SPC BENCHMARK 1™
FULL DISCLOSURE REPORT

DATACORE SOFTWARE CORPORATION
DATACORE SANSYMPHONY 10.0
(DUAL NODE, HIGH AVAILABILITY, HYPER-CONVERGED)

SPC-1 V1.14

Submitted for Review: June 13, 2016
Submission Identifier: A00178

First Edition – June 2016

Trademarks

Table of Contents

AUDIT CERTIFICATION

Ben Treiber
DataCore Software Corporation
Worldwide Headquarters
Corporate Park
6300 NW 5th Way
Fort Lauderdale, FL 33309

June 13, 2016

The SPC Benchmark 1™ Reported Data listed below for the DataCore SANsymphony 10.0 *(Dual Node, High Availability, Hyper-converged)*was produced in compliance with the SPC Benchmark 1™ v1.14 Onsite Audit requirements.

SPC Benchmark 1™ v1.14 Reported Data	
Tested Storage Product (TSP) Name: DataCore SANsymphony 10.0*(Dual Node, High Availability, Hyper-converged)*	
Metric	**Reported Result**
SPC-1 IOPS™	1,201,961.83
SPC-1 Price-Performance	$0.10/SPC-1 IOPS™
Total ASU Capacity	3,325.000 GB
Data Protection Level	Protected 2 *(mirroring)*
Total Price (including three-year maintenance)	$115,142.76
Currency Used	U.S. Dollars
Target Country for availability, sales and support	USA

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

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

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

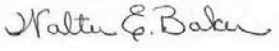
AUDIT CERTIFICATION *(CONT.)*

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

Audit Notes:

There are no audit notes.

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

DataCore™
SOFTWARE

Date: May 26th, 2016

From: Roni Putra
DataCore Software Corporation
Worldwide Headquarters
Corporate Park
6300 NW 5th Way
Ft. Lauderdale, FL 33309
Phone: (954)-377-6000
FAX: (954) 938-7953

To: Walter E. Baker
Storage Performance Council (SPC) Auditor
Gradient Systems, Inc.
643 Bair Island Road, Suite 103
Redwood City, CA 94063-2755
Phone: 650.556.9380
FAX: 650.556.9385

Subject: SPC-1 Letter of Good Faith for DataCore SANsymphony 10.0 – Dual Node, High Availability

DataCore Software Corporation is the SPC-1 Test Sponsor for the above listed product. To the best of our knowledge and belief, the required SPC-1 benchmark results and materials we have submitted for that product are complete, accurate, and in full compliance with V1.14 of the SPC-1 benchmark specification.

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

Signed: Date:

May 26th , 2016

Roni Putra, Vice President and CTO
DataCore Software Corporation

Date of Signature

DataCore Software Corporation
6300 NW 5th Way
Fort Lauderdale, FL 33309
T 954.377.6000 **F** 954.938.7953
info@datacore.com

www.datacore.com

EXECUTIVE SUMMARY

Test Sponsor and Contact Information

Test Sponsor and Contact Information	
Test Sponsor Primary Contact	DataCore Software Corporation – http://www.datacore.com Ben Treiber – ben.treiber@datacore.com Worldwide Headquarters Corporate Park 6300 NW 5th Way Ft. Lauderdale, FL 33309 Phone: (954) 377-6000 FAX: (954) 938-7953
Test Sponsor Alternate Contact	DataCore Software Corporation – http://www.datacore.com Roni Putra – roni.putra@datacore.com Worldwide Headquarters Corporate Park 6300 NW 5th Way Ft. Lauderdale, FL 33309 Phone: (954) 377-6000 FAX: (954) 938-7953
Auditor	Storage Performance Council – http://www.storageperformance.org Walter E. Baker – AuditService@StoragePerformance.org Gradient Systems, Inc. 643 Bair Island Road, Suite 103 Redwood City, CA 94063 Phone: (650) 556-9380 FAX: (650) 556-9385

Revision Information and Key Dates

Revision Information and Key Dates	
SPC-1 Specification revision number	V1.14
SPC-1 Workload Generator revision number	V2.3.0
Date Results were first used publicly	June 13, 2016
Date the FDR was submitted to the SPC	June 13, 2016
Date the Priced Storage Configuration is available for shipment to customers	September 6, 2016
Date the TSC completed audit certification	June 13, 2016

Tested Storage Product (TSP) Description

SANsymphony provides a flexible software platform that has been proven in enterprise environments for over a decade. Because it is designed from the outset as parallel storage software, SANsymphony is uniquely able to scale to its underlying hardware environment and to do so in both conventional storage topologies and in more recent converged environments.

The Tested Storage Configuration (TSC) used to produce this SPC-1 Result was a converged configuration consisting of two Lenovo x3650 servers acting both as a dual-node, high availability, hyperconverged SANsymphony configuration and as the SPC-1 Host Systems providing the reported SPC-1 I/O requests.

This version of SANsymphony improved on the performance demonstrated by the current SANsymphony-V 10.0 SPC-1 Result[1]. In addition, the dual-node SANsymphony configuration provided an additional level of protection by synchronously mirroring the SPC-1 Data Repository across directly connected, redundant Fibre Channel (FC) ports between the two nodes, which resulted in active-active storage for the SPC-1 Test Runs executing on the TSC.

[1] The SPC-1 Executive Summary and Full Disclosure Report for the Data Core SANsymphony-V 10.0 (459,290.87 SPC-1 IOPS) are available at: http://www.storageperformance.org/results/benchmark_results_spc1_active/#a00164

Summary of Results

SPC-1 Reported Data	
Tested Storage Product (TSP) Name: DataCore SANsymphony 10.0 *(Dual Node, High Availability, Hyper-converged)*	
Metric	**Reported Result**
SPC-1 IOPS™	1,201,961.83
SPC-1 Price-Performance™	$0.10/SPC-1 IOPS™
Total ASU Capacity	3,325.000 GB
Data Protection Level	Protected 2 *(Mirroring)*
Total Price	$115,142.76
Currency Used	U.S. Dollars
Target Country for availability, sales and support	USA

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

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

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

A **Data Protection Level** of **Protected 2** using ***Mirroring*** configures two or more identical copies of user data.

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

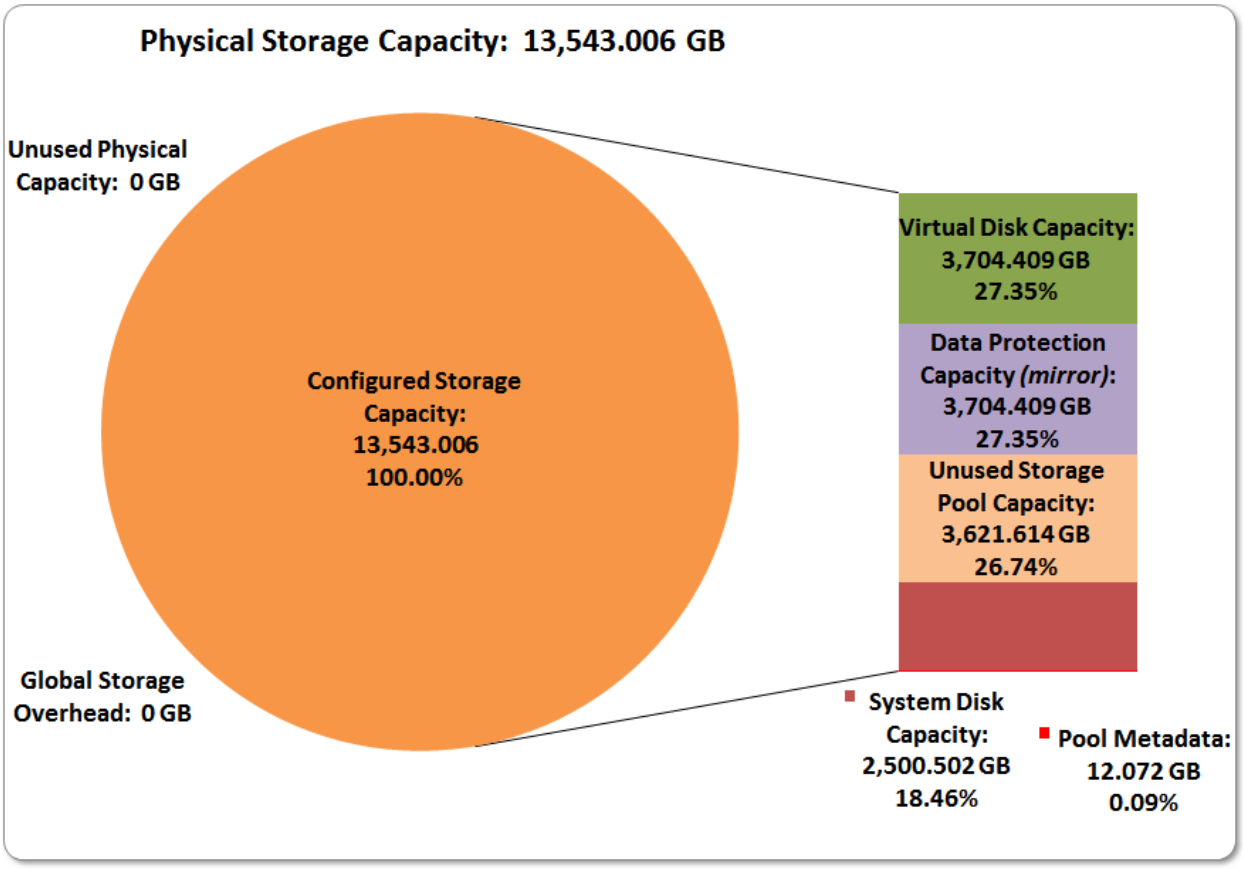
Total Price includes the cost of the Priced Storage Configuration plus three years of hardware maintenance and software support as detailed on page 17.

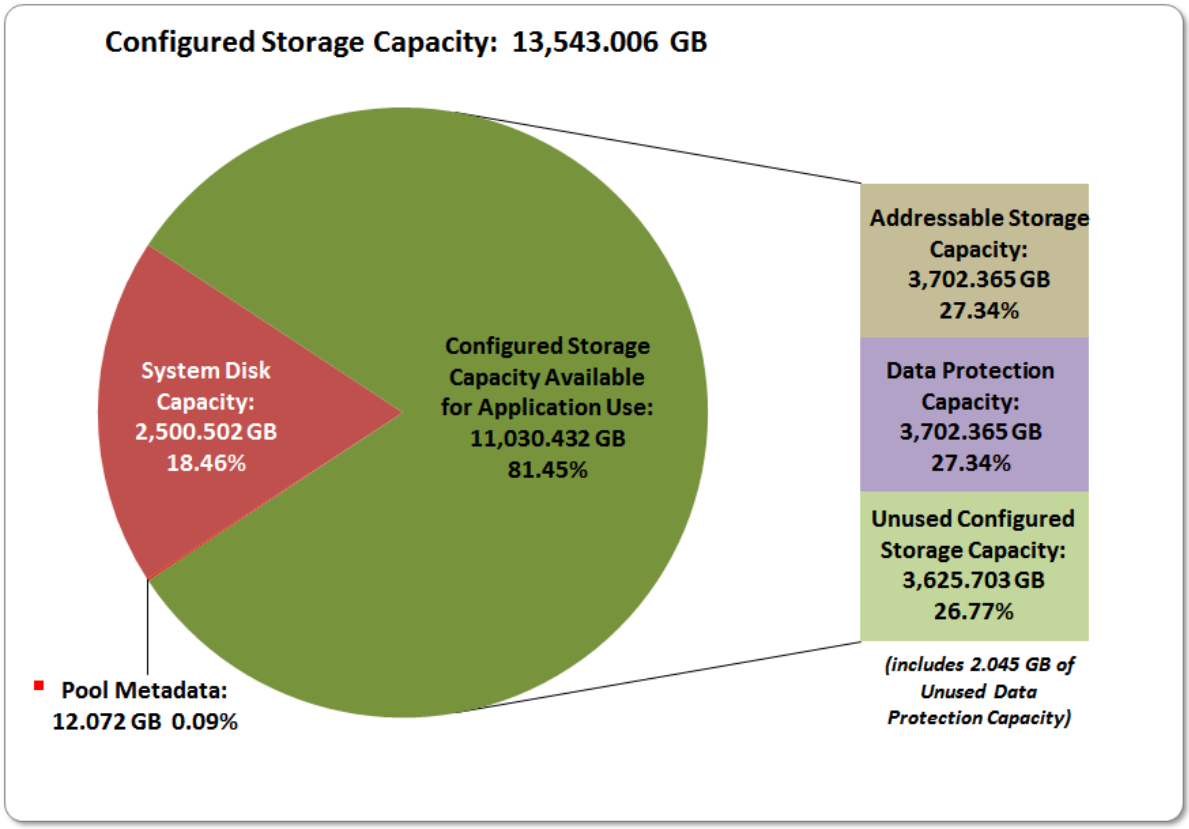
Currency Used is formal name for the currency used in calculating the **Total Price** and **SPC-1 Price-Performance™**. That currency may be the local currency of the **Target Country** or the currency of a difference country *(non-local currency).*

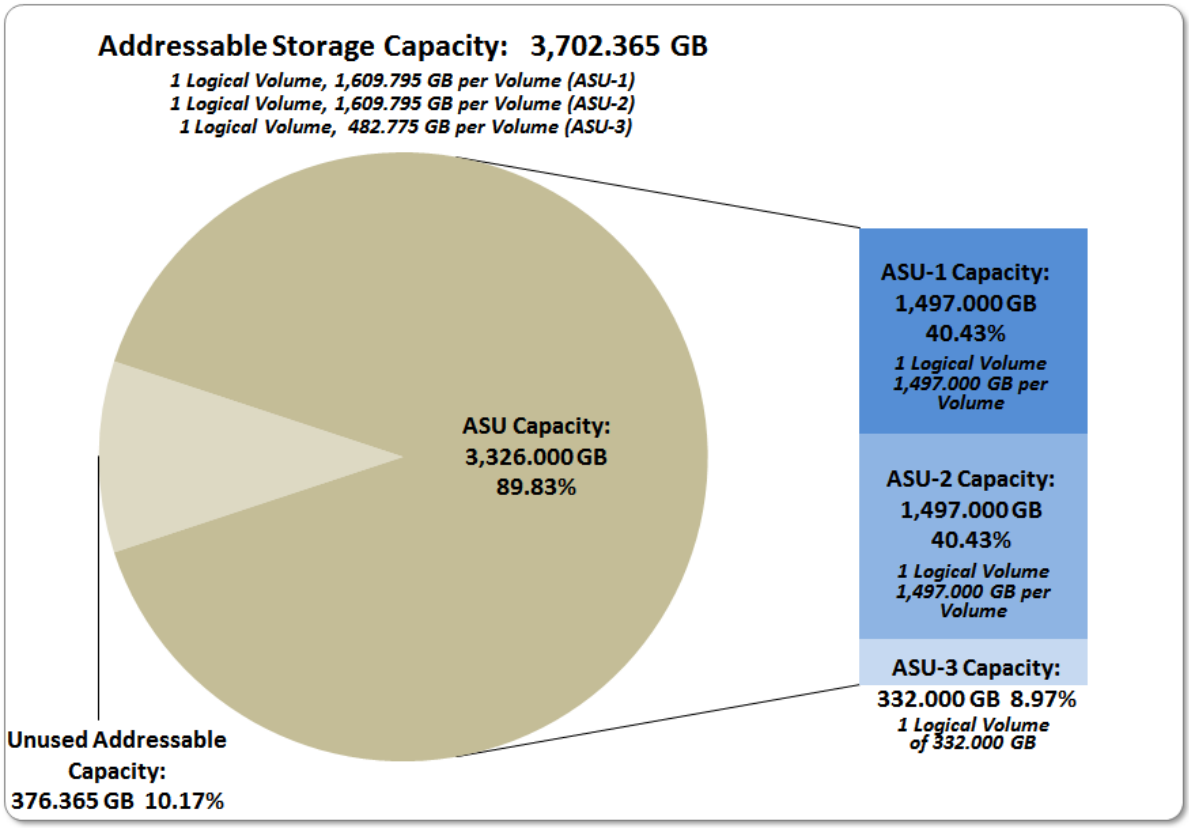
The **Target Country** is the country in which the Priced Storage Configuration is available for sale and in which the required hardware maintenance and software support is provided either directly from the Test Sponsor or indirectly via a third-party supplier.

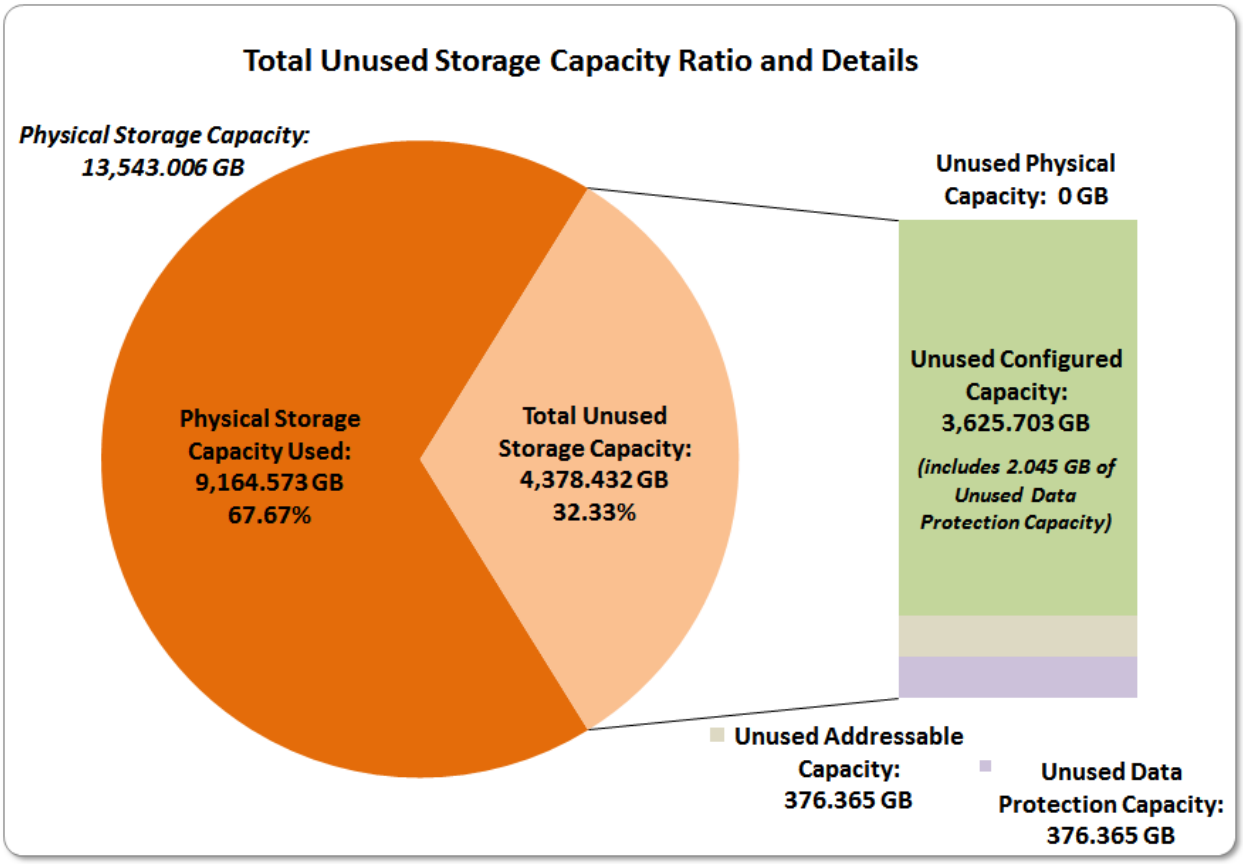
Storage Capacities, Relationships, and Utilization

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




Configured Storage Capacity: 13,543.006 GB
System Disk Capacity: 2,500.502 GB 18.46%
Configured Storage Capacity Available for Application Use: 11,030.432 GB 81.45%
Pool Metadata: 12.072 GB 0.09%
Addressable Storage Capacity: 3,702.365 GB 27.34%
Data Protection Capacity: 3,702.365 GB 27.34%
Unused Configured Storage Capacity: 3,625.703 GB 26.77%
(includes 2.045 GB of Unused Data Protection Capacity)


Addressable Storage Capacity: 3,702.365 GB
1 Logical Volume, 1,609.795 GB per Volume (ASU-1)
1 Logical Volume, 1,609.795 GB per Volume (ASU-2)
1 Logical Volume, 482.775 GB per Volume (ASU-3)
ASU Capacity: 3,326.000 GB 89.83%
Unused Addressable Capacity: 376.365 GB 10.17%
ASU-1 Capacity: 1,497.000 GB 40.43%
1 Logical Volume 1,497.000 GB per Volume
ASU-2 Capacity: 1,497.000 GB 40.43%
1 Logical Volume 1,497.000 GB per Volume
ASU-3 Capacity: 332.000 GB 8.97%
1 Logical Volume of 332.000 GB



SPC-1 Storage Capacity Utilization	
Application Utilization	32.33%
Protected Application Utilization	49.12%
Unused Storage Ratio	32.33%

Application Utilization: Total ASU Capacity *(3,326.000 GB)* divided by Physical Storage Capacity *(13,543.006 GB)*.

Protected Application Utilization: (Total ASU Capacity *(3,326.000 GB)* plus total Data Protection Capacity *(3,704.509 GB)* minus unused Data Protection Capacity *(378.409 GB)*) divided by Physical Storage Capacity *(13,543.006 GB)*.

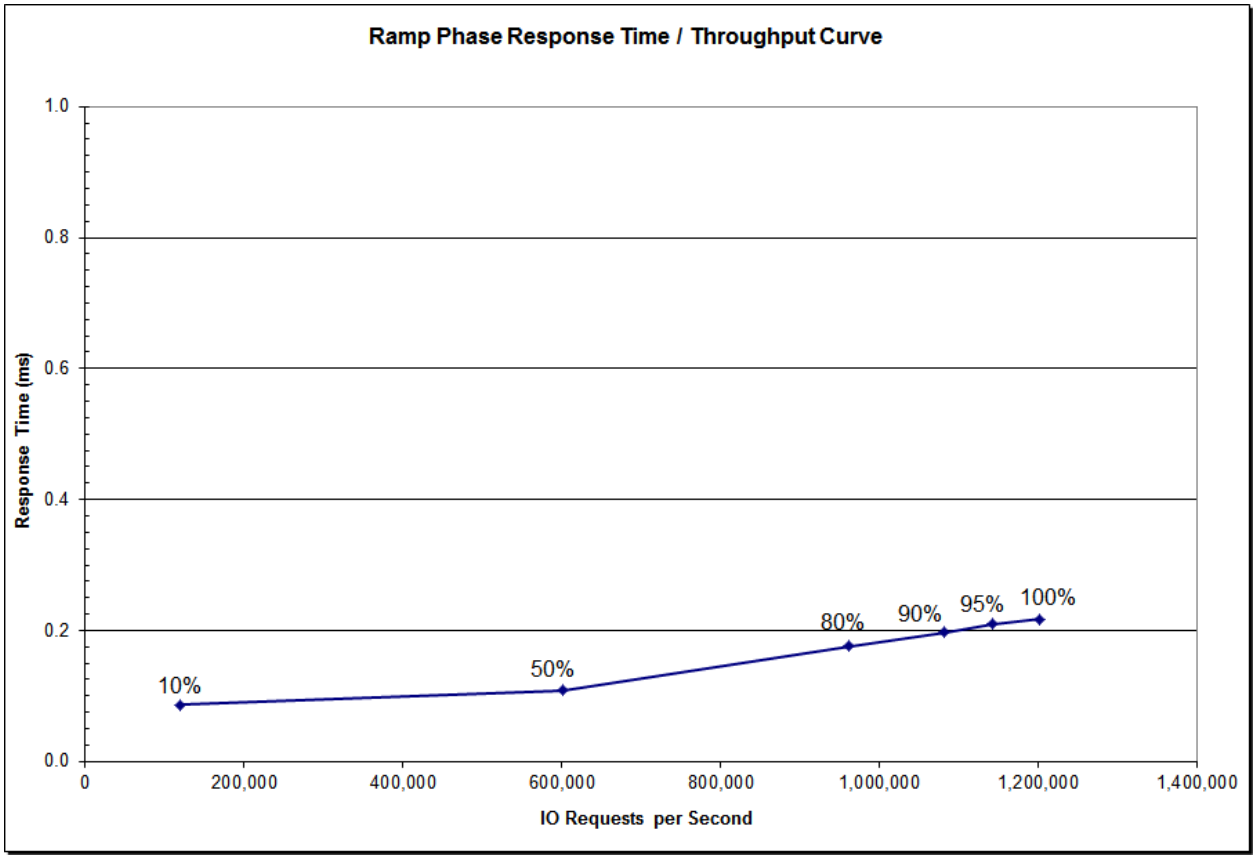
Unused Storage Ratio: Total Unused Capacity *(4,378.432 GB)* divided by Physical Storage Capacity *(13,543.006 GB)* and may not exceed 45%.

Detailed information for the various storage capacities and utilizations is available on pages 25-26.

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	120,251.64	601,236.81	961,968.26	1,082,314.26	1,142,289.52	1,201,961.83
Average Response Time (ms):						
All ASUs	0.09	0.11	0.18	0.20	0.21	0.22
ASU-1	0.08	0.10	0.18	0.20	0.22	0.22
ASU-2	0.12	0.15	0.22	0.24	0.26	0.28
ASU-3	0.09	0.10	0.14	0.17	0.18	0.19
Reads	0.10	0.13	0.24	0.26	0.27	0.27
Writes	0.08	0.10	0.14	0.16	0.17	0.18

Priced Storage Configuration Pricing

The Priced Storage Configuration pricing information is not embedded in this document due it size and format. The pricing information is available via the following hyperlink:

Priced Storage Configuration Pricing

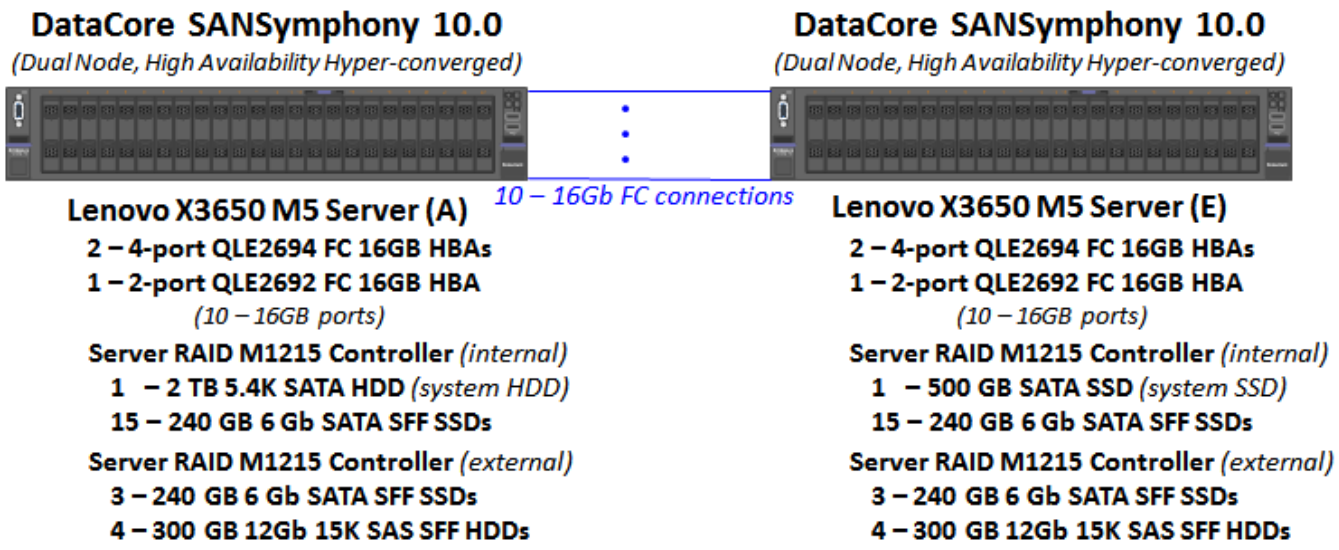
The above pricing includes hardware maintenance and software support for three years, 7 days per week, 24 hours per day. The hardware maintenance and software support provides the following:

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

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

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

Priced Storage Configuration Diagram



Priced Storage Configuration Components

Priced Storage Configuration
DataCore SANsymphony 10.0 ***(Dual Node, High Availability, Hyper-converged)***
1 – Lenovo X3650 M5 Server, each with: 2 – Intel® Xeon® 2.30 GHz E5-2696 V3 processors each with 18 cores, 45 MB Intel Smart Cache 768 GB main memory *(508 GB configured for DataCore SANsymphony 10.0)* Windows 2008 R2 Enterprise Server w/SP1 PCIe 2 – 4-port QLE2694 16Gb HBAs 1 – 2-port QLE2692 16Gb HBA *(10 – 16Gb ports total)* 1 – Server RAID M1215 SAS/SATA Controller *(internal)* 1 – Server RAID M1215 SAS/SATA Controllers *(external)* 1 – System Disk (*connected to the internal controller)* Server A: 2TB, 5.4K SATA HDD Server E: Samsun 850EVO 2.5” 500GB SATA III SSD 1 – Samsung 850EVO 2.5” 500GB SATA III SSD *(Server E system SSD, connected to the internal controller)* 15 – 240 GB, 6 Gb SATA SFF SSDs *(Samsung SM863 MZ-7KM204E)* *(connected to the internal controller)* 3 – 240 GB, 6 Gb SATA SFF SSDs *(Samsung SM863 MZ-7KM204E)* *(connected to the external controller)* 4 – 300 GB, 12 Gb, 15K SAS SFF HDDs (HGST Ultrastar C15K600) *(connected to the external controller)*
2 – APC Smart UPS X 1500VA Rack/Tower 120V – SMX1500RM2UAPC

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 21 *(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 storage network is illustrated on page 21 *(Benchmark Configuration/Tested Storage Configuration Diagram)*.

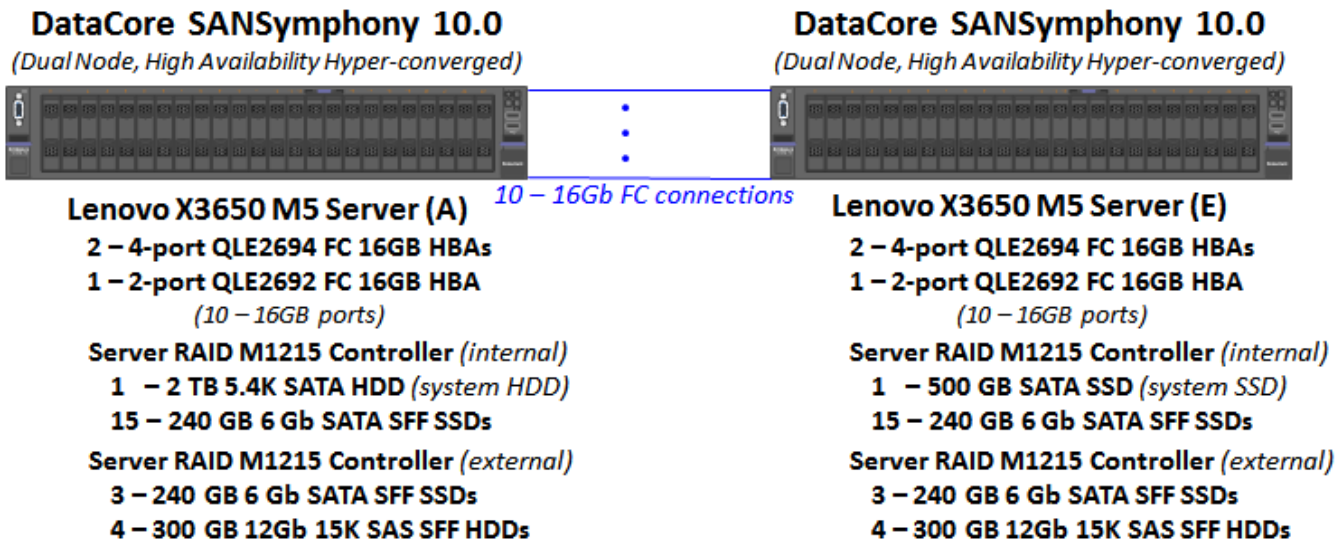
Host System(s) and Tested Storage Configuration (TSC) Table of Components

Clause 9.4.3.4.3

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

The Host System(s) and TSC table of components may be found on page 22 *(Host System and Tested Storage Configuration Components)*.

Benchmark Configuration/Tested Storage Configuration Diagram



Host System and Tested Storage Configuration Components

Host Systems and Tested Storage Configuration
DataCore SANsymphony 10.0 ***(Dual Node, High Availability, Hyper-converged)***
1 – Lenovo X3650 M5 Server, each with: 2 – Intel® Xeon® 2.30 GHz E5-2696 V3 processors each with 18 cores, 45 MB Intel Smart Cache 768 GB main memory *(508 GB configured for DataCore SANsymphony 10.0)* Windows 2008 R2 Enterprise Server w/SP1 PCIe 2 – 4-port QLE2694 16Gb HBAs 1 – 2-port QLE2692 16Gb HBA *(10 – 16Gb ports total)* 1 – Server RAID M1215 SAS/SATA Controller *(internal)* 1 – Server RAID M1215 SAS/SATA Controllers *(external)* 1 – System Disk (*connected to the internal controller)* Server A: 2TB, 5.4K SATA HDD Server E: Samsun 850EVO 2.5” 500GB SATA III SSD 1 – Samsung 850EVO 2.5” 500GB SATA III SSD *(Server E system SSD, connected to the internal controller)* 15 – 240 GB, 6 Gb SATA SFF SSDs *(Samsung SM863 MZ-7KM204E)* *(connected to the internal controller)* 3 – 240 GB, 6 Gb SATA SFF SSDs *(Samsung SM863 MZ-7KM204E)* *(connected to the external controller)* 4 – 300 GB, 12 Gb, 15K SAS SFF HDDs (HGST Ultrastar C15K600) *(connected to the external controller)*
2 – APC Smart UPS X 1500VA Rack/Tower 120V – SMX1500RM2UAPC

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 66 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 70 contains the detailed information that describes how to create and configure the logical TSC.

SPC-1 Workload Generator Storage Configuration

Clause 9.4.3.5.3

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

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

ASU Pre-Fill

Clause 5.3.3

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

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

SPC-1 DATA REPOSITORY

This portion of the Full Disclosure Report presents the detailed information that fully documents the various SPC-1 storage capacities and mappings used in the Tested Storage Configuration. SPC-1 Data Repository Definitions on page 62 contains definitions of terms specific to the SPC-1 Data Repository.

Storage Capacities and Relationships

Clause 9.4.3.6.1

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

SPC-1 Storage Capacities

The Physical Storage Capacity consisted of 13,543.006 GB distributed over 36 solid state drives (SSDs) each with a formatted capacity of 240.055 GB, 8 disk drives (HDDs) each with a formatted capacity of 300.068 GB, disk drive with a formatted capacity of 500.105 GB and 1 disk drive with a formatted capacity of 2,000.396 GB. There was 0.000 GB (0.00%) of Unused Storage within the Physical Storage Capacity. Global Storage Overhead consisted of 0,000 GB (0.00%) of the Physical Storage Capacity. There was 3,625.703 GB (26.77%) of Unused Storage within the Configured Storage Capacity. The Total ASU Capacity utilized 89.93% of the Addressable Storage Capacity resulting in 376.365 GB (10.17%) of Unused Storage within the Addressable Storage Capacity. The Data Protection *(Mirroring)* capacity was 3,704.409 GB of which 3,326.000 GB was utilized. The total Unused Storage capacity was 4,002.068 GB.

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

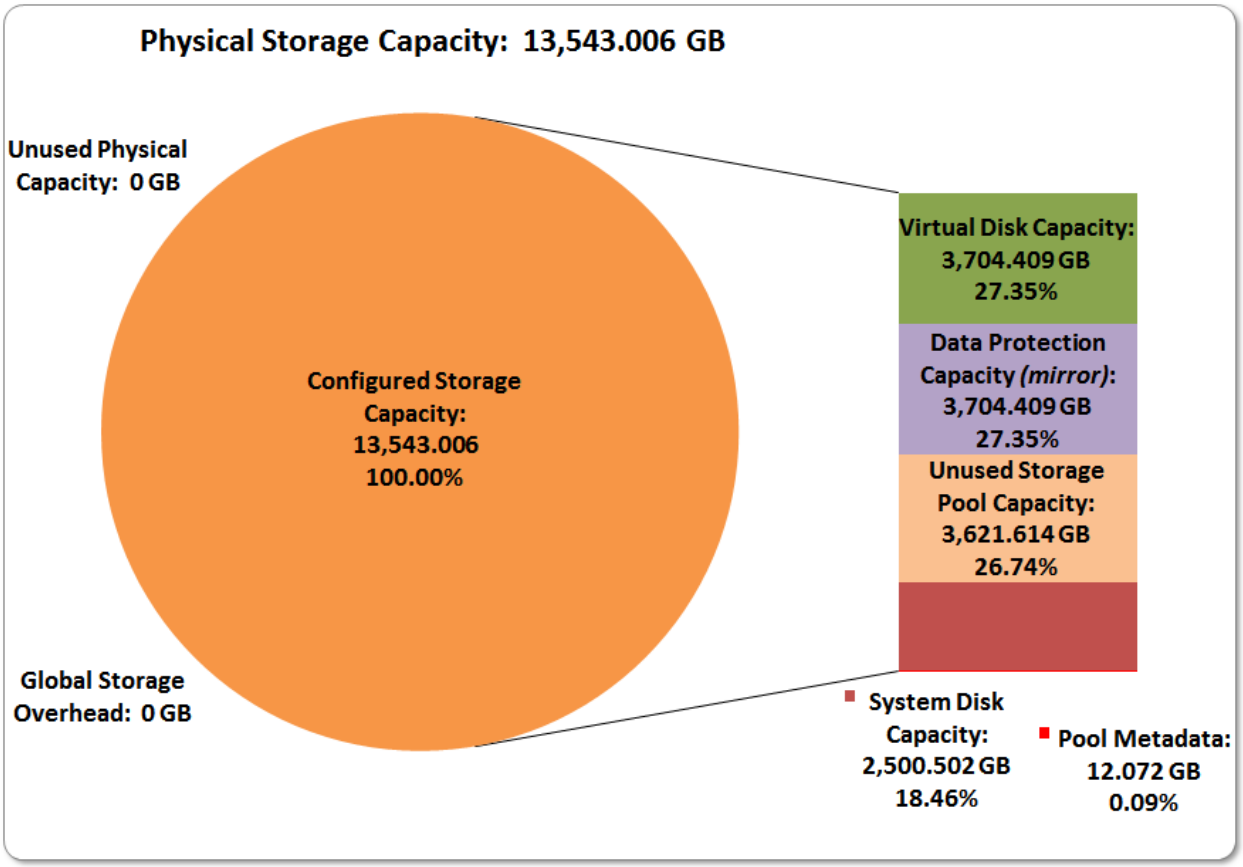
SPC-1 Storage Capacities

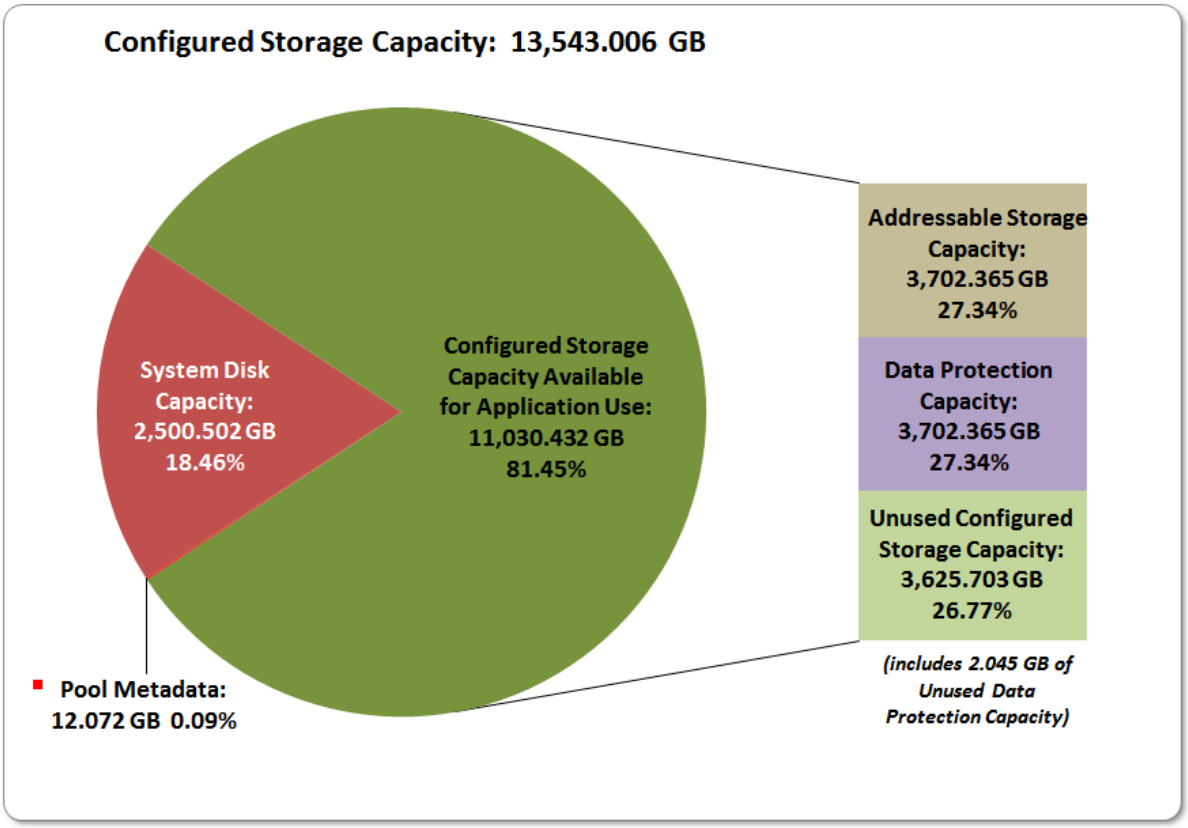
Storage Hierarchy Component	Units	Capacity
Total ASU Capacity	Gigabytes (GB)	3,326.000
Addressable Storage Capacity	Gigabytes (GB)	3,702.365
Configured Storage Capacity	Gigabytes (GB)	13,543.006
Physical Storage Capacity	Gigabytes (GB)	13,543.006
Data Protection *(Mirroring)*	Gigabytes (GB)	3,704.409
Required Storage *(system disk, metadata)*	Gigabytes (GB)	2,512.573
Global Storage Overhead	Gigabytes (GB)	0.000
Total Unused Storage	Gigabytes (GB)	4,002.068

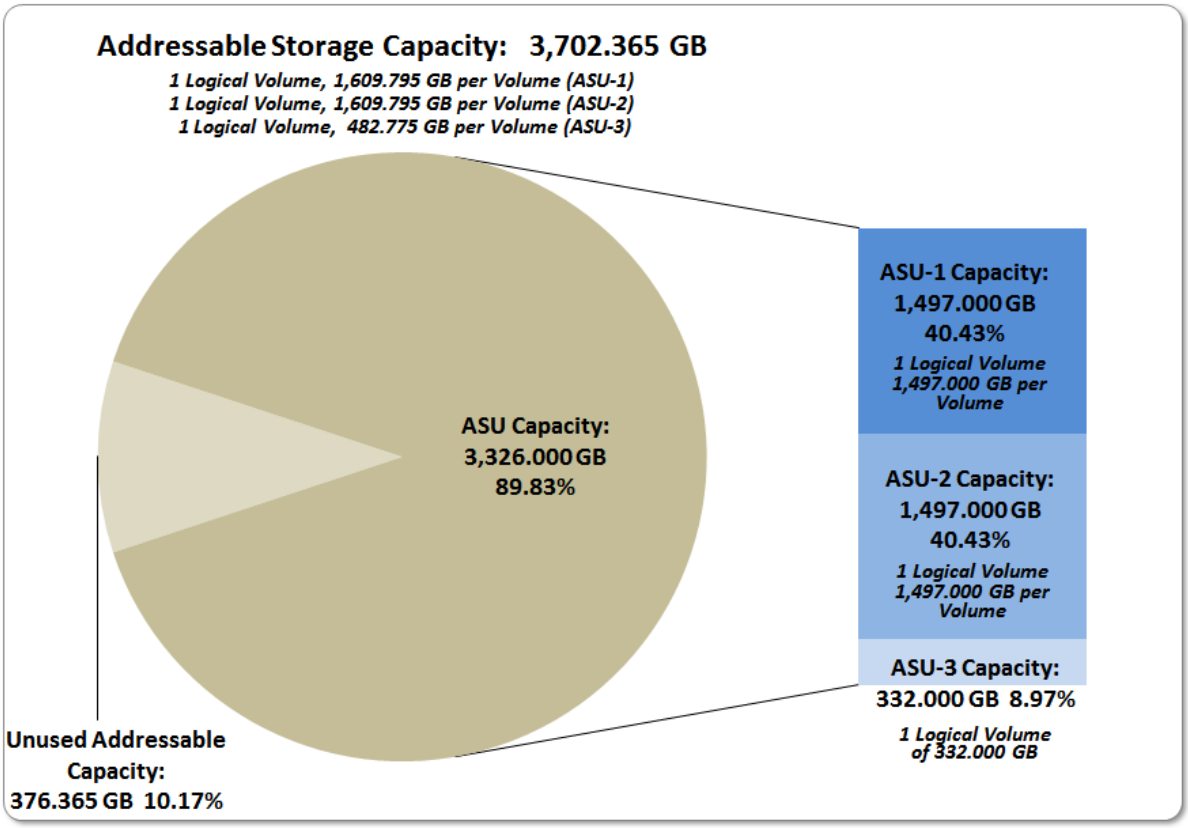
SPC-1 Storage Hierarchy Ratios

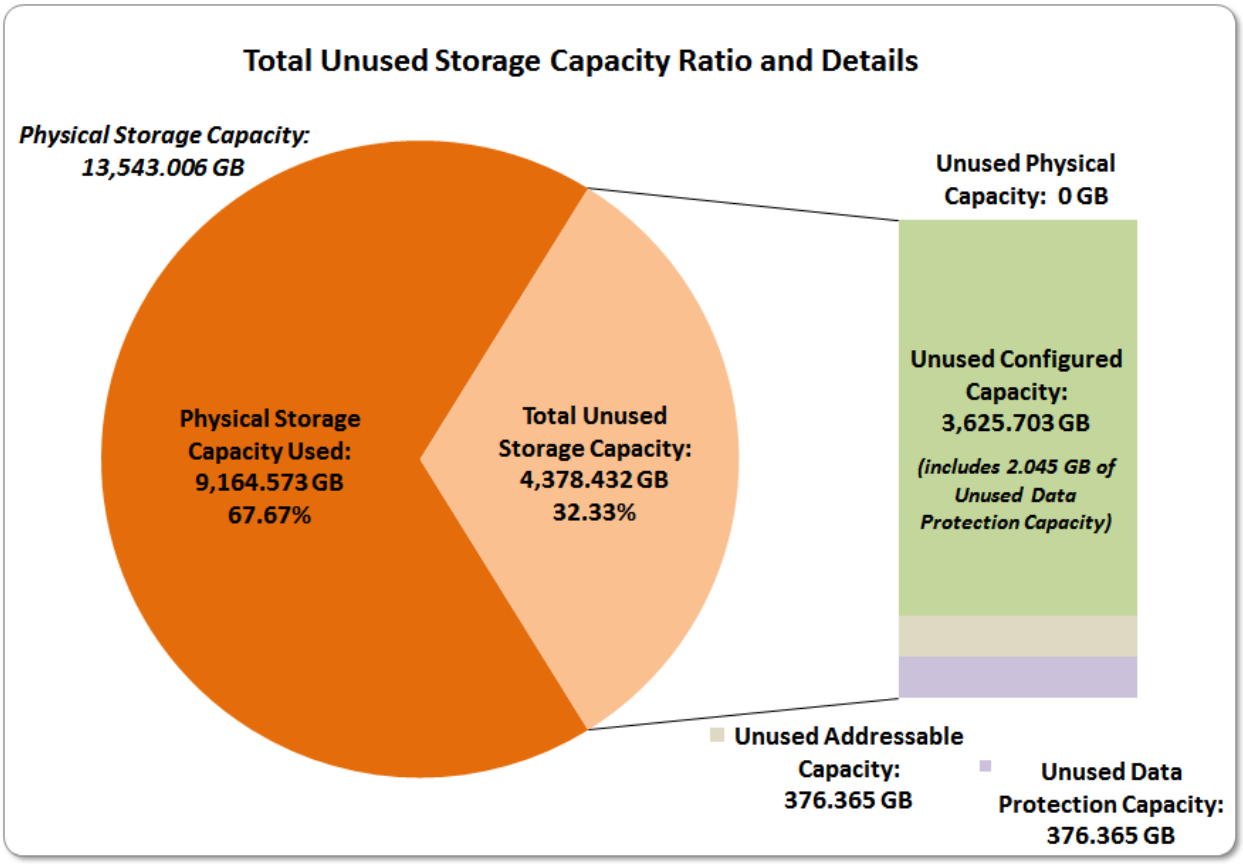
	Addressable Storage Capacity	Configured Storage Capacity	Physical Storage Capacity
Total ASU Capacity	89.83%	24.56%	24.56%
Required for Data Protection *(Mirroring)*		27.35%	27.35%
Addressable Storage Capacity		27.34%	27.34%
Required Storage *(system disk, metadata)*		18.55%	18.55%
Configured Storage Capacity			100.00%
Global Storage Overhead			0.00%
Unused Storage:			
Addressable	10.17%		
Configured		26.77%	
Physical			0.00%

SPC-1 Storage Capacity Charts




Configured Storage Capacity: 13,543.006 GB
System Disk Capacity: 2,500.502 GB 18.46%
Configured Storage Capacity Available for Application Use: 11,030.432 GB 81.45%
Pool Metadata: 12.072 GB 0.09%
Addressable Storage Capacity: 3,702.365 GB 27.34%
Data Protection Capacity: 3,702.365 GB 27.34%
Unused Configured Storage Capacity: 3,625.703 GB 26.77%
(includes 2.045 GB of Unused Data Protection Capacity)


Addressable Storage Capacity: 3,702.365 GB
1 Logical Volume, 1,609.795 GB per Volume (ASU-1)
1 Logical Volume, 1,609.795 GB per Volume (ASU-2)
1 Logical Volume, 482.775 GB per Volume (ASU-3)
ASU Capacity: 3,326.000 GB 89.83%
Unused Addressable Capacity: 376.365 GB 10.17%
ASU-1 Capacity: 1,497.000 GB 40.43%
1 Logical Volume 1,497.000 GB per Volume
ASU-2 Capacity: 1,497.000 GB 40.43%
1 Logical Volume 1,497.000 GB per Volume
ASU-3 Capacity: 332.000 GB 8.97%
1 Logical Volume of 332.000 GB



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	24.56%
Protected Application Utilization	49.12%
Unused Storage Ratio	32.33%

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 (1,497.000 GB)
1 Logical Volume 1,609.795 GB per Logical Volume (1,497.000 GB used per Logical Volume)
ASU-2 (GB)
1 Logical Volume 1,609.795 GB per Logical Volume (1,497.000 GB used per Logical Volume)
ASU-3 (GB)
1 Logical Volume 482.775 GB per Logical Volume (332.000 GB used per Logical Volume)

The Data Protection Level used for all Logical Volumes was **Protected 2** using ***Mirroring*** as described on page 12. See "ASU Configuration" in the **IOPS Test Results File** for more detailed configuration information.

SPC-1 BENCHMARK EXECUTION RESULTS

This portion of the Full Disclosure Report documents the results of the various SPC-1 Tests, Test Phases, and Test Runs. An SPC-1 glossary on page 62 contains definitions of terms specific to the SPC-1 Tests, Test Phases, and Test Runs.

Clause 5.4.3

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

SPC-1 Tests, Test Phases, and Test Runs

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

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

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

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

"Ramp-Up" Test Runs

Clause 5.3.13

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

Clause 5.3.13.3

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

Clause 9.4.3.7.1

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

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

Primary Metrics Test – Sustainability Test Phase

Clause 5.4.4.1.1

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

Clause 5.4.4.1.2

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

Clause 5.4.4.1.4

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

Clause 9.4.3.7.2

For the Sustainability Test Phase the FDR shall contain:

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

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 82.

Sustainability Test Results File

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

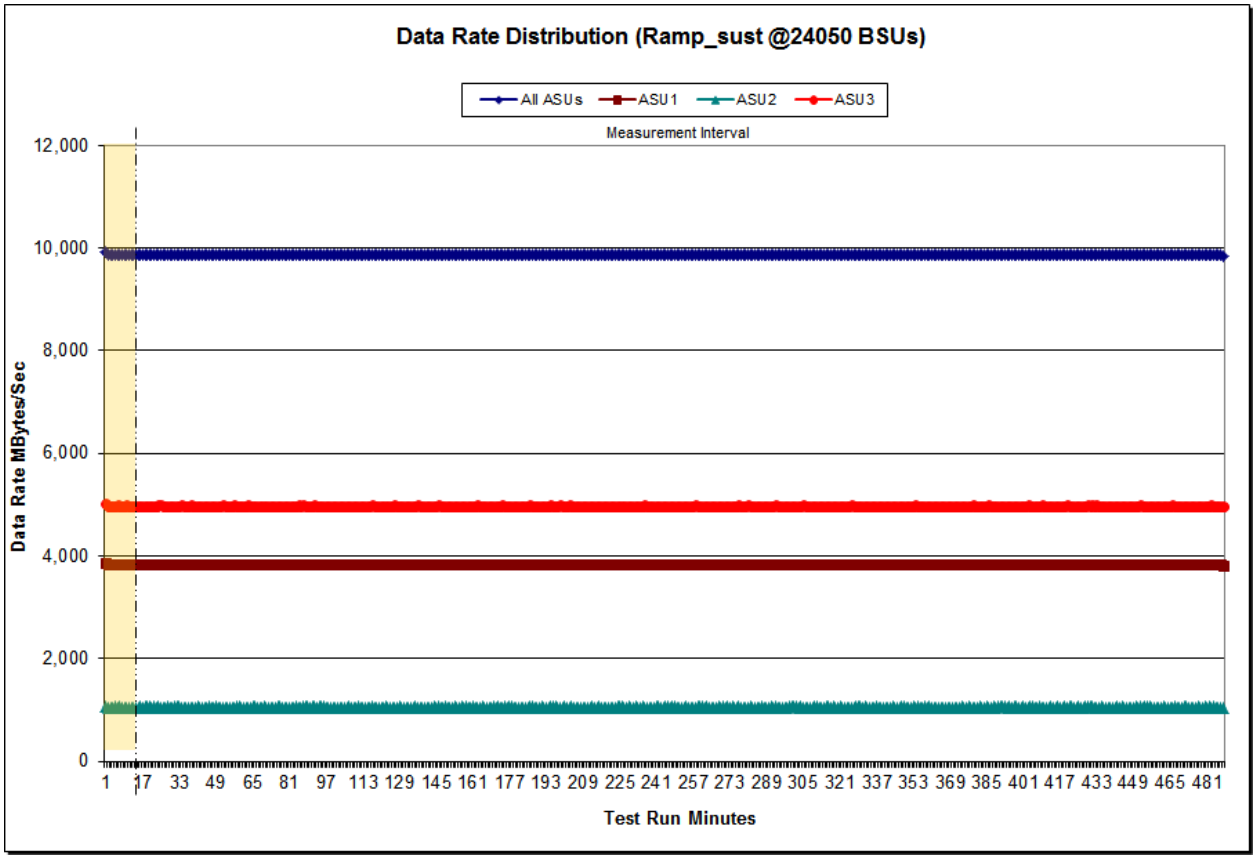
Sustainability Test Results File

Sustainability – Data Rate Distribution Data (MB/second)

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

Sustainability Data Rate Distribution Data Table

Sustainability – Data Rate Distribution Graph

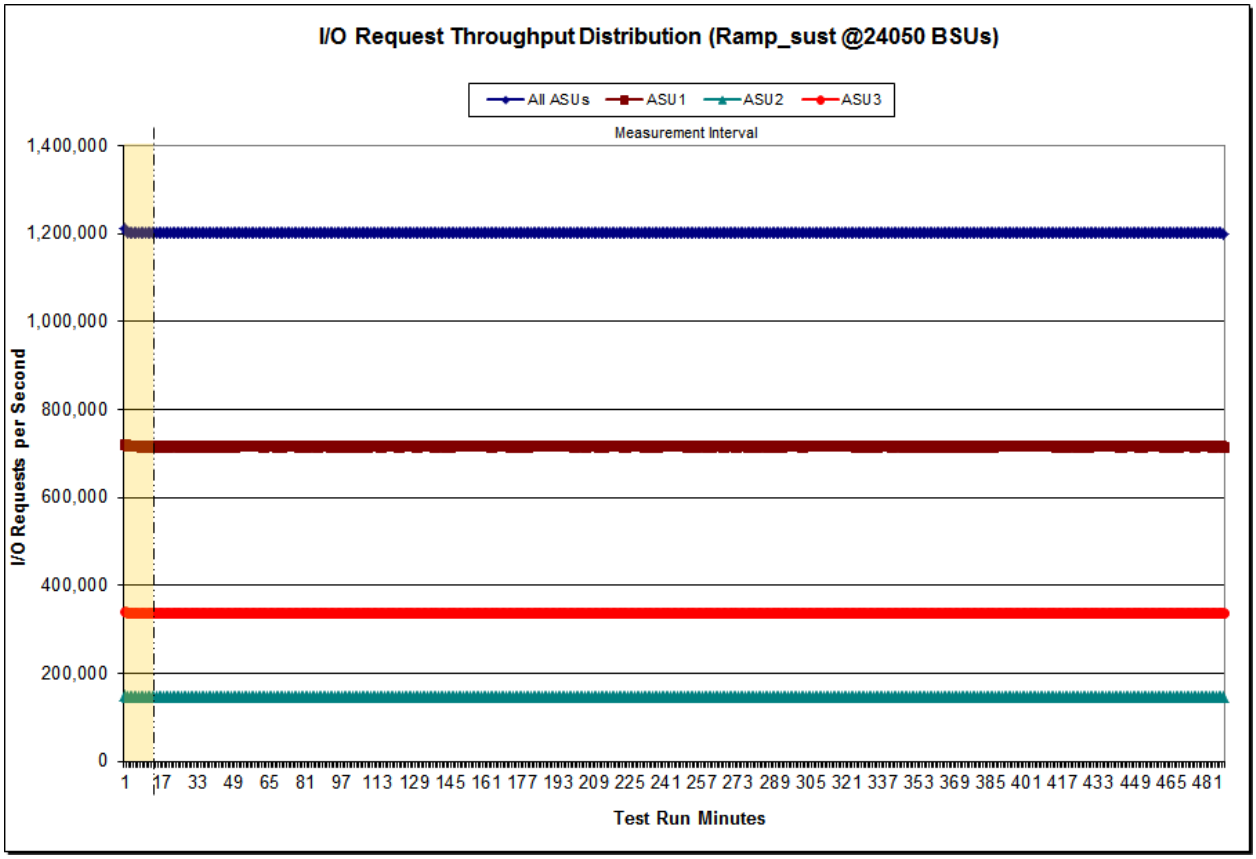


Sustainability – I/O Request Throughput Distribution Data

The Sustainability I/O Request Throughput Data table 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

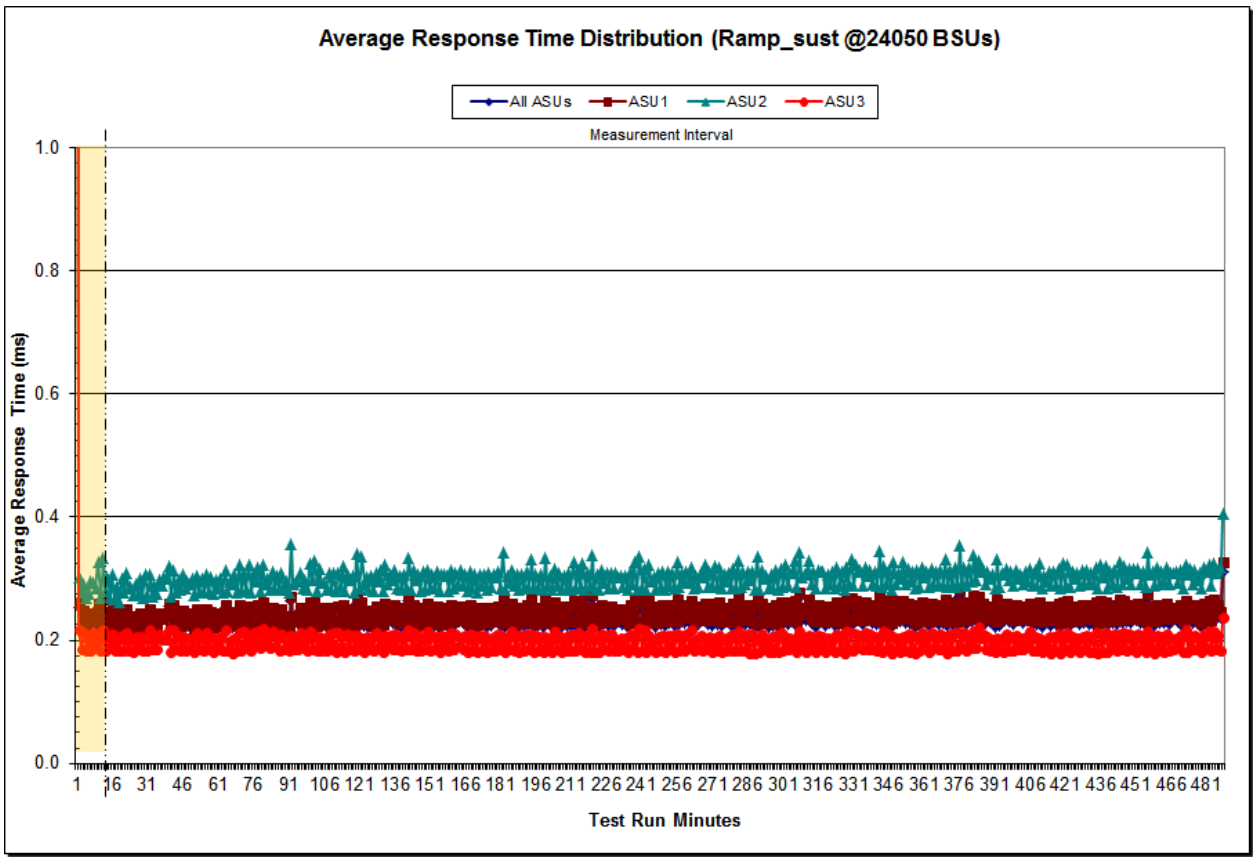


Sustainability – Average Response Time (ms) Distribution Data

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

Sustainability Average Response Time Distribution Data Table

Sustainability – Average Response Time (ms) Distribution Graph

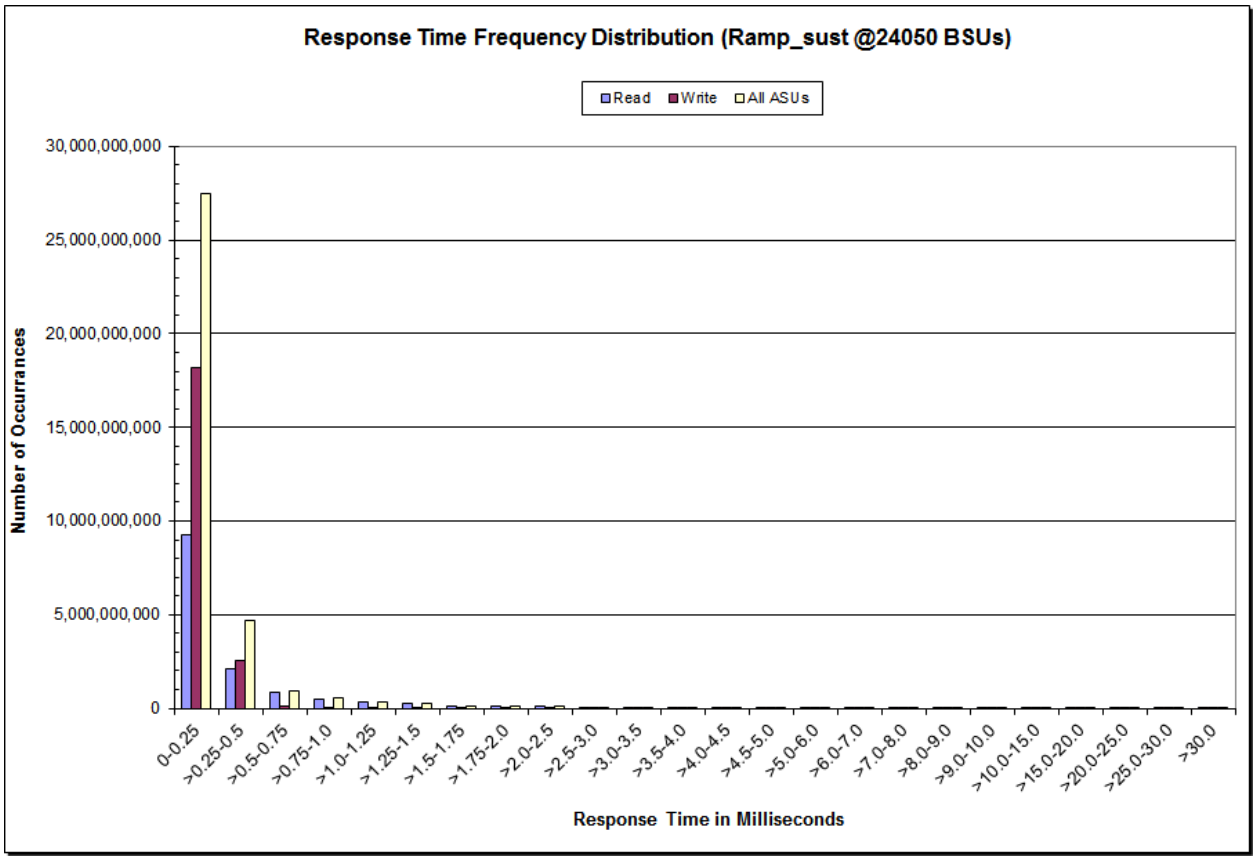


Sustainability – Response Time Frequency Distribution Data

The Sustainability Response Time Frequency Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Sustainability Response Time Frequency Distribution Data Table

Sustainability – Response Time Frequency Distribution Graph



Sustainability – Measured Intensity Multiplier and Coefficient of Variation

Clause 3.4.3

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

Clauses 5.1.10 and 5.3.15.2

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

Clause 5.3.15.3

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

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

Primary Metrics Test – IOPS Test Phase

Clause 5.4.4.2

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

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

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

Clause 9.4.3.7.3

For the IOPS Test Phase the FDR shall contain:

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

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 82.

IOPS Test Results File

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

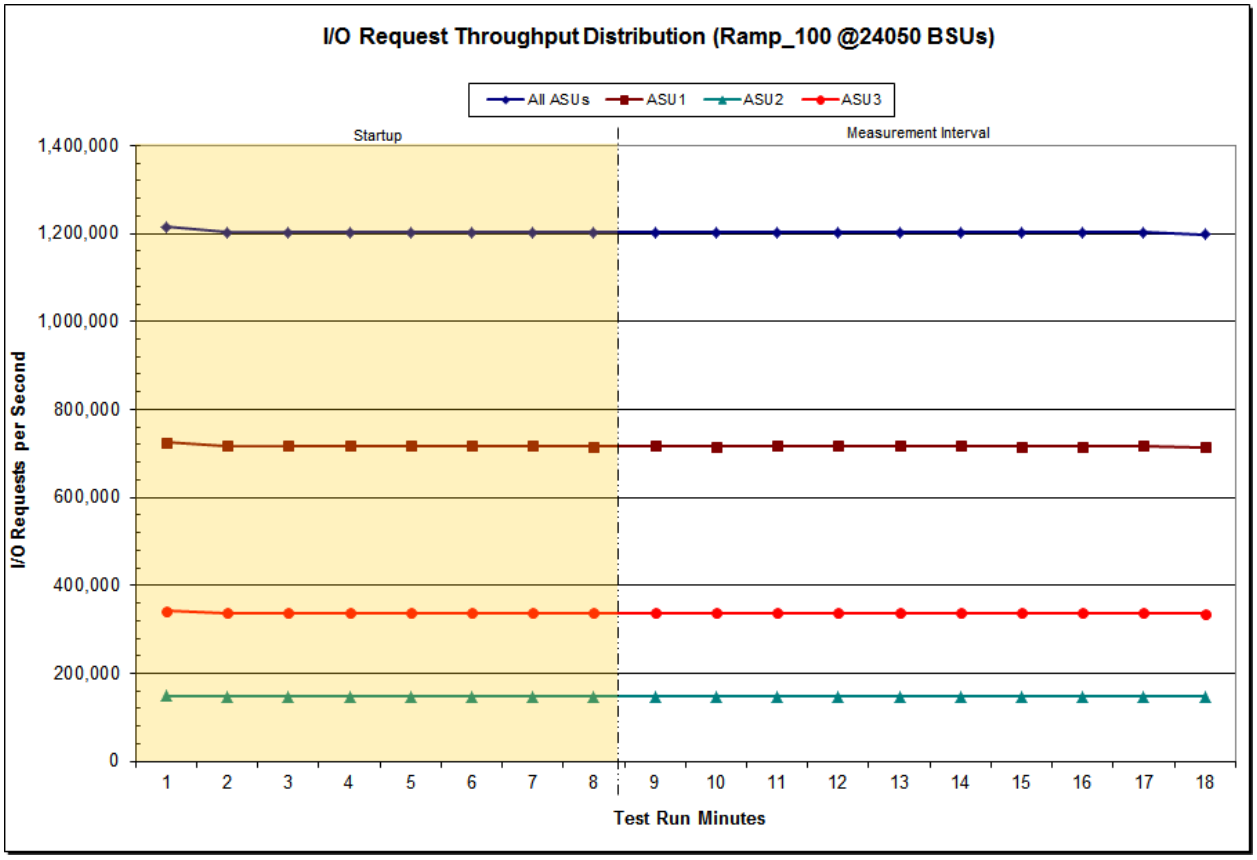
IOPS Test Results File

IOPS Test Run – I/O Request Throughput Distribution Data

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

I/O Request Throughput Distribution Data Table

IOPS Test Run – I/O Request Throughput Distribution Graph

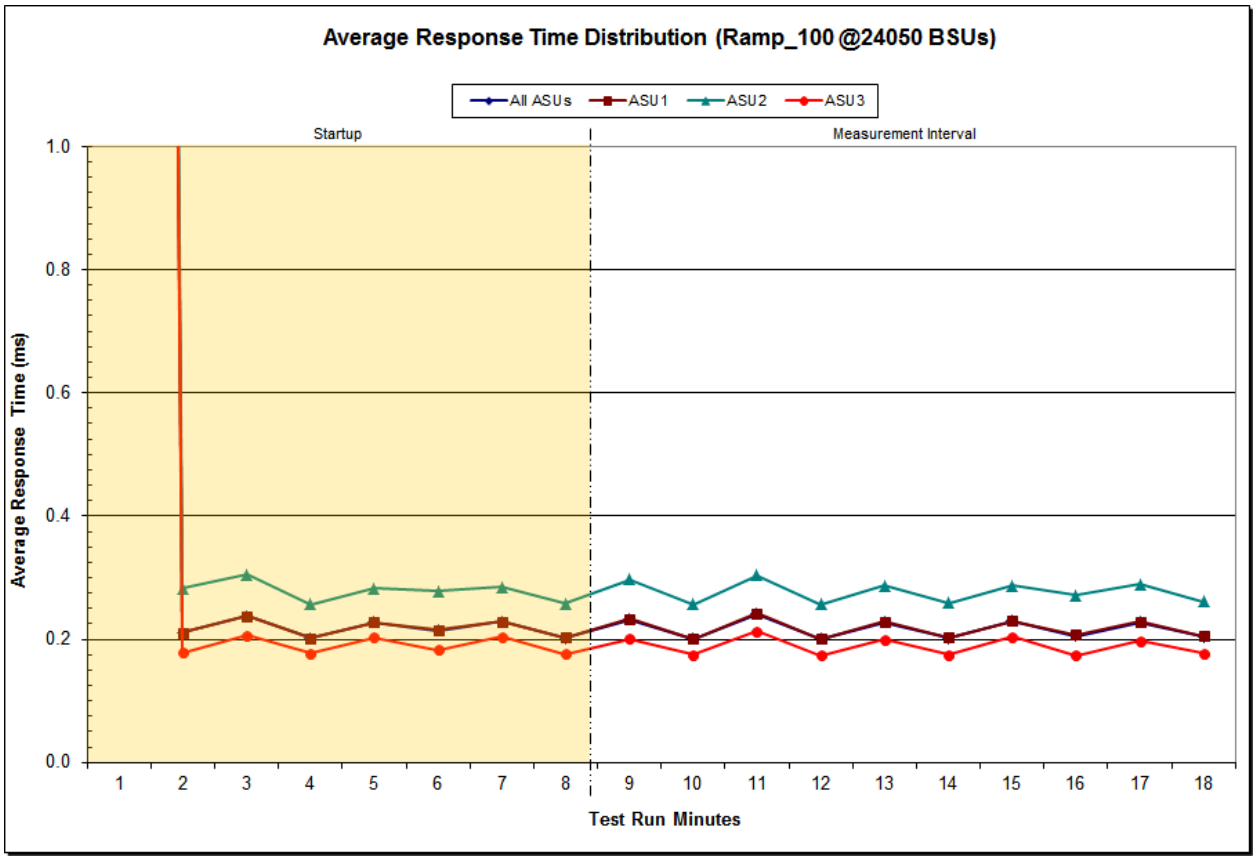


IOPS Test Run – Average Response Time (ms) Distribution Data

The IOPS Test Run – Average Response Time Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

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

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

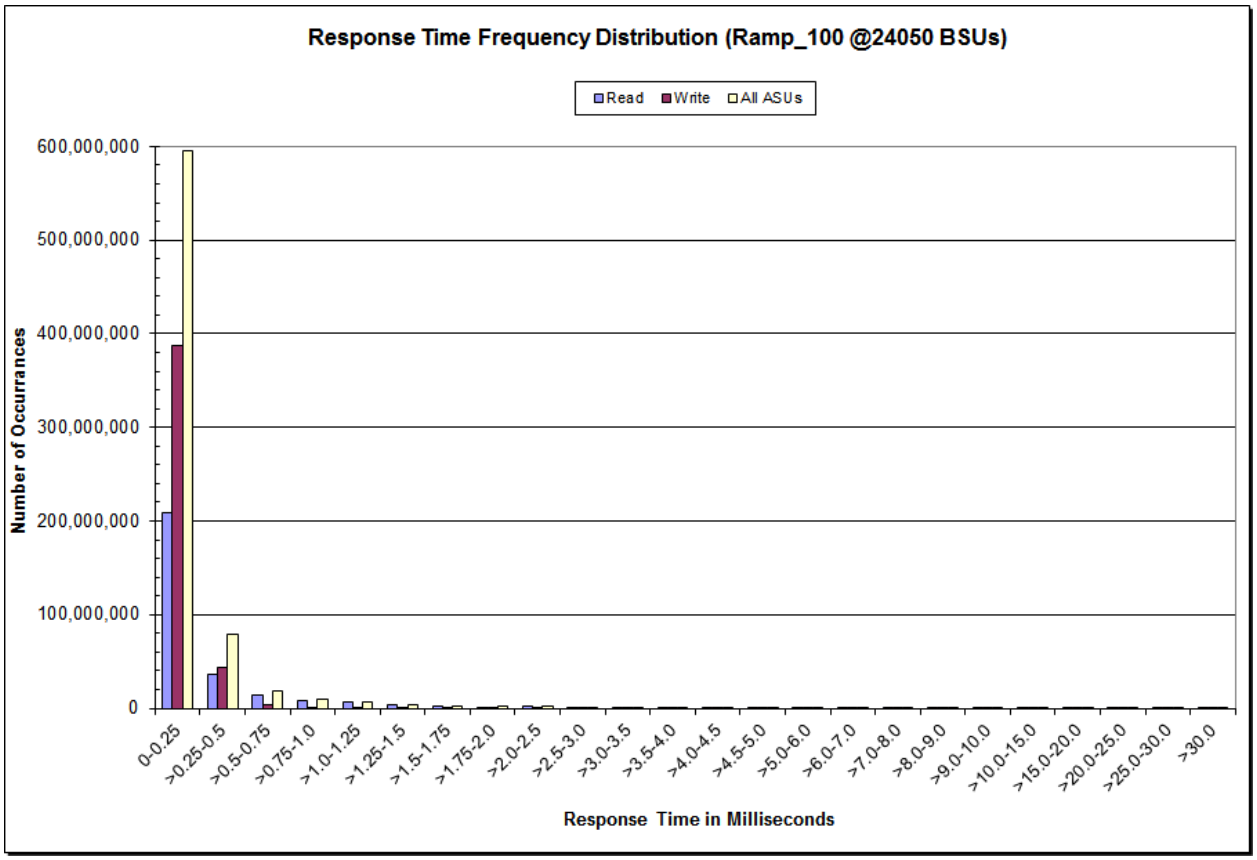


IOPS Test Run –Response Time Frequency Distribution Data

The IOPS Test Run –Response Time Frequency Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

IOPS Test Run –Response Time Frequency Distribution Data Table

IOPS Test Run –Response Time Frequency Distribution Graph



IOPS Test Run – I/O Request Information

I/O Requests Completed in the Measurement Interval
721,143,327
I/O Requests Completed with Response Time = or < 30 ms
721,127,626
I/O Requests Completed with Response Time > 30 ms
15,701

IOPS Test Run – Measured Intensity Multiplier and Coefficient of Variation

Clause 3.4.3

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

Clauses 5.1.10 and 5.3.15.2

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

Clause 5.3.15.3

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

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

Primary Metrics Test – Response Time Ramp Test Phase

Clause 5.4.4.3

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

The five Response Time Ramp Test Runs, in conjunction with the IOPS Test Run (100%), demonstrate the relationship between Average Response Time and I/O Request Throughput for the Tested Storage Configuration (TSC) as illustrated in the response time/throughput curve on page 16.

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

Clause 9.4.3.7.4

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

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

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 82.

Response Time Ramp Test Results File

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

95% Load Level

90% Load Level

80% Load Level

50% Load Level

10% Load Level

Response Time Ramp Distribution (IOPS) Data

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

In addition to appearing below, the Response Time Ramp Distribution (IOPS) Data table is available via the following URL.

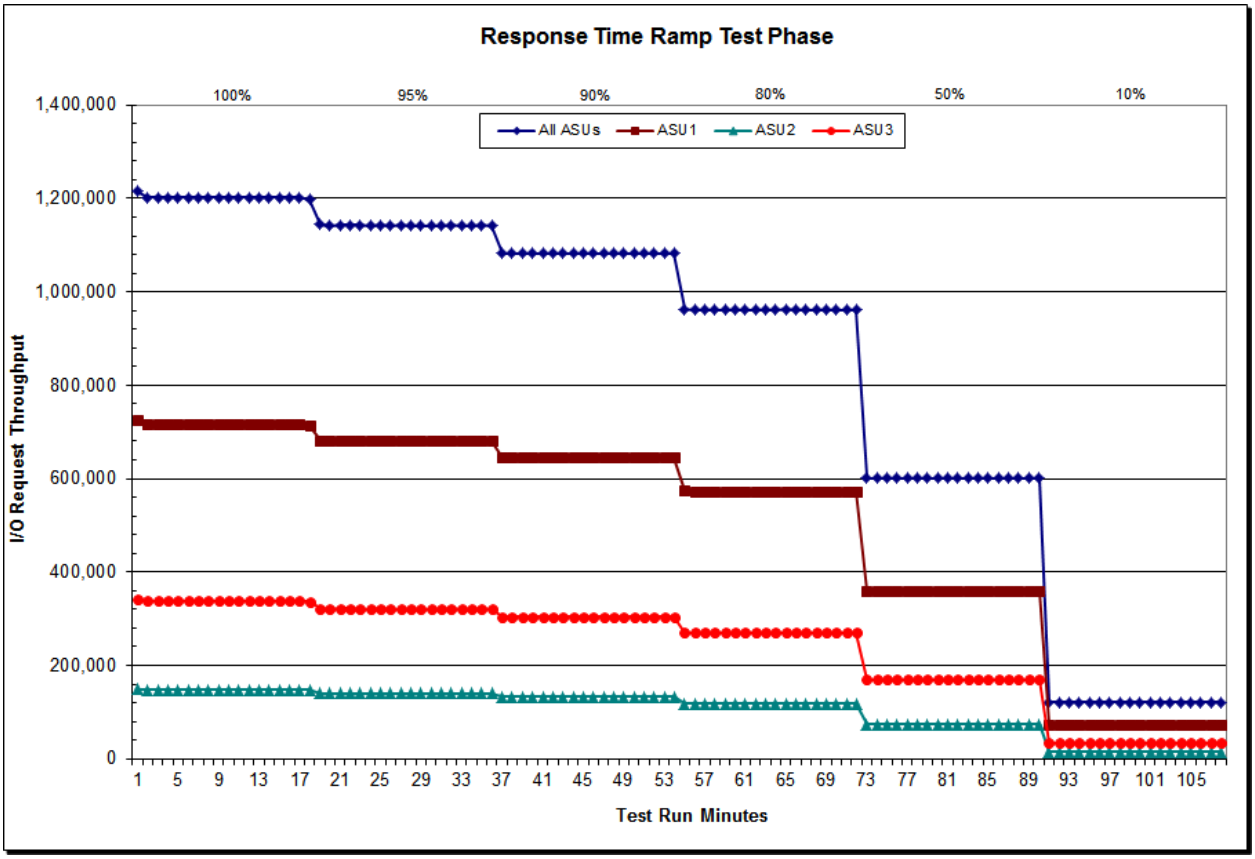
IOPS Test Run –Response Time Frequency Distribution Data Table

100% Load Level: 24,050 BSUs	Start	Stop	Interval	Duration	95% Load Level: 22,847 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	22:25:50	22:33:51	0-7	0:08:01	Start-Up/Ramp-Up	22:49:49	22:57:50	0-7	0:08:01
Measurement Interval	22:33:51	22:43:51	8-17	0:10:00	Measurement Interval	22:57:50	23:07:50	8-17	0:10:00
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3	*(60 second intervals)*	All ASUs	ASU-1	ASU-2	ASU-3
0	1,215,710.75	724,517.50	149,549.15	341,644.10	0	1,144,808.93	682,303.63	140,749.65	321,755.65
1	1,202,478.42	716,620.60	147,885.80	337,972.02	1	1,142,256.85	680,761.78	140,547.35	320,947.72
2	1,202,407.23	716,669.02	147,840.97	337,897.25	2	1,142,486.30	680,922.52	140,521.27	321,042.52
3	1,202,470.77	716,689.33	147,975.72	337,805.72	3	1,142,366.40	680,794.30	140,535.15	321,036.95
4	1,202,681.23	716,905.48	147,940.28	337,835.47	4	1,142,424.65	680,900.78	140,556.22	320,967.65
5	1,202,388.50	716,624.02	147,956.20	337,808.28	5	1,142,251.80	680,746.92	140,499.25	321,005.63
6	1,202,724.25	716,674.07	148,042.88	338,007.30	6	1,142,540.12	680,861.02	140,599.22	321,079.88
7	1,202,456.07	716,545.93	147,938.52	337,971.62	7	1,142,152.68	680,697.33	140,472.02	320,983.33
8	1,202,622.15	716,796.22	147,901.70	337,924.23	8	1,142,352.95	681,071.57	140,426.38	320,855.00
9	1,202,370.95	716,576.13	147,901.72	337,893.10	9	1,142,224.90	680,767.22	140,472.77	320,984.92
10	1,202,570.80	716,764.60	147,892.05	337,914.15	10	1,142,185.80	680,712.05	140,523.32	320,950.43
11	1,202,647.50	716,720.43	147,919.38	338,007.68	11	1,142,132.05	680,742.53	140,424.75	320,964.77
12	1,202,365.25	716,613.93	147,910.62	337,840.70	12	1,142,333.33	680,909.12	140,520.62	320,903.60
13	1,202,381.82	716,612.87	147,935.03	337,833.92	13	1,142,359.18	680,779.93	140,489.47	321,089.78
14	1,202,332.25	716,491.40	147,911.23	337,929.62	14	1,142,083.78	680,640.53	140,490.43	320,952.82
15	1,202,432.78	716,549.55	147,861.87	338,021.37	15	1,142,218.03	680,819.43	140,482.07	320,916.53
16	1,202,712.70	716,805.90	147,933.50	337,973.30	16	1,142,661.30	681,103.90	140,487.32	321,070.08
17	1,197,182.13	713,506.43	147,282.18	336,393.52	17	1,142,343.83	681,000.30	140,434.58	320,908.95
Average	***1,201,961.83***	***716,343.75***	***147,844.93***	***337,773.16***	***Average***	***1,142,289.52***	***680,854.66***	***140,475.17***	***320,959.69***
90% Load Level: 21,645 BSUs	**Start**	**Stop**	**Interval**	**Duration**	**80% Load Level: 19,240 BSUs**	**Start**	**Stop**	**Interval**	**Duration**
Start-Up/Ramp-Up	23:13:21	23:13:21	0-7	0:00:00	Start-Up/Ramp-Up	23:36:32	23:44:33	0-7	0:08:01
Measurement Interval	23:13:21	23:13:21	8-17	0:00:00	Measurement Interval	23:44:33	23:54:33	8-17	0:10:00
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3	*(60 second intervals)*	All ASUs	ASU-1	ASU-2	ASU-3
0	1,083,229.57	645,608.97	133,185.90	304,434.70	0	962,770.58	573,705.57	118,446.30	270,618.72
1	1,082,297.57	645,007.63	133,132.58	304,157.35	1	961,838.72	573,152.98	118,348.82	270,336.92
2	1,082,153.08	644,970.60	133,117.48	304,065.00	2	962,051.68	573,466.70	118,297.87	270,287.12
3	1,082,418.32	645,120.05	133,165.77	304,132.50	3	962,074.67	573,374.27	118,387.27	270,313.13
4	1,082,086.18	644,881.93	133,103.52	304,100.73	4	961,933.25	573,310.58	118,338.13	270,284.53
5	1,082,100.22	645,018.42	133,067.15	304,014.65	5	961,963.05	573,363.58	118,237.50	270,361.97
6	1,082,307.57	645,227.77	133,014.65	304,065.15	6	962,008.87	573,403.63	118,405.32	270,199.92
7	1,082,458.57	645,194.65	133,131.95	304,131.97	7	961,801.12	573,310.45	118,256.93	270,233.73
8	1,082,250.82	644,938.35	133,185.42	304,127.05	8	962,002.90	573,373.47	118,295.47	270,333.97
9	1,082,270.30	644,948.53	133,086.88	304,234.88	9	961,777.33	573,294.05	118,286.07	270,197.22
10	1,082,379.32	645,144.25	133,136.75	304,098.32	10	962,142.50	573,363.20	118,380.43	270,398.87
11	1,082,364.58	645,122.10	133,135.20	304,107.28	11	961,967.65	573,274.85	118,370.23	270,322.57
12	1,082,252.32	645,073.42	133,014.38	304,164.52	12	961,988.13	573,351.65	118,330.72	270,305.77
13	1,082,510.85	645,139.45	133,241.55	304,129.85	13	961,942.18	573,331.88	118,356.22	270,254.08
14	1,082,118.58	644,867.32	133,076.05	304,175.22	14	961,885.67	573,275.45	118,344.02	270,266.20
15	1,082,266.97	645,016.90	133,153.17	304,096.90	15	961,861.80	573,249.42	118,305.20	270,307.18
16	1,082,410.77	645,184.80	133,010.27	304,215.70	16	962,035.08	573,348.33	118,321.87	270,364.88
17	1,082,318.08	645,091.35	133,098.55	304,128.18	17	962,079.30	573,357.45	118,325.02	270,396.83
Average	***1,082,314.26***	***645,052.65***	***133,113.82***	***304,147.79***	***Average***	***961,968.26***	***573,321.98***	***118,331.52***	***270,314.76***

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

50% Load Level: 12,025 BSUs	Start	Stop	Interval	Duration	10% Load Level: 2,405 BSUs	Start	Stop	Interval	Duration
Start-Up/Ramp-Up	23:57:56	0:05:57	0-7	0:08:01	Start-Up/Ramp-Up	0:17:02	0:25:03	0-7	0:08:01
Measurement Interval	0:05:57	0:15:57	8-17	0:10:00	Measurement Interval	0:25:03	0:35:03	8-17	0:10:00
(60 second intervals)	All ASUs	ASU-1	ASU-2	ASU-3	*(60 second intervals)*	All ASUs	ASU-1	ASU-2	ASU-3
0	601,547.68	358,558.28	73,956.95	169,032.45	0	120,268.68	71,642.45	14,803.15	33,823.08
1	601,455.75	358,438.62	74,008.97	169,008.17	1	120,234.13	71,663.27	14,756.33	33,814.53
2	601,141.33	358,258.12	73,939.32	168,943.90	2	120,250.88	71,729.95	14,765.65	33,755.28
3	601,433.88	358,441.73	73,960.63	169,031.52	3	120,206.05	71,654.42	14,779.45	33,772.18
4	601,394.83	358,447.87	73,961.82	168,985.15	4	120,199.77	71,623.48	14,779.12	33,797.17
5	601,163.93	358,379.43	73,829.08	168,955.42	5	120,267.07	71,654.10	14,784.75	33,828.22
6	601,156.42	358,204.90	73,984.25	168,967.27	6	120,208.33	71,646.07	14,804.00	33,758.27
7	601,287.78	358,325.67	73,969.55	168,992.57	7	120,229.60	71,660.65	14,782.73	33,786.22
8	601,145.02	358,289.25	73,948.35	168,907.42	8	120,282.75	71,705.50	14,782.45	33,794.80
9	601,306.98	358,388.47	73,952.85	168,965.67	9	120,242.07	71,691.42	14,761.82	33,788.83
10	601,021.97	358,180.13	73,944.45	168,897.38	10	120,206.88	71,673.38	14,760.08	33,773.42
11	601,319.37	358,353.45	73,998.47	168,967.45	11	120,262.27	71,687.13	14,790.38	33,784.75
12	601,348.77	358,303.20	73,980.17	169,065.40	12	120,301.77	71,736.02	14,789.62	33,776.13
13	601,418.30	358,406.50	73,986.27	169,025.53	13	120,233.80	71,632.07	14,791.08	33,810.65
14	601,247.48	358,280.03	73,955.58	169,011.87	14	120,174.88	71,614.58	14,798.07	33,762.23
15	601,334.07	358,443.57	73,953.57	168,936.93	15	120,301.37	71,718.95	14,784.12	33,798.30
16	601,089.12	358,305.80	73,902.07	168,881.25	16	120,248.78	71,621.62	14,817.15	33,810.02
17	601,137.00	358,240.73	73,949.70	168,946.57	17	120,261.83	71,674.35	14,777.12	33,810.37
Average	***601,236.81***	***358,319.11***	***73,957.15***	***168,960.55***	***Average***	***120,251.64***	***71,675.50***	***14,785.19***	***33,790.95***

Response Time Ramp Distribution (IOPS) Graph

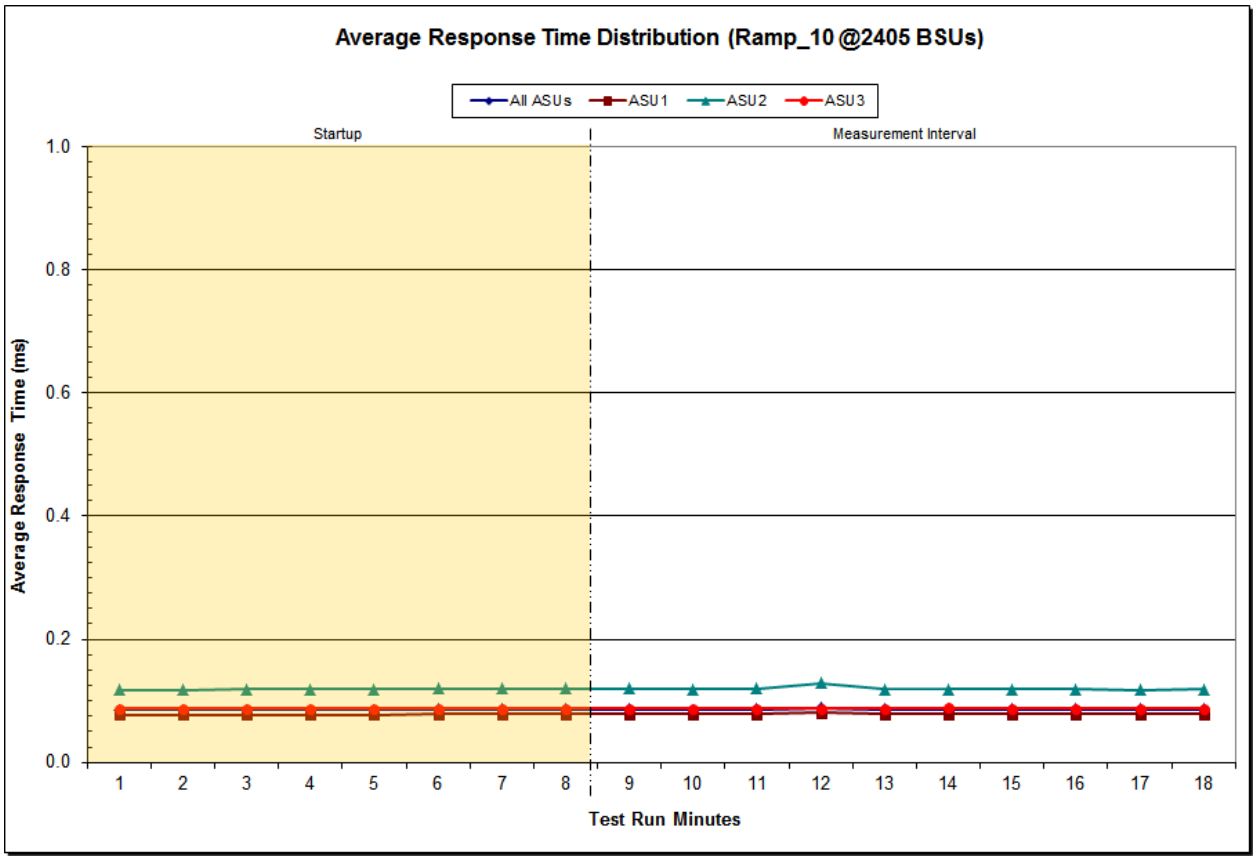


SPC-1 LRT™ Average Response Time (ms) Distribution Data

The SPC-1 LRT™ Average Response Time Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

SPC-1 LRT™ Average Response Time Distribution Data Table

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



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

Clause 3.4.3

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

Clauses 5.1.10 and 5.3.15.2

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

Clause 5.3.15.3

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

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.02810
COV	0.002	0.001	0.001	0.001	0.002	0.002	0.001	0.000

Repeatability Test

Clause 5.4.5

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

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

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

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

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

Clause 9.4.3.7.5

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

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

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 82.

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	***1,201,961.83***
Repeatability Test Phase 1	1,202,397.81
Repeatability Test Phase 2	1,202,384.07

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

	SPC-1 LRT™
Primary Metrics	***0.09***
Repeatability Test Phase 1	0.09
Repeatability Test Phase 2	0.09

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

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

Repeatability Test Phase 1, Test Run 1 (LRT)

Repeatability Test Phase 1, Test Run 2 (IOPS)

Repeatability Test Phase 2, Test Run 1 (LRT)

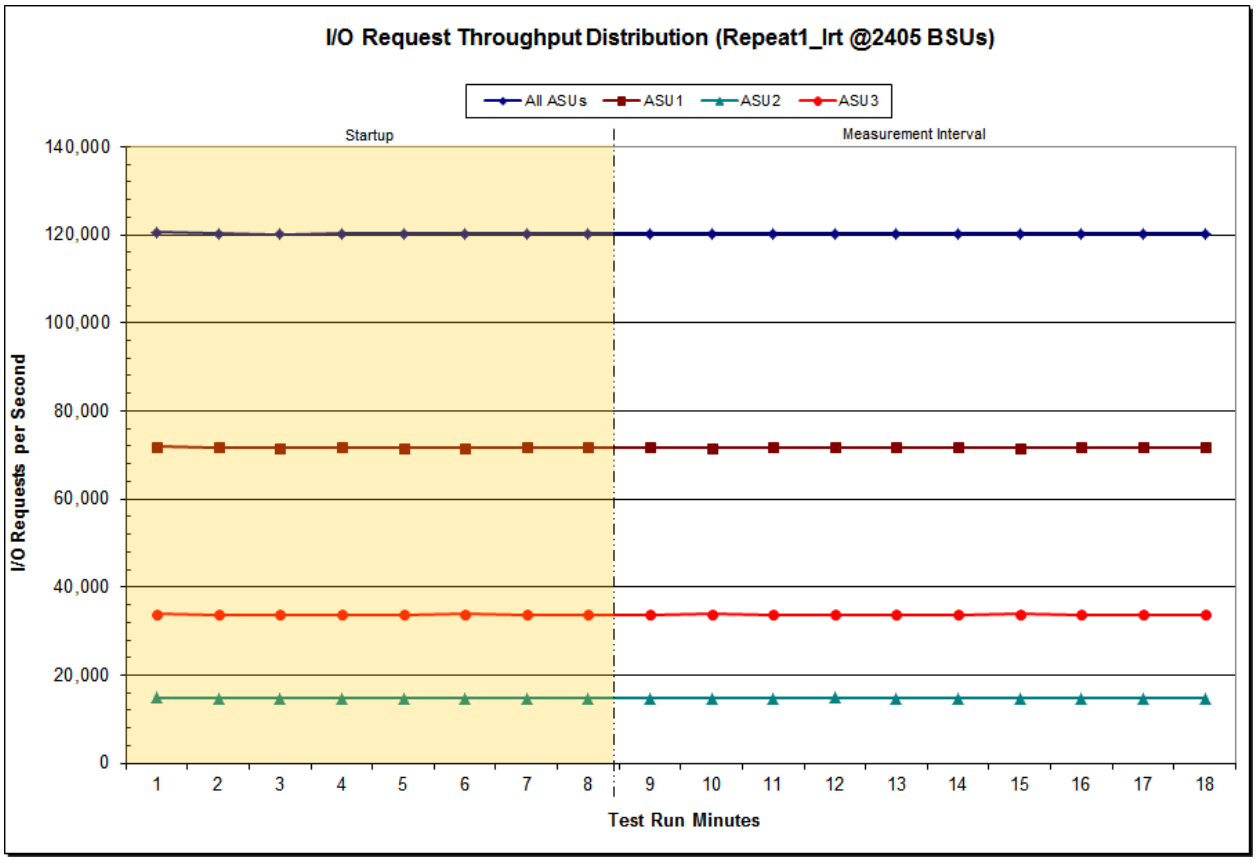
Repeatability Test Phase 2, Test Run 2 (IOPS)

Repeatability 1 LRT – I/O Request Throughput Distribution Data

The Repeatability 1 LRT – I/O Request Throughput Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Repeatability 1 LRT – I/O Request Throughput Distribution Data Table

Repeatability 1 LRT – I/O Request Throughput Distribution Graph

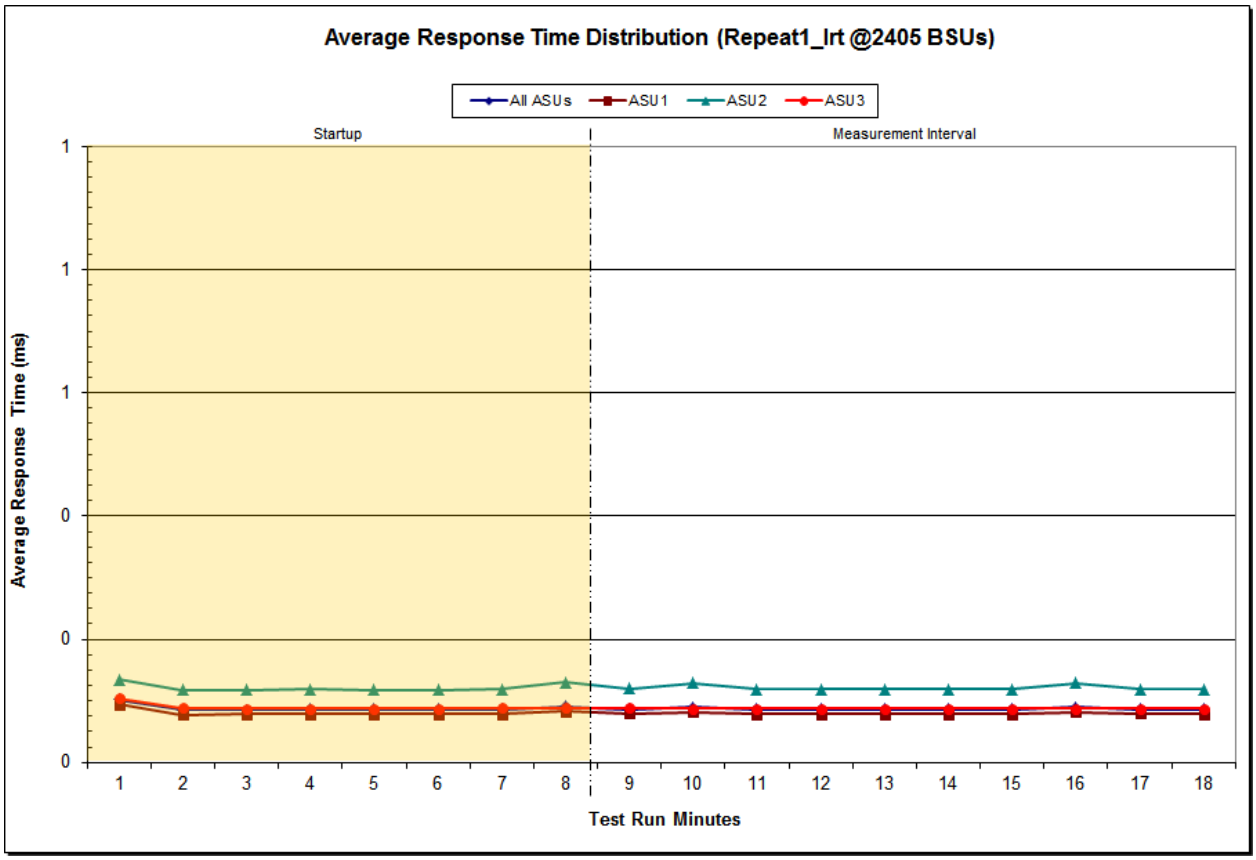


Repeatability 1 LRT –Average Response Time (ms) Distribution Data

The Repeatability 1 LRT – Average Response Time Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Repeatability 1 LRT – Average Response Time Distribution Data Table

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

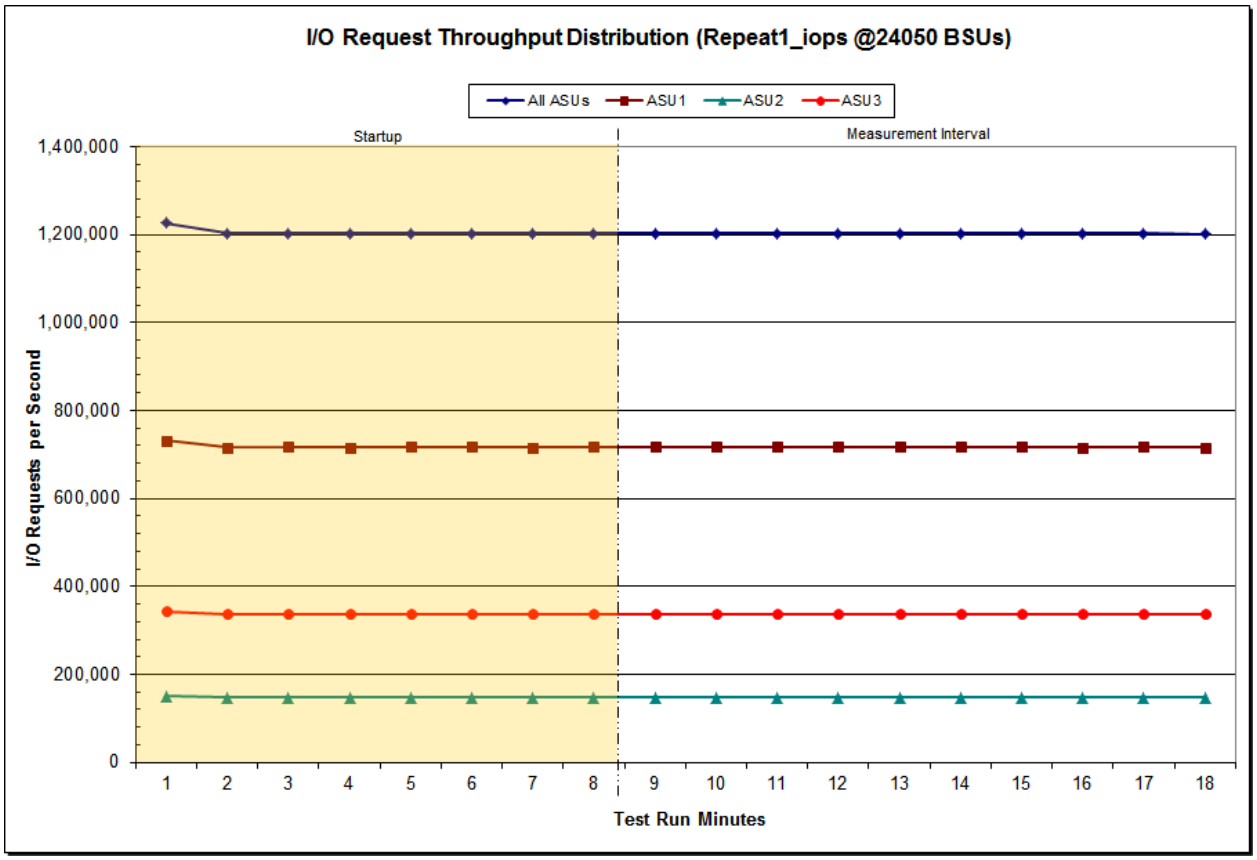


Repeatability 1 IOPS – I/O Request Throughput Distribution Data

The Repeatability 1 IOPS – I/O Request Throughput Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Repeatability 1 IOPS – I/O Request Throughput Distribution Data Table

Repeatability 1 IOPS – I/O Request Throughput Distribution Graph

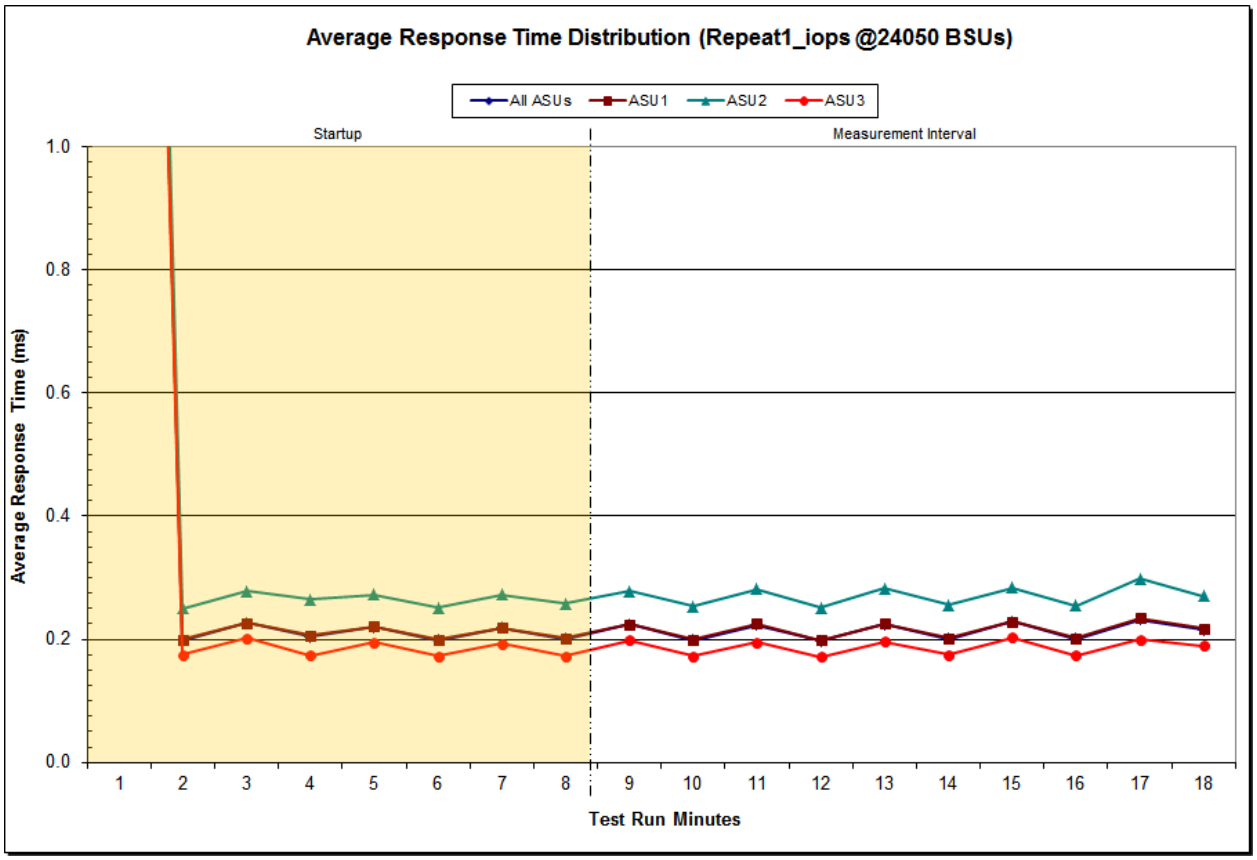


Repeatability 1 IOPS –Average Response Time (ms) Distribution Data

The Repeatability 1 IOPS – Average Response Time Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Repeatability 1 IOPS – Average Response Time Distribution Data Table

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

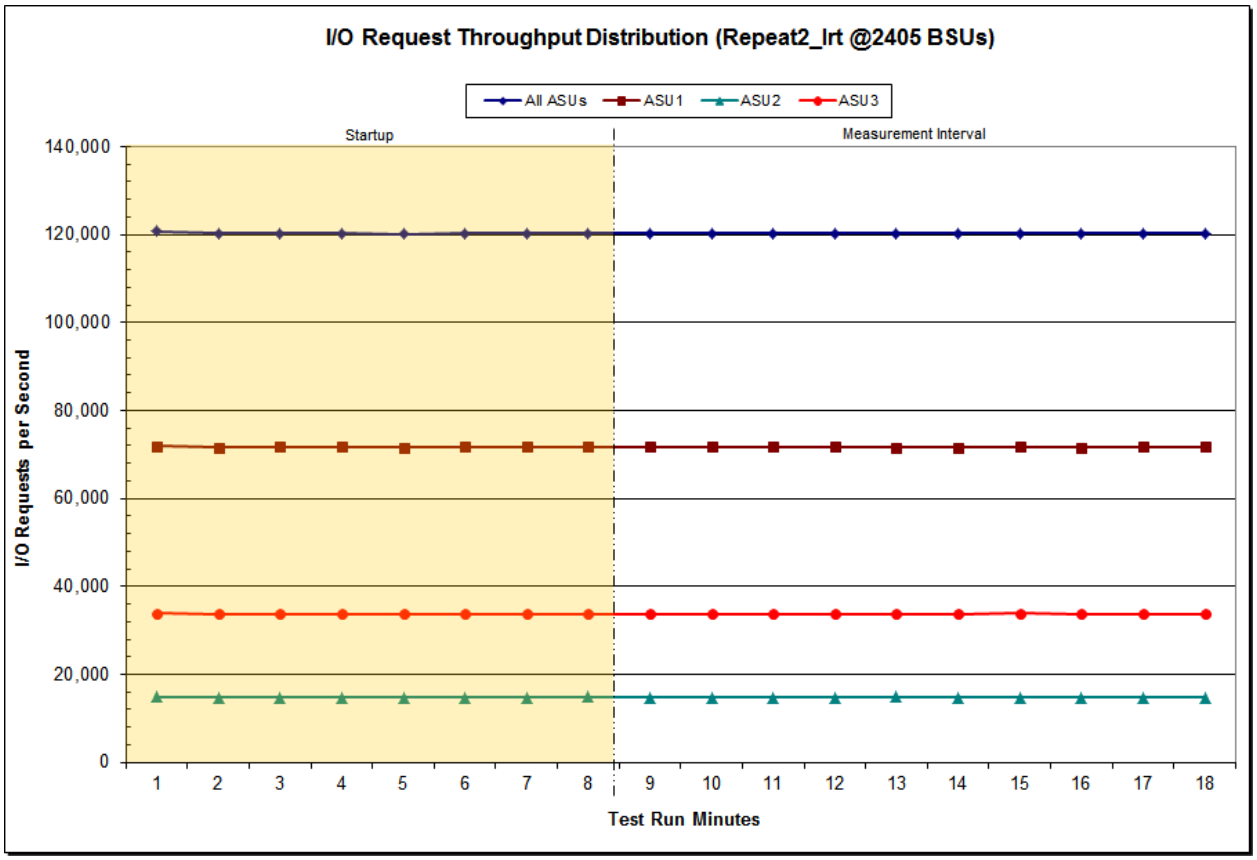


Repeatability 2 LRT – I/O Request Throughput Distribution Data

The Repeatability 2 LRT – I/O Request Throughput Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Repeatability 2 LRT – I/O Request Throughput Distribution Data Table

Repeatability 2 LRT – I/O Request Throughput Distribution Graph

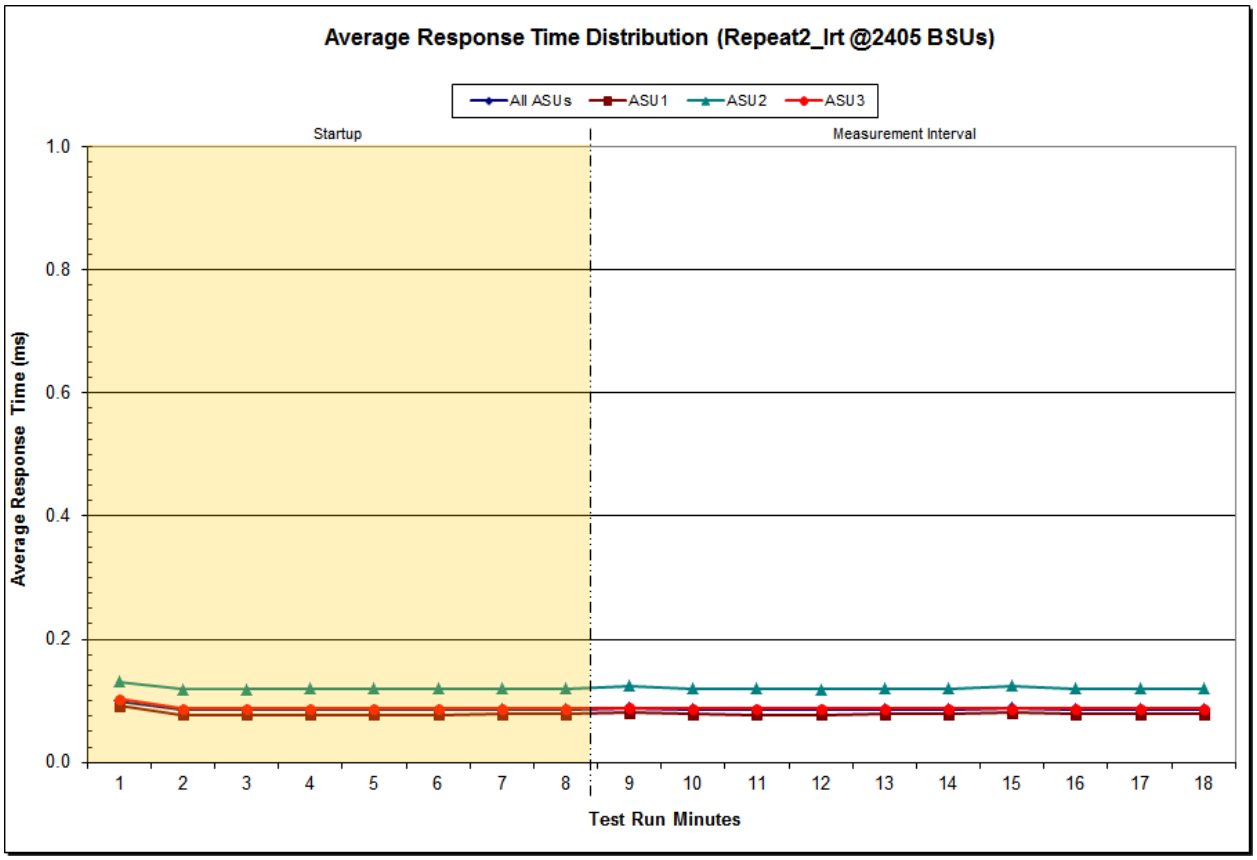


Repeatability 2 LRT –Average Response Time (ms) Distribution Data

The Repeatability 2 LRT – Average Response Time Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Repeatability 2 LRT – Average Response Time Distribution Data Table

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

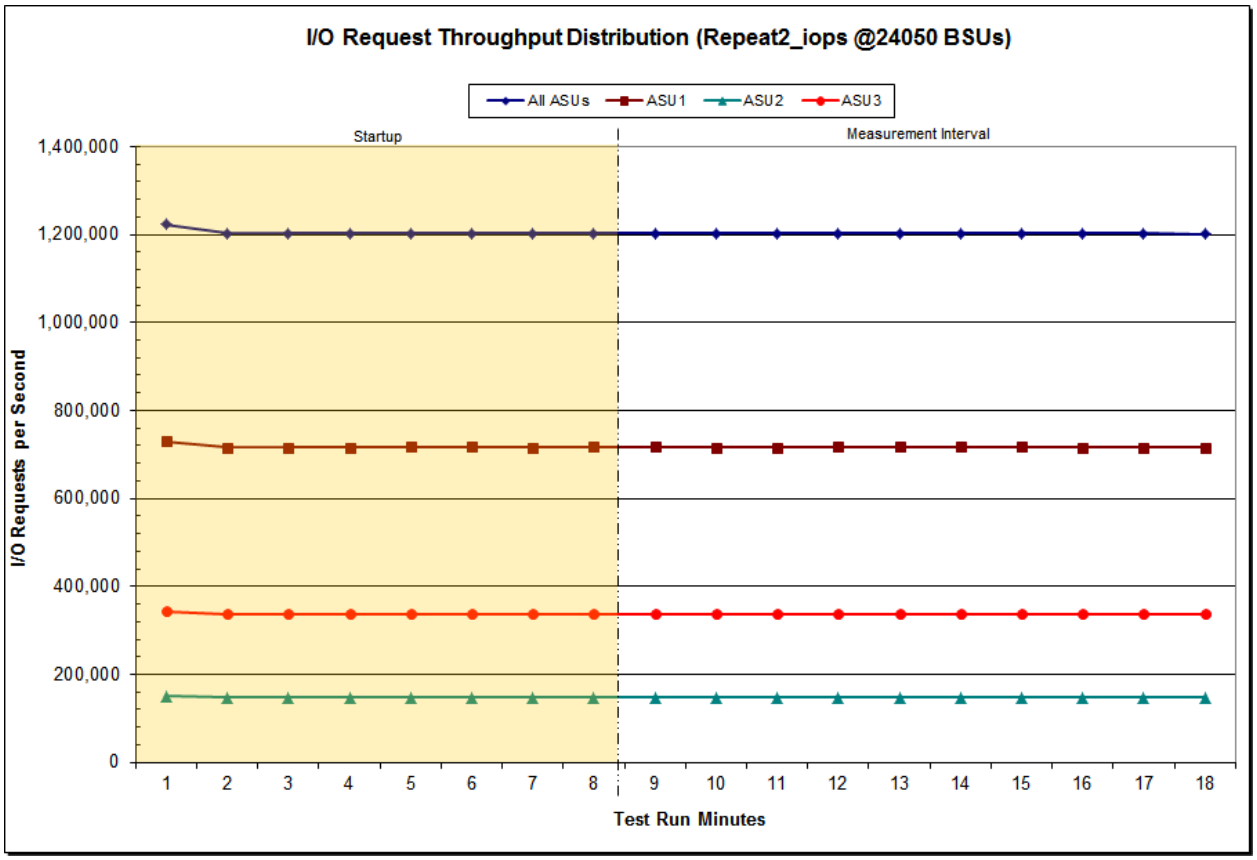


Repeatability 2 IOPS – I/O Request Throughput Distribution Data

The Repeatability 2 IOPS – I/O Request Throughput Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Repeatability 2 IOPS – I/O Request Throughput Distribution Data Table

Repeatability 2 IOPS – I/O Request Throughput Distribution Graph

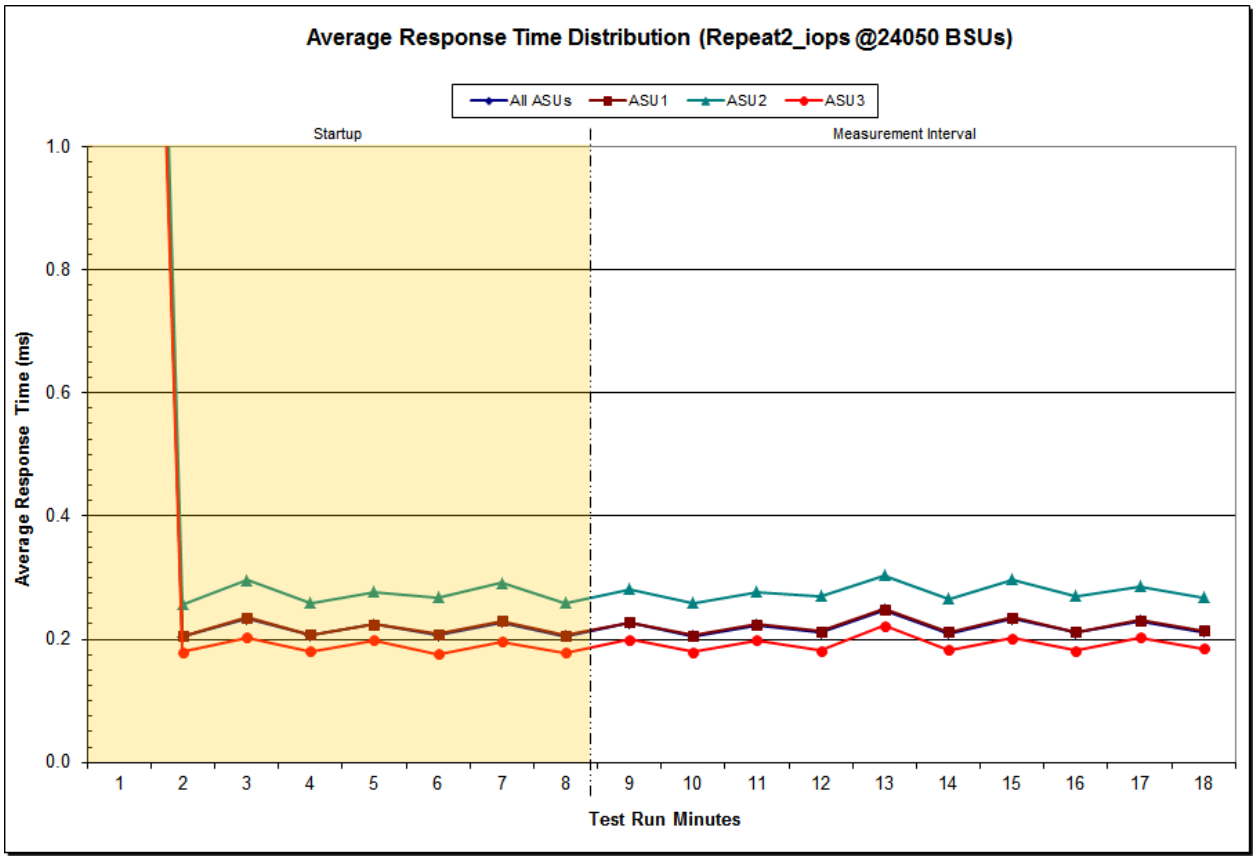


Repeatability 2 IOPS –Average Response Time (ms) Distribution Data

The Repeatability 2 IOPS – Average Response Time Distribution Data table is not embedded in this document due to its size. The table is available via the following URL:

Repeatability 2 IOPS – Average Response Time Distribution Data Table

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



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

Clause 3.4.3

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

Clauses5.1.10 and 5.3.15.2

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

Clause 5.3.15.3

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

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0700	0.2100	0.0180	0.0700	0.0350	0.2810
COV	0.002	0.001	0.001	0.001	0.003	0.002	0.002	0.001

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	0.001	0.000	0.001	0.000	0.000	0.001	0.001	0.000

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

	ASU1-1	ASU1-2	ASU1-3	ASU1-4	ASU2-1	ASU2-2	ASU2-3	ASU3-1
IM	***0.0350***	***0.2810***	***0.0700***	***0.2100***	***0.0180***	***0.0700***	***0.0350***	***0.2810***
MIM	0.0350	0.2810	0.0701	0.2099	0.0180	0.0700	0.0350	0.2810
COV	0.002	0.001	0.001	0.001	0.002	0.001	0.002	0.001

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

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

Data Persistence Test

Clause 6

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

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

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

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

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

Clause 9.4.3.8

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

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

SPC-1 Workload Generator Input Parameters

The SPC-1 Workload Generator input parameters for the Sustainability, IOPS, Response Time Ramp, Repeatability, and Persistence Test Runs are documented in Appendix E: SPC-1 Workload Generator Input Parameters on Page 82.

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	735,311,504
Total Number of Logical Blocks Verified	231,718,480
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 9.4.3.9

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

The entire Priced Storage Configuration, as documented in this Full Disclosure Report, will become available on September 6, 2016 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 17.

TESTED STORAGE CONFIGURATION (TSC) AND PRICED STORAGE CONFIGURATION DIFFERENCES

Clause 9.4.3.3.8

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

A list of all differences between the Tested Storage Configuration (TSC) and Priced Storage Configuration may be found in the Executive Summary portion of this document on page 17.

ANOMALIES OR IRREGULARITIES

Clause 9.4.3.10

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

There were no anomalies or irregularities encountered during the SPC-1 Onsite Audit of the DataCore SANsymphony 10.0.

APPENDIX A: SPC-1 GLOSSARY

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SPC-1 Data Repository Definitions

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

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

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

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

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

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

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

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

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

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

SPC-1 Data Protection Levels

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

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

SPC-1 Test Execution Definitions

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

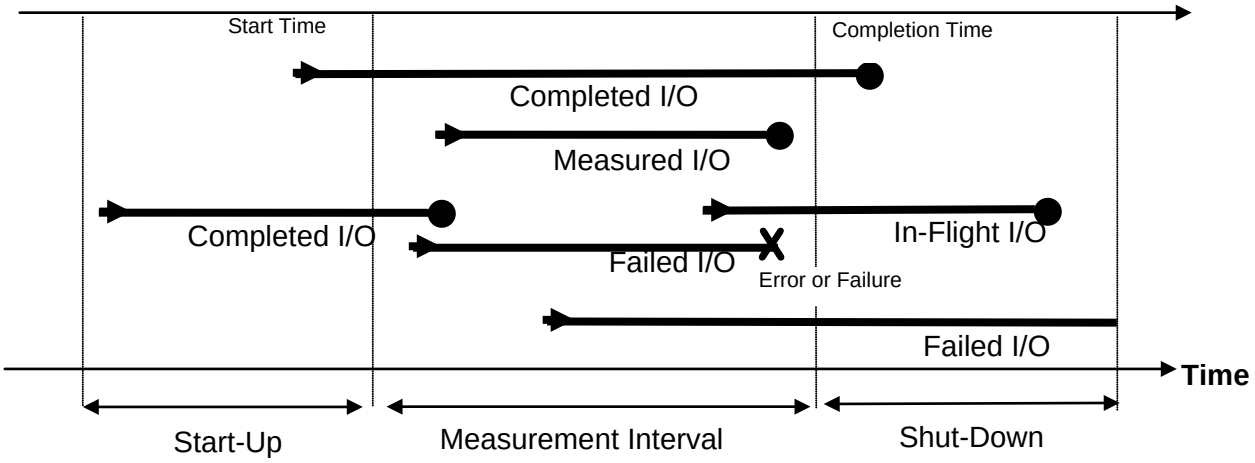
Steady State: The consistent and sustainable throughput of the TSC. During this period the load presented to the TSC by the Workload Generator is constant.

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

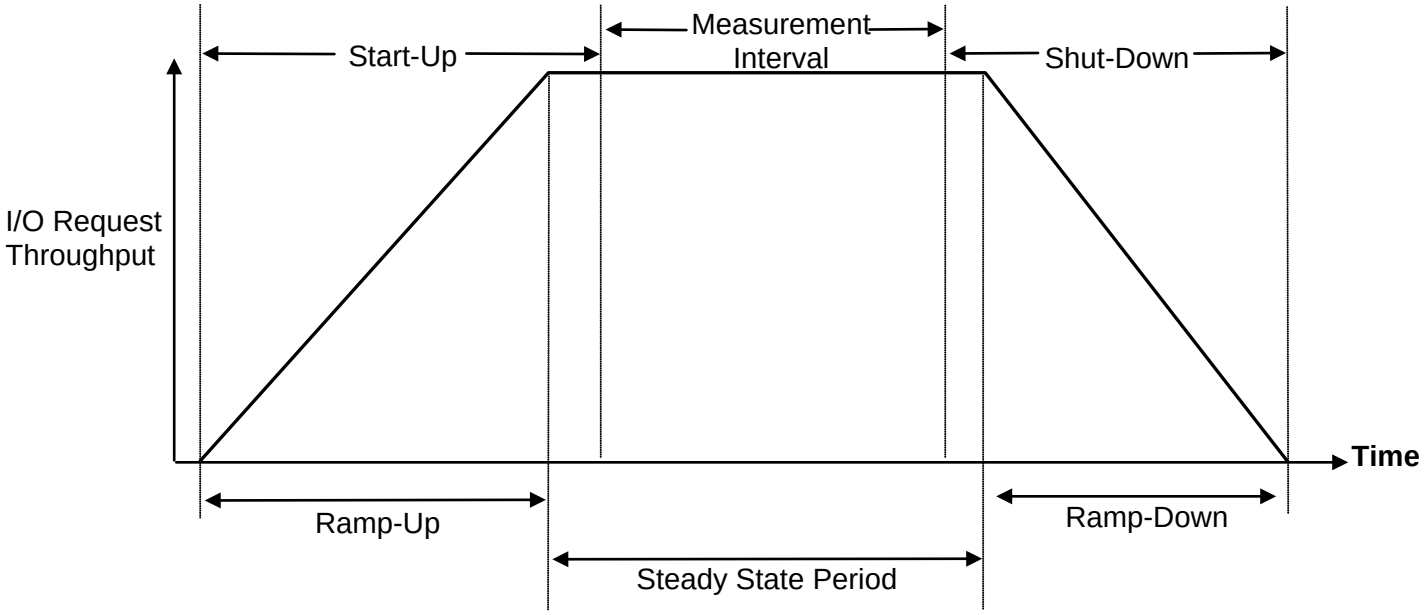
Test Run: The execution of SPC-1 for the purpose of producing or supporting an SPC-1 test result. SPC-1 Test Runs may have a finite and measured Ramp-Up period, Start-Up period, Shut-Down period, and Ramp-Down period as illustrated in the "SPC-1 Test Run Components" below. All SPC-1 Test Runs shall have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

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

I/O Completion Types



SPC-1 Test Run Components



APPENDIX B: CUSTOMER TUNABLE PARAMETERS AND OPTIONS

Windows 2008 Server Registry Settings

The following Windows 2008 Server registry settings were either changed from their default values or added if they did not exist. Settings changed from their default values will have the default value listed in parenthesis and new settings will be annotated with "(new)".

SANsymphony-V Cache Settings for defined Virtual Disks

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsCache\Parameters]
(will be applied to each virtual disk)

- **WriteSizeStop=dword:00000000** ***(00040000)***
  Maximum amount of dirty data (cache memory locations that have changed but not committed to the backend) in the cache.
- **ReadAhead=dword:00000001** ***(00000001 use default)***
  Enable prefetching. Setting means no prefetching
- **IoQueueCount=dword:00000400** ***(00000020)***
  Max number of outstanding IOs to the backend
- **IoQueueSize=dword:00035fff** ***(00002000)***
  Max number of blocks that can be outstanding to the backend
- **AllowRewrite=dword:00000002** ***(00000000)***
  Enable rewrite. When set, an initiator can overwrite dirty data.
- **CmdPoolSize=dword:00010000** ***(new)***
  Number of storage commands that may be simultaneously in progress (global to the entire storage node)
- **MaxLowWaterMark=dword:00000400** ***(new)***
  Point at which the cache is critically short of available blocks and will begin to return busy status to requests.
- **WriteThruLowWaterMark=dword:00010000** ***(new)***
  Point at which the cache will begin to process all write requests by synchronizing them with the backend. (begins writethrough)
- **Workermode=dword:00000004 (new)**

Worker threads for cache processing.

- **WriteQueueCount=dword:000000b4** *(new)*
- **WriteQueueSize=dword:00014054** *(new)*

SANsymphony-V Cache settings for defined specific Virtual Disks

Virtual Disks are constructed of mirrored pairs of logical disks from each of the two DataCore Servers (PServer-A and PServer-E). These registry setting changes must be made on both systems in the appropriate keys as indicated for each server.

PServer-A:
(HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsCache\Parameters\V.{a9c89d1c-f9cb-11e5-852d-98be94f958e3}-00000007

PServer-E:
(HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsCache\Parameters\V.{06b5c27c-f9d4-11e5-a97c-42f2e9afc857}-00000004

ASU3_1=""

- **IoQueueCount=dword:00000001** ***(00000020)***
  Max number of outstanding IOs to the backend
- **IoQueueSize=dword:00075000** ***(00002000)***
  Max number of blocks that can be outstanding to the backend
- **WriteQueueCount=dword:00000001** ***(IoQueueCount)***
  Max number of outstanding writes to the backend
- **WriteQueueSize=dword:00075000** ***(IoQueueSize)***
  Max number of write blocks that can be outstanding to the backend

PServer-A:
(HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsCache\Parameters\V.{a9c89d1c-f9cb-11e5-852d-98be94f958e3}-00000008

PServer-E:
(HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsCache\Parameters\V.{06b5c27c-f9d4-11e5-a97c-42f2e9afc857}-00000005

ASU3_2=""

- **IoQueueCount=dword:00000001** ***(00000020)***
  Max number of outstanding IOs to the backend
- **IoQueueSize=dword:00075000** ***(00002000)***
  Max number of blocks that can be outstanding to the backend
- **WriteQueueCount=dword:00000001** ***(IoQueueCount)***
  Max number of outstanding writes to the backend
- **WriteQueueSize=dword:00075000** ***(IoQueueSize)***
  Max number of write blocks that can be outstanding to the backend

PServer-A:
(HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsCache\Parameters\V.{a9c89d1c-f9cb-11e5-852d-98be94f958e3}-00000009

PServer-E:
(HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsCache\Parameters\V.{06b5c27c-f9d4-11e5-a97c-42f2e9afc857}-00000006

ASU3_3=""

- **IoQueueCount=dword:00000001** ***(00000020)***
  Max number of outstanding IOs to the backend
- **IoQueueSize=dword:00075000** ***(00002000)***
  Max number of blocks that can be outstanding to the backend
- **WriteQueueCount=dword:00000001** ***(IoQueueCount)***
  Max number of outstanding writes to the backend
- **WriteQueueSize=dword:00075000** ***(IoQueueSize)***
  Max number of write blocks that can be outstanding to the backend

SANsymphony-V Poller settings

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsPoll\Parameters

- **LoadHighWaterMark=dword:00000002** ***(00000028)***
  Determines when a new scheduler instance is spawned.
- **LoadLowWaterMark=dword:00000001** ***(0000000a)***
  Determines when a scheduler instance is retired.
- **MinPollers=dword:00000034** ***(00000002) (new)*** [1]
  Minimum number of schedulers
- **CpuAffinity=dword:00000002** ***(00000000) (new)*** [1]
  Hint that determines scheduler to CPU affinity. (1st on CPU1, 2nd on CPU2, etc.)
- **MaxPollers=dword:00000034** ***(0000000a) (new)*** [1]
  Allows max pollers to increase to 34 and addition of logical Ports to increase to 34

SANsymphony-V settings for Pools

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsPool\Parameters

- **DisableAutoTrim=dword:00000001** ***(00000000) (new)*** [1]
  Disable Auto Trim Storage Allocation Units are not reclaimed automatically when filled with zeros.
- **EnableAsyncIo=dword:00000000 (new)**

  Enables the use of system worker threadsfor issuing IOs to a physical disk.

SANsymphony-V settings for specific Pools

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DcsPool\Parameters\KickWorkerOnIoCompletion

(Set per pool by specifying the Pool's GUID, Notify scheduler on physical disk IO completion.)

- **{b0027ac9-0e09-11e6-a03e-98be94f958e3}=dword:00000001** ***(00000020)*** ***PSA_SSD_Pool1*** *(Side 1)*
- **{ab988879-0e09-11e6-a77c-40f2e9afc853}=dword:00000001 (00000020)** ***PSE_SSD_Pool1*** *(Side 2)*
- **{b0027ace-0e09-11e6-a03e-98be94f958e3}=dword:00000001** ***(00000020)*** ***PSA_SSD_Pool2****, (Side 1)*
- **{ab98887e-0e09-11e6-a77c-40f2e9afc853}=dword:00000001 (00000020)** ***PSE_SSD_Pool2*****,** *(Side 2)*
- **{b0027ad2-0e09-11e6-a03e-98be94f958e3}=dword:00000001** ***(00000020)*** ***PSA_SSD_Pool3****, (Side 1)*
- **{ab988882-0e09-11e6-a77c-40f2e9afc853}=dword:00000001 (00000020)** ***PSE_SSD_Pool3,*** *(Side 2)*
- **{b0027cf4-0e09-11e6-a03e-98be94f958e3}=dword:00000001** ***(00000020)*** ***PSA_SSD_Pool4****, (Side 1)*
- **{ab988886-0e09-11e6-a77c-40f2e9afc853}=dword:00000001 (00000020)** ***PSE_SSD_Pool4*****,** *(Side 2)*
- **{b0027ada-0e09-11e6-a03e-98be94f958e3}=dword:00000001** ***(00000020)*** ***PSA_SSD_Pool5****, (Side 1)*
- **{ab98888a-0e09-11e6-a77c-40f2e9afc853}=dword:00000001 (00000020)** ***PSE_SSD_Pool5,*** *(Side 2)*
- **{b0027ade-0e09-11e6-a03e-98be94f958e3}=dword:00000001** ***(00000020)*** ***PSA_SSD_Pool6****, (Side 1)*
- **{ab98888e-0e09-11e6-a77c-40f2e9afc853}=dword:00000001 (00000020)** ***PSE_SSD_Pool6*****,** *(Side 2)*

NOTE:

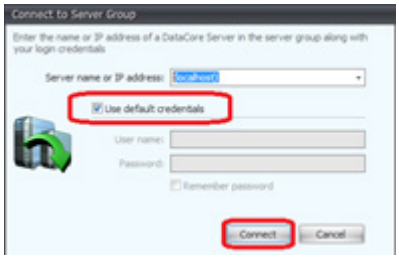
[1] On initial inspection using the registry editor, these registry entries will be absent. In their absence, the software will use the default values (given in parenthesis). The user must add those entries with the values documented to override these defaults.

APPENDIX C: TESTED STORAGE CONFIGURATION (TSC) CREATION

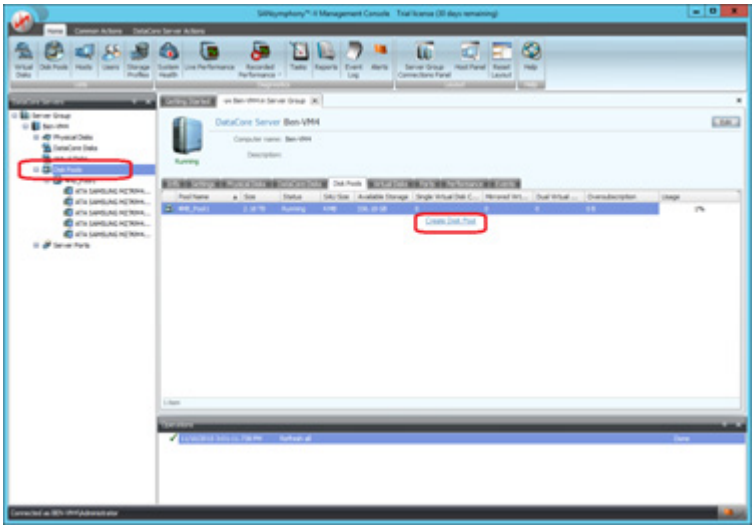
1. **Install DataCore SANsymphony 10.0 Software:**
   The software installation is performed by running a self-extracting executable file.

   After installation, the DataCore Console can be accessed from the Windows "start" menu or by using the desktop shortcut with the DataCore logo. Online help for using the DataCore Console is located at http://www.datacore.com/SSV-Webhelp/ *(refer the help topic, SANSymphony-V Management Console for more information).*

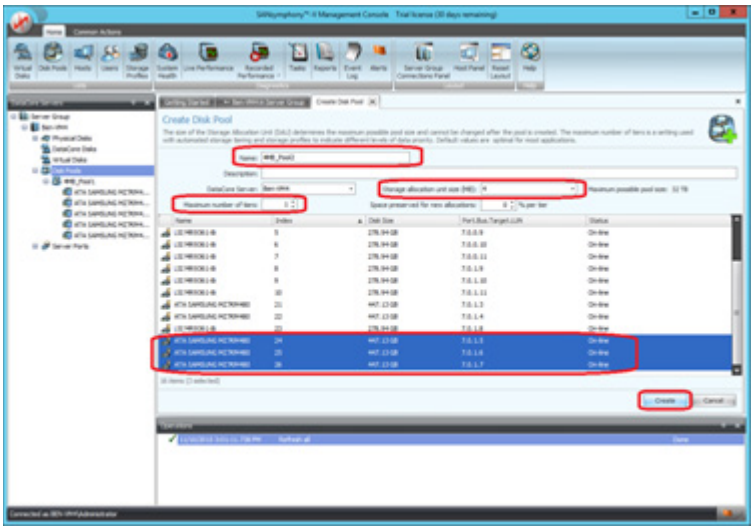
2. Open and log in to the DataCore Console using the following steps. At the end of this step, you will be logged into the management and configuration console:
   a. On the system desktop, double-click on the "SANsymphony" icon to start the DataCore Console.
   b. Select the "Use default credentials checkbox and click Connect to proceed to the DataCore Console:



3. Create pools from managed physical disk resources with the following steps:
   a. In the DataCore Console, click **Disk Pools** under the appropriate Server instance (either PServer-A or PServer-E as indicated in the Server column of the table in step 3.b) in the left panel labeled **DataCore Servers**, then click **Create Disk Pool** in the main panel to initiate the next step.



b. On the subsequent screen, in the main panel, complete the **Name:**, **Storage allocation unit size (MB):**, set the **Maximum number of tiers:** to 1, and select the proper type and amount of physical disks in accordance with the table listed below the following screenshot. Click the **Create** button to create the pool after selecting the disks. *(Note: **4MB_Pool1** has been created as an example).* Repeat this step for each of the 14 pools that are to be created:

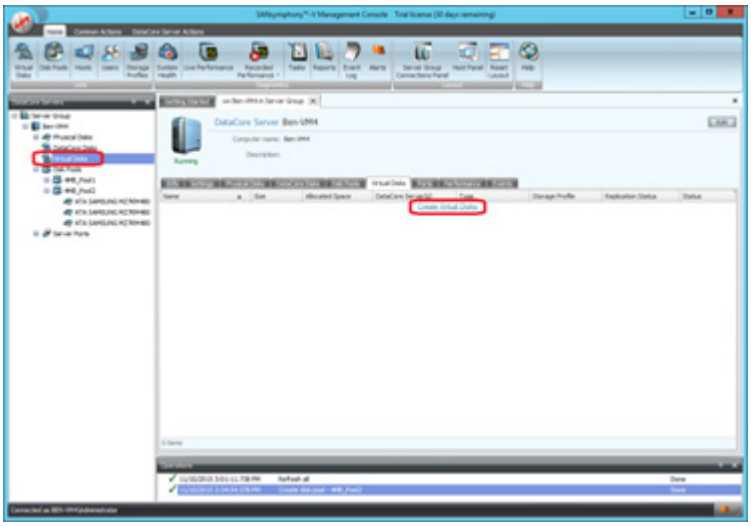


Pool Number	Server	PoolName	Storage Allocation Unit Size(MB)	Number of physical Disks to select (in lower panel)	Type of Physical Disk (from "Name" column in lower panel)
1	PServer-A	PsA_SSD_Pool1	4	3	ATA SAMSUNG MZ7KM240
2	PServer-A	PsA_SSD_Pool2	4	3	ATA SAMSUNG MZ7KM240
3	PServer-A	PsA_SSD_Pool3	4	3	ATA SAMSUNG MZ7KM240
4	PServer-A	PsA_SSD_Pool4	4	3	ATA SAMSUNG MZ7KM240
5	PServer-A	PsA_SSD_Pool5	4	3	ATA SAMSUNG MZ7KM240
6	PServer-A	PsA_SSD_Pool6	4	3	ATA SAMSUNG MZ7KM240
7	PServer-A	PsA_HDD_Pool1	32	4	HGST HUC156030CSS200
8	PServer-E	PsE_SSD_Pool1	4	3	ATA SAMSUNG MZ7KM240
9	PServer-E	PsE_SSD_Pool2	4	3	ATA SAMSUNG MZ7KM240
10	PServer-E	PsE_SSD_Pool3	4	3	ATA SAMSUNG MZ7KM240
11	PServer-E	PsE_SSD_Pool4	4	3	ATA SAMSUNG MZ7KM240
12	PServer-E	PsE_SSD_Pool5	4	3	ATA SAMSUNG MZ7KM240
13	PServer-E	PsE_SSD_Pool6	4	3	ATA SAMSUNG MZ7KM240
14	PServer-E	PsE_HDD_Pool1	32	4	HGST HUC156030CSS200

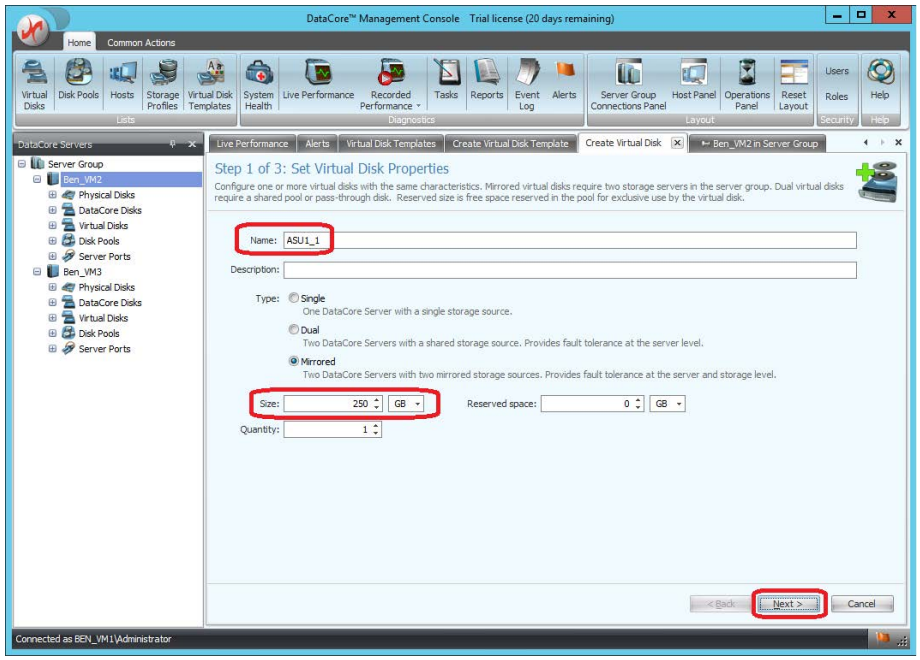
4. Create 15 Mirrored Virtual Disks repeating steps **4a** through **4d** 15 times, once for each of the virtual disks to be created, using the details in the following table. The virtual disks created in this step will be mapped and used to create the SPC-1 Logical Volumes that comprise the SPC-1 ASUs.

Virtual Disk Number	Name:	Size: (GB)	Source Pool 1:	Source Pool 2:
1	ASU1_1	250	PsA_SSD_Pool1	PsE_SSD_Pool1
2	ASU1_2	250	PsA_SSD_Pool2	PsE_SSD_Pool2
3	ASU1_3	250	PsA_SSD_Pool3	PsE_SSD_Pool3
4	ASU1_4	250	PsA_SSD_Pool4	PsE_SSD_Pool4
5	ASU1_5	250	PsA_SSD_Pool5	PsE_SSD_Pool5
6	ASU1_6	250	PsA_SSD_Pool6	PsE_SSD_Pool6
7	ASU2_1	250	PsA_SSD_Pool1	PsE_SSD_Pool1
8	ASU2_2	250	PsA_SSD_Pool2	PsE_SSD_Pool2
9	ASU2_3	250	PsA_SSD_Pool3	PsE_SSD_Pool3
10	ASU2_4	250	PsA_SSD_Pool4	PsE_SSD_Pool4
11	ASU2_5	250	PsA_SSD_Pool5	PsE_SSD_Pool5
12	ASU2_6	250	PsA_SSD_Pool6	PsE_SSD_Pool6
13	ASU3_1	150	PsE_HDD_Pool1	PsA_HDD_Pool1
14	ASU3_2	150	PsE_HDD_Pool1	PsA_HDD_Pool1
15	ASU3_3	150	PsE_HDD_Pool1	PsA_HDD_Pool1

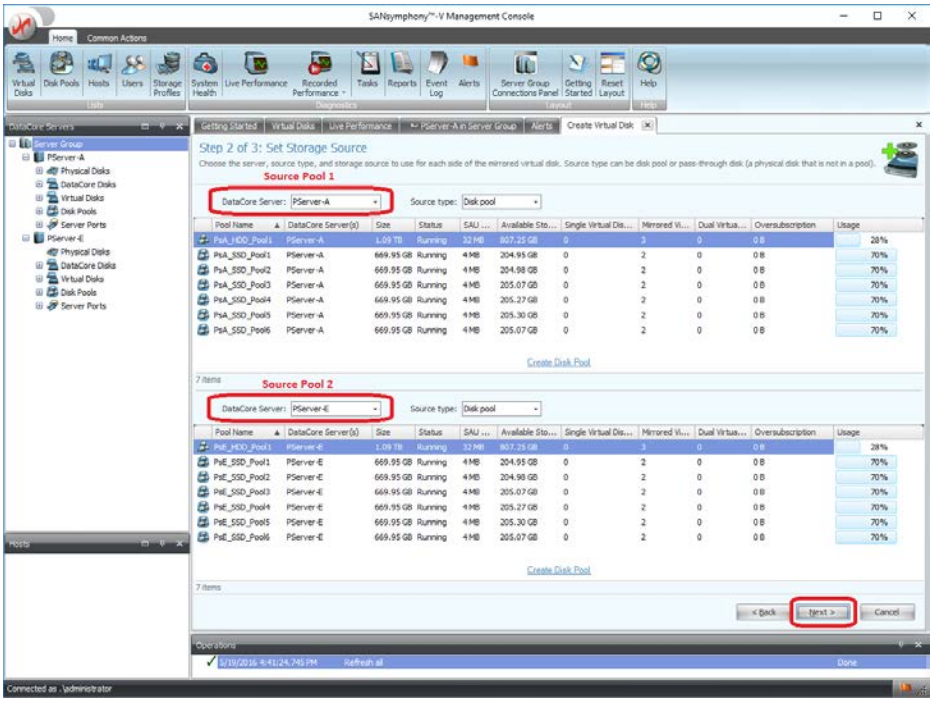
a. In the DataCore Console, click **Virtual Disks** in the left panel labeled **DataCore Servers**, then click **Create Virtual Disks** in the main panel.



b. On the subsequent screen, complete the **Name:** and the **Size (GB):** as described in the table above. Click the **Next** button to proceed to the next step. *(Note: **ASU1_1** is being created as an example).*



c. On the subsequent screen, set the storage sources by clicking on the appropriate entry *(Refer to the above table in step 4.)* under the **Source Pool** columns labeled **1** and **2** and proceed to the next step by clicking the **Next** button,



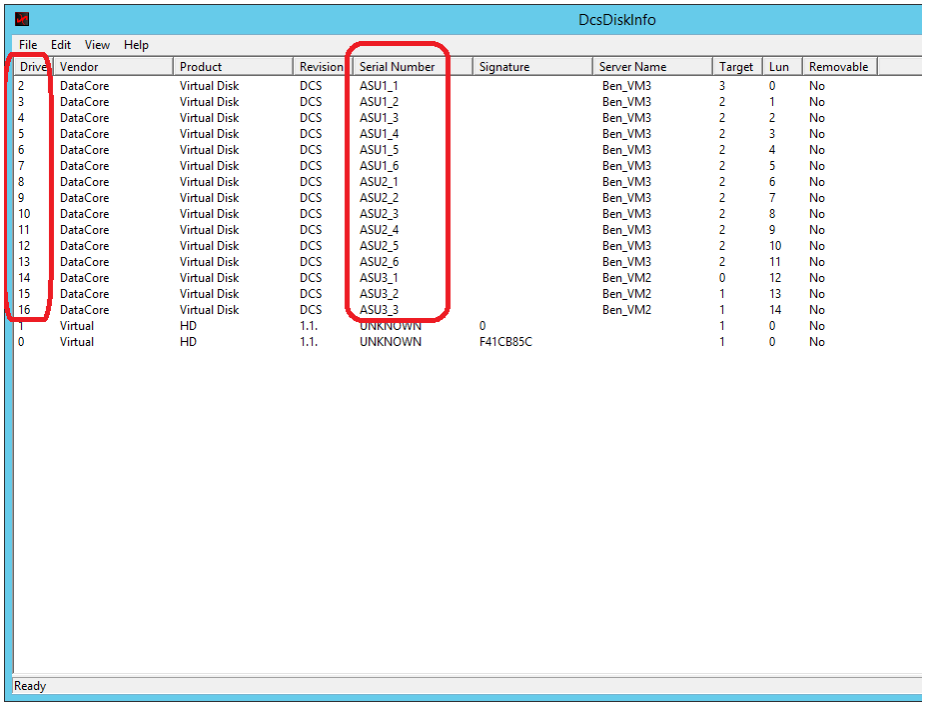
5. Use the Windows 2008 Server registry editor to make the changes documented in *Appendix B: Customer Tunable Parameters and Options* on page 66.

6. Double-click the desktop icon labeled "SANsymphony-V cmdlet shell to open it and execute the script **Map_FrontEnd_DPS.ps1**. This script maps the virtual disks created in step 4 to the host via the logical ports.

7. On host **Pserver-A**, use the following steps to create disk stripes:

   a. From the **C:\ProgramFiles\DataCore\SANsymphony** directory on **Pserver-A** or **Pserver-E** copy **DcsDiskInfo.exe** and **DcsMakeClusterDisk.exe** to the desktop.

   b. From the desktop, execute **DcsDiskInfo.exe** by double clicking it. This utility will open a window to display a list of all disks detected on the system.



   c. Find the 15 virtual disk names (in the **Serial Number** column) given in step 4 and note the Drive number (in the **Drive** column) for each of the logical disks used in the benchmark.

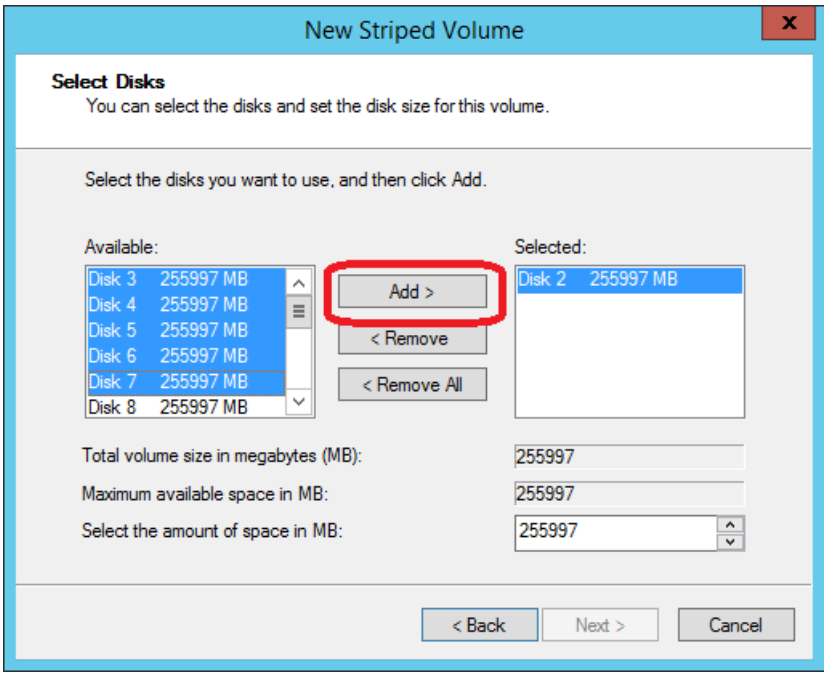
d. From the Windows start menu, open Windows Disk Management.

i. "online" all disks identified in step **7b**.

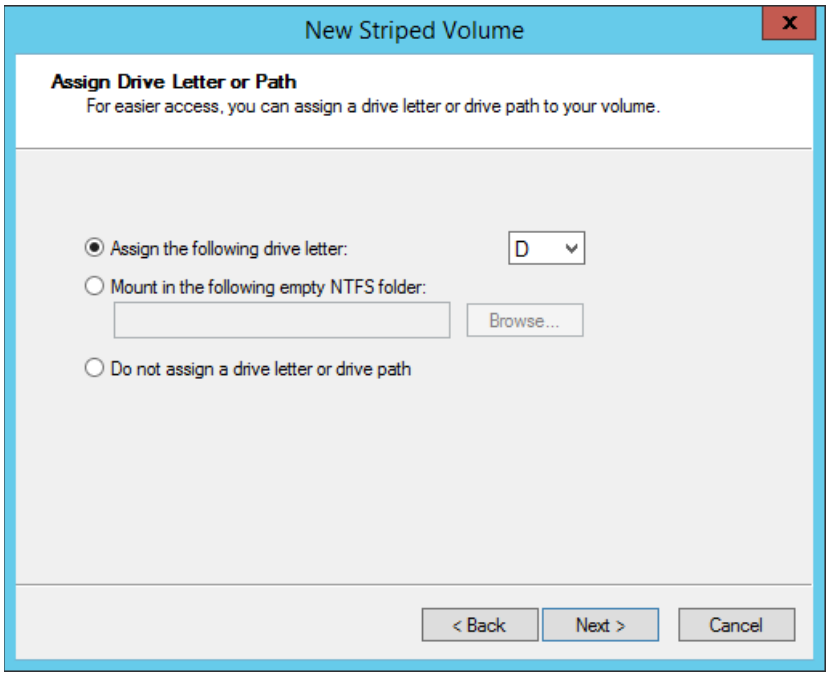
ii. Pick one of those disks "onlined" in the previous step and "initialize" as GPT, all 15 disks.

iii. Pick the lowest numbered disk of those "onlined" in the previous step for ASU-1 and select **New Striped Volume**, click **next** so that the **New Striped Volume** dialog box appears. In that dialogue, select the 5 additional available disks identified for ASU-1 in step **7b**. *(The below example is used to select disks for ASU-1.)*

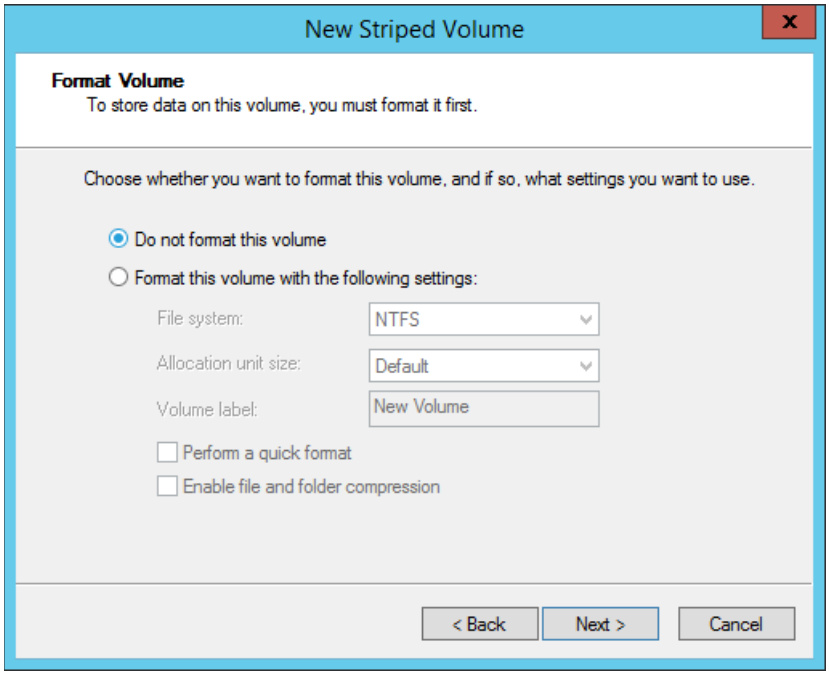
iv. Click the **Add** button, to add the additional "disks". Use all available space, which is indicated in the "**Select the amount of space in MB**" selection and then click the **Next** button which will have become actionable.



v. Assign drive letter **D** in the next dialog. Then click the **Next** button.

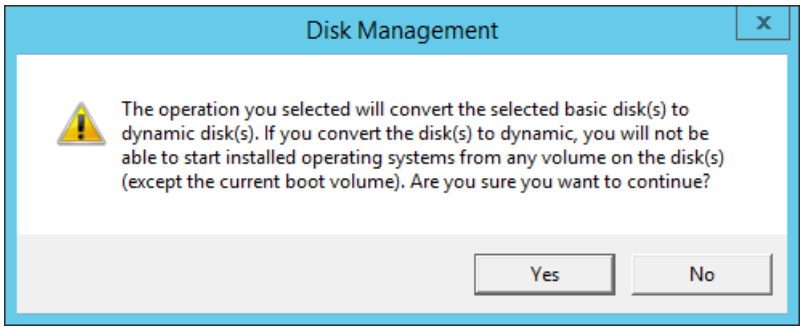


vi. Select **Do not format this volume**. Then click the **Next** button.



vii. Verify the striped volume is correct and click the **Finish** button to complete the creation of the striped volume.

viii. Click **Yes** in the dialog to confirm conversion of the basic disks to dynamic.



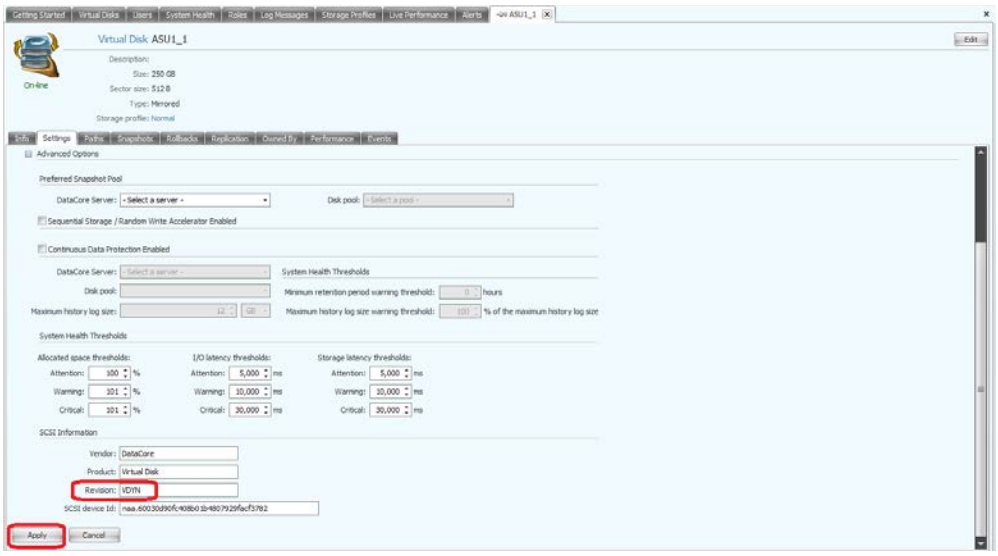
ix. Repeat steps **7.d.iii.-vii.** for ASU-2 and ASU-3 making sure to use letters **E** and **F** in steps **7.d.v**. respectively.

e. The following steps are used to share stripe sets of the 15 disks for the SPC-1 Logical Volumes among the two hosts in benchmark configuration.

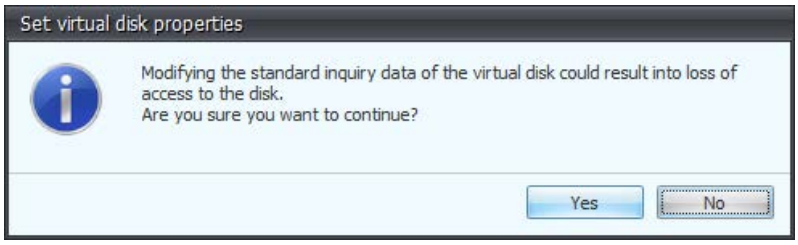
i. Run **cmd.exe** as Administrator to start a CLI session and for each disk identified in step **7b**, run **DcsMakeClusterDisk.exe <*arg*>** where **<*arg*>** is a disk number. This command is executed 15 times using 2-16 as arguments.

ii. In the DataCore Console:

1) Go to the **Details, Settings** tab for a Virtual Disk identified in step **7b**. to change **SCSI Revision** from **DCS** to **VDYN** and then click the **Apply** button. This change allows additional hosts in this TSC to mount the striped drives of the SPC-1 Logical Volumes. The example shown is for ASU1_1.



2) Click the "Yes" button to acknowledge the warning.



3) Repeat steps **7.e ii.1** and **7.e ii.2** for the remaining 14 Virtual Disks (ASU1_2 – ASU3_3).

f. Log into the other server, **Pserver-E**. From the Windows start menu, open **Windows Disk Management**.

i. On the action ribbon, rescan disks to make "foreign" disks appear.

ii. Right click on one of the disks, and click on **Import Foreign Disks**.

iii. Repeat steps 8.f.i.-ii. until all foreign disks are imported.

iv. On the action ribbon, rescan disks.

g. Log back into server **Pserver-A**. From the Windows start menu, open Windows Disk Management.

On the action ribbon, rescan disks to assure that the striped disks (**D:**, **E:** and **F:**) remain visible.

Map_FrontEnd_DPS.ps1

```
Connect-DcsServer


Write-Host "Loopback and FC mappings for all ASU's"

Write-Host "ASU1_1"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU1_1 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU1_1"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU1_1 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU1_2"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU1_2 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU1_2"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU1_2 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU1_3"
```

```
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU1_3 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU1_3"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU1_3 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU1_4"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU1_4 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU1_4"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU1_4 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU1_5"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU1_5 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU1_5"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU1_5 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU1_6"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU1_6 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU1_6"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU1_6 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU2_1"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU2_1 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU2_1"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU2_1 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU2_2"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU2_2 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU2_2"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU2_2 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU2_3"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU2_3 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU2_3"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU2_3 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU2_4"
```

```
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU2_4 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU2_4"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU2_4 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU2_5"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU2_5 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU2_5"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU2_5 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU2_6"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU2_6 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU2_6"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU2_6 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10


Write-Host "ASU3_1"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU3_1 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU3_1"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU3_1 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU3_2"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU3_2 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU3_2"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU3_2 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Write-Host "ASU3_3"
Serve-DcsVirtualDisk -Machine Pserver-A -VirtualDisk ASU3_3 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-A

Write-Host "ASU3_3"
Serve-DcsVirtualDisk -Machine Pserver-E -VirtualDisk ASU3_3 -InitiatorPort "Loopback
Port 1" -TargetPort "Loopback Port 1" -server Pserver-E
Start-Sleep -Seconds 10

Disconnect-DcsServer
```

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

ASU Pre-Fill

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

```
compratio=1
sd=default,threads=8
sd=asu1_1,lun=\\.\D:,size=1497000000000,threads=4
sd=asu2_1,lun=\\.\E:,size=1497000000000,threads=4
sd=asu3_1,lun=\\.\F:,size=332000000000,threads=4
wd=default,rdpct=0,seek=-1,xfersize=512k
wd=wd1,sd=asu1_1
wd=wd2,sd=asu2_1
wd=wd3,sd=asu3_1
rd=asuprefill,wd=wd*,iorate=max,elapsed=24h,interval=60
```

Primary Metrics, and Repeatability Tests

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

```
javaparms="-Xmx512m -XX:+UseG1GC"
host=master
slaves=(h1,h122,h2,h123,h3,h124,h4,h125,h5,h126,h6,h127,h7,h128,h8,h129,h9,h130,h10,
h131,h11,h132,h12,h133,h13,h134,h14,h135,h15,h136,h16,h137,h17,h138,h18,h139,h19,h14
0,h20,h141,h21,h142,h22,h143,h23,h144,h24,h145,h25,h146,h26,h147,h27,h148,h28,h149,h
29,h150,h30,h151,h31,h152,h32,h153,h33,h154,h34,h155,h35,h156,h36,h157,h37,h158,h38,
h159,h39,h160,h40,h161,h41,h162,h42,h163,h43,h164,h44,h165,h45,h166,h46,h167,h47,h16
8,h48,h169,h49,h170,h50,h171,h51,h172,h52,h173,h53,h174,h54,h175,h55,h176,h56,h177,h
57,h178,h58,h179,h59,h180,h60,h181,h61,h182,h62,h183,h63,h184,h64,h185,h65,h186,h66,
h187,h67,h188,h68,h189,h69,h190,h70,h191,h71,h192,h72,h193,h73,h194,h74,h195,h75,h19
6,h76,h197,h77,h198,h78,h199,h79,h200,h80,h201,h81,h202,h82,h203,h83,h204,h84,h205,h
85,h206,h86,h207,h87,h208,h88,h209,h89,h210,h90,h211,h91,h212,h92,h213,h93,h214,h94,
h215,h95,h216,h96,h217,h97,h218,h98,h219,h99,h220,h100,h221,h101,h222,h102,h223,h103
,h224,h104,h225,h105,h226,h106,h227,h107,h228,h108,h229,h109,h230,h110,h231,h111,h23
2,h112,h233,h113,h234,h114,h235,h115,h236,h116,h237,h117,h238,h118,h239,h119,h240,h1
20,h241,h121)
sd=asu1_1,lun=\\?\D:,size=1497g
sd=asu2_1,lun=\\?\E:,size=1497g
sd=asu3_1,lun=\\?\F:,size=332g
```

SPC-1 Persistence Test

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

```
javaparms="-Xmx512m -XX:+UseG1GC"
sd=asu1_1,lun=\\?\D:,size=1497g
sd=asu2_1,lun=\\?\E:,size=1497g
sd=asu3_1,lun=\\?\F:,size=332g
```

APPENDIX E: SPC-1 WORKLOAD GENERATOR INPUT PARAMETERS

The first 'master' script, **RunSPC1.bat**, was executed to:

- Dynamically determine the number of CPUs in each NUMA node and specific affinity mask based on the benchmark configuration (**:setcpus**). A value of 6 CPUs was used for this benchmark execution.
- Set various benchmark-related environment variables (**config.bat**).
- Generate the required SPC-1 Workload Generator configuration files required for the Test Run (**:config**, **:configure**, **:discover**, **:slavecfg**, **:updatecfg**).
  - The appropriate 'master' **spc1.cfg** files for the benchmark execution. *(The same file is used for the Primary Metrics and Repeatability Tests (multiple Slave JVMs) and a separate file for the SPC-1 Persistence Test (one Slave JVM)*
  - A configuration file, **spc1.cfg**, for each Slave JVM, which is derived from the appropriate 'master' configuration file.
- Using the appropriate arguments, execute the following the following in an uninterrupted sequence:
  - Terminate any pre-existing Slave JVMs prior to executing the ASU pre-fill or SPC-1 Test Run (**:killjvms**).
  - Start the required number of Slave JVMs and bind each Slave JVM to a set of CPUs located on the same NUMA node (**:startjvms**, **:startlocal**, **:setmask**).
  - Activate the appropriate 'master' configuration file, **spc1.cfg** (**:mastercfg**).
  - Execute the following, based invocation arguments (**:benchmark**):
    - Generate the first set of storage configuration information required for audit (**:sysinfo**).
    - The required ASU pre-fill (**:prefill**).
    - The Primary Metrics Test
      (Sustainability Test Phase, IOPS Test Phase, and Response Time Ramp Test Phase).
    - Terminate and restart the Slave JVMs prior to each Repeatability Test Phase (**:restartjvms**).
    - The Repeatability Test
      (Repeatability Test Phase 1 and Repeatability Test Phase 2).
    - Terminate the Slave JVM used in the prior Tests (**:killjvms**)
    - SPC-1 Persistence Test Run 1 *(write phase)* (**:persist1**).
      *(The **spc2persist** variable in **RunSPC1.bat** was initialized as " ", meaning null value, so the SPC-2 Persistence Test was not used as a substitute for the maximum BSU level SPC-1 Persistence Test. The SPC-1 Persistence Test Run 1 (write phase) was executed with the maximum BSU level specified of which the SPC-1 Workload Generator used 25% of that value.)*
    - Create the second 'master' script, **Persist2.bat**.

The second 'master' script, **Persist2.bat**, is executed after completion of the required TSC power off/power on cycle to:

- Invoke, **RunSPC1.bat**, with the appropriate arguments to execute the SPC-1 Persistence Test Run 2 *(read phase)* (**:persist2**).
- Generate the second set of storage configuration information required for the audit (**:sysinfo**).
- Collect and 'package' all of the output for audit submission (**:package**).

RunSPC1.bat

```
@echo off
@setlocal enabledelayedexpansion

set rate=30200
set runtime=28800
set ramptime=180
set warmuptime=
set hostlist=
set sdslist=
set prefix=
set spc2persist=
set javaparms=-Xmx512m -XX:+UseG1GC
set spc2javaparms=-Xmx1536m -Xms512m -Xss96k
set masterjavaparms=-Xmx10240m
set masteraffinity=
set psauth=-accepteula -u Administrator -p Datacore1
set priority=/low
set setsize=6
set basecpu=
set exec=
set resume=
set bsusperjvm=100
set maxstart=100

cd c:\BenchmarkRun
set rundir=c:\BenchmarkRun
set share=BenchmarkRun

set spc1path=c:\spc\spc1
set spc2path=c:\spc\spc2
set CLASSPATH=%spc1path%

call :setcpus
if "%cpuspernuma%"=="28" set setsize=4
if "%cpuspernuma%"=="32" set setsize=4

set cmd=%0
if exist %cmd%.bat set cmd=%cmd%.bat

set setsizearg=
set javaparmsarg=

set x=%~1
if "%x%"=="-d" (
    set exec=echo
    shift
```

```
)

set x=%~1
if "%x%"=="-r" (
    set resume=true
    shift
)

set /a x=%1+0
if not "%x%"=="0" (
    set setsizearg=%x%
    shift
)

set x=%~1
if "%x:~0,1%"=="-" (
    set javaparmsarg=%x%
    shift
)

set master=%COMPUTERNAME%

if not "%1"=="slave" (
    if "%hostlist%"=="" set hostlist=%master%
    if exist config.bat call config.bat
)

if not "%hostlist%"=="" set hostlist=%hostlist:(= %
if not "%hostlist%"=="" set hostlist=%hostlist:)= %
if not "%hostlist%"=="" set hostlist=%hostlist:,= %

if "%sdslist%"=="" set sdslist=%master%
if not "%sdslist%"=="" set sdslist=%sdslist:(= %
if not "%sdslist%"=="" set sdslist=%sdslist:)= %
if not "%sdslist%"=="" set sdslist=%sdslist:,= %

set dflag=
if "%exec%"=="echo" set dflag=-d

if not "!%master%_setsize!"=="" set setsize=!%master%_setsize!
if not "%setsizearg%"=="" set setsize=%setsizearg%
if not "%javaparmsarg%"=="" set javaparms=%javaparmsarg%

if "%resume%"=="" call :killjvms

if "%1"=="kill" goto :eof
if "%1 %2"=="slave kill" goto :eof

if "%1 %2"=="slave info" (
    call :localinfo %3
    goto :eof
)

set /a setspernuma=cpuspernuma/setsize
set /a v=setspernuma*setsize

if not "%v%"=="%cpuspernuma%" (
    echo %master%: Warning: Per-NUMA CPU count is not a multiple of the set size
)

if not "!%master%_basecpu!"=="" set basecpu=!%master%_basecpu!
set /a v=basecpu"%%"setsize
```

```
if not "%v%"=="0" (
    echo Error: basecpu is not a multiple of the set size
    goto :eof
)

set bsu=0
if "%2"=="-b" set /a bsu=%3
if "%1"=="" set /a bsu=rate
set /a jvms="(bsu+bsusperjvm-1)/bsusperjvm"

if "%1"=="persist1" set /a jvms=0
if "%1"=="persist2" set /a jvms=0
if "%1"=="prefill" set /a jvms=0
if "%hostlist%"=="" set /a jvms=0

call :config %1
if errorlevel 1 goto :error
if "%1"=="config" goto :done

call :setoutput %1 2>&1

if "%1"=="prefill" goto :prefill
if "%1"=="package" goto :package

if "%1"=="slave" (
    call :startlocal %2 %3 %4 %5 2>&1
    goto :eof
)

if "%jvms%"=="0" set hostlist=%master%

call :setjavaparms
if "%1"=="" goto :benchmark

if not "%jvms%"=="0" (
    set CLASSPATH=%CLASSPATH%\..\DataCore;%CLASSPATH%
)

call :startjvms
if errorlevel 1 goto :error

call :mastercfg

if "%1"=="persist1" (
    call :persist1 %2 %3
    goto :done
)

if "%1"=="persist2" (
    call :persist2
    call :sysinfo End
    goto :done
)

if "%masteraffinity%"=="" (
    %exec% java %javaparms% %masterjavaparms% %1 %2 %3 %4 %5 %6 %7 %8 %9
) else (
    %exec% start /wait %masteraffinity% cmd /c "java %javaparms% %masterjavaparms%
%1 %2 %3 %4 %5 %6 %7 %8 %9"
)

:done
call :killjvms
```

```
del /q /f disk.info volume.info spc1.multi >nul: 2>&1
goto :eof

:error
del /q /f disk.info volume.info spc1.multi >nul: 2>&1
exit /b 1


rem
rem Set output folder
rem

:setoutput
(
    set stamp=%COMPUTERNAME%_%DATE:~4,2%-
%DATE:~7,2%_%TIME:~0,2%%TIME:~3,2%%TIME:~6,2%
    set stamp=!stamp: =0!
    set stamp=!stamp:X3650=!
    set stamp=!stamp:x3650=!

    set output=metrics
    if "%1"=="range" set output=rangetest
    if "%1"=="metrics" set output=metrics
    if "%1"=="persist1" set output=persistence1
    if "%1"=="persist2" set output=persistence2
    if "%1"=="repeat1" set output=repeatability1
    if "%1"=="repeat2" set output=repeatability2
    if "%1"=="prefill" set output=prefill.out
    if "%1"=="slave" set output=host
    if "%1"=="SPCout" set output=SPCout
    if "%1"=="package" set output=Full_Runs\!stamp!_Run

    if not "%resume%"=="" (
        if not exist !output! mkdir !output!
        exit /b 0
    )

    if not "%1"=="package" (
        if exist !output! (
            mkdir Previous_Runs >nul: 2>&1
            set s=!stamp!
            if exist !output!\stamp set /p s= < !output!\stamp
            move /y !output! Previous_Runs\!output!_!s! >nul: 2>&1
            if exist !output! (
                robocopy !output! Previous_Runs\!output!_!s! /s /move >nul: 2>&1
                rmdir /s /q !output! >nul: 2>&1
            )
        )
    )

    if not exist !output! mkdir !output!
    echo !stamp! > !output!\stamp
)
exit /b 0


rem
rem Set java parameters
rem

:setjavaparms
(
    set args=
```

```
    for /f delims^=^"^ tokens^=2 %%i in ('findstr javaparms= spc1.cfg') do (
        for %%x in (%%i) do (
            set v=%%x
            echo " !args! !javaparms! " > java.parms
            for /f %%y in ('findstr /c:" %%x " java.parms') do set v=
            if not "!v!"=="" set args=!args! !v!
            del /q /f java.parms >nul: 2>&1
        )
    )

    set javaparms=!args! !javaparms!
    set javaparms=!javaparms:~1!
)
exit /b 0


rem
rem Run benchmark
rem

:benchmark
(
    for %%o in (SPCout range metrics repeat1 repeat2 persist1 prefill) do (
        call :setoutput %%o
    )

    call :sysinfo Start
    call :prefill

    set output=metrics
    if not "%warmuptime%"=="" set output=rangetest

    call :startjvms
    if errorlevel 1 goto :error
    call :mastercfg

    if not "%warmuptime%"=="" (
        set /a warmuprate=rate/2
        %exec% java %javaparms% %masterjavaparms% range -b !warmuprate! -s
%ramptime% -t %warmuptime%
        call :restartjvms metrics
        if errorlevel 1 goto :error
    )

    %exec% java %javaparms% %masterjavaparms% metrics -b %rate% -s %ramptime% -t
%runtime%

    call :restartjvms repeatability1
    if errorlevel 1 goto :error

    %exec% java %javaparms% %masterjavaparms% repeat1 -b %rate% -s %ramptime%

    call :restartjvms repeatability2
    if errorlevel 1 goto :error

    %exec% java %javaparms% %masterjavaparms% repeat2 -b %rate% -s %ramptime%
    call :killjvms

    for %%h in (%hostlist%) do (
        if /I "%%h" NEQ "%master%" (
            xcopy /s /q /r /y /z /i \\%%h\%share%\host !output! >nul:
        )
```

```
    )

    set /a jvms=0
    set output=persistence1

    call :mastercfg
    call :persist1 -b %rate%

    echo call RunSPC1 %dflag% persist2 > Persist2.bat
    echo call RunSPC1 %dflag% package >> Persist2.bat
    echo pause >> Persist2.bat
    pause
)
exit /b 0


rem
rem Restart JVMs
rem

:restartjvms
(
    call :killjvms

    for %%h in (%hostlist%) do (
        if /I "%%h" NEQ "%master%" (
            xcopy /s /q /r /y /z /i \\%%h\%share%\host %output% >nul:
            copy /Y /Z %output%\spc1-%%h.cfg \\%%h\%share%\spc1-%master%.cfg >nul:
        )
    )

    set output=%1
    copy /y spc1.cfg %1 > nul:

    call :startjvms
    if errorlevel 1 goto :error
)
exit /b 0


rem
rem Package benchmark output
rem

:package
(
    move /y rangetest %output%
    move /y metrics %output%
    move /y persistence1 %output%
    move /y persistence2 %output%
    move /y prefill.out %output%
    move /y repeatability1 %output%
    move /y repeatability2 %output%
    move /y SPCOut %output%
    move /y HW_INFO %output%
    move /y asu*.map %output%
    move /y asu*.jnl %output%
    copy /y spc1.cfg %output%
    copy /y spc1.parm %output%
    copy /y *.config %output%
    copy /y *.ps1 %output%
    copy /y *.bat %output%
```

```
    for %%h in (%sdslist%) do (
        robocopy /e \\%%h\%share%\HW_INFO %output%\HW_INFO >nul:
    )
)
exit /b 0


rem
rem Get system info
rem

:sysinfo
(
    for %%h in (%sdslist%) do (
        echo Saving system configuration on %%h ^(takes a long time^) ...

        if /I "%%h" NEQ "%master%" (
            copy /Y /Z GetInfo.ps1 \\%%h\%share%\GetInfo.ps1 >nul:
            copy /Y /Z .\ibm_utl_dsa_dsyte1d-9.61_portable_windows_x86-64.exe
\\%%h\%share%\ibm_utl_dsa_dsyte1d-9.61_portable_windows_x86-64.exe > nul:
            .\psexec \\%%h %psauth% -c -f %cmd% %dflag% slave info %1 2>nul: < nul:
        ) else (
            call :localinfo %1
        )
    )
)
exit /b 0


rem
rem Get local info
rem

:localinfo
(
    if "%1"=="Start" rmdir /s /q HW_INFO >nul: 2>&1

    set sys=%COMPUTERNAME%
    mkdir HW_INFO\!sys!_%1 >nul: 2>&1

    powershell.exe .\GetInfo.ps1 -NoLogo -NonInteractive >
HW_INFO\!sys!_psOutput%1.txt
    rem %exec% .\ibm_utl_dsa_dsyte1d-9.61_portable_windows_x86-64.exe -b -d
HW_INFO\!sys!_%1 -v >> HW_INFO\!sys!_psOutput%1.txt
)
exit /b 0


rem
rem Run prefill
rem

:prefill
(
    (
        echo compratio=1
        echo sd=default,threads=8

        for /f "delims==\, tokens=2,4,5,*" %%i in ('findstr sd= spc1.cfg') do (
            set x=%%k
            set x=!x:~1,1!
            set tail=
            if "!x!"==":" set tail=%%l
```

```
            if not "!tail!"=="" set tail=!tail: =!
            if not "!tail!"=="" set tail=,!tail!
            echo sd=%%i,lun=\\%%j\%%k!tail!,threads=4
        )

        echo wd=default,rdpct=0,seek=-1,xfersize=512k
        set /a wd=1

        for /f "delims==, tokens=2" %%i in ('findstr sd= spc1.cfg') do (
            echo wd=wd!wd!,sd=%%i
            set /a wd=wd+1
        )

        echo rd=asuprefill,wd=wd*,iorate=max,elapsed=24h,interval=60
    ) > prefill.cfg

    %exec% call C:\spc\vdbench\vdbench.bat -f prefill.cfg -o %output%
    move /y prefill.cfg %output% >nul:
)
goto :done


rem
rem Run Persist1
rem

:persist1
(
    set /a persistrate=%2
    if not "%spc2persist%"=="" set /a persistrate=persistrate/10

    %exec% java %javaparms% %masterjavaparms% persist1 -b !persistrate!
    if "%spc2persist%"=="" exit /b 0

    set /a streams=^(%2+2^)/30
    set /a spc2jvms=^(streams+599^)/600

    (
        echo host=localhost,jvms=!spc2jvms!,maxstreams=!streams!

        for /f "delims==\, tokens=2,4,5,*" %%i in ('findstr sd= spc1.cfg') do (
            set tail=%%l
            if not "!tail!"=="" set tail=!tail: =!
            if not "!tail!"=="" set tail=,!tail!
            echo sd=%%i,lun=\\%%j\%%k!tail!
        )

        echo maxlatestart=1
        echo reportinginterval=5
        echo segmentlength=512m
        echo
rd=default,rampup=180,periods=90,measurement=300,runout=0,rampdown=0,buffers=1
        echo rd=default,rdpct=0,xfersize=1024k
        echo rd=TR1-5s_SPC-2-persist-w,streams=!streams!
    ) > %output%\spc2.cfg

    set CLASSPATH=%spc2path%
    %exec% java %spc2javaparms% vdbench -w SPC2 -f %output%\spc2.cfg -init -o
%output%\persistinit
    %exec% java %spc2javaparms% vdbench -w SPC2 -f %output%\spc2.cfg -o
%output%\persist1-spc2
)
exit /b 0
```

```
rem
rem Run Persist2
rem

:persist2
(
    if "%spc2persist%"=="" (
        %exec% java %javaparms% %masterjavaparms% persist2
        exit /b 0
    )

    (
        for /f "delims==\, tokens=2,4,5,*" %%i in ('findstr sd= spc1.cfg') do (
            set tail=%%l
            if not "!tail!"=="" set tail=!tail: =!
            if not "!tail!"=="" set tail=,!tail!
            echo sd=%%i,lun=\\%%j\%%k!tail!
        )

        echo maxlatestart=1
        echo reportinginterval=5
        echo segmentlength=512m
        echo rd=default,buffers=1,rdpct=100,xfersize=1024k
        echo rd=TR1-200s_SPC-2-persist-r
    ) > %output%\spc2.cfg

    set CLASSPATH=%spc2path%
    %exec% java %spc2javaparms% vdbench -w SPC2 -f %output%\spc2.cfg -o
%output%\persist2-spc2
)
exit /b 0


rem
rem Start JVMs
rem

:startjvms
(
    set /a hosts=0
    for %%i in (%hostlist%) do set /a hosts+=1

    set jparms=!javaparms!
    if not "!jparms!"=="" set jparms="!jparms!"

    set /a i=1
    set /a resid=jvms

    for %%h in (%hostlist%) do (
        set /a hostjvms="(!%%h_rate!+bsusperjvm-1)/bsusperjvm"
        set /a "n=(resid+hosts-1)/hosts"
        set /a hosts-=1

        if !hostjvms! GTR 0 (
            if !hostjvms! GTR !resid! (
                set /a n=resid
            ) else (
                set /a n=hostjvms
            )
        )
```

```
        if !n! GTR 0 (
            echo Starting !n! on %%h ...
            if /I "%%h" NEQ "%master%" (
                set rflag=
                for /L %%j in (1,%maxstart%,!n!) do (
                    set /a x=n-%%j+1
                    if !x! GTR %maxstart% set /a x=%maxstart%
                    .\psexec \\%%h %psauth% -c -f %cmd% %dflag% !rflag!
!%%h_setsize! !jparms! slave %master% !i! !x! !%%h_basecpu! 2>nul:
                    set /a i+=x
                    set rflag=-r
                )
                copy /Y /Z \\%%h\%share%\spc1-%master%.cfg %output%\spc1-%%h.cfg
>nul:
                del /q /f \\%%h\%share%\spc1-%master%.cfg >nul: 2>&1
            ) else (
                call :startlocal %master% !i! !n! %basecpu%
                set /a i+=n
            )
        )

        set /a resid-=n
    )

    if !resid! GTR 0 (
        echo Error: Could not start enough JVMs - check per-host rates
        goto :error
    )
)
exit /b 0


rem
rem Start local JVMs
rem

:startlocal
(
    set /a baseset="(%4+0)/setsize"
    set /a sets="(cpus/setsize-baseset)"

    for /L %%i in (1,1,%3) do (
        set /a host=%2+%%i-1
        set host=h!host!

        mkdir %output%\!host! >nul: 2>&1

        (
            echo Master=%1
            echo Host=!host!
            if /I "%COMPUTERNAME%" NEQ "%1" (
                findstr sd= spc1-%1.cfg
            ) else (
                findstr sd= spc1.cfg
            )
        ) > %output%\!host!\!host!.cfg

        if "!baseset!"=="0" (
            set /a numa=%%i"%%"numas
            set /a setnum=%%i/numas"%%"setspernuma
        ) else (
            set /a setnum=sets-"("%%i-1")%%"sets-1+baseset
            set /a numa=setnum/setspernuma
```

```
            set /a setnum=setnum"%%"setspernuma
        )

        call :setmask !setsize! !setnum!
        %exec% start %priority% /node !numa! /affinity !mask! /min java %javaparms%
spc1 -f %output%\!host!\!host!.cfg -o %output%\!host!
    )
)
exit /b 0


rem
rem Setup master configuration file
rem

:mastercfg
(
    set /a hosts=0
    for %%i in (%hostlist%) do set /a hosts+=1

    set slaves=
    set /a "n=(jvms+hosts-1)/hosts"
    for /L %%i in (1,1,!n!) do for /L %%j in (%%i,!n!,%jvms%) do set
slaves=!slaves!,h%%j
    set slaves=^(!slaves:~1!^)

    findstr javaparms= spc1.cfg > spc1.multi

    if not "%jvms%"=="0" (
        echo host=master >> spc1.multi
        echo slaves=!slaves! >> spc1.multi
    )

    findstr sd= spc1.cfg >> spc1.multi

    move /y spc1.multi spc1.cfg >nul:
    copy /y spc1.cfg %output% > nul:
)
exit /b 0


rem
rem Setup configuration files
rem

:config
(
    if "%1"=="slave" exit /b 0
    if "%1"=="package" exit /b 0

    call :configure %master% %1
    if errorlevel 1 goto :error

    move /y spc1.multi spc1.cfg >nul:

    if "%jvms%"=="0" exit /b 0

    for %%h in (%hostlist%) do (
        if /I "%%h" NEQ "%master%" (
            call :slavecfg %%h %1
            if errorlevel 1 goto :error
        )
    )
```

```
)
exit /b 0


rem
rem Update configuration files on slave system
rem

:slavecfg
(
    call :configure %1 %2
    if errorlevel 1 goto :error

    copy /Y /Z spc1.multi \\%1\%share%\spc1-%master%.cfg >nul:
    del /q /f spc1.multi >nul: 2>&1
)
exit /b 0


rem
rem Configure drives
rem

:configure
(
    echo Configuring %1 ...

    for /l %%i in (6,-1,1) do (
        call :discover %1 %%i %2
        call :updatecfg %1 %%i
        if not errorlevel 1 exit /b 0
    )

    goto :error
)
exit /b 0


rem
rem Update drives in spc1.cfg
rem

:updatecfg
(
    findstr javaparms= spc1.cfg > spc1.multi

    if /I "%1" EQU "%master%" (
        findstr host= spc1.cfg >> spc1.multi
        findstr slaves= spc1.cfg >> spc1.multi
    )

    for /f "delims==\, tokens=2,4,5,*" %%i in ('findstr sd= spc1.cfg') do (
        set drive=
        set tail=%%l
        if not "!tail!"=="" set tail=!tail: =!
        if not "!tail!"=="" set tail=,!tail!

        set x=%%k
        set x=!x:~1,1!

        if not "!x!"==":" (
            for /f "delims=, tokens=1" %%x in ('findstr /i /r ",%prefix%%%i[^0-9]"
disk.info') do set drive=%%x
```

```
            if not "!drive!"=="" echo sd=%%i,lun=\\%%j\PhysicalDrive!drive!!tail! >>
spc1.multi
        )

        if "!x!"==":" (
            set mount=%%k
            set mount=!mount::=!

            for /f "tokens=3,4" %%x in ('findstr /i /c:" !mount! " volume.info') do
(
                if /I "%%x %%y" EQU "!mount! RAW" set drive=%%x
            )

            if not "!drive!"=="" (echo sd=%%i,lun=\\%%j\!drive!:!tail!) >>
spc1.multi
        )

        if "!drive!"=="" (
            if "%2"=="1" echo Error: Volume %%i on %1 not found or not RAW
            goto :error
        )
    )

    del /q /f disk.info volume.info >nul: 2>&1
)
exit /b 0


rem
rem Discover disks
rem

:discover
(
    set rescan=
    if "%2"=="3" set rescan=true
    if "%2_%3"=="6_" set rescan=true
    if "%2_%3"=="6_config" set rescan=true
    if "%2_%3"=="6_persist2" set rescan=true

    if /I "%1" EQU "%master%" (
        set run=.\psexec \\localhost %psauth%
        if not exist "c:\windows\system32\drivers\dcssup.sys" (
            .\devcon rescan
        ) else (
            if not "!rescan!"=="" .\devcon rescan
        )
    ) else (
        set run=.\psexec \\%1 %psauth%
        !run! -f -c .\devcon rescan 2>nul:
    )

    if not "!rescan!"=="" (
        !run! cmd /c "echo rescan | diskpart" >nul: 2>&1
        !run! cmd /c "echo list disk | diskpart" 2>nul: > disk.info

        for /f "tokens=2" %%x in ('findstr "line Foreign" disk.info') do (
            !run! cmd /c "(echo select disk %%x && echo offline disk && echo online
disk && echo attributes disk clear readonly) | diskpart" >nul: 2>&1
        )
    )

    !run! cmd /c "echo list disk | diskpart" 2>nul: > disk.info
```

```
    for /f "tokens=2" %%x in ('findstr "Offline" disk.info') do (
        !run! cmd /c "(echo select disk %%x && echo online disk) | diskpart" >nul:
2>&1
    )

    set disk=
    for /f "tokens=2" %%x in ('findstr "Foreign" disk.info') do set disk=%%x
    if not "!disk!"=="" !run! cmd /c "(echo select disk !disk! && echo import) |
diskpart" >nul: 2>&1

    del /q /f disk.info >nul: 2>&1

    if /I "%1" EQU "%master%" (
        .\DcsCDiskInfo | findstr Virtual > disk.info
    ) else (
        .\psexec \\%1 %psauth% -f -c .\DcsCDiskInfo 2>nul: | findstr Virtual >
disk.info
    )

    !run! cmd /c "echo list volume | diskpart" > volume.info 2>nul:
)
exit /b 0


rem
rem Kill JVMs
rem

:killjvms
(
    set kill=

    if "%exec%"=="" (
        for %%h in (localhost %hostlist% %sdslist%) do (
            tskill /a /server:%%h java >nul: 2>&1
            if not errorlevel 1 set kill=true
            tskill /a /server:%%h javaw >nul: 2>&1
            if not errorlevel 1 set kill=true
        )
    )

    if not "!kill!"=="" goto :killjvms
)
exit /b 0


rem
rem Set CPU configuration
rem

:setcpus
(
    set /a numas=%HIGHESTNUMANODENUMBER%+1
    if "!numas!"=="1" (
        for /l %%i in (1,1,8) do (
            start /node %%i cmd /c exit > numa.log 2>&1
            for /f %%x in ('type numa.log') do goto :numadone
            set /a numas+=1
        )
    :numadone
        del /f /q numa.log >nul: 2>&1
    )
```

```
    set /a cpus=NUMBER_OF_PROCESSORS
    if "!cpus!"=="36" set /a cpus=72
    if "!cpus!"=="60" set /a cpus=120

    set /a cpuspernuma=cpus/numas
)
exit /b 0


rem
rem Set affinity mask
rem

:setmask
(
    if "%1"=="1" (
        if "%2"=="0" set mask=000000001
        if "%2"=="1" set mask=000000002
        if "%2"=="2" set mask=000000004
        if "%2"=="3" set mask=000000008
        if "%2"=="4" set mask=000000010
        if "%2"=="5" set mask=000000020
        if "%2"=="6" set mask=000000040
        if "%2"=="7" set mask=000000080
        if "%2"=="8" set mask=000000100
        if "%2"=="9" set mask=000000200
        if "%2"=="10" set mask=000000400
        if "%2"=="11" set mask=000000800
        if "%2"=="12" set mask=000001000
        if "%2"=="13" set mask=000002000
        if "%2"=="14" set mask=000003000
        if "%2"=="15" set mask=000004000
        if "%2"=="16" set mask=000010000
        if "%2"=="17" set mask=000020000
        if "%2"=="18" set mask=000040000
        if "%2"=="19" set mask=000080000
        if "%2"=="20" set mask=000100000
        if "%2"=="21" set mask=000200000
        if "%2"=="22" set mask=000400000
        if "%2"=="23" set mask=000800000
        if "%2"=="24" set mask=001000000
        if "%2"=="25" set mask=002000000
        if "%2"=="26" set mask=004000000
        if "%2"=="27" set mask=008000000
        if "%2"=="28" set mask=010000000
        if "%2"=="29" set mask=020000000
        if "%2"=="30" set mask=040000000
        if "%2"=="31" set mask=080000000
        if "%2"=="32" set mask=100000000
        if "%2"=="33" set mask=200000000
        if "%2"=="34" set mask=400000000
        if "%2"=="35" set mask=800000000
        exit /b 0
    )

    if "%1"=="2" (
        if "%2"=="0" set mask=000000003
        if "%2"=="1" set mask=00000000C
        if "%2"=="2" set mask=000000030
        if "%2"=="3" set mask=0000000C0
        if "%2"=="4" set mask=000000300
        if "%2"=="5" set mask=000000C00
```

```
        if "%2"=="6" set mask=000003000
        if "%2"=="7" set mask=00000C000
        if "%2"=="8" set mask=000030000
        if "%2"=="9" set mask=0000C0000
        if "%2"=="10" set mask=000300000
        if "%2"=="11" set mask=000C00000
        if "%2"=="12" set mask=003000000
        if "%2"=="13" set mask=00c000000
        if "%2"=="14" set mask=030000000
        if "%2"=="15" set mask=0c0000000
        if "%2"=="16" set mask=300000000
        if "%2"=="17" set mask=c00000000
        exit /b 0
    )

    if "%1"=="3" (
        if "%2"=="0" set mask=000000007
        if "%2"=="1" set mask=000000038
        if "%2"=="2" set mask=0000001c0
        if "%2"=="3" set mask=000000e00
        if "%2"=="4" set mask=000007000
        if "%2"=="5" set mask=000038000
        if "%2"=="6" set mask=0001C0000
        if "%2"=="7" set mask=000e00000
        if "%2"=="8" set mask=007000000
        if "%2"=="9" set mask=038000000
        if "%2"=="10" set mask=1c0000000
        if "%2"=="11" set mask=e00000000
        exit /b 0
    )

    if "%1"=="4" (
        if "%2"=="0" set mask=00000000f
        if "%2"=="1" set mask=0000000f0
        if "%2"=="2" set mask=000000f00
        if "%2"=="3" set mask=00000f000
        if "%2"=="4" set mask=0000f0000
        if "%2"=="5" set mask=000f00000
        if "%2"=="6" set mask=00f000000
        if "%2"=="7" set mask=0f0000000
        if "%2"=="8" set mask=f00000000
        exit /b 0
    )

    if "%1"=="6" (
        if "%2"=="0" set mask=00000003f
        if "%2"=="1" set mask=000000fC0
        if "%2"=="2" set mask=00003f000
        if "%2"=="3" set mask=000fC0000
        if "%2"=="4" set mask=03f000000
        if "%2"=="5" set mask=fC0000000
        exit /b 0
    )

    if "%1"=="7" (
        if "%2"=="0" set mask=00000007f
        if "%2"=="1" set mask=000003f80
        if "%2"=="2" set mask=0001fc000
        if "%2"=="3" set mask=00fe00000
        if "%2"=="4" set mask=7f0000000
        exit /b 0
    )
```

```
    if "%1"=="8" (
        if "%2"=="0" set mask=0000000ff
        if "%2"=="1" set mask=00000ff00
        if "%2"=="2" set mask=000ff0000
        if "%2"=="3" set mask=0ff000000
        exit /b 0
    )

    if "%1"=="9" (
        if "%2"=="0" set mask=0000001ff
        if "%2"=="1" set mask=00003fe00
        if "%2"=="2" set mask=007fc0000
        if "%2"=="3" set mask=ff8000000
        exit /b 0
    )

    if "%1"=="12" (
        if "%2"=="0" set mask=000000fff
        if "%2"=="1" set mask=000fff000
        if "%2"=="2" set mask=fff000000
        exit /b 0
    )

    if "%1"=="14" (
        if "%2"=="0" set mask=000003fff
        if "%2"=="1" set mask=00fffc000
        exit /b 0
    )

    if "%1"=="18" (
        if "%2"=="0" set mask=00003ffff
        if "%2"=="1" set mask=ffffc0000
        exit /b 0
    )

    set mask=invalid
)
exit /b 1
```

config.bat

This script sets various benchmark-related environment variables.

```
set rate=24050
set ramptime=480

set hostlist=Pserver-E, Pserver-A
set sdslist=Pserver-E, Pserver-A

set PServer-E_basecpu=42
set PServer-A_basecpu=42
set PServer-E_setsize=6
set PServer-A_setsize=6

rem set masteraffinity=/node 0 /affinity 3f
set masteraffinity=/node 1
rem set PServer-E_rate=16000
```

Persist2.bat

This script will invoke **RunSPC1.bat** to execute the Persistence Test Run 2 *(read phase)* and this invoke **RunSPC1.bat** again to 'package' the output.

```
call RunSPC1  persist2
call RunSPC1  package
pause
```

APPENDIX F: THIRD-PARTY QUOTATION

Priced Storage Configuration

The third-party quotation is not embedded in this document due it size and format. The quotation is available via the following hyperlink:

Third-Party Quotation