FUJITSU

THE POSSIBILITIES ARE INFINITE

SPC BENCHMARK 1™
FULL DISCLOSURE REPORT

FUJITSU LIMITED
FUJITSU STORAGE SYSTEMS ETERNUS DX440 S2

SPC-1 V1.12

Submitted for Review: June 26, 2012
Submission Identifier: A00117

First Edition – June 2012

Trademarks

Table of Contents

AUDIT CERTIFICATION

C.A. (Sandy) Wilson
Fujitsu Limited
1250 East Arques Ave.
P.O. Box 3470
Sunnyvale, CA 94088 3470

June 26, 2012

The SPC Benchmark 1™ Reported Data listed below for the Fujitsu Storage Systems ETERNUS DX440 S2 was produced in compliance with the SPC Benchmark 1™ v1.12 Remote Audit requirements.

SPC Benchmark 1™ v1.12 Reported Data	
Tested Storage Product (TSP) Name: Fujitsu Storage Systems ETERNUS DX440 S2	
Metric	**Reported Result**
SPC-1 IOPS™	102,989.38
SPC-1 Price-Performance	$8.62/SPC-1 IOPS™
Total ASU Capacity	70,224.000 GB
Data Protection Level	Protected *(Mirroring)*
Total TSC Price (including three-year maintenance)	$888,235.30

The following SPC Benchmark 1™ Remote 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 information supplied by Fujitsu Limited:
  - ✓ 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.
- The total Application Storage Unit (ASU) Capacity was filled with random data, using an auditor approved tool, prior to execution of the SPC-1 Tests.
- An appropriate diagram of the Benchmark Configuration (BC)/Tested Storage Configuration (TSC).

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

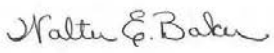
AUDIT CERTIFICATION *(CONT.)*

- Listings and commands to configure the Benchmark Configuration/Tested Storage Configuration, including customer tunable parameters that were changed from default values.
- SPC-1 Workload Generator commands and parameters used for the audited SPC Test Runs.
- The following Host System requirements were verified by information supplied by Fujitsu Limited:
  - ✓ The type of each Host System including the number of processors and main memory.
  - ✓ The presence and version number of the SPC-1 Workload Generator on each Host System.
  - ✓ The TSC boundary within each Host System.
- The Test Results Files and resultant Summary Results Files received from Fujitsu Limited for each of following were authentic, accurate, and compliant with all of the requirements and constraints of Clauses 4 and 5 of the SPC-1 Benchmark Specification:
  - ✓ Data Persistence Test
  - ✓ Sustainability Test Phase
  - ✓ IOPS Test Phase
  - ✓ Response Time Ramp Test Phase
  - ✓ Repeatability Test
- There were no differences between the Tested Storage Configuration and Priced Storage Configuration.
- The submitted pricing information met all of the requirements and constraints of Clause 8 of the SPC-1 Benchmark Specification.
- The Full Disclosure Report (FDR) met all of the requirements in Clause 9 of the SPC-1 Benchmark Specification.
- This successfully audited SPC measurement is not subject to an SPC Confidential Review.

Audit Notes:

There were no audit notes or exceptions.

Respectfully,



Walter E. Baker
SPC Auditor

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

LETTER OF GOOD FAITH

FUJITSU

Kanagawa-ken,Kawasaki-shi,Nakahara-ku,Kamikodanaka,4-1-1,JAPAN211-8588
Phone: 044-754-3640

June 11, 2012
From: Shigeo Konno, Fujitsu Limited

To: Walter E. Baker, SPC Auditor
Gradient Systems, Inc.
643 Bair Island Road, Suite 103
Redwood City, CA 94063-2755. U.S.A.

Contact Information: Carrel A. (Sandy)Wilson
Fujitsu America, Inc.
1250 East Arques Ave. PO Box 3470
Sunnyvale, CA 94088, U.S.A.

Subject: SPC-1 Letter of Good Faith for the ETERNUS DX440 S2

Fujitsu Limited 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 V1.12 of the SPC-1 benchmark specification.

In addition, we have reported any items in the Benchmark Configuration and execution of the benchmark necessary to reproduce the reported results even if the items are not explicitly required to be disclosed by the SPC-1 benchmark specification.

Signed: Date:

Shigeo Konno
General Manager, Storage System Division

EXECUTIVE SUMMARY

Test Sponsor and Contact Information

Test Sponsor and Contact Information	
Test Sponsor Primary Contact	Fujitsu Limited – http://www.fujitsu.com/services/computing/storage/ Fujitsu America, Inc. C.A. (Sandy) Wilson Sandy_Wilson@us.fujitsu.com 1250 East Arques Ave PO Box 3470 Sunnyvale, CA 94088-3470 Phone: (916) 434-8593
Test Sponsor Alternate Contact	Fujitsu Limited – http://www.fujitsu.com/services/computing/storage/ Fujitsu America, Inc. Kun Katsumata Kun_Katsumata@us.fujitsu.com 1250 East Arques Ave PO Box 3470 Sunnyvale, CA 94088-3470 Phone: (408) 746-6415
Test Sponsor Alternate Contact	Fujitsu Limited http://www.fujitsu.com/services/computing/storage/ Shigeo Konno konno.shigeo@jp.fujitsu.com 1-1 Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa-ken 211-8588, Japan Phone: (044) 754-3632 FAX: (044) 754-3719
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	V2.2.0
Date Results were first used publicly	June 26, 2012
Date the FDR was submitted to the SPC	June 26, 2012
Date the Priced Storage Configuration is available for shipment to customers	currently available
Date the TSC completed audit certification	June 26, 2012

Tested Storage Product (TSP) Description

The Fujitsu ETERNUS DX440 S2 is a flexible, highly reliable storage array, equipped with redundant components to provide uncompromised availability to mid-market requirements. A mixture of 300GB, 450GB, 600GB, and 900GB 10krpm plus 1TB Nearline 2.5" SAS drives are offered. In addition, 300GB, 450GB, and 600GB 15krpm SAS drives, as well as 1TB, 2TB, and 3TB Nearline SAS drives may be used, up to a maximum of 960 drives. SSD drives are available in 100GB, 200GB, and 400GB sizes. The drives may be arranged in a variety of RAID groups, including RAID1, RAID1+0(10), RAID5, RAID6, and RAID5+0(50).

The product is offered with Fibre Channel *(as tested)*, iSCSI *(both 1Gbps and 10Gbps)*, and FCoE host connections. Up to 8 Channel Adapters (CAs) can be installed, 4 on each the two Control Modules (CMs), with mixed types permitted. Most types have two ports on each CA, however a FC CA is offered with 4 ports. In addition, a number of different snapshot and replication facilities, native disk data encryption, and MAID capabilities are available.

Summary of Results

SPC-1 Reported Data	
Tested Storage Product (TSP) Name: Fujitsu Storage Systems ETERNUS DX440 S2	
Metric	**Reported Result**
SPC-1 IOPS™	102,989.38
SPC-1 Price-Performance™	$8.62/SPC-1 IOPS™
Total ASU Capacity	70,224.000 GB
Data Protection Level	Protected *(Mirroring)*
Total TSC Price (including three-year maintenance)	$888,235.30

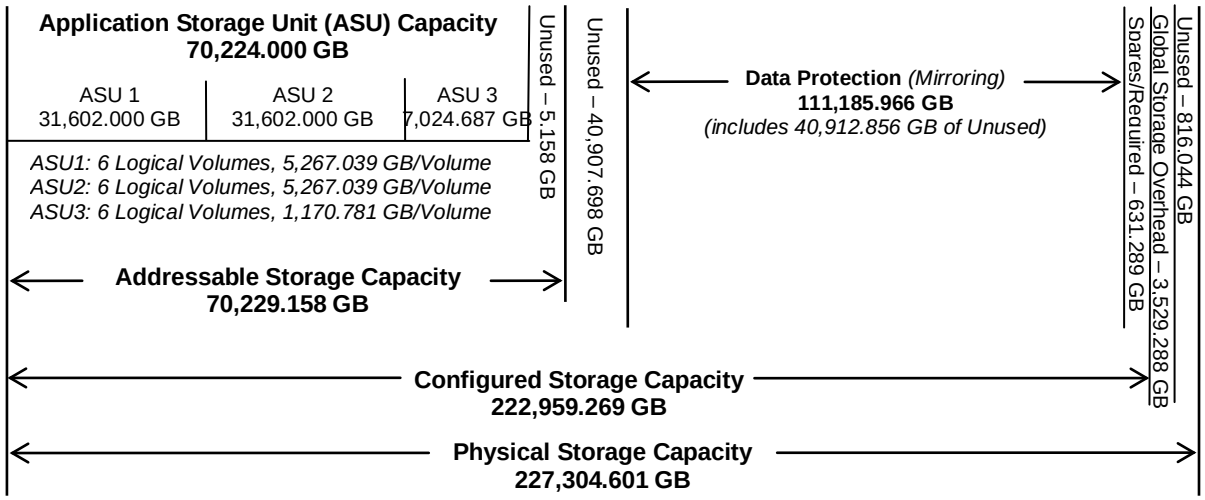
SPC-1 IOPS™ represents the maximum I/O Request Throughput at the 100% load point.

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 **Protected** using ***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	30.89%
Protected Application Utilization	61.81%
Unused Storage Ratio	36.36%

Application Utilization: Total ASU Capacity *(70,224.000 GB)* divided by Physical Storage Capacity *(227,304.601 GB)*

Protected Application Utilization: Total ASU Capacity *(70,224.000 GB)* plus total Data Protection Capacity *(111,185.966 GB)* minus unused Data Protection Capacity *(40,912.856 GB)* divided by Physical Storage Capacity *(227,304.601 GB)*

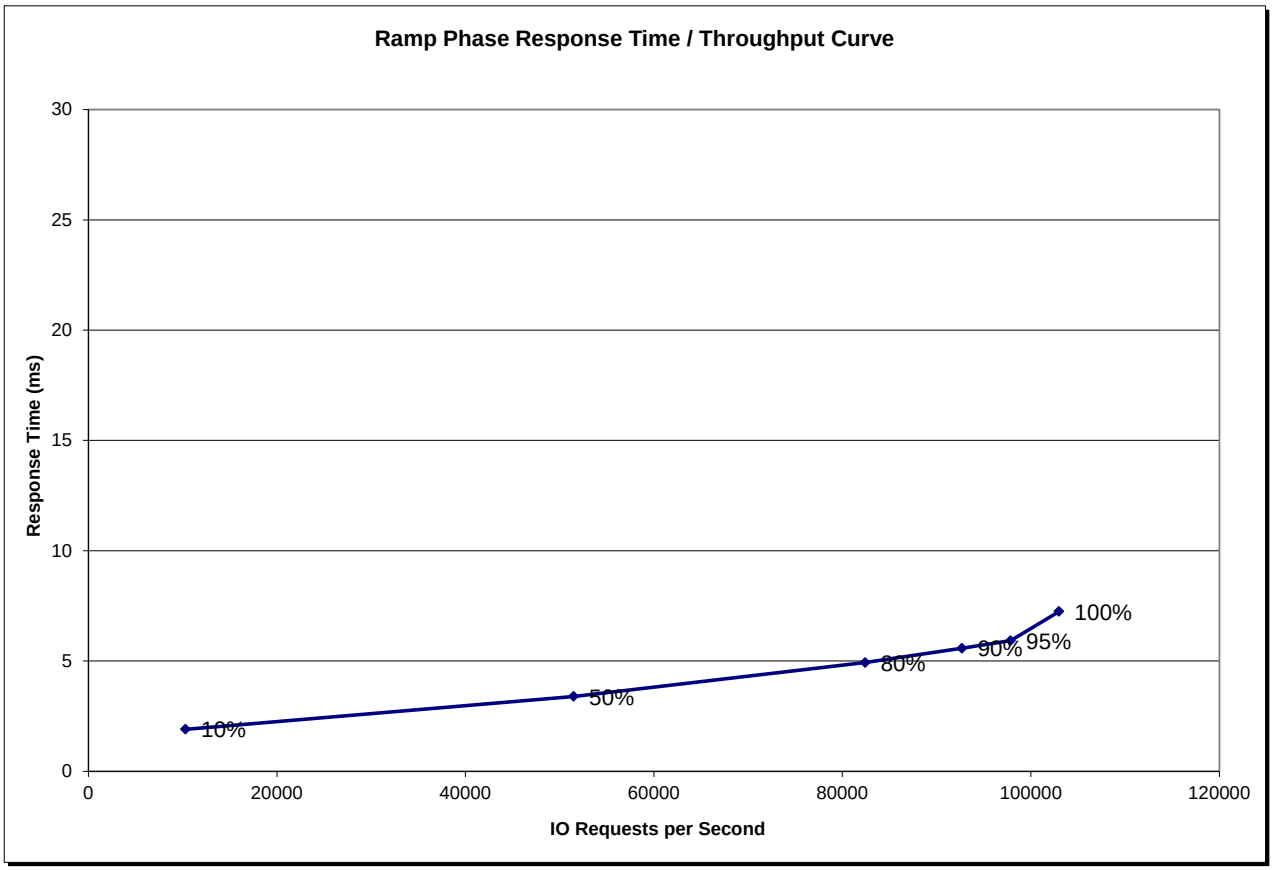
Unused Storage Ratio: Total Unused Capacity *(82,646.914 GB)* divided by Physical Storage Capacity *(227,304.601 GB)* and may not exceed 45%.

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	10,307.53	51,487.03	82,413.86	92,697.84	97,855.65	102,989.38
Average Response Time (ms):						
All ASUs	1.90	3.39	4.93	5.58	5.92	7.25
ASU-1	2.66	4.33	6.10	6.84	7.25	7.87
ASU-2	1.55	2.72	4.03	4.61	4.91	5.39
ASU-3	0.45	1.71	2.84	3.32	3.56	6.73
Reads	4.15	6.01	8.17	9.09	9.60	10.27
Writes	0.44	1.69	2.82	3.30	3.53	5.28

Priced Storage Configuration Pricing

Product ID	Product Name	Qty	Unit List Price	Extended LP	Discount %	Discounted Price
ET442SAU	DX440 S2 Base System Rackmount (AC200V, 3RU)	1	$39,515.00	$39,515.00	30%	$27,660.50
ETNHF24	FC Host Interface, pair - 8 ports (2/4/8 Gbps, Host/Remote Connect)	2	$10,300.00	$20,600.00	30%	$14,420.00
ETNM86	48GB Cache Memory for DX440 S2 (8GB two sets of 3)	2	$36,000.00	$72,000.00	30%	$50,400.00
ETNAD2CU	Drive Enclosure (2.5" HDD) Rackmount (AC200V, 2RU)	32	$6,000.00	$192,000.00	30%	$134,400.00
ETND3HC	300GB/10krpm 2.5" Disk Drives	762	$865.00	$659,130.00	30%	$461,391.00
ETNRKC2U	Base Rack - Standard (42RU) with Front & Rear doors, side panels without Earthquake stabilizer	1	$3,150.00	$3,150.00	30%	$2,205.00
ETNRKD2U	Expansion Rack - Standard (42RU) with Front & Rear doors, no side panels without Earthquake stabilizer	1	$3,150.00	$3,150.00	30%	$2,205.00
ETNC2Q6-L	Rack-to-rack Extension Cables (6 meters)	2	$715.00	$1,430.00	30%	$1,001.00
ETNP16U-L	Power Distribution Unit for DX (AC240V, 30A - 8 enclosures, 2RU)	5	$1,520.00	$7,600.00	30%	$5,320.00
S6361-F3631-L2	QLogic 8Gbps Dual Port Fibre Channel Host Bus Adapter	8	$1,920.00	$15,360.00	10%	$13,824.00
61-343827-003	Fibre Channel Cable LC-LC 3 m	16	$132.00	$2,112.00	30%	$1,478.40
	(Provide 24 hour per day / 7days per week 4 hour response maintenance for 36 months)					
	36 months, Enhanced Plus	1	$248,472.00	$248,472.00	30%	$173,930.40
	SFPs are included.			**Total:**		**$888,235.30**

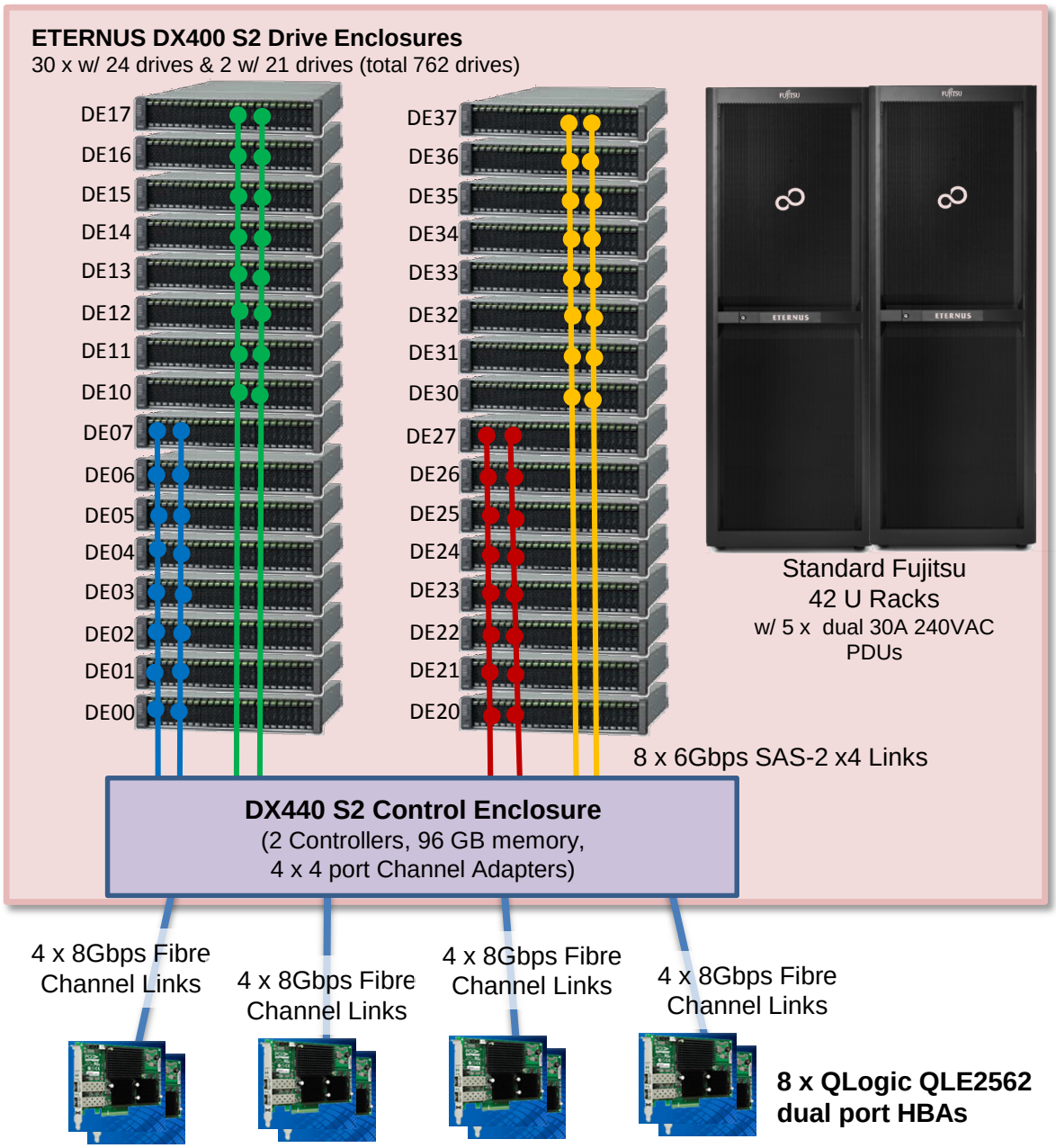
The above pricing includes hardware maintenance and software support for three years, 7 days per week, 24 hours per day. The hardware maintenance and software support provides the following:

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

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

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

Priced Storage Configuration Diagram



Priced Storage Configuration Components

Priced Storage Configuration
8 – QLogic QLE2562 dual port FC HBAs *(8 Gbps)*
Fujitsu Storage Systems ETERNUS DX440 S2 2 – Controller Modules, each with: 48 GB cache *(96 GB total)* Flash Memory power fail protection 2 – Channel Adapter modules, each with 4 –8 Gbps Fibre Channel ports *(front-end Host connections, 8 total and 8 used)* *(16 connections available and 16 used with both controllers)* 4 – SAS Expander Drive interfaces, QSFP SAS-2 *(backend connections to first drive enclosure)* *(8 backend connections total, 8 used)*
32 –ETERNUS DX400 S2 Drive Enclosures, each with 2 – I/O Modules, each with SAS Expander Drive interface, QSFP SAS-2 *(2 total, 2 used)*
762 – 300 GB 10K RPM 2.5” SAS Disk Drives: *24 disk drives in each of 30 ETERNUS DX400 S2 Drive Enclosures* *21 disk drives in each of 2 ETERNUS DX80 S2 Drive Enclosures*
2 – standard Fujitsu 42U racks with a total of 5 dual 30A 240VAC PDUs

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 Benchmark Configuration (BC)/Tested Storage Configuration (TSC) utilized direct attached storage.

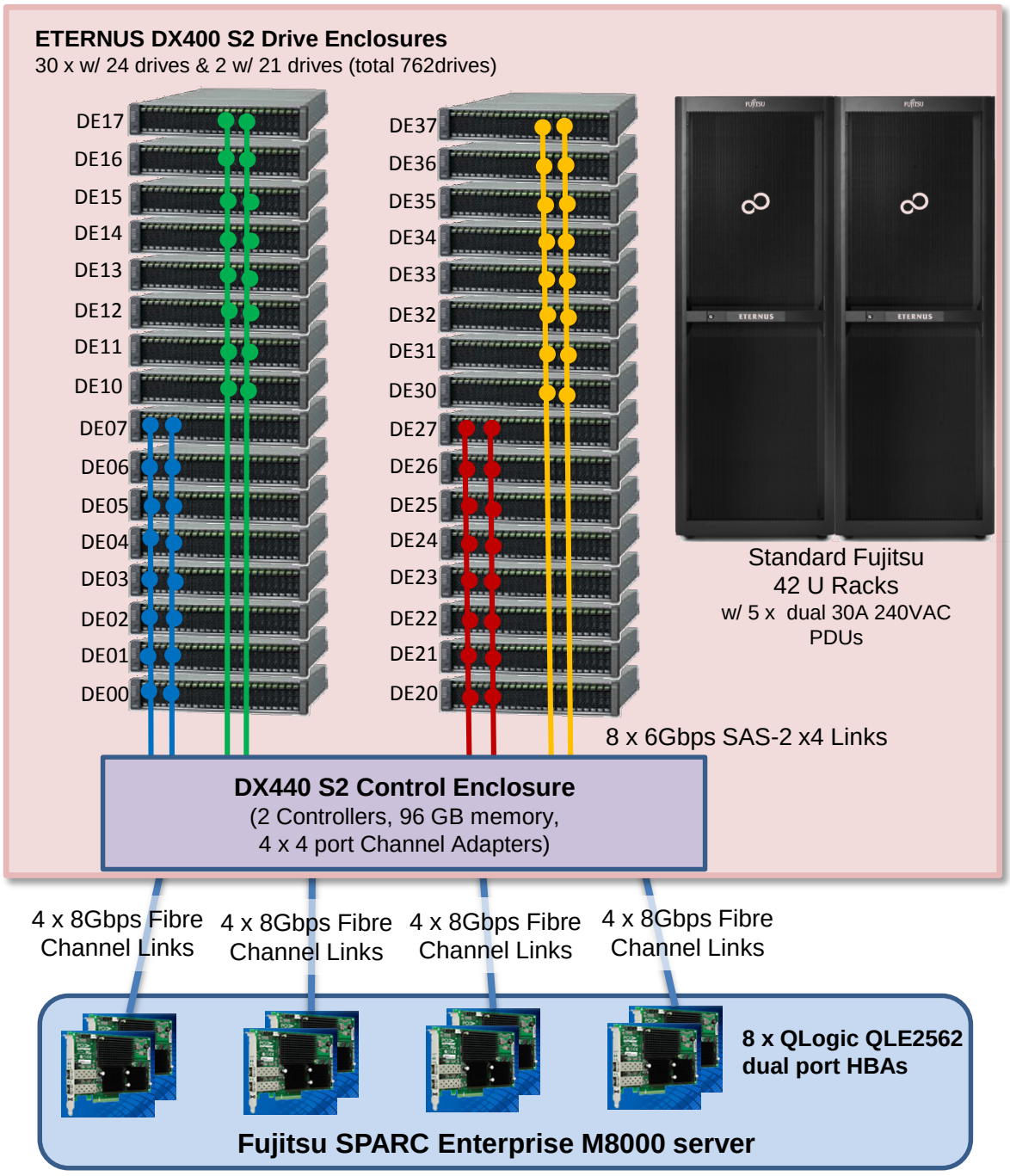
Host System and Tested Storage Configuration (TSC) Table of Components

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 *(Host Systems and Tested Storage Configuration Components).*

Benchmark Configuration/Tested Storage Configuration Diagram



Host Systems and Tested Storage Configuration Components

Host System:	Tested Storage Configuration (TSC):
Fujitsu SPARC Enterprise M8000 server 16 – SPARC64 VI CPU chips, each with 128 KB L1 instruction cache, 128 KB L1 data cache, 6 MB L2 cache 512 GB main memory Solaris 10 PCI-X	8 – !Logic QLE2562 dual port FC HBAs *(8 Gbps)*
	Fujitsu Storage Systems ETERNUS DX440 S2 2 – Controller Modules, each with: 48 GB cache *(96 GB total)* Flash Memory power fail protection 4 – Channel Adapter modules, each with 4 –8 Gbps Fibre Channel ports *(front-end Host connections, 8 total, 8 used)* *(16 connections available and 16 used with both controllers)* 4 – SAS 2 x4 Expander Drive interfaces, QSFP SAS-2 *(backend connections to first drive enclosure)* *(8 backend connections total, 8 used)*
	32 –ETERNUS DX400 S2 Drive Enclosures, each with 2 – I/O Modules, each with SAS 2 x4 Expander Drive interfaces, QSFP SAS-2 *(2 total, 2 used)*
	762 – 300 GB 10K RPM 2.5” SAS Disk Drives: *24 disk drives in each of 30 drive enclosures* *21 disk drives in each of 2 drive enclosures*
	2 – standard Fujitsu 40U racks with a total of 5 dual 30A 240VAC PDUs

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 60 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 61 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 90.

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 56 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)	70,224.000
Addressable Storage Capacity	Gigabytes (GB)	70,229.158
Configured Storage Capacity	Gigabytes (GB)	222,959.269
Physical Storage Capacity	Gigabytes (GB)	227,304.601
Data Protection *(Mirroring)*	Gigabytes (GB)	111,185.966
Required Storage *(spares, overhead/metadata)*	Gigabytes (GB)	631.289
Global Storage Overhead	Gigabytes (GB)	3,529.288
Total Unused Storage	Gigabytes (GB)	82,646.914

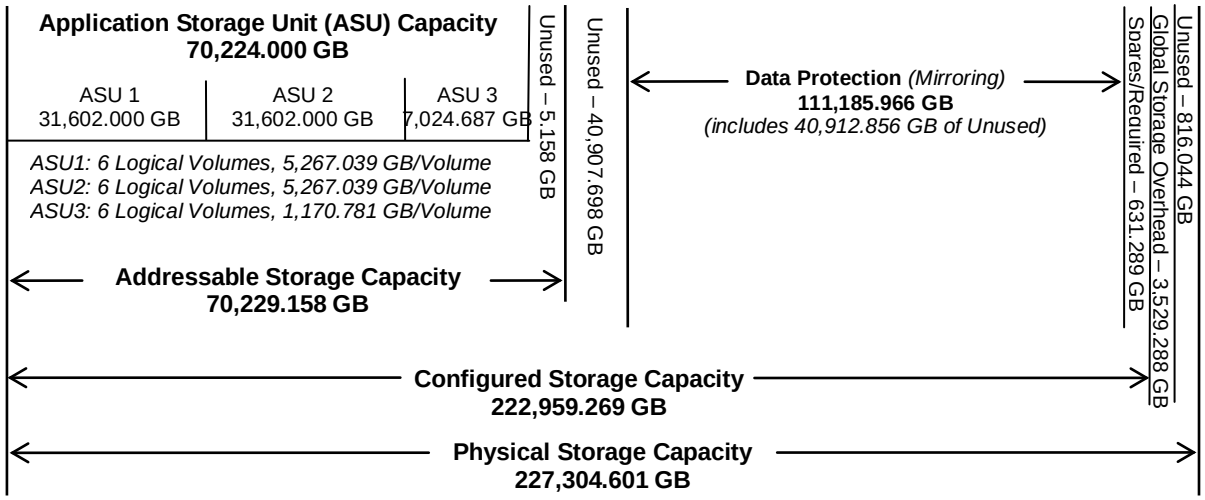
SPC-1 Storage Hierarchy Ratios

	Addressable Storage Capacity	**Configured Storage Capacity**	**Physical Storage Capacity**
Total ASU Capacity	99.99%	31.50%	30.89%
Required for Data Protection *(Mirroring)*		49.87%	48.91%
Addressable Storage Capacity		31.50%	30.90%
Required Storage *(overhead/metadata)*		0.28%	0.28%
Configured Storage Capacity			98.09%
Global Storage Overhead			1.55%
Unused Storage:			
Addressable	0.01%		
Configured		36.70%	
Physical			0.36%

The Physical Storage Capacity consisted of 227,304.601 GB distributed over 762 disk drives, each with a formatted capacity of 298.300 GB. There was 816.044 GB (0.36%) of Unused Storage within the Physical Storage Capacity. Global Storage Overhead consisted of 3,529.288 GB (1.55%) of the Physical Storage Capacity. There was 81,825.712 GB (36.70%) of Unused Storage within the Configured Storage Capacity. The Total ASU Capacity utilized 99.99% of the Addressable Storage Capacity resulting in 5.158 GB (0.01%) of Unused Storage within the Addressable Storage Capacity. The Data Protection *(Mirroring)* capacity was 111,185.966 GB of which 70,273.110 GB was utilized. The total Unused Storage capacity was 82,646.914 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 (31,602.000 GB)	**ASU-2 (31,602.000 GB)**	**ASU-3 (7,020.000 GB)**
6 Logical Volumes 5,267.039 GB per Logical Volume (5,267.000 GB used per Logical Volume)	6 Logical Volumes 5,267.039 GB per Logical Volume (5,267.000 GB used per Logical Volume)	6 Logical Volumes 1,170.781 GB per Logical Volume (1,170.000 GB 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	30.89%
Protected Application Utilization	61.81%
Unused Storage Ratio	36.36%

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 57 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 92.

Sustainability Test Results File

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

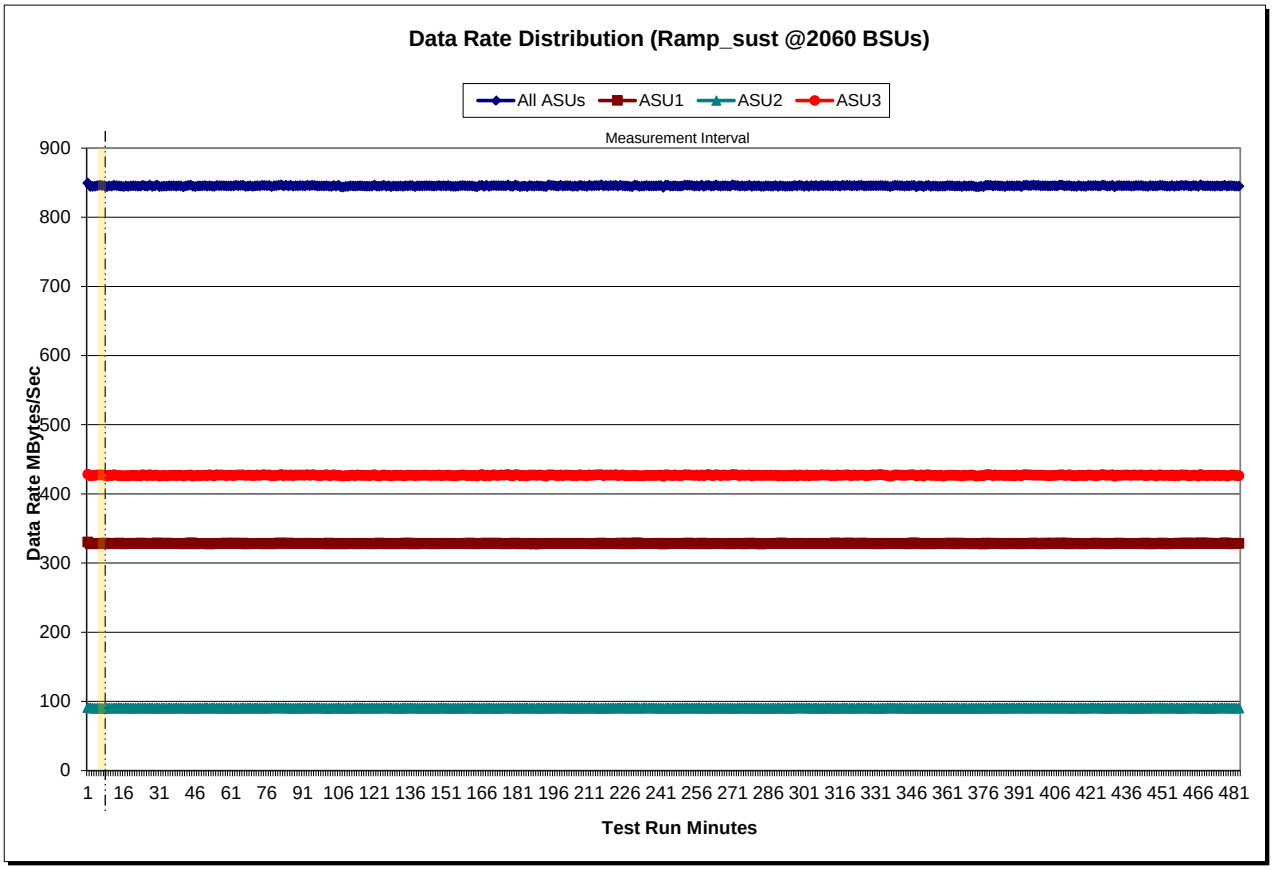
Sustainability Test Results File

Sustainability – Data Rate Distribution Data *(MB/second)*

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

Sustainability Data Rate Table

Sustainability – Data Rate Distribution Graph

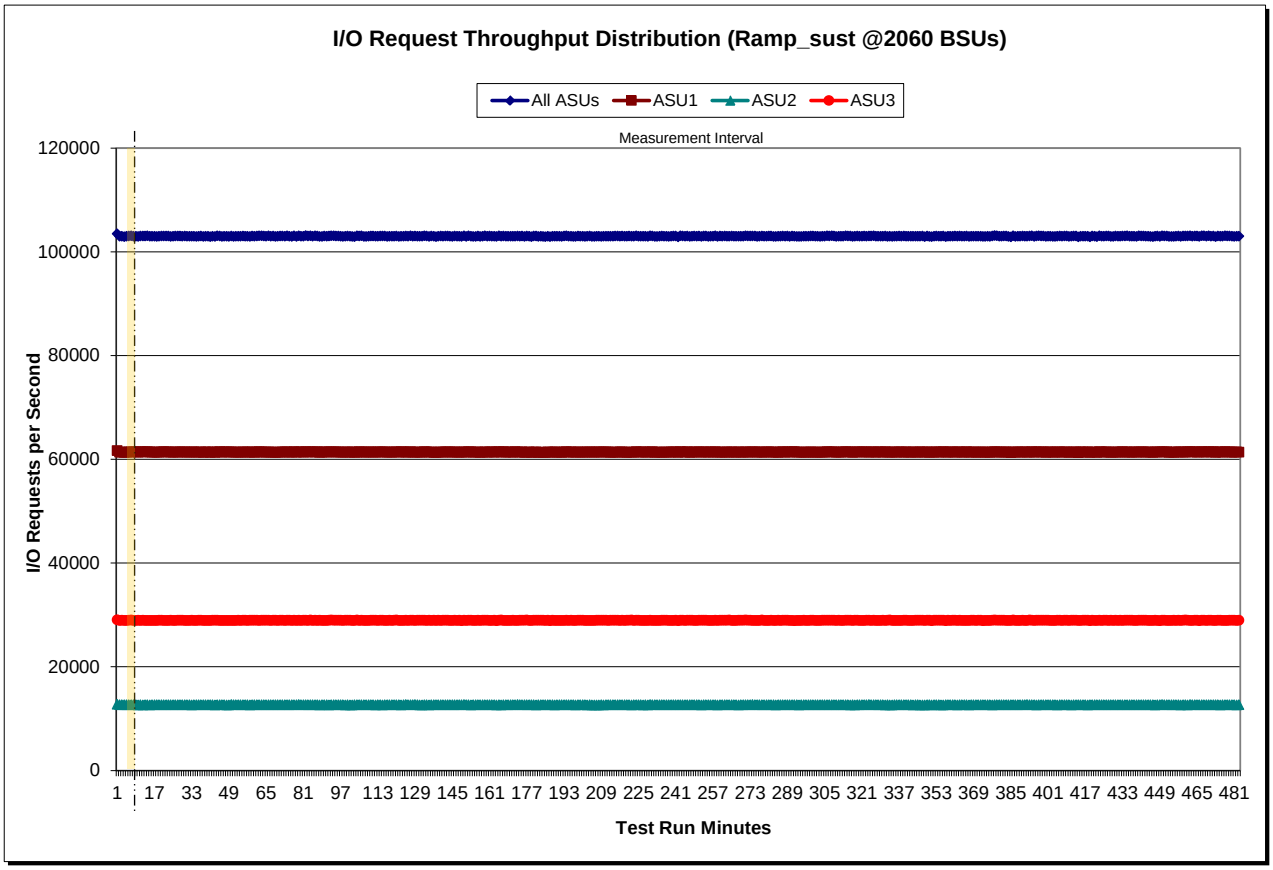


Sustainability – I/O Request Throughput Distribution Data

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

Sustainability I/O Request Throughput Table

Sustainability – I/O Request Throughput Distribution Graph

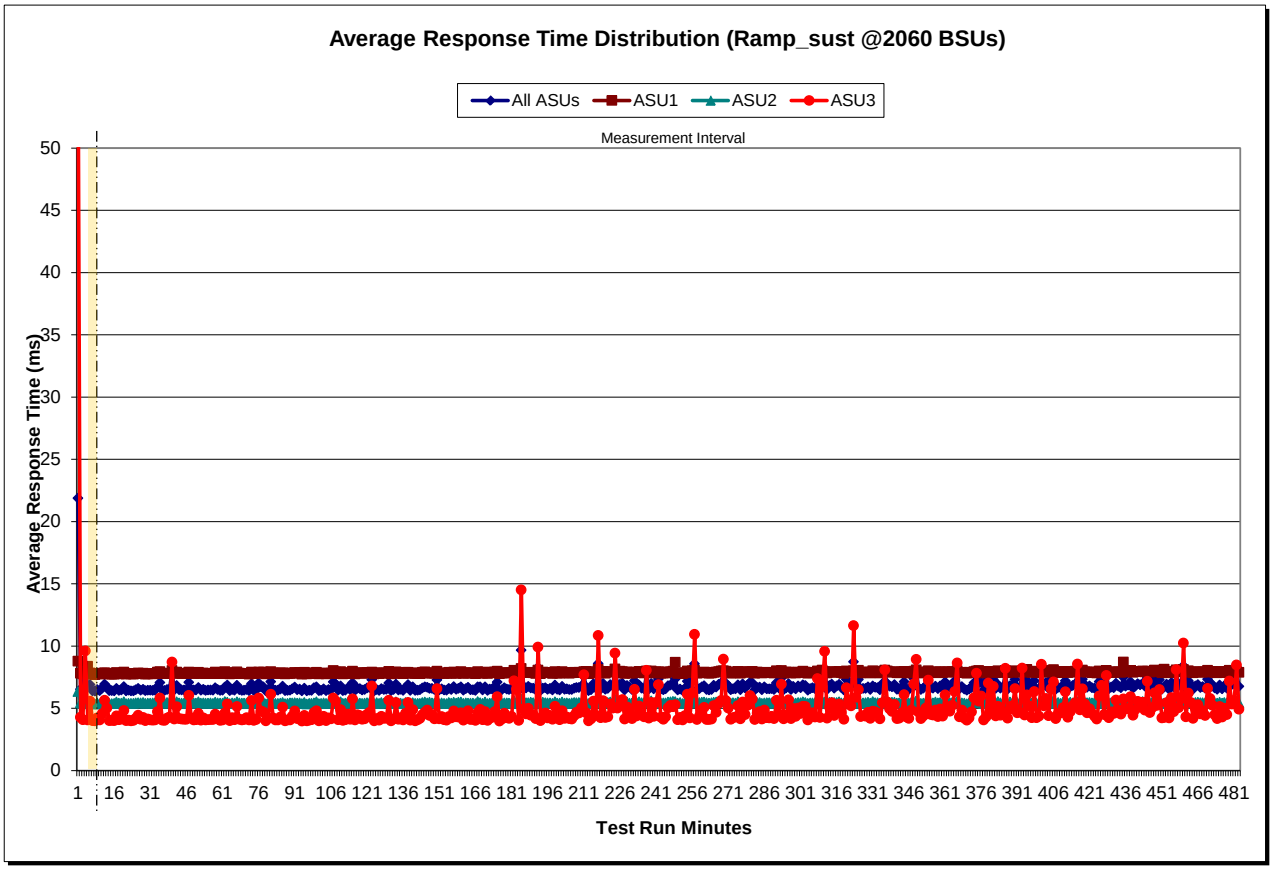


Sustainability – Average Response Time (ms) Distribution Data

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

Sustainability Average Response Time Table

Sustainability – Average Response Time (ms) Distribution Graph



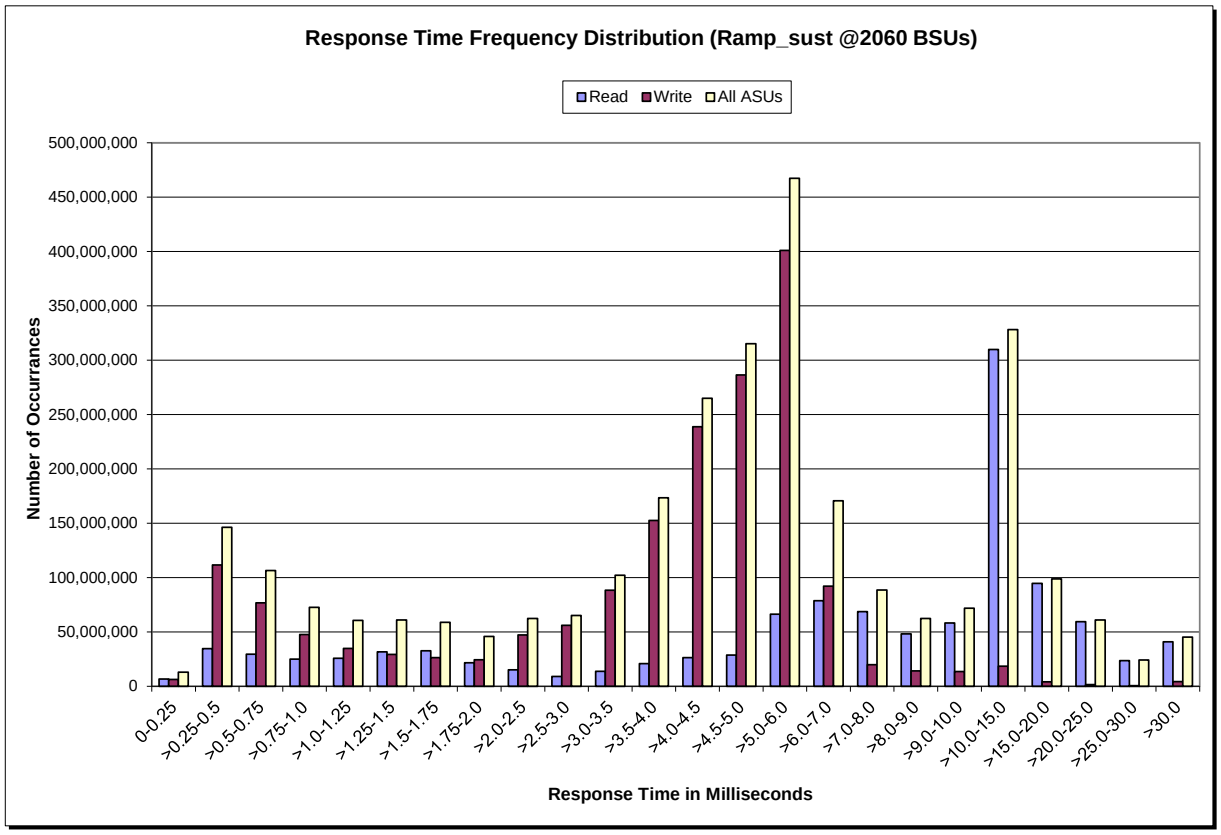
Sustainability – Response Time Frequency Distribution Data

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	6,670,703	34,608,921	29,593,162	25,100,089	25,878,767	31,634,731	32,601,282	21,558,914
Write	6,383,559	111,696,199	76,838,584	47,632,009	34,773,339	29,362,457	26,292,245	24,353,281
All ASUs	13,054,262	146,305,120	106,431,746	72,732,098	60,652,106	60,997,188	58,893,527	45,912,195
ASU1	8,669,525	77,122,412	54,956,428	39,570,167	35,072,678	37,561,396	37,152,018	27,516,409
ASU2	1,813,077	19,770,365	15,480,122	10,836,395	9,361,167	9,817,903	9,612,260	7,207,736
ASU3	2,571,660	49,412,343	35,995,196	22,325,536	16,218,261	13,617,889	12,129,249	11,188,050

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	15,205,082	9,080,425	13,758,212	20,814,073	26,285,990	28,723,795	66,261,975	78,686,616
Write	47,195,541	56,051,229	88,323,537	152,542,261	238,675,432	286,416,201	400,997,524	92,024,625
All ASUs	62,400,623	65,131,654	102,081,749	173,356,334	264,961,422	315,139,996	467,259,499	170,711,241
ASU1	31,890,086	30,672,450	48,110,906	81,289,593	122,746,365	144,935,751	224,303,458	105,360,495
ASU2	8,983,805	9,298,213	14,920,918	24,547,622	34,364,611	37,032,294	50,625,190	19,907,779
ASU3	21,526,732	25,160,991	39,049,925	67,519,119	107,850,446	133,171,951	192,330,851	45,442,967

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	68,711,683	48,147,204	58,176,631	309,735,872	94,596,128	59,434,904	23,695,189	40,964,722
Write	19,951,729	14,181,811	13,637,061	18,460,974	4,217,133	1,676,940	534,872	4,259,103
All ASUs	88,663,412	62,329,015	71,813,692	328,196,846	98,813,261	61,111,844	24,230,061	45,223,825
ASU1	67,191,680	47,280,261	55,784,645	281,391,434	90,115,165	56,801,417	22,771,212	39,715,978
ASU2	11,579,937	8,105,730	9,379,725	38,669,302	7,351,653	3,567,812	1,109,408	1,524,007
ASU3	9,891,795	6,943,024	6,649,322	8,136,110	1,346,443	742,615	349,441	3,983,840

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 92.

IOPS Test Results File

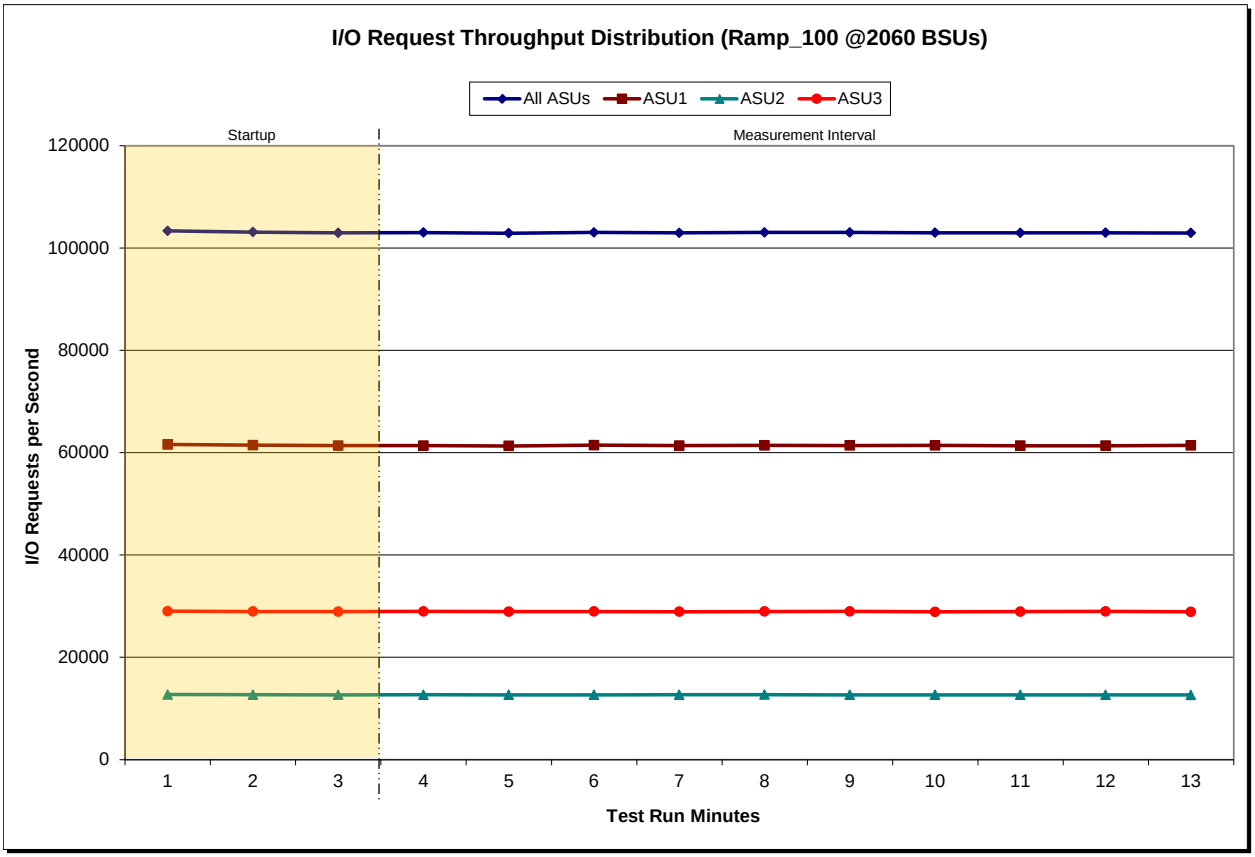
A link to the test results file generated from the IOPS Test Run is listed below.

IOPS Test Results File

IOPS Test Run – I/O Request Throughput Distribution Data

2,060 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	8:31:29	8:34:29	0-2	0:03:00
Measurement Interval	8:34:29	8:44:29	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	103,345.97	61,597.38	12,722.37	29,026.22
1	103,105.18	61,461.22	12,684.58	28,959.38
2	102,963.37	61,373.87	12,657.88	28,931.62
3	103,026.42	61,381.30	12,675.53	28,969.58
4	102,899.85	61,305.38	12,654.27	28,940.20
5	103,053.58	61,453.40	12,657.47	28,942.72
6	102,964.28	61,376.10	12,676.25	28,911.93
7	103,050.03	61,410.28	12,696.60	28,943.15
8	103,035.98	61,391.83	12,673.53	28,970.62
9	102,979.32	61,418.42	12,667.03	28,893.87
10	102,958.92	61,365.48	12,659.05	28,934.38
11	102,987.25	61,356.20	12,668.25	28,962.80
12	102,938.12	61,412.82	12,648.45	28,876.85
Average	***102,989.38***	***61,387.12***	***12,667.64***	***28,934.61***

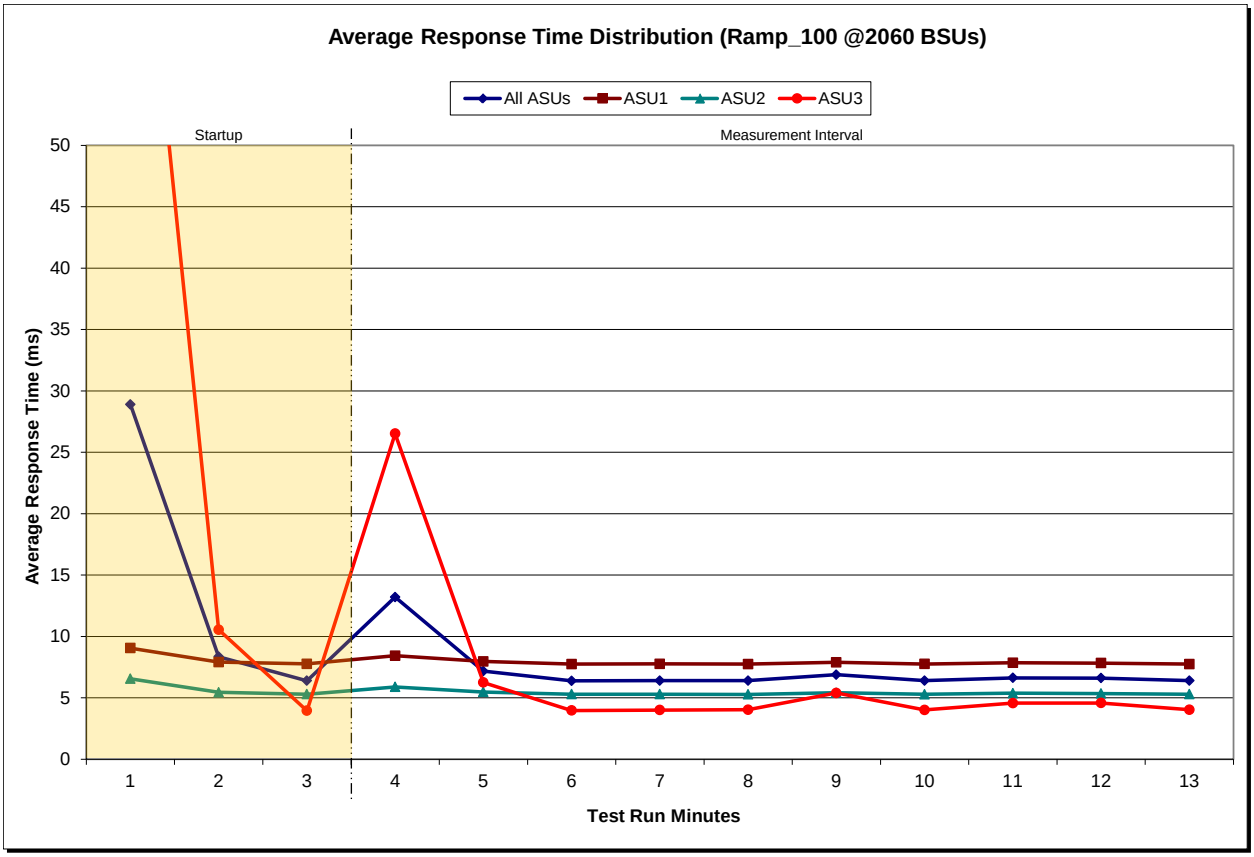
IOPS Test Run – I/O Request Throughput Distribution Graph



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

2,060 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	8:31:29	8:34:29	0-2	0:03:00
Measurement Interval	8:34:29	8:44:29	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	28.89	9.05	6.55	80.78
1	8.34	7.91	5.44	10.54
2	6.39	7.76	5.30	3.95
3	13.20	8.43	5.89	26.51
4	7.17	7.96	5.47	6.25
5	6.38	7.75	5.29	3.96
6	6.40	7.76	5.30	4.00
7	6.40	7.74	5.27	4.03
8	6.88	7.89	5.41	5.40
9	6.40	7.76	5.28	4.01
10	6.63	7.85	5.38	4.57
11	6.61	7.82	5.35	4.58
12	6.40	7.74	5.29	4.03
Average	***7.25***	***7.87***	***5.39***	***6.73***

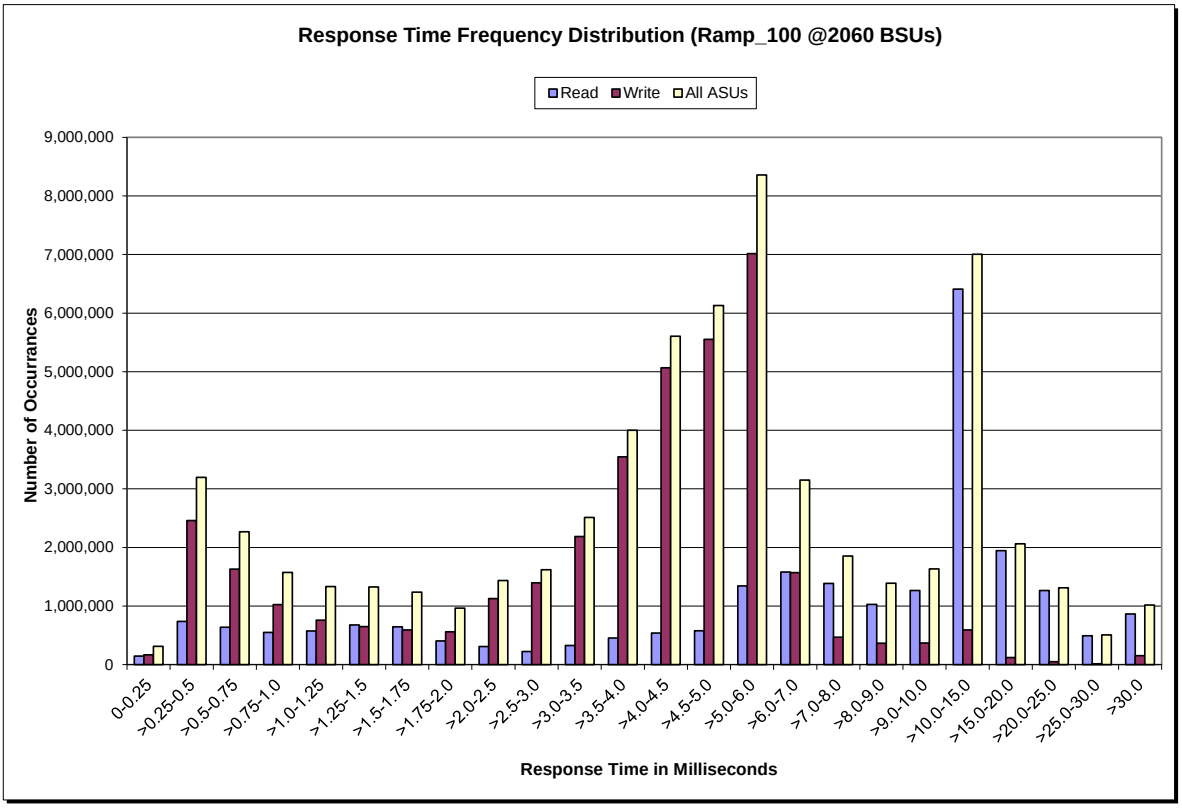
IOPS Test Run – Average Response Time (ms) Distribution Graph



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	146,377	736,433	636,722	549,963	574,966	675,501	643,846	403,815
Write	164,998	2,459,423	1,629,955	1,022,442	758,486	648,916	592,881	560,977
All ASUs	311,375	3,195,856	2,266,677	1,572,405	1,333,452	1,324,417	1,236,727	964,792
ASU1	202,119	1,673,320	1,173,276	858,957	772,138	811,411	764,715	559,017
ASU2	42,480	431,945	330,729	235,458	208,394	213,152	198,842	148,385
ASU3	66,776	1,090,591	762,672	477,990	352,920	299,854	273,170	257,390
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	308,275	224,741	326,038	454,090	539,036	577,053	1,343,589	1,580,517
Write	1,127,992	1,394,994	2,185,694	3,546,484	5,066,421	5,551,388	7,013,621	1,570,832
All ASUs	1,436,267	1,619,735	2,511,732	4,000,574	5,605,457	6,128,441	8,357,210	3,151,349
ASU1	719,061	764,355	1,182,752	1,868,599	2,595,693	2,830,489	4,077,599	1,999,583
ASU2	203,846	230,285	361,166	550,725	705,721	706,658	907,162	375,488
ASU3	513,360	625,095	967,814	1,581,250	2,304,043	2,591,294	3,372,449	776,278
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	1,386,597	1,026,222	1,265,300	6,409,702	1,943,626	1,263,281	493,104	865,551
Write	466,830	364,224	366,876	593,518	118,780	48,212	13,883	150,765
All ASUs	1,853,427	1,390,446	1,632,176	7,003,220	2,062,406	1,311,493	506,987	1,016,316
ASU1	1,382,622	1,033,543	1,243,315	5,920,288	1,864,744	1,212,162	475,324	846,741
ASU2	242,275	180,856	213,820	828,830	157,794	78,043	22,712	25,755
ASU3	228,530	176,047	175,041	254,102	39,868	21,288	8,951	143,820

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
61,792,937	60,776,621	1,016,316

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	0.0701	0.2100	0.0180	0.0700	0.0350	0.2809
COV	0.001	0.001	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 92.

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 - 2,060 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	8:31:29	8:34:29	0-2	0:03:00
Measurement Interval	8:34:29	8:44:29	3-12	0:10:00

(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	103,345.97	61,597.38	12,722.37	29,026.22
1	103,105.18	61,461.22	12,684.58	28,959.38
2	102,963.37	61,373.87	12,657.88	28,931.62
3	103,026.42	61,381.30	12,675.53	28,969.58
4	102,899.85	61,305.38	12,654.27	28,940.20
5	103,053.58	61,453.40	12,657.47	28,942.72
6	102,964.28	61,376.10	12,676.25	28,911.93
7	103,050.03	61,410.28	12,696.60	28,943.15
8	103,035.98	61,391.83	12,673.53	28,970.62
9	102,979.32	61,418.42	12,667.03	28,893.87
10	102,958.92	61,365.48	12,659.05	28,934.38
11	102,987.25	61,356.20	12,668.25	28,962.80
12	102,938.12	61,412.82	12,648.45	28,876.85
Average	***102,989.38***	***61,387.12***	***12,667.64***	***28,934.61***

95% Load Level - 1,957 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	8:46:19	8:49:19	0-2	0:03:00
Measurement Interval	8:49:19	8:59:19	3-12	0:10:00

(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	98,206.17	58,517.52	12,082.07	27,606.58
1	97,784.52	58,261.87	12,035.98	27,486.67
2	97,857.67	58,299.55	12,034.30	27,523.82
3	97,802.27	58,307.52	12,013.45	27,481.30
4	97,877.53	58,330.03	12,042.77	27,504.73
5	97,855.78	58,315.88	12,040.38	27,499.52
6	97,806.92	58,293.10	12,035.38	27,478.43
7	97,853.18	58,329.03	12,040.22	27,483.93
8	97,798.47	58,319.15	12,024.55	27,454.77
9	97,823.23	58,284.28	12,030.12	27,508.83
10	97,870.37	58,295.58	12,038.13	27,536.65
11	97,937.05	58,406.17	12,029.07	27,501.82
12	97,931.73	58,346.20	12,054.53	27,531.00
Average	***97,855.65***	***58,322.70***	***12,034.86***	***27,498.10***

90% Load Level - 1,855 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	9:01:08	9:04:08	0-2	0:03:00
Measurement Interval	9:04:08	9:14:08	3-12	0:10:00

(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	93,007.15	55,440.68	11,422.30	26,144.17
1	92,755.35	55,287.93	11,421.62	26,045.80
2	92,730.27	55,257.00	11,417.28	26,055.98
3	92,634.57	55,228.50	11,389.03	26,017.03
4	92,724.73	55,307.25	11,389.07	26,028.42
5	92,688.42	55,218.68	11,408.40	26,061.33
6	92,711.87	55,247.25	11,403.62	26,061.00
7	92,673.77	55,251.17	11,379.72	26,042.88
8	92,715.32	55,272.33	11,403.88	26,039.10
9	92,764.52	55,270.67	11,406.73	26,087.12
10	92,693.32	55,242.13	11,394.45	26,056.73
11	92,640.02	55,206.97	11,397.38	26,035.67
12	92,731.92	55,246.10	11,422.50	26,063.32
Average	***92,697.84***	***55,249.11***	***11,399.48***	***26,049.26***

80% Load Level - 1,648 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	9:15:55	9:18:55	0-2	0:03:00
Measurement Interval	9:18:55	9:28:55	3-12	0:10:00

(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	82,746.60	49,305.93	10,198.40	23,242.27
1	82,427.95	49,142.38	10,118.90	23,166.67
2	82,426.40	49,142.83	10,126.95	23,156.62
3	82,433.05	49,127.23	10,129.70	23,176.12
4	82,398.88	49,117.00	10,137.67	23,144.22
5	82,397.42	49,150.82	10,128.68	23,117.92
6	82,393.48	49,136.08	10,117.60	23,139.80
7	82,469.57	49,151.23	10,155.07	23,163.27
8	82,421.88	49,128.08	10,127.38	23,166.42
9	82,402.33	49,126.15	10,113.48	23,162.70
10	82,471.40	49,130.15	10,157.68	23,183.57
11	82,362.17	49,053.82	10,140.37	23,167.98
12	82,388.40	49,104.08	10,117.13	23,167.18
Average	***82,413.86***	***49,122.47***	***10,132.48***	***23,158.92***

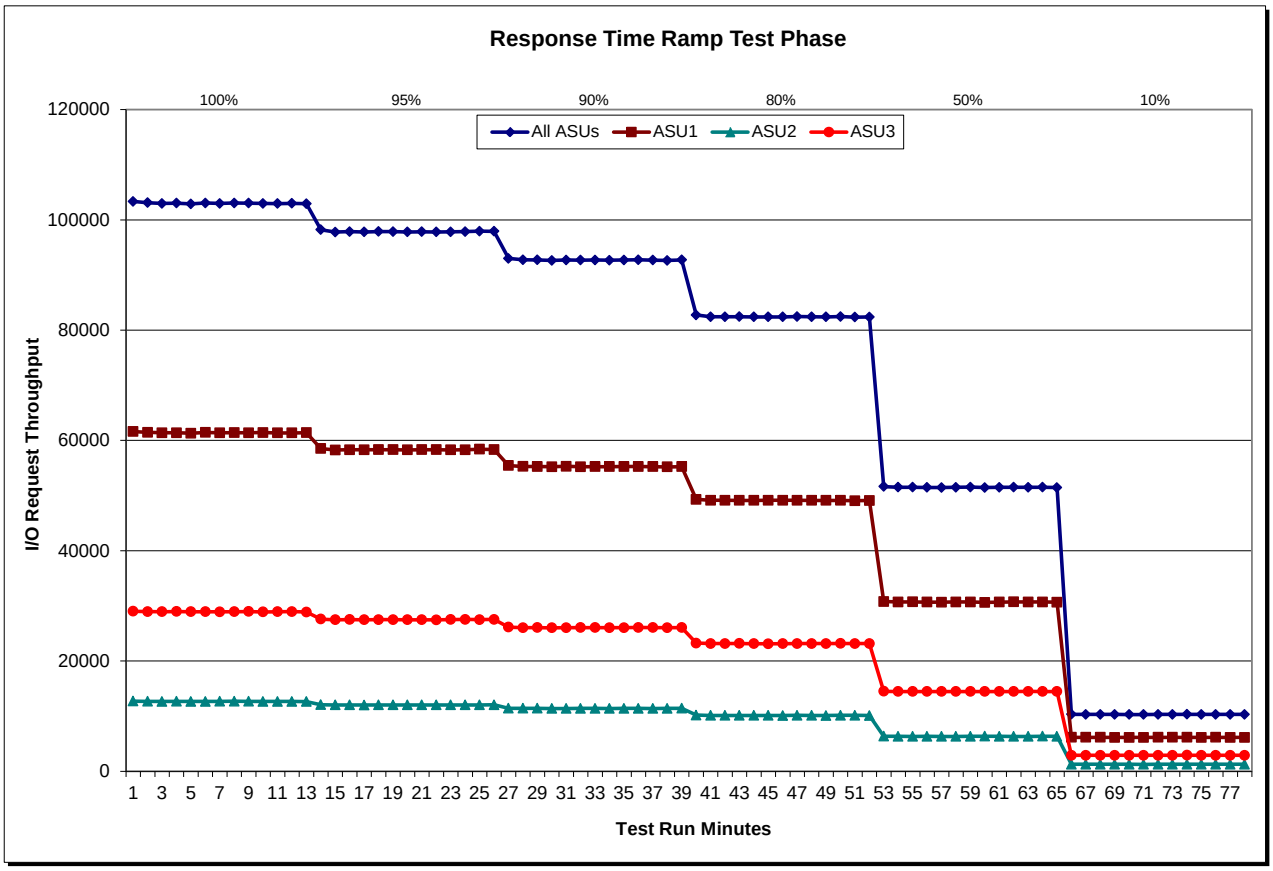
50% Load Level - 1,030 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	9:30:38	9:33:38	0-2	0:03:00
Measurement Interval	9:33:38	9:43:38	3-12	0:10:00

(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	51,641.67	30,772.10	6,354.17	14,515.40
1	51,516.73	30,707.65	6,347.65	14,461.43
2	51,511.82	30,722.18	6,324.62	14,465.02
3	51,468.82	30,671.38	6,332.18	14,465.25
4	51,451.52	30,650.32	6,325.72	14,475.48
5	51,502.87	30,701.32	6,327.10	14,474.45
6	51,525.82	30,707.53	6,345.90	14,472.38
7	51,447.15	30,627.62	6,354.48	14,465.05
8	51,498.73	30,680.40	6,344.70	14,473.63
9	51,509.90	30,711.05	6,324.20	14,474.65
10	51,501.22	30,695.93	6,326.82	14,478.47
11	51,507.68	30,691.28	6,350.53	14,465.87
12	51,456.58	30,657.37	6,334.00	14,465.22
Average	***51,487.03***	***30,679.42***	***6,336.56***	***14,471.05***

10% Load Level - 206 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	9:45:14	9:48:14	0-2	0:03:00
Measurement Interval	9:48:14	9:58:14	3-12	0:10:00

(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3
0	10,324.27	6,163.83	1,267.52	2,892.92
1	10,310.00	6,156.52	1,266.45	2,887.03
2	10,318.55	6,163.43	1,271.13	2,883.98
3	10,312.63	6,144.05	1,268.73	2,899.85
4	10,299.77	6,139.87	1,266.78	2,893.12
5	10,289.82	6,128.13	1,261.45	2,900.23
6	10,312.92	6,147.08	1,266.83	2,899.00
7	10,303.75	6,153.33	1,263.15	2,887.27
8	10,333.77	6,147.27	1,270.92	2,915.58
9	10,301.42	6,139.47	1,267.58	2,894.37
10	10,314.07	6,152.73	1,265.05	2,896.28
11	10,301.95	6,141.57	1,266.12	2,894.27
12	10,305.25	6,133.93	1,270.12	2,901.20
Average	***10,307.53***	***6,142.74***	***1,266.67***	***2,898.12***

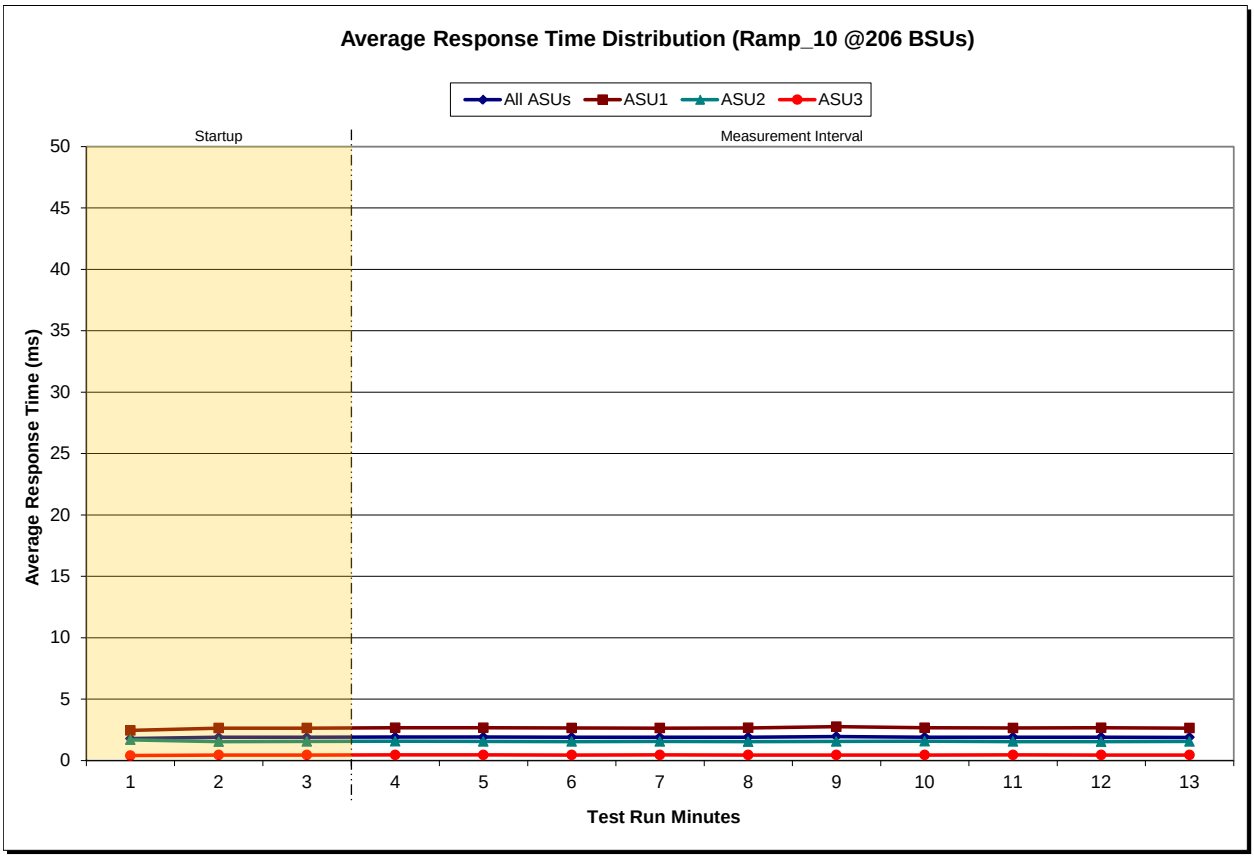
Response Time Ramp Distribution (IOPS) Graph



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

206 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	9:45:14	9:48:14	0-2	0:03:00
Measurement Interval	9:48:14	9:58:14	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	1.78	2.45	1.68	0.40
1	1.89	2.64	1.53	0.44
2	1.89	2.64	1.55	0.45
3	1.91	2.67	1.57	0.45
4	1.91	2.66	1.55	0.45
5	1.89	2.65	1.55	0.45
6	1.89	2.64	1.55	0.46
7	1.90	2.66	1.54	0.45
8	1.96	2.75	1.55	0.45
9	1.90	2.66	1.57	0.45
10	1.90	2.65	1.54	0.45
11	1.90	2.66	1.54	0.45
12	1.89	2.63	1.55	0.45
Average	***1.90***	***2.66***	***1.55***	***0.45***

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	0.2809	0.0699	0.2101	0.0180	0.0699	0.0350	0.2812
COV	0.007	0.002	0.004	0.003	0.012	0.003	0.008	0.002

Repeatability Test

Clause 5.4.5

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

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

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

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

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

Clause 9.4.3.7.4

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 92.

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	***102,989.38***
Repeatability Test Phase 1	102,988.59
Repeatability Test Phase 2	103,025.77

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	***1.90***
Repeatability Test Phase 1	1.89
Repeatability Test Phase 2	1.89

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)...

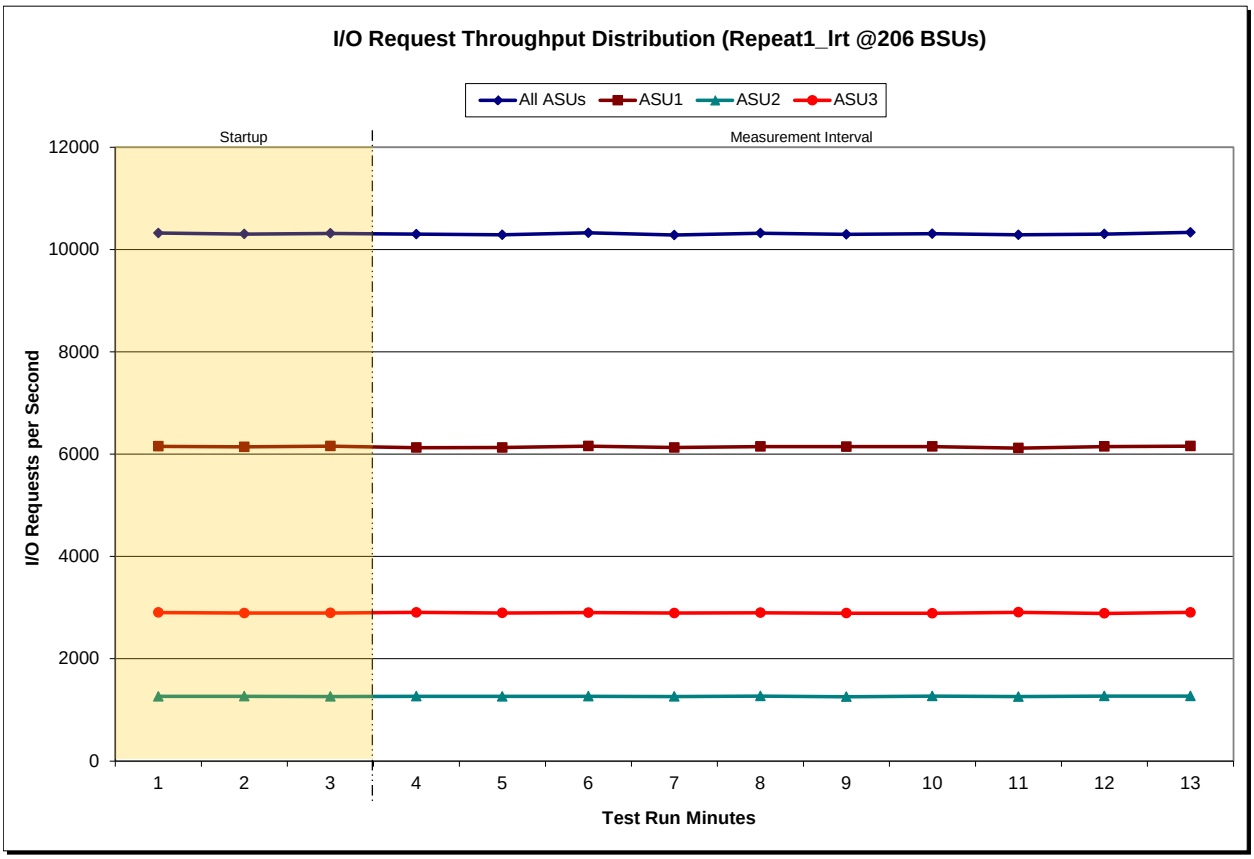
A link to the test result file generated from each Repeatability Test Run is listed below.

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

206 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	11:30:31	11:33:31	0-2	0:03:00
Measurement Interval	11:33:31	11:43:31	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	10,321.18	6,152.80	1,264.97	2,903.42
1	10,301.87	6,141.00	1,267.00	2,893.87
2	10,315.63	6,156.35	1,263.08	2,896.20
3	10,299.25	6,127.83	1,266.20	2,905.22
4	10,288.03	6,128.90	1,263.93	2,895.20
5	10,326.68	6,156.68	1,267.37	2,902.63
6	10,282.77	6,128.63	1,262.08	2,892.05
7	10,319.13	6,148.63	1,271.35	2,899.15
8	10,296.18	6,146.28	1,258.62	2,891.28
9	10,308.05	6,148.00	1,270.88	2,889.17
10	10,286.42	6,117.72	1,260.75	2,907.95
11	10,303.32	6,147.75	1,269.80	2,885.77
12	10,335.58	6,157.23	1,271.48	2,906.87
Average	***10,304.54***	***6,140.77***	***1,266.25***	***2,897.53***

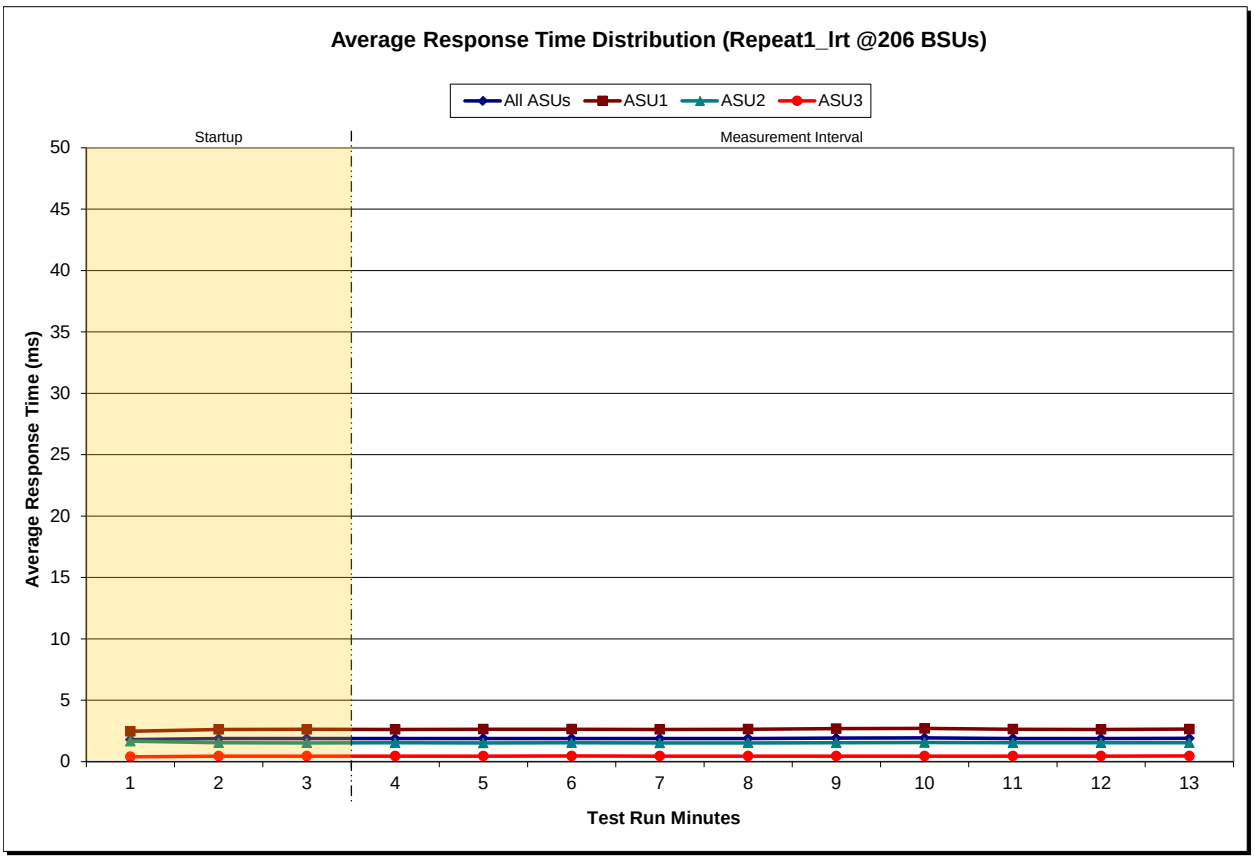
Repeatability 1 LRT – I/O Request Throughput Distribution Graph



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

206 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	11:30:31	11:33:31	0-2	0:03:00
Measurement Interval	11:33:31	11:43:31	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	1.79	2.47	1.66	0.39
1	1.87	2.62	1.54	0.45
2	1.88	2.63	1.53	0.45
3	1.87	2.62	1.53	0.45
4	1.88	2.64	1.52	0.44
5	1.89	2.64	1.53	0.45
6	1.87	2.62	1.52	0.45
7	1.89	2.64	1.52	0.45
8	1.91	2.68	1.53	0.45
9	1.93	2.70	1.55	0.45
10	1.88	2.64	1.53	0.45
11	1.88	2.62	1.53	0.44
12	1.89	2.64	1.53	0.45
Average	***1.89***	***2.64***	***1.53***	***0.45***

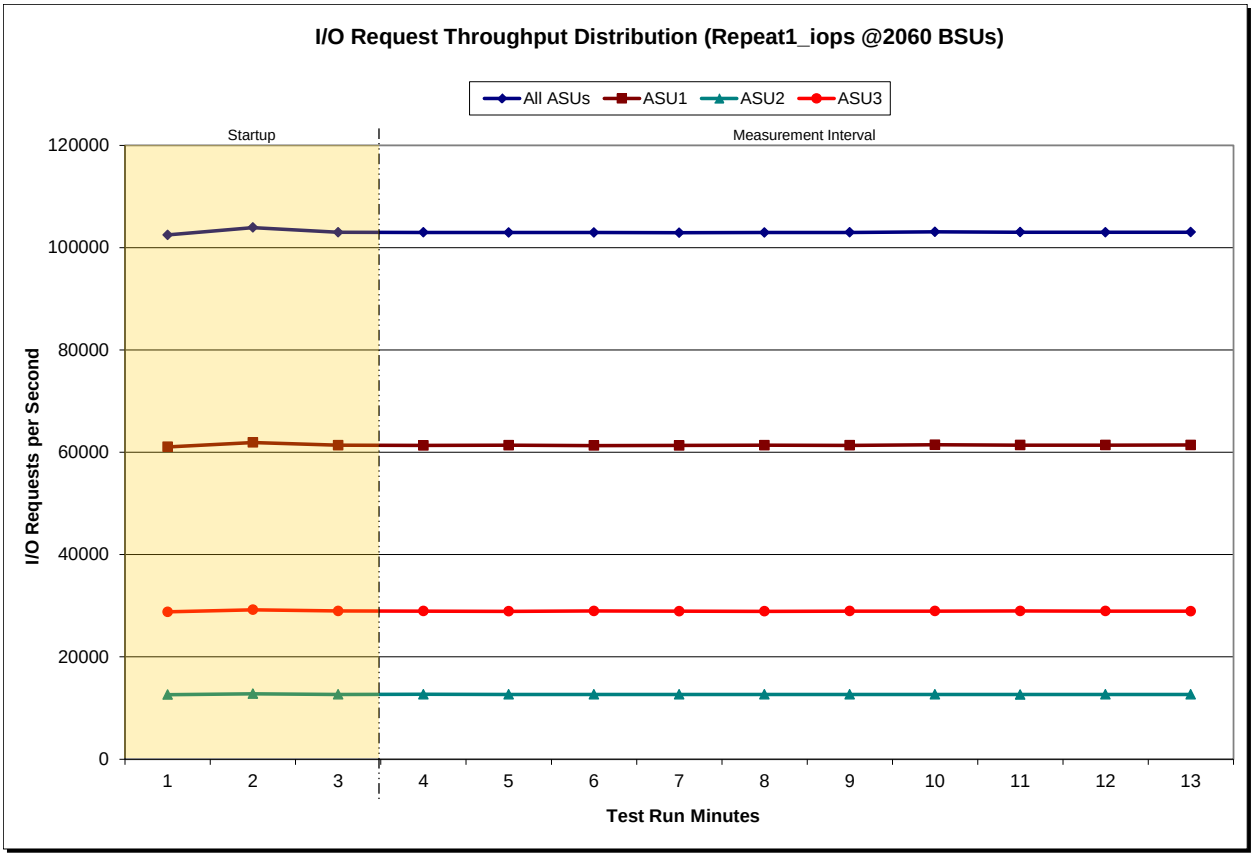
Repeatability 1 LRT –Average Response Time (ms) Distribution Graph



Repeatability 1 IOPS – I/O Request Throughput Distribution Data

2,060 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	11:45:20	11:48:20	0-2	0:03:00
Measurement Interval	11:48:20	11:58:20	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	102,478.22	61,060.70	12,624.88	28,792.63
1	103,923.52	61,923.50	12,791.55	29,208.47
2	103,005.23	61,372.72	12,663.20	28,969.32
3	102,973.72	61,341.85	12,687.88	28,943.98
4	102,952.07	61,379.10	12,672.83	28,900.13
5	102,966.28	61,322.20	12,666.93	28,977.15
6	102,925.40	61,343.32	12,656.20	28,925.88
7	102,962.78	61,381.47	12,662.47	28,918.85
8	102,974.57	61,361.25	12,664.33	28,948.98
9	103,080.63	61,454.02	12,673.12	28,953.50
10	103,026.60	61,402.87	12,652.82	28,970.92
11	103,006.77	61,405.20	12,653.03	28,948.53
12	103,017.10	61,419.28	12,673.15	28,924.67
Average	***102,988.59***	***61,381.06***	***12,666.28***	***28,941.26***

Repeatability 1 IOPS – I/O Request Throughput Distribution Graph

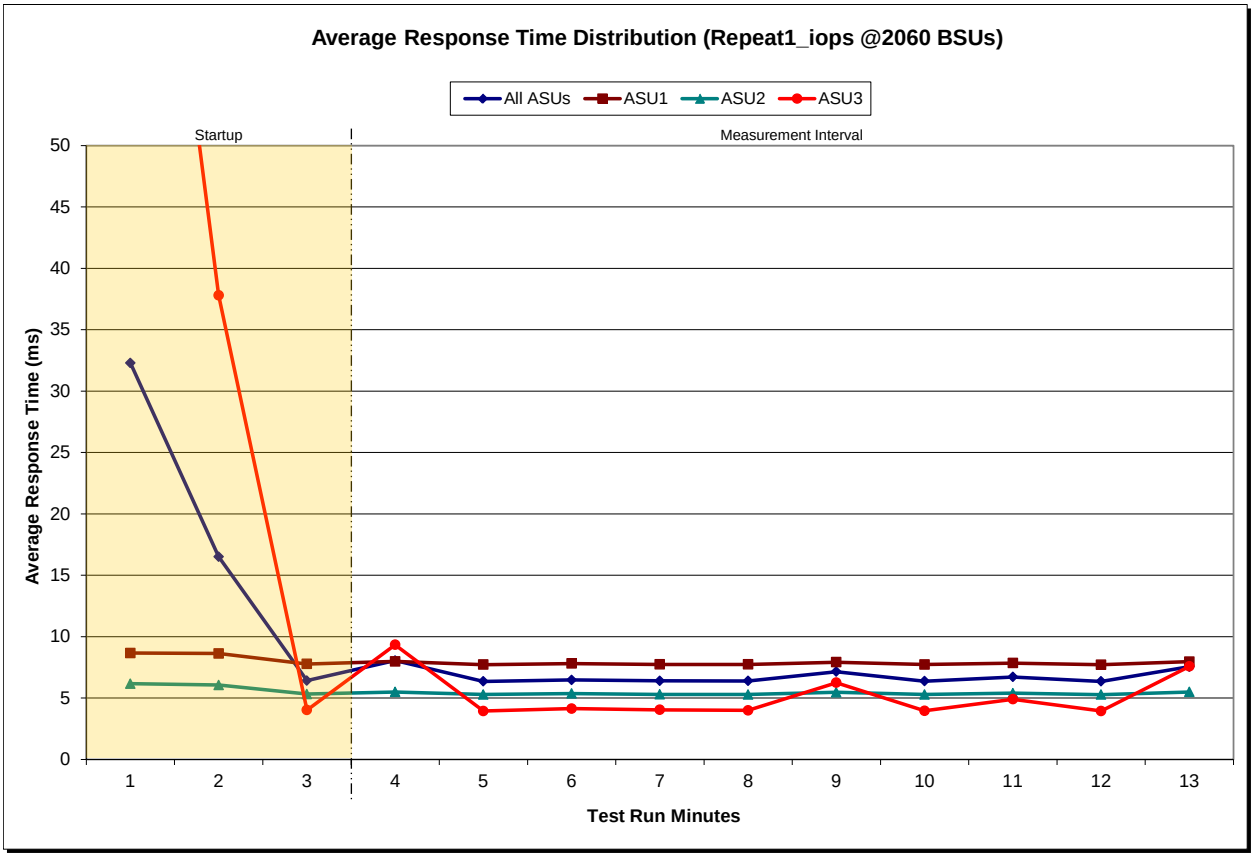


Repeatability 1 IOPS –Average Response Time (ms) Distribution Data

2,060 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	11:45:20	11:48:20	0-2	0:03:00
Measurement Interval	11:48:20	11:58:20	3-12	0:10:00

60 second intervals	All ASUs	ASU1	ASU2	ASU3
0	32.28	8.66	6.16	93.82
1	16.51	8.63	6.06	37.80
2	6.42	7.77	5.32	4.02
3	8.05	7.97	5.49	9.34
4	6.36	7.71	5.29	3.94
5	6.47	7.80	5.37	4.14
6	6.40	7.73	5.30	4.04
7	6.39	7.74	5.29	3.99
8	7.15	7.92	5.47	6.24
9	6.37	7.73	5.29	3.96
10	6.72	7.85	5.40	4.90
11	6.35	7.71	5.27	3.94
12	7.55	7.96	5.49	7.57
Average	***6.78***	***7.81***	***5.37***	***5.21***

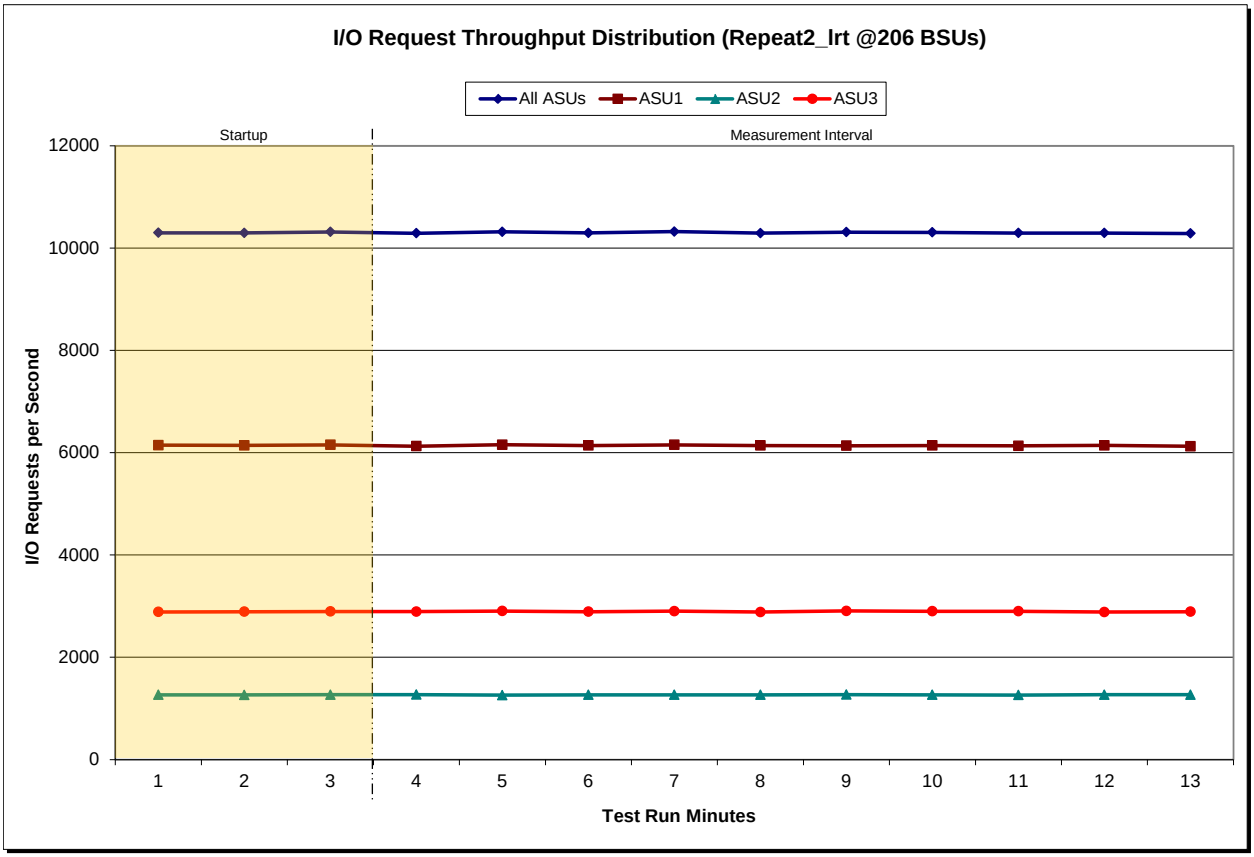
Repeatability 1 IOPS –Average Response Time (ms) Distribution Graph



Repeatability 2 LRT – I/O Request Throughput Distribution Data

206 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	12:00:07	12:03:07	0-2	0:03:00
Measurement Interval	12:03:07	12:13:07	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	10,298.10	6,146.03	1,265.68	2,886.38
1	10,296.70	6,141.52	1,263.28	2,891.90
2	10,314.43	6,152.00	1,267.50	2,894.93
3	10,289.60	6,127.57	1,269.88	2,892.15
4	10,318.22	6,153.97	1,260.13	2,904.12
5	10,295.98	6,140.37	1,265.62	2,890.00
6	10,322.00	6,153.28	1,266.85	2,901.87
7	10,290.55	6,138.80	1,266.37	2,885.38
8	10,311.33	6,134.68	1,271.23	2,905.42
9	10,306.52	6,139.03	1,267.25	2,900.23
10	10,293.90	6,132.83	1,262.65	2,898.42
11	10,293.42	6,141.17	1,267.50	2,884.75
12	10,284.42	6,125.75	1,268.23	2,890.43
Average	***10,300.59***	***6,138.75***	***1,266.57***	***2,895.28***

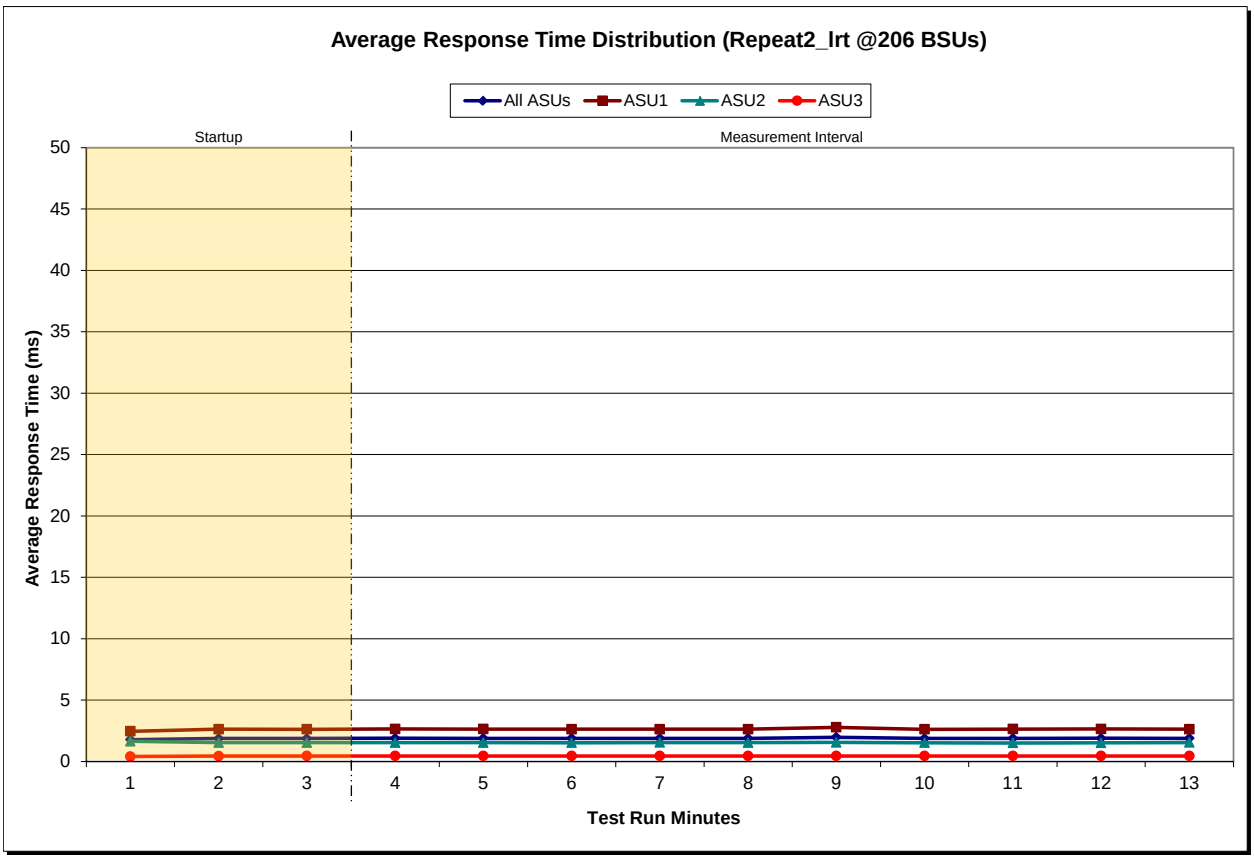
Repeatability 2 LRT – I/O Request Throughput Distribution Graph



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

206 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	12:00:07	12:03:07	0-2	0:03:00
Measurement Interval	12:03:07	12:13:07	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	1.78	2.46	1.64	0.40
1	1.88	2.63	1.53	0.44
2	1.88	2.62	1.53	0.44
3	1.89	2.65	1.53	0.45
4	1.89	2.63	1.53	0.45
5	1.88	2.63	1.52	0.45
6	1.88	2.63	1.55	0.45
7	1.88	2.63	1.53	0.45
8	1.97	2.78	1.55	0.45
9	1.88	2.62	1.53	0.45
10	1.88	2.64	1.51	0.45
11	1.89	2.65	1.52	0.45
12	1.88	2.63	1.54	0.45
Average	***1.89***	***2.65***	***1.53***	***0.45***

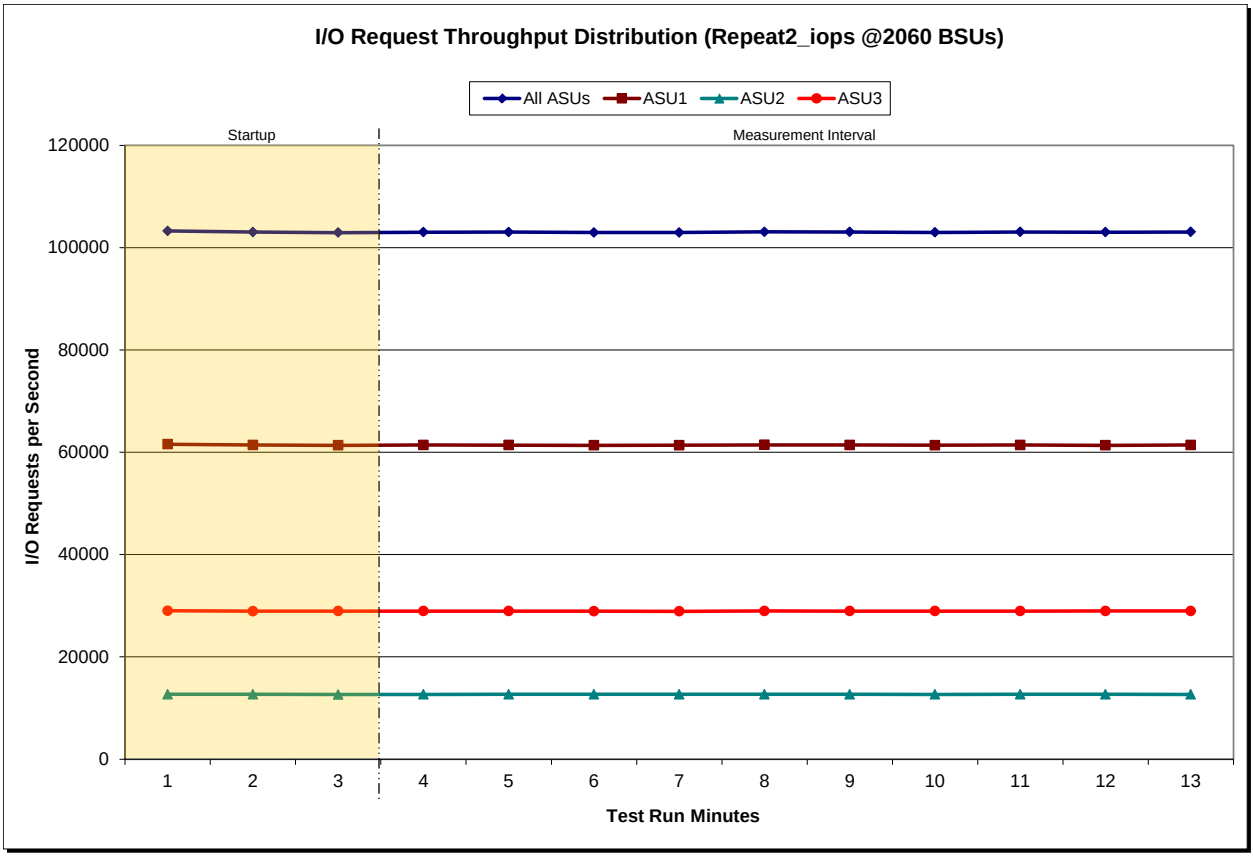
Repeatability 2 LRT –Average Response Time (ms) Distribution Graph



Repeatability 2 IOPS – I/O Request Throughput Distribution Data

2,060 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	12:14:56	12:17:56	0-2	0:03:00
Measurement Interval	12:17:56	12:27:56	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	103,269.12	61,566.55	12,682.22	29,020.35
1	103,046.03	61,417.22	12,691.60	28,937.22
2	102,942.32	61,354.18	12,636.67	28,951.47
3	103,026.57	61,410.95	12,667.92	28,947.70
4	103,034.83	61,393.63	12,683.45	28,957.75
5	102,951.88	61,347.88	12,676.17	28,927.83
6	102,965.37	61,376.02	12,676.33	28,913.02
7	103,088.30	61,435.98	12,678.93	28,973.38
8	103,058.38	61,424.43	12,679.52	28,954.43
9	102,984.52	61,374.75	12,658.63	28,951.13
10	103,062.83	61,419.57	12,683.18	28,960.08
11	103,014.85	61,353.83	12,694.67	28,966.35
12	103,070.17	61,426.22	12,673.22	28,970.73
Average	***103,025.77***	***61,396.33***	***12,677.20***	***28,952.24***

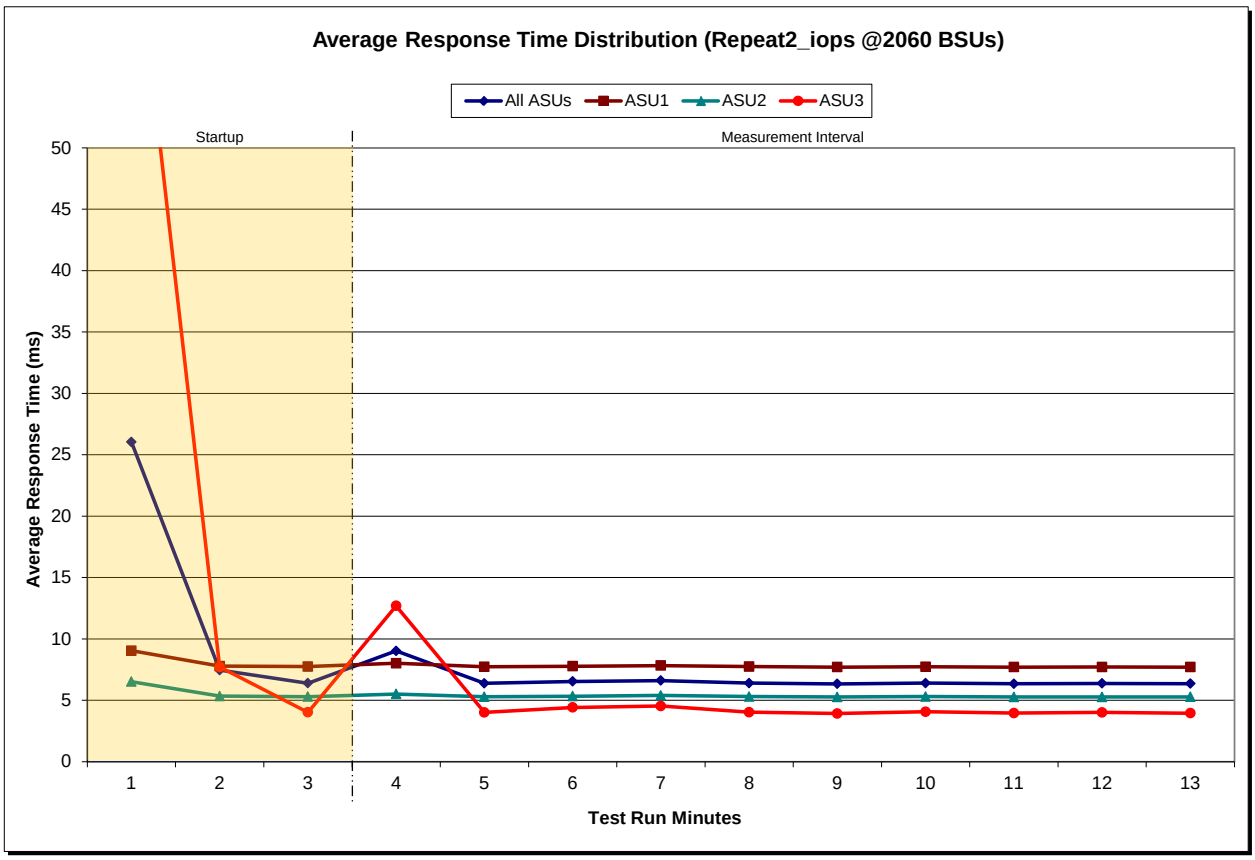
Repeatability 2 IOPS – I/O Request Throughput Distribution Graph



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

2,060 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	12:14:56	12:17:56	0-2	0:03:00
Measurement Interval	12:17:56	12:27:56	3-12	0:10:00
60 second intervals	**All ASUs**	**ASU1**	**ASU2**	**ASU3**
0	26.03	9.04	6.51	70.59
1	7.46	7.78	5.34	7.71
2	6.39	7.74	5.29	4.02
3	9.02	8.02	5.51	12.69
4	6.38	7.73	5.29	4.01
5	6.53	7.77	5.33	4.41
6	6.60	7.83	5.40	4.54
7	6.40	7.74	5.30	4.03
8	6.34	7.70	5.27	3.92
9	6.40	7.73	5.30	4.06
10	6.35	7.70	5.27	3.96
11	6.37	7.71	5.27	4.01
12	6.35	7.70	5.27	3.95
Average	***6.67***	***7.76***	***5.32***	***4.96***

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	0.2810	0.0700	0.2099	0.0180	0.0700	0.0349	0.2812
COV	0.007	0.002	0.004	0.003	0.008	0.004	0.008	0.003

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

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
COV	0.002	0.000	0.001	0.001	0.003	0.001	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	0.0350	0.2809	0.0701	0.2100	0.0180	0.0699	0.0350	0.2811
COV	0.006	0.002	0.005	0.002	0.011	0.005	0.004	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	0.2809	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
COV	0.001	0.001	0.001	0.001	0.004	0.002	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 92.

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	171,350,816
Total Number of Logical Blocks Verified	126,630,960
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	512
Number of Failed I/O Requests in the process of the Test	0

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 Fujitsu Storage Systems ETERNUS DX440 S2 as documented in this Full Disclosure Report is currently available for customer purchase and shipment.

PRICING INFORMATION

Clause 9.4.3.3.6

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

Pricing information may be found in the Priced Storage Configuration Pricing section on page 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 differences between the Tested Storage Configuration (TSC) and the Priced Storage Configuration.

A list of all differences between the Tested Storage Configuration (TSC) and Priced Storage Configuration may be found in the Executive Summary portion of this document on page 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 Remote Audit of the Fujitsu Storage Systems ETERNUS DX440 S2 .

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.

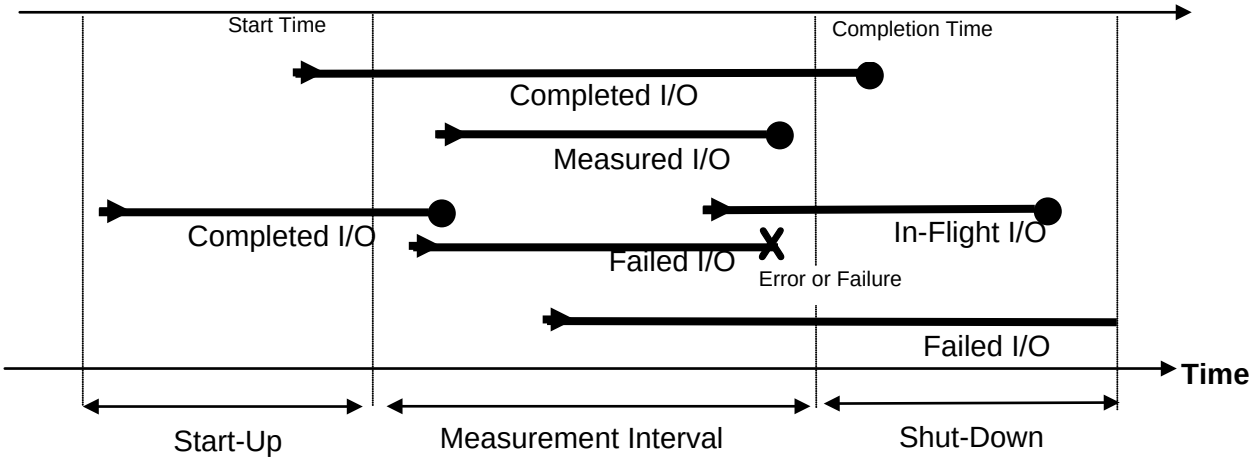
Test: A collection of Test Phases and or Test Runs sharing a common objective.

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

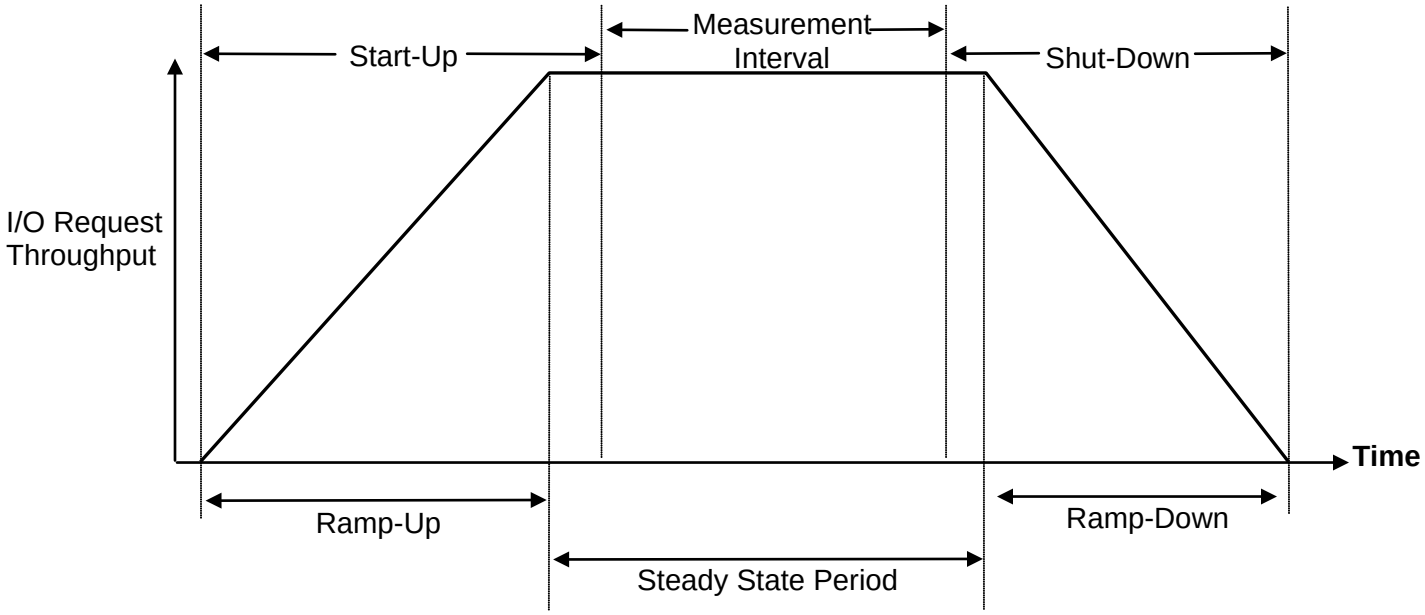
period, Shut-Down period, and Ramp-Down period as illustrated in the "SPC-1 Test Run Components" below. All SPC-1 Test Runs shall have a Steady State period and a Measurement Interval.

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

QLogic HBA Driver

The following parameters were set in the QLogic HBA driver configuration file, **qlc.conf**, on the Host System:

pci-max-read-request=2048
(This parameter sets the maximum request of a read request.)

ddi-forceattach=1
(This parameter value forces the Solaris kernel to load all instances of the driver during boot.)

Solaris Operating System

The Solaris system parameter file, **etc/system**, included the following entry:

ssd:ssd_max_throttle = 12
(This entry sets the maximum number of requests per target LUN.)

Both of the above changes were made during the initial setup of the Solaris Host System.

APPENDIX C: TESTED STORAGE CONFIGURATION (TSC) CREATION

The standard Fujitsu Command Line tool (CLI) was used to create the ETERNUS DX440 S2 SPC-1 configuration. The Solaris Host System had the **cygwin** packages, **expect** and **openssh**, installed to enable execution the CLI scripts referenced below.

The 'master' script, **doFDRcfg.sh**, was executed, which in turn, invoked the scripts, **DX440S2_R10_95rg_570lun_120524.exp** and **makesol**.

The **DX440S2_R10_95rg_570lun_120524.exp** script completed steps 1-4, described below for the initial 8 host port configuration. The **makesol** script completed step 5 described below.

An additional 8 host ports were added to the configuration and the script, **DX440S2_R10_95rg_570lun_16ch_120604.exp**, was executed to remap to the 16 host ports.

Each **expect** script included the **docli** procedure, which was used to issue the CLI commands to the array. That procedure used **ssh** for communication with the array. A second procedure in the script, **doexit**, was used to conclude the execution sequence at the end of the script.

Step 1 – Creation of RAID Groups

A total of 95 RAID Groups were created, according to the configuration plan, **DX440S2_R10_95rg_570lun_16ch_120604.xlsx**, which is typically prepared in concert with a Fujitsu SE. Each RAID Group was made up of 8 disk drives in a RAID10 (4+4) configuration and assigned to a specific CM and CPU within the CM for operational control. The RAID Groups were named R10-0 through R10-94.

Step 2 – Creation of the Global Hot Spares

Two drives were designated as the Global Hot Spares in slot 20 of DE27 & DE37, per the configuration plan.

Step 2 – Creation of the Logical Volumes

Within each of the RAID Groups, six Logical Volumes were created in each of RAID Groups RG10-0 through RG10-10 *(11 groups, 66 LVs)* with size set to 101,532MB each for ASU-3. Six Logical Volumes were created in each of RAID Groups RG-11 through RG10-65 *(55 groups, 330 LVs)* with size set to 91,356MB each for ASU-1. Six Logical Volumes were created in each of RAID Groups R10-66 through R10-94 *(29 groups, 174 LVs)* with size set to 173,265MB each for ASU2.

Step 4 – Assignment of LUN Mapping to the Solaris Host Systems

The **DX440S2_R10_95rg_570lun_120524.exp** script provided mapping to eight host ports. An additional 8 host ports were added to the configuration and the script, **DX440S2_R10_95rg_570lun_16ch_120604.exp**, was executed to remap to the 16 host ports.

First all of the previous mapping between Host LUNs and the defined Logical Volumes was released. Then port LUN mapping was assigned for each of the Logical Volumes using all four ports on each of the two Channel Adapters (CA) in each of the two Controller Modules (CM). LUNs 0-35 were assigned to all of the ports except ports 0 & 2 on CM1 CA1 which were each assigned LUNs 0-32. Each Logical Volume was assigned one Host addressing LUN through one of the 16 channels between the Solaris Host system and the ETERNUS storage array

Step 5 – Creation of the Solaris Logical Volumes

The input file, **DX440S2_R10_95rg_570lun_16ch_120602_svmake_higuchi.txt**, was used by the **makesol** script to create the Solaris Logical Volumes (SVMs) that comprise the SPC-1 ASUs. The input text file provides the specific list of LUNs used to make up each of the SVM: 6 SVMs for ASU-1, 6 for ASU-2, and 6 for ASU-3.

The list of LUNs in each SVM are selected according to the above referenced configuration plan with the LUN addresses recognized by Solaris. Each SVM includes one LUN (LV) from each of the RAID Groups assigned to the specific ASU as noted is Step 3 above. The assignments are verified with the content of the **metastat** report from Solaris that links back to the specific volume assignments within the ETERNUS storage array

Referenced Scripts and Input File

doFDRcfg.sh

```
#!/usr/bin/bash
#
# Do the configuration steps required for the SPC1 benchmark
#
# create tmp directory for spc1 if it does not exist
if [ ! -d /tmp/spc1 ]; then
mkdir /tmp/spc1
fi
#
# confID uniquely identifies the configuration of the array
confID=DX440S2_R10_95rg_570lun_120524
#
# obtain cjobID based on the timestamp
# cjobID uniquely identifies the configuration job
cjobID=C`date +%y%m%d%H%M%S`
#
echo job start time `date` > /tmp/spc1/${cjobID}_message.txt
echo This is an array configuration job >> /tmp/spc1/${cjobID}_message.txt
echo job confID=$confID >> /tmp/spc1/${cjobID}_message.txt
echo job cjobID=$cjobID >> /tmp/spc1/${cjobID}_message.txt
./sendstatus.sh "Starting Configuration Job=${cjobID}" ${cjobID}_message.txt
#
# Configure Array using the Expect script to issue CLI commands
./sendstatus.sh "Starting Eternus CLI script for configuration Job=${cjobID}"
${cjobID}_message.txt
cd ../configure
./${confID}.exp
cd ../scripts
./sendstatus.sh "Completed Eternus CLI script for configuration Job=${cjobID}"
${cjobID}_message.txt
#
./sendstatus.sh "Starting makesol script for solaris VM creation Job=${cjobID}"
${cjobID}_message.txt
cd ../solvm;
# Create SVM (Solaris Volume Manager) Volumes
./makesol ${confID}_svmake.txt
metastat >${confID}_metastat.txt
cd ../scripts
./sendstatus.sh "Completed makesol script for solaris VM creation Job=${cjobID}"
${cjobID}_message.txt
#
# Wait for physical format to complete
#
PollingInterval=1200 #wait 20 minutes to check format status
./sendstatus.sh "Waiting for physical format to complete Job=${cjobID}"
${cjobID}_message.txt
#
./showFormatStatus.exp dx440s2 root /tmp/spc1/fmt_${cjobID}.txt
#
LUNS=`grep Available /tmp/spc1/fmt_${cjobID}.txt|wc|awk '{print $1}'`
while [ $LUNS -gt 0 ]; do
 echo "-------------------------------------------"  >>
/tmp/spc1/${cjobID}_message.txt
 cat  /tmp/spc1/fmt_${cjobID}.txt >> /tmp/spc1/${cjobID}_message.txt
 ./sendstatus.sh "Currently formatting $LUNS LUNS Job=${cjobID}"
${cjobID}_message.txt
 sleep $PollingInterval
```

```
  ./showFormatStatus.exp dx440s2 root /tmp/spc1/fmt_${cjobID}.txt
 LUNS=`grep Available /tmp/spc1/fmt_${cjobID}.txt |wc |awk '{print $1}'`
done
./sendstatus.sh "Physical format complete please proceed. Job=${cjobID}"
${cjobID}_message.txt
```

DX440S2_R10_95rg_570lun_120524.exp

```
#!/usr/local/bin/expect
# script to setup initial configuration for dx440s2
# for SPC-2 benchmark
set timeout 600
set user root
set password root
spawn ssh dx440s2 -l $user
expect "CLI>"
# procedure to execute dx440s2 cli command
proc docli { cmd args} {
send "$cmd $args\r"
expect "CLI>"
}
# procedure to exit
proc doexit {} {
send "exit \r"
}

#D440S2 RAID10(4+4)x94RG,570lun SPC1-BMT, 2012.04.27

#  Create raid-groups (0 - 94)

docli create raid-group -name RG10-0 -disks 2000-2003,3000-3003 -level 10 -assigned-
cm cm0cpu0
docli create raid-group -name RG10-1 -disks 0000-0003,1000-1003 -level 10 -assigned-
cm cm0cpu1
docli create raid-group -name RG10-2 -disks 2004-2007,3004-3007 -level 10 -assigned-
cm cm1cpu0
docli create raid-group -name RG10-3 -disks 0004-0007,1004-1007 -level 10 -assigned-
cm cm1cpu1
docli create raid-group -name RG10-4 -disks 2008-2011,3008-3011 -level 10 -assigned-
cm cm0cpu0
docli create raid-group -name RG10-5 -disks 0008-0011,1008-1011 -level 10 -assigned-
cm cm0cpu1
docli create raid-group -name RG10-6 -disks 2012-2015,3012-3015 -level 10 -assigned-
cm cm1cpu0
docli create raid-group -name RG10-7 -disks 0012-0015,1012-1015 -level 10 -assigned-
cm cm1cpu1
docli create raid-group -name RG10-8 -disks 2016-2019,3016-3019 -level 10 -assigned-
cm cm0cpu0
docli create raid-group -name RG10-9 -disks 0016-0019,1016-1019 -level 10 -assigned-
cm cm0cpu1

docli create raid-group -name RG10-10 -disks 2020-2023,3020-3023 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-11 -disks 0020-0023,1020-1023 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-12 -disks 2100-2103,3100-3103 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-13 -disks 0100-0103,1100-1103 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-14 -disks 2104-2107,3104-3107 -level 10 -
assigned-cm cm1cpu0
```

```
docli create raid-group -name RG10-15 -disks 0104-0107,1104-1107 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-16 -disks 2108-2111,3108-3111 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-17 -disks 0108-0111,1108-1111 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-18 -disks 2112-2115,3112-3115 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-19 -disks 0112-0115,1112-1115 -level 10 -
assigned-cm cm1cpu1

docli create raid-group -name RG10-20 -disks 2116-2119,3116-3119 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-21 -disks 0116-0119,1116-1119 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-22 -disks 2120-2123,3120-3123 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-23 -disks 0120-0123,1120-1123 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-24 -disks 2200-2203,3200-3203 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-25 -disks 0200-0203,1200-1203 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-26 -disks 2204-2207,3204-3207 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-27 -disks 0204-0207,1204-1207 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-28 -disks 2208-2211,3208-3211 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-29 -disks 0208-0211,1208-1211 -level 10 -
assigned-cm cm0cpu1

docli create raid-group -name RG10-30 -disks 2212-2215,3212-3215 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-31 -disks 0212-0215,1212-1215 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-32 -disks 2216-2219,3216-3219 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-33 -disks 0216-0219,1216-1219 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-34 -disks 2220-2223,3220-3223 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-35 -disks 0220-0223,1220-1223 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-36 -disks 2300-2303,3300-3303 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-37 -disks 0300-0303,1300-1303 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-38 -disks 2304-2307,3304-3307 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-39 -disks 0304-0307,1304-1307 -level 10 -
assigned-cm cm1cpu1

docli create raid-group -name RG10-40 -disks 2308-2311,3308-3311 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-41 -disks 0308-0311,1308-1311 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-42 -disks 2312-2315,3312-3315 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-43 -disks 0312-0315,1312-1315 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-44 -disks 2316-2319,3316-3319 -level 10 -
assigned-cm cm0cpu0
```

```
docli create raid-group -name RG10-45 -disks 0316-0319,1316-1319 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-46 -disks 2320-2323,3320-3323 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-47 -disks 0320-0323,1320-1323 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-48 -disks 2400-2403,3400-3403 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-49 -disks 0400-0403,1400-1403 -level 10 -
assigned-cm cm0cpu1

docli create raid-group -name RG10-50 -disks 2404-2407,3404-3407 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-51 -disks 0404-0407,1404-1407 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-52 -disks 2408-2411,3408-3411 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-53 -disks 0408-0411,1408-1411 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-54 -disks 2412-2415,3412-3415 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-55 -disks 0412-0415,1412-1415 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-56 -disks 2416-2419,3416-3419 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-57 -disks 0416-0419,1416-1419 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-58 -disks 2420-2423,3420-3423 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-59 -disks 0420-0423,1420-1423 -level 10 -
assigned-cm cm1cpu1

docli create raid-group -name RG10-60 -disks 2500-2503,3500-3503 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-61 -disks 0500-0503,1500-1503 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-62 -disks 2504-2507,3504-3507 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-63 -disks 0504-0507,1504-1507 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-64 -disks 2508-2511,3508-3511 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-65 -disks 0508-0511,1508-1511 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-66 -disks 2512-2515,3512-3515 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-67 -disks 0512-0515,1512-1515 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-68 -disks 2516-2519,3516-3519 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-69 -disks 0516-0519,1516-1519 -level 10 -
assigned-cm cm0cpu1

docli create raid-group -name RG10-70 -disks 2520-2523,3520-3523 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-71 -disks 0520-0523,1520-1523 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-72 -disks 2600-2603,3600-3603 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-73 -disks 0600-0603,1600-1603 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-74 -disks 2604-2607,3604-3607 -level 10 -
assigned-cm cm1cpu0
```

```
docli create raid-group -name RG10-75 -disks 0604-0607,1604-1607 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-76 -disks 2608-2611,3608-3611 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-77 -disks 0608-0611,1608-1611 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-78 -disks 2612-2615,3612-3615 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-79 -disks 0612-0615,1612-1615 -level 10 -
assigned-cm cm1cpu1

docli create raid-group -name RG10-80 -disks 2616-2619,3616-3619 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-81 -disks 0616-0619,1616-1619 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-82 -disks 2620-2623,3620-3623 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-83 -disks 0620-0623,1620-1623 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-84 -disks 2700-2703,3700-3703 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-85 -disks 0700-0703,1700-1703 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-86 -disks 2704-2707,3704-3707 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-87 -disks 0704-0707,1704-1707 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-88 -disks 2708-2711,3708-3711 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-89 -disks 0708-0711,1708-1711 -level 10 -
assigned-cm cm0cpu1

docli create raid-group -name RG10-90 -disks 2712-2715,3712-3715 -level 10 -
assigned-cm cm1cpu0
docli create raid-group -name RG10-91 -disks 0712-0715,1712-1715 -level 10 -
assigned-cm cm1cpu1
docli create raid-group -name RG10-92 -disks 2716-2719,3716-3719 -level 10 -
assigned-cm cm0cpu0
docli create raid-group -name RG10-93 -disks 0716-0719,1716-1719 -level 10 -
assigned-cm cm0cpu1
docli create raid-group -name RG10-94 -disks 2720-2723,3720-3723 -level 10 -
assigned-cm cm1cpu1


#  Assign two Global Hot Spare drives

docli set global-spare -disks 2720,3720

#  Create Logical Volumes - six within each RAID Group

#    Logical Volumes for ASU3 using 11 RAID Groups
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-0 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-1 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-2 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-3 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-4 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-5 -type open -size
101532mb
```

```
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-6 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-7 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-8 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-9 -type open -size
101532mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-10 -type open -size
101532mb

#    Logical Volumes for ASU1 using 55 RAID Groups
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-11 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-12 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-13 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-14 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-15 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-16 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-17 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-18 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-19 -type open -size
91356mb

docli create volume -name RG10-_Vol -count 6 -rg-name RG10-20 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-21 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-22 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-23 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-24 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-25 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-26 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-27 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-28 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-29 -type open -size
91356mb

docli create volume -name RG10-_Vol -count 6 -rg-name RG10-30 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-31 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-32 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-33 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-34 -type open -size
91356mb
```

```
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-35 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-36 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-37 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-38 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-39 -type open -size
91356mb

docli create volume -name RG10-_Vol -count 6 -rg-name RG10-40 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-41 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-42 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-43 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-44 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-45 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-46 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-47 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-48 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-49 -type open -size
91356mb

docli create volume -name RG10-_Vol -count 6 -rg-name RG10-50 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-51 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-52 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-53 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-54 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-55 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-56 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-57 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-58 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-59 -type open -size
91356mb

docli create volume -name RG10-_Vol -count 6 -rg-name RG10-60 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-61 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-62 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-63 -type open -size
91356mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-64 -type open -size
91356mb
```

```
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-65 -type open -size
91356mb

#    Logical Volumes for ASU2 using 29 RAID Groups
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-66 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-67 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-68 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-69 -type open -size
173265mb

docli create volume -name RG10-_Vol -count 6 -rg-name RG10-70 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-71 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-72 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-73 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-74 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-75 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-76 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-77 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-78 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-79 -type open -size
173265mb

docli create volume -name RG10-_Vol -count 6 -rg-name RG10-80 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-81 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-82 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-83 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-84 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-85 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-86 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-87 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-88 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-89 -type open -size
173265mb

docli create volume -name RG10-_Vol -count 6 -rg-name RG10-90 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-91 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-92 -type open -size
173265mb
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-93 -type open -size
173265mb
```

```
docli create volume -name RG10-_Vol -count 6 -rg-name RG10-94 -type open -size
173265mb

#  Create Port Mapping for CM0 CA0 Ports 0 & 1
docli set mapping -port 000 -volume-number 0-5,48-53,96-101,144-149,192-197,240-245
-lun 0-35
docli set mapping -port 000 -volume-number 288-293,336-341,384-389,432-437,480-
485,528-533 -lun 36-71
docli set mapping -port 001 -volume-number 6-11,54-59,102-107,150-155,198-203,246-
251 -lun 0-35
docli set mapping -port 001 -volume-number 294-299,342-347,390-395,438-443,486-
491,534-539 -lun 36-71

#  Create Port Mapping for CM0 CA1 Ports 0 & 1
docli set mapping -port 010 -volume-number 24-29,72-77,120-125,168-173,216-221,264-
269 -lun 0-35
docli set mapping -port 010 -volume-number 312-317,360-365,408-413,456-461,504-
509,552-557 -lun 36-71
docli set mapping -port 011 -volume-number 30-35,78-83,126-131,174-179,222-227,270-
275 -lun 0-35
docli set mapping -port 011 -volume-number 318-323,366-371,414-419,462-467,510-
515,558-563 -lun 36-71

#  Create Port Mapping for CM1 CA0 Ports 0 & 1
docli set mapping -port 100 -volume-number 12-17,60-65,108-113,156-161,204-209,252-
257 -lun 0-35
docli set mapping -port 100 -volume-number 300-305,348-353,396-401,444-449,492-
497,540-545 -lun 36-71
docli set mapping -port 101 -volume-number 18-23,66-71,114-119,162-167,210-215,258-
263 -lun 0-35
docli set mapping -port 101 -volume-number 306-311,354-359,402-407,450-455,498-
503,546-551 -lun 36-71

#  Create Port Mapping for CM1 CA1 Ports 0 & 1
docli set mapping -port 110 -volume-number 36-41,84-89,132-137,180-185,228-233,276-
281 -lun 0-35
docli set mapping -port 110 -volume-number 324-329,372-377,420-425,468-473,516-521 -
lun 36-65
docli set mapping -port 111 -volume-number 42-47,90-95,138-143,186-191,234-239,282-
287 -lun 0-35
docli set mapping -port 111 -volume-number 330-335,378-383,426-431,474-479,522-
527,564-569 -lun 36-71

#exit from CLI

doexit
```

makesol

```
#!/bin/ksh
# Usage: usage
#           makesol configFile
#
LABELFILE="/tmp/makesollabel"
STATFILE="/tmp/makesolstat"
AWK=nawk
usage()
{
        echo "\nUsage: $0 configFile\n"
        exit 1
}

labelDisk()
{
     echo "l" > $LABELFILE
     echo "q" >> $LABELFILE
     format -s -f $LABELFILE $1
}

checkStat()
{
     typeset -i i=0
     del1=`grep $1 $STATFILE|$AWK '{ print $1 }'`
     if [ "$del1" != "" ] ; then
            for del in $del1
            do
                   i=0
                   while (( $i < $delete ))
                   do
                          if [ ${DELETE[(($i+1))]} == $del ] ; then
                                 break
                          fi
                          i=$i+1
                   done
                   if (( $i == $delete )) ; then
                          delete=$delete+1
                          DELETE[$delete]=$del
                   fi
            done
     fi
}

getDiskSlice()
{
     vDisks=""
     for disk in ${DISKS[$1]}
     do
            ndisk=`echo $disk|$AWK 'BEGIN { FS="s"} ; { print $1 }'`
            vDisks=$vDisks$ndisk"s"$2" "
     done
}

makevol()
{
     typeset -i count=0
     typeset -i i=0
     typeset -i vcount
        tmp=`/usr/sbin/metastat -p|$AWK '{ print substr( $1, 2, length($1)-1 )}'`
```

```
      if [ "$tmp" == "" ] ; then
            i=0
      else
            for dgroup in $tmp
            do
                  if (( $dgroup > $i )) ; then
                        i=$dgroup
                  fi
            done
            i=$i+1
      fi
      while (( $count < $groups ))
      do
            count=$count+1
 #echo "/usr/sbin/metainit d$i 1 ${DISK_COUNT[$count]} ${DISKS[$count]}
 ${STRIPE[$count]}"
            tmp=`/usr/sbin/metainit d$i 1 ${DISK_COUNT[$count]} ${DISKS[$count]}
 ${STRIPE[$count]}`
            i=$i+1
            if [ "${VCOUNT[$count]}" != "" ] ; then
                  vcount=1
                  while (( $vcount < ${VCOUNT[$count]} ))
                  do
                        getSlice $vcount
                        getDiskSlice $count $num
                        tmp=`/usr/sbin/metainit d$i 1 ${DISK_COUNT[$count]} $vDisks
 ${STRIPE[$count]}`
                        i=$i+1
                        vcount=$vcount+1
                  done
            fi
      done
 }

 checkDisk()
 {
      typeset -i i=0
      tmp=$1"s"
      test=`grep $tmp /etc/vfstab`
      if [ "$test" != "" ] ; then
            echo "Found disk $1 in /etc/vfstab, we really shouldn't use it here"
            exit 4
      fi
      while (( $i < $groups ))
      do
            i=$i+1
            for disk in ${DISKS[$i]}
            do
                  tmp=$1"s0"
                  if [ "$disk" == $tmp ] ; then
                        echo "disk $1 repeated at line $lineno"
                        exit 4
                  fi
            done
         done
         disks=$disks+1
      part=$1"s0"
      DISKS[$groups]=${DISKS[$groups]}$part" "
      tmp=`prtvtoc -h /dev/dsk/$part 2>/dev/null`
      if [ $? != 0 ] ; then
            labelDisk $part
            tmp=`prtvtoc -h /dev/dsk/$part 2>/dev/null`
            if [ $? != 0 ] ; then
```

```
                    echo "prtvtoc failed for $part"
                    exit 4
             fi
      fi
      checkStat $1"s"
}

getSlice()
{
      num=0
         case $1 in
         0)
             num=0
                  ;;
         1)
                  num=1
                  ;;
         2|3|4|5|6)
                  (( num=$1+1 ))
                  ;;
         esac
}

setVtoc()
{
      typeset -i count=0
         typeset -i i=0
         while (( $i < $groups ))
         do
                  i=$i+1
                  for disk in ${DISKS[$i]}
                  do
                    if [ "${VCOUNT[$i]}" != "" ] ; then
                           sectors=`prtvtoc /dev/dsk/$disk 2>/dev/null|grep
"accessible cylinders"|$AWK '{ print $2 }'`
                           seccyl=`prtvtoc /dev/dsk/$disk 2>/dev/null|grep
"sectors/cylinder"|$AWK '{ print $2 }'`
                           (( sectors=$sectors-1 ))
                    fi
                    tmp=`prtvtoc -h /dev/dsk/$disk 2>/dev/null`
                    set $tmp
                    while (( $# > 5 ))
                    do
                           if (( $1 == 2 )) ; then
                                  if [ "${VCOUNT[$i]}" == "" ] ; then
                                         echo "0 4 $3 $4 $5 $6" > $LABELFILE
                                  else
                                         echo "* labelfile" > $LABELFILE
                                         (( secCount=$sectors/${VCOUNT[$i]} ))
                                         count=0
                                                  (( sc=$secCount*$seccyl ))
                                                  fs=$seccyl
                                         while (( $count < ${VCOUNT[$i]} ))
                                         do
                                                (( ls=$fs+$sc ))
                                                getSlice $count
                                                echo "$num 4 $3 $fs $sc $ls" >>
$LABELFILE
                                                           count=$count+1
                                                (( fs=$fs+$sc ))
                                         done
                                  fi
                                  echo "$1 $2 $3 $4 $5 $6" >> $LABELFILE
```

```
                                tmp=`fmthard -s $LABELFILE /dev/rdsk/$disk`
                                break
                        fi
                        shift 6
                  done
            done
      done
}

delGroups()
{
     typeset -i i=0
     if [ $DELETE_ALL == "yes" ] ; then
           tmp=`/usr/sbin/metastat -p |$AWK '{ print $1 }'`
           for del in $tmp
           do
                 tmp=`/usr/sbin/metaclear $del`
                 if [ $? != 0 ] ; then
                       echo "Failed to delete volume $del"
                       exit 4
                 fi
           done
           return
     fi
     while (( $i < $delete ))
     do
           i=$i+1
           tmp=`/usr/sbin/metaclear ${DELETE[$i]}`
           if [ $? != 0 ] ; then
                 echo "Failed to delete volume ${DELETE[$i]}"
                 exit 4
           fi
     done
}

addDisks()
{
        typeset -i diskNum=0
        typeset -i count=$name
        typeset -i jump=1
        diskNum=${label#*d}
        if (( $diskNum < 10 ))
        then
                diskPrefix=`echo $label|awk '{ print substr( $1, 0, length($1)-1 )
}'`
        elif (( $diskNum < 100 ))
        then
                diskPrefix=`echo $label|awk '{ print substr( $1, 0, length($1)-2 )
}'`
        else
                diskPrefix=`echo $label|awk '{ print substr( $1, 0, length($1)-3 )
}'`
        fi
        if [ "$skip" != "" ]
        then
                jump=$skip
        fi
        count=$count-1
        while [ $count != 0 ]
        do
                count=$count-1
                diskNum=$diskNum+$jump
                diskName=$diskPrefix$diskNum
```

```
                checkDisk $diskName
        done

}

checkConfig()
{
        typeset -i lineno=1
        invg="no"
     DELETE_ALL="no"
        while read -r label name skip
        do
                case $label in
                "VOLUME_GROUP:")
                        VGNAME=$VGNAME$name" "
                        invg="yes"
                        groups=$groups+1
                        getSize="yes"
                        ;;
                "#")
                        ;;
                "")
                        ;;
            "VOLUME")
                        if [ "$invg" != "yes" ]
                        then
                                echo "invalid line in config file line=$lineno
data=\"$label $name\""
                         echo "VOLUME line must be in a volume_group definition"
                                exit 4
                        fi
                   tmp=`echo $name|grep ^[1-7]$`
                   if [ "$tmp" == "" ] ; then
                         echo "invalid line in config file line=$lineno
data=\"$label $name\""
                         echo "VOLUME count must be from 1-7"
                         exit 4
                   fi
                   VCOUNT[groups]=$name
                   ;;
                "STRIPE")
                        if [ "$invg" != "yes" ]
                        then
                                echo "invalid line in config file line=$lineno
data=\"$label $name\""
                                echo "STRIPE line must be in a volume_group
definition"
                                exit 4
                        fi
                        STRIPE[groups]="-i $name"
                        ;;
            "DELETE_ALL")
                   DELETE_ALL="yes"
                   ;;
                "END")
                   DISK_COUNT[$groups]=$disks
                   disks=0
                        invg="no"
                        ;;
                *)
                        if [ "$invg" != "yes" ]
                        then
```

```
                                echo "invalid line in config file line=$lineno
data=\"$label $name\""
                                exit 4
                        fi
                        diskName=$label
                        checkDisk $diskName
                        if [ "$name" != "" ]
                        then
                                addDisks
                        fi
                esac
                lineno=$lineno+1
        done < $CONFIG
}


# main()

        typeset -i delete=0
        typeset -i groups=0
        typeset -i disks=0
        test=`uname -a|grep "Linux"`
        if [ "$test" != "" ]
        then
                AWK=awk
        fi
     case $# in
     1)
           CONFIG=$1
           echo "Doing solvm config from $1"
           ;;
     *)
           usage
           ;;
     esac
     tmp=`/usr/sbin/metadb`
     if [ "$tmp" == "" ] ; then
           echo "No replica database is defined"
           exit 4
     fi
     tmp=`/usr/sbin/metastat -p > $STATFILE`
     checkConfig
     delGroups
     setVtoc
     makevol
```

DX440S2_R10_95rg_570lun_16ch_120602_svmake_higuchi.txt

```
DELETE_ALL
VOLUME_GROUP: asu1-1 (d0)
STRIPE 8m
VOLUME 1
c4t0d3
c5t0d3
c6t0d3
c7t0d3
c8t0d3
c1t0d6
c2t0d6
c3t0d6
c4t0d6
c5t0d6
c6t0d6
c7t0d6
c8t0d6
c1t0d9
c2t0d9
c3t0d9
c4t0d9
c5t0d9
c6t0d9
c7t0d9
c8t0d9
c1t0d12
c2t0d12
c3t0d12
c4t0d12
c5t0d12
c6t0d12
c7t0d12
c8t0d12
c1t0d15
c2t0d15
c3t0d15
c4t0d15
c5t0d15
c6t0d15
c7t0d15
c8t0d15
c1t0d18
c2t0d18
c3t0d18
c4t0d18
c5t0d18
c6t0d18
c7t0d18
c8t0d18
c1t0d21
c2t0d21
c3t0d21
c4t0d21
c5t0d21
c6t0d21
c7t0d21
c8t0d21
c1t0d24
c2t0d24
```

```
END
VOLUME_GROUP: asu1-2 (d1)
STRIPE 8m
VOLUME 1
c4t0d4
c5t0d4
c6t0d4
c7t0d4
c8t0d4
c1t0d7
c2t0d7
c3t0d7
c4t0d7
c5t0d7
c6t0d7
c7t0d7
c8t0d7
c1t0d10
c2t0d10
c3t0d10
c4t0d10
c5t0d10
c6t0d10
c7t0d10
c8t0d10
c1t0d13
c2t0d13
c3t0d13
c4t0d13
c5t0d13
c6t0d13
c7t0d13
c8t0d13
c1t0d16
c2t0d16
c3t0d16
c4t0d16
c5t0d16
c6t0d16
c7t0d16
c8t0d16
c1t0d19
c2t0d19
c3t0d19
c4t0d19
c5t0d19
c6t0d19
c7t0d19
c8t0d19
c1t0d22
c2t0d22
c3t0d22
c4t0d22
c5t0d22
c6t0d22
c7t0d22
c8t0d22
c1t0d25
c2t0d25
END
VOLUME_GROUP: asu1-3 (d2)
STRIPE 8m
VOLUME 1
```

```
  c4t0d5
  c5t0d5
  c6t0d5
  c7t0d5
  c8t0d5
  c1t0d8
  c2t0d8
  c3t0d8
  c4t0d8
  c5t0d8
  c6t0d8
  c7t0d8
  c8t0d8
  c1t0d11
  c2t0d11
  c3t0d11
  c4t0d11
  c5t0d11
  c6t0d11
  c7t0d11
  c8t0d11
  c1t0d14
  c2t0d14
  c3t0d14
  c4t0d14
  c5t0d14
  c6t0d14
  c7t0d14
  c8t0d14
  c1t0d17
  c2t0d17
  c3t0d17
  c4t0d17
  c5t0d17
  c6t0d17
  c7t0d17
  c8t0d17
  c1t0d20
  c2t0d20
  c3t0d20
  c4t0d20
  c5t0d20
  c6t0d20
  c7t0d20
  c8t0d20
  c1t0d23
  c2t0d23
  c3t0d23
  c4t0d23
  c5t0d23
  c6t0d23
  c7t0d23
  c8t0d23
  c1t0d26
  c2t0d26
  END
  VOLUME_GROUP: asu1-4 (d3)
  STRIPE 8m
  VOLUME 1
  c21t0d3
  c22t0d3
  c23t0d3
  c24t0d3
```

```
  c25t0d3
  c18t0d6
  c19t0d6
  c20t0d6
  c21t0d6
  c22t0d6
  c23t0d6
  c24t0d6
  c25t0d6
  c18t0d9
  c19t0d9
  c20t0d9
  c21t0d9
  c22t0d9
  c23t0d9
  c24t0d9
  c25t0d9
  c18t0d12
  c19t0d12
  c20t0d12
  c21t0d12
  c22t0d12
  c23t0d12
  c24t0d12
  c25t0d12
  c18t0d15
  c19t0d15
  c20t0d15
  c21t0d15
  c22t0d15
  c23t0d15
  c24t0d15
  c25t0d15
  c18t0d18
  c19t0d18
  c20t0d18
  c21t0d18
  c22t0d18
  c23t0d18
  c24t0d18
  c25t0d18
  c18t0d21
  c19t0d21
  c20t0d21
  c21t0d21
  c22t0d21
  c23t0d21
  c24t0d21
  c25t0d21
  c18t0d24
  c19t0d24
  END
  VOLUME_GROUP: asu1-5 (d4)
  STRIPE 8m
  VOLUME 1
  c21t0d4
  c22t0d4
  c23t0d4
  c24t0d4
  c25t0d4
  c18t0d7
  c19t0d7
  c20t0d7
```

```
c21t0d7
c22t0d7
c23t0d7
c24t0d7
c25t0d7
c18t0d10
c19t0d10
c20t0d10
c21t0d10
c22t0d10
c23t0d10
c24t0d10
c25t0d10
c18t0d13
c19t0d13
c20t0d13
c21t0d13
c22t0d13
c23t0d13
c24t0d13
c25t0d13
c18t0d16
c19t0d16
c20t0d16
c21t0d16
c22t0d16
c23t0d16
c24t0d16
c25t0d16
c18t0d19
c19t0d19
c20t0d19
c21t0d19
c22t0d19
c23t0d19
c24t0d19
c25t0d19
c18t0d22
c19t0d22
c20t0d22
c21t0d22
c22t0d22
c23t0d22
c24t0d22
c25t0d22
c18t0d25
c19t0d25
END
VOLUME_GROUP: asu1-6 (d5)
STRIPE 8m
VOLUME 1
c21t0d5
c22t0d5
c23t0d5
c24t0d5
c25t0d5
c18t0d8
c19t0d8
c20t0d8
c21t0d8
c22t0d8
c23t0d8
c24t0d8
```

```
c25t0d8
c18t0d11
c19t0d11
c20t0d11
c21t0d11
c22t0d11
c23t0d11
c24t0d11
c25t0d11
c18t0d14
c19t0d14
c20t0d14
c21t0d14
c22t0d14
c23t0d14
c24t0d14
c25t0d14
c18t0d17
c19t0d17
c20t0d17
c21t0d17
c22t0d17
c23t0d17
c24t0d17
c25t0d17
c18t0d20
c19t0d20
c20t0d20
c21t0d20
c22t0d20
c23t0d20
c24t0d20
c25t0d20
c18t0d23
c19t0d23
c20t0d23
c21t0d23
c22t0d23
c23t0d23
c24t0d23
c25t0d23
c18t0d26
c19t0d26
END
VOLUME_GROUP: asu2-1 (d6)
STRIPE 8m
VOLUME 1
c3t0d24
c4t0d24
c5t0d24
c6t0d24
c7t0d24
c8t0d24
c1t0d27
c2t0d27
c3t0d27
c4t0d27
c5t0d27
c6t0d27
c7t0d27
c8t0d27
c1t0d30
c2t0d30
```

```
  c3t0d30
  c4t0d30
  c5t0d30
  c6t0d30
  c7t0d30
  c8t0d30
  c1t0d33
  c2t0d33
  c3t0d33
  c4t0d33
  c5t0d33
  c6t0d33
  c8t0d33
  END
  VOLUME_GROUP: asu2-2 (d7)
  STRIPE 8m
  VOLUME 1
  c3t0d25
  c4t0d25
  c5t0d25
  c6t0d25
  c7t0d25
  c8t0d25
  c1t0d28
  c2t0d28
  c3t0d28
  c4t0d28
  c5t0d28
  c6t0d28
  c7t0d28
  c8t0d28
  c1t0d31
  c2t0d31
  c3t0d31
  c4t0d31
  c5t0d31
  c6t0d31
  c7t0d31
  c8t0d31
  c1t0d34
  c2t0d34
  c3t0d34
  c4t0d34
  c5t0d34
  c6t0d34
  c8t0d34
  END
  VOLUME_GROUP: asu2-3 (d8)
  STRIPE 8m
  VOLUME 1
  c3t0d26
  c4t0d26
  c5t0d26
  c6t0d26
  c7t0d26
  c8t0d26
  c1t0d29
  c2t0d29
  c3t0d29
  c4t0d29
  c5t0d29
  c6t0d29
  c7t0d29
```

```
  c8t0d29
  c1t0d32
  c2t0d32
  c3t0d32
  c4t0d32
  c5t0d32
  c6t0d32
  c7t0d32
  c8t0d32
  c1t0d35
  c2t0d35
  c3t0d35
  c4t0d35
  c5t0d35
  c6t0d35
  c8t0d35
  END
  VOLUME_GROUP: asu2-4 (d9)
  STRIPE 8m
  VOLUME 1
  c20t0d24
  c21t0d24
  c22t0d24
  c23t0d24
  c24t0d24
  c25t0d24
  c18t0d27
  c19t0d27
  c20t0d27
  c21t0d27
  c22t0d27
  c23t0d27
  c24t0d27
  c25t0d27
  c18t0d30
  c19t0d30
  c20t0d30
  c21t0d30
  c22t0d30
  c23t0d30
  c24t0d30
  c25t0d30
  c18t0d33
  c19t0d33
  c20t0d33
  c21t0d33
  c22t0d33
  c23t0d33
  c25t0d33
  END
  VOLUME_GROUP: asu2-5 (d10)
  STRIPE 8m
  VOLUME 1
  c20t0d25
  c21t0d25
  c22t0d25
  c23t0d25
  c24t0d25
  c25t0d25
  c18t0d28
  c19t0d28
  c20t0d28
  c21t0d28
```

```
  c22t0d28
  c23t0d28
  c24t0d28
  c25t0d28
  c18t0d31
  c19t0d31
  c20t0d31
  c21t0d31
  c22t0d31
  c23t0d31
  c24t0d31
  c25t0d31
  c18t0d34
  c19t0d34
  c20t0d34
  c21t0d34
  c22t0d34
  c23t0d34
  c25t0d34
  END
  VOLUME_GROUP: asu2-6 (d11)
  STRIPE 8m
  VOLUME 1
  c20t0d26
  c21t0d26
  c22t0d26
  c23t0d26
  c24t0d26
  c25t0d26
  c18t0d29
  c19t0d29
  c20t0d29
  c21t0d29
  c22t0d29
  c23t0d29
  c24t0d29
  c25t0d29
  c18t0d32
  c19t0d32
  c20t0d32
  c21t0d32
  c22t0d32
  c23t0d32
  c24t0d32
  c25t0d32
  c18t0d35
  c19t0d35
  c20t0d35
  c21t0d35
  c22t0d35
  c23t0d35
  c25t0d35
  END
  VOLUME_GROUP: asu3-1 (d12)
  STRIPE 8m
  VOLUME 1
  c1t0d0
  c2t0d0
  c3t0d0
  c4t0d0
  c5t0d0
  c6t0d0
  c7t0d0
```

```
c8t0d0
c1t0d3
c2t0d3
c3t0d3
END
VOLUME_GROUP: asu3-2 (d13)
STRIPE 8m
VOLUME 1
c1t0d1
c2t0d1
c3t0d1
c4t0d1
c5t0d1
c6t0d1
c7t0d1
c8t0d1
c1t0d4
c2t0d4
c3t0d4
END
VOLUME_GROUP: asu3-3 (d14)
STRIPE 8m
VOLUME 1
c1t0d2
c2t0d2
c3t0d2
c4t0d2
c5t0d2
c6t0d2
c7t0d2
c8t0d2
c1t0d5
c2t0d5
c3t0d5
END
VOLUME_GROUP: asu3-4 (d15)
STRIPE 8m
VOLUME 1
c18t0d0
c19t0d0
c20t0d0
c21t0d0
c22t0d0
c23t0d0
c24t0d0
c25t0d0
c18t0d3
c19t0d3
c20t0d3
END
VOLUME_GROUP: asu3-5 (d16)
STRIPE 8m
VOLUME 1
c18t0d1
c19t0d1
c20t0d1
c21t0d1
c22t0d1
c23t0d1
c24t0d1
c25t0d1
c18t0d4
c19t0d4
```

```
c20t0d4
END
VOLUME_GROUP: asu3-6 (d17)
STRIPE 8m
VOLUME 1
c18t0d2
c19t0d2
c20t0d2
c21t0d2
c22t0d2
c23t0d2
c24t0d2
c25t0d2
c18t0d5
c19t0d5
c20t0d5
END
```

DX440S2_R10_95rg_570lun_16ch_120604.exp

```
#!/usr/local/bin/expect
# script to setup initial configuration for dx440s2
# for SPC-2 benchmark
set timeout 600
set user root
set password root
spawn ssh dx440s2 -l $user
expect "CLI>"
# procedure to execute dx440s2 cli command
proc docli { cmd args} {
send "$cmd $args\r"
expect "CLI>"
}
# procedure to exit
proc doexit {} {
send "exit \r"
}

#D440S2 RAID10(4+4)x94RG,570lun SPC1-BMT, 16 channels 120602

# Remove the existing mapping
docli release mapping -port all

#  Create Port Mapping for CM0 CA0 Ports 0 & 3
docli set mapping -port 000 -volume-number 0-2,48-50,96-98,144-146,192-194,240-242 -
lun 0-17
docli set mapping -port 000 -volume-number 288-290,336-338,384,386,432-434,480-
482,528-530 -lun 18-35
docli set mapping -port 001 -volume-number 6-8,54-56,102-104,150-152,198-200,246-248
-lun 0-17
docli set mapping -port 001 -volume-number 294-296,342-344,390-392,438-440,486-
488,534-536 -lun 18-35
docli set mapping -port 002 -volume-number 3-5,51-53,99-101,147-149,159-197,243-245
-lun 0-17
docli set mapping -port 002 -volume-number 291-293,339-341,387-389,435-437,483-
485,531-533 -lun 18-35
docli set mapping -port 003 -volume-number 9-11,57-59,105-107,153-155,201-203,249-
251 -lun 0-17
docli set mapping -port 003 -volume-number 297-299,345-347,393-395,441-443,489-
491,537-539 -lun 18-35

#  Create Port Mapping for CM0 CA1 Ports 0 & 3
```

```
docli set mapping -port 010 -volume-number 24-26,72-74,120-122,168-170,216-218,264-
266 -lun 0-17
docli set mapping -port 010 -volume-number 312-314,360-302,408-410,456-458,504-
508,552-554 -lun 18-35
docli set mapping -port 011 -volume-number 30-32,78-80,126-128,174-176,222-224,270-
272 -lun 0-17
docli set mapping -port 011 -volume-number 318-320,366-368,414-416,462-464,510-
512,558-560 -lun 18-35
docli set mapping -port 012 -volume-number 27-29,75-77,123-125,171-173,219-221,267-
269 -lun 0-17
docli set mapping -port 012 -volume-number 315-317,363-365,411-413,459-431,507-
509,555-557 -lun 18-35
docli set mapping -port 013 -volume-number 33-35,81-83,129-131,177-179,225-227,273-
275 -lun 0-17
docli set mapping -port 013 -volume-number 321-323,369-371,417-419,465-467,513-
515,561-563 -lun 18-35

#  Create Port Mapping for CM1 CA0 Ports 0 & 3
docli set mapping -port 100 -volume-number 12-14,60-62,108-110,156-158,204-206,252-
254 -lun 0-17
docli set mapping -port 100 -volume-number 300-302,348-350,396-398,444-446,492-
494,540-542 -lun 18-35
docli set mapping -port 101 -volume-number 18-20,66-68,114-116,162-164,210-212,258-
260 -lun 0-17
docli set mapping -port 101 -volume-number 306-308,354-356,402-404,450-452,498-
500,546-548 -lun 18-35
docli set mapping -port 102 -volume-number 15-17,63-65,111-113,159-161,207-209,255-
257 -lun 0-17
docli set mapping -port 102 -volume-number 303-305,351-353,399-401,447-449,495-
497,543-545 -lun 18-35
docli set mapping -port 103 -volume-number 21-23,69-71,117-119,165-167,213-215,261-
263 -lun 0-17
docli set mapping -port 103 -volume-number 309-311,357-359,405-407,453-455,501-
503,549-551 -lun 18-35

#  Create Port Mapping for CM1 CA1 Ports 0 & 3
docli set mapping -port 110 -volume-number 36-38,84-86,132-134,180-182,228-230,276-
278 -lun 0-17
docli set mapping -port 110 -volume-number 324-326,372-374,420-422,468-470,516-518 -
lun 18-32
docli set mapping -port 111 -volume-number 42-44,90-92,138-140,186-188,234-236,282-
284 -lun 0-17
docli set mapping -port 111 -volume-number 330-332,378-380,426-428,474-476,522-
524,564-566 -lun 18-35
docli set mapping -port 112 -volume-number 39-41,87-89,135-137,183-185,231-233,279-
281 -lun 0-17
docli set mapping -port 112 -volume-number 327-329,375-377,423-425,471-473,519-521 -
lun 18-32
docli set mapping -port 113 -volume-number 45-47,93-95,141-143,189-191,237-239,285-
287 -lun 0-17
docli set mapping -port 113 -volume-number 333-335,381-383,429-431,477-479,525-
527,567-569 -lun 18-35

#exit from CLI

doexit
```

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

Primary Metrics and Repeatability Tests

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

Common Command Lines – Primary Metrics and Repeatability Tests

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

```
host=master

slaves=(slave01,slave02,slave03,slave04,slave05,slave06,slave07,slave08,slave09,slav
e10,slave11,slave12,slave13,slave14,slave15,slave16,slave17,slave18,slave19,slave20,
slave21)

sd=asu1_1,lun=/dev/md/rdsk/d0,size=5267g
sd=asu1_2,lun=/dev/md/rdsk/d1,size=5267g
sd=asu1_3,lun=/dev/md/rdsk/d2,size=5267g
sd=asu1_4,lun=/dev/md/rdsk/d3,size=5267g
sd=asu1_5,lun=/dev/md/rdsk/d4,size=5267g
sd=asu1_6,lun=/dev/md/rdsk/d5,size=5267g
sd=asu2_1,lun=/dev/md/rdsk/d6,size=5267g
sd=asu2_2,lun=/dev/md/rdsk/d7,size=5267g
sd=asu2_3,lun=/dev/md/rdsk/d8,size=5267g
sd=asu2_4,lun=/dev/md/rdsk/d9,size=5267g
sd=asu2_5,lun=/dev/md/rdsk/d10,size=5267g
sd=asu2_6,lun=/dev/md/rdsk/d11,size=5267g
sd=asu3_1,lun=/dev/md/rdsk/d12,size=1170g
sd=asu3_2,lun=/dev/md/rdsk/d13,size=1170g
sd=asu3_3,lun=/dev/md/rdsk/d14,size=1170g
sd=asu3_4,lun=/dev/md/rdsk/d15,size=1170g
sd=asu3_5,lun=/dev/md/rdsk/d16,size=1170g
sd=asu3_6,lun=/dev/md/rdsk/d17,size=1170g
```

Primary Metrics Test: Sustainability Test Phase/Test Run

```
rd=sustain,bsus=2060,startup=180,elapsed=28800,interval=60
```

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

```
rd=ramp_100,bsus=2060,startup=180,elapsed=600,interval=60
```

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

```
rd=ramp_95,bsus=1957,startup=180,elapsed=600,interval=60
```

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

```
rd=ramp_90,bsus=1854,startup=180,elapsed=600,interval=60
```

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

```
rd=ramp_80,bsus=1648,startup=180,elapsed=600,interval=60
```

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

```
rd=ramp_50,bsus=1030,startup=180,elapsed=600,interval=60
```

Primary Metrics Test: Response Time Ramp Test Phase *(10% Test Run)*

```
rd=ramp_10,bsus=206,startup=180,elapsed=600,interval=60
```

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

```
rd=repeat1_lrt,bsus=206,startup=180,elapsed=600,interval=60
```

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

```
rd=repeat1_iops,bsus=2060,startup=180,elapsed=600,interval=60
```

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

```
rd=repeat2_lrt,bsus=206,startup=180,elapsed=600,interval=60
```

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

```
rd=repeat2_iops,bsus=2060,startup=180,elapsed=600,interval=60
```

Persistence Test

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

```
javaparms="-d64 -Xmx8192m -Xms8192m -Xss512k"
sd=asu1_1,lun=/dev/md/rdsk/d0,size=5267g
sd=asu1_2,lun=/dev/md/rdsk/d1,size=5267g
sd=asu1_3,lun=/dev/md/rdsk/d2,size=5267g
sd=asu1_4,lun=/dev/md/rdsk/d3,size=5267g
sd=asu1_5,lun=/dev/md/rdsk/d4,size=5267g
sd=asu1_6,lun=/dev/md/rdsk/d5,size=5267g
sd=asu2_1,lun=/dev/md/rdsk/d6,size=5267g
sd=asu2_2,lun=/dev/md/rdsk/d7,size=5267g
sd=asu2_3,lun=/dev/md/rdsk/d8,size=5267g
sd=asu2_4,lun=/dev/md/rdsk/d9,size=5267g
sd=asu2_5,lun=/dev/md/rdsk/d10,size=5267g
sd=asu2_6,lun=/dev/md/rdsk/d11,size=5267g
sd=asu3_1,lun=/dev/md/rdsk/d12,size=1170g
sd=asu3_2,lun=/dev/md/rdsk/d13,size=1170g
sd=asu3_3,lun=/dev/md/rdsk/d14,size=1170g
sd=asu3_4,lun=/dev/md/rdsk/d15,size=1170g
sd=asu3_5,lun=/dev/md/rdsk/d16,size=1170g
sd=asu3_6,lun=/dev/md/rdsk/d17,size=1170g
```

APPENDIX E: SPC-1 WORKLOAD GENERATOR INPUT PARAMETERS

'Master' Execution Script for Primary Metrics Test, Repeatability Test, and Persistence Test Run 1

The following script, invoked from a standard command shell window on the Host System, was used to execute the required ASU pre-fill, 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. The script also included the appropriate commands to capture the detailed TSC profile listings required for a Remote Audit.

The Slave JVMs were started prior to each Test Run (**start_slaves.sh**) and terminated at the completion of each Test Run (**stop_slaves.sh**) to address a Slave JVM execution issue.

```
#
# Script consists of Part 1 of the FDR job
#  - Save "Before" logs
#  - Prefill
#  - Metricss
#  - Repeatability1/2
#  - Persistence1
#  - Power Down
# This script incorporates lowlevel scripts by Walter Baker.
#
# Absolute path of Work Directories
ROOT=/SPC1_work
# directory for prefill step
PREFILL=${ROOT}/prefill
# main work directory for SPC1 benchmark
SPC1=${ROOT}/spc1
SCRIPTS=${ROOT}/scripts
SPCTMP=/tmp/spc1
# Lowlevel files/scripts from Walter are placed under subdirectory
LOWLEVEL=${SCRIPTS}/lowlevel

#
# helper functions
#

#
# assertSuccess($bmt,$jobID, $confID )
#
# $bmt = benchmark type ( metrics, repeatablity1, repeatablity2, persistence1,
persistence2)
# $jobID = jobID
# $confID = configuration ID
# Assert that previous run was successful, by presense of Vdbench .. sucessfully
line
# in summary.html file.  If the file does not exist or the line does not exist
# stop the job zip up the data and quit
#
assertSuccess() {
  bmt=$1
  jobID=$2
  confID=$3
  # check to see if summary.html contains the VdBench
  if [ -f ${SPC1}/${jobID}/${bmt}/summary.html ]; then
```

```
      grep Vdbench ${SPC1}/${jobID}/${bmt}/summary.html >
${SPCTMP}/${JOBID}${bmt}success.txt
      if [ -s  ${SPCTMP}/${JOBID}success.txt ]; then
        if [  `awk '{print $5}' ${SPCTMP}/${JOBID}${bmt}success.txt ` ==
"Successfully" ]; then
           return
        fi
      fi
  fi
  # At this point the gather the data and exit from the Job
  # Save after log
  #
  ${SCRIPTS}/sendstatus.sh "${bmt} Unsuccessful Job=${jobID}" ${jobID}_message.txt
  ${SCRIPTS}/sendstatus.sh "Started save of afterA log for Job=${jobID}"
${jobID}_message.txt
  ./exportLog.exp ${jobID}_afterA
  cp /tmp/spc1/*${jobID}_afterA  /SPC1_work/scripts/
  ${SCRIPTS}/sendstatus.sh "Completed save of afterA log for Job=${jobID}"
${jobID}_message.txt
  # Collect Archive
  ./zipUp.sh ${confID} ${jobID}
  # Send Archive to repository
  ./ftpToEternity.exp SPC1_Benchmark DX440S2 ${date} ${jobID}.zip
  ${SCRIPTS}/sendstatus.sh "Aborted FDR. Partial in Eternity
~share3/SPC1Benchmark/DX440S2/${date} jobID=${jobID}" ${jobID}_message.txt
  exit
}

# create tmp directory for spc1 if it does not exist
if [ ! -d /tmp/spc1 ]; then
mkdir /tmp/spc1
fi
#
# Do prefill/persisetnce 1 required for the FDR run
#
# confID uniquely identifies the configuration of the array
confID=DX440S2_R10_95rg_570lun_16ch_120604
# obtain jobID based on the timestamp
# jobID uniquely identifies the benchmark run
jobID=J`date +%y%m%d%H%M%S`
# save confID and jobID for Part 2 use
echo $confID > /tmp/spc1/lastconfID
echo $jobID > /tmp/spc1/lastjobID
# bsu=each business scale unit corresponds to 50 iops
bsu=2060
#
echo FDR job w/Lowlevel   start time `date` > /tmp/spc1/${jobID}_message.txt
echo job confID=$confID >> /tmp/spc1/${jobID}_message.txt
echo job jobID=$jobID >> /tmp/spc1/${jobID}_message.txt
${SCRIPTS}/sendstatus.sh "Starting FDR Job=${jobID}" ${jobID}_message.txt
# datecode for directory
date=`date +%Y%m%d`
#
# Save beforeP log
#
${SCRIPTS}/sendstatus.sh "Starting BeforeP log save Job=${jobID}"
${jobID}_message.txt
${SCRIPTS}/exportLog.exp ${jobID}_beforeP
cp /tmp/${jobID}_before* /SPC1_work/
${SCRIPTS}/sendstatus.sh "Completed BeforeP log save Job=${jobID}"
${jobID}_message.txt
#
# Perform Prefill operation
```

```
#
${SCRIPTS}/sendstatus.sh "Starting Prefill step for Job=${jobID}"
${jobID}_message.txt
date
# move to prefill context
cd ${PREFILL}
sh -x ./doPrefill.sh ${confID} ${jobID}
${SCRIPTS}/sendstatus.sh "Completed Prefill step for Job=${jobID}"
${jobID}_message.txt
#
# Save beforeF log
#
${SCRIPTS}/sendstatus.sh "Starting BeforeF log save Job=${jobID}"
${jobID}_message.txt
${SCRIPTS}/exportLog.exp ${jobID}_beforeF
cp /tmp/${jobID}_before* /SPC1_work/
${SCRIPTS}/sendstatus.sh "Completed BeforeF log save Job=${jobID}"
${jobID}_message.txt
#
#
# Start sequence of SPC1 benchmark jobs
#
# Setup the Java environment variables for SPC1
CLASSPATH=/usr/local/spc/spc1;export CLASSPATH
LD_LIBRARY_PATH=/usr/local/spc/spc1;export LD_LIBRARY_PATH
# setup the unique execution context(directory) for SPC1 job
cd ${SPC1}
mkdir ${jobID}
cd ${jobID}
# Copy the Low level parameter files from Walter for Metrics and Repeatabiliy1/2
steps
cp ${LOWLEVEL}/sustain.txt .
cp ${LOWLEVEL}/ramp*.txt .
cp ${LOWLEVEL}/repeat*_*.txt .
# Copy the standard  parameter files for Persistence1/2 steps
# copy the spc1.cfg file associated with the configID with S (single sever)
cp ../spc1_${confID}_S.cfg spc1.cfg
#
${SCRIPTS}/sendstatus.sh "Starting Metrics test step for Job=${jobID}"
${jobID}_message.txt
#metrics
${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  sustain
${LOWLEVEL}/start_slaves.sh sustain
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1   -f sustain.txt -o sustain
SPCOut
${LOWLEVEL}/stop_slaves.sh

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  ramp100
${LOWLEVEL}/start_slaves.sh ramp100
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f ramp100.txt -o ramp100
SPCOut
${LOWLEVEL}/stop_slaves.sh

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  ramp095
${LOWLEVEL}/start_slaves.sh ramp095
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f ramp095.txt -o ramp095
SPCOut
${LOWLEVEL}/stop_slaves.sh

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  ramp090
${LOWLEVEL}/start_slaves.sh ramp090
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f ramp090.txt -o ramp090
SPCOut
```

```
${LOWLEVEL}/stop_slaves.sh

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  ramp080
${LOWLEVEL}/start_slaves.sh ramp080
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f ramp080.txt -o ramp080
SPCOut
${LOWLEVEL}/stop_slaves.sh

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  ramp050
${LOWLEVEL}/start_slaves.sh ramp050
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f ramp050.txt -o ramp050
SPCOut
${LOWLEVEL}/stop_slaves.sh

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  ramp010
${LOWLEVEL}/start_slaves.sh ramp010
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f ramp010.txt -o ramp010
SPCOut
${LOWLEVEL}/stop_slaves.sh

${SCRIPTS}/sendstatus.sh "Completed Metrics test step for Job=${jobID}"
${jobID}_message.txt

#repeat-1
${SCRIPTS}/sendstatus.sh "Starting Repeatablity test 1 step for Job=${jobID}"
${jobID}_message.txt

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  repeat1_lrt
${LOWLEVEL}/start_slaves.sh repeat1_lrt
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f repeat1_lrt.txt -o
repeat1_lrt SPCOut
${LOWLEVEL}/stop_slaves.sh

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  repeat1_iops
${LOWLEVEL}/start_slaves.sh repeat1_iops
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f repeat1_iops.txt -o
repeat1_iops SPCOut
${LOWLEVEL}/stop_slaves.sh

${SCRIPTS}/sendstatus.sh "Completed Repeatablity test 1 step for Job=${jobID}"
${jobID}_message.txt

#repeat-2
${SCRIPTS}/sendstatus.sh "Starting Repeatablity test 2 step for Job=${jobID}"
${jobID}_message.txt

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}  repeat2_lrt
${LOWLEVEL}/start_slaves.sh repeat2_lrt
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f repeat2_lrt.txt -o
repeat2_lrt SPCOut
${LOWLEVEL}/stop_slaves.sh

${LOWLEVEL}/setup_slave_dirs.sh ${jobID}   repeat2_iops
${LOWLEVEL}/start_slaves.sh repeat2_iops
java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -w SPC1  -f repeat2_iops.txt -o
repeat2_iops SPCOut
${LOWLEVEL}/stop_slaves.sh

${SCRIPTS}/sendstatus.sh "Completed Repeatablity test 2 step for Job=${jobID}"
${jobID}_message.txt

#persist-1
```

```
${SCRIPTS}/sendstatus.sh "Starting Persistence test 1 step for Job=${jobID}"
${jobID}_message.txt
java -d64 -Xmx4096m -Xms4096m -Xss512k persist1 -b $bsu
${SCRIPTS}/sendstatus.sh "Completed Persistence test 1 step for Job=${jobID}"
${jobID}_message.txt
# move back to Script context
cd ${SCRIPTS}
# Perform Power cycle
${SCRIPTS}/sendstatus.sh "Array is Ready for manual Power cycle Job=${jobID} "
${jobID}_message.txt
# save nohup.out part1
sync
cp ./nohup.out nohup1_${jobID}.out
```

Referenced Scripts

The following scripts were invoked from the 'master' execution script.

setup_slave_dirs.sh

This script creates an output directory for each Slave JVM for each SPC-1 Test Run.

```
#!/usr/local/bin/bash
# parameter jobID
jobID=$1
STEP=$2
ROOT=/SPC1_work
SCRIPTS=${ROOT}/scripts
LOWLEVEL=$SCRIPTS/lowlevel
SPCDIR=${ROOT}/spc1/${jobID}
cd ${SPCDIR}
mkdir ${STEP}
cd ${STEP}
mkdir slave01;cp ${LOWLEVEL}/slave01.cfg slave01/
mkdir slave02;cat slave01/slave01.cfg |sed s/slave01/slave02/ > slave02/slave02.cfg
mkdir slave03;cat slave01/slave01.cfg |sed s/slave01/slave03/ > slave03/slave03.cfg
mkdir slave04;cat slave01/slave01.cfg |sed s/slave01/slave04/ > slave04/slave04.cfg
mkdir slave05;cat slave01/slave01.cfg |sed s/slave01/slave05/ > slave05/slave05.cfg
mkdir slave06;cat slave01/slave01.cfg |sed s/slave01/slave06/ > slave06/slave06.cfg
mkdir slave07;cat slave01/slave01.cfg |sed s/slave01/slave07/ > slave07/slave07.cfg
mkdir slave08;cat slave01/slave01.cfg |sed s/slave01/slave08/ > slave08/slave08.cfg
mkdir slave09;cat slave01/slave01.cfg |sed s/slave01/slave09/ > slave09/slave09.cfg
mkdir slave10;cat slave01/slave01.cfg |sed s/slave01/slave10/ > slave10/slave10.cfg
mkdir slave11;cat slave01/slave01.cfg |sed s/slave01/slave11/ > slave11/slave11.cfg
mkdir slave12;cat slave01/slave01.cfg |sed s/slave01/slave12/ > slave12/slave12.cfg
mkdir slave13;cat slave01/slave01.cfg |sed s/slave01/slave13/ > slave13/slave13.cfg
mkdir slave14;cat slave01/slave01.cfg |sed s/slave01/slave14/ > slave14/slave14.cfg
mkdir slave15;cat slave01/slave01.cfg |sed s/slave01/slave15/ > slave15/slave15.cfg
mkdir slave16;cat slave01/slave01.cfg |sed s/slave01/slave16/ > slave16/slave16.cfg
mkdir slave17;cat slave01/slave01.cfg |sed s/slave01/slave17/ > slave17/slave17.cfg
mkdir slave18;cat slave01/slave01.cfg |sed s/slave01/slave18/ > slave18/slave18.cfg
mkdir slave19;cat slave01/slave01.cfg |sed s/slave01/slave19/ > slave19/slave19.cfg
mkdir slave20;cat slave01/slave01.cfg |sed s/slave01/slave20/ > slave20/slave20.cfg
mkdir slave21;cat slave01/slave01.cfg |sed s/slave01/slave21/ > slave21/slave21.cfg
```

start_slaves.sh

This script starts all of the specified Slave JVMs.

```
#!/usr/local/bin/bash
CLASSPATH=/usr/local/spc/spc1;export CLASSPATH
LD_LIBRARY_PATH=/usr/local/spc/spc1;export LD_LIBRARY_PATH
STEP=$1
cd $STEP
(cd slave01;java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave01.cfg &>
slave01_console_out.txt) &
sleep 1
(cd slave02;java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave02.cfg &>
slave02_console_out.txt) &
sleep 1
(cd slave03; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave03.cfg &>
slave03_console_out.txt) &
sleep 1
(cd slave04; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave04.cfg &>
slave04_console_out.txt) &
sleep 1
(cd slave05; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave05.cfg &>
slave05_console_out.txt )&
sleep 1
(cd slave06; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave06.cfg &>
slave06_console_out.txt )&
sleep 1
(cd slave07; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave07.cfg &>
slave07_console_out.txt )&
sleep 1
(cd slave08; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave08.cfg &>
slave08_console_out.txt )&
sleep 1
(cd slave09; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave09.cfg &>
slave09_console_out.txt )&
sleep 1
(cd slave10; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave10.cfg &>
slave10_console_out.txt )&
sleep 1
(cd slave11; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave11.cfg &>
slave11_console_out.txt )&
sleep 1
(cd slave12; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave12.cfg &>
slave12_console_out.txt )&
sleep 1
(cd slave13; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave13.cfg &>
slave13_console_out.txt )&
sleep 1
(cd slave14; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave14.cfg &>
slave14_console_out.txt )&
sleep 1
(cd slave15; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave15.cfg &>
slave15_console_out.txt )&
sleep 1
(cd slave16; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave16.cfg &>
slave16_console_out.txt )&
sleep 1
(cd slave17; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave17.cfg &>
slave17_console_out.txt )&
sleep 1
```

```
(cd slave18; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave18.cfg &>
slave18_console_out.txt )&
sleep 1
(cd slave19; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave19.cfg &>
slave19_console_out.txt )&
sleep 1
(cd slave20; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave20.cfg &>
slave20_console_out.txt )&
sleep 1
(cd slave21; java -d64 -Xmx4096m -Xms4096m -Xss512k spc1 -f slave21.cfg &>
slave21_console_out.txt )&
```

stop_slaves.sh

```
This script terminates all of the specified Slave JVMs.
#!/usr/local/bin/bash
# kill  slaves if any (any one running spc local version of java)
for j in `ps -ef |grep "[0-9]:[0-9][0-9] /usr/local/spc/java" |awk '{print $2}' -`
do
   kill $j
done
```

Persistence Test Run 2

The following script, invoked from a standard command shell window on a single Host System, executes Persistence Test Run 2

```
#!/usr/bin/bash
#
# Do All steps required for the FDR run
#
#
# obtain jobID and confID saved from part 1
#
# check to see if previous context exists
# create tmp directory for spc1 if it does not exist
if [ ! -d /tmp/spc1 ]; then
 echo Error!
 exit
else
 confID=`cat /tmp/spc1/lastconfID`
 jobID=`cat /tmp/spc1/lastjobID`
fi
# Absolute path of Work Directories
ROOT=/SPC1_work
# directory for prefill step
PREFILL=${ROOT}/prefill
# main work directory for SPC1 benchmark
SPC1=${ROOT}/spc1
SCRIPTS=${ROOT}/scripts
SPCTMP=/tmp/spc1
# Lowlevel files/scripts from Walter are placed under subdirectory
LOWLEVEL=${SCRIPTS}/lowlevel
# bsu=each business scale unit corresponds to 50 iops
bsu=2060
# get current date for the directory
date=`date +%Y%m%d`
#
# Run persistence 2
```

```
#
# move to the Benchmark execution context
cd ${SPC1}
cd ${jobID}
CLASSPATH=/usr/local/spc/spc1;export CLASSPATH
LD_LIBRARY_PATH=/usr/local/spc/spc1;export LD_LIBRARY_PATH
${SCRIPTS}/sendstatus.sh "Starting Persistence2 step for Job=${jobID}"
${jobID}_message.txt
#persist-2
java -d64 -Xmx8192m -Xms8192m -Xss512k persist2
${SCRIPTS}/sendstatus.sh "Completed Persistence2 step for Job=${jobID}"
${jobID}_message.txt
# move out of the  Benchmark execution contect
cd ..
# move back to scripts context
cd ../scripts/
#
# Save after log
#
${SCRIPTS}/sendstatus.sh "Started save of AFTER log for Job=${jobID}"
${jobID}_message.txt
${SCRIPTS}/exportLog.exp ${jobID}_after
cp /tmp/spc1/*${jobID}_after*  /SPC1_work/scripts/
${SCRIPTS}/sendstatus.sh "Completed save of AFTER log for Job=${jobID}"
${jobID}_message.txt
# Collect Archive
${SCRIPTS}/zipUp.sh ${confID} ${jobID}
# Send Archive to repository
${SCRIPTS}/ftpToEternity.exp SPC1_Benchmark DX440S2 ${date} ${jobID}.zip
${SCRIPTS}/sendstatus.sh "Completed FDR. All data in Eternity
~share3/SPC1Benchmark/DX440S2/${date} jobID=${jobID}" ${jobID}_message.txt
```