.92	81,698.43	16,860.23	38,528.25
17	137,132.95	81,725.85	16,885.93	38,521.17
18	137,012.65	81,655.53	16,861.63	38,495.48
19	137,076.67	81,714.30	16,852.70	38,509.67
20	137,097.92	81,693.02	16,872.15	38,532.75
21	137,015.35	81,656.43	16,862.25	38,496.67
22	137,098.62	81,730.18	16,869.88	38,498.55
23	137,051.35	81,664.28	16,843.78	38,543.28
24	137,094.38	81,758.60	16,857.48	38,478.30
25	137,065.78	81,674.83	16,851.73	38,539.22
26	137,032.92	81,668.93	16,866.87	38,497.12
27	137,067.57	81,676.20	16,888.33	38,503.03
28	137,022.10	81,690.85	16,833.00	38,498.25
29	137,110.22	81,738.72	16,874.38	38,497.12
30	137,045.48	81,705.40	16,850.37	38,489.72
31	137,061.12	81,731.80	16,848.58	38,480.73
32	137,069.53	81,696.08	16,852.75	38,520.70
33	137,004.48	81,630.30	16,863.45	38,510.73
34	137,066.12	81,682.33	16,859.38	38,524.40
35	137,127.40	81,741.73	16,873.42	38,512.25
36	137,042.28	81,688.97	16,833.85	38,519.47
37	137,104.93	81,692.17	16,860.15	38,552.62
38	137,122.00	81,738.75	16,845.38	38,537.87
39	137,018.62	81,672.60	16,873.92	38,472.10
Average	137,066.20	81,698.01	16,856.13	38,512.06

95% Load Level - 2604 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	0:53:09	1:23:09	0-29	0:30:00
Measurement Interval	1:23:09	1:33:09	30-39	0:10:00

(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	130,338.95	77,677.98	16,019.48	36,641.48
1	130,164.60	77,570.58	16,013.82	36,580.20
2	127,974.82	76,299.43	15,746.88	35,928.50
3	132,348.77	78,928.82	16,247.52	37,172.43
4	130,161.60	77,590.93	16,001.30	36,569.37
5	130,229.02	77,582.83	16,024.75	36,621.43
6	130,202.55	77,583.63	16,025.47	36,593.45
7	130,320.42	77,671.58	16,017.87	36,630.97
8	130,287.60	77,663.07	16,038.10	36,586.43
9	130,213.30	77,628.20	16,013.47	36,571.63
10	130,250.93	77,652.32	16,016.35	36,582.27
11	130,171.67	77,602.72	15,993.35	36,575.60
12	130,237.03	77,618.35	16,038.18	36,580.50
13	130,196.78	77,584.93	16,015.45	36,596.40
14	130,122.53	77,569.12	16,001.17	36,552.25
15	130,257.08	77,604.92	16,017.38	36,634.78
16	130,200.62	77,585.42	16,014.62	36,600.58
17	130,155.03	77,606.07	15,985.65	36,563.32
18	130,246.23	77,665.05	16,012.10	36,569.08
19	130,188.73	77,565.27	16,015.83	36,607.63
20	130,192.78	77,591.72	15,993.38	36,607.68
21	130,220.97	77,571.52	16,028.45	36,621.00
22	130,183.38	77,580.23	16,002.88	36,600.27
23	130,246.15	77,622.48	16,023.95	36,599.72
24	130,147.33	77,557.70	16,013.47	36,576.17
25	130,128.05	77,571.00	16,009.75	36,547.30
26	130,125.75	77,546.15	15,994.62	36,584.98
27	130,244.53	77,608.08	16,032.43	36,604.02
28	130,182.47	77,583.07	16,014.78	36,584.62
29	130,252.03	77,660.55	16,021.07	36,570.42
30	130,201.70	77,622.90	15,988.43	36,590.37
31	130,185.17	77,565.05	16,029.48	36,590.63
32	130,168.42	77,553.60	16,013.33	36,601.48
33	130,221.45	77,582.58	16,046.90	36,591.97
34	130,165.83	77,598.47	15,993.12	36,574.25
35	130,261.18	77,665.33	16,005.22	36,590.63
36	130,129.73	77,551.07	16,024.02	36,554.65
37	130,196.32	77,598.28	15,994.28	36,603.75
38	130,168.95	77,564.63	16,006.05	36,598.27
39	130,246.38	77,650.02	16,014.38	36,581.98
Average	130,194.51	77,595.19	16,011.52	36,587.80

Response Time Ramp Distribution (IOPS) Data *(cont.)*

90% Load Level - 2467 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	1:36:30	2:06:30	0-29	0:30:00
Measurement Interval	2:06:30	2:16:30	30-39	0:10:00
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	123,466.78	73,582.18	15,221.17	34,663.43
1	123,333.68	73,538.53	15,176.57	34,618.58
2	123,369.80	73,517.62	15,193.53	34,658.65
3	123,338.95	73,495.75	15,167.28	34,675.92
4	123,390.72	73,515.77	15,179.38	34,695.57
5	123,341.12	73,486.82	15,200.47	34,653.83
6	123,392.75	73,515.28	15,191.00	34,686.47
7	123,309.38	73,477.63	15,166.97	34,664.78
8	123,352.13	73,513.80	15,186.72	34,651.62
9	123,338.57	73,510.42	15,169.82	34,658.33
10	123,396.20	73,518.55	15,188.90	34,688.75
11	123,329.52	73,577.93	15,148.70	34,602.88
12	123,444.75	73,596.18	15,162.93	34,685.63
13	123,313.18	73,511.15	15,173.30	34,628.73
14	123,377.73	73,553.67	15,180.67	34,643.40
15	123,333.87	73,492.10	15,198.55	34,643.22
16	123,286.17	73,442.87	15,182.12	34,661.18
17	123,428.88	73,559.32	15,205.13	34,664.43
18	123,300.83	73,449.10	15,192.38	34,659.35
19	123,275.78	73,508.03	15,149.03	34,618.72
20	123,426.68	73,550.18	15,201.37	34,675.13
21	123,437.22	73,528.05	15,186.63	34,722.53
22	123,351.80	73,507.85	15,199.02	34,644.93
23	123,359.08	73,496.75	15,176.10	34,686.23
24	123,290.90	73,464.58	15,157.55	34,668.77
25	123,343.23	73,511.83	15,152.47	34,678.93
26	123,273.95	73,474.50	15,146.60	34,652.85
27	123,308.72	73,508.37	15,158.73	34,641.62
28	123,302.28	73,467.57	15,162.43	34,672.28
29	123,338.05	73,501.40	15,171.42	34,665.23
30	123,466.05	73,598.48	15,174.67	34,692.90
31	123,371.83	73,514.27	15,181.43	34,676.13
32	123,321.73	73,488.62	15,187.67	34,645.45
33	123,363.87	73,514.25	15,157.78	34,691.83
34	123,312.68	73,473.95	15,177.58	34,661.15
35	123,299.22	73,462.82	15,202.48	34,633.92
36	123,336.88	73,496.13	15,177.38	34,663.37
37	123,362.43	73,566.20	15,173.37	34,622.87
38	123,336.47	73,509.15	15,152.68	34,674.63
39	123,380.22	73,509.92	15,180.30	34,690.00
Average	123,355.14	73,513.38	15,176.54	34,665.23

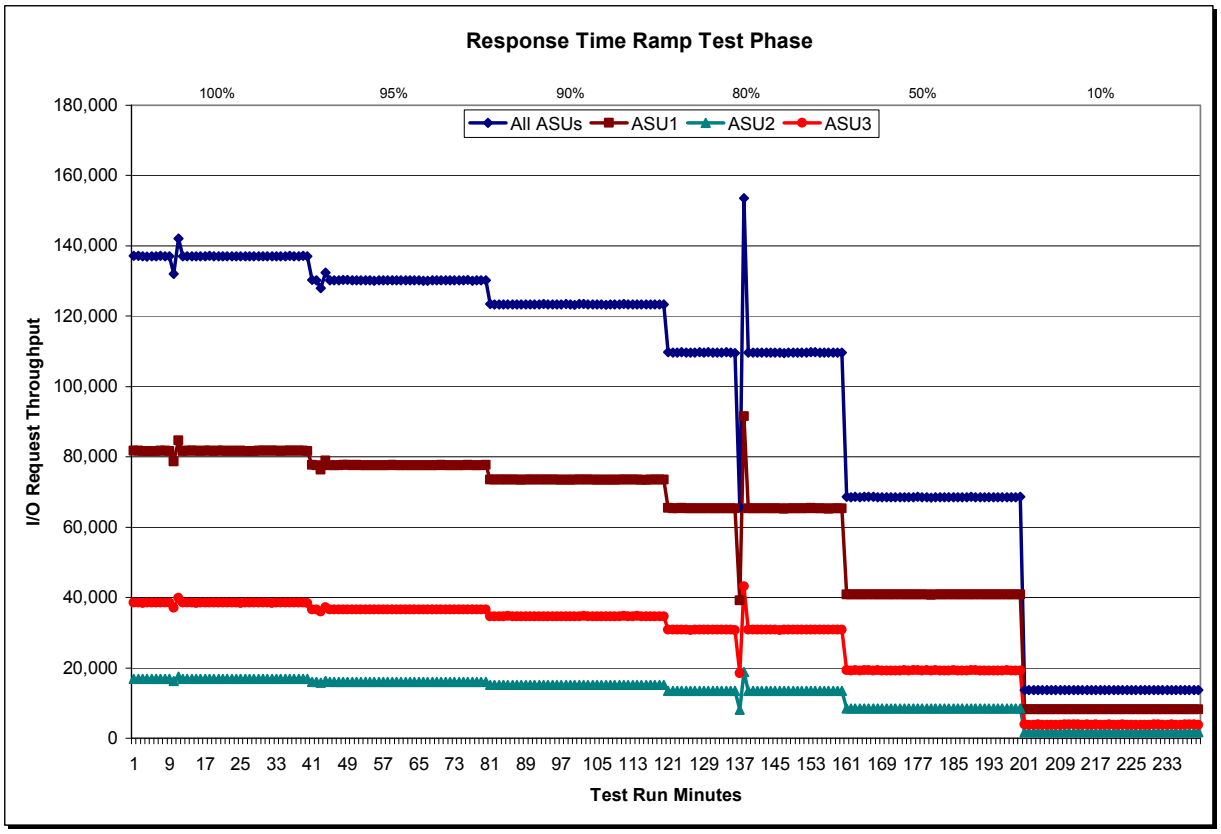
80% Load Level - 2193 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	2:19:24	2:49:24	0-29	0:30:00
Measurement Interval	2:49:24	2:59:24	30-39	0:10:00
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	109,775.12	65,464.43	13,486.47	30,824.22
1	109,655.85	65,357.55	13,482.93	30,815.37
2	109,695.78	65,399.80	13,494.62	30,801.37
3	109,748.05	65,409.35	13,509.83	30,828.87
4	109,642.13	65,355.22	13,500.55	30,786.37
5	109,621.73	65,378.60	13,468.72	30,774.42
6	109,682.03	65,318.72	13,515.88	30,847.43
7	109,719.23	65,394.27	13,519.87	30,805.10
8	109,622.95	65,327.42	13,493.77	30,801.77
9	109,705.78	65,387.95	13,496.48	30,821.35
10	109,613.88	65,344.42	13,488.95	30,780.52
11	109,605.75	65,329.55	13,442.90	30,833.30
12	109,666.60	65,360.93	13,493.47	30,812.20
13	109,701.88	65,381.23	13,469.45	30,851.20
14	109,681.40	65,376.60	13,484.93	30,819.87
15	109,552.35	65,319.02	13,498.60	30,734.73
16	65,740.40	39,169.47	8,094.82	18,476.12
17	153,529.72	91,478.67	18,865.42	43,185.63
18	109,689.73	65,382.77	13,472.43	30,834.53
19	109,686.90	65,361.80	13,516.13	30,808.97
20	109,667.65	65,389.27	13,478.72	30,799.67
21	109,689.08	65,366.43	13,509.23	30,813.42
22	109,629.30	65,330.02	13,488.97	30,810.32
23	109,607.72	65,321.80	13,461.67	30,824.25
24	109,681.33	65,361.23	13,460.20	30,859.90
25	109,594.75	65,349.52	13,482.13	30,763.10
26	109,530.67	65,278.85	13,447.33	30,804.48
27	109,673.80	65,389.50	13,484.90	30,799.40
28	109,584.57	65,340.70	13,462.30	30,781.57
29	109,646.05	65,360.85	13,483.28	30,801.92
30	109,600.60	65,327.25	13,480.70	30,792.65
31	109,676.30	65,397.55	13,477.82	30,800.93
32	109,713.18	65,429.53	13,493.58	30,790.07
33	109,718.50	65,354.02	13,505.10	30,859.38
34	109,625.27	65,323.20	13,486.22	30,815.85
35	109,661.82	65,336.52	13,508.72	30,816.58
36	109,600.60	65,274.27	13,496.88	30,829.45
37	109,615.53	65,322.35	13,497.95	30,795.23
38	109,625.95	65,333.42	13,494.28	30,798.25
39	109,681.30	65,328.48	13,510.87	30,841.95
Average	109,651.91	65,342.66	13,495.21	30,814.04

Response Time Ramp Distribution (IOPS) Data *(cont.)*

50% Load Level - 1371 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	3:01:13	3:31:13	0-29	0:30:00
Measurement Interval	3:31:13	3:41:13	30-39	0:10:00
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	68,587.28	40,880.88	8,433.00	19,273.40
1	68,542.93	40,846.85	8,432.00	19,264.08
2	68,591.75	40,884.40	8,434.55	19,272.80
3	68,484.10	40,817.83	8,403.05	19,263.22
4	68,623.38	40,905.32	8,439.28	19,278.78
5	68,589.82	40,863.70	8,445.85	19,280.27
6	68,614.02	40,914.23	8,437.08	19,262.70
7	68,582.82	40,854.60	8,449.40	19,278.82
8	68,531.15	40,830.25	8,439.05	19,261.85
9	68,558.02	40,873.30	8,435.20	19,249.52
10	68,477.58	40,820.87	8,403.23	19,253.48
11	68,508.53	40,817.72	8,440.93	19,249.88
12	68,553.27	40,855.88	8,439.77	19,257.62
13	68,561.63	40,850.78	8,422.97	19,287.88
14	68,472.80	40,823.25	8,428.22	19,221.33
15	68,558.53	40,820.92	8,458.33	19,279.28
16	68,609.00	40,891.27	8,435.97	19,281.77
17	68,536.40	40,874.93	8,443.37	19,218.10
18	68,573.52	40,853.15	8,423.70	19,296.67
19	68,454.73	40,799.63	8,415.85	19,239.25
20	68,538.32	40,837.88	8,425.32	19,275.12
21	68,485.05	40,817.13	8,402.73	19,265.18
22	68,540.02	40,866.13	8,431.03	19,242.85
23	68,531.80	40,868.25	8,426.12	19,237.43
24	68,557.70	40,840.07	8,444.47	19,273.17
25	68,549.75	40,862.00	8,431.42	19,256.33
26	68,516.00	40,836.95	8,427.20	19,251.85
27	68,528.08	40,827.20	8,446.25	19,254.63
28	68,621.23	40,874.08	8,443.73	19,303.42
29	68,555.22	40,833.95	8,439.93	19,281.33
30	68,515.57	40,824.77	8,431.33	19,259.47
31	68,544.73	40,860.82	8,411.58	19,272.33
32	68,566.82	40,860.77	8,445.28	19,260.77
33	68,554.62	40,891.00	8,407.68	19,255.93
34	68,530.77	40,854.60	8,433.53	19,242.63
35	68,535.60	40,859.62	8,419.92	19,256.07
36	68,531.38	40,825.72	8,431.50	19,274.17
37	68,516.85	40,829.38	8,430.65	19,256.82
38	68,537.52	40,846.17	8,434.48	19,256.87
39	68,592.07	40,883.95	8,443.93	19,264.18
Average	68,542.59	40,853.68	8,428.99	19,259.92

10% Load Level - 274 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	3:41:55	4:11:55	0-29	0:30:00
Measurement Interval	4:11:55	4:21:55	30-39	0:10:00
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	13,713.88	8,171.53	1,687.53	3,854.82
1	13,671.52	8,140.98	1,682.37	3,848.17
2	13,705.08	8,175.72	1,682.88	3,846.48
3	13,697.97	8,157.20	1,685.10	3,855.67
4	13,685.82	8,155.02	1,681.63	3,849.17
5	13,684.43	8,161.38	1,682.72	3,840.33
6	13,708.43	8,176.77	1,684.92	3,846.75
7	13,678.42	8,144.27	1,679.95	3,854.20
8	13,685.12	8,151.57	1,685.30	3,848.25
9	13,711.87	8,175.55	1,676.85	3,859.47
10	13,721.50	8,175.35	1,685.48	3,860.67
11	13,716.02	8,163.95	1,690.08	3,861.98
12	13,707.83	8,165.17	1,687.03	3,855.63
13	13,695.10	8,156.15	1,692.38	3,846.57
14	13,722.42	8,164.38	1,687.20	3,870.83
15	13,679.60	8,167.48	1,681.67	3,830.45
16	13,714.47	8,170.18	1,684.78	3,859.50
17	13,695.15	8,154.07	1,688.07	3,853.02
18	13,704.05	8,176.45	1,676.18	3,851.42
19	13,726.87	8,176.27	1,688.53	3,862.07
20	13,711.67	8,184.20	1,686.63	3,840.83
21	13,691.12	8,156.22	1,685.30	3,849.60
22	13,687.55	8,147.02	1,685.98	3,854.55
23	13,684.30	8,155.87	1,684.82	3,843.62
24	13,694.53	8,158.07	1,682.22	3,854.25
25	13,697.68	8,173.25	1,672.37	3,852.07
26	13,686.00	8,159.72	1,684.77	3,841.52
27	13,692.47	8,171.30	1,681.92	3,839.25
28	13,676.40	8,158.02	1,678.77	3,839.62
29	13,708.25	8,169.35	1,683.55	3,855.35
30	13,709.27	8,164.33	1,683.45	3,861.48
31	13,670.82	8,143.87	1,681.05	3,845.90
32	13,686.62	8,158.82	1,693.97	3,833.83
33	13,718.17	8,179.32	1,682.47	3,856.38
34	13,696.50	8,164.87	1,685.55	3,846.08
35	13,711.82	8,176.85	1,681.23	3,853.73
36	13,704.12	8,158.15	1,690.55	3,855.42
37	13,697.12	8,157.40	1,682.52	3,857.20
38	13,722.23	8,168.37	1,695.57	3,858.30
39	13,708.27	8,172.43	1,688.23	3,847.60
Average	13,702.49	8,164.44	1,686.46	3,851.59

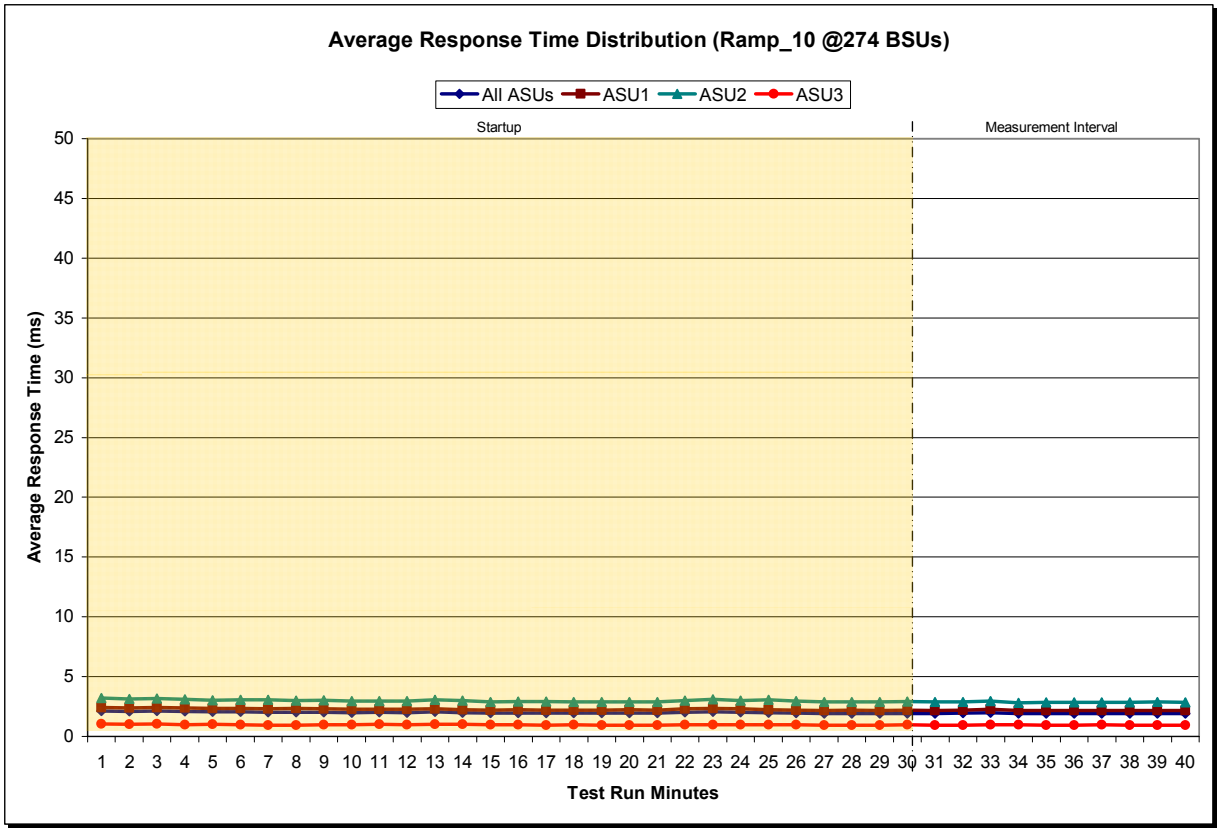
Response Time Ramp Distribution (IOPS) Graph



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

274 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	3:41:55	4:11:55	0-29	0:30:00
Measurement Interval	4:11:55	4:21:55	29-39	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	2.13	2.42	3.20	1.03
1	2.08	2.37	3.12	1.01
2	2.11	2.40	3.17	1.05
3	2.07	2.38	3.11	0.96
4	2.04	2.33	3.03	1.01
5	2.05	2.34	3.05	0.98
6	2.02	2.31	3.04	0.95
7	2.02	2.32	2.99	0.95
8	2.03	2.31	3.04	0.99
9	1.99	2.27	2.95	0.97
10	2.01	2.28	2.96	1.02
11	1.99	2.27	2.97	0.96
12	2.04	2.32	3.07	1.00
13	1.97	2.22	2.97	1.00
14	1.94	2.21	2.90	0.96
15	1.96	2.22	2.93	0.97
16	1.93	2.20	2.91	0.95
17	1.93	2.20	2.89	0.95
18	1.94	2.21	2.89	0.95
19	1.94	2.22	2.90	0.95
20	1.93	2.19	2.87	0.95
21	2.01	2.29	3.00	0.97
22	2.05	2.34	3.09	0.99
23	2.01	2.31	2.97	0.96
24	1.98	2.24	3.04	0.97
25	1.94	2.20	2.95	0.97
26	1.92	2.18	2.87	0.95
27	1.92	2.18	2.87	0.95
28	1.91	2.16	2.86	0.95
29	1.93	2.18	2.91	0.95
30	1.92	2.18	2.89	0.94
31	1.93	2.19	2.89	0.95
32	1.99	2.26	2.95	0.99
33	1.90	2.16	2.79	0.97
34	1.90	2.15	2.83	0.95
35	1.91	2.16	2.85	0.95
36	1.90	2.15	2.85	0.96
37	1.90	2.16	2.84	0.95
38	1.91	2.16	2.88	0.95
39	1.90	2.15	2.85	0.95
Average	***1.91***	***2.17***	***2.86***	***0.95***

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 5.3.13.2

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 5.3.13.3

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	0.0699	0.2100	0.0181	0.0700	0.0350	0.2811
COV	0.005	0.001	0.004	0.002	0.007	0.004	0.007	0.002

Repeatability Test

Clause 5.4.5

The Repeatability Test demonstrates the repeatability and reproducibility of the SPC-1 IOPS™ primary metric and 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 9.4.3.7.4

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 105.

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	***137,066.20***
Repeatability Test Phase 1	137,069.80
Repeatability Test Phase 2	137,032.15

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	***1.91 ms***
Repeatability Test Phase 1	1.91 ms
Repeatability Test Phase 2	1.94 ms

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 minus one (1) millisecond (ms)..

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

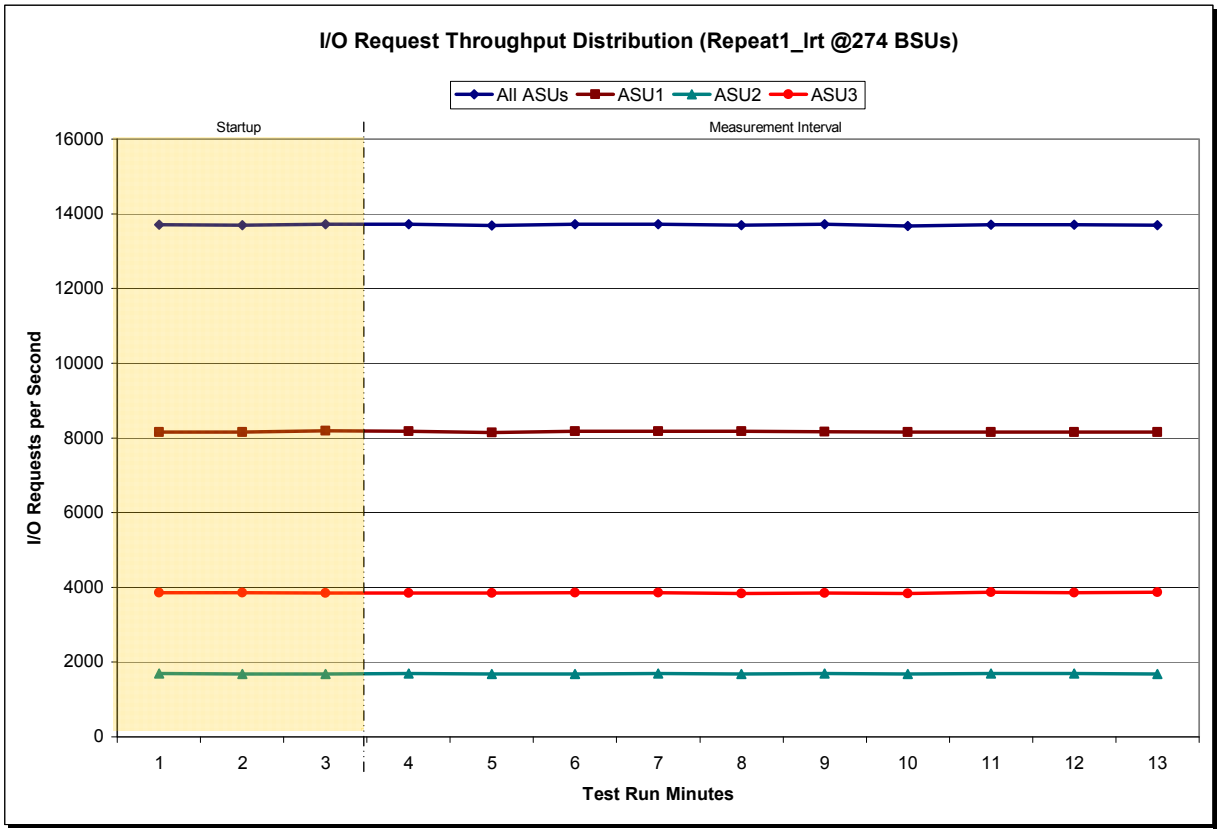
Repeatability Test Phase 1, Test Run 1 *(LRT)*
Repeatability Test Phase 1, Test Run 2 *(IOPS)*
Repeatability Test Phase 2, Test Run 1 *(LRT)*
Repeatability Test Phase 2, Test Run 2 *(IOPS)*

Repeatability 1 LRT – I/O Request Throughput Distribution Data

274 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	4:24:00	4:27:00	0-2	0:03:00
Measurement Interval	4:27:00	4:37:00	3-12	0:10:00

60 second intervals	All ASUs	ASU1	ASU2	ASU3
0	13,703.10	8,157.42	1,688.32	3,857.37
1	13,693.68	8,152.15	1,680.82	3,860.72
2	13,717.32	8,187.10	1,681.73	3,848.48
3	13,716.68	8,176.78	1,689.82	3,850.08
4	13,681.82	8,146.00	1,685.35	3,850.47
5	13,721.43	8,172.95	1,687.12	3,861.37
6	13,720.53	8,177.52	1,688.97	3,854.05
7	13,698.15	8,174.68	1,684.37	3,839.10
8	13,716.62	8,172.30	1,691.65	3,852.67
9	13,677.95	8,158.97	1,681.85	3,837.13
10	13,710.23	8,154.95	1,689.97	3,865.32
11	13,708.50	8,158.50	1,696.23	3,853.77
12	13,701.55	8,153.43	1,681.77	3,866.35
Average	***13,705.35***	***8,164.61***	***1,687.71***	***3,853.03***

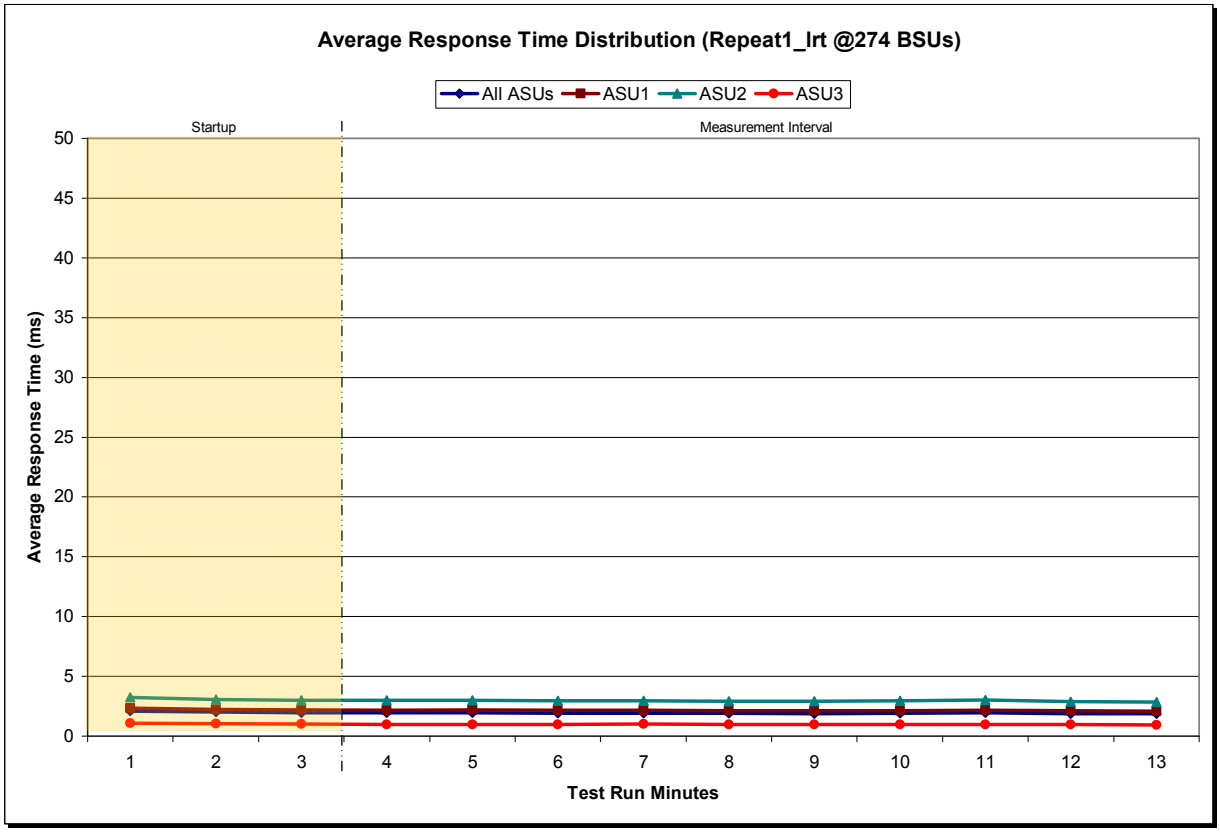
Repeatability 1 LRT – I/O Request Throughput Distribution Graph



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

274 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	4:24:00	4:27:00	0-2	0:03:00
Measurement Interval	4:27:00	4:37:00	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	2.10	2.35	3.23	1.09
1	2.00	2.24	3.05	1.04
2	1.96	2.20	2.98	1.00
3	1.94	2.18	2.98	0.98
4	1.94	2.18	2.98	0.97
5	1.91	2.15	2.94	0.97
6	1.92	2.15	2.96	1.00
7	1.91	2.14	2.92	0.97
8	1.89	2.11	2.93	0.96
9	1.89	2.11	2.95	0.96
10	1.93	2.15	3.03	0.97
11	1.88	2.11	2.89	0.95
12	1.85	2.08	2.85	0.95
Average	***1.91***	***2.13***	***2.94***	***0.97***

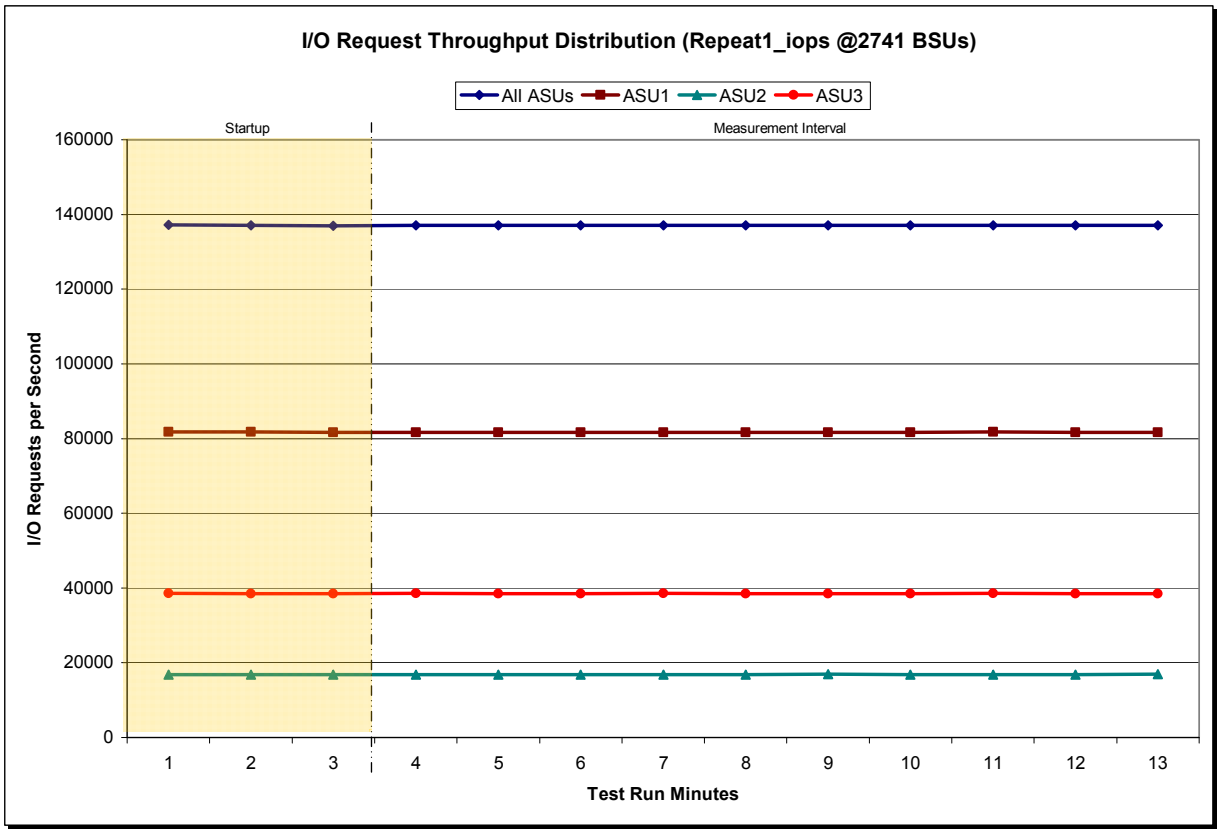
Repeatability 1 LRT –Average Response Time (ms) Distribution Graph



Repeatability 1 IOPS – I/O Request Throughput Distribution Data

2741 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	4:39:32	4:42:33	0-2	0:03:01
Measurement Interval	4:42:33	4:52:33	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	137,142.35	81,761.00	16,818.67	38,562.68
1	137,111.33	81,753.83	16,829.32	38,528.18
2	136,944.97	81,664.45	16,855.72	38,424.80
3	137,119.32	81,700.08	16,853.75	38,565.48
4	137,039.23	81,664.18	16,861.52	38,513.53
5	137,077.07	81,714.37	16,842.00	38,520.70
6	137,069.53	81,632.95	16,869.63	38,566.95
7	137,057.27	81,675.62	16,863.15	38,518.50
8	137,067.95	81,702.30	16,876.02	38,489.63
9	137,071.12	81,709.58	16,873.10	38,488.43
10	137,121.32	81,735.92	16,831.88	38,553.52
11	137,028.67	81,690.15	16,830.35	38,508.17
12	137,046.57	81,689.38	16,876.43	38,480.75
Average	***137,069.80***	***81,691.45***	***16,857.78***	***38,520.57***

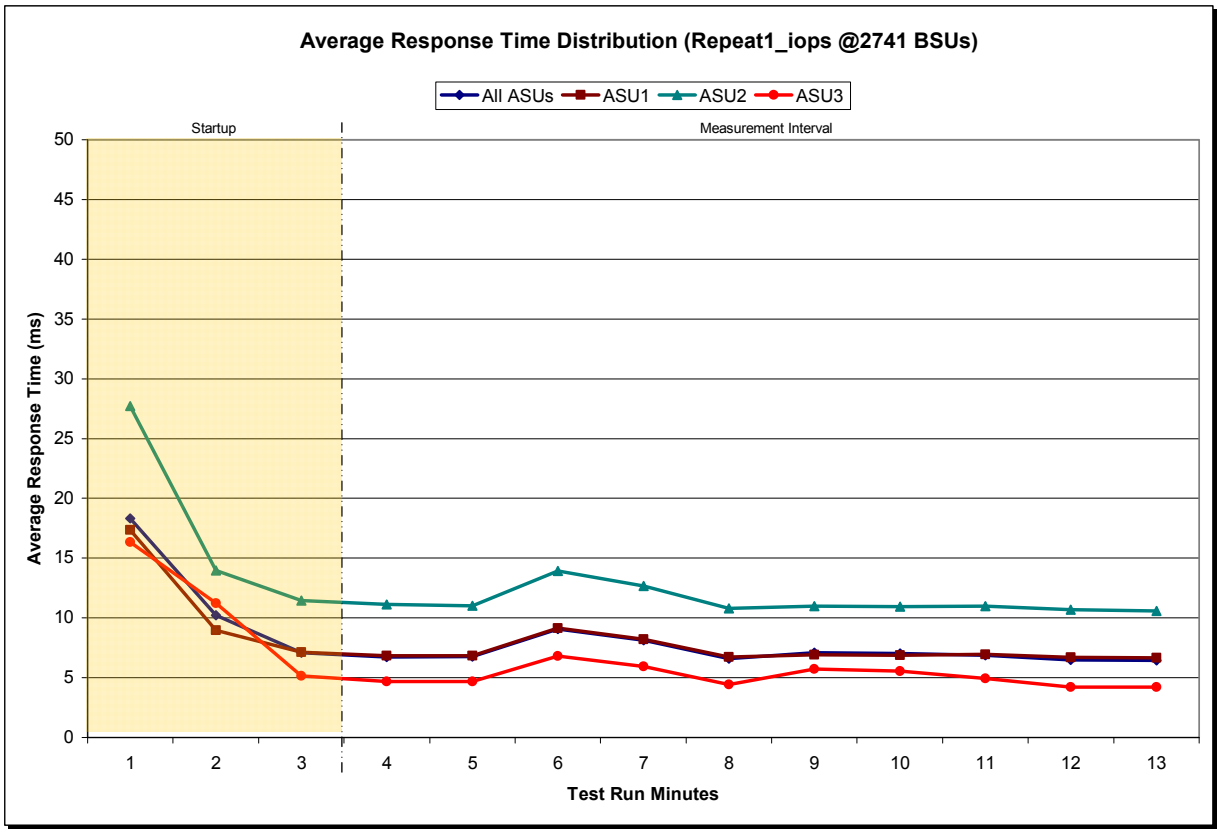
Repeatability 1 IOPS – I/O Request Throughput Distribution Graph



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

2741 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	4:39:32	4:42:33	0-2	0:03:01
Measurement Interval	4:42:33	4:52:33	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	18.34	17.34	27.73	16.36
1	10.21	8.95	13.96	11.25
2	7.10	7.12	11.45	5.14
3	6.75	6.82	11.14	4.66
4	6.76	6.84	11.03	4.70
5	9.08	9.15	13.95	6.80
6	8.12	8.21	12.68	5.95
7	6.59	6.73	10.81	4.43
8	7.09	6.93	10.99	5.74
9	7.00	6.88	10.94	5.55
10	6.87	6.94	10.97	4.93
11	6.49	6.69	10.69	4.22
12	6.46	6.67	10.60	4.20
Average	***7.12***	***7.19***	***11.38***	***5.12***

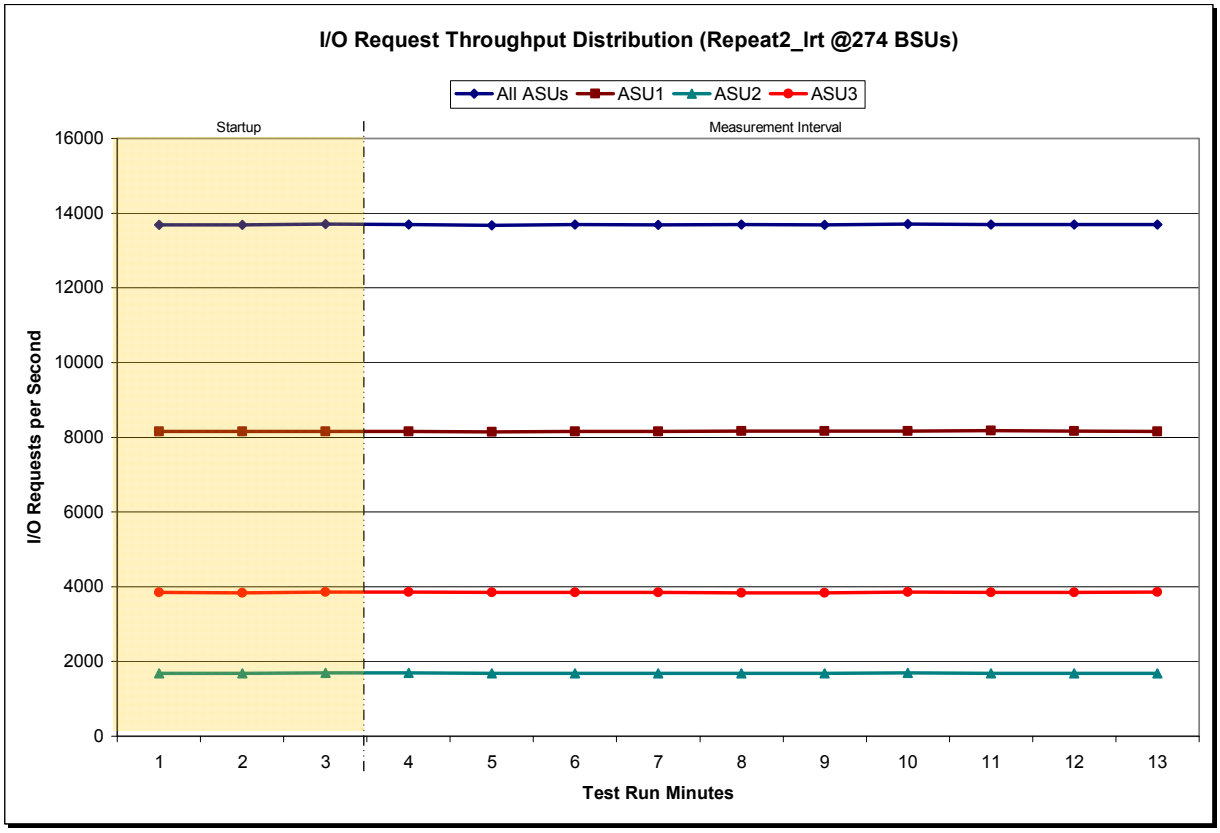
Repeatability 1 IOPS –Average Response Time (ms) Distribution Graph



Repeatability 2 LRT – I/O Request Throughput Distribution Data

274 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	4:54:40	4:57:40	0-2	0:03:00
Measurement Interval	4:57:40	5:07:40	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	13,686.78	8,150.87	1,685.45	3,850.47
1	13,682.80	8,158.78	1,686.32	3,837.70
2	13,708.70	8,157.75	1,695.47	3,855.48
3	13,695.58	8,152.72	1,689.00	3,853.87
4	13,674.22	8,139.90	1,683.75	3,850.57
5	13,694.30	8,160.85	1,683.17	3,850.28
6	13,681.07	8,151.67	1,679.77	3,849.63
7	13,694.75	8,171.50	1,684.42	3,838.83
8	13,687.15	8,164.97	1,682.07	3,840.12
9	13,713.25	8,162.98	1,690.15	3,860.12
10	13,692.58	8,172.82	1,677.80	3,841.97
11	13,698.40	8,165.57	1,683.45	3,849.38
12	13,695.40	8,153.90	1,682.20	3,859.30
Average	***13,692.67***	***8,159.69***	***1,683.58***	***3,849.41***

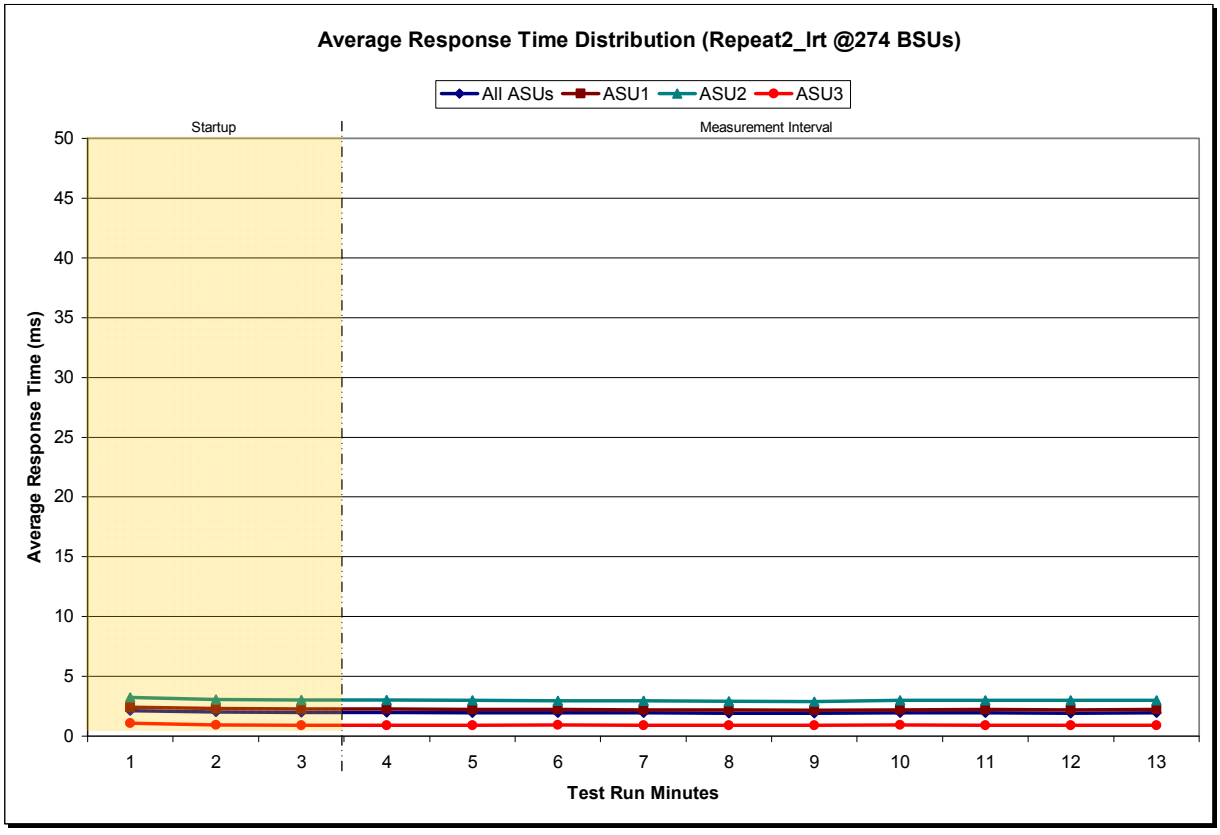
Repeatability 2 LRT – I/O Request Throughput Distribution Graph



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

274 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	4:54:40	4:57:40	0-2	0:03:00
Measurement Interval	4:57:40	5:07:40	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	2.14	2.41	3.24	1.08
1	2.01	2.30	3.06	0.94
2	1.99	2.28	3.02	0.92
3	1.97	2.27	3.01	0.90
4	1.95	2.24	2.98	0.89
5	1.96	2.24	2.97	0.94
6	1.93	2.21	2.95	0.88
7	1.92	2.21	2.92	0.89
8	1.90	2.17	2.89	0.89
9	1.94	2.19	2.99	0.94
10	1.95	2.23	2.99	0.91
11	1.92	2.18	2.99	0.90
12	1.95	2.23	3.00	0.91
Average	***1.94***	***2.22***	***2.97***	***0.91***

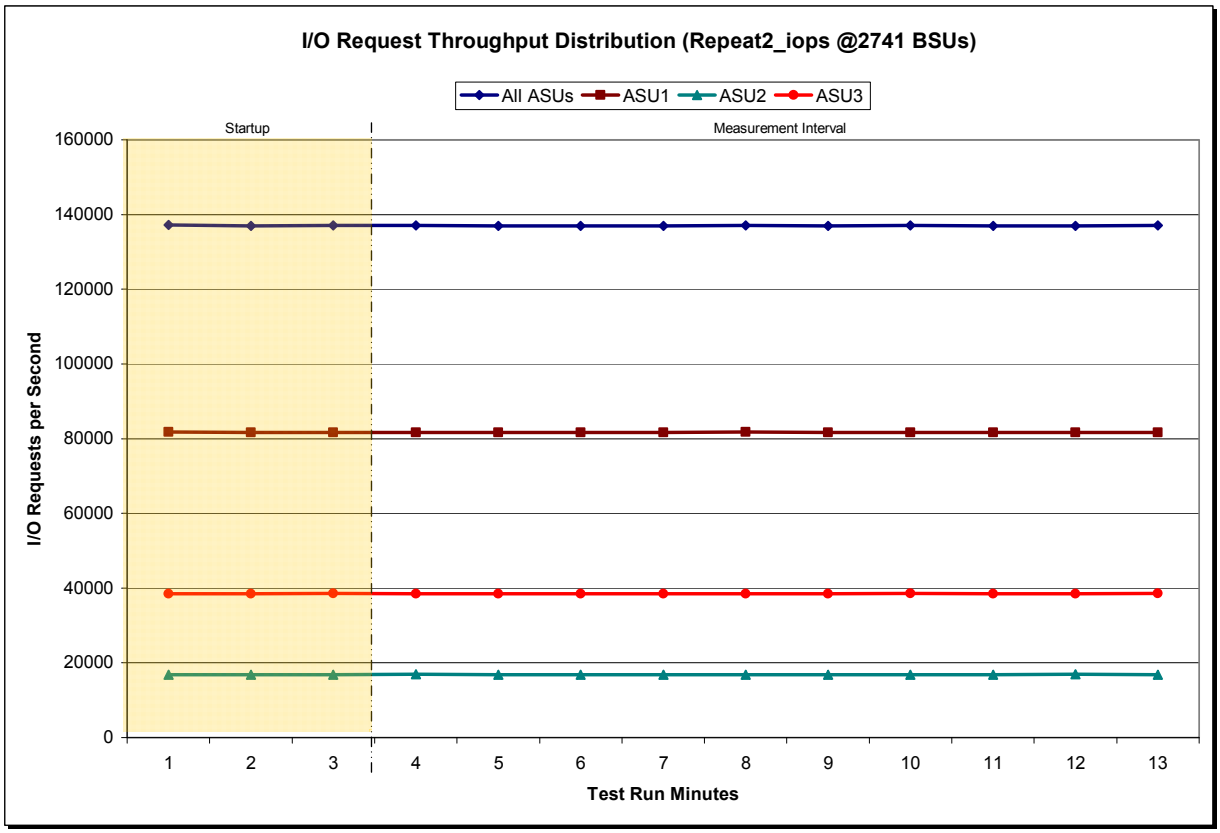
Repeatability 2 LRT –Average Response Time (ms) Distribution Graph



Repeatability 2 IOPS – I/O Request Throughput Distribution Data

2741 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	5:10:12	5:13:13	0-2	0:03:01
Measurement Interval	5:13:13	5:23:13	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	137,154.62	81,758.72	16,865.65	38,530.25
1	137,010.82	81,646.02	16,837.22	38,527.58
2	137,067.50	81,695.40	16,823.33	38,548.77
3	137,090.27	81,679.73	16,883.90	38,526.63
4	136,981.18	81,663.10	16,830.47	38,487.62
5	136,974.97	81,616.48	16,857.00	38,501.48
6	137,015.28	81,643.82	16,871.48	38,499.98
7	137,119.93	81,734.62	16,858.35	38,526.97
8	136,993.93	81,631.87	16,846.42	38,515.65
9	137,025.08	81,630.33	16,852.70	38,542.05
10	137,000.40	81,620.18	16,867.43	38,512.78
11	137,015.77	81,653.40	16,877.50	38,484.87
12	137,104.65	81,699.48	16,866.00	38,539.17
Average	***137,032.15***	***81,657.30***	***16,861.13***	***38,513.72***

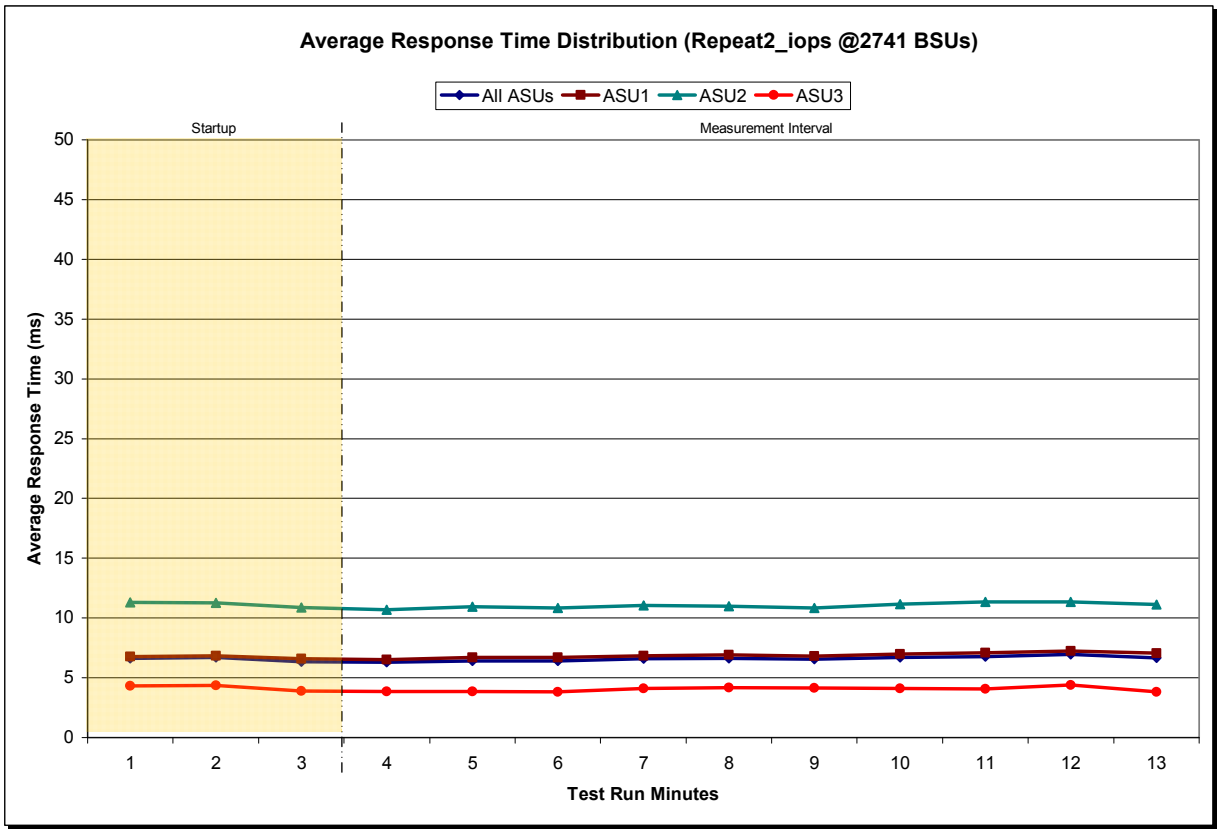
Repeatability 2 IOPS – I/O Request Throughput Distribution Graph



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

2741 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	5:10:12	5:13:13	0-2	0:03:01
Measurement Interval	5:13:13	5:23:13	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	6.63	6.75	11.32	4.32
1	6.68	6.83	11.28	4.36
2	6.35	6.58	10.89	3.87
3	6.29	6.53	10.70	3.87
4	6.41	6.69	10.95	3.84
5	6.39	6.69	10.83	3.83
6	6.59	6.84	11.05	4.09
7	6.64	6.91	10.98	4.17
8	6.55	6.81	10.82	4.15
9	6.69	6.99	11.16	4.11
10	6.77	7.11	11.33	4.06
11	6.95	7.24	11.36	4.39
12	6.64	7.05	11.13	3.82
Average	***6.59***	***6.88***	***11.03***	***4.03***

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.*

Clauses 5.1.10 and 5.3.13.2

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 5.3.13.3

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	0.2808	0.0700	0.2099	0.0181	0.0700	0.0351	0.2811
COV	0.006	0.002	0.004	0.002	0.007	0.002	0.005	0.002

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	0.2809	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
COV	0.002	0.000	0.001	0.001	0.002	0.002	0.002	0.001

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	0.0351	0.2812	0.0700	0.2097	0.0181	0.0699	0.0350	0.2811
COV	0.005	0.002	0.004	0.002	0.007	0.002	0.006	0.002

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	0.2811
COV	0.002	0.001	0.002	0.000	0.002	0.001	0.002	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.*

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 105.

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	640,545
Total Number of Logical Blocks Verified	631,527
Total Number of Logical Blocks that Failed Verification	0
Time Duration for Writing Test Logical Blocks	10 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 9.2.4.9

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 Oracle Sun ZFS Storage 7420c Appliance as documented in this Full Disclosure Report will become available January 1, 2012 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 found in the Priced Storage Configuration Pricing section on page 14.

TESTED STORAGE CONFIGURATION (TSC) AND PRICED STORAGE CONFIGURATION DIFFERENCES

Clause 9.4.3.3.7

The Executive Summary shall contain a pricing a list of all differenced 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 14.

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 Onsite Audit of the Oracle Sun ZFS Storage 7420c Appliance .

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: This level will ensure data protection in the event of a single point of failure of any configured storage device. A brief description of the data protection utilized is included in the Executive Summary.

Unprotected: No claim of data protection is asserted in the event of a single point of failure.

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.

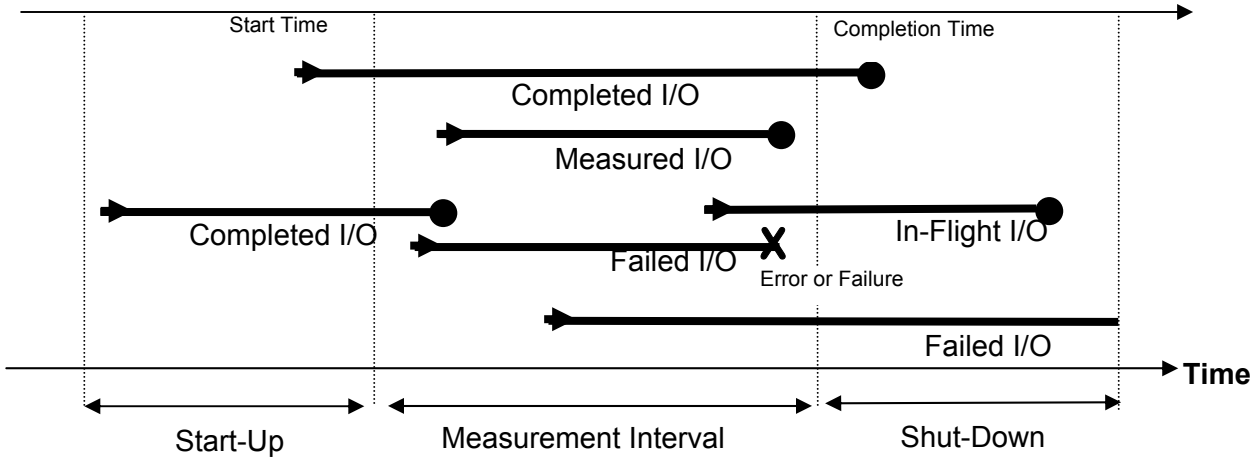
Test: A collection of Test Phases and or Test Runs sharing a common objective.

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

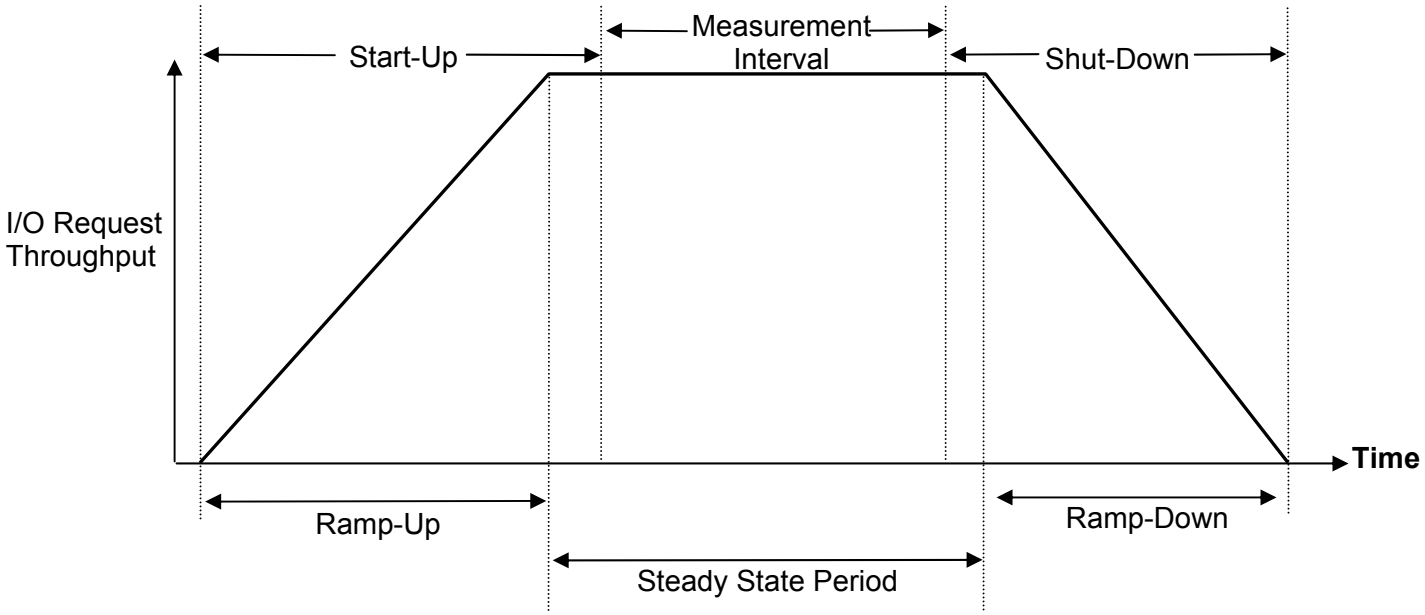
period, Shut-Down period, and Ramp-Down period as illustrated in the "SPC-1 Test Run Components" below. All SPC-1 Test Runs shall have a Steady State period and a Measurement Interval.

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

The following entries were added to the ***/etc/system*** file on the Oracle SPARC T3-1 Host System:

```
# Defines Max number of Page IO requests that can be queued.  Default 256
set maxpgio = 65536

# Maximum number of pages per sec that the system looks at when pressure is
# highest.  Default  The lessor of 64 Mbytes and 1/2 of physical memory
set fastscan = 65536

#ufs_HW specifies High Water number of bytes outstanding on a single file.
# default 8 x 1024 X 1024 for ufs_LW and 16 X 1024 X 1024 for ufs_HW
set ufs:ufs_HW = 20971520

#ufs_LW is the Low Water number of bytes outstanding on a single file
set ufs:ufs_LW = 15728640

# Along with tune_t_fsflushr, autoup controls the amount of memory
# examined for dirty pages in each invocation and frequency
# of file system synchronizing operations default 30
set autoup = 1024

# Defines the time between dirty page flushes in seconds. Default 1
set tune_t_fsflushr = 1

# Sets the depth of the Syncq (number of messages) before destination
#  STREAMS queue generates a QFULL message
set sq_max_size = 100

# Max outstanding number of requests to a Lun target
set ssd:ssd_max_throttle=1000

# Max physical bytes to be allocated in a file
set maxphys = 8388608

# Second max SD max outstanding number of requests to a lun target
set sd:sd_max_throttle=1000
set ssd:sd_max_throttle=1000

# Two entries, one for SPARC and x86
set sd:sd_xbuf_active_limit=2048
set ssd:ssd_xbuf_active_limit=2048

#Improve the resolution of the cpu clock for better IO driver sleeps
# default 0 (disabled)
set hires_tick=1
```

APPENDIX C: TESTED STORAGE CONFIGURATION (TSC) CREATION

Assign Host Names and IP Addresses

The Quick Start instructions shipped with the 7420c provide the details for assigning host names and IP addresses.

Configure the Tested Storage Configuration (TSC)

The 7420c includes storage controllers, referenced in this section as 'A' and 'B'. The TSC will be configured using the scripts described below, which are performed on the Host System via the "root" user. All referenced scripts will appear at the end of the appendix.

Build the 7420 Cluster

The script, **Build-7420-Cluster.sh**, will invoke scripts to:

- build RAID pools on each controller
- create 28 volumes on each controller
- format and align LUNs

Build RAID Pools

The two scripts, **NEW-35d_build_12tray_4pool_1L_a.sh** and **NEW-35d_build_12tray_4pool_1L_b.sh**, will create four RAID pools per controller.

Create Volumes

The two scripts, **build-vols-12T-4P_a.sh** and **build-vols-12T-4P_b.sh**, will create 28 volumes on each controller. All 56 volumes are accessible by both controllers.

Format and Align LUNs

The 56 volumes created in the previous step are viewed as 56 LUNs by the Host System. The two scripts, **format_12T_4P_a.sh** and **format_12T_4P_b.sh**, will invoke the scripts, **get_luns44.sh**, **get_luns4.sh**, **get_luns33.sh**, **get_luns3.sh**, **get_luns22.sh**, **get_luns2.sh**, **get_luns11.sh** and **get_luns1.sh**, to obtain a volume listing from the two controllers in order to format and align the LUNs.

Create the SPC-1 Logical Volumes

The final script, **make_meta_12T_4P_Cluster.sh**, will create three SPC-1 Logical Volumes, one for each of the three SPC-1 ASUs, using the 48 LUNs prefixed with **asu1_**, **asu2_**, and **asu3_**. The remaining 8 LUNs, labeled **asu_4**, will be used to create a fourth logical volume used ~~as an offset to align~~ to optimally place the three SPC-1 Logical Volumes for free space management. That fourth logical volume is deleted prior to benchmark execution.

The **make_meta_12T_4P_Cluster.sh** script will invoke the scripts, **get_luns44.sh**, **get_luns4.sh**, **get_luns33.sh**, **get_luns3.sh**, **get_luns22.sh**, **get_luns2.sh**, **get_luns11.sh**

and **get_luns1.sh**, to obtain a volume listing from the two controllers in order to create the four logical volumes.

In addition, this final script is used to create various documentation listings and configuration files used in the benchmark execution.

Build-7420-Cluster.sh

```
#!/bin/bash


#
# Remove pools and create pools for SPC1 on the "A" node
NEW-35d_build_12tray_4pool_1L_a.sh
sleep 10
## Remove pools and create pools for SPC1 on the "B" node
NEW-35d_build_12tray_4pool_1L_b.sh
sleep 10
#
#
# Build volumes on both Storage Nodes
build-vols-12T-4P_a.sh
sleep 10
build-vols-12T-4P_b.sh
#
#
cfgadm -la
devfsadm -C
devfsadm
#
# Fomat new Luns and align
format_12T_4P_a.sh
sleep 10
format_12T_4P_b.sh
#
#
# Create Metadevices and create prtvtoc.txt file
# Also creates A-head_hw_list.txt and B-head_hw_list.txt
# and all Parameter files
make_meta_12T_4P_Cluster.sh
#
#
```

NEW-35d_build_12tray_4pool_1L_a.sh

```
#!/bin/bash
#
# Build four pools on a 12-tray 7000 cluster
# Assumption is 4 pools currently present
#
# 4 Pools = ASU1/2/3 32 drives | 2 log | 1 cache
#
AR_HOSTNAME=sbm-q112-7420a

echo "Tesing connectivity to $AR_HOSTNAME..."

# Check that host is up
ping $AR_HOSTNAME > /dev/null
if [ $? = 1 ]
   then
     echo "Unable to contact appliance. Please check hostname and network
connectivity."
```

```
exit
fi

#:<<'COMMENT'
echo "Removing old pools..."
ssh -T root@$AR_HOSTNAME <<EOF
script
   run('configuration storage');
   run('unconfig pool1');
   run('confirm done');
   run('done');
   run('configuration storage');
   run('unconfig pool2');
   run('confirm done');
   run('done');
   run('configuration storage');
   run('unconfig pool3');
   run('confirm done');
   run('done');
   run('configuration storage');
   run('unconfig pool4');
   run('confirm done');
   run('done');
EOF
#COMMENT

sleep 10


echo "Building pool 1..."
ssh -T root@$AR_HOSTNAME <<EOF
script
   run('configuration storage');
   run('config pool1');
   run('set 0-cache=1');
   run('set 1-data=3');
   run('set 1-log=1');
   run('set 2-data=3');
   run('set 3-data=3');
   run('set 4-data=2');
   run('set 5-data=3');
   run('set 6-data=3');
   run('set 7-data=3');
   run('set 8-data=3');
   run('set 9-data=3');
   run('set 10-data=3');
   run('set 11-data=3');
   run('set 12-data=3');
   run('done');
   run('set profile=mirror_nspf');
   run('set log_profile=log_stripe');
   run('set cache_profile=cache_stripe');
   run('done');
   run('done');
EOF
#:<<'COMMENT'

sleep 10

echo "Building pool 2..."
ssh -T root@$AR_HOSTNAME <<EOF
script
   run('configuration storage');
```

```
    run('config pool2');
    run('set 0-cache=1');
    run('set 1-data=2');
    run('set 1-log=1');
    run('set 2-data=3');
    run('set 3-data=3');
    run('set 4-data=3');
    run('set 5-data=3');
    run('set 6-data=3');
    run('set 7-data=3');
    run('set 8-data=3');
    run('set 9-data=3');
    run('set 10-data=3');
    run('set 11-data=3');
    run('set 12-data=3');
    run('done');
    run('set profile=mirror_nspf');
    run('set log_profile=log_stripe');
    run('set cache_profile=cache_stripe');
    run('confirm done');
    run('done');
EOF

sleep 10

echo "Building pool 3..."
ssh -T root@$AR_HOSTNAME <<EOF
script
    run('configuration storage');
    run('config pool3');
    run('set 0-cache=1');
    run('set 1-data=3');
    run('set 1-log=1');
    run('set 2-data=3');
    run('set 3-data=3');
    run('set 4-data=2');
    run('set 5-data=3');
    run('set 6-data=3');
    run('set 7-data=3');
    run('set 8-data=3');
    run('set 9-data=3');
    run('set 10-data=3');
    run('set 11-data=3');
    run('set 12-data=3');
    run('done');
    run('set profile=mirror_nspf');
    run('set log_profile=log_stripe');
    run('set cache_profile=cache_stripe');
    run('confirm done');
    run('done');
EOF

sleep 10

echo "Building pool 4..."
ssh -T root@$AR_HOSTNAME <<EOF
script
    run('configuration storage');
    run('config pool4');
    run('set 0-cache=1');
    run('set 1-data=2');
    run('set 1-log=1');
    run('set 2-data=3');
```

```
    run('set 3-data=3');
    run('set 4-data=3');
    run('set 5-data=3');
    run('set 6-data=3');
    run('set 7-data=3');
    run('set 8-data=3');
    run('set 9-data=3');
    run('set 10-data=3');
    run('set 11-data=3');
    run('set 12-data=3');
    run('done');
    run('set profile=mirror_nspf');
    run('set log_profile=log_stripe');
    run('set cache_profile=cache_stripe');
    run('confirm done');
    run('done');
EOF

#COMMENT

exit
```

NEW-35d_build_12tray_4pool_1L_b.sh

```
#!/bin/bash
#
# Build four pools on a 12-tray 7000 cluster
# Assumption is 4 pools currently present
#
# 4 Pools = ASU1/2/3 32 drives | 2 log | 1 cache
#
AR_HOSTNAME=sbm-q112-7420b

echo "Tesing connectivity to $AR_HOSTNAME..."

# Check that host is up
ping $AR_HOSTNAME > /dev/null
if [ $? = 1 ]
   then
     echo "Unable to contact appliance. Please check hostname and network
connectivity."
exit
fi

#:<<'COMMENT'
echo "Removing old pools..."
ssh -T root@$AR_HOSTNAME <<EOF
script
   run('configuration storage');
   run('unconfig pool11');
   run('confirm done');
   run('done');
   run('configuration storage');
   run('unconfig pool22');
   run('confirm done');
   run('done');
   run('configuration storage');
   run('unconfig pool33');
   run('confirm done');
   run('done');
   run('configuration storage');
   run('unconfig pool44');
```

```
    run('confirm done');
    run('done');
EOF
#COMMENT

sleep 10

#:<<'COMMENT'

echo "Building pool 11..."
ssh -T root@$AR_HOSTNAME <<EOF
script
    run('configuration storage');
    run('config pool11');
    run('set 0-cache=1');
    run('set 1-data=3');
    run('set 2-data=3');
    run('set 3-data=3');
    run('set 4-data=2');
    run('set 4-log=1');
    run('set 5-data=3');
    run('set 6-data=3');
    run('set 7-data=3');
    run('set 8-data=3');
    run('set 9-data=3');
    run('set 10-data=3');
    run('set 11-data=3');
    run('set 12-data=3');
    run('done');
    run('set profile=mirror_nspf');
    run('set log_profile=log_stripe');
    run('set cache_profile=cache_stripe');
    run('done');
    run('done');
EOF

sleep 10

echo "Building pool 22..."
ssh -T root@$AR_HOSTNAME <<EOF
script
    run('configuration storage');
    run('config pool22');
    run('set 0-cache=1');
    run('set 1-data=2');
    run('set 2-data=3');
    run('set 3-data=3');
    run('set 4-data=3');
    run('set 4-log=1');
    run('set 5-data=3');
    run('set 6-data=3');
    run('set 7-data=3');
    run('set 8-data=3');
    run('set 9-data=3');
    run('set 10-data=3');
    run('set 11-data=3');
    run('set 12-data=3');
    run('done');
    run('set profile=mirror_nspf');
    run('set log_profile=log_stripe');
    run('set cache_profile=cache_stripe');
    run('confirm done');
    run('done');
```

```
EOF

sleep 10

echo "Building pool 33..."
ssh -T root@$AR_HOSTNAME <<EOF
script
   run('configuration storage');
   run('config pool33');
   run('set 0-cache=1');
   run('set 1-data=3');
   run('set 2-data=3');
   run('set 3-data=3');
   run('set 4-data=2');
   run('set 4-log=1');
   run('set 5-data=3');
   run('set 6-data=3');
   run('set 7-data=3');
   run('set 8-data=3');
   run('set 9-data=3');
   run('set 10-data=3');
   run('set 11-data=3');
   run('set 12-data=3');
   run('done');
   run('set profile=mirror_nspf');
   run('set log_profile=log_stripe');
   run('set cache_profile=cache_stripe');
   run('confirm done');
   run('done');
EOF

sleep 10

echo "Building pool 44..."
ssh -T root@$AR_HOSTNAME <<EOF
script
   run('configuration storage');
   run('config pool44');
   run('set 0-cache=1');
   run('set 1-data=2');
   run('set 2-data=3');
   run('set 3-data=3');
   run('set 4-data=3');
   run('set 4-log=1');
   run('set 5-data=3');
   run('set 6-data=3');
   run('set 7-data=3');
   run('set 8-data=3');
   run('set 9-data=3');
   run('set 10-data=3');
   run('set 11-data=3');
   run('set 12-data=3');
   run('done');
   run('set profile=mirror_nspf');
   run('set log_profile=log_stripe');
   run('set cache_profile=cache_stripe');
   run('confirm done');
   run('done');
EOF

#COMMENT

exit
```

build-vols-12T-4P_a.sh

```
#!/bin/bash
#
#
# This script will create volumes on the 7000 appliance
# It takes one argument, which is the hostname of the Fishworks appliance.
# This is a specific config for the SPC-1 benchmark; edit varibles below...
#

ASU1_SIZE=621g
ASU2_SIZE=621g
ASU3_SIZE=138g
ASU4_SIZE=1024g

ASU1_BLOCKSIZE=16k
ASU2_BLOCKSIZE=16k
ASU3_BLOCKSIZE=128k
ASU4_BLOCKSIZE=128k

AR_HOSTNAME=sbm-q112-7420a

echo "Tesing connectivity to $AR_HOSTNAME..."

# Check that host is up
ping $AR_HOSTNAME > /dev/null
if [ $? = 1 ]
   then
     echo "Unable to contact appliance. Please check hostname and network
connectivity."
exit
fi

echo "Building LUNs..."

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool1');
   run('select default');
   run('lun asu1_1');
   run('set volsize="$ASU1_SIZE"');
   run('set volblocksize="$ASU1_BLOCKSIZE"');
   run('commit');
   run('lun asu1_2');
   run('set volsize="$ASU1_SIZE"');
   run('set volblocksize="$ASU1_BLOCKSIZE"');
   run('commit');
   run('lun asu_4');
   run('set volsize="$ASU4_SIZE"');
   run('set volblocksize="$ASU4_BLOCKSIZE"');
   run('commit');
   run('lun asu2_1');
   run('set volsize="$ASU2_SIZE"');
   run('set volblocksize="$ASU2_BLOCKSIZE"');
   run('commit');
   run('lun asu2_2');
   run('set volsize="$ASU2_SIZE"');
   run('set volblocksize="$ASU2_BLOCKSIZE"');
   run('commit');
```

```
run('lun asu3_1');
run('set volsize="$ASU3_SIZE"');
run('set volblocksize="$ASU3_BLOCKSIZE"');
run('commit');
run('lun asu3_2');
run('set volsize="$ASU3_SIZE"');
run('set volblocksize="$ASU3_BLOCKSIZE"');
run('commit');
print(run('list lun'))
run('done');
run('done');
run('shares');
run('set pool=pool2');
run('select default');
run('lun asu1_1');
run('set volsize="$ASU1_SIZE"');
run('set volblocksize="$ASU1_BLOCKSIZE"');
run('commit');
run('lun asu1_2');
run('set volsize="$ASU1_SIZE"');
run('set volblocksize="$ASU1_BLOCKSIZE"');
run('commit');
run('lun asu_4');
run('set volsize="$ASU4_SIZE"');
run('set volblocksize="$ASU4_BLOCKSIZE"');
run('commit');
run('lun asu2_1');
run('set volsize="$ASU2_SIZE"');
run('set volblocksize="$ASU2_BLOCKSIZE"');
run('commit');
run('lun asu2_2');
run('set volsize="$ASU2_SIZE"');
run('set volblocksize="$ASU2_BLOCKSIZE"');
run('commit');
run('lun asu3_1');
run('set volsize="$ASU3_SIZE"');
run('set volblocksize="$ASU3_BLOCKSIZE"');
run('commit');
run('lun asu3_2');
run('set volsize="$ASU3_SIZE"');
run('set volblocksize="$ASU3_BLOCKSIZE"');
run('commit');
print(run('list lun'))
run('done');
run('done');
run('shares');
run('set pool=pool3');
run('select default');
run('lun asu1_1');
run('set volsize="$ASU1_SIZE"');
run('set volblocksize="$ASU1_BLOCKSIZE"');
run('commit');
run('lun asu1_2');
run('set volsize="$ASU1_SIZE"');
run('set volblocksize="$ASU1_BLOCKSIZE"');
run('commit');
run('lun asu_4');
run('set volsize="$ASU4_SIZE"');
run('set volblocksize="$ASU4_BLOCKSIZE"');
run('commit');
run('lun asu2_1');
run('set volsize="$ASU2_SIZE"');
run('set volblocksize="$ASU2_BLOCKSIZE"');
```

```
    run('commit');
    run('lun asu2_2');
    run('set volsize="$ASU2_SIZE"');
    run('set volblocksize="$ASU2_BLOCKSIZE"');
    run('commit');
    run('lun asu3_1');
    run('set volsize="$ASU3_SIZE"');
    run('set volblocksize="$ASU3_BLOCKSIZE"');
    run('commit');
    run('lun asu3_2');
    run('set volsize="$ASU3_SIZE"');
    run('set volblocksize="$ASU3_BLOCKSIZE"');
    run('commit');
    print(run('list lun'))
    run('done');
    run('done');
    run('shares');
    run('set pool=pool4');
    run('select default');
    run('lun asu1_1');
    run('set volsize="$ASU1_SIZE"');
    run('set volblocksize="$ASU1_BLOCKSIZE"');
    run('commit');
    run('lun asu1_2');
    run('set volsize="$ASU1_SIZE"');
    run('set volblocksize="$ASU1_BLOCKSIZE"');
    run('commit');
    run('lun asu_4');
    run('set volsize="$ASU4_SIZE"');
    run('set volblocksize="$ASU4_BLOCKSIZE"');
    run('commit');
    run('lun asu2_1');
    run('set volsize="$ASU2_SIZE"');
    run('set volblocksize="$ASU2_BLOCKSIZE"');
    run('commit');
    run('lun asu2_2');
    run('set volsize="$ASU2_SIZE"');
    run('set volblocksize="$ASU2_BLOCKSIZE"');
    run('commit');
    run('lun asu3_1');
    run('set volsize="$ASU3_SIZE"');
    run('set volblocksize="$ASU3_BLOCKSIZE"');
    run('commit');
    run('lun asu3_2');
    run('set volsize="$ASU3_SIZE"');
    run('set volblocksize="$ASU3_BLOCKSIZE"');
    run('commit');
    print(run('list lun'))
    run('done');
EOF

exit
```

build-vols-12T-4P_b.sh

```
#!/bin/bash
#
#
# This script will create volumes on the 7000 appliance
# It takes one argument, which is the hostname of the Fishworks appliance.
# This is a specific config for the SPC-1 benchmark; edit varibles below...
#

ASU1_SIZE=621g
ASU2_SIZE=621g
ASU3_SIZE=138g
ASU4_SIZE=1024g

ASU1_BLOCKSIZE=16k
ASU2_BLOCKSIZE=16k
ASU3_BLOCKSIZE=128k
ASU4_BLOCKSIZE=128k

AR_HOSTNAME=sbm-q112-7420b

echo "Tesing connectivity to $AR_HOSTNAME..."

# Check that host is up
ping $AR_HOSTNAME > /dev/null
if [ $? = 1 ]
   then
     echo "Unable to contact appliance. Please check hostname and network
connectivity."
exit
fi

echo "Building LUNs..."

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool11');
   run('select default');
   run('lun asu1_1');
   run('set volsize="$ASU1_SIZE"');
   run('set volblocksize="$ASU1_BLOCKSIZE"');
   run('commit');
   run('lun asu1_2');
   run('set volsize="$ASU1_SIZE"');
   run('set volblocksize="$ASU1_BLOCKSIZE"');
   run('commit');
   run('lun asu_4');
   run('set volsize="$ASU4_SIZE"');
   run('set volblocksize="$ASU4_BLOCKSIZE"');
   run('commit');
   run('lun asu2_1');
   run('set volsize="$ASU2_SIZE"');
   run('set volblocksize="$ASU2_BLOCKSIZE"');
   run('commit');
   run('lun asu2_2');
   run('set volsize="$ASU2_SIZE"');
   run('set volblocksize="$ASU2_BLOCKSIZE"');
   run('commit');
```

```
run('lun asu3_1');
run('set volsize="$ASU3_SIZE"');
run('set volblocksize="$ASU3_BLOCKSIZE"');
run('commit');
run('lun asu3_2');
run('set volsize="$ASU3_SIZE"');
run('set volblocksize="$ASU3_BLOCKSIZE"');
run('commit');
print(run('list lun'))
run('done');
run('done');
run('shares');
run('set pool=pool22');
run('select default');
run('lun asu1_1');
run('set volsize="$ASU1_SIZE"');
run('set volblocksize="$ASU1_BLOCKSIZE"');
run('commit');
run('lun asu1_2');
run('set volsize="$ASU1_SIZE"');
run('set volblocksize="$ASU1_BLOCKSIZE"');
run('commit');
run('lun asu_4');
run('set volsize="$ASU4_SIZE"');
run('set volblocksize="$ASU4_BLOCKSIZE"');
run('commit');
run('lun asu2_1');
run('set volsize="$ASU2_SIZE"');
run('set volblocksize="$ASU2_BLOCKSIZE"');
run('commit');
run('lun asu2_2');
run('set volsize="$ASU2_SIZE"');
run('set volblocksize="$ASU2_BLOCKSIZE"');
run('commit');
run('lun asu3_1');
run('set volsize="$ASU3_SIZE"');
run('set volblocksize="$ASU3_BLOCKSIZE"');
run('commit');
run('lun asu3_2');
run('set volsize="$ASU3_SIZE"');
run('set volblocksize="$ASU3_BLOCKSIZE"');
run('commit');
print(run('list lun'))
run('done');
run('done');
run('shares');
run('set pool=pool33');
run('select default');
run('lun asu1_1');
run('set volsize="$ASU1_SIZE"');
run('set volblocksize="$ASU1_BLOCKSIZE"');
run('commit');
run('lun asu1_2');
run('set volsize="$ASU1_SIZE"');
run('set volblocksize="$ASU1_BLOCKSIZE"');
run('commit');
run('lun asu_4');
run('set volsize="$ASU4_SIZE"');
run('set volblocksize="$ASU4_BLOCKSIZE"');
run('commit');
run('lun asu2_1');
run('set volsize="$ASU2_SIZE"');
run('set volblocksize="$ASU2_BLOCKSIZE"');
```

```
    run('commit');
    run('lun asu2_2');
    run('set volsize="$ASU2_SIZE"');
    run('set volblocksize="$ASU2_BLOCKSIZE"');
    run('commit');
    run('lun asu3_1');
    run('set volsize="$ASU3_SIZE"');
    run('set volblocksize="$ASU3_BLOCKSIZE"');
    run('commit');
    run('lun asu3_2');
    run('set volsize="$ASU3_SIZE"');
    run('set volblocksize="$ASU3_BLOCKSIZE"');
    run('commit');
    print(run('list lun'))
    run('done');
    run('done');
    run('shares');
    run('set pool=pool44');
    run('select default');
    run('lun asu1_1');
    run('set volsize="$ASU1_SIZE"');
    run('set volblocksize="$ASU1_BLOCKSIZE"');
    run('commit');
    run('lun asu1_2');
    run('set volsize="$ASU1_SIZE"');
    run('set volblocksize="$ASU1_BLOCKSIZE"');
    run('commit');
    run('lun asu_4');
    run('set volsize="$ASU4_SIZE"');
    run('set volblocksize="$ASU4_BLOCKSIZE"');
    run('commit');
    run('lun asu2_1');
    run('set volsize="$ASU2_SIZE"');
    run('set volblocksize="$ASU2_BLOCKSIZE"');
    run('commit');
    run('lun asu2_2');
    run('set volsize="$ASU2_SIZE"');
    run('set volblocksize="$ASU2_BLOCKSIZE"');
    run('commit');
    run('lun asu3_1');
    run('set volsize="$ASU3_SIZE"');
    run('set volblocksize="$ASU3_BLOCKSIZE"');
    run('commit');
    run('lun asu3_2');
    run('set volsize="$ASU3_SIZE"');
    run('set volblocksize="$ASU3_BLOCKSIZE"');
    run('commit');
    print(run('list lun'))
    run('done');
EOF

exit
```

format_12T_4P_a.sh

```
#!/usr/bin/bash
#
# Format disks for 7000 setup
#
# To do: test and debug this rough draft
#

#set -x

AR_HOSTNAME=sbm-q112-7420a

echo "Testing connectivity to $AR_HOSTNAME..."

# Check that host is up
ping $AR_HOSTNAME > /dev/null
if [ $? = 1 ]
   then
     echo "Unable to contact appliance. Please check hostname and network
connectivity."
exit
fi

./get_luns1.sh $AR_HOSTNAME > lun_list1.txt
./get_luns2.sh $AR_HOSTNAME > lun_list2.txt
./get_luns3.sh $AR_HOSTNAME > lun_list3.txt
./get_luns4.sh $AR_HOSTNAME > lun_list4.txt

ASU1_1=/dev/rdsk/c0t`grep asu1_1 lun_list1.txt | awk '{print $3}'`d0s2
ASU1_2=/dev/rdsk/c0t`grep asu1_1 lun_list2.txt | awk '{print $3}'`d0s2
ASU1_3=/dev/rdsk/c0t`grep asu1_1 lun_list3.txt | awk '{print $3}'`d0s2
ASU1_4=/dev/rdsk/c0t`grep asu1_1 lun_list4.txt | awk '{print $3}'`d0s2
ASU1_5=/dev/rdsk/c0t`grep asu1_2 lun_list1.txt | awk '{print $3}'`d0s2
ASU1_6=/dev/rdsk/c0t`grep asu1_2 lun_list2.txt | awk '{print $3}'`d0s2
ASU1_7=/dev/rdsk/c0t`grep asu1_2 lun_list3.txt | awk '{print $3}'`d0s2
ASU1_8=/dev/rdsk/c0t`grep asu1_2 lun_list4.txt | awk '{print $3}'`d0s2

ASU2_1=/dev/rdsk/c0t`grep asu2_1 lun_list1.txt | awk '{print $3}'`d0s2
ASU2_2=/dev/rdsk/c0t`grep asu2_1 lun_list2.txt | awk '{print $3}'`d0s2
ASU2_3=/dev/rdsk/c0t`grep asu2_1 lun_list3.txt | awk '{print $3}'`d0s2
ASU2_4=/dev/rdsk/c0t`grep asu2_1 lun_list4.txt | awk '{print $3}'`d0s2
ASU2_5=/dev/rdsk/c0t`grep asu2_2 lun_list1.txt | awk '{print $3}'`d0s2
ASU2_6=/dev/rdsk/c0t`grep asu2_2 lun_list2.txt | awk '{print $3}'`d0s2
ASU2_7=/dev/rdsk/c0t`grep asu2_2 lun_list3.txt | awk '{print $3}'`d0s2
ASU2_8=/dev/rdsk/c0t`grep asu2_2 lun_list4.txt | awk '{print $3}'`d0s2

ASU3_1=/dev/rdsk/c0t`grep asu3_1 lun_list1.txt | awk '{print $3}'`d0s2
ASU3_2=/dev/rdsk/c0t`grep asu3_1 lun_list2.txt | awk '{print $3}'`d0s2
ASU3_3=/dev/rdsk/c0t`grep asu3_1 lun_list3.txt | awk '{print $3}'`d0s2
ASU3_4=/dev/rdsk/c0t`grep asu3_1 lun_list4.txt | awk '{print $3}'`d0s2
ASU3_5=/dev/rdsk/c0t`grep asu3_2 lun_list1.txt | awk '{print $3}'`d0s2
ASU3_6=/dev/rdsk/c0t`grep asu3_2 lun_list2.txt | awk '{print $3}'`d0s2
ASU3_7=/dev/rdsk/c0t`grep asu3_2 lun_list3.txt | awk '{print $3}'`d0s2
ASU3_8=/dev/rdsk/c0t`grep asu3_2 lun_list4.txt | awk '{print $3}'`d0s2

ASU4_1=/dev/rdsk/c0t`grep asu_4 lun_list1.txt | awk '{print $3}'`d0s2
ASU4_2=/dev/rdsk/c0t`grep asu_4 lun_list2.txt | awk '{print $3}'`d0s2
ASU4_3=/dev/rdsk/c0t`grep asu_4 lun_list3.txt | awk '{print $3}'`d0s2
ASU4_4=/dev/rdsk/c0t`grep asu_4 lun_list4.txt | awk '{print $3}'`d0s2
```

```
echo "Refreshing device list..."
devfsadm -C;cfgadm -al;devfsadm

echo "Formatting LUNs..."

format -e $ASU1_1<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU1_2<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU1_3<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU1_4<<EOF
n
la
```

```
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU1_5<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU1_6<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU1_7<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
```

```
q
q
EOF
format -e $ASU1_8<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU2_1<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU2_2<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU2_3<<EOF
n
la
1
y
p
```

```
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU2_4<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU2_5<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU2_6<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
```

```
format -e $ASU2_7<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU2_8<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU3_1<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF

format -e $ASU3_2<<EOF
n
la
1
y
p
0


256
```

```
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_3<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF

format -e $ASU3_4<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_5<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_6<<EOF
n
la
1
```

```
y
p
0


256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_7<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_8<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF

format -e $ASU4_1<<EOF
n
la
1
y
p
0


256
1023.9gb
la
1
y
q
```

```
q
EOF
format -e $ASU4_2<<EOF
n
la
1
y
p
0


256
1023.9gb
la
1
y
q
q
EOF


format -e $ASU4_3<<EOF
n
la
1
y
p
0


256
1023.9gb
la
1
y
q
q
EOF

format -e $ASU4_4<<EOF
n
la
1
y
p
0


256
1023.9gb
la
1
y
q
q
EOF

rm lun_list*.txt

#echo $ASU1_2
#echo $ASU_4
#echo $ASU3_1
#echo $ASU3_2
```

```
#echo $ASU2_1
#echo $ASU2_2

exit
```

format_12T_4P_b.sh

```
#!/usr/bin/bash
#
# Format disks for 7000 setup
#
# To do: test and debug this rough draft
#

#set -x

AR_HOSTNAME=sbm-q112-7420b

echo "Testing connectivity to $AR_HOSTNAME..."

# Check that host is up
ping $AR_HOSTNAME > /dev/null
if [ $? = 1 ]
   then
     echo "Unable to contact appliance. Please check hostname and network
connectivity."
exit
fi

./get_luns11.sh $AR_HOSTNAME > lun_list1.txt
./get_luns22.sh $AR_HOSTNAME > lun_list2.txt
./get_luns33.sh $AR_HOSTNAME > lun_list3.txt
./get_luns44.sh $AR_HOSTNAME > lun_list4.txt

ASU1_1=/dev/rdsk/c0t`grep asu1_1 lun_list1.txt | awk '{print $3}'`d0s2
ASU1_2=/dev/rdsk/c0t`grep asu1_1 lun_list2.txt | awk '{print $3}'`d0s2
ASU1_3=/dev/rdsk/c0t`grep asu1_1 lun_list3.txt | awk '{print $3}'`d0s2
ASU1_4=/dev/rdsk/c0t`grep asu1_1 lun_list4.txt | awk '{print $3}'`d0s2
ASU1_5=/dev/rdsk/c0t`grep asu1_2 lun_list1.txt | awk '{print $3}'`d0s2
ASU1_6=/dev/rdsk/c0t`grep asu1_2 lun_list2.txt | awk '{print $3}'`d0s2
ASU1_7=/dev/rdsk/c0t`grep asu1_2 lun_list3.txt | awk '{print $3}'`d0s2
ASU1_8=/dev/rdsk/c0t`grep asu1_2 lun_list4.txt | awk '{print $3}'`d0s2

ASU2_1=/dev/rdsk/c0t`grep asu2_1 lun_list1.txt | awk '{print $3}'`d0s2
ASU2_2=/dev/rdsk/c0t`grep asu2_1 lun_list2.txt | awk '{print $3}'`d0s2
ASU2_3=/dev/rdsk/c0t`grep asu2_1 lun_list3.txt | awk '{print $3}'`d0s2
ASU2_4=/dev/rdsk/c0t`grep asu2_1 lun_list4.txt | awk '{print $3}'`d0s2
ASU2_5=/dev/rdsk/c0t`grep asu2_2 lun_list1.txt | awk '{print $3}'`d0s2
ASU2_6=/dev/rdsk/c0t`grep asu2_2 lun_list2.txt | awk '{print $3}'`d0s2
ASU2_7=/dev/rdsk/c0t`grep asu2_2 lun_list3.txt | awk '{print $3}'`d0s2
ASU2_8=/dev/rdsk/c0t`grep asu2_2 lun_list4.txt | awk '{print $3}'`d0s2

ASU3_1=/dev/rdsk/c0t`grep asu3_1 lun_list1.txt | awk '{print $3}'`d0s2
ASU3_2=/dev/rdsk/c0t`grep asu3_1 lun_list2.txt | awk '{print $3}'`d0s2
ASU3_3=/dev/rdsk/c0t`grep asu3_1 lun_list3.txt | awk '{print $3}'`d0s2
ASU3_4=/dev/rdsk/c0t`grep asu3_1 lun_list4.txt | awk '{print $3}'`d0s2
ASU3_5=/dev/rdsk/c0t`grep asu3_2 lun_list1.txt | awk '{print $3}'`d0s2
ASU3_6=/dev/rdsk/c0t`grep asu3_2 lun_list2.txt | awk '{print $3}'`d0s2
ASU3_7=/dev/rdsk/c0t`grep asu3_2 lun_list3.txt | awk '{print $3}'`d0s2
ASU3_8=/dev/rdsk/c0t`grep asu3_2 lun_list4.txt | awk '{print $3}'`d0s2

ASU4_1=/dev/rdsk/c0t`grep asu_4 lun_list1.txt | awk '{print $3}'`d0s2
```

```
ASU4_2=/dev/rdsk/c0t`grep asu_4 lun_list2.txt | awk '{print $3}'`d0s2
ASU4_3=/dev/rdsk/c0t`grep asu_4 lun_list3.txt | awk '{print $3}'`d0s2
ASU4_4=/dev/rdsk/c0t`grep asu_4 lun_list4.txt | awk '{print $3}'`d0s2

echo "Refreshing device list..."
devfsadm -C;cfgadm -al;devfsadm

echo "Formatting LUNs..."

format -e $ASU1_1<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU1_2<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU1_3<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
```

```
format -e $ASU1_4<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU1_5<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU1_6<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU1_7<<EOF
n
la
1
y
p
0


256
620.9gb
```

```
la
1
y
q
q
EOF
format -e $ASU1_8<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU2_1<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU2_2<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU2_3<<EOF
n
la
```

```
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU2_4<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU2_5<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU2_6<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
```

```
q
q
EOF
format -e $ASU2_7<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF
format -e $ASU2_8<<EOF
n
la
1
y
p
0


256
620.9gb
la
1
y
q
q
EOF

format -e $ASU3_1<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF

format -e $ASU3_2<<EOF
n
la
1
y
p
0
```

```
256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_3<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF

format -e $ASU3_4<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_5<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_6<<EOF
```

```
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_7<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF
format -e $ASU3_8<<EOF
n
la
1
y
p
0


256
137.9gb
la
1
y
q
q
EOF

format -e $ASU4_1<<EOF
n
la
1
y
p
0


256
1023.9gb
la
```

```
1
y
q
q
EOF
format -e $ASU4_2<<EOF
n
la
1
y
p
0


256
1023.9gb
la
1
y
q
q
EOF


format -e $ASU4_3<<EOF
n
la
1
y
p
0


256
1023.9gb
la
1
y
q
q
EOF

format -e $ASU4_4<<EOF
n
la
1
y
p
0


256
1023.9gb
la
1
y
q
q
EOF

rm lun_list*.txt

#echo $ASU1_2
```

```
#echo $ASU_4
#echo $ASU3_1
#echo $ASU3_2
#echo $ASU2_1
#echo $ASU2_2

exit
```

make_meta_12T_4P_Cluster.sh

```
#!/bin/bash

# To do: add split ASU4 LUNs for more stable precondition
# To do: test and debug this rough draft

AR_HOSTNAME_A=sbm-q112-7420a
AR_HOSTNAME_B=sbm-q112-7420b

./get_luns1.sh $AR_HOSTNAME_A > lun_list1.txt
./get_luns2.sh $AR_HOSTNAME_A > lun_list2.txt
./get_luns3.sh $AR_HOSTNAME_A > lun_list3.txt
./get_luns4.sh $AR_HOSTNAME_A > lun_list4.txt
./get_luns11.sh $AR_HOSTNAME_B > lun_list11.txt
./get_luns22.sh $AR_HOSTNAME_B > lun_list22.txt
./get_luns33.sh $AR_HOSTNAME_B > lun_list33.txt
./get_luns44.sh $AR_HOSTNAME_B > lun_list44.txt
#
cat lun_list*.txt > 7420_lun_list.txt
#
set -x
# 'a' host LUNs:
ASU1_1=c0t`grep asu1_1 lun_list1.txt | awk '{print $3}'`d0s0
ASU1_2=c0t`grep asu1_1 lun_list11.txt | awk '{print $3}'`d0s0
ASU1_3=c0t`grep asu1_1 lun_list2.txt | awk '{print $3}'`d0s0
ASU1_4=c0t`grep asu1_1 lun_list22.txt | awk '{print $3}'`d0s0
ASU1_5=c0t`grep asu1_1 lun_list3.txt | awk '{print $3}'`d0s0
ASU1_6=c0t`grep asu1_1 lun_list33.txt | awk '{print $3}'`d0s0
ASU1_7=c0t`grep asu1_1 lun_list4.txt | awk '{print $3}'`d0s0
ASU1_8=c0t`grep asu1_1 lun_list44.txt | awk '{print $3}'`d0s0
ASU1_9=c0t`grep asu1_2 lun_list1.txt | awk '{print $3}'`d0s0
ASU1_10=c0t`grep asu1_2 lun_list11.txt | awk '{print $3}'`d0s0
ASU1_11=c0t`grep asu1_2 lun_list2.txt | awk '{print $3}'`d0s0
ASU1_12=c0t`grep asu1_2 lun_list22.txt | awk '{print $3}'`d0s0
ASU1_13=c0t`grep asu1_2 lun_list3.txt | awk '{print $3}'`d0s0
ASU1_14=c0t`grep asu1_2 lun_list33.txt | awk '{print $3}'`d0s0
ASU1_15=c0t`grep asu1_2 lun_list4.txt | awk '{print $3}'`d0s0
ASU1_16=c0t`grep asu1_2 lun_list44.txt | awk '{print $3}'`d0s0


ASU2_1=c0t`grep asu2_1 lun_list1.txt | awk '{print $3}'`d0s0
ASU2_2=c0t`grep asu2_1 lun_list11.txt | awk '{print $3}'`d0s0
ASU2_3=c0t`grep asu2_1 lun_list2.txt | awk '{print $3}'`d0s0
ASU2_4=c0t`grep asu2_1 lun_list22.txt | awk '{print $3}'`d0s0
ASU2_5=c0t`grep asu2_1 lun_list3.txt | awk '{print $3}'`d0s0
ASU2_6=c0t`grep asu2_1 lun_list33.txt | awk '{print $3}'`d0s0
ASU2_7=c0t`grep asu2_1 lun_list4.txt | awk '{print $3}'`d0s0
ASU2_8=c0t`grep asu2_1 lun_list44.txt | awk '{print $3}'`d0s0
ASU2_9=c0t`grep asu2_2 lun_list1.txt | awk '{print $3}'`d0s0
ASU2_10=c0t`grep asu2_2 lun_list11.txt | awk '{print $3}'`d0s0
ASU2_11=c0t`grep asu2_2 lun_list2.txt | awk '{print $3}'`d0s0
ASU2_12=c0t`grep asu2_2 lun_list22.txt | awk '{print $3}'`d0s0
ASU2_13=c0t`grep asu2_2 lun_list3.txt | awk '{print $3}'`d0s0
```

```
ASU2_14=c0t`grep asu2_2 lun_list33.txt | awk '{print $3}'`d0s0
ASU2_15=c0t`grep asu2_2 lun_list4.txt | awk '{print $3}'`d0s0
ASU2_16=c0t`grep asu2_2 lun_list44.txt | awk '{print $3}'`d0s0


ASU3_1=c0t`grep asu3_1 lun_list1.txt | awk '{print $3}'`d0s0
ASU3_2=c0t`grep asu3_1 lun_list11.txt | awk '{print $3}'`d0s0
ASU3_3=c0t`grep asu3_1 lun_list2.txt | awk '{print $3}'`d0s0
ASU3_4=c0t`grep asu3_1 lun_list22.txt | awk '{print $3}'`d0s0
ASU3_5=c0t`grep asu3_1 lun_list3.txt | awk '{print $3}'`d0s0
ASU3_6=c0t`grep asu3_1 lun_list33.txt | awk '{print $3}'`d0s0
ASU3_7=c0t`grep asu3_1 lun_list4.txt | awk '{print $3}'`d0s0
ASU3_8=c0t`grep asu3_1 lun_list44.txt | awk '{print $3}'`d0s0
ASU3_9=c0t`grep asu3_2 lun_list1.txt | awk '{print $3}'`d0s0
ASU3_10=c0t`grep asu3_2 lun_list11.txt | awk '{print $3}'`d0s0
ASU3_11=c0t`grep asu3_2 lun_list2.txt | awk '{print $3}'`d0s0
ASU3_12=c0t`grep asu3_2 lun_list22.txt | awk '{print $3}'`d0s0
ASU3_13=c0t`grep asu3_2 lun_list3.txt | awk '{print $3}'`d0s0
ASU3_14=c0t`grep asu3_2 lun_list33.txt | awk '{print $3}'`d0s0
ASU3_15=c0t`grep asu3_2 lun_list4.txt | awk '{print $3}'`d0s0
ASU3_16=c0t`grep asu3_2 lun_list44.txt | awk '{print $3}'`d0s0
# 'b' host LUNs:

#
ASU_1=c0t`grep asu_4 lun_list1.txt | awk '{print $3}'`d0s0
ASU_2=c0t`grep asu_4 lun_list11.txt | awk '{print $3}'`d0s0
ASU_3=c0t`grep asu_4 lun_list2.txt | awk '{print $3}'`d0s0
ASU_4=c0t`grep asu_4 lun_list22.txt | awk '{print $3}'`d0s0
ASU_5=c0t`grep asu_4 lun_list3.txt | awk '{print $3}'`d0s0
ASU_6=c0t`grep asu_4 lun_list33.txt | awk '{print $3}'`d0s0
ASU_7=c0t`grep asu_4 lun_list4.txt | awk '{print $3}'`d0s0
ASU_8=c0t`grep asu_4 lun_list44.txt | awk '{print $3}'`d0s0

# Clear any old metadevices
echo "Clearing any old metadevices..."
metaclear -a
metastat

mv /etc/lvm/md.tab /etc/lvm/md.old

echo "" > /etc/lvm/md.tab

echo "d1 1 16 \\" >> /etc/lvm/md.tab
echo $ASU1_2" \\" >> /etc/lvm/md.tab
echo $ASU1_1" \\" >> /etc/lvm/md.tab
echo $ASU1_4" \\" >> /etc/lvm/md.tab
echo $ASU1_3" \\" >> /etc/lvm/md.tab
echo $ASU1_6" \\" >> /etc/lvm/md.tab
echo $ASU1_5" \\" >> /etc/lvm/md.tab
echo $ASU1_8" \\" >> /etc/lvm/md.tab
echo $ASU1_7" \\" >> /etc/lvm/md.tab
echo $ASU1_10" \\" >> /etc/lvm/md.tab
echo $ASU1_9" \\" >> /etc/lvm/md.tab
echo $ASU1_12" \\" >> /etc/lvm/md.tab
echo $ASU1_11" \\" >> /etc/lvm/md.tab
echo $ASU1_14" \\" >> /etc/lvm/md.tab
echo $ASU1_13" \\" >> /etc/lvm/md.tab
echo $ASU1_16" \\" >> /etc/lvm/md.tab
echo $ASU1_15" \\" >> /etc/lvm/md.tab
echo "-i 1m" >> /etc/lvm/md.tab

echo "d2 1 16 \\" >> /etc/lvm/md.tab
echo $ASU2_2" \\" >> /etc/lvm/md.tab
```

```
echo $ASU2_1" \\" >> /etc/lvm/md.tab
echo $ASU2_4" \\" >> /etc/lvm/md.tab
echo $ASU2_3" \\" >> /etc/lvm/md.tab
echo $ASU2_6" \\" >> /etc/lvm/md.tab
echo $ASU2_5" \\" >> /etc/lvm/md.tab
echo $ASU2_8" \\" >> /etc/lvm/md.tab
echo $ASU2_7" \\" >> /etc/lvm/md.tab
echo $ASU2_10" \\" >> /etc/lvm/md.tab
echo $ASU2_9" \\" >> /etc/lvm/md.tab
echo $ASU2_12" \\" >> /etc/lvm/md.tab
echo $ASU2_11" \\" >> /etc/lvm/md.tab
echo $ASU2_14" \\" >> /etc/lvm/md.tab
echo $ASU2_13" \\" >> /etc/lvm/md.tab
echo $ASU2_16" \\" >> /etc/lvm/md.tab
echo $ASU2_15" \\" >> /etc/lvm/md.tab
echo "-i 1m" >> /etc/lvm/md.tab

echo "d3 1 16 \\" >> /etc/lvm/md.tab
echo $ASU3_2" \\" >> /etc/lvm/md.tab
echo $ASU3_1" \\" >> /etc/lvm/md.tab
echo $ASU3_4" \\" >> /etc/lvm/md.tab
echo $ASU3_3" \\" >> /etc/lvm/md.tab
echo $ASU3_6" \\" >> /etc/lvm/md.tab
echo $ASU3_5" \\" >> /etc/lvm/md.tab
echo $ASU3_8" \\" >> /etc/lvm/md.tab
echo $ASU3_7" \\" >> /etc/lvm/md.tab
echo $ASU3_10" \\" >> /etc/lvm/md.tab
echo $ASU3_9" \\" >> /etc/lvm/md.tab
echo $ASU3_12" \\" >> /etc/lvm/md.tab
echo $ASU3_11" \\" >> /etc/lvm/md.tab
echo $ASU3_14" \\" >> /etc/lvm/md.tab
echo $ASU3_13" \\" >> /etc/lvm/md.tab
echo $ASU3_16" \\" >> /etc/lvm/md.tab
echo $ASU3_15" \\" >> /etc/lvm/md.tab
echo "-i 1m" >> /etc/lvm/md.tab

metainit d4 1 8 $ASU_1 $ASU_2 $ASU_3 $ASU_4 $ASU_5 $ASU_6 $ASU_7 $ASU_8 -i 1m

metainit -a
rm lun_list*
metastat
sleep 3
echo " Creating prtvtoc.txt "
cat /etc/lvm/md.tab | grep c0 | sed 's/\\//g' > disklist.txt

prtvtoc /dev/md/rdsk/d1 > vtoc_out.txt
prtvtoc /dev/md/rdsk/d2 >> vtoc_out.txt
prtvtoc /dev/md/rdsk/d3 >> vtoc_out.txt

for x in `cat disklist.txt`
 do
    prtvtoc /dev/rdsk/$x >> vtoc_out.txt
done
mv vtoc_out.txt prtvtoc.txt

./get_hw.sh $AR_HOSTNAME_A > A-head_hw_list.txt
./get_hw.sh $AR_HOSTNAME_B > B-head_hw_list.txt
#
#
# Remove old Parameter files
mv spc1.cfg spc1.cfg-old
mv pre.txt pre.txt-old
mv p1.txt p1.txt-old
```

```
mv p2.txt p2.txt-old
#
# Build Parameter files
#
# Build Prefill parameter file
echo "compratio=1" > pre.txt
echo "sd=default,th=16 #,range=(30,100)" >> pre.txt
echo "sd=sd1,lun=/dev/md/rdsk/d4  # asu_4" >> pre.txt
echo "sd=sd2,lun=/dev/md/rdsk/d1  # asu_1" >> pre.txt
echo "sd=sd3,lun=/dev/md/rdsk/d2  # asu_2" >> pre.txt
echo "sd=sd4,lun=/dev/md/rdsk/d3  # asu_3" >> pre.txt

echo "" >> pre.txt
echo "wd=wd1,sd=sd1,seekpct=eof,rdpct=0,xfersize=1m #asu1" >> pre.txt
echo "wd=wd2,sd=sd2,seekpct=eof,rdpct=0,xfersize=1m #asu4" >> pre.txt
echo "wd=wd3,sd=sd3,seekpct=eof,rdpct=0,xfersize=1m #asu2" >> pre.txt
echo "wd=wd4,sd=sd4,seekpct=eof,rdpct=0,xfersize=1m #asu3" >> pre.txt
echo "" >> pre.txt
echo "rd=rd1,wd=wd1,elapsed=72h,interval=60,iorate=max" >> pre.txt
echo "rd=rd2,wd=wd2,elapsed=72h,interval=60,iorate=max" >> pre.txt
echo "rd=rd3,wd=wd3,elapsed=72h,interval=60,iorate=max" >> pre.txt
echo "rd=rd4,wd=wd4,elapsed=72h,interval=60,iorate=max" >> pre.txt

#
# Build SPC1.cfg parameter file
echo "# Cluster, 7420, 16 luns per ASU 300g 15K drives" > spc1.cfg
echo "sd=asu1_1,lun=/dev/md/rdsk/d1,size=10666970710016" >> spc1.cfg
echo "sd=asu2_1,lun=/dev/md/rdsk/d2,size=10666970710016" >> spc1.cfg
echo "sd=asu3_1,lun=/dev/md/rdsk/d3,size=2369093894144" >> spc1.cfg
#
# Create Persist 1 and Persist 2 parameter files
# Persist 1 is first
echo " Persist 1 file is being created"
echo " "
sleep 3
touch p1.txt
echo " " >> p1.txt
echo " # Persist 1  " >> p1.txt
echo " " >> p1.txt
echo "host=localhost,jvms=4,maxstreams=100,java=(java,-Xmx1024m) " >> p1.txt
echo "sd=default,host=localhost " >> p1.txt
echo " " >> p1.txt
echo "sd=asu1_1,lun=/dev/md/rdsk/d1,size=10666970710016" >> p1.txt
echo "sd=asu2_1,lun=/dev/md/rdsk/d2,size=10666970710016" >> p1.txt
echo "sd=asu3_1,lun=/dev/md/rdsk/d3,size=2369093894144" >> p1.txt
echo " " >>p1.txt
echo "maxlatestart=1" >>p1.txt
echo "reportinginterval=5" >>p1.txt
echo "segmentlength=512m " >>p1.txt
echo " " >>p1.txt
echo
"rd=default,rampup=180,periods=90,measurement=300,runout=0,rampdown=0,buffers=1"
>>p1.txt
echo "rd=default,rdpct=0,xfersize=1024k" >>p1.txt
echo "rd=TR1-93s_SPC-2-persist-w,streams=93" >>p1.txt
#
# Now we start building the persist 2 file
#
# Persist 2 is Next
echo " Persist 2 file is being created"
echo " "
sleep 3
touch p2.txt
```

```
echo " " >> p2.txt
echo " # Persistence Test Run 2  " >> p2.txt
echo " " >> p2.txt
echo "host=localhost,jvms=4,maxstreams=100,java=(java,-Xmx1024m) " >> p2.txt
echo "sd=default,host=localhost " >> p2.txt
echo " " >> p2.txt
echo "sd=asu1_1,lun=/dev/md/rdsk/d1,size=10666970710016" >> p2.txt
echo "sd=asu2_1,lun=/dev/md/rdsk/d2,size=10666970710016" >> p2.txt
echo "sd=asu3_1,lun=/dev/md/rdsk/d3,size=2369093894144" >> p2.txt
echo " " >>p2.txt
echo "maxlatestart=1" >>p2.txt
echo "reportinginterval=5" >>p2.txt
echo "segmentlength=512m " >>p2.txt
echo "maxpersistenceerrors=10" >>p2.txt
echo " " >>p2.txt
echo "*corruptstreams=3" >>p2.txt
echo "rd=default,buffers=1,rdpct=100,xfersize=1024k" >>p2.txt
echo "rd=TR1-93s_SPC-2-persist-r" >>p2.txt
#
# Copy Parameter files to run direstory
cp spc1.cfg ../
cp pre.txt ../
cp p1.txt ../
cp p2.txt ../
```

get_luns44.sh

```
#!/bin/bash

AR_HOSTNAME=$1

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool44');
   run('select default');
   print(run('list lun'))
   run('done');
EOF

exit
```

get_luns4.sh

```
#!/bin/bash

AR_HOSTNAME=$1

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool4');
   run('select default');
   print(run('list lun'))
   run('done');
EOF

exit
```

get_luns33.sh

```
#!/bin/bash

AR_HOSTNAME=$1

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool33');
   run('select default');
   print(run('list lun'))
   run('done');
EOF

exit
```

get_luns3.sh

```
#!/bin/bash

AR_HOSTNAME=$1

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool3');
   run('select default');
   print(run('list lun'))
   run('done');
EOF

exit
```

get_luns22.sh

```
#!/bin/bash

AR_HOSTNAME=$1

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool22');
   run('select default');
   print(run('list lun'))
   run('done');
EOF

exit
```

get_luns2.sh

```
#!/bin/bash

AR_HOSTNAME=$1

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool2');
   run('select default');
   print(run('list lun'))
   run('done');
```

```
EOF

exit
```

get_luns11.sh

```
#!/bin/bash

AR_HOSTNAME=$1

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool11');
   run('select default');
   print(run('list lun'))
   run('done');
EOF

exit
```

get_luns1.sh

```
#!/bin/bash

AR_HOSTNAME=$1

ssh -T root@$AR_HOSTNAME <<EOF
script
   run('shares');
   run('set pool=pool1');
   run('select default');
   print(run('list lun'))
   run('done');
EOF

exit
```

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

The content of SPC-1 Workload Generator command and parameter file, used in this benchmark to execute the Primary Metrics and Repeatability, is listed below.

```
# Cluster, 7420, 16 luns per ASU 300g 15K drives
sd=asu1_1,lun=/dev/md/rdsk/d1,size=10666970710016
sd=asu2_1,lun=/dev/md/rdsk/d2,size=10666970710016
sd=asu3_1,lun=/dev/md/rdsk/d3,size=2369093894144
```

Persistence Test

The content of SPC-2 Workload Generator command and parameter files, used in this benchmark to execute the Persistence Test, is listed below.

Persistence Test Run 1 *(write phase)*

```
# Persist 1

host=localhost,jvms=4,maxstreams=100,java=(java,-Xmx1024m)
sd=default,host=localhost

sd=asu1_1,lun=/dev/md/rdsk/d1,size=10666970710016
sd=asu2_1,lun=/dev/md/rdsk/d2,size=10666970710016
sd=asu3_1,lun=/dev/md/rdsk/d3,size=2369093894144

maxlatestart=1
reportinginterval=5
segmentlength=512m

rd=default,rampup=180,periods=90,measurement=300,runout=0,rampdown=0,buffers=1
rd=default,rdpct=0,xfersize=1024k
rd=TR1-93s_SPC-2-persist-w,streams=93
```

Persistence Test Run 2 *(read phase)*

```
# Persistence Test Run 2

host=localhost,jvms=4,maxstreams=100,java=(java,-Xmx1024m)
sd=default,host=localhost

sd=asu1_1,lun=/dev/md/rdsk/d1,size=10666970710016
sd=asu2_1,lun=/dev/md/rdsk/d2,size=10666970710016
sd=asu3_1,lun=/dev/md/rdsk/d3,size=2369093894144

maxlatestart=1
reportinginterval=5
segmentlength=512m
maxpersistenceerrors=10

*corruptstreams=3
rd=default,buffers=1,rdpct=100,xfersize=1024k
rd=TR1-93s_SPC-2-persist-r
```

APPENDIX E: SPC-1 WORKLOAD GENERATOR INPUT PARAMETERS

Primary Metrics Test, Repeatability Test, and Persistence Test Run 1

The following scripts was used to 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)*, and Persistence Test Run 1 in an uninterrupted sequence.

```
#!/usr/bin/ksh

#### SPC1/e Sun ZFS Storage 7420 Appliance ###
# storage=sbm-q112-7420a, sbm-q112-7420b, host=sbm-q112-t31a
#
# 9-9-2011 Run-1 At 137K SPC1e IOPS NEW KIT
#
script=run-Cluster-7420.sh
output=Full-run
basedir=/spc/output/spc1/sbm-q112-7420a/t31a/Audit-Run-4
outdir=$basedir/$output

mkdir -p $outdir

# Comment this next line if running persist 2
cp $script $outdir
# Uncomment this next line if running persist 2
#cp $script $outdir/$script-P2
#
# Get a metastat from the hosts
touch $outdir/metastat-output.txt
metastat >> $outdir/metastat-output.txt

## Copy config scripts and all config files to the output directory
cp -r Config-7420-Cluster $outdir/
##
#
# Preconditioning
########
#
#/vdbench/vdbench503rc11/vdbench -f pre.txt -o $outdir/pre-SH-1a/
#sleep 5
#cp pre.txt $outdir
#echo " reboot 7420a/b cluster, then kill the sleep cmd "
#exit
#sleep 600000000000000000
##
bsu=2741
STEP=200
startup=180
while [[ $bsu -le 2741 ]]
do
Outdir=$outdir/${bsu}
mkdir -p $Outdir
cp spc1.cfg $Outdir
cp /etc/system $Outdir
#
####
echo "Power run will start Now"
####
### Start the power meter
echo " Start Power Meter"
```

```
echo "Current output directory is " $outdir
sleep 180
#### Starting Pre-Range Phase for ten Minutes
java -Xmx1024m -Xss256k range -b $bsu -s $startup -t 600
mv rangetest $Outdir/pre-rangetest
####
#### Starting Idle Phase - Sleeping for 35 Minutes
echo " Starting Idle Phase - Sleeping for 35 Minutes "
sleep 2100
####
#### Starting Post-Range Phase 10% of workload
echo " Starting Post-Range Phase 10% of workload "
java -Xmx1024m -Xss256k range -b 274 -s $startup -t 600
mv rangetest $Outdir/post-rangetest
####
#### Sleep 10 seconds then start metrics phases
#
#
sleep 10
java -Xmx1024m -Xss256k spc1 -f spc1.cfg sustain.txt -o sustain SPCOut
sleep 10
java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp100.txt -o ramp100 SPCOut
sleep 10
java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp095.txt -o ramp095 SPCOut
sleep 10
java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp090.txt -o ramp090 SPCOut
sleep 10
java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp080.txt -o ramp080 SPCOut
sleep 10
java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp050.txt -o ramp050 SPCOut
sleep 10
java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp010.txt -o ramp010 SPCOut
sleep 10
#
#
#
java -Xmx1024m -Xss256k repeat1 -b $bsu -s $startup
sleep 3
java -Xmx1024m -Xss256k repeat2 -b $bsu -s $startup
sleep 3
echo " Turn off power meter"
sleep 180
mv sustain $Outdir
cp sustain.txt $Outdir
sleep 3
mv ramp100 $Outdir
cp ramp100.txt $Outdir
sleep 3
mv ramp095 $Outdir
cp ramp095.txt $Outdir
sleep 3
mv ramp090 $Outdir
cp ramp090.txt $Outdir
sleep 3
mv ramp080 $Outdir
cp ramp080.txt $Outdir
sleep 3
mv ramp050 $Outdir
cp ramp050.txt $Outdir
sleep 3
mv ramp010 $Outdir
cp ramp010.txt $Outdir
sleep 3
```

```
mv repeatability1 $Outdir
sleep 3
mv repeatability2 $Outdir
sleep 3
mv SPCOut $Outdir
cp SPC1.cfg $Outdir
sleep 3
##
## SPC2 Persist Runs
#
/spc/spc2/spc2 -f p1.txt -o $Outdir/init -init
##
sleep 10
/spc/spc2/spc2 -f p1.txt -o $Outdir/persist1/
sleep 5
cp p1.txt $Outdir
sleep 10
####
####
####
#/spc/spc2/spc2 -f p2.txt -o $Outdir/persist2/
#sleep 5
#cp p2.txt $Outdir
#sleep 5
#
#
bsu=`expr $bsu + $STEP`
echo "All done .... sleeping for 5 minutes"
sleep 3
done
#
cd $basedir ;/bin/chmod -R 777 $output ;/usr/bin/zip -r $output.zip $output
#cd $basedir ;/bin/chmod -R 777 $output ;mv $output.zip $output.zip-P1 ;/usr/bin/zip
-r $output.zip $output
```

sustain.txt

```
rd=sustain,bsus=2741,startup=5400,elapsed=28800,interval=60
```

ramp100.txt

```
rd=ramp_100,bsus=2741,startup=1800,elapsed=600,interval=60
```

ramp095.txt

```
rd=ramp_95,bsus=2604,startup=1800,elapsed=600,interval=60
```

ramp090.txt

```
rd=ramp_90,bsus=2467,startup=1800,elapsed=600,interval=60
```

ramp080.txt

```
rd=ramp_80,bsus=2193,startup=1800,elapsed=600,interval=60
```

ramp050.txt

```
rd=ramp_50,bsus=1371,startup=1800,elapsed=600,interval=60
```

ramp010.txt

```
rd=ramp_10,bsus=274,startup=1800,elapsed=600,interval=60
```

Persistence Test Run 2

The following script was used to execute Persistence Test Run 2.

```
#!/usr/bin/ksh

#### SPC1/e Sun ZFS Storage 7420 Appliance ###
# storage=sbm-q112-7420a, sbm-q112-7420b, host=sbm-q112-t31a
#
# 9-10-2011 Run-1 At 137K SPC1e IOPS NEW KIT
#
script=run-Cluster-7420.sh
output=Full-run
basedir=/spc/output/spc1/sbm-q112-7420a/t31a/Audit-Run-4
outdir=$basedir/$output

#mkdir -p $outdir

# Comment this next line if running persist 2
#cp $script $outdir
# Uncomment this next line if running persist 2
cp $script $outdir/$script-P2
#
# Get a metastat from the hosts
touch $outdir/metastat-output.txt
metastat >> $outdir/metastat-output.txt

## Copy config scripts and all config files to the output directory
#cp -r Config-7420-Cluster $outdir/
##
#
# Preconditioning
########
#
#/vdbench/vdbench503rc11/vdbench -f pre.txt -o $outdir/pre-SH-1a/
#sleep 5
#cp pre.txt $outdir
#echo " reboot 7420a/b cluster, then kill the sleep cmd "
#exit
#sleep 600000000000000000
##
bsu=2741
STEP=200
startup=180
while [[ $bsu -le 2741 ]]
do
Outdir=$outdir/${bsu}
#mkdir -p $Outdir
#cp spc1.cfg $Outdir
#cp /etc/system $Outdir
#
####
#echo "Power run will start Now"
#####
#### Start the power meter
#echo " Start Power Meter"
#echo "Current output directory is " $outdir
#sleep 180
##### Starting Pre-Range Phase for ten Minutes
#java -Xmx1024m -Xss256k range -b $bsu -s $startup -t 600
#mv rangetest $Outdir/pre-rangetest
#####
```

```
##### Starting Idle Phase - Sleeping for 35 Minutes
#echo " Starting Idle Phase - Sleeping for 35 Minutes "
#sleep 2100
#####
##### Starting Post-Range Phase 10% of workload
#echo " Starting Post-Range Phase 10% of workload "
#java -Xmx1024m -Xss256k range -b 274 -s $startup -t 600
#mv rangetest $Outdir/post-rangetest
#####
##### Sleep 10 seconds then start metrics phases
##
##
#sleep 10
#java -Xmx1024m -Xss256k spc1 -f spc1.cfg sustain.txt -o sustain SPCOut
#sleep 10
#java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp100.txt -o ramp100 SPCOut
#sleep 10
#java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp095.txt -o ramp095 SPCOut
#sleep 10
#java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp090.txt -o ramp090 SPCOut
#sleep 10
#java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp080.txt -o ramp080 SPCOut
#sleep 10
#java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp050.txt -o ramp050 SPCOut
#sleep 10
#java -Xmx1024m -Xss256k spc1 -f spc1.cfg ramp010.txt -o ramp010 SPCOut
#sleep 10
##
##
##
#java -Xmx1024m -Xss256k repeat1 -b $bsu -s $startup
#sleep 3
#java -Xmx1024m -Xss256k repeat2 -b $bsu -s $startup
#sleep 3
#echo " Turn off power meter"
#sleep 180
#mv sustain $Outdir
#cp sustain.txt $Outdir
#sleep 3
#mv ramp100 $Outdir
#cp ramp100.txt $Outdir
#sleep 3
#mv ramp095 $Outdir
#cp ramp095.txt $Outdir
#sleep 3
#mv ramp090 $Outdir
#cp ramp090.txt $Outdir
#sleep 3
#mv ramp080 $Outdir
#cp ramp080.txt $Outdir
#sleep 3
#mv ramp050 $Outdir
#cp ramp050.txt $Outdir
#sleep 3
#mv ramp010 $Outdir
#cp ramp010.txt $Outdir
#sleep 3
#mv repeatability1 $Outdir
#sleep 3
#mv repeatability2 $Outdir
#sleep 3
#mv SPCOut $Outdir
#cp SPC1.cfg $Outdir
```

```
#sleep 3
###
### SPC2 Persist Runs
##
#/spc/spc2/spc2 -f p1.txt -o $Outdir/init -init
###
#sleep 10
#/spc/spc2/spc2 -f p1.txt -o $Outdir/persist1/
#sleep 5
#cp p1.txt $Outdir
#sleep 10
#####
#####
#####
/spc/spc2/spc2 -f p2.txt -o $Outdir/persist2/
sleep 5
cp p2.txt $Outdir
sleep 5
#
#
bsu=`expr $bsu + $STEP`
echo "All done .... sleeping for 5 minutes"
sleep 3
done
#
#cd $basedir ;/bin/chmod -R 777 $output ;/usr/bin/zip -r $output.zip $output
cd $basedir ;/bin/chmod -R 777 $output ;mv $output.zip $output.zip-P1 ;/usr/bin/zip
-r $output.zip $output
```