SPC Benchmark 2™
Full Disclosure Report

Kaminario, Inc.
Kaminario K2 *(K2F00000700)*

SPC-2™ V1.5

Submitted for Review: November 20, 2013
Submission Identifier: B00068

First Edition – November 2013

THE INFORMATION CONTAINED IN THIS DOCUMENT IS DISTRIBUTED ON AN AS IS BASIS WITHOUT ANY WARRANTY EITHER EXPRESS OR IMPLIED. The use of this information or the implementation of any of these techniques is the customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by Kaminario, Inc. for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

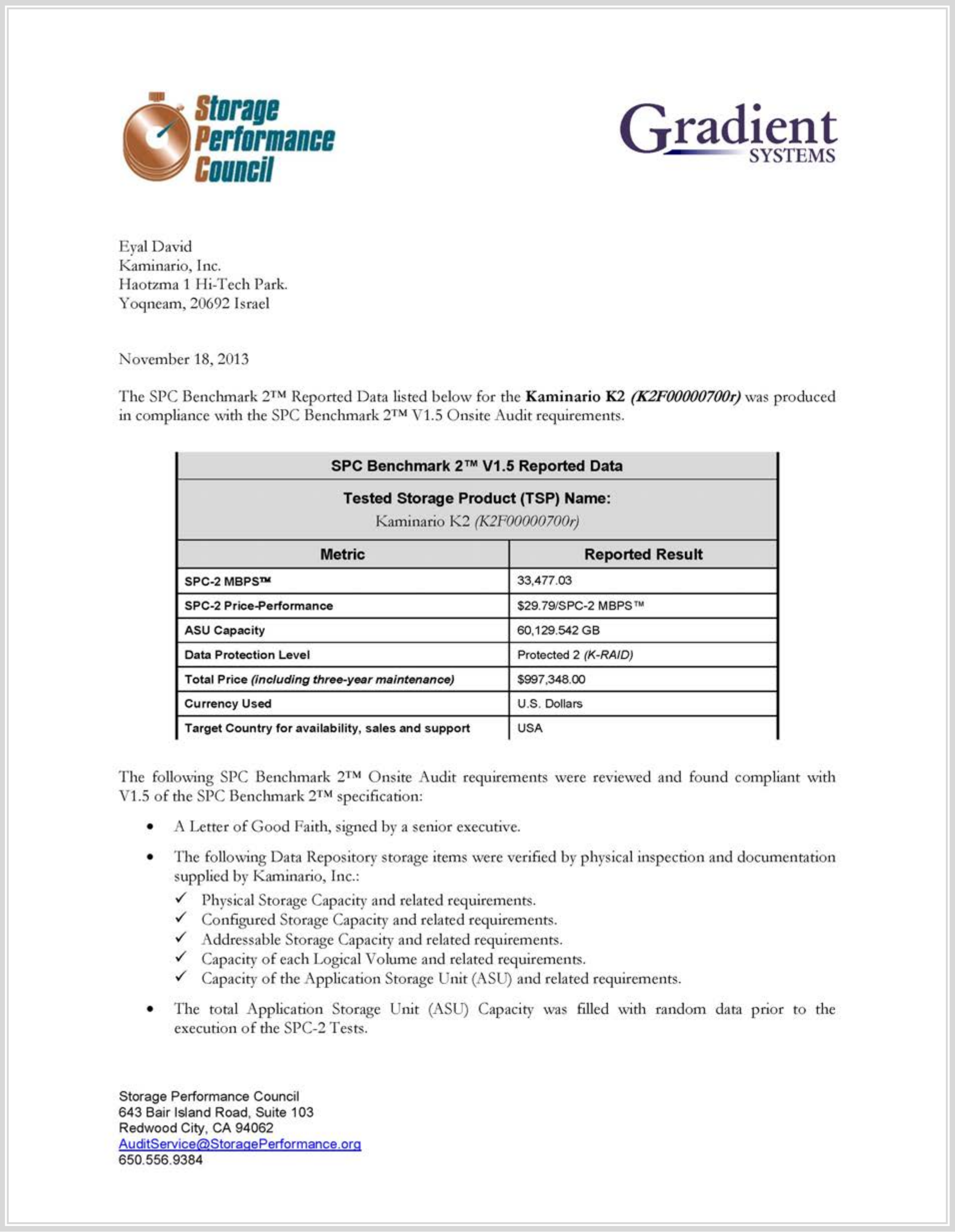
This publication was produced in the United States. Kaminario, Inc. may not offer the products, services, or features discussed in this document in other countries, and the information is subject to change with notice. Consult your local Kaminario, Inc. representative for information on products and services available in your area.

Trademarks

SPC Benchmark 2, SPC-2, SPC-2 MBPS, and SPC-2 Price-Performance are trademarks of the Storage Performance Council. Kaminario, K2 and SPEAR are trademarks or registered trademarks of Kaminario, Inc. in the United States and other countries. All other brands, trademarks, and product names are the property of their respective owners.

Table of Contents

AUDIT CERTIFICATION



Eyal David
Kaminario, Inc.
Haotzma 1 Hi-Tech Park.
Yoqneam, 20692 Israel

November 18, 2013

The SPC Benchmark 2™ Reported Data listed below for the **Kaminario K2** ***(K2F00000700r)*** was produced in compliance with the SPC Benchmark 2™ V1.5 Onsite Audit requirements.

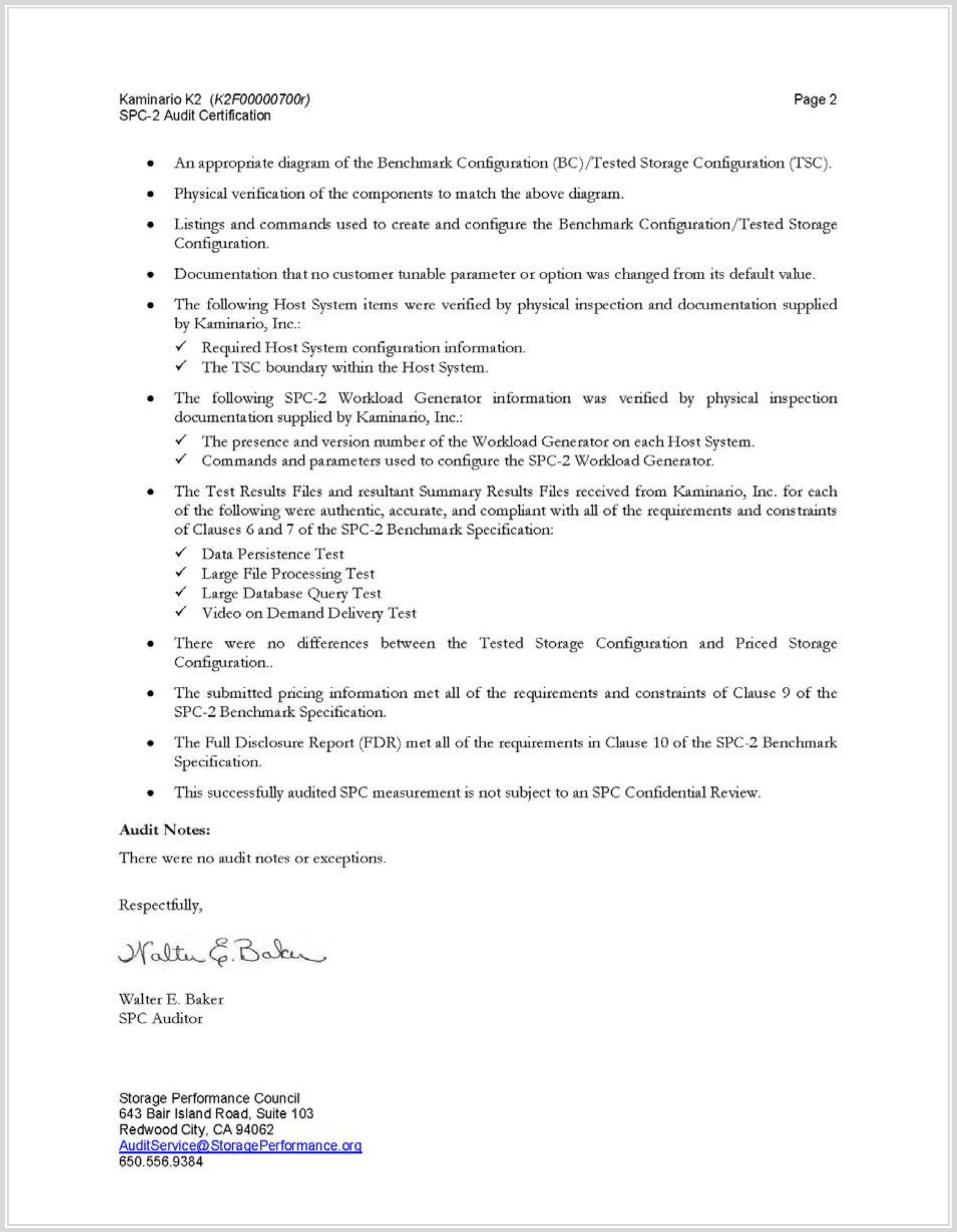
SPC Benchmark 2™ V1.5 Reported Data	
Tested Storage Product (TSP) Name: Kaminario K2 *(K2F00000700r)*	
Metric	**Reported Result**
SPC-2 MBPS™	33,477.03
SPC-2 Price-Performance	$29.79/SPC-2 MBPS™
ASU Capacity	60,129.542 GB
Data Protection Level	Protected 2 *(K-RAID)*
Total Price ***(including three-year maintenance)***	$997,348.00
Currency Used	U.S. Dollars
Target Country for availability, sales and support	USA

The following SPC Benchmark 2™ Onsite Audit requirements were reviewed and found compliant with V1.5 of the SPC Benchmark 2™ specification:

- A Letter of Good Faith, signed by a senior executive.
- The following Data Repository storage items were verified by physical inspection and documentation supplied by Kaminario, Inc.:
  - ✓ Physical Storage Capacity and related requirements.
  - ✓ Configured Storage Capacity and related requirements.
  - ✓ Addressable Storage Capacity and related requirements.
  - ✓ Capacity of each Logical Volume and related requirements.
  - ✓ Capacity of the Application Storage Unit (ASU) and related requirements.
- The total Application Storage Unit (ASU) Capacity was filled with random data prior to the execution of the SPC-2 Tests.

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

AUDIT CERTIFICATION *(CONT.)*



- An appropriate diagram of the Benchmark Configuration (BC)/Tested Storage Configuration (TSC).
- Physical verification of the components to match the above diagram.
- Listings and commands used to create and configure the Benchmark Configuration/Tested Storage Configuration.
- Documentation that no customer tunable parameter or option was changed from its default value.
- The following Host System items were verified by physical inspection and documentation supplied by Kaminario, Inc.:
  - ✓ Required Host System configuration information.
  - ✓ The TSC boundary within the Host System.
- The following SPC-2 Workload Generator information was verified by physical inspection documentation supplied by Kaminario, Inc.:
  - ✓ The presence and version number of the Workload Generator on each Host System.
  - ✓ Commands and parameters used to configure the SPC-2 Workload Generator.
- The Test Results Files and resultant Summary Results Files received from Kaminario, Inc. for each of the following were authentic, accurate, and compliant with all of the requirements and constraints of Clauses 6 and 7 of the SPC-2 Benchmark Specification:
  - ✓ Data Persistence Test
  - ✓ Large File Processing Test
  - ✓ Large Database Query Test
  - ✓ Video on Demand Delivery Test
- There were no differences between the Tested Storage Configuration and Priced Storage Configuration..
- The submitted pricing information met all of the requirements and constraints of Clause 9 of the SPC-2 Benchmark Specification.
- The Full Disclosure Report (FDR) met all of the requirements in Clause 10 of the SPC-2 Benchmark Specification.
- This successfully audited SPC measurement is not subject to an SPC Confidential Review.

Audit Notes:

There were no audit notes or exceptions.

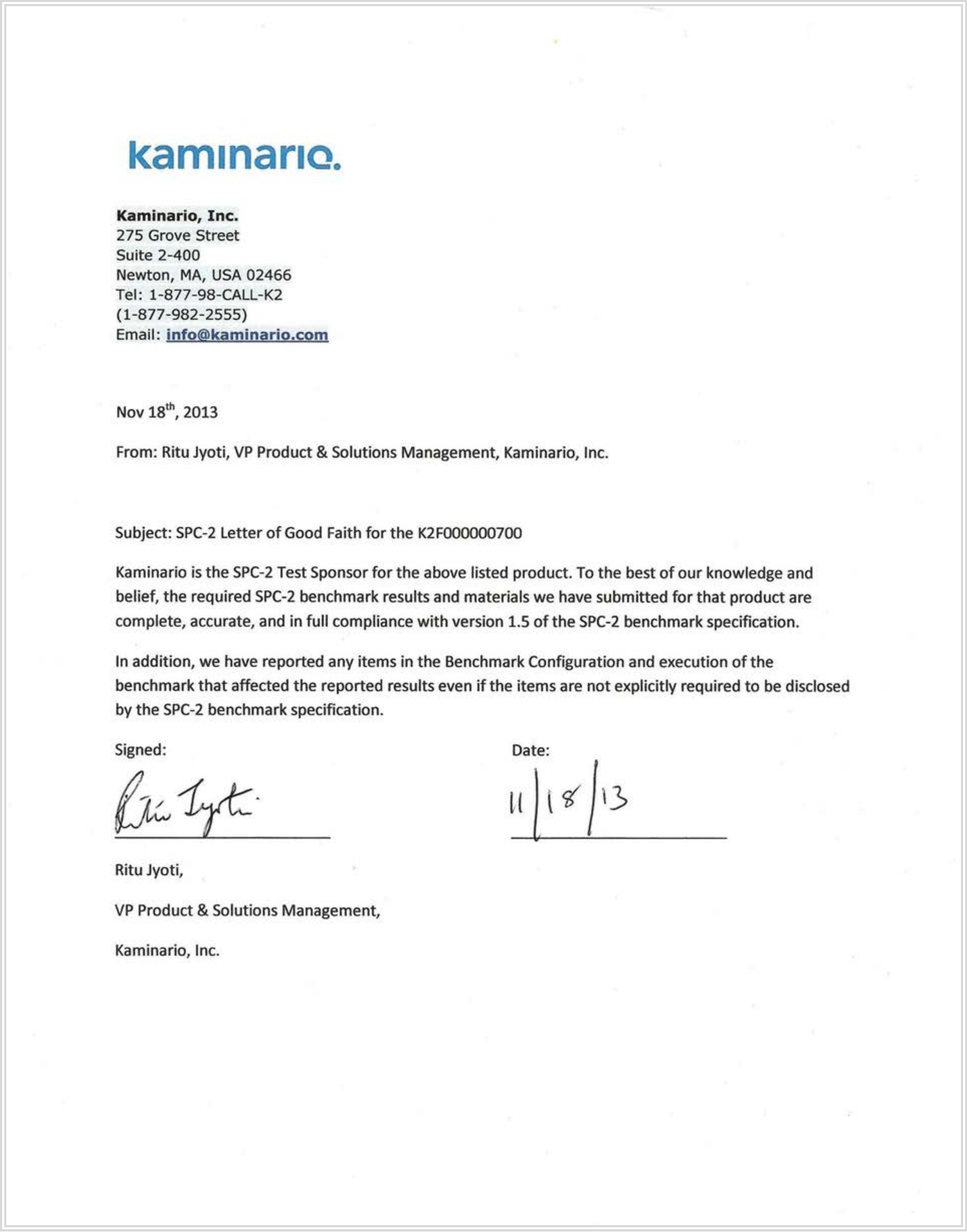
Respectfully,

Walter E. Baker

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



kaminario.

Kaminario, Inc.
275 Grove Street
Suite 2-400
Newton, MA, USA 02466
Tel: 1-877-98-CALL-K2
(1-877-982-2555)
Email: info@kaminario.com

Nov 18th, 2013

From: Ritu Jyoti, VP Product & Solutions Management, Kaminario, Inc.

Subject: SPC-2 Letter of Good Faith for the K2F000000700

Kaminario is the SPC-2 Test Sponsor for the above listed product. To the best of our knowledge and belief, the required SPC-2 benchmark results and materials we have submitted for that product are complete, accurate, and in full compliance with version 1.5 of the SPC-2 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-2 benchmark specification.

Signed: Ritu Jyoti

Date: 11/18/13

Ritu Jyoti,

VP Product & Solutions Management,

Kaminario, Inc.

EXECUTIVE SUMMARY

Test Sponsor and Contact Information

Test Sponsor and Contact Information	
Test Sponsor Primary Contact	Kaminario, Inc. – www.kaminario.com Eyal David – eyal.david@kaminario.com Haotzma 1 Hi-Tech Park Yoqneam 20692 Israel Phone: +972 72 222 4495 FAX: +972 4 959 05511
Test Sponsor Alternate Contact	Kaminario, Inc. – www.kaminario.com Gur Rosen Haotzma 1 Hi-Tech Park Yoqneam 20692 Israel Phone: +972 72 222 7217 FAX: +972 4 959 -0551
Auditor	Storage Performance Council – http://www.storageperformance.org Walter E. Baker – AuditService@StoragePerformance.org 643 Bair Island Road, Suite 103 Redwood City, CA 94063 Phone: (650) 556-9384 FAX: (650) 556-9385

Revision Information and Key Dates

Revision Information and Key Dates	
SPC-2 Specification revision number	V1.5
SPC-2 Workload Generator revision number	V1.2
Date Results were first used publicly	November 20, 2013
Date FDR was submitted to the SPC	November 20, 2013
Date the TSC will be available for shipment to customers	currently available
Date the TSC completed audit certification	November 18, 2013

Tested Storage Product (TSP) Description

Kaminario K2 is an enterprise class general purpose MLC Flash array that eliminates I/O and throughput bottlenecks and dramatically reduces latency to accelerate applications. The K2 is consistently fast, highly available, cost effective, and easy to deploy storage. The K2 is a fundamentally better way to store performance sensitive data.

SPC-2 Reported Data

SPC-2 Reported Data consists of three groups of information:

- The following SPC-2 Primary Metrics, which characterize the overall benchmark result:
  - SPC-2 MBPS™
  - SPC-2 Price Performance™
  - Application Storage Unit (ASU) Capacity
- Supplemental data to the SPC-2 Primary Metrics.
  - Total Price
  - Data Protection Level
  - Currency Used
  - Target Country
- Reported Data for each SPC Test: Large File Processing (LFP), Large Database Query (LDQ), and Video on Demand Delivery (VOD) Test.

SPC-2 MBPS™ represents the aggregate data rate, in megabytes per second, of all three SPC-2 workloads: Large File Processing (LFP), Large Database Query (LDQ), and Video on Demand (VOD).

SPC-2 Price-Performance™ is the ratio of **Total Price** to **SPC-2 MBPS™**.

ASU *(Application Storage Unit)* **Capacity** represents the total storage capacity available to be read and written in the course of executing the SPC-2 benchmark.

Total Price includes the cost of the Priced Storage Configuration plus three years of hardware maintenance and software support as detailed on page 18.

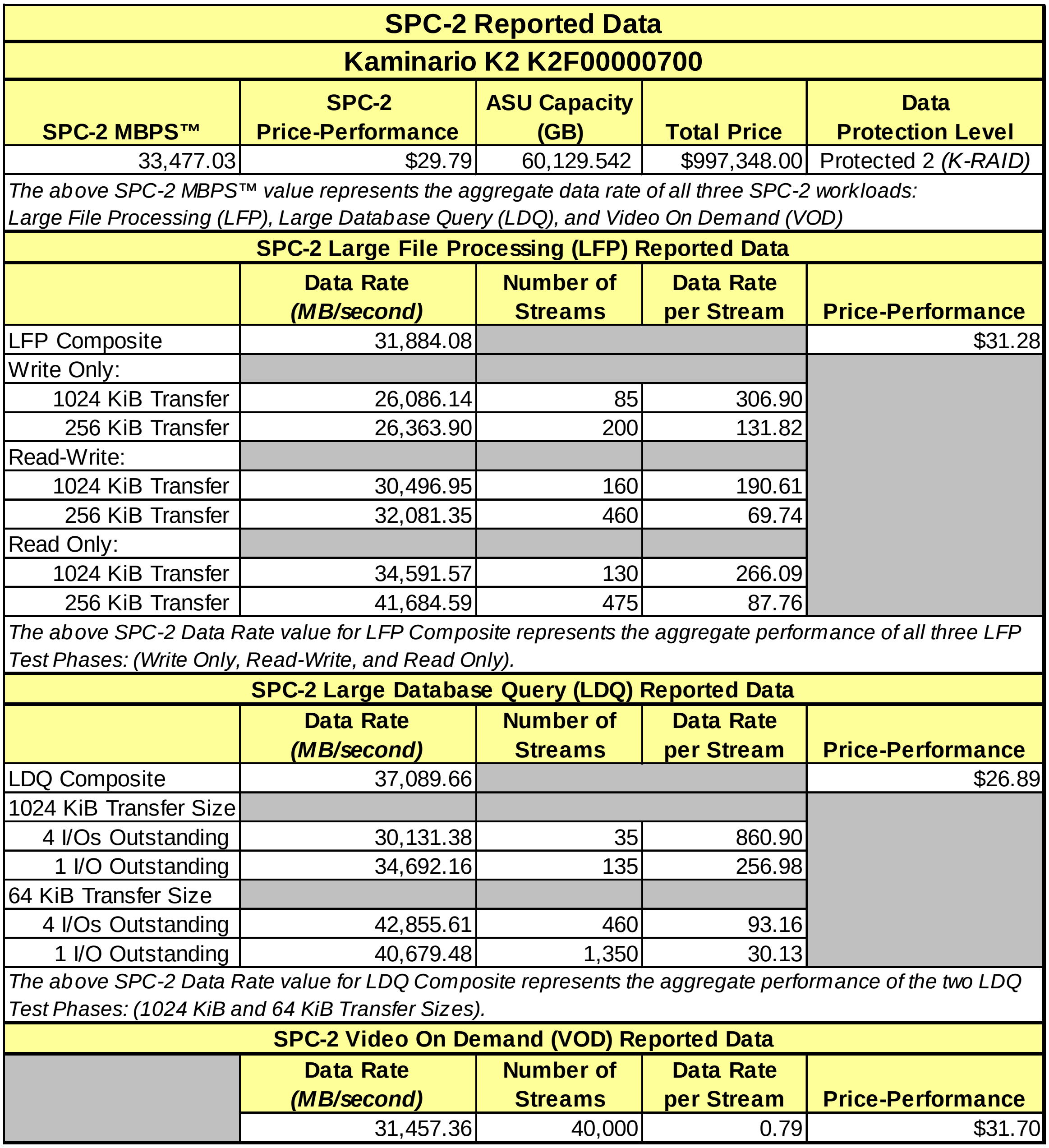
A **Data Protection Level** of **Protected 2** using ***K-RAID,*** which consists of RAID 10 during normal operation, where half of the SSD storage is allocated for data mirroring. During failures, the data is mirrored to the KMS storage capacity (HDDs).

> ***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-2 Data Repository.*

Currency Used is formal name for the currency used in calculating the **Total Price** and **SPC-2 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.

SPC-2 Reported Data *(continued)*

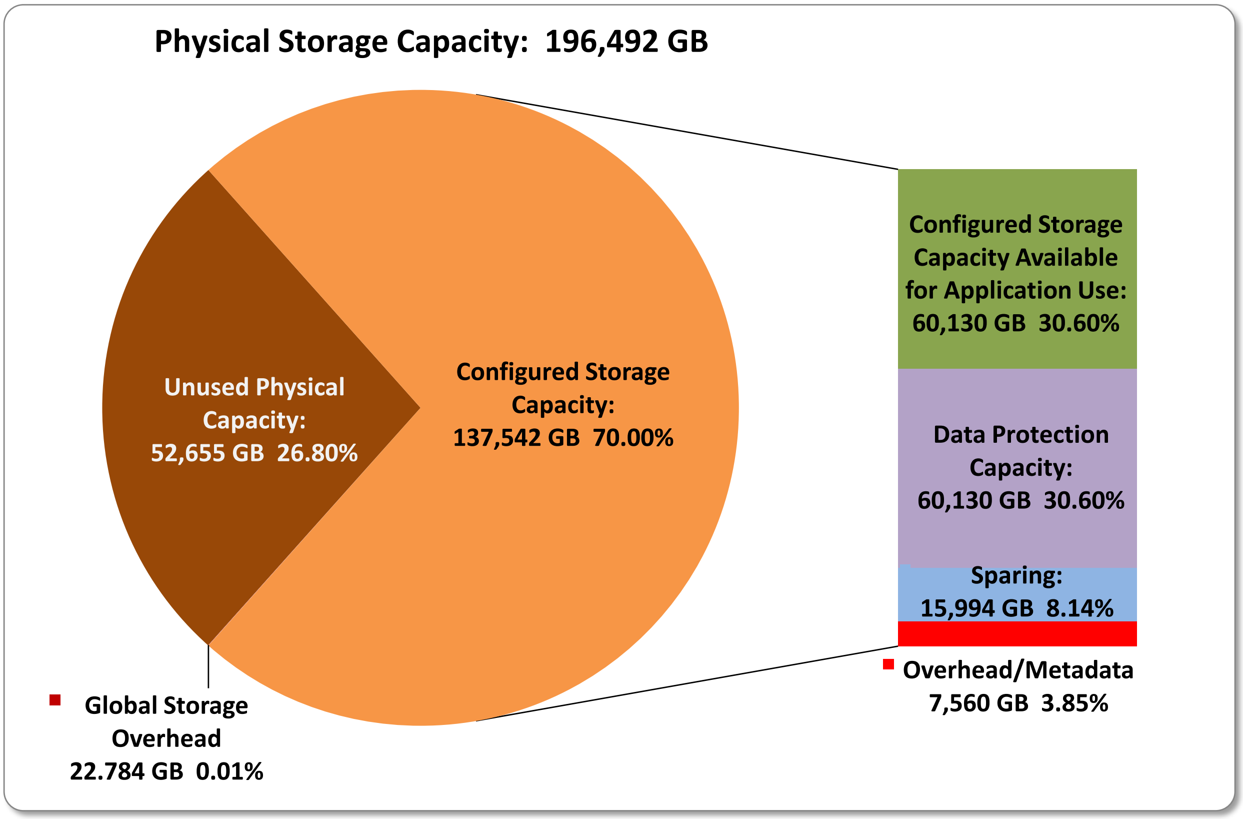


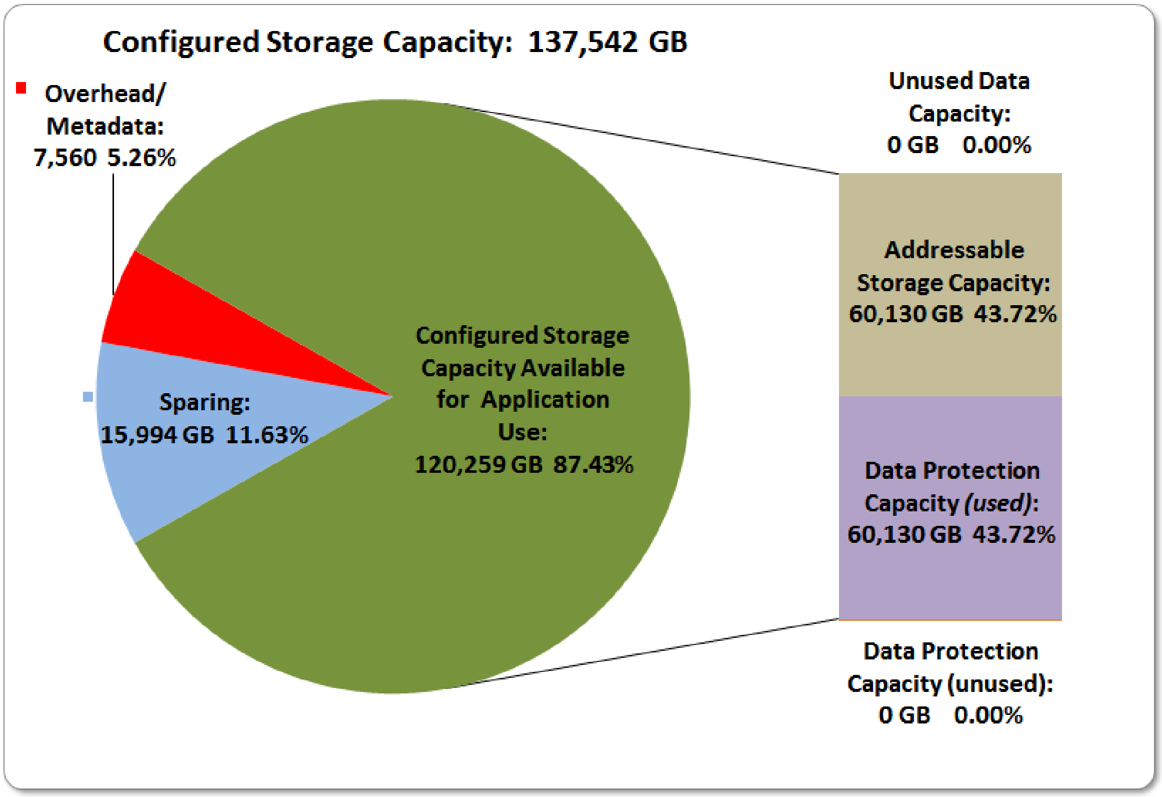
SPC-2 Reported Data Kaminario K2 K2F00000700				
SPC-2 MBPS™	**SPC-2 Price-Performance**	**ASU Capacity (GB)**	**Total Price**	**Data Protection Level**
33,477.03	$29.79	60,129.542	$997,348.00	Protected 2 *(K-RAID)*
The above SPC-2 MBPS™ value represents the aggregate data rate of all three SPC-2 workloads: Large File Processing (LFP), Large Database Query (LDQ), and Video On Demand (VOD)				
SPC-2 Large File Processing (LFP) Reported Data				
	Data Rate *(MB/second)*	**Number of Streams**	**Data Rate per Stream**	**Price-Performance**
LFP Composite	31,884.08			$31.28
Write Only:				
1024 KiB Transfer	26,086.14	85	306.90	
256 KiB Transfer	26,363.90	200	131.82	
Read-Write:				
1024 KiB Transfer	30,496.95	160	190.61	
256 KiB Transfer	32,081.35	460	69.74	
Read Only:				
1024 KiB Transfer	34,591.57	130	266.09	
256 KiB Transfer	41,684.59	475	87.76	
The above SPC-2 Data Rate value for LFP Composite represents the aggregate performance of all three LFP Test Phases: (Write Only, Read-Write, and Read Only).				
SPC-2 Large Database Query (LDQ) Reported Data				
	Data Rate *(MB/second)*	**Number of Streams**	**Data Rate per Stream**	**Price-Performance**
LDQ Composite	37,089.66			$26.89
1024 KiB Transfer Size				
4 I/Os Outstanding	30,131.38	35	860.90	
1 I/O Outstanding	34,692.16	135	256.98	
64 KiB Transfer Size				
4 I/Os Outstanding	42,855.61	460	93.16	
1 I/O Outstanding	40,679.48	1,350	30.13	
The above SPC-2 Data Rate value for LDQ Composite represents the aggregate performance of the two LDQ Test Phases: (1024 KiB and 64 KiB Transfer Sizes).				
SPC-2 Video On Demand (VOD) Reported Data				
	Data Rate *(MB/second)*	**Number of Streams**	**Data Rate per Stream**	**Price-Performance**
	31,457.36	40,000	0.79	$31.70

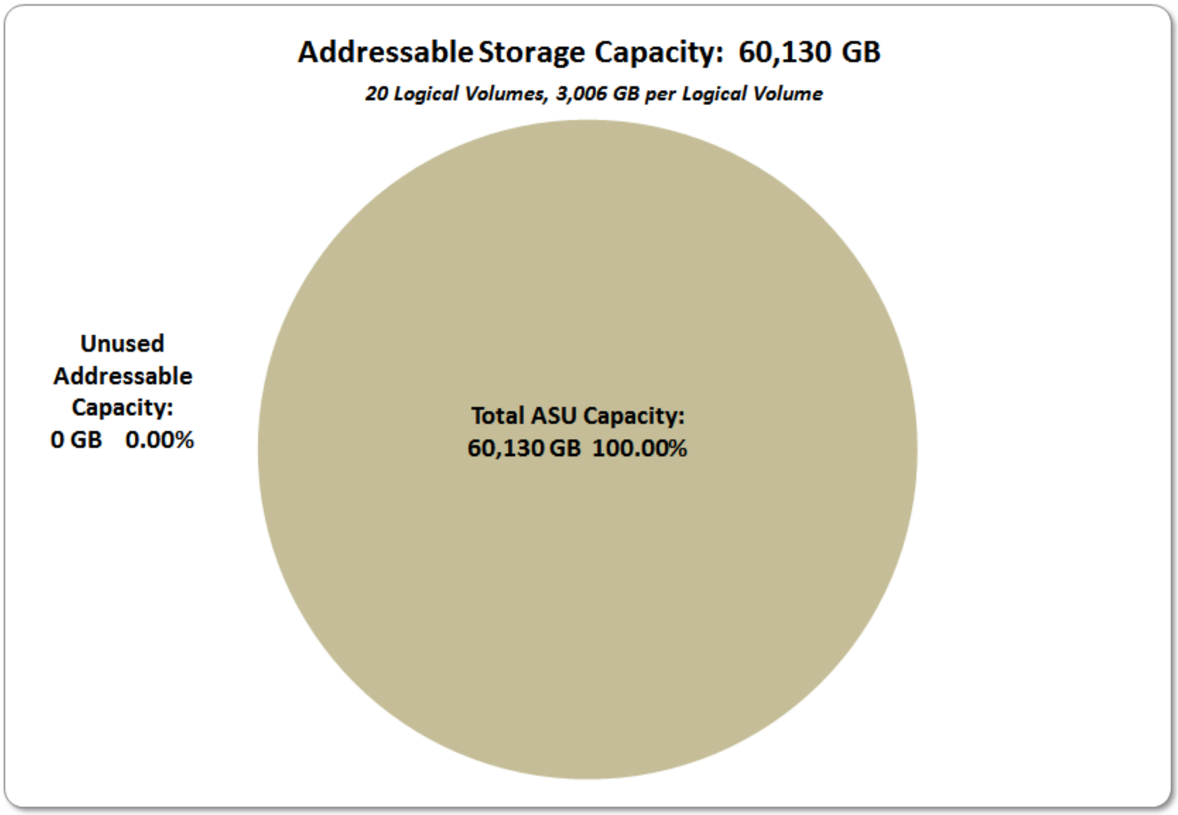
Storage Capacities, Relationships and Utilization

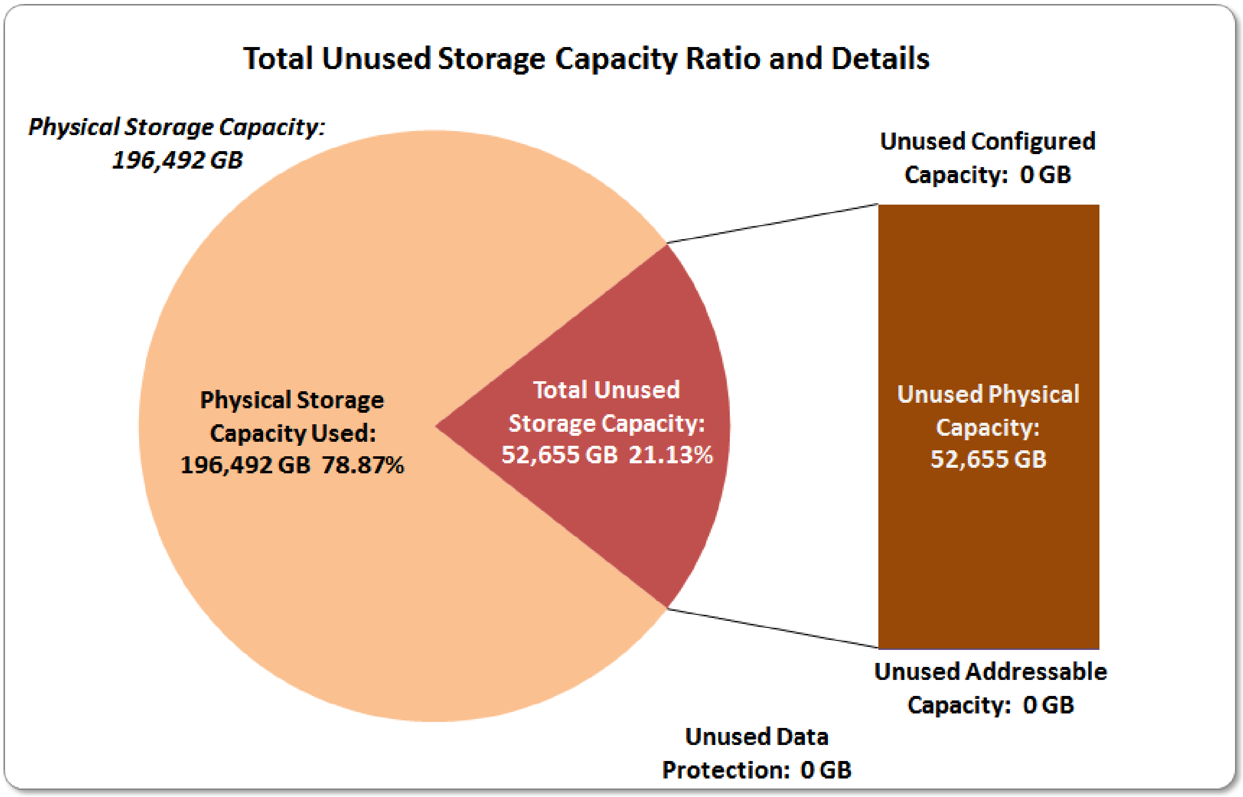
The following four charts and table document the various storage capacities, used in this benchmark, and their relationships, as well as the storage utilization values required to be reported.

The capacity values in each of the following four charts are listed as integer values, for readability, rather than the decimal values listed elsewhere in this document.




Configured Storage Capacity: 137,542 GB
🟥 Overhead/ Metadata: 7,560 5.26%
Sparing: 15,994 GB 11.63%
Configured Storage Capacity Available for Application Use: 120,259 GB 87.43%
Unused Data Capacity: 0 GB 0.00%
Addressable Storage Capacity: 60,130 GB 43.72%
Data Protection Capacity *(used)*: 60,130 GB 43.72%
Data Protection Capacity (unused): 0 GB 0.00%


Addressable Storage Capacity: 60,130 GB
20 Logical Volumes, 3,006 GB per Logical Volume
Unused Addressable Capacity: 0 GB 0.00%
Total ASU Capacity: 60,130 GB 100.00%



SPC-2 Storage Capacity Utilization	
Application Utilization	30.60%
Protected Application Utilization	61.20%
Unused Storage Ratio	26;80%

Application Utilization: Total ASU Capacity *(60,129.542 GB)* divided by Physical Storage Capacity *(196,491.768 GB).*

Protected Application Utilization: Total ASU Capacity *(60,129.542 GB)* plus total Data Protection Capacity *(60,1129.542 GB)* minus unused Data Protection Capacity *(0.000 GB)* divided by Physical Storage Capacity *(196,491.768 GB).*

Unused Storage Ratio: Total Unused Capacity *(52,655.440 GB)* divided by Physical Storage Capacity *(196,491.768 GB)* and may not exceed 45%.

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

Priced Storage Configuration Pricing

Quantity	Item	Description	Unit Price	Price
1	K2F000000700**	Kaminario K2 Flash 7 K-Blocks with 86.49TB total usable capacity		730,000.00
1	Three years maintenance	4 hours mission ciritcal		255,000.00
56	T54-M11FF-10	WesternWire FC cable LC-LC 3m	8.00	448.00
28	QME2572	QLogic QME2572 8Gbps Fibre Channel I/O Card	425.00	11,900.00
	Total System Price:			**997,348.00**

The above pricing includes the following:

- Acknowledgement of new and existing hardware and/or software problems within four 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.

K2F000000700** Line Item Components

The K2F line item in the above pricing includes the following components:

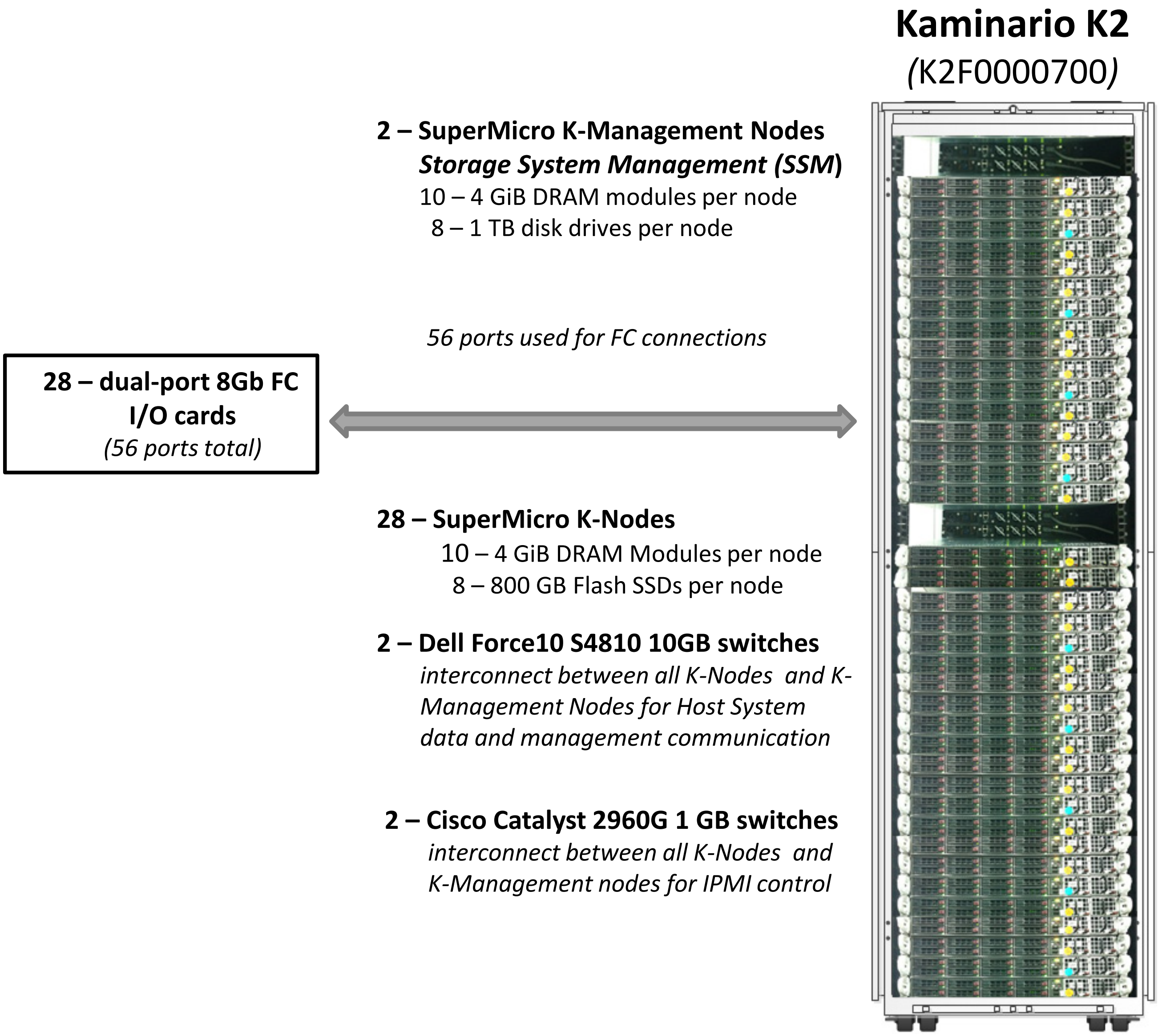
- **28 K-Nodes**: *SuperMicro SYS-1027R-72BRFTP1-EI007*:
  - Each K-Node includes eight 800 GB solid state storage devices (SSD), which provide the storage capacity for the primary and mirror SPC-1 ASUs.
  - Each K-node also runs an IO-director process responsible for exposing the data volumes to the Host Systems, connected via Fibre Channel.
- **2 K-Management Nodes, Storage System Management (SSM)** *SuperMicro SYS-1027R-72BRFTP1-EI007.*

  The SSM modules provide storage installation, configuration and monitoring functionality. Each SSM module included eight 1 TB HDDs that serve as spare backup capacity for the system.
- **2 Dell Force10 S4810 10GB switches** - Interconnects all K-nodes for the purpose of sending Host System data between the K-nodes and for supporting management communication.
- **2 Cisco Catalyst 2960G 1 GB switches** - Interconnects all K-nodes to the K-management node for the purpose IPMI protocol control over the K-nodes.
- **1 Rack:** Used to house all of the above components.

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

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

Priced Storage Configuration Diagram



Kaminario K2

(K2F0000700)

2 – SuperMicro K-Management Nodes
Storage System Management (SSM)
10 – 4 GiB DRAM modules per node
8 – 1 TB disk drives per node

56 ports used for FC connections

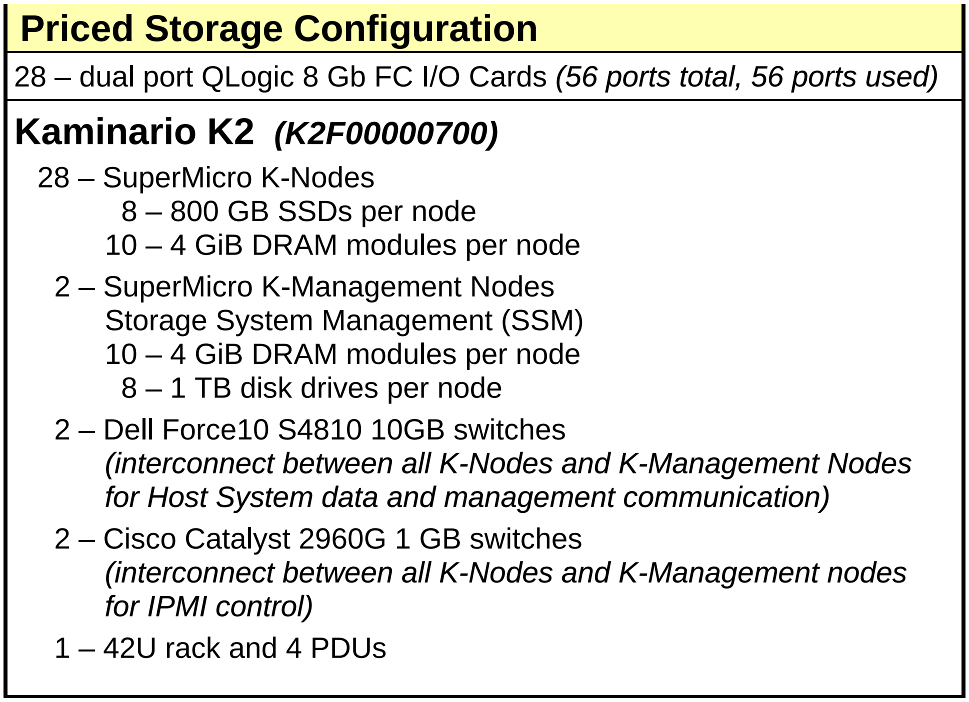
28 – dual-port 8Gb FC I/O cards
(56 ports total)

28 – SuperMicro K-Nodes
10 – 4 GiB DRAM Modules per node
8 – 800 GB Flash SSDs per node

2 – Dell Force10 S4810 10GB switches
interconnect between all K-Nodes and K-Management Nodes for Host System data and management communication

2 – Cisco Catalyst 2960G 1 GB switches
interconnect between all K-Nodes and K-Management nodes for IPMI control

Priced Storage Configuration Components



Priced Storage Configuration

28 – dual port QLogic 8 Gb FC I/O Cards *(56 ports total, 56 ports used)*

Kaminario K2 ***(K2F00000700)***

28 – SuperMicro K-Nodes
8 – 800 GB SSDs per node
10 – 4 GiB DRAM modules per node

2 – SuperMicro K-Management Nodes
Storage System Management (SSM)
10 – 4 GiB DRAM modules per node
8 – 1 TB disk drives per node

2 – Dell Force10 S4810 10GB switches
(interconnect between all K-Nodes and K-Management Nodes for Host System data and management communication)

2 – Cisco Catalyst 2960G 1 GB switches
(interconnect between all K-Nodes and K-Management nodes for IPMI control)

1 – 42U rack and 4 PDUs

CONFIGURATION INFORMATION

This portion of the Full Disclosure Report documents and illustrates the detailed information necessary to recreate the Benchmark Configuration (BC), including the Tested Storage Configuration (TSC), so that the SPC-2 benchmark result produced by the BC may be independently reproduced.

In each of the following sections of this document, the appropriate Full Disclosure Report requirement, from the SPC-2 benchmark specification, is stated in italics followed by the information to fulfill the stated requirement.

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

Clause 10.6.6

The FDR will contain a one page BC/TSC diagram that illustrates all major components of the BC/TSC.

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

Storage Network Configuration

Clause 10.6.6.1

If a storage network was configured as a part of the Tested Storage Configuration and the Benchmark Configuration described in Clause 10.6.6 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 10.11.

The Benchmark Configuration (BC)/Tested Storage Configuration (TSC) was configured with local storage and, as such, did not employ a storage network.

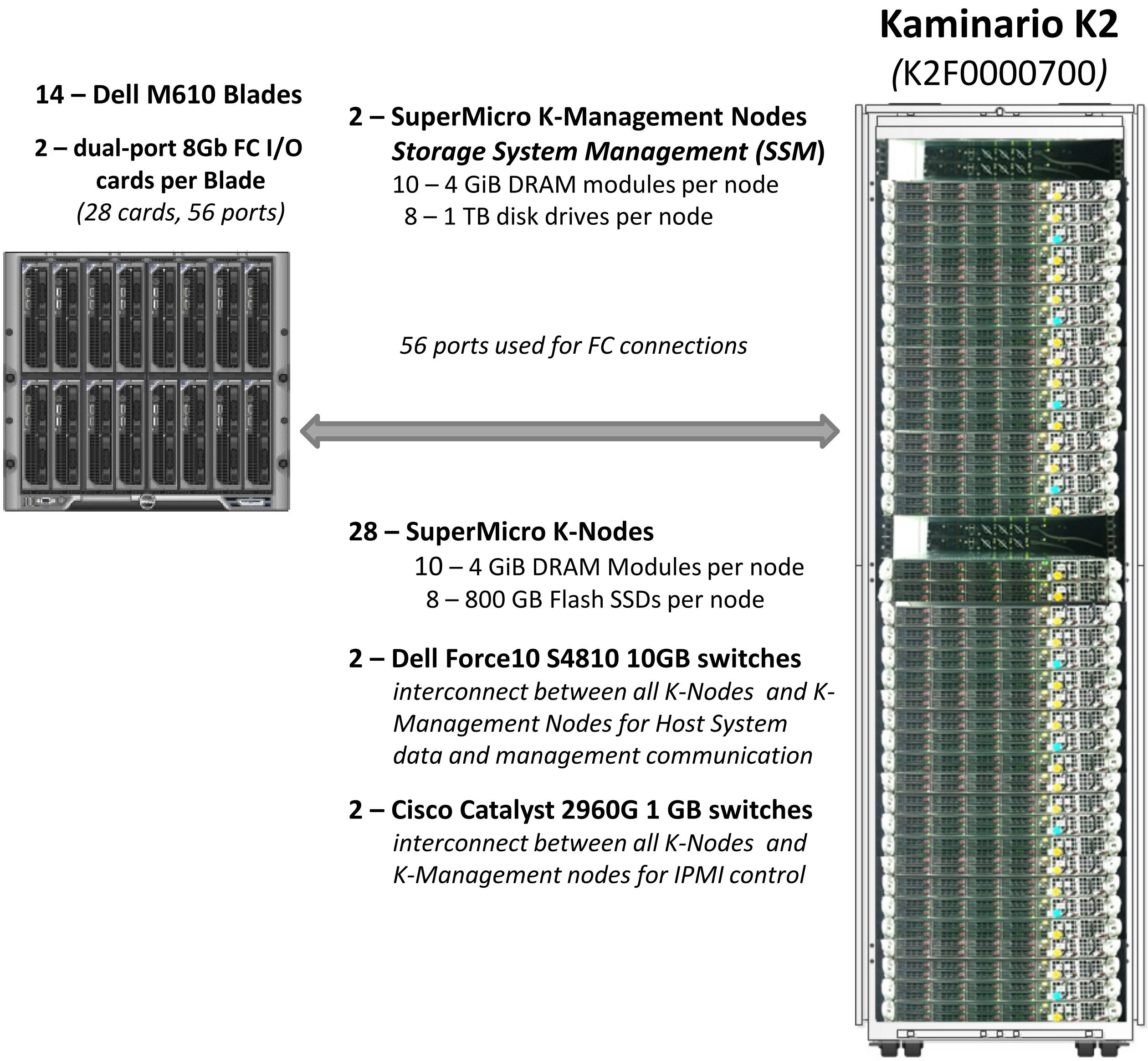
Host System and Tested Storage Configuration Table

Clause 10.6.6.2

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

The components that comprise each Host System and the Tested Storage Configuration are listed in the table that appears on page 23 (*Host System and Tested Storage Configuration Components*).

Benchmark Configuration/Tested Storage Configuration Diagram



Kaminario K2

*(*K2F0000700*)*

14 – Dell M610 Blades

2 – dual-port 8Gb FC I/O cards per Blade

(28 cards, 56 ports)

2 – SuperMicro K-Management Nodes

Storage System Management (SSM**)**

10 – 4 GiB DRAM modules per node

8 – 1 TB disk drives per node

56 ports used for FC connections

28 – SuperMicro K-Nodes

10 – 4 GiB DRAM Modules per node

8 – 800 GB Flash SSDs per node

2 – Dell Force10 S4810 10GB switches

interconnect between all K-Nodes and K-Management Nodes for Host System data and management communication

2 – Cisco Catalyst 2960G 1 GB switches

interconnect between all K-Nodes and K-Management nodes for IPMI control

Host System and Tested Storage Configuration Components

Host Systems

14 – Dell M610 Blades, each with:

- 2 – 6 core Intel® Xeon® Processors @ 2.53 GHz
- 96 GB main memory
- VMware vSphere ESX5.1.0
- Single guest VM: Windows 2008 Server R2 w/SP1
- Windows 2008 MPIO

Tested Storage Configuration (TSC)

28 – dual port QLogic 8 Gb FC I/O Cards *(56 ports total, 56 ports used)*

Kaminario K2 ***(K2F00000700)***

- 28 – SuperMicro K-Nodes
  - 8 – 800 GB SSDs per node
  - 10 – 4 GiB DRAM modules per node
- 2 – SuperMicro K-Management Nodes
  - Storage System Management (SSM)
  - 10 – 4 GiB DRAM modules per node
  - 8 – 1 TB disk drives per node
- 2 – Dell Force10 S4810 10GB switches
  (interconnect between all K-Nodes and K-Management Nodes for Host System data and management communication)
- 2 – Cisco Catalyst 2960G 1 GB switches
  (interconnect between all K-Nodes and K-Management nodes for IPMI control)
- 1 – 42U rack and 4 PDUs

Customer Tunable Parameters and Options

Clause 10.6.7.1

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 68 contains the customer tunable parameters and options that have been altered from their default values for this benchmark.

Tested Storage Configuration (TSC) Creation and Configuration

Clause 10.6.7.2

The Full Disclosure Report must include sufficient information to recreate the logical representation of the Tested Storage Configuration (TSC). In addition to customer tunable parameters and options (Clause10.6.6.1), 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 10.6.5.7 and the Storage Network Configuration Diagram in Clause 10.6.5.8.*
  - ➢ *The logical representation of the TSC, configured from the above components that will be presented to the SPC-2 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 69 contains the detailed information that describes how to create and configure the logical TSC.

SPC-2 Workload Generator Storage Configuration

Clause 10.6.7.3

The Full Disclosure Report will include all SPC-2 Workload Generator storage configuration commands and parameters used in the SPC-2 benchmark measurement.

The SPC-2 Workload Generator storage configuration commands and parameters for this measurement appear in Appendix D: SPC-2 Workload Generator Storage Commands and Parameter Files on page 72.

ASU Pre-Fill

Clause 6.3.3

The SPC-2 ASU is required to be completely filled with specified content prior to the execution of audited SPC-2 Tests. The content is required to consist of random data pattern such as that produced by an SPC recommended tool.

…

Clause 6.3.3.3

The required ASU pre-fill must be executed as the first step in the uninterrupted benchmark execution sequence described in Clause 6.4.2. That uninterrupted sequence will consist of: ASU Pre-Fill, Large File Processing, Large Database Query, Video on Demand Delivery and Persistence Test Run 1. The only exception to this requirement is described in Clause 6.3.3.4.

Clause 6.3.3.4

If approved by the Auditor, the Test Sponsor may complete the required ASU pre-fill prior to the execution of the audited SPC-2 Tests and not as part of the SPC-2 Test execution sequence.

The Auditor will verify the required random data pattern content in the ASU prior to the execution of the audited SPC-2 Tests. If that verification fails, the Test Sponsor is required to reload the specified content to the ASU.

The configuration file used to complete the required ASU pre-fill appears in Appendix D: SPC-2 Workload Generator Storage Commands and Parameter Files on page 72.

SPC-2 DATA REPOSITORY

This portion of the Full Disclosure Report presents the detailed information that fully documents the various SPC-2 storage capacities and mappings used in the Tested Storage Configuration. SPC-2 Data Repository Definitions on page 63 contains definitions of terms specific to the SPC-2 Data Repository.

In each of the following sections of this document, the appropriate Full Disclosure Report requirement, from the SPC-2 benchmark specification, is stated in italics followed by the information to fulfill the stated requirement.

SPC-2 Storage Capacities and Relationships

Clause 10.6.8.1

Two tables and four charts documenting the storage capacities and relationships of the SPC-2 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-2 Storage Capacities

The Physical Storage Capacity consisted of 196,491.768 GB consisted of the following:

- 224 solid state storage devices *(flash)*, each with a formatted capacity of 800 GB *(179,200.00 GB total)*, which contained primary and mirror copies of the SPC-2 ASU.
- 300 solid state storage devices *(DRAM)*, each with a formatted capacity of 4.295 GB *(1,288.490 total)*, which is used for caching and runtime metadata.
- 16 disk drives, each with a formatted capacity of 1,000.205 GB *(16,003.278 GB)*, which is used as sparing/backup capacity for mirroring data during a failure.

There was 58,926.951 GB (29.99%) of Unused Storage within the Physical Storage Capacity. Global Storage Overhead consisted of 22.784 GB (0.01%) of the Physical Storage Capacity. There was 0.000 GB (0.00%) of Unused Storage within the Configured Storage Capacity. The Total ASU Capacity utilized 100% of the Addressable Storage Capacity resulting in 0.000 GB (0.00%) of Unused Storage within the Addressable Storage Capacity. The Data Protection *(K-RAID)* capacity was 60,129.542 GB of which 60,129.542 GB was utilized. The total Unused Storage capacity was 58,926.951 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.

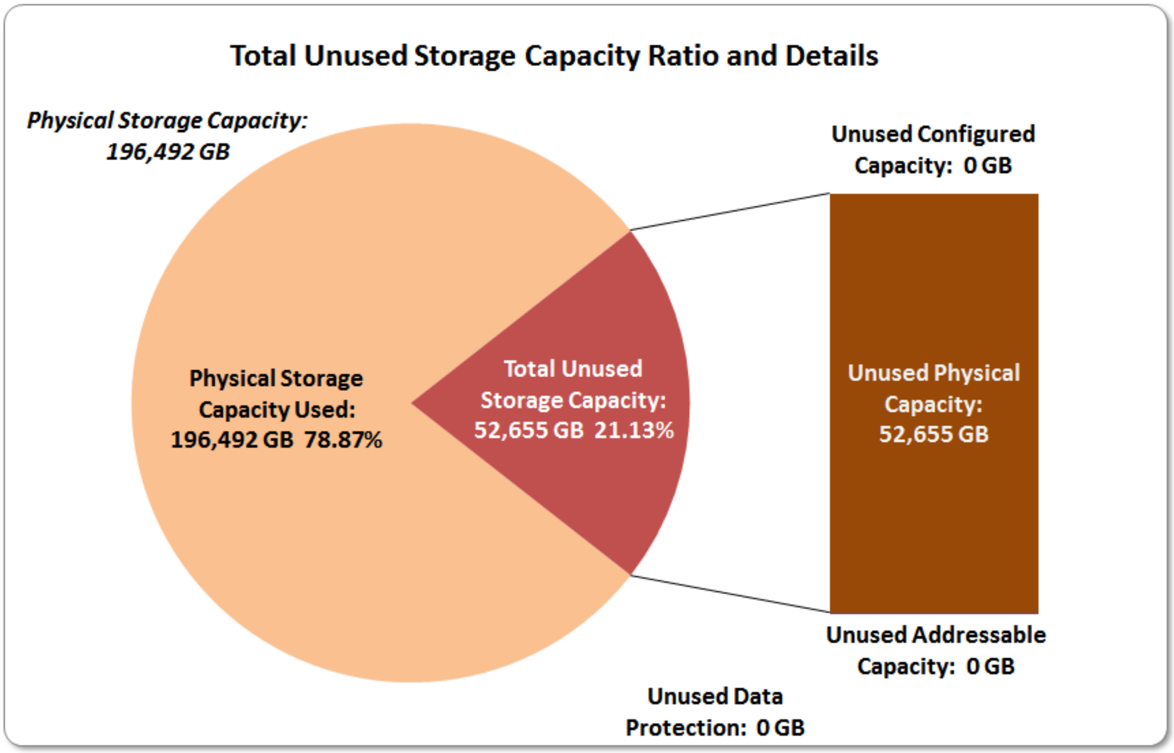
SPC-2 Storage Capacities

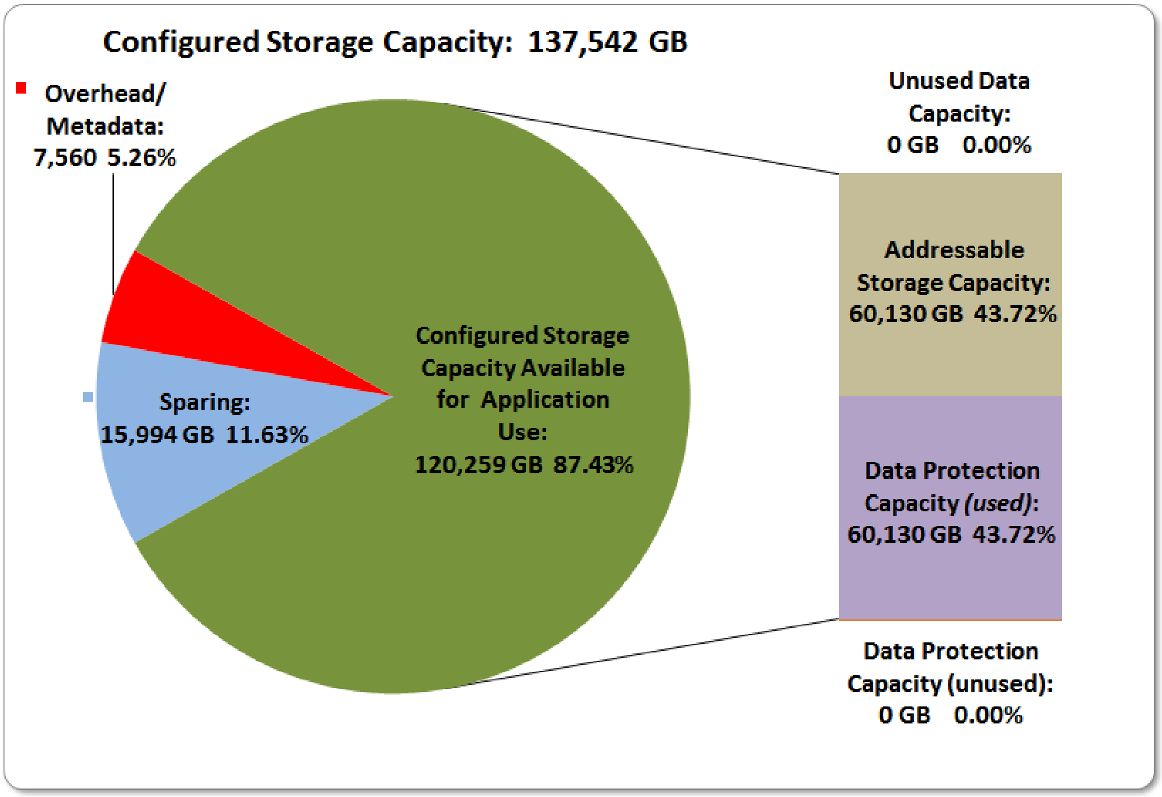
Storage Hierarchy Component	Units	Capacity
Total ASU Capacity	Gigabytes (GB)	60,129.542
Addressable Storage Capacity	Gigabytes (GB)	60,129.542
Configured Storage Capacity	Gigabytes (GB)	137,542.033
Physical Storage Capacity	Gigabytes (GB)	196,491.768
Data Protection *(K-RAID)*	Gigabytes (GB)	60,129.542
Required Storage *(sparing, metadata)*	Gigabytes (GB)	23,554.460
Global Storage Overhead	Gigabytes (GB)	22.784
Total Unused Storage	Gigabytes (GB)	52,655.440

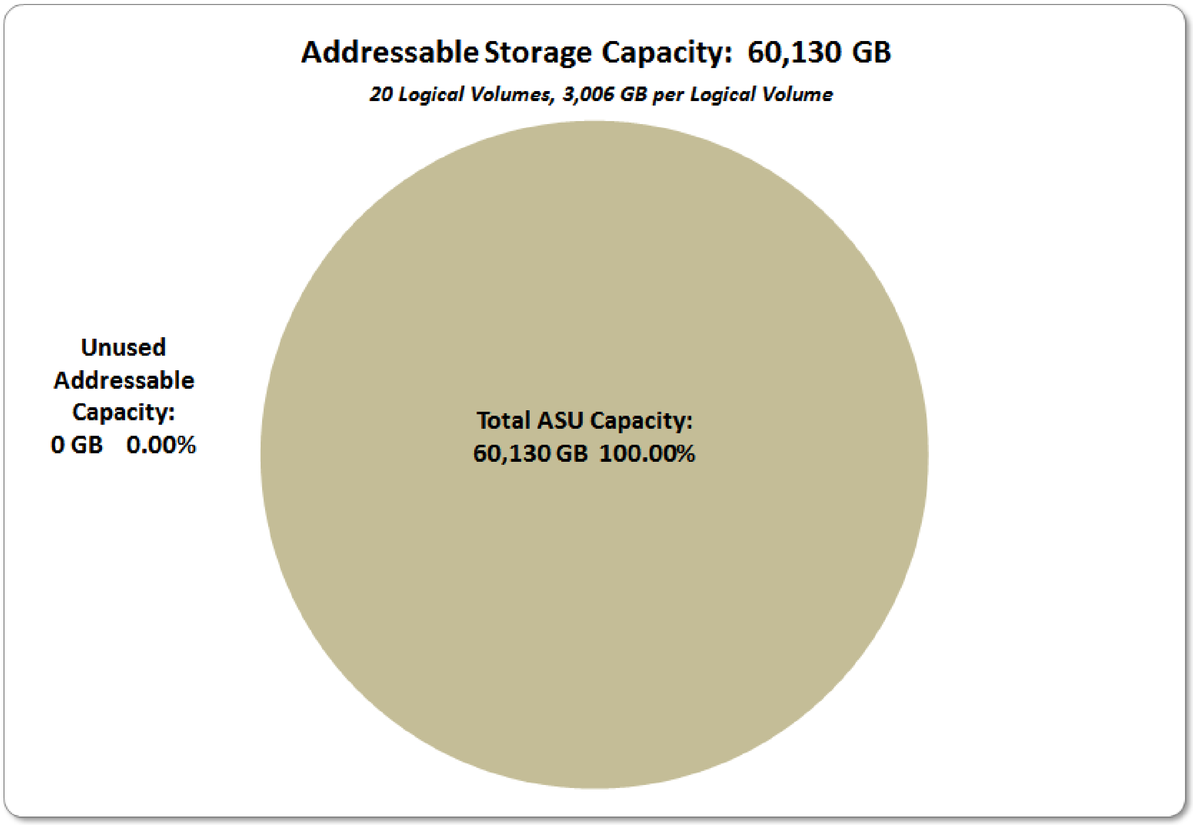
SPC-2 Storage Hierarchy Ratios

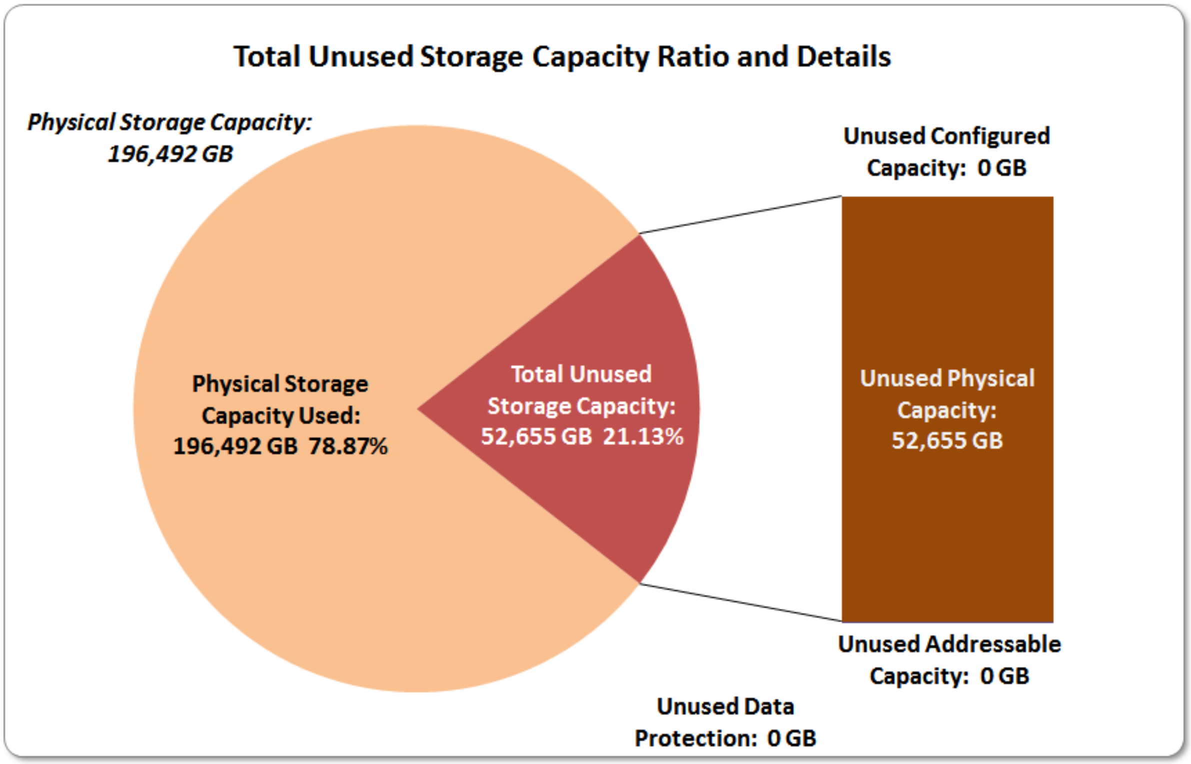
	Addressable Storage Capacity	Configured Storage Capacity	Physical Storage Capacity
Total ASU Capacity	100.00%	41.81%	30.60%
Data Protection *(mirroring)*		41.81%	30.60%
Addressable Storage Capacity		41..81%	30.60%
Required Storage *(spares, overhead)*		16.38%	11.99%
Configured Storage Capacity			73.19%
Global Storage Overhead			0.01%
Unused Storage:			
Addressable	0.00%		
Configured		0.00%	
Physical			26.80%

SPC-1 Storage Capacity Charts






Addressable Storage Capacity: 60,130 GB
20 Logical Volumes, 3,006 GB per Logical Volume
Unused Addressable Capacity: 0 GB 0.00%
Total ASU Capacity: 60,130 GB 100.00%


Total Unused Storage Capacity Ratio and Details
Physical Storage Capacity: 196,492 GB
Unused Configured Capacity: 0 GB
Physical Storage Capacity Used: 196,492 GB 78.87%
Total Unused Storage Capacity: 52,655 GB 21.13%
Unused Physical Capacity: 52,655 GB
Unused Addressable Capacity: 0 GB
Unused Data Protection: 0 GB

Storage Capacity Utilization

Clause 10.6.8.2

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

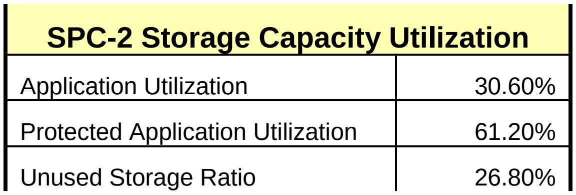
***Application Utilization** is defined as Total ASU Capacity divided by Physical Storage Capacity.*

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-2 Storage Capacity Utilization	
Application Utilization	30.60%
Protected Application Utilization	61.20%
Unused Storage Ratio	26.80%

Logical Volume Capacity and ASU Mapping

Clause 10.6.8.3

A table illustrating the capacity of the Application Storage Unit (ASU) and the mapping of Logical Volumes to ASU will be provided in the FDR. Capacity must be stated in gigabytes (GB) as a value with a minimum of two digits to the right of the decimal point. Each Logical Volume will be sequenced in the table from top to bottom per its position in the contiguous address space of the ASU. Each Logical Volume entry will list its total capacity, the portion of that capacity used for the ASU, and any unused capacity.

Logical Volume (LV) Capacity and Mapping			
ASU (60,129.542 GB)			
	Total Capacity *(GB)*	Capacity Used *(GB)*	Capacity Unused *(GB)*
Logical Volumes 1-20	3,006.477 per LV	3,006.477 per LV	0.000 per LV

See the Storage Definition (sd) entries in Appendix D: SPC-2 Workload Generator Storage Commands and Parameter Files on page 72 for more detailed configuration information.

SPC-2 BENCHMARK EXECUTION RESULTS

This portion of the Full Disclosure Report documents the results of the various SPC-2 Tests, Test Phases, Test Run Sequences, and Test Runs. An SPC-2 glossary on page 63 contains definitions of terms specific to the SPC-2 Data Repository.

In each of the following sections of this document, the appropriate Full Disclosure Report requirement, from the SPC-2 benchmark specification, is stated in italics followed by the information to fulfill the stated requirement.

SPC-2 Tests, Test Phases, Test Run Sequences, and Test Runs

The SPC-2 benchmark consists of the following Tests, Test Phases, Test Run Sequences, and Test Runs:

- **Data Persistence Test**
  - Data Persistence Test Run 1
  - Data Persistence Test Run 2

- **Large File Processing Test**
  - WRITE ONLY Test Phase
    - Test Run Sequence 1
      - ✓ Test Run 1 – 1024 KiB Transfer – maximum number of Streams
      - ✓ Test Run 2 – 1024 KiB Transfer – 50% of Test Run 1's Streams value
      - ✓ Test Run 3 – 1024 KiB Transfer – 25% of Test Run 1's Streams value
      - ✓ Test Run 4 – 1024 KiB Transfer – 12.5% of Test Run 1's Streams value
      - ✓ Test Run 5 – 1024 KiB Transfer – single (1) Stream
    - Test Run Sequence 2
      - ✓ Test Run 6 – 256 KiB Transfer – maximum number of Streams
      - ✓ Test Run 7 – 256 KiB Transfer – 50% of Test Run 6's Streams value
      - ✓ Test Run 8 – 256 KiB Transfer – 25% of Test Run 6's Streams value
      - ✓ Test Run 9 – 256 KiB Transfer – 12.5% of Test Run 6's Streams value
      - ✓ Test Run 10 – 256 KiB Transfer – single (1) Stream
  - READ-WRITE Test Phase
    - Test Run Sequence 3
      - ✓ Test Run 11 – 1024 KiB Transfer – maximum number of Streams
      - ✓ Test Run 12 – 1024 KiB Transfer – 50% of Test Run 11's Streams value
      - ✓ Test Run 13 – 1024 KiB Transfer – 25% of Test Run 11's Streams value
      - ✓ Test Run 14 – 1024 KiB Transfer – 12.5% of Test Run 11's Streams value
      - ✓ Test Run 15 – 1024 KiB Transfer – single (1) Stream
    - Test Run Sequence 4
      - ✓ Test Run 16 – 256 KiB Transfer – maximum number of Streams
      - ✓ Test Run 17 – 256 KiB Transfer – 50% of Test Run 16's Streams value
      - ✓ Test Run 18 – 256 KiB Transfer – 25% of Test Run 16's Streams value
      - ✓ Test Run 19 – 256 KiB Transfer – 12.5% of Test Run 16's Streams value
      - ✓ Test Run 20 – 256 KiB Transfer – single (1) Stream
  - READ ONLY Test Phase
    - Test Run Sequence 5
      - ✓ Test Run 21 – 1024 KiB Transfer – maximum number of Streams

  - ✓ Test Run 22 – 1024 KiB Transfer – 50% of Test Run 21's Streams value
  - ✓ Test Run 23 – 1024 KiB Transfer – 25% of Test Run 21's Streams value
  - ✓ Test Run 24 – 1024 KiB Transfer – 12.5% of Test Run 21's Streams value
  - ✓ Test Run 25 – 1024 KiB Transfer – single (1) Stream
- Test Run Sequence 6
  - ✓ Test Run 26 – 256 KiB Transfer – maximum number of Streams
  - ✓ Test Run 27 – 256 KiB Transfer – 50% of Test Run 26's Streams value
  - ✓ Test Run 28 – 256 KiB Transfer – 25% of Test Run 26's Streams value
  - ✓ Test Run 29 – 256 KiB Transfer – 12.5% of Test Run 26's Streams value
  - ✓ Test Run 30 – 256 KiB Transfer – single (1) Stream

- ▪ **Large Database Query Test**
  - ➢ 1024 KIB TRANSFER SIZE Test Phase
    - Test Run Sequence 1
      - ✓ Test Run 1 – 4 I/O Requests Outstanding – maximum number of Streams
      - ✓ Test Run 2 – 4 I/O Requests Outstanding – 50% of Test Run 1's Streams value
      - ✓ Test Run 3 – 4 I/O Requests Outstanding – 25% of Test Run 1's Streams value
      - ✓ Test Run 4 – 4 I/O Requests Outstanding – 12.5% of Test Run 1's Streams value
      - ✓ Test Run 5 – 4 I/O Requests Outstanding – single (1) Stream
    - Test Run Sequence 2
      - ✓ Test Run 6 – 1 I/O Request Outstanding – maximum number of Streams
      - ✓ Test Run 7 – 1 I/O Request Outstanding – 50% of Test Run 6's Streams value
      - ✓ Test Run 8 – 1 I/O Request Outstanding – 25% of Test Run 6's Streams value
      - ✓ Test Run 9 – 1 I/O Request Outstanding – 12.5% of Test Run 6's Streams value
      - ✓ Test Run 10 – 1 I/O Request Outstanding – single (1) Stream
  - ➢ 64 KIB TRANSFER SIZE Test Phase
    - Test Run Sequence 3
      - ✓ Test Run 11 – 4 I/O Requests Outstanding – maximum number of Streams
      - ✓ Test Run 12 – 4 I/O Requests Outstanding – 50% of Test Run 11's Streams value
      - ✓ Test Run 13 – 4 I/O Requests Outstanding – 25% of Test Run 11's Streams value
      - ✓ Test Run 14 – 4 I/O Requests Outstanding – 12.5% of Test Run 11's Streams value
      - ✓ Test Run 15 – 4 I/O Requests Outstanding – single (1) Stream
    - Test Run Sequence 4
      - ✓ Test Run 16 – 1 I/O Request Outstanding – maximum number of Streams
      - ✓ Test Run 17 – 1 I/O Request Outstanding – 50% of Test Run 16's Streams value
      - ✓ Test Run 18 – 1 I/O Request Outstanding – 25% of Test Run 16's Streams value
      - ✓ Test Run 19 – 1 I/O Request Outstanding – 12.5% of Test Run 16's Streams value
      - ✓ Test Run 20 – 1 I/O Request Outstanding – single (1) Stream

- ▪ **Video on Demand Delivery Test**
  - ➢ Video on Demand Delivery Test Run

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 Tests may be executed in any sequence.

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

Large File Processing Test

Clause 6.4.3.1

The Large File Processing Test consists of the I/O operations associated with the type of applications, in a wide range of fields, which require simple sequential processing of one or more large files. Specific examples of those types of applications include scientific computing and large-scale financial processing

Clause 6.4.3.2

The Large File Processing Test has three Test Phases, which shall be executed in the following uninterrupted sequence:

1. *WRITE ONLY*
2. *READ-WRITE*
3. *READ ONLY*

The BC shall not be restarted or manually disturbed, altered, or adjusted during the execution of the Large File Processing Test. If power is lost to the BC during this Test all results shall be rendered invalid and the Test re-run in its entirety.

Clause 10.6.9.1

The Full Disclosure Report will contain the following content for the Large File Processing Test:

1. *A listing of the SPC-2 Workload Generator commands and parameters used to execute each of the Test Runs in the Large File Processing Test.*
2. *The human readable SPC-2 Test Results File for each of the Test Runs in the Large File Processing Test.*
3. *The following three tables:*
   - *Average Data Rate: The average Data Rate, in MB per second for the Measurement Interval of each Test Run in the Large File Processing Test.*
   - *Average Data Rate per Stream: The average Data Rate per Stream, in MB per second, for the Measurement Interval of each Test Run in the Large File Processing Test.*
   - *Average Response Time: The average response time, in milliseconds (ms), for the Measurement Interval of each Test Run in the Large File Processing Test.*
4. *Average Data Rate, Average Data Rate per Stream and Average Response Time graphs as defined in Clauses 10.1.1, 10.1.2 and 10.1.3.*

SPC-2 Workload Generator Commands and Parameters

The SPC-2 Workload Generator commands and parameters for the Large File Processing Test Runs are documented in Appendix E: SPC-2 Workload Generator Execution Commands and Parameters on Page 83.

SPC-2 Test Results File

A link to the SPC-2 Test Results file generated from the Large File Processing Test Runs is listed below.

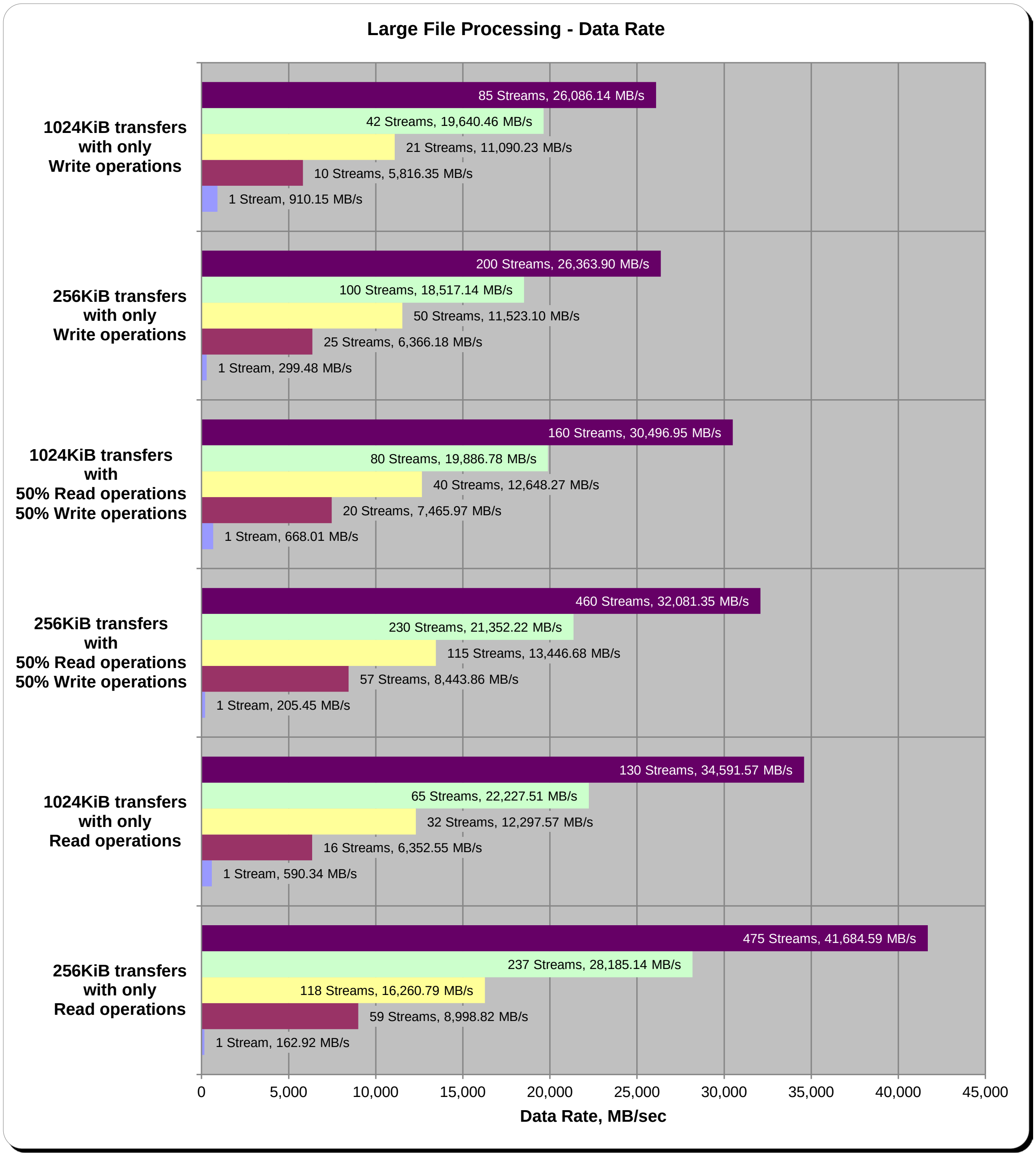
SPC-2 Large File Processing Test Results File

SPC-2 Large File Processing Average Data Rates (MB/s)

The average Data Rate (MB/s) for each Test Run in the three Test Phases of the SPC-2 Large File Processing Test is listed in the table below as well as illustrated in the following graph.

Test Run Sequence	**1 Stream**	**10 Streams**	**21 Streams**	**42 Streams**	**85 Streams**
Write 1024KiB	910.15	5,816.35	11,090.23	19,640.46	26,086.14
Test Run Sequence	**1 Stream**	**25 Streams**	**50 Streams**	**100 Streams**	**200 Streams**
Write 256KiB	299.48	6,366.18	11,523.10	18,517.14	26,363.90
Test Run Sequence	**1 Stream**	**20 Streams**	**40 Streams**	**80 Streams**	**160 Streams**
Read/Write 1024KiB	668.01	7,465.97	12,648.27	19,886.78	30,496.95
Test Run Sequence	**1 Stream**	**57 Streams**	**115 Streams**	**230 Streams**	**460 Streams**
Read/Write 256KiB	205.45	8,443.86	13,446.68	21,352.22	32,081.35
Test Run Sequence	**1 Stream**	**16 Streams**	**32 Streams**	**65 Streams**	**130 Streams**
Read 1024KiB	590.34	6,352.55	12,297.57	22,227.51	34,591.57
Test Run Sequence	**1 Stream**	**59 Streams**	**118 Streams**	**237 Streams**	**475 Streams**
Read 256KiB	162.92	8,998.82	16,260.79	28,185.14	41,684.59

SPC-2 Large File Processing Average Data Rates Graph

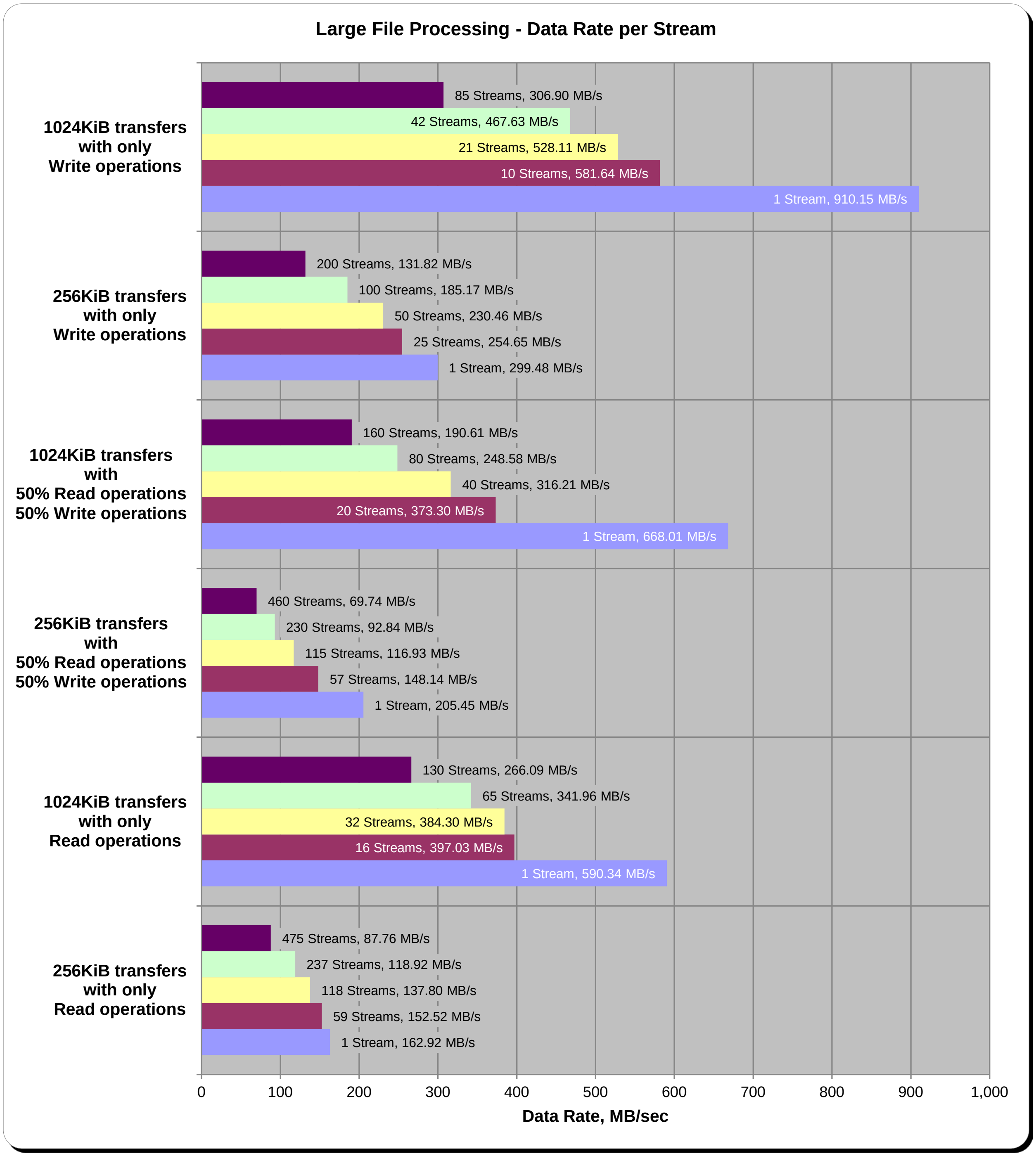


SPC-2 Large File Processing Average Data Rate per Stream

The average Data Rate per Stream for each Test Run in the three Test Phases of the SPC-2 Large File Processing Test is listed in the table below as well as illustrated in the following graph.

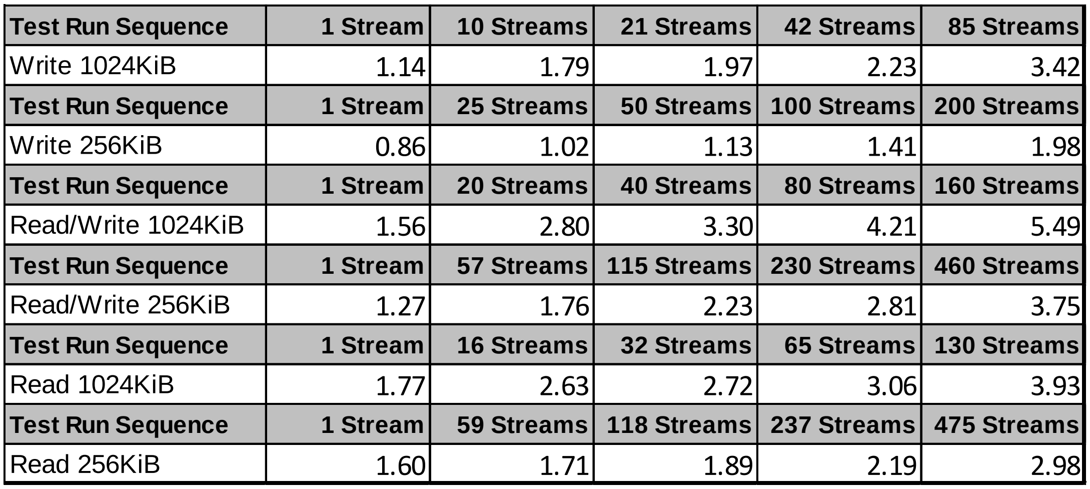
Test Run Sequence	**1 Stream**	**10 Streams**	**21 Streams**	**42 Streams**	**85 Streams**
Write 1024KiB	910.15	581.64	528.11	467.63	306.90
Test Run Sequence	**1 Stream**	**25 Streams**	**50 Streams**	**100 Streams**	**200 Streams**
Write 256KiB	299.48	254.65	230.46	185.17	131.82
Test Run Sequence	**1 Stream**	**20 Streams**	**40 Streams**	**80 Streams**	**160 Streams**
Read/Write 1024KiB	668.01	373.30	316.21	248.58	190.61
Test Run Sequence	**1 Stream**	**57 Streams**	**115 Streams**	**230 Streams**	**460 Streams**
Read/Write 256KiB	205.45	148.14	116.93	92.84	69.74
Test Run Sequence	**1 Stream**	**16 Streams**	**32 Streams**	**65 Streams**	**130 Streams**
Read 1024KiB	590.34	397.03	384.30	341.96	266.09
Test Run Sequence	**1 Stream**	**59 Streams**	**118 Streams**	**237 Streams**	**475 Streams**
Read 256KiB	162.92	152.52	137.80	118.92	87.76

SPC-2 Large File Processing Average Data Rate per Stream Graph



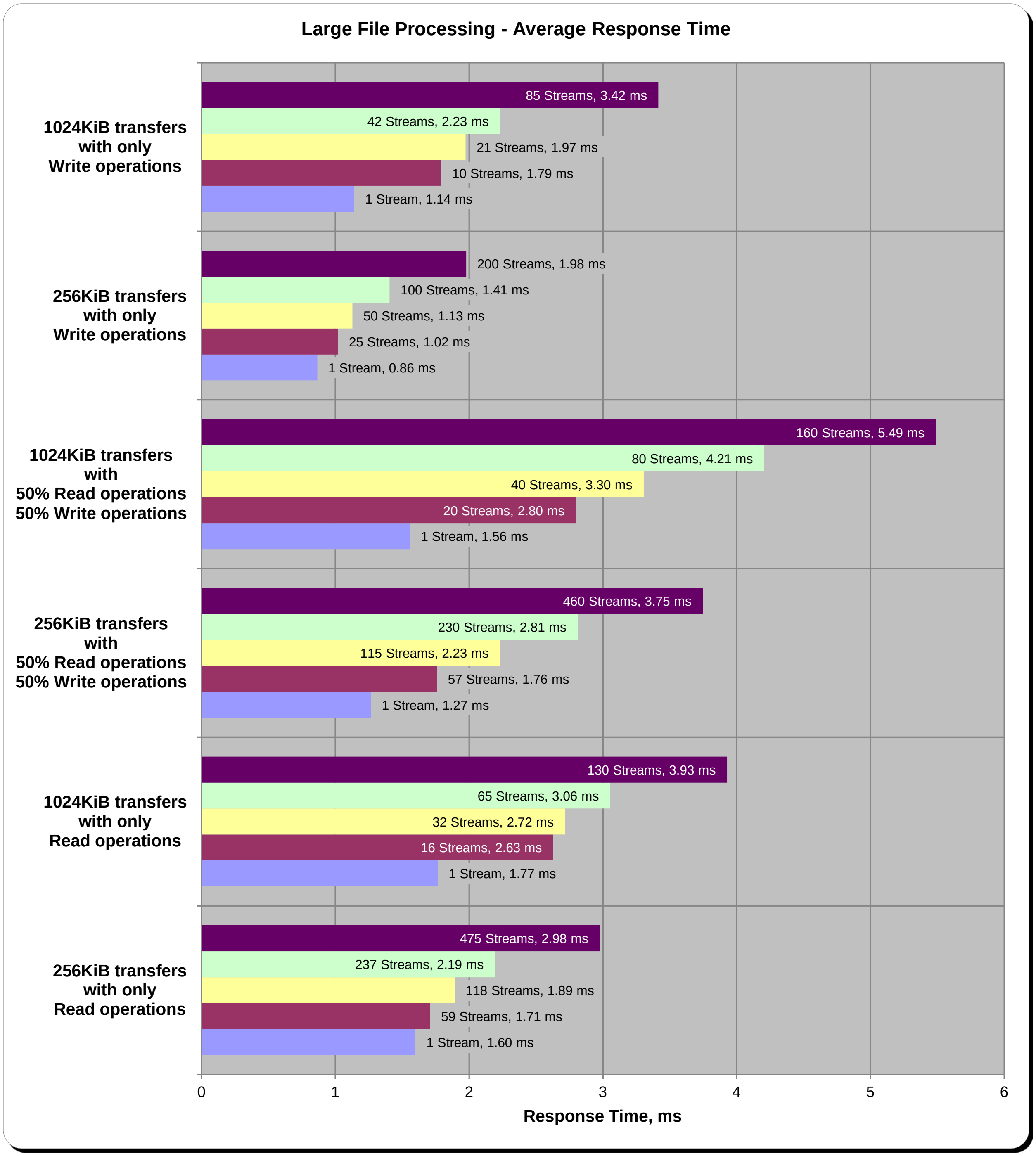
SPC-2 Large File Processing Average Response Time

The average Response Time, milliseconds (ms), for each Test Run in the three Test Phases of the SPC-2 Large File Processing Test is listed in the table below as well as illustrated in the following graph.



Test Run Sequence	1 Stream	10 Streams	21 Streams	42 Streams	85 Streams
Write 1024KiB	1.14	1.79	1.97	2.23	3.42
Test Run Sequence	**1 Stream**	**25 Streams**	**50 Streams**	**100 Streams**	**200 Streams**
Write 256KiB	0.86	1.02	1.13	1.41	1.98
Test Run Sequence	**1 Stream**	**20 Streams**	**40 Streams**	**80 Streams**	**160 Streams**
Read/Write 1024KiB	1.56	2.80	3.30	4.21	5.49
Test Run Sequence	**1 Stream**	**57 Streams**	**115 Streams**	**230 Streams**	**460 Streams**
Read/Write 256KiB	1.27	1.76	2.23	2.81	3.75
Test Run Sequence	**1 Stream**	**16 Streams**	**32 Streams**	**65 Streams**	**130 Streams**
Read 1024KiB	1.77	2.63	2.72	3.06	3.93
Test Run Sequence	**1 Stream**	**59 Streams**	**118 Streams**	**237 Streams**	**475 Streams**
Read 256KiB	1.60	1.71	1.89	2.19	2.98

SPC-2 Large File Processing Average Response Time Graph



Large File Processing Test – WRITE ONLY Test Phase

Clause 10.6.9.1.1

1. *A table that will contain the following information for each ”WRITE ONLY, 1024 KiB Transfer Size” Test Run:*
    - *The number of Streams specified.*
    - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
2. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the “WRITE ONLY, 1024 KiB Transfer Size” Test Runs as specified in Clauses 10.1.4 – 10.1.6.*
3. *A table that will contain the following information for each ”WRITE ONLY, 256 KiB Transfer Size” Test Run:*
    - *The number of Streams specified.*
    - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
4. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the “WRITE ONLY, 256 KiB Transfer Size” Test Runs as specified in Clauses 10.1.4 – 10.1.6.*

A hyperlink for each of the above tables and graphs may appear in the FDR to provide access to the table or graph.

A hyperlink to a table with the SPC-2 “Large File Processing/WRITE ONLY/1024 KiB Transfer Size” Test Run data appears on the next page. That entry is followed by hyperlinks to graphs illustrating the average Data Rate, average Data Rate per Stream, and average Response Time produced by the same Test Runs. The table and graphs present the data at five-second intervals.

Immediately following the above SPC-2 “Large File Processing/WRITE ONLY/1024 KiB Transfer Size” entries will be hyperlinks for SPC-2 “Large File Processing/WRITE ONLY/256 KiB Transfer Size” table and graphs. The table contains the Test Run data and the graphs illustrate the average Data Rate, average Data Rate per Stream, and average Response Time produced by the Test Runs.

SPC-2 “Large File Processing/WRITE ONLY/1024 KiB Transfer Size” Test Run Data

SPC-2 “Large File Processing/WRITE ONLY/1024 KiB Transfer Size” Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 “Large File Processing/WRITE ONLY/1024 KiB Transfer Size” Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 “Large File Processing/WRITE ONLY/1024 KiB Transfer Size” graphs
(four pages, 1 graph per page)

SPC-2 “Large File Processing/WRITE ONLY/256 KiB Transfer Size” Test Run Data

SPC-2 “Large File Processing/WRITE ONLY/256 KiB Transfer Size” Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 “Large File Processing/WRITE ONLY/256 KiB Transfer Size” Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 “Large File Processing/WRITE ONLY/256 KiB Transfer Size” graphs
(four pages, 1 graph per page)

Large File Processing Test – READ-WRITE Test Phase

Clause 10.6.9.1.2

1. *A table that will contain the following information for each "READ-WRITE, 1024 KiB Transfer Size" Test Run:*
   - *The number of Streams specified.*
   - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
2. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the "READ-WRITE, 1024 KiB Transfer Size" Test Runs as specified in Clauses 10.1.4 – 10.1.6.*
3. *A table that will contain the following information for each " READ-WRITE, 256 KiB Transfer Size" Test Run:*
   - *The number of Streams specified.*
   - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
4. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the "READ-WRITE, 256 KiB Transfer Size" Test Runs as specified in Clauses 10.1.4 – 10.1.6.*

A hyperlink for each of the above tables and graphs may appear in the FDR to provide access to the table or graph.

A hyperlink to a table with the SPC-2 "Large File Processing/READ-WRITE/1024 KiB Transfer Size" Test Run data appears on the next page. That entry is followed by hyperlinks to graphs illustrating the average Data Rate, average Data Rate per Stream, and average Response Time produced by the same Test Runs. The table and graphs present the data at five-second intervals.

Immediately following the above SPC-2 "Large File Processing/READ-WRITE/1024 KiB Transfer Size" entries will be hyperlinks for SPC-2 "Large File Processing/READ-WRITE/256 KiB Transfer Size" table and graphs. The table contains the Test Run data and the graphs illustrate the average Data Rate, average Data Rate per Stream, and average Response Time produced by the Test Runs.

SPC-2 "Large File Processing/READ-WRITE/1024 KiB Transfer Size" Test Run Data

SPC-2 "Large File Processing/READ-WRITE/1024 KiB Transfer Size" Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 "Large File Processing/READ-WRITE/1024 KIB Transfer Size" Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 "Large File Processing/READ-WRITE/1024 KiB Transfer Size" graphs
(four pages, 1 graph per page)

SPC-2 "Large File Processing/READ-WRITE/256 KiB Transfer Size" Test Run Data

SPC-2 "Large File Processing/READ-WRITE/256 KiB Transfer Size" Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 "Large File Processing/READ-WRITE/256 KiB Transfer Size" Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 "Large File Processing/READ-WRITE/256 KiB Transfer Size" graphs
(four pages, 1 graph per page)

Large File Processing Test – READ ONLY Test Phase

Clause 10.6.9.1.3

1. *A table that will contain the following information for each "READ ONLY, 1024 KiB Transfer Size" Test Run:*
   - *The number of Streams specified.*
   - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
2. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the "READ ONLY, 1024 KiB Transfer Size" Test Runs as specified in Clauses 10.1.4 – 10.1.6.*
3. *A table that will contain the following information for each "READ ONLY, 256 KiB Transfer Size" Test Run:*
   - *The number of Streams specified.*
   - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
4. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the "READ ONLY, 256 KiB Transfer Size" Test Runs as specified in Clauses 10.1.4 – 10.1.6.*

A hyperlink for each of the above tables and graphs may appear in the FDR to provide access to the table or graph.

A hyperlink to a table with the SPC-2 "Large File Processing/READ ONLY/1024 KiB Transfer Size" Test Run data appears on the next page. That entry is followed by hyperlinks to graphs illustrating the average Data Rate, average Data Rate per Stream, and average Response Time produced by the same Test Runs. The table and graphs present the data at five-second intervals.

Immediately following the above SPC-2 "Large File Processing/READ ONLY/1024 KiB Transfer Size" entries will be hyperlinks for SPC-2 "Large File Processing/READ ONLY/256 KiB Transfer Size" table and graphs. The table contains the Test Run data and the graphs illustrate the average Data Rate, average Data Rate per Stream, and average Response Time produced by the Test Runs.

SPC-2 "Large File Processing/READ ONLY/1024 KiB Transfer Size" Test Run Data

SPC-2 "Large File Processing/READ ONLY/1024 KiB Transfer Size" Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 "Large File Processing/READ ONLY/1024 KIB Transfer Size" Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 "Large File Processing/READ ONLY/1024 KiB Transfer Size" graphs
(four pages, 1 graph per page)

SPC-2 "Large File Processing/READ ONLY/256 KiB Transfer Size" Test Run Data

SPC-2 "Large File Processing/READ ONLY/256 KiB Transfer Size" Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 "Large File Processing/READ ONLY/256 KiB Transfer Size" Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 "Large File Processing/READ ONLY/256 KiB Transfer Size" graphs
(four pages, 1 graph per page)

Large Database Query Test

Clause 6.4.4.1

The Large Database Query Test is comprised of a set of I/O operations representative of scans or joins of large relational tables such as those performed for data mining or business intelligence.

Clause 6.4.4.2

The Large Database Query Test has two Test Phases, which shall be executed in the following uninterrupted sequence:

1. *1024 KIB TRANSFER SIZE*
2. *64 KIB TRANSFER SIZE*

The BC shall not be restarted or manually disturbed, altered, or adjusted during the execution of the Large File Processing Test. If power is lost to the BC during this Test all results shall be rendered invalid and the Test re-run in its entirety.

Clause 10.6.9.2

The Full Disclosure Report will contain the following content for the Large Database Query Test:

1. *A listing of the SPC-2 Workload Generator commands and parameters used to execute each of the Test Runs in the Large Database Query Test.*
2. *The human readable SPC-2 Test Results File for each of the Test Runs in the Large Database Query Test.*
3. *A table that contains the following information for each Test Run in the two Test Phases of the Large Database Query Test:*
    - *Average Data Rate: The average Data Rate, in MB per second for the Measurement Interval of each Test Run in the Large Database Query Test.*
    - *Average Data Rate per Stream: The average Data Rate per Stream, in MB per second, for the Measurement Interval of each Test Run in the Large Database Query Test.*
    - *Average Response Time: The average response time, in milliseconds (ms), for the Measurement Interval of each Test Run in the Large Database Query Test.*
4. *Average Data Rate, Average Data Rate per Stream and Average Response time graphs as defined in Clauses 10.1.1, 10.1.2 and 10.1.3.*

SPC-2 Workload Generator Commands and Parameters

The SPC-2 Workload Generator commands and parameters for the Large Database Query Test Runs are documented in Appendix E: SPC-2 Workload Generator Execution Commands and Parameters on Page 83.

SPC-2 Test Results File

A link to the SPC-2 Test Results file generated from the Large Database Query Test Runs is listed below.

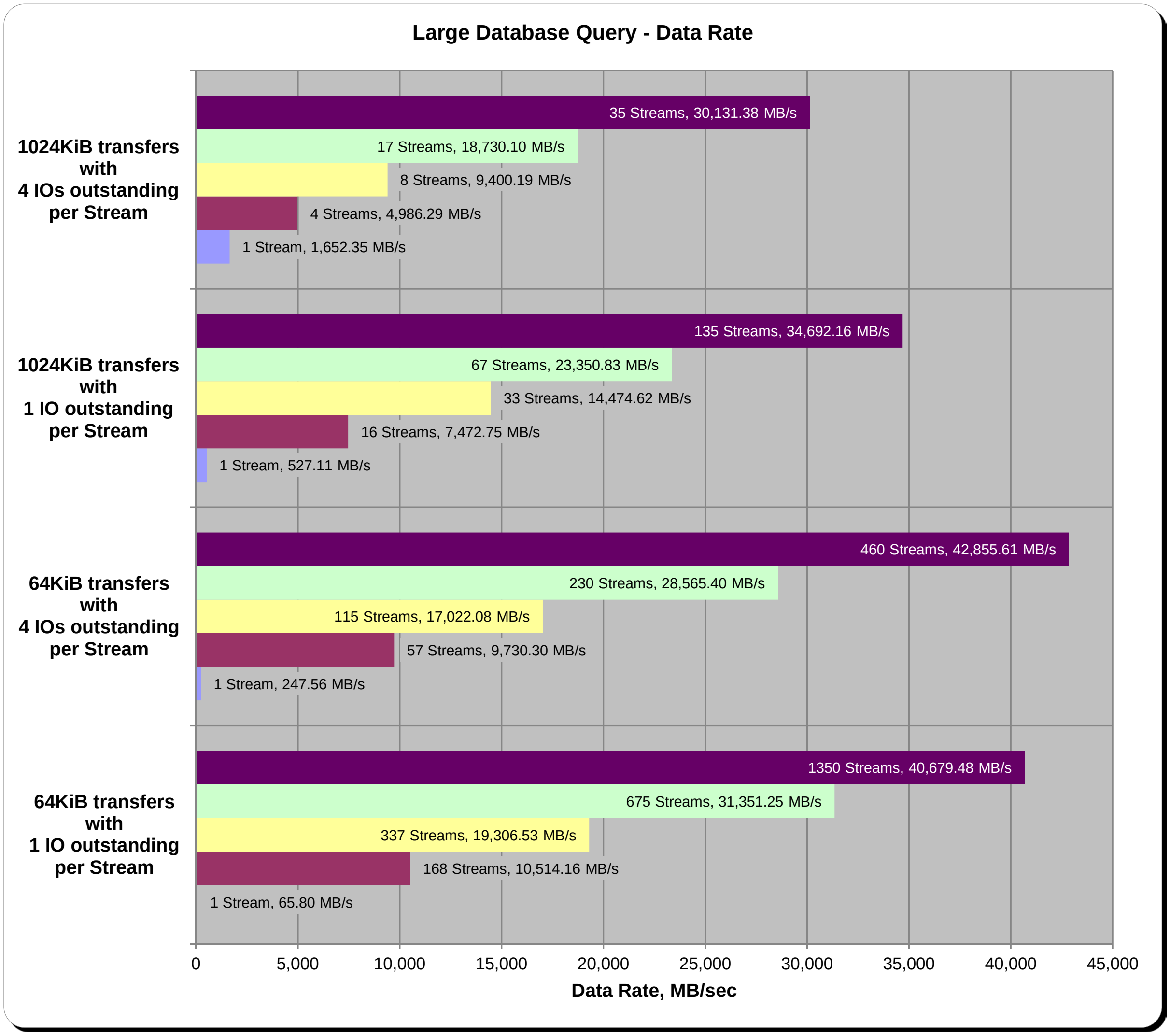
SPC-2 Large Database Query Test Results File

SPC-2 Large Database Query Average Data Rates (MB/s)

The average Data Rate (MB/s) for each Test Run in the two Test Phases of the SPC-2 Large Database Query Test is listed in the table below as well as illustrated in the following graph.

Test Run Sequence	1 Stream	4 Streams	8 Streams	17 Streams	35 Streams
1024KiB w/ 4 IOs/Stream	1,652.35	4,986.29	9,400.19	18,730.10	30,131.38
Test Run Sequence	**1 Stream**	**16 Streams**	**33 Streams**	**67 Streams**	**135 Streams**
1024KiB w/ 1 IO/Stream	527.11	7,472.75	14,474.62	23,350.83	34,692.16
Test Run Sequence	**1 Stream**	**57 Streams**	**115 Streams**	**230 Streams**	**460 Streams**
64KiB w/ 4 IOs/Stream	247.56	9,730.30	17,022.08	28,565.40	42,855.61
Test Run Sequence	**1 Stream**	**168 Streams**	**337 Streams**	**675 Streams**	**1350 Streams**
64KiB w/ 1 IO/Stream	65.80	10,514.16	19,306.53	31,351.25	40,679.48

SPC-2 Large Database Query Average Data Rates Graph

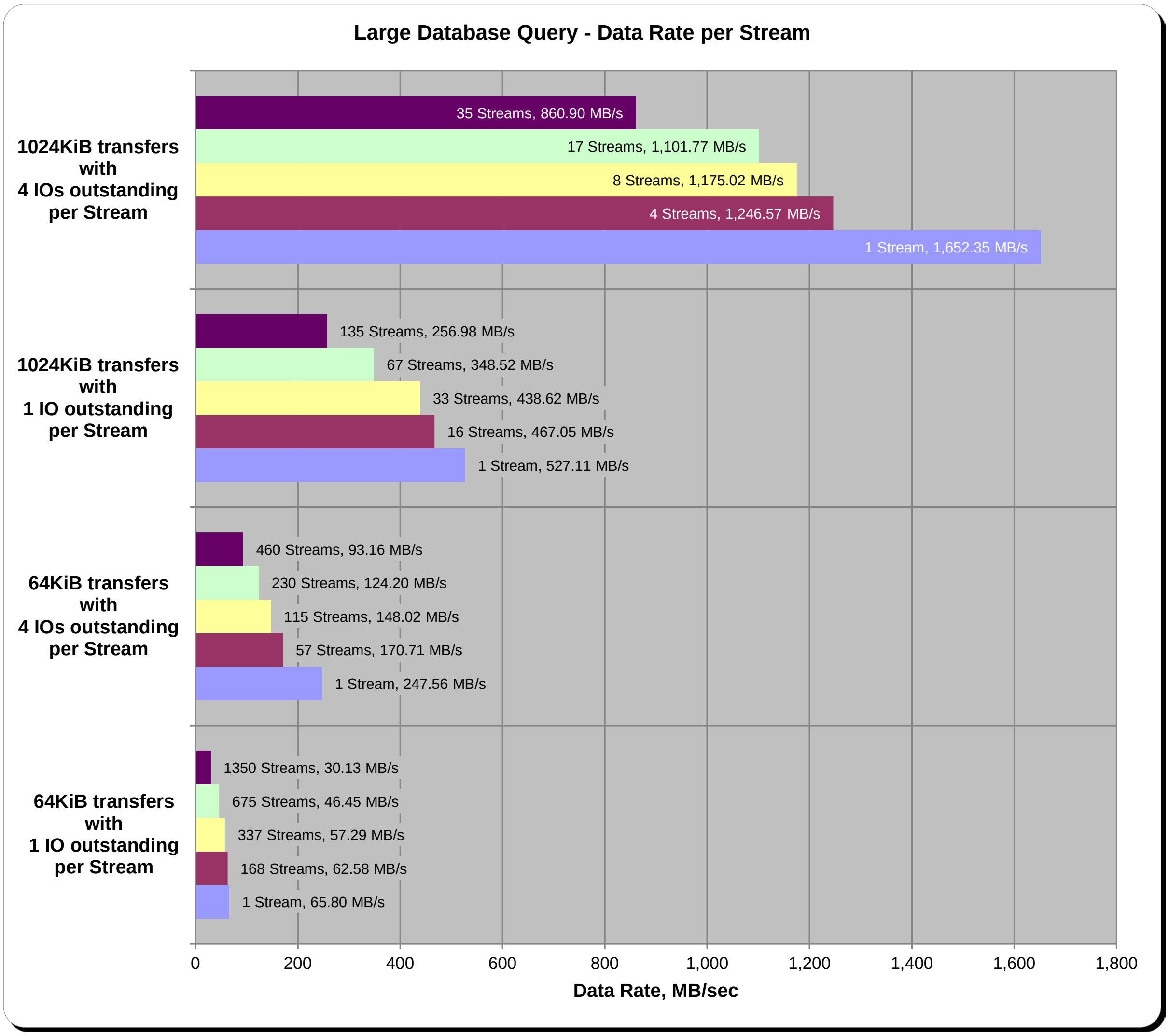


SPC-2 Large Database Query Average Data Rate per Stream

The average Data Rate per Stream for each Test Run in the two Test Phases of the SPC-2 Large Database Query Test is listed in the table below as well as illustrated in the following graph.

Test Run Sequence	**1 Stream**	**4 Streams**	**8 Streams**	**17 Streams**	**35 Streams**
1024KiB w/ 4 IOs/Stream	1,652.35	1,246.57	1,175.02	1,101.77	860.90
Test Run Sequence	**1 Stream**	**16 Streams**	**33 Streams**	**67 Streams**	**135 Streams**
1024KiB w/ 1 IO/Stream	527.11	467.05	438.62	348.52	256.98
Test Run Sequence	**1 Stream**	**57 Streams**	**115 Streams**	**230 Streams**	**460 Streams**
64KiB w/ 4 IOs/Stream	247.56	170.71	148.02	124.20	93.16
Test Run Sequence	**1 Stream**	**168 Streams**	**337 Streams**	**675 Streams**	**1350 Streams**
64KiB w/ 1 IO/Stream	65.80	62.58	57.29	46.45	30.13

SPC-2 Large Database Query Average Data Rate per Stream Graph

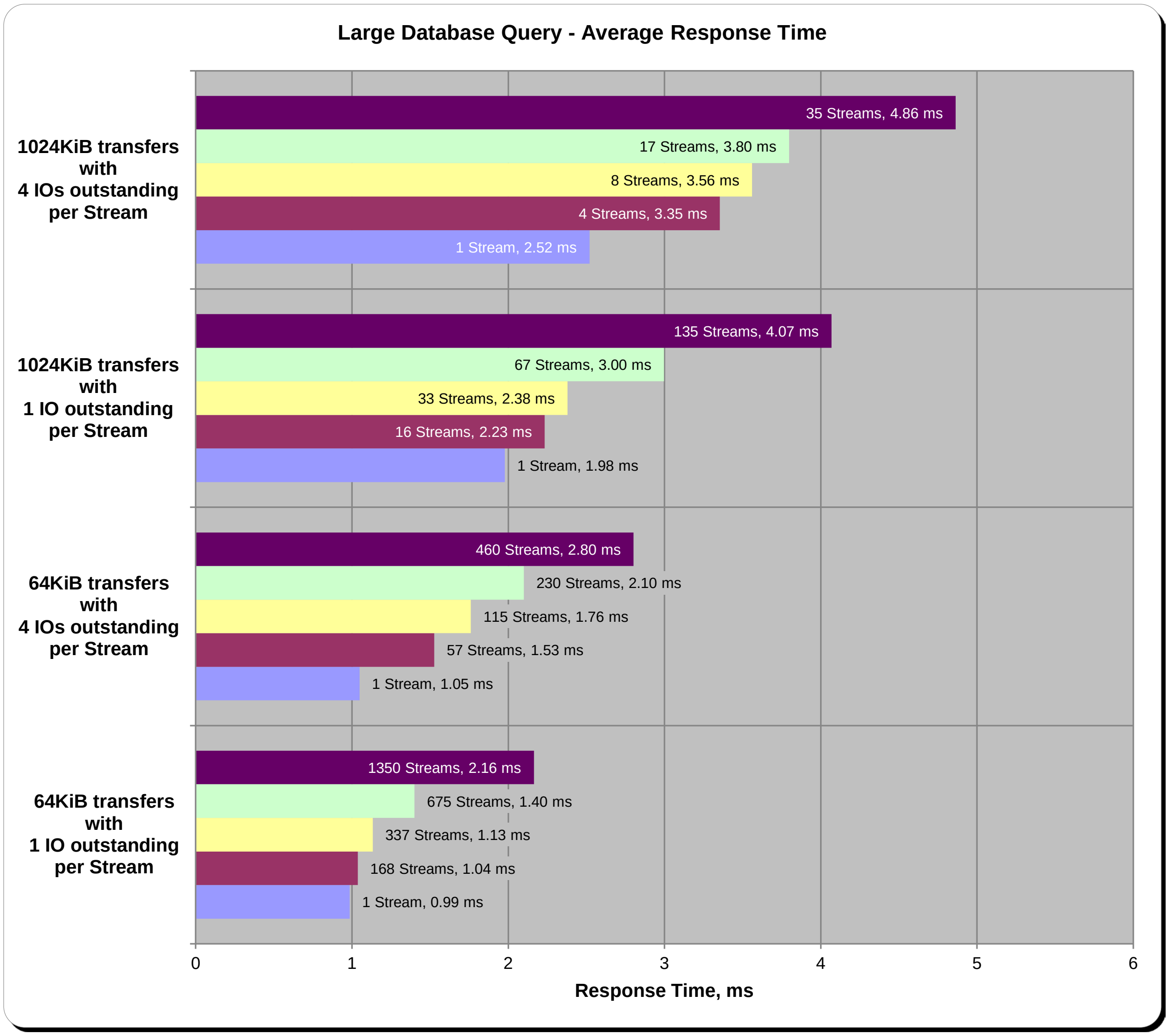


SPC-2 Large Database Query Average Response Time

The average Response Time, in milliseconds, for each Test Run in the two Test Phases of the SPC-2 Large Database Query Test is listed in the table below as well as illustrated in the following graph.

Test Run Sequence	1 Stream	4 Streams	8 Streams	17 Streams	35 Streams
1024KiB w/ 4 IOs/Stream	2.52	3.35	3.56	3.80	4.86
Test Run Sequence	**1 Stream**	**16 Streams**	**33 Streams**	**67 Streams**	**135 Streams**
1024KiB w/ 1 IO/Stream	1.98	2.23	2.38	3.00	4.07
Test Run Sequence	**1 Stream**	**57 Streams**	**115 Streams**	**230 Streams**	**460 Streams**
64KiB w/ 4 IOs/Stream	1.05	1.53	1.76	2.10	2.80
Test Run Sequence	**1 Stream**	**168 Streams**	**337 Streams**	**675 Streams**	**1350 Streams**
64KiB w/ 1 IO/Stream	0.99	1.04	1.13	1.40	2.16

SPC-2 Large Database Query Average Response Time Graph



Large Database Query Test – 1024 KIB TRANSFER SIZE Test Phase

Clause 10.6.9.2.1

1. *A table that will contain the following information for each "1024 KiB Transfer Size, 4 Outstanding I/Os" Test Run:*
   - *The number of Streams specified.*
   - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
2. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the "1024 KiB Transfer Size, 4 Outstanding I/Os" Test Runs as specified in Clauses 10.1.4 – 10.1.6.*
3. *A table that will contain the following information for each "1024 KiB Transfer Size, 1 Outstanding I/O" Test Run:*
   - *The number of Streams specified.*
   - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
4. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the "1024 KiB Transfer Size, 1 Outstanding I/O" Test Runs as specified in Clauses 10.1.4 – 10.1.6.*

A hyperlink for each of the above tables and graphs may appear in the FDR to provide access to the table or graph.

A hyperlink to a table with the SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/4 Outstanding I/Os" Test Run data appears on the next page. That entry is followed by hyperlinks to graphs illustrating the average Data Rate, average Data Rate per Stream, and average Response Time produced by the same Test Runs. The table and graphs present the data at five-second intervals.

Immediately following the above SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/4 Outstanding I/Os" entries will be hyperlinks for SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/1 Outstanding I/O" table and graphs. The table contains the Test Run data and the graphs illustrate the average Data Rate, average Data Rate per Stream, and average Response Time produced by the Test Runs.

SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/4 Outstanding I/Os" Test Run Data

SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/4 Outstanding I/Os" Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/4 Outstanding I/Os" Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/4 Outstanding I/Os" graphs
(four pages, 1 graph per page)

SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/1 Outstanding I/O" Test Run Data

SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/1 Outstanding I/O" Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/1 Outstanding I/O" Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 "Large Database Query/1024 KIB TRANSFER SIZE/1 Outstanding I/O" graphs
(four pages, 1 graph per page)

Large Database Query Test – 64 KIB TRANSFER SIZE Test Phase

Clause 10.6.9.2.2

1. *A table that will contain the following information for each "64 KiB Transfer Size, 4 Outstanding I/Os" Test Run:*
   - *The number of Streams specified.*
   - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
2. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the "64 KiB Transfer Size, 4 Outstanding I/Os" Test Runs as specified in Clauses 10.1.4 – 10.1.6.*
3. *A table that will contain the following information for each "64 KiB Transfer Size, 1 Outstanding I/O" Test Run:*
   - *The number of Streams specified.*
   - *The Average Data Rate, Average Data Rate per Stream, and Average Response Time reported at five second intervals.*
4. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the "64 KiB Transfer Size, 1 Outstanding I/O" Test Runs as specified in Clauses 10.1.4 – 10.1.6.*

A hyperlink for each of the above tables and graphs may appear in the FDR to provide access to the table or graph.

A hyperlink to a table with the SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/4 Outstanding I/Os" Test Run data appears on the next page. That entry is followed by hyperlinks to graphs illustrating the average Data Rate, average Data Rate per Stream, and average Response Time produced by the same Test Runs. The table and graphs present the data at five-second intervals.

Immediately following the above SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/4 Outstanding I/Os" entries will be hyperlinks for SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/1 Outstanding I/O" table and graphs. The table contains the Test Run data and the graphs illustrate the average Data Rate, average Data Rate per Stream, and average Response Time produced by the Test Runs.

SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/4 Outstanding I/Os" Test Run Data

SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/4 Outstanding I/Os" Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/4 Outstanding I/Os" Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/4 Outstanding I/Os" graphs
(four pages, 1 graph per page)

SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/1 Outstanding I/O" Test Run Data

SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/1 Outstanding I/O" Test Run Data Tables: Ramp-Up, Measurement Interval, Run-Out, and Ramp-Down Periods *(3 pages)*

SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/1 Outstanding I/O" Graphs

Average Data Rate – Complete Test Run

Average Data Rate – Measurement Interval (MI) Only

Average Data Rate per Stream

Average Response Time

SPC-2 "Large Database Query/64 KIB TRANSFER SIZE/1 Outstanding I/O" graphs
(four pages, 1 graph per page)

Video on Demand Delivery Test

Clause 6.4.5.1

The Video on Demand Delivery Test represents the I/O operations required to enable individualized video entertainment for a community of subscribers, which draw from a digital film library.

Clause 6.4.5.2

The Video on Demand Delivery Test consists of one (1) Test Run.

The BC shall not be restarted or manually disturbed, altered, or adjusted during the execution of the Video on Demand Delivery Test. If power is lost to the BC during this Test all results shall be rendered invalid and the Test re-run in its entirety.

Clause 10.6.9.3

The Full Disclosure Report will contain the following content for the Video on Demand Delivery Test:

1. *A listing of the SPC-2 Workload Generator commands and parameters used to execute the Test Run in the Video on Demand Delivery Test.*
2. *The human readable SPC-2 Test Results File for the Test Run in the Video on Demand Delivery Test.*
3. *A table that contains the following information for the Test Run in the Video on Demand Delivery Test:*
   - *The number Streams specified.*
   - *The Ramp-Up duration in seconds.*
   - *The Measurement Interval duration in seconds.*
   - *The average data rate, in MB per second, for the Measurement Interval.*
   - *The average data rate, in MB per second, per Stream for the Measurement Interval.*
4. *A table that contains the following information for the single Video on Demand Delivery Test Run:*
   - *The number Streams specified.*
   - *The average data rate, average data rate per stream, average Response Time, and Maximum Response Time reported at 60 second intervals.*
5. *Average Data Rate by Intervals, Average Data Rate per Stream by Intervals, and Average Response Time by Intervals graphs for the single Video on Demand Delivery Test Run as specified in Clause 10.1.8.*
6. *A Maximum Response Time (intervals) graph as specified in Clause 10.1.8.*

SPC-2 Workload Generator Commands and Parameters

The SPC-2 Workload Generator commands and parameters for the Video on Demand Delivery Test Run are documented in Appendix E: SPC-2 Workload Generator Execution Commands and Parameters on Page 83..

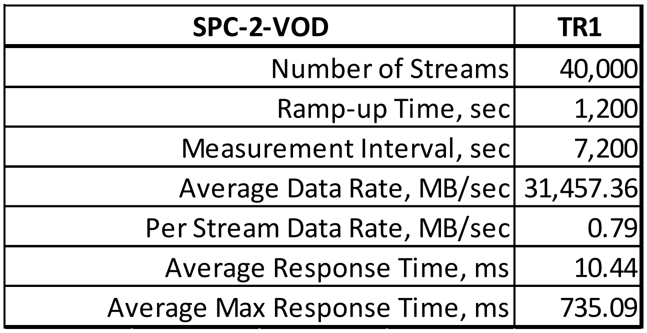
SPC-2 Test Results File

A link to the SPC-2 Test Results file generated from the Video on Demand Delivery Test Run is listed below.

SPC-2 Video on Demand Delivery Test Results File

SPC-2 Video on Demand Delivery Test Run Data

The number of Streams specified, Ramp-Up duration in seconds, Measurement Interval duration in seconds, average Data Rate for the Measurement Interval, and average Data Rate per Stream for the Measurement Interval are listed in the following table.



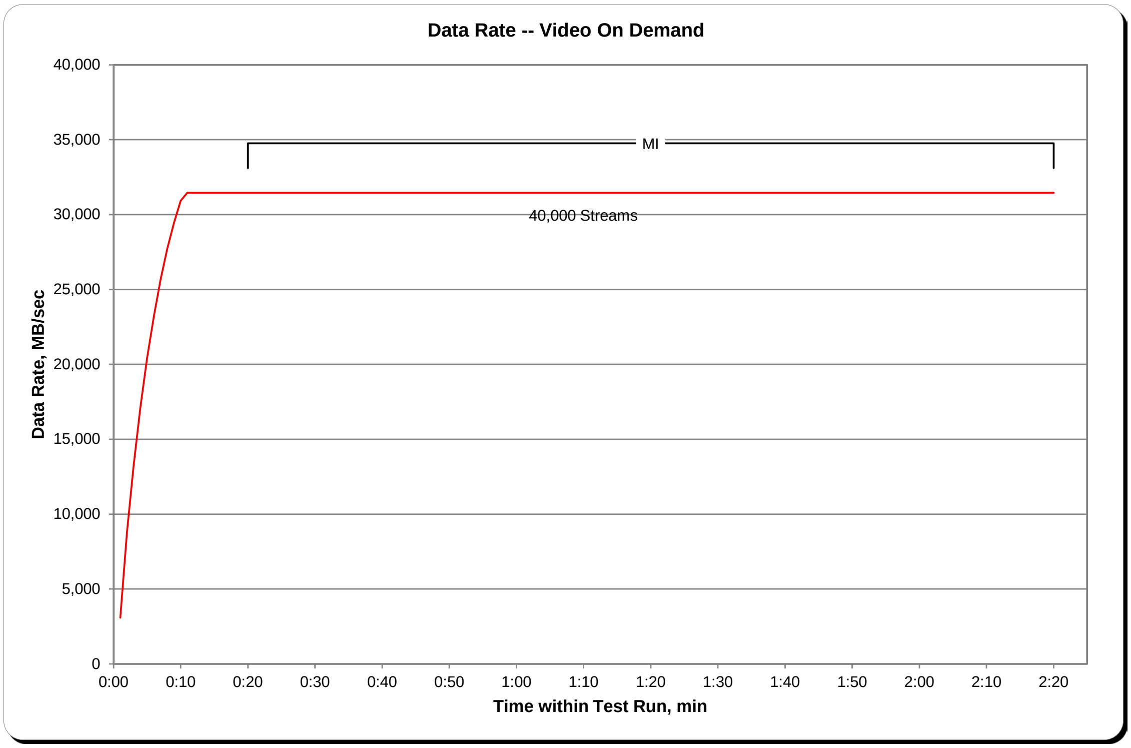
SPC-2-VOD	TR1
Number of Streams	40,000
Ramp-up Time, sec	1,200
Measurement Interval, sec	7,200
Average Data Rate, MB/sec	31,457.36
Per Stream Data Rate, MB/sec	0.79
Average Response Time, ms	10.44
Average Max Response Time, ms	735.09

Video on Demand Delivery Test – TEST RUN DATA BY INTERVAL

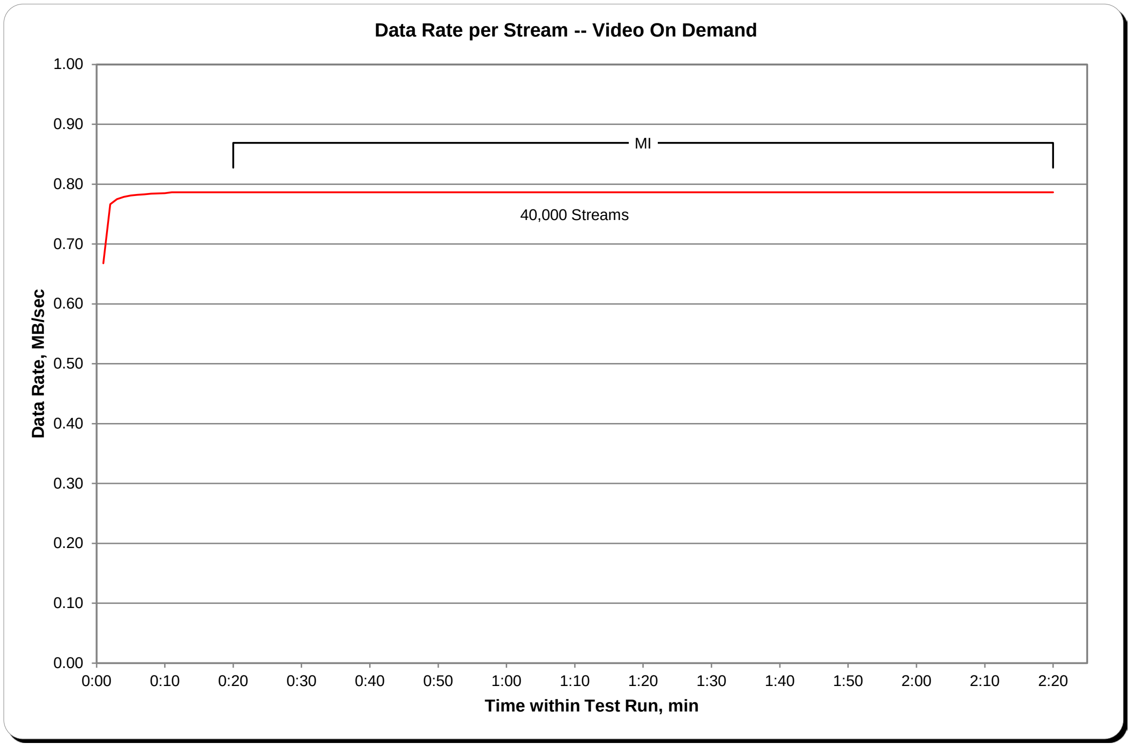
The SPC-2 Video on Demand Delivery Test Run data is contained in the table that appears below. That table is followed by graphs illustrating the average Data Rate and average Data Rate per Stream produced by the same Test Runs. The table and graphs present the data at sixty second intervals.

TR1	40,000 Streams				TR1	40,000 Streams				TR1	40,000 Streams			
Test Run Sequence Time	Data Rate, MB/sec	Data Rate / Stream, MB/sec	Response Time, ms	Maximum Response Time, ms	Test Run Sequence Time	Data Rate, MB/sec	Data Rate / Stream, MB/sec	Response Time, ms	Maximum Response Time, ms	Test Run Sequence Time	Data Rate, MB/sec	Data Rate / Stream, MB/sec	Response Time, ms	Maximum Response Time, ms
0:01:00	3,091.48	0.67	1.99	51.60	0:48:00	31,457.76	0.79	10.39	746.36	1:35:00	31,457.55	0.79	10.46	400.59
0:02:00	8,788.88	0.77	2.34	57.70	0:49:00	31,457.52	0.79	10.46	666.55	1:36:00	31,457.00	0.79	10.49	945.12
0:03:00	13,259.96	0.77	3.05	275.46	0:50:00	31,456.95	0.79	10.37	827.83	1:37:00	31,457.35	0.79	10.55	936.83
0:04:00	17,101.94	0.78	4.09	597.52	0:51:00	31,457.89	0.79	10.27	608.56	1:38:00	31,457.53	0.79	10.44	735.19
0:05:00	20,393.10	0.78	5.20	614.00	0:52:00	31,457.17	0.79	10.37	721.68	1:39:00	31,458.07	0.79	10.54	716.34
0:06:00	23,225.29	0.78	6.36	507.07	0:53:00	31,457.06	0.79	10.35	646.27	1:40:00	31,456.30	0.79	10.54	343.11
0:07:00	25,667.21	0.78	7.59	627.07	0:54:00	31,457.35	0.79	10.37	754.42	1:41:00	31,456.95	0.79	10.56	828.74
0:08:00	27,719.03	0.78	8.63	515.57	0:55:00	31,458.41	0.79	10.29	949.55	1:42:00	31,456.94	0.79	10.40	512.43
0:09:00	29,438.59	0.78	9.46	762.62	0:56:00	31,457.45	0.79	10.37	495.26	1:43:00	31,458.24	0.79	10.35	515.98
0:10:00	30,920.47	0.78	10.35	836.35	0:57:00	31,457.41	0.79	10.33	530.76	1:44:00	31,456.94	0.79	10.29	848.69
0:11:00	31,457.89	0.79	10.31	827.39	0:58:00	31,456.43	0.79	10.28	1,165.20	1:45:00	31,457.65	0.79	10.32	1,067.06
0:12:00	31,456.05	0.79	10.25	625.62	0:59:00	31,456.51	0.79	10.31	623.33	1:46:00	31,457.36	0.79	10.32	983.87
0:13:00	31,457.67	0.79	10.21	725.37	1:00:00	31,458.36	0.79	10.34	833.69	1:47:00	31,457.38	0.79	10.29	716.79
0:14:00	31,455.73	0.79	10.25	736.60	1:01:00	31,456.14	0.79	10.39	1,062.75	1:48:00	31,456.65	0.79	10.42	621.06
0:15:00	31,458.54	0.79	10.36	860.54	1:02:00	31,457.82	0.79	10.46	563.13	1:49:00	31,456.57	0.79	10.33	828.43
0:16:00	31,457.78	0.79	10.34	398.66	1:03:00	31,457.85	0.79	10.38	845.39	1:50:00	31,457.59	0.79	10.32	886.85
0:17:00	31,457.71	0.79	10.30	728.39	1:04:00	31,456.97	0.79	10.38	754.35	1:51:00	31,458.72	0.79	10.41	540.88
0:18:00	31,457.88	0.79	10.38	632.72	1:05:00	31,456.76	0.79	10.45	1,056.01	1:52:00	31,457.32	0.79	10.32	632.22
0:19:00	31,457.02	0.79	10.24	841.89	1:06:00	31,457.74	0.79	10.44	513.47	1:53:00	31,456.71	0.79	10.26	731.89
0:20:00	31,455.58	0.79	10.41	548.82	1:07:00	31,455.94	0.79	10.46	603.33	1:54:00	31,459.69	0.79	10.26	938.08
0:21:00	31,455.85	0.79	10.34	860.49	1:08:00	31,456.19	0.79	10.52	750.58	1:55:00	31,456.10	0.79	10.33	739.57
0:22:00	31,457.60	0.79	10.38	523.26	1:09:00	31,457.64	0.79	10.46	726.85	1:56:00	31,456.18	0.79	10.32	748.37
0:23:00	31,457.16	0.79	10.29	621.57	1:10:00	31,457.16	0.79	10.49	836.36	1:57:00	31,459.22	0.79	10.30	672.84
0:24:00	31,458.44	0.79	10.36	834.51	1:11:00	31,458.96	0.79	10.53	851.44	1:58:00	31,457.80	0.79	10.37	734.19
0:25:00	31,457.87	0.79	10.39	953.23	1:12:00	31,456.05	0.79	10.49	756.48	1:59:00	31,458.22	0.79	10.25	650.19
0:26:00	31,458.06	0.79	10.53	767.43	1:13:00	31,459.78	0.79	10.45	727.89	2:00:00	31,457.67	0.79	10.34	739.94
0:27:00	31,458.40	0.79	10.53	781.93	1:14:00	31,457.26	0.79	10.61	666.43	2:01:00	31,458.35	0.79	10.40	836.34
0:28:00	31,457.22	0.79	10.57	729.19	1:15:00	31,456.11	0.79	10.49	630.53	2:02:00	31,458.34	0.79	10.46	531.23
0:29:00	31,457.02	0.79	10.45	838.43	1:16:00	31,456.49	0.79	10.53	653.05	2:03:00	31,456.15	0.79	10.47	838.51
0:30:00	31,458.16	0.79	10.52	850.62	1:17:00	31,458.61	0.79	10.52	533.27	2:04:00	31,458.66	0.79	10.46	630.73
0:31:00	31,457.06	0.79	10.54	951.32	1:18:00	31,455.45	0.79	10.55	343.68	2:05:00	31,457.46	0.79	10.40	698.64
0:32:00	31,458.71	0.79	10.47	1,081.54	1:19:00	31,456.54	0.79	10.51	532.17	2:06:00	31,456.75	0.79	10.39	878.61
0:33:00	31,456.65	0.79	10.51	530.27	1:20:00	31,456.87	0.79	10.48	742.39	2:07:00	31,458.28	0.79	10.42	1,049.91
0:34:00	31,457.36	0.79	10.46	946.44	1:21:00	31,459.00	0.79	10.55	634.36	2:08:00	31,454.96	0.79	10.48	513.22
0:35:00	31,457.68	0.79	10.46	1,172.32	1:22:00	31,457.14	0.79	10.54	562.86	2:09:00	31,457.55	0.79	10.45	732.92
0:36:00	31,457.12	0.79	10.49	740.60	1:23:00	31,458.31	0.79	10.44	346.74	2:10:00	31,456.96	0.79	10.41	733.86
0:37:00	31,457.24	0.79	10.51	846.06	1:24:00	31,458.19	0.79	10.48	531.64	2:11:00	31,456.14	0.79	10.47	726.38
0:38:00	31,457.04	0.79	10.52	534.53	1:25:00	31,457.49	0.79	10.53	443.85	2:12:00	31,458.73	0.79	10.51	844.06
0:39:00	31,456.89	0.79	10.52	973.66	1:26:00	31,457.52	0.79	10.56	944.71	2:13:00	31,456.74	0.79	10.50	645.66
0:40:00	31,458.07	0.79	10.64	960.32	1:27:00	31,458.79	0.79	10.51	736.34	2:14:00	31,458.23	0.79	10.46	519.20
0:41:00	31,456.01	0.79	10.42	952.05	1:28:00	31,457.66	0.79	10.59	513.07	2:15:00	31,456.43	0.79	10.56	843.98
0:42:00	31,456.62	0.79	10.60	1,163.95	1:29:00	31,457.84	0.79	10.52	849.03	2:16:00	31,457.55	0.79	10.44	723.52
0:43:00	31,458.33	0.79	10.46	623.49	1:30:00	31,456.98	0.79	10.54	945.36	2:17:00	31,457.37	0.79	10.48	560.41
0:44:00	31,455.64	0.79	10.53	758.91	1:31:00	31,456.66	0.79	10.55	835.91	2:18:00	31,457.71	0.79	10.45	857.90
0:45:00	31,459.47	0.79	10.52	833.48	1:32:00	31,458.24	0.79	10.42	846.43	2:19:00	31,456.81	0.79	10.41	633.93
0:46:00	31,457.09	0.79	10.37	951.92	1:33:00	31,456.81	0.79	10.53	328.99	2:20:00	31,457.79	0.79	10.49	529.53
0:47:00	31,457.46	0.79	10.40	744.87	1:34:00	31,455.75	0.79	10.44	628.11	0:00:00	0.00	0.00	0.00	0.00

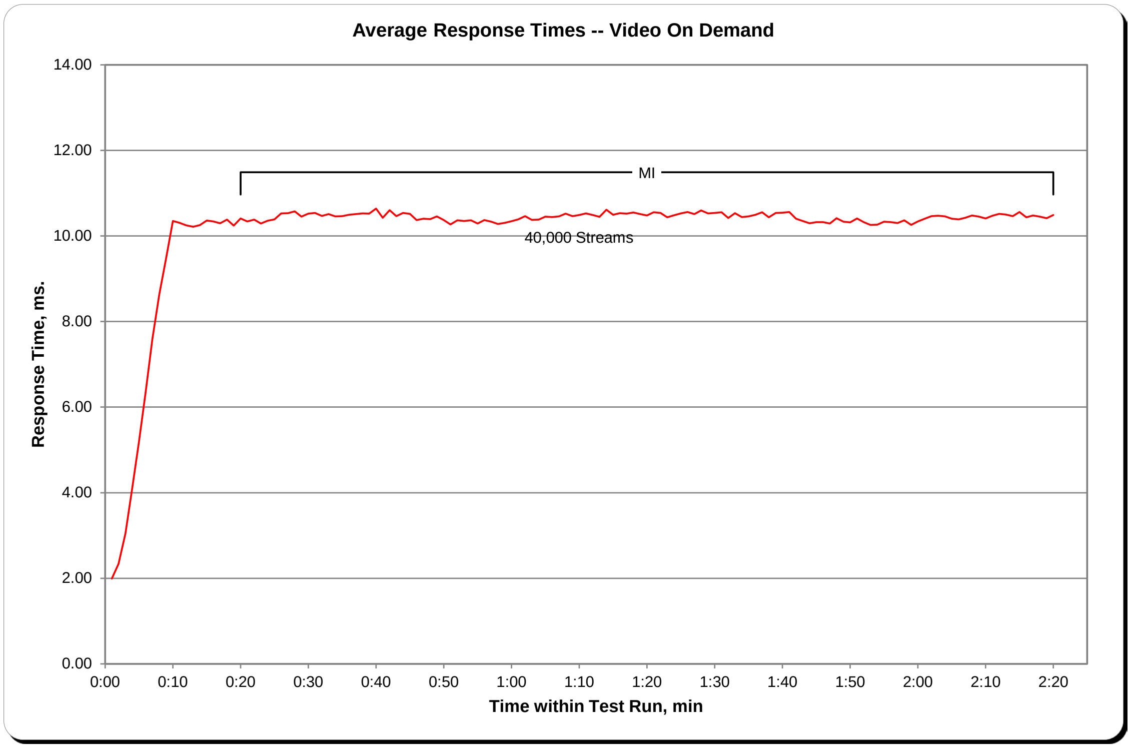
SPC-2 Video on Demand Delivery Average Data Rate Graph



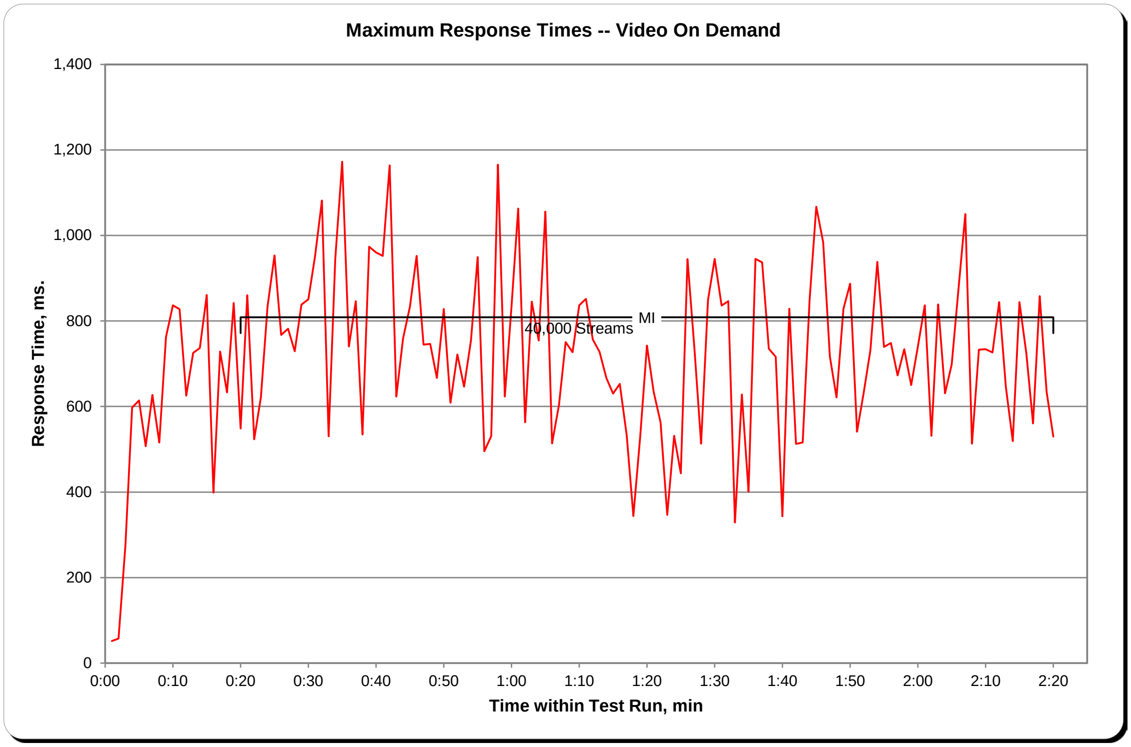
SPC-2 Video on Demand Delivery Average Data Rate per Stream Graph



SPC-2 Video on Demand Delivery Average Response Time Graph



SPC-2 Video on Demand Delivery Maximum Response Time Graph



Data Persistence Test

Clause 7

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-2 Workload Generator will write a specific pattern at randomly selected locations throughout the Total ASU Capacity (Persistence Test Run 1). The SPC-2 Workload Generator will retain the information necessary to later validate the pattern written at each location.

The Tested Storage Configuration 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.

Restart the TSC, and if the Host System(s) were shutdown and powered off, restart the Host System(s).

The SPC-2 Workload Generator will utilize the retained data from Persistence Test Run 1 to verify (Persistence Run 2) the bit patterns written in Persistence Test Run 1 and their corresponding location.

Clause 10.6.9.4

The Full Disclosure Report will contain the following content for the Data Persistence Test:

1. *A listing of the SPC-2 Workload Generator commands and parameters used to execute each of the Test Runs in the Persistence Test.*
2. *The human readable SPC-2 Test Results File for each of the Test Runs in the Data Persistence Test.*
3. *A table from the successful Persistence Test, which contains the results from the test.*

SPC-2 Workload Generator Commands and Parameters

The SPC-2 Workload Generator commands and parameters for the Persistence Test Runs are documented in Appendix E: SPC-2 Workload Generator Execution Commands and Parameters on Page 83.

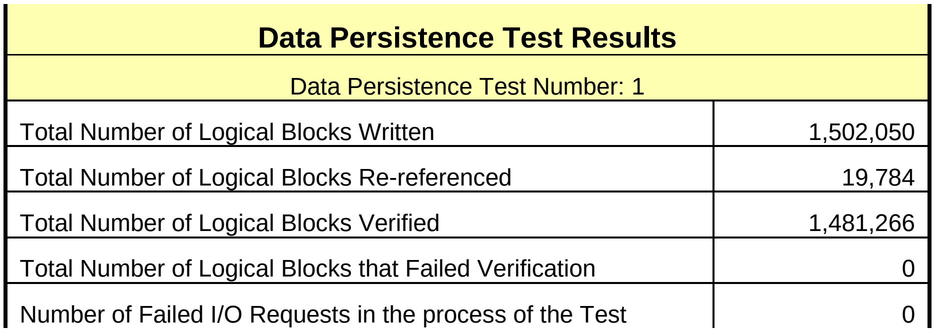
Data Persistence Test Results File

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

Persistence 1 Test Run Results File

Persistence 2 Test Run Results File

Data Persistence Test Results



Data Persistence Test Results	
Data Persistence Test Number: 1	
Total Number of Logical Blocks Written	1,502,050
Total Number of Logical Blocks Re-referenced	19,784
Total Number of Logical Blocks Verified	1,481,266
Total Number of Logical Blocks that Failed Verification	0
Number of Failed I/O Requests in the process of the Test	0

PRICED STORAGE CONFIGURATION AVAILABILITY DATE

Clause 10.6.9

The committed delivery date 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 must be the date at which all components are committed to be available. All availability dates, whether for individual components or for the Priced Storage Configuration as a whole, must be disclosed to a precision of one day.

The Availability Data shall be stated in either a combination of specific alphanumeric month, numeric day and numeric year or as "Currently Available".

The Kaminario K2, as documented in this SPC-2 Full Disclosure Report, is currently available for customer purchase and shipment.

ANOMALIES OR IRREGULARITIES

Clause 10.6.12

The FDR shall include a clear and complete description of any anomalies or irregularities encountered in the course of executing the SPC-2 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-2 Onsite Audit of the Kaminario K2.

APPENDIX A: SPC-2 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-2 Data Repository Definitions

Total ASU Capacity: The total storage capacity read and written in the course of executing the SPC-2 benchmark.

Application Storage Unit (ASU): The logical interface between the storage and SPC-2 Workload Generator. The ASU is 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-2 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-2 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 ASU.

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 sum of unused storage capacity within the Physical Storage Capacity, Configured Storage Capacity, and Addressable Storage Capacity.

SPC-2 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-2 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-2 Data Repository.

SPC-2 Test Execution Definitions

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

Completion Time: The time recorded by the Workload Generator when an I/O Request is completed by the Tested Storage Configuration (TSC) as signaled by System Software.

Data Rate: The data volume, in MB, transferred by all Measured I/O Requests in an SPC-2 Test Run divided by the length of the Test Run in seconds.

Failed I/O Request: Any I/O Request issued by the SPC-2 Workload Generator that meets one of the following conditions *(see "I/O Completion Types" illustrated below)*:

- The I/O Request was signaled as failed by System Software.
- The I/O Request started within the Measurement Interval, but did not complete prior to the end of the appropriate Run-Out period..
- The I/O Request started within the Run-Out period, but did not complete prior to the end of the appropriate Ramp-Down period.

I/O Request Throughput: The total number of Measured I/O Requests in an SPC-2 Test Run divided by the duration of the Measurement Interval in seconds.

Measured I/O Request: A Completed I/O Request that begins (Start Time) within a Measurement Interval and completes (Completion Time) prior to the end of the appropriate Ramp Down *(see "I/O Completion Types" illustrated below).*

Measurement Interval: A specified, contiguous period of time, after the TSC has reached Steady State, when data is collected by the Workload Generator to produce the test results for a SPC-2 Test Run *(see "SPC-2 Test Run Components" illustrated below, Test Run 1: T_2-T_3 and Test Run 2: T_7-T_8).*

Outstanding I/O Requests: The Outstanding I/O Requests parameter specifies the maximum number of concurrent I/O Requests, associated with a give Stream, which have been issued but not yet completed. *(Clause 3.4.4 of the SPC-2 Benchmark Specification).*

Ramp-Down: A specified, contiguous period of time in which the TSC is required to complete I/O Requests started but not completed during the preceding Run-Out period. Ramp-Down begins at the end of the preceding Run-Out period *(see "SPC-2 Test Run Components" illustrated below, Test Run 1: T_4-T_5 and Test Run 2: T_9-T_{10}).* The Workload Generator will not submit any I/O Requests during the Ramp-Down.

Ramp-Up: A specified, contiguous period of 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. The Ramp-Up period ends at the beginning of the Measurement Interval *(see "SPC-2 Test Run Components" illustrated below, Test Run 1: T_0-T_2 and Test Run 2: T_5-T_7).*

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

Run-Out: A specified, contiguous period of time in which the TSC is required to complete I/O Requests started but not completed during the preceding Measurement Interval. The Run-Out period begins at the end of the preceding Measurement Interval and is a component of the Steady State period *(see "SPC-2 Test Run Components" illustrated below, Test Run 1: T_3-T_4 and Test Run 2: T_9-T_{10}).* The Workload Generator will continue to submit I/O Requests at the Test Run's specified rate during the Run-Out period.

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 TSC.

Steady State: The period during which the workload presented to the TSC by the SPC-2 Workload Generator is constant and the resulting TSC I/O Request Throughput is both consistent and sustainable. The Steady State period includes both the Measurement Interval and Run-Out periods *(see "SPC-2 Test Run Components" illustrated below, Test Run 1: T_1-T_4 and Test Run 2: T_6-T_9).*

Steady State is achieved only after caches in the TSC have filled and as a result the I/O Request Throughput of the TSC has stabilized.

Stream: A collection of Stream Segments that started within a Test Run.

Stream Segment: A sequentially organized pattern of I/O requests, which transfers a contiguous range of data.

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

Test Phase: A collection of one or more SPC-2 Test Runs sharing a common objective and intended to be run in a specific sequence.

Test Run: The execution of SPC-2 that produces specific SPC-2 test results. SPC-2 Test Runs have specified, measured Ramp-Up, Measurement Interval, Run-Out and Ramp-Down periods. "SPC-2 Test Run Components" *(see below)* illustrates the Ramp-Up, Steady State, Measurement Interval, Run-Out, and Ramp-Down components contained in two uninterrupted SPC-2 Test Runs *(Test Run 1: T_0-T_5 and Test Run 2: T_5-T_{10}).*

Test Run Sequence: A related sequence of Large File Processing (LFP) or Large Database Query (LDQ) Test Runs. Each Test Run Sequence will consist of five Test Runs, which vary the number of Streams as follows:

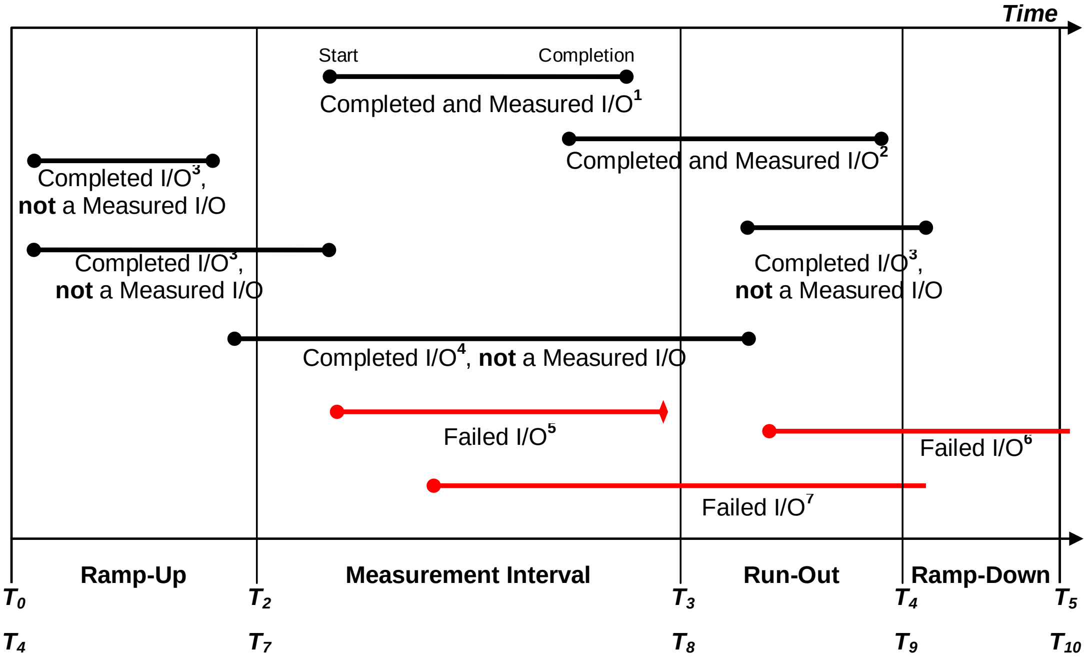
- Test Run 1: Maximum number of Streams, which is selected by the Test Sponsor
- Test Run 2: 50% of the maximum number of Streams used in Test Run 1.
- Test Run 3: 25% of the maximum number of Streams used in Test Run 1.
- Test Run 4: 12.5% of the maximum number of Streams used in Test Run 1.
- Test Run 5: 1 Stream.

Each of the five Test Runs in a Test Run Sequence will share the same attributes with the exception of the number of Streams. For example:

- Large File Processing, Read, 1024 KiB Transfer Size: Maximum Streams
- Large File Processing, Read, 1024 KiB Transfer Size: 50% of Maximum Streams
- Large File Processing, Read, 1024 KiB Transfer Size: 25% of Maximum Streams
- Large File Processing, Read, 1024 KiB Transfer Size: 12.5% of Maximum Streams
- Large File Processing, Read, 1024 KiB Transfer Size: 1 Stream

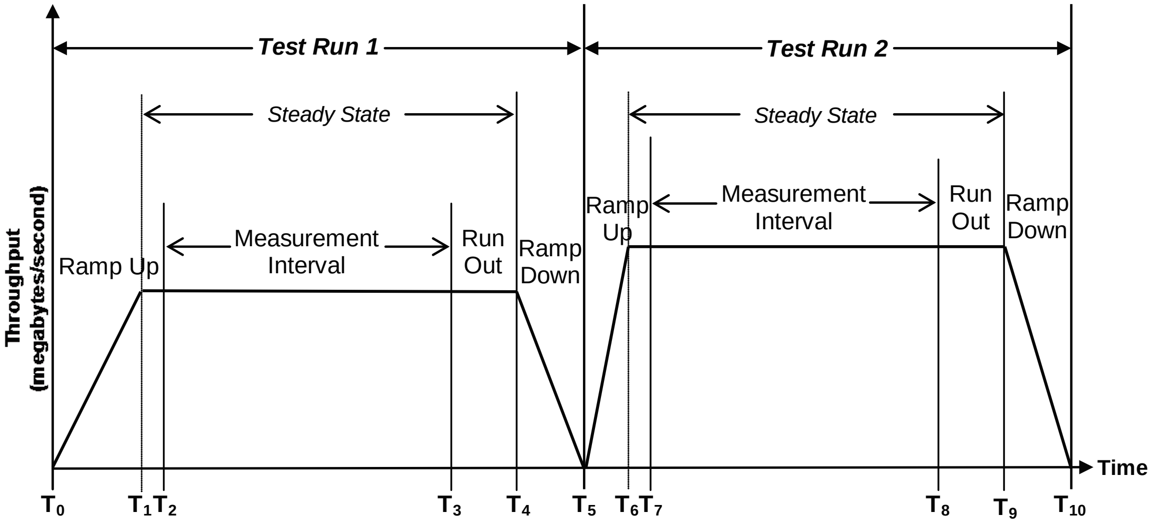
Transfer Size: The Transfer Size parameter specifies the number of bytes in KiB to transfer. *(Clause 3.4.7 of the SPC-2 Benchmark Specification)*

I/O Completion Types


Time
Start
Completion
Completed and Measured I/O1
Completed and Measured I/O2
Completed I/O3, not a Measured I/O
Completed I/O3, not a Measured I/O
Completed I/O3, not a Measured I/O
Completed I/O4, not a Measured I/O
Failed I/O5
Failed I/O6
Failed I/O7
Ramp-Up
Measurement Interval
Run-Out
Ramp-Down
T0 T2 T3 T4 T5
T4 T7 T8 T9 T10

***Completed and Measured I/O*[1]**: *I/O started and completed within the Measurement Interval.*
***Completed and Measured I/O*[2]**: *I/O started within the Measurement Interval and completed within Ramp Down.*
***Completed I/O*[3]**: *I/O started before or after the Measurement Interval – not measured.*
***Completed I/O*[4]**: *I/O started before and completed after the Measurement Interval – not measured.*
***Failed I/O*[5]**: *Signaled as failed by System Software.*
***Failed I/O*[6]**: *I/O did not complete prior to the end of Ramp-Down.*
***Failed I/O*[7]**: *I/O did not complete prior to the end of Run-Out.*

SPC-2 Test Run Components



APPENDIX B: CUSTOMER TUNABLE PARAMETERS AND OPTIONS

There were no customer tunable parameters or options changed from their default values for the SPC-1 measurements.

APPENDIX C: TESTED STORAGE CONFIGURATION (TSC) CREATION

All referenced scripts will appear at the end of this appendix in the TSC Creation/Configuration Scripts section.

Log into the K2 CLI with an SSH client from any Host System and execute the following two scripts and single command to create and configure the SPC-1 ASUs:

- **tsc_creation-hosts.txt**
  - ➢ Creates a host group,
  - ➢ Creates a list of Host Systems and associates them to the newly created host group.
  - ➢ Configures the WWPNs of the FC HBA on each Host System to be mapped to the volumes created by the next script,
- **tsc_creation-volumes.txt**
  Create the volumes (SPC-1 Logical Volumes) that comprise the SPC-1 ASUs.
- Map the newly created volumes to the host group with the following command:
  volume mapping create host_group=spc volumes=”asu1_01, asu1_02, asu1_03, asu1_04, asu1_05, asu1_06, asu1_07, asu1_08, asu1_09,asu2_01, asu2_02, asu2_03, asu2_04, asu2_05, asu2_06, asu2_07, asu2_08, asu2_09, asu3_01, asu3_02”

TSC Creation/Configuration Scripts

tsc_creation-hosts.txt

```
volume host-group-create name=spc

volume host-create host_group=spc os_type=ESX name=winperf01
volume host-create host_group=spc os_type=ESX name=winperf04
volume host-create host_group=spc os_type=ESX name=winperf05
volume host-create host_group=spc os_type=ESX name=winperf06
volume host-create host_group=spc os_type=ESX name=winperf07
volume host-create host_group=spc os_type=ESX name=winperf08
volume host-create host_group=spc os_type=ESX name=winperf09
volume host-create host_group=spc os_type=ESX name=winperf10
volume host-create host_group=spc os_type=ESX name=winperf11
volume host-create host_group=spc os_type=ESX name=winperf12
volume host-create host_group=spc os_type=ESX name=winperf13
volume host-create host_group=spc os_type=ESX name=winperf14
volume host-create host_group=spc os_type=ESX name=winperf15
volume host-create host_group=spc os_type=ESX name=winperf16



volume host-change silent=true add_fc_port=50024f40c0370300 name=winperf01
volume host-change silent=true add_fc_port=50024f40c0370301 name=winperf01
volume host-change silent=true add_fc_port=50024f40b20c2b00 name=winperf01
volume host-change silent=true add_fc_port=50024f40b20c2b01 name=winperf01
volume host-change silent=true add_fc_port=2100001b32975f56 name=winperf04
volume host-change silent=true add_fc_port=2101001b32b75f56 name=winperf04
volume host-change silent=true add_fc_port=50024f40b15a0a00 name=winperf04
volume host-change silent=true add_fc_port=50024f40b15a0a01 name=winperf04
volume host-change silent=true add_fc_port=50024f40b15a0200 name=winperf05
volume host-change silent=true add_fc_port=50024f40b15a0201 name=winperf05
volume host-change silent=true add_fc_port=21000024ff45a618 name=winperf05
volume host-change silent=true add_fc_port=21000024ff45a619 name=winperf05
volume host-change silent=true add_fc_port=2100001b329d14ab name=winperf06
volume host-change silent=true add_fc_port=2101001b32bd14ab name=winperf06
volume host-change silent=true add_fc_port=21000024ff45a4d8 name=winperf06
volume host-change silent=true add_fc_port=21000024ff45a4d9 name=winperf06
volume host-change silent=true add_fc_port=50024f40dddd0200 name=winperf07
volume host-change silent=true add_fc_port=50024f40dddd0201 name=winperf07
volume host-change silent=true add_fc_port=21000024ff2efad4 name=winperf07
volume host-change silent=true add_fc_port=21000024ff2efad5 name=winperf07
volume host-change silent=true add_fc_port=50024f40b15a0900 name=winperf08
volume host-change silent=true add_fc_port=50024f40b15a0901 name=winperf08
volume host-change silent=true add_fc_port=21000024ff34add2 name=winperf08
volume host-change silent=true add_fc_port=21000024ff34add3 name=winperf08
volume host-change silent=true add_fc_port=21000024ff3565ec name=winperf09
volume host-change silent=true add_fc_port=21000024ff3565ed name=winperf09
volume host-change silent=true add_fc_port=2100001b32885a8e name=winperf09
volume host-change silent=true add_fc_port=2101001b32a85a8e name=winperf09
volume host-change silent=true add_fc_port=21000024ff356602 name=winperf10
volume host-change silent=true add_fc_port=21000024ff356603 name=winperf10
volume host-change silent=true add_fc_port=50024f40b21b1100 name=winperf10
volume host-change silent=true add_fc_port=50024f40b21b1101 name=winperf10
volume host-change silent=true add_fc_port=50024f40b21b1b00 name=winperf11
volume host-change silent=true add_fc_port=50024f40b21b1b01 name=winperf11
volume host-change silent=true add_fc_port=50024f40dddd0100 name=winperf11
volume host-change silent=true add_fc_port=50024f40dddd0101 name=winperf11
volume host-change silent=true add_fc_port=2100001b3297b756 name=winperf12
volume host-change silent=true add_fc_port=2101001b32b7b756 name=winperf12
volume host-change silent=true add_fc_port=2100001b329c9b53 name=winperf12
volume host-change silent=true add_fc_port=2101001b32bc9b53 name=winperf12
volume host-change silent=true add_fc_port=2100001b329d85a3 name=winperf13
volume host-change silent=true add_fc_port=2101001b32bd85a3 name=winperf13
volume host-change silent=true add_fc_port=2100001b3288468e name=winperf13
volume host-change silent=true add_fc_port=2101001b32a8468e name=winperf13
volume host-change silent=true add_fc_port=21000024ff45a65c name=winperf14
volume host-change silent=true add_fc_port=21000024ff45a65d name=winperf14
volume host-change silent=true add_fc_port=21000024ff34ae80 name=winperf14
volume host-change silent=true add_fc_port=21000024ff34ae81 name=winperf14
volume host-change silent=true add_fc_port=21000024ff45a400 name=winperf15
volume host-change silent=true add_fc_port=21000024ff45a401 name=winperf15
volume host-change silent=true add_fc_port=50024f40b15a0b00 name=winperf15
volume host-change silent=true add_fc_port=50024f40b15a0b01 name=winperf15
volume host-change silent=true add_fc_port=21000024ff45a3e6 name=winperf16
volume host-change silent=true add_fc_port=21000024ff45a3e7 name=winperf16
volume host-change silent=true add_fc_port=50024f40dddd0a00 name=winperf16
volume host-change silent=true add_fc_port=50024f40dddd0a01 name=winperf16
```

tsc_creation-volumes.txt

```
volume volume-create name=asu1_01 size=3000GB
volume volume-create name=asu1_02 size=3000GB
volume volume-create name=asu1_03 size=3000GB
volume volume-create name=asu1_04 size=3000GB
volume volume-create name=asu1_05 size=3000GB
volume volume-create name=asu1_06 size=3000GB
volume volume-create name=asu1_07 size=3000GB
volume volume-create name=asu1_08 size=3000GB
volume volume-create name=asu1_09 size=3000GB
volume volume-create name=asu2_01 size=3000GB
volume volume-create name=asu2_02 size=3000GB
volume volume-create name=asu2_03 size=3000GB
volume volume-create name=asu2_04 size=3000GB
volume volume-create name=asu2_05 size=3000GB
volume volume-create name=asu2_06 size=3000GB
volume volume-create name=asu2_07 size=3000GB
volume volume-create name=asu2_08 size=3000GB
volume volume-create name=asu2_09 size=3000GB
volume volume-create name=asu3_01 size=3000GB
volume volume-create name=asu3_02 size=3000GB
```

APPENDIX D: SPC-2 WORKLOAD GENERATOR STORAGE COMMANDS AND PARAMETER FILES

ASU Pre-Fill

Each Host System was assigned one or more LUNs *(SPC-2 Logical Volumes)* to pre-fill as a distinct execution sequence. This approach partitioned the pre-fill work, by LUN, across the Host Systems, allowing pre-fill sequences to execute simultaneously with each Host System responsible for its assigned LUN(s) rather than all LUNs. The result is a significant reduction in the time required to complete the entire ASU pre-fill operation.

The content of each command and parameter file used by each Host System is listed below.

Host System: win01, LUN: PHYSICALDRIVE7

```
compratio=1
sd=asu2_05,lun=\\.\PHYSICALDRIVE7,threads=32
wd=wd_asu2_05,sd=asu2_05,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win04, LUNs: PHYSICALDRIVE23, PHYSICALDRIVE15

```
compratio=1
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,threads=32
sd=asu2_06,lun=\\.\PHYSICALDRIVE15,threads=32
wd=wd_aasu1_01,sd=aasu1_01,rdpct=0,seek=-1,xfersize=128K
wd=wd_asu2_06,sd=asu2_06,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win05, LUNs: PHYSICALDRIVE24, PHYSICALDRIVE9

```
compratio=1
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,threads=32
sd=asu2_07,lun=\\.\PHYSICALDRIVE9,threads=32
wd=wd_asu1_02,sd=asu1_02,rdpct=0,seek=-1,xfersize=128K
wd=wd_asu2_07,sd=asu2_07,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win06, LUNs: PHYSICALDRIVE5, PHYSICALDRIVE3

```
compratio=1
sd=asu1_03,lun=\\.\PHYSICALDRIVE5,threads=32
sd=asu2_08,lun=\\.\PHYSICALDRIVE3,threads=32
wd=wd_asu1_03,sd=asu1_03,rdpct=0,seek=-1,xfersize=128K
wd=wd_asu2_08,sd=asu2_08,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win07, LUNs: PHYSICALDRIVE30, PHYSICALDRIVE31

```
compratio=1
sd=asu1_04,lun=\\.\PHYSICALDRIVE30,threads=32
sd=asu2_09,lun=\\.\PHYSICALDRIVE31,threads=32
wd=wd_asu1_04,sd=asu1_04,rdpct=0,seek=-1,xfersize=128K
wd=wd_asu2_09,sd=asu2_09,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win08, LUNs: PHYSICALDRIVE18, PHYSICALDRIVE19

```
compratio=1
sd=asu1_05,lun=\\.\PHYSICALDRIVE18,threads=32
sd=asu3_01,lun=\\.\PHYSICALDRIVE19,threads=32
wd=wd_asu1_05,sd=asu1_05,rdpct=0,seek=-1,xfersize=128K
wd=wd_asu3_01,sd=asu3_01,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win09, LUNs: PHYSICALDRIVE27, PHYSICALDRIVE3

```
compratio=1
sd=asu1_05,lun=\\.\PHYSICALDRIVE18,threads=32
sd=asu3_01,lun=\\.\PHYSICALDRIVE19,threads=32
wd=wd_asu1_05,sd=asu1_05,rdpct=0,seek=-1,xfersize=128K
wd=wd_asu3_01,sd=asu3_01,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win10, LUNs: PHYSICALDRIVE37

```
compratio=1
sd=asu1_07,lun=\\.\PHYSICALDRIVE37,threads=32
wd=wd_asu1_07,sd=asu1_07,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win11, LUN: PHYSICALDRIVE39

```
compratio=1
sd=asu1_08,lun=\\.\PHYSICALDRIVE39,threads=32
wd=wd_asu1_08,sd=asu1_08,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win12, LUN: PHYSICALDRIVE37

```
compratio=1
sd=asu1_09,lun=\\.\PHYSICALDRIVE37,threads=32
wd=wd_asu1_09,sd=asu1_09,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win13, LUN: PHYSICALDRIVE7

```
compratio=1
sd=asu2_01,lun=\\.\PHYSICALDRIVE7,threads=32
wd=wd_asu2_01,sd=asu2_01,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win14, LUN: PHYSICALDRIVE26

```
compratio=1
sd=asu2_02,lun=\\.\PHYSICALDRIVE26,threads=32
wd=wd_asu2_02,sd=asu2_02,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win15, LUN: PHYSICALDRIVE14

```
compratio=1
sd=asu2_03,lun=\\.\PHYSICALDRIVE14,threads=32
wd=wd_asu2_03,sd=asu2_03,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Host System: win16, LUN: PHYSICALDRIVE29

```
compratio=1
sd=asu2_04,lun=\\.\PHYSICALDRIVE29,threads=32
wd=wd_asu2_04,sd=asu2_04,rdpct=0,seek=-1,xfersize=128K
rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10
```

Common Commands/Parameters – LFP, LDQ and VOD

The following command/parameter lines appear in each of the command and parameter files for the Large File Processing (LFP), Large Database Query (LDQ) and Video on Demand (VOD) Tests. The command lines are only listed below to eliminate redundancy.

```
host=localhost,jvms=4,maxstreams=1500,java=("C:\Program
Files\Java\jre7\bin\java.exe","-Xmx5000m -Xms1536m -Xss96k"),shell=spc2

host=(win04,win04),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win05,win05),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win06,win06),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win07,win07),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win08,win08),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win09,win09),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win10,win10),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win11,win11),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win12,win12),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win13,win13),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win14,win14),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win15,win15),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500

host=(win16,win16),java=("C:\Program Files\Java\jre7\bin\java.exe","-Xmx5000m -
Xms1536m -Xss96k"),shell=spc2,jvms=4,maxstreams=1500


sd=default,host=localhost
sd=aasu1_01,lun=\\.\PHYSICALDRIVE3,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE4,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE15,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE13,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE17,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE16,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE14,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE20,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE19,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE11,size=3006477107200
```

```
sd=asu2_02,lun=\\.\PHYSICALDRIVE9,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE12,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE10,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE7,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE6,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE8,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE18,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE5,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE21,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE22,size=3006477107200

sd=default,host=win04
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE9,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE7,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE17,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE5,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE11,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE10,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE6,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE13,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE14,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE3,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE12,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE15,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE8,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE16,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE4,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE18,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE19,size=3006477107200

sd=default,host=win05
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE16,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE4,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE14,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE11,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE10,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE6,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE7,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE13,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE12,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE5,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE3,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE17,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE8,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE9,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE15,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE18,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE19,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE20,size=3006477107200

sd=default,host=win06
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE5,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE7,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE6,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE8,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE11,size=3006477107200
```

```
sd=asu1_08,lun=\\.\PHYSICALDRIVE10,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE17,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE16,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE15,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE9,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE13,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE14,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE12,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE3,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE4,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE18,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE19,size=3006477107200

sd=default,host=win07
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE28,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE30,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE40,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE27,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE36,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE37,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE39,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE33,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE38,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE26,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE32,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE34,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE35,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE29,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE31,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE41,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE42,size=3006477107200

sd=default,host=win08
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE15,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE7,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE18,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE6,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE17,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE14,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE11,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE8,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE12,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE4,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE10,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE5,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE13,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE9,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE16,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE3,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE19,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE20,size=3006477107200

sd=default,host=win09
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE38,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE33,size=3006477107200
```

```
sd=asu1_05,lun=\\.\PHYSICALDRIVE34,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE27,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE29,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE32,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE36,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE39,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE30,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE35,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE26,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE37,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE31,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE40,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE28,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE41,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE3,size=3006477107200

sd=default,host=win10
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE40,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE27,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE32,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE30,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE37,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE28,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE36,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE29,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE38,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE35,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE26,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE33,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE31,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE34,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE39,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE41,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE42,size=3006477107200

sd=default,host=win11
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE31,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE32,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE40,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE28,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE39,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE33,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE37,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE34,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE29,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE38,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE36,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE26,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE35,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE30,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE27,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE41,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE42,size=3006477107200

sd=default,host=win12
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
```

```
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE38,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE31,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE29,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE33,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE28,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE39,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE37,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE27,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE34,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE36,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE40,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE30,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE35,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE32,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE26,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE41,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE3,size=3006477107200

sd=default,host=win13
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE15,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE10,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE12,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE16,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE17,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE13,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE9,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE7,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE14,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE3,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE5,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE8,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE11,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE4,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE6,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE18,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE19,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE20,size=3006477107200

sd=default,host=win14
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE34,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE36,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE29,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE35,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE37,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE33,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE27,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE26,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE39,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE31,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE40,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE30,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE41,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE28,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE32,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE38,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE3,size=3006477107200
```

```
sd=default,host=win15
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE7,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE13,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE17,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE3,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE6,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE4,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE15,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE12,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE10,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE14,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE9,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE16,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE8,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE18,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE5,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE11,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE19,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE20,size=3006477107200

sd=default,host=win16
sd=aasu1_01,lun=\\.\PHYSICALDRIVE23,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE24,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE25,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE28,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE37,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE32,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE39,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE34,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE36,size=3006477107200
sd=asu2_01,lun=\\.\PHYSICALDRIVE27,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE31,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE26,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE29,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE40,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE30,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE35,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE33,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE38,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE41,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE42,size=3006477107200
```

Common Commands/Parameters – Persistence

The following command/parameter lines appear in each of the command and parameter files for the SPC-2 Persistence Test. The command lines are only listed below to eliminate redundancy.

```
host=localhost,jvms=8,maxstreams=200
sd=aasu1_01,lun=\\.\PHYSICALDRIVE3,size=3006477107200
sd=asu1_02,lun=\\.\PHYSICALDRIVE4,size=3006477107200
sd=asu1_03,lun=\\.\PHYSICALDRIVE15,size=3006477107200
sd=asu1_04,lun=\\.\PHYSICALDRIVE13,size=3006477107200
sd=asu1_05,lun=\\.\PHYSICALDRIVE17,size=3006477107200
sd=asu1_06,lun=\\.\PHYSICALDRIVE16,size=3006477107200
sd=asu1_07,lun=\\.\PHYSICALDRIVE14,size=3006477107200
sd=asu1_08,lun=\\.\PHYSICALDRIVE20,size=3006477107200
sd=asu1_09,lun=\\.\PHYSICALDRIVE19,size=3006477107200
```

```
sd=asu2_01,lun=\\.\PHYSICALDRIVE11,size=3006477107200
sd=asu2_02,lun=\\.\PHYSICALDRIVE9,size=3006477107200
sd=asu2_03,lun=\\.\PHYSICALDRIVE12,size=3006477107200
sd=asu2_04,lun=\\.\PHYSICALDRIVE10,size=3006477107200
sd=asu2_05,lun=\\.\PHYSICALDRIVE7,size=3006477107200
sd=asu2_06,lun=\\.\PHYSICALDRIVE6,size=3006477107200
sd=asu2_07,lun=\\.\PHYSICALDRIVE8,size=3006477107200
sd=asu2_08,lun=\\.\PHYSICALDRIVE18,size=3006477107200
sd=asu2_09,lun=\\.\PHYSICALDRIVE5,size=3006477107200
sd=asu3_01,lun=\\.\PHYSICALDRIVE21,size=3006477107200
sd=asu3_02,lun=\\.\PHYSICALDRIVE22,size=3006477107200
```

Large File Processing Test (LFP)

```
* Large File Processing Test (LFP)
```

Common Commands/Parameters – LFP, LDQ and VOD

```
maxlatestart=0
reportinginterval=5
segmentlength=512m
rd=default,rampup=180,periods=90,measurement=180,runout=45,rampdown=15,buffers=1

* LFP, "write" Test Phase
* Test Run Sequence 1
rd=default,rdpct=0,xfersize=1024k
rd=TR1-s85_SPC-2-FP,streams=85
rd=TR2-s42_SPC-2-FP,streams=42
rd=TR3-s21_SPC-2-FP,streams=21
rd=TR4-s10_SPC-2-FP,streams=10
rd=TR5-s1_SPC-2-FP,streams=1

* Test Run Sequence 2
rd=default,xfersize=256k
rd=TR6-s200_SPC-2-FP,streams=200
rd=TR7-s100_SPC-2-FP,streams=100
rd=TR8-s50_SPC-2-FP,streams=50
rd=TR9-s25_SPC-2-FP,streams=25
rd=TR10-s1_SPC-2-FP,streams=1

* LFP, "read-write" Test Phase
* Test Run Sequence 3
rd=default,rdpct=50,xfersize=1024k
rd=TR11-s160_SPC-2-FP,streams=160
rd=TR12-s80_SPC-2-FP,streams=80
rd=TR13-s40_SPC-2-FP,streams=40
rd=TR14-s20_SPC-2-FP,streams=20
rd=TR15-s1_SPC-2-FP,streams=1

* Test Run Sequence 4
rd=default,xfersize=256k
rd=TR16-s460_SPC-2-FP,streams=460
rd=TR17-s230_SPC-2-FP,streams=230
rd=TR18-s115_SPC-2-FP,streams=115
rd=TR19-s57_SPC-2-FP,streams=57
rd=TR20-s1_SPC-2-FP,streams=1

* LFP, "read" Test Phase
* Test Run Sequence 5
rd=default,rdpct=100,xfersize=1024k
rd=TR21-s130_SPC-2-FP,streams=130
rd=TR22-s65_SPC-2-FP,streams=65
```

```
rd=TR23-s32_SPC-2-FP,streams=32
rd=TR24-s16_SPC-2-FP,streams=16
rd=TR25-s1_SPC-2-FP,streams=1

* Test Run Sequence 6
rd=default,xfersize=256k
rd=TR26-s475_SPC-2-FP,streams=475
rd=TR27-s237_SPC-2-FP,streams=237
rd=TR28-s118_SPC-2-FP,streams=118
rd=TR29-s59_SPC-2-FP,streams=59
rd=TR30-s1_SPC-2-FP,streams=1
```

Large Database Query Test (LDQ)

```
* Large Database Query Test (LDQ)
```

Common Commands/Parameters – LFP, LDQ and VOD

```
maxlatestart=0
reportinginterval=5
segmentlength=512m
rd=default,rdpct=99,rampup=180,periods=90,measurement=180,runout=45,rampdown=15

* LDQ, 1024 KiB Test Phase
* Test Run Sequence 1
rd=default,xfersize=1024k,buffers=4
rd=TR1-s35_SPC-2-DQ,streams=35
rd=TR2-s17_SPC-2-DQ,streams=17
rd=TR3-s8_SPC-2-DQ,streams=8
rd=TR4-s4_SPC-2-DQ,streams=4
rd=TR5-s1_SPC-2-DQ,streams=1
* Test Run Sequence 2
rd=default,buffers=1
rd=TR6-s135_SPC-2-DQ,streams=135
rd=TR7-s67_SPC-2-DQ,streams=67
rd=TR8-s33_SPC-2-DQ,streams=33
rd=TR9-s16_SPC-2-DQ,streams=16
rd=TR10-s1_SPC-2-DQ,streams=1

* LDQ, 64 KiB Test Phase
* Test Run Sequence 3
rd=default,xfersize=64k,buffers=4
rd=TR11-s460_SPC-2-DQ,streams=460
rd=TR12-s230_SPC-2-DQ,streams=230
rd=TR13-s115_SPC-2-DQ,streams=115
rd=TR14-s57_SPC-2-DQ,streams=57
rd=TR15-s1_SPC-2-DQ,streams=1
* Test Run Sequence 4
rd=default,buffers=1
rd=TR16-s1350_SPC-2-DQ,streams=1350
rd=TR17-s675_SPC-2-DQ,streams=675
rd=TR18-s337_SPC-2-DQ,streams=337
rd=TR19-s168_SPC-2-DQ,streams=168
rd=TR20-s1_SPC-2-DQ,streams=1
```

Video on Demand Delivery (VOD)

```
* Video on Demand Test (VOD)
```

Common Commands/Parameters – LFP, LDQ and VOD

```
maxlatestart=0
videosegmentduration=1200
maxlatevod=0
reportinginterval=5
rd=default,rampup=1200,periods=600,measurement=7200,runout=45,rampdown=15,buffers=8
rd=TR1-s40000_SPC-2-VOD,streams=40000
```

SPC-2 Persistence Test Run 1 (write phase)

Common Commands/Parameters – Persistence

```
maxlatestart=1
reportinginterval=5
segmentlength=512m
rd=default,rampup=180,periods=90,measurement=300,runout=0,rampdown=0,buffers=1
rd=default,rdpct=0,xfersize=1024k
rd=TR1_SPC-2-persist-w,streams=200
```

SPC-2 Persistence Test Run 2 (read phase)

Common Commands/Parameters – Persistence

```
maxlatestart=1
reportinginterval=5
segmentlength=512m
maxpersistenceerrors=10
rd=default,buffers=1,rdpct=100,xfersize=1024k
rd=TR1_SPC-2-persist-r
```

APPENDIX E: SPC-2 WORKLOAD GENERATOR EXECUTION COMMANDS AND PARAMETERS

‘Master’ Execution Script

This ‘master’ script is invoked from a “start-up” script, **tests_sequence**, which is listed below. This script is responsible for the execution of the required ASU Pre-Fill, Large File Processing (LFP), Large Database Query (LDQ), Video on Demand Delivery and Persistence Tests.

```
#!/bin/bash


ROOTDIR='/cygdrive/c/spc/run/'
JAVA=$ROOTDIR"java7 -Xmx12000m -Xms12000m"
SPC2="/cygdrive/c/spc/spc2/spc2.bat"
BSU=12420
PERSIST1_FACTOR=10
SUSTAIN_TIME=86400
TIME=600
STARTUP=180
INTERVAL=60
#STREAMS=$[$BSU/30 + 50]
STREAMS=200
DISKS=$ROOTDIR"disks"
DISKS_SIZES=$ROOTDIR"disks_sizes"
PERSISTDIR=$ROOTDOIR"persist/"
PERSIST_W='persist_write'
PERSIST_R='persist_read'
PHASES='sustain:100 ramp_100:100 ramp_95:95 ramp_90:90 ramp_80:80 ramp_50:50
ramp_10:10 repeat1_lrt:10 repeat1_iops:100 repeat2_lrt:10 repeat2_iops:100'

mkdir -p $PERSISTDIR
cd $ROOTDIR

echo "Updating disk mapping"
./all_update_disks
sleep 3

echo "Preparing files"
# creating main tests files
for phase in $PHASES; do
     NAME=${phase%:*}
     FACTOR=${phase#*:}
     if [ $NAME = "sustain" ];
     then
            ACTUAL_TIME=$SUSTAIN_TIME
     else
            ACTUAL_TIME=$TIME
     fi
     cat $ROOTDIR"spc1.cfg" > $NAME".txt"
     echo "rd=$NAME,bsus=$[$BSU * $FACTOR /
100],startup=$STARTUP,elapsed=$ACTUAL_TIME,interval=$INTERVAL" >> $NAME".txt"
done
# creating persistence test files
cd $PERSISTDIR
echo "host=localhost,jvms=8,maxstreams=$STREAMS" > $PERSIST_W".txt"
cat $DISKS_SIZES >> $PERSIST_W".txt"
echo "maxlatestart=1" >> $PERSIST_W".txt"
echo "reportinginterval=5" >> $PERSIST_W".txt"
echo "segmentlength=512m" >> $PERSIST_W".txt"
echo
"rd=default,rampup=180,periods=90,measurement=300,runout=0,rampdown=0,buffers=1" >>
$PERSIST_W".txt"
echo "rd=default,rdpct=0,xfersize=1024k" >> $PERSIST_W".txt"
echo "rd=TR1_SPC-2-persist-w,streams=$STREAMS" >> $PERSIST_W".txt"

echo "host=localhost,jvms=8,maxstreams=$STREAMS" > $PERSIST_R".txt"
cat $DISKS_SIZES >> $PERSIST_R".txt"
echo "maxlatestart=1" >> $PERSIST_R".txt"
echo "reportinginterval=5" >> $PERSIST_R".txt"
echo "segmentlength=512m" >> $PERSIST_R".txt"
echo "maxpersistenceerrors=10" >> $PERSIST_R".txt"
echo "rd=default,buffers=1,rdpct=100,xfersize=1024k" >> $PERSIST_R".txt"
echo "rd=TR1_SPC-2-persist-r" >> $PERSIST_R".txt"
cd $ROOTDIR

case "$1"  in

     prepssd)
            echo "Running vdbench pre-fill from multiple hosts"
            ./all_prepssd
            ./all_killslaves
            ;;
     main)
            echo "Running test with low level commands"
            for phase in $PHASES; do
                  if [ $phase == 'restart' ]
                  then
                         echo "Restarting slaves"
                         #Calling all_runslaves will kill and then run the slaves on
each host"
                         ./all_runslaves
                         sleep 10
                  else
                         ./all_runslaves
                         sleep 10
                         NAME=${phase%:*}
                         CMD="$JAVA spc1 -w SPC1 -f $NAME.txt -o $NAME SPCOut"
                         echo $CMD
                         $CMD
                         RC=$?

                         # move the slave directories aside
                         ./all_killslaves
                         ./all_run_cmd "rm -rf "$ROOTDIR"slaves_$NAME/"
                         ./all_run_cmd "mv "$ROOTDIR"slaves "$ROOTDIR"slaves_$NAME/"

                         # Check the run return code
                         if [ $RC -gt 0 ]
                         then
                                echo "Test ended with error code $RC. Stopping run."
                                exit $RC
                         fi
                         if [ `tail $ROOTDIR$NAME/logfile.html | grep Error | wc -l`
-gt 0  ]
                         then
                                echo "Found an error - waiting"
                                sleep 300
                                if [ `grep "Task IO_task stopped after 3 minutes"
/tmp/slaves.win* | wc -l` -gt 0 ]
                                then
```

```
                                  echo "There was a stuck slave. we can
continue"
                            else
                                  echo "An error detected in phase $NAME"
                                  exit 1
                            fi
                      fi

                      # Remake the slave tree
                      ./all_run_cmd ""$ROOTDIR"mktree"
                      sleep 10

                fi
          done
          ;;
    spc2_persist_w)
          echo "Running persist_w"
          ./all_killslaves
          cd $PERSISTDIR
          $SPC2 -f $PERSIST_W".txt" -o init -init
          $SPC2 -f $PERSIST_W".txt" -o $PERSIST_W
          ;;
    spc2_persist_r)
          echo "Running persist_r"
          ./all_killslaves
          cd $PERSISTDIR
          $SPC2 -f $PERSIST_R".txt" -o $PERSIST_R
          ;;
    spc1_persist1)
          echo "Running persist1"
          ./all_killslaves
          cp disks $PERSISTDIR"spc1.cfg"
          cd $PERSISTDIR
          $JAVA persist1 -b $[$BSU * $PERSIST1_FACTOR / 100];
          ;;
    spc1_persist2)
          echo "Running persist2"
          ./all_killslaves
          cp disks $PERSISTDIR"spc1.cfg"
          cd $PERSISTDIR
          $JAVA persist2;
          ;;
    spc2_main)
          echo "creating SPC-2 files"
          source createspc2_files
          echo "Running spc-2 main"
          ./all_killslaves
          sleep 2
          ./all_startremote
          $SPC2 -f lfp.txt -o init -init
          ./all_run_cmd 'rm -rf /cygdrive/c/spc/run/LFP/*'
          $SPC2 -f lfp.txt -o LFP
          ./all_run_cmd 'rm -rf /cygdrive/c/spc/run/LDQ/*'
          $SPC2 -f ldq.txt -o LDQ
           ./all_run_cmd 'rm -rf /cygdrive/c/spc/run/VOD/*'
          $SPC2 -f vod.txt -o VOD
          ./all_copy_spc2_slaves
          ;;
    *)
          echo "no stage supplied -
prepssd|main|spc2_persist_w|spc2_persist_r|spc1_persist1|spc1_persist2|spc2_main"
          ;;
esac
```

Referenced Scripts

tests_sequence

This script invokes the above 'master' script. The various parameters in this script will determine what portion of the 'master' script is executed.

The first invocation of the script (***./tests_squence spc2_main***), listed below, will execute the required ASU Pre-Fill, Large File Processing (LFP), Large Database Query (LDQ), Video on Demand Delivery (VOD) and SPC-2 Persistence Test Run 1.

```
#!/bin/bash
ROOTDIR='/cygdrive/c/spc/run/'
cd $ROOTDIR

case "$1"  in
     spc1_main | spc2_main)
            echo "Collecting inventory"
            echo "Inventory start: `date`" > inventory.pre
            ssh root@kcs499 "echo system server-show table=inventory |
/opt/km/cli/km-cli" >> inventory.pre
            echo "Inventory end: `date`" >> inventory.pre
            sleep 300
            ./tests prepssd
            case "$1"  in
                   spc1_main)
                          ./tests main
                          ./tests spc1_persist1
                          ;;
                   spc2_main)
                          ./tests spc2_main
                          ;;
            esac
            ./tests spc2_persist_w
            ;;
     post)
            ./tests spc2_persist_r
            sleep 300
            echo "Collecting inventory"
            echo "Inventory start: `date`" > inventory.post
            ssh root@kcs499 "echo system server-show table=inventory |
/opt/km/cli/km-cli" >> inventory.post
            echo "Inventory end: `date`" >> inventory.post
            ;;
esac
```

A second invocation, (***./tests_squence post***), of the above script will execute SPC-2 Persistence Test Run 2 after the required TSC power off/power on cycle.

hosts

Contains a list of the current Host Systems and the mapping from the K2 volumes to the SPC-2 ASU.

```
#!/bin/bash

SLAVES=18
MASTER="win01"
```

```
hosts_slaves="win01:$SLAVES win04:$SLAVES win05:$SLAVES win06:$SLAVES win07:$SLAVES
win08:$SLAVES win09:$SLAVES win10:$SLAVES win11:$SLAVES win12:$SLAVES win13:$SLAVES
win14:$SLAVES win15:$SLAVES win16:$SLAVES"

hosts_slaves="`echo $hosts_slaves | tr ' ' '\n' | grep -v :0 | xargs`"


disk_map='"asu1_01:bcac0001 asu1_02:bcac0002 asu1_03:bcac0003 asu1_04:bcac0004
asu1_05:bcac0005 asu1_06:bcac0006 asu1_07:bcac0007 asu1_08:bcac0008 asu1_09:bcac0009
asu2_01:bcac000a asu2_02:bcac000b asu2_03:bcac000c asu2_04:bcac000d asu2_05:bcac000e
asu2_06:bcac000f asu2_07:bcac0010 asu2_08:bcac0011 asu2_09:bcac0012 asu3_01:bcac0013
asu3_02:bcac0014"'
```

all_update_disks

Invokes the **update_disks** script for each Host System.

```
#!/bin/bash

source hosts
I=0
for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     SLAVES=${host_slave#*:}
     if [ $SLAVES -gt 0 ]; then
            ssh -n -f $NAME "/cygdrive/c/spc/run/update_disks $disk_map $I `echo
$hosts_slaves | wc -w` $1 2>/tmp/update_disks.err" > /tmp/update_disks.$NAME &
     fi
     I=$[ $I + 1 ]
done
wait
echo "Disk count , LB policy and path count:"
grep -A 1000 New  /tmp/update_disks.* | grep count | sort | uniq -c
echo "Done updating disks"
```

update_disks

This script:

- Rescans the LUNs
- Creates the mapping between the K2 volumes and LUNs using the **get_disks** script
- Ensures the LUNs are all online and initialized
- Ensures the load balancing policy is LDQ using the **LB_policy_tool.vbs** script
- Creates the directories for Slave JVM output and recreates the Slave JVM configuration files using the **mktree** script
- Creates the 'master' SPC-1 configuration file ("spc1.cfg") using the **createspc1cfg** script

```
#!/bin/bash

FILENAME=diskpart_script
ROOTDIR='/cygdrive/c/spc/run/'
BASHSCRIPT=$ROOTDIR$FILENAME

# rescan for disks
echo 'rescan' > $BASHSCRIPT
schtasks.exe /Run /TN run_diskpart_script
```

```
# get the kaminario disks
/cygdrive/c/spc/run/get_disks "$1" $2 $3

if [ $4 ]; then
     # make all disks online
     rm -f $BASHSCRIPT
     for disk in `cat $ROOTDIR"disks"`; do
            echo "SELECT DISK="`echo $disk |  grep -o  'PHYSICALDRIVE[0-9]*' | cut  -
-complement -c 1-13` >> $BASHSCRIPT
            echo "ATTRIBUTES DISK CLEAR READONLY NOERR" >> $BASHSCRIPT
            echo "ONLINE DISK NOERR" >> $BASHSCRIPT
            echo "CONVERT MBR NOERR" >> $BASHSCRIPT
     done

     schtasks.exe /Run /TN run_diskpart_script
fi

# Set the LB policy to LQD
 cscript.exe "C:\spc\run\LB_policy_tool.vbs" 3

# Remake the slave tree
$ROOTDIR"mktree"

# Create the spc config file
$ROOTDIR"createspc1cfg"
```

get_disks

Uses ***wmic*** to create the map from K2 volumes name on the windows machine to the ASUs. This script prepares the list of SD entries to be used in the configuration files. The script also created the configuration file for the ASU pre-fill phase. Each server is given a distinct and mutual exclusive set of disks to work with in the ASU pre-fill phase.

```
#!/bin/bash

if [ $# == 0 ]; then
     echo "Missing disk map (e.g:  \"asu1_1:c0210001 asu2_1:c0210002
asu3_1:c0200002\")"
     exit
fi
ROOTDIR='/cygdrive/c/spc/run/'
DISKS=$ROOTDIR"disks"
DISKS_SIZES=$ROOTDIR"disks_sizes"
PREPSSD=$ROOTDIR"prepssd.txt"
WDTMP=$ROOTDIR"wd.tmp"
rm -f $DISKS
rm -f "$DISKS"_tmp
rm -f $WDTMP
rm -f $DISKS_SIZES
rm -f "$DISKS_SIZES"_tmp
echo "compratio=1" > $PREPSSD

wmic_disks="`wmic diskdrive get name,serialnumber,model,size | tr -s ' ' ':' | grep
KMNRIO | cut -d ':' -f 6-8`"
ind=0
for l in $wmic_disks; do
     DISK=${l%:*:*}
     SERIAL_SIZE=${l#*:}
     SERIAL=${SERIAL_SIZE%:*}
     SIZE=${SERIAL_SIZE#*:}
     SIZE=`/cygdrive/c/sg_utils/sg_readcap.exe  $DISK | grep size | cut -d " " -f 6`
```

```
      echo "A: " $ind $3 $2
      for map in $1 ; do
             ASU=${map%:*}
             ID=${map#*:}
             if [ "$SERIAL" = "$ID" ]
             then
                    ind=$[ $ind + 1 ]
                    echo sd=$ASU,lun=$DISK >> "$DISKS"_tmp
                    echo sd=$ASU,lun=$DISK,size=$SIZE >> "$DISKS_SIZES"_tmp
                    if [ "`expr $ind % $3`" = "$2" ]
                    then
                          echo "B: " $ind $3 $2
                          echo sd=$ASU,lun=$DISK,threads=32 >> $PREPSSD
                          echo wd=wd_$ASU,sd=$ASU,rdpct=0,seek=-1,xfersize=128K >>
$WDTMP
                    fi
             fi
      done
done

cat "$DISKS"_tmp | sort > $DISKS
cat "$DISKS_SIZES"_tmp | sort > $DISKS_SIZES

cat $WDTMP >> $PREPSSD
echo rd=PREPSSD,wd=wd*,iorate=max,elapsed=36000,interval=10 >> $PREPSSD
```

LB_policy_tool.vbs

VB script to Set the LB MPIO policy of the disks to LQD.

```
Option Explicit



const LB_POLICY_DONT_CHANGE           = -1
const LB_POLICY_ROUND_ROBIN           = 2
const LB_POLICY_LEAST_QUEUE_DEPTH     = 4

'''Get the user selection
'''=====================
Dim args
Dim strMsg
Dim strInput
dim intNewValue
Dim myFSO, WriteStuff

strMsg = "Load blanace control tool for MPIO" & vbCr & vbCr & _
      " {1} Query the current load balance policy" & vbcr & vbCr & _
      " {2} Set load balance policy to ROUND ROBIN" & vbCr & vbCr & _
      " {3} Change load blance policy ro LEAST QUEUE DEPTH"

args = WScript.Arguments.Count
If args < 1 then
'''  Wscript.Echo(strMsg)
  WScript.Quit
end If


select Case WScript.Arguments.Item(0)
      Case "1"
             intNewValue = LB_POLICY_DONT_CHANGE
      Case "2"
```

```
            intNewValue = LB_POLICY_ROUND_ROBIN
     Case "3"
            intNewValue = LB_POLICY_LEAST_QUEUE_DEPTH
     Case Else
            if (strInput <> "") Then
                  wScript.echo  "unsupprted command  (" & strInput & ")"
            end if
            wScript.Quit
End Select



'''Query the current LB policy
'''==========================

Dim objWMIService, objProcess, colProcess, objProcess2, colProcess2, Policy(1000)
Dim strComputer, strList
Dim strRes, strPolicy
Dim i

strComputer = "."


Set objWMIService = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\\" _
& strComputer & "\root\wmi")

Set colProcess = objWMIService.ExecQuery _
("Select * from DSM_QueryLBPolicy")

strRes="LB policies before update" & vbCr & vbCr
WScript.Echo strRes
i=0
For Each objProcess in colProcess
     Set Policy(i) = objProcess.LoadBalancePolicy
     strPolicy = Policy(i).LoadBalancePolicy
     select case strPolicy
            case LB_POLICY_ROUND_ROBIN
                  strPolicy = strPolicy & " (ROUND ROBIN)"
            case LB_POLICY_LEAST_QUEUE_DEPTH
                  strPolicy = strPolicy & " (LEAST QUEUE DEPTH)"
     end select
     strRes = "LoadBalancePolicy= "& strPolicy & " Path count=" &
Policy(i).DSMPathCount & vbCr
     WScript.Echo strRes
     i=i+1
Next

'''Set the new policy
'''=================

'''If no change is required quit
If (intNewValue = LB_POLICY_DONT_CHANGE      ) Then wScript.Quit


Set colProcess = objWMIService.ExecQuery _
("Select * from DSM_LB_Operations")

WScript.Echo ""
WScript.Echo "New LB policies" & vbCr & vbCr
i=0
For Each objProcess in colProcess
     Policy(i).LoadBalancePolicy = intNewValue
```

```
      strPolicy = Policy(i).LoadBalancePolicy
      select case strPolicy
            case LB_POLICY_ROUND_ROBIN
                  strPolicy = strPolicy & " (ROUND ROBIN)"
            case LB_POLICY_LEAST_QUEUE_DEPTH
                  strPolicy = strPolicy & " (LEAST QUEUE DEPTH)"
      end select

      objProcess.DsmSetLoadBalancePolicy(Policy(i))
      WScript.Echo "Disk Name: " & objProcess.InstanceName
      strRes = "LoadBalancePolicy= "& strPolicy & " Path count=" &
Policy(i).DSMPathCount
      WScript.Echo strRes
      WScript.Echo ""

      i=i+1
Next


WScript.Quit
```

mktree

create the run directories for the slaves with the configuration file

```
#!/bin/bash

NUMOFSLAVES=50
HOST=`hostname |  tr '[A-Z]' '[a-z]' | tr -d '\n' | tr -d '\r'`
ROOTDIR='/cygdrive/c/spc/run/slaves/'

rm -rf $ROOTDIR
mkdir $ROOTDIR

MASTER='win01'

FILENAME=$HOST"_"
STARTDIR=100

ENDDIR=$[$STARTDIR + $NUMOFSLAVES]
for NDIR in `seq $STARTDIR $ENDDIR`; do

        DIR=$ROOTDIR$FILENAME$NDIR
        CFGFILE=$DIR/$FILENAME$NDIR.txt
        HOSTNAME=$FILENAME$NDIR
        /bin/mkdir -p $DIR
        /bin/echo "master=$MASTER" > $CFGFILE
        /bin/echo "host=$HOSTNAME" >> $CFGFILE
           cat $ROOTDIR"../disks" >> $CFGFILE
done
```

createspc1cfg

Creates the SPC-1 configuration file, for use with SPC-1.

```
#!/bin/bash

ROOTDIR='/cygdrive/c/spc/run/'
source $ROOTDIR"hosts"

CFGFILE=$ROOTDIR"spc1.cfg"
SLAVELIST=""
```

```
for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     SLAVES=${host_slave#*:}
     FIRSTSLAVE=100

     LASTSLAVE=$[$FIRSTSLAVE    + $SLAVES - 1]
     for SLAVENUM in `seq $FIRSTSLAVE $LASTSLAVE`; do
            SLAVENAME=$NAME"_"$SLAVENUM
            SLAVELIST="$SLAVELIST,$SLAVENAME"
     done
done
SLAVELIST=${SLAVELIST:1}

/bin/echo "host=master" > $CFGFILE
/bin/echo "slaves=($SLAVELIST)" >> $CFGFILE
cat $ROOTDIR"disks" >> $CFGFILE
```

createspc2_files

Creates the SPC-2 command and parameter files.

```
# create hosts and disk config

ROOTDIR='/cygdrive/c/spc/run/'
DISKS_SIZES=$ROOTDIR"disks_sizes"
source hosts
SPC2_HOSTS=spc2_hosts
SPC2_DISKS=spc2_disks
LFP=lfp.txt
LDQ=ldq.txt
VOD=vod.txt

LFP_WRITE_1024=85
LFP_WRITE_256=200
LFP_RW_1024=160
LFP_RW_256=460
LFP_READ_1024=130
LFP_READ_256=475
LDQ_1024_4=35
LDQ_1024_1=135
LDQ_64_4=460
LDQ_64_1=1350
VOD_STREAMS=40000


echo "host=localhost,jvms=4,maxstreams=1500,java=(\"C:\Program
Files\Java\jre7\bin\java.exe\",\"-Xmx5000m -Xms1536m -Xss96k\"),shell=spc2" >
$SPC2_HOSTS
echo "sd=default,host=localhost" > $SPC2_DISKS
cat $DISKS_SIZES >> $SPC2_DISKS
echo "" >> $SPC2_DISKS
for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     SLAVES=${host_slave#*:}
     if [ $NAME != "win01" -a $SLAVES -gt 0 ]; then
            echo "host=($NAME,$NAME),java=(\"C:\Program
Files\Java\jre7\bin\java.exe\",\"-Xmx5000m -Xms1536m -
Xss96k\"),shell=spc2,jvms=4,maxstreams=1500" >> $SPC2_HOSTS
            echo "sd=default,host=$NAME" >> $SPC2_DISKS
            ssh $NAME "cat $DISKS_SIZES " >> $SPC2_DISKS
            echo "" >> $SPC2_DISKS
     fi
done

# LARGE FILE PROCESSING
echo "* Large File Processing Test (LFP)" > $LFP
cat $SPC2_HOSTS >> $LFP
echo "" >> $LFP
cat $SPC2_DISKS >> $LFP

echo "
maxlatestart=0
reportinginterval=5
segmentlength=512m
rd=default,rampup=180,periods=90,measurement=180,runout=45,rampdown=15,buffers=1

* LFP, \"write\" Test Phase
* Test Run Sequence 1
rd=default,rdpct=0,xfersize=1024k
rd=TR1-s"$LFP_WRITE_1024"_SPC-2-FP,streams=$LFP_WRITE_1024
rd=TR2-s$[$LFP_WRITE_1024/2]_SPC-2-FP,streams=$[$LFP_WRITE_1024/2]
rd=TR3-s$[$LFP_WRITE_1024/4]_SPC-2-FP,streams=$[$LFP_WRITE_1024/4]
rd=TR4-s$[$LFP_WRITE_1024/8]_SPC-2-FP,streams=$[$LFP_WRITE_1024/8]
rd=TR5-s1_SPC-2-FP,streams=1

* Test Run Sequence 2
rd=default,xfersize=256k
rd=TR6-s"$LFP_WRITE_256"_SPC-2-FP,streams=$LFP_WRITE_256
rd=TR7-s$[$LFP_WRITE_256/2]_SPC-2-FP,streams=$[$LFP_WRITE_256/2]
rd=TR8-s$[$LFP_WRITE_256/4]_SPC-2-FP,streams=$[$LFP_WRITE_256/4]
rd=TR9-s$[$LFP_WRITE_256/8]_SPC-2-FP,streams=$[$LFP_WRITE_256/8]
rd=TR10-s1_SPC-2-FP,streams=1

* LFP, \"read-write\" Test Phase
* Test Run Sequence 3
rd=default,rdpct=50,xfersize=1024k
rd=TR11-s"$LFP_RW_1024"_SPC-2-FP,streams=$LFP_RW_1024
rd=TR12-s$[$LFP_RW_1024/2]_SPC-2-FP,streams=$[$LFP_RW_1024/2]
rd=TR13-s$[$LFP_RW_1024/4]_SPC-2-FP,streams=$[$LFP_RW_1024/4]
rd=TR14-s$[$LFP_RW_1024/8]_SPC-2-FP,streams=$[$LFP_RW_1024/8]
rd=TR15-s1_SPC-2-FP,streams=1

* Test Run Sequence 4
rd=default,xfersize=256k
rd=TR16-s"$LFP_RW_256"_SPC-2-FP,streams=$LFP_RW_256
rd=TR17-s$[$LFP_RW_256/2]_SPC-2-FP,streams=$[$LFP_RW_256/2]
rd=TR18-s$[$LFP_RW_256/4]_SPC-2-FP,streams=$[$LFP_RW_256/4]
rd=TR19-s$[$LFP_RW_256/8]_SPC-2-FP,streams=$[$LFP_RW_256/8]
rd=TR20-s1_SPC-2-FP,streams=1

* LFP, \"read\" Test Phase
* Test Run Sequence 5
rd=default,rdpct=100,xfersize=1024k
rd=TR21-s"$LFP_READ_1024"_SPC-2-FP,streams=$LFP_READ_1024
rd=TR22-s$[$LFP_READ_1024/2]_SPC-2-FP,streams=$[$LFP_READ_1024/2]
rd=TR23-s$[$LFP_READ_1024/4]_SPC-2-FP,streams=$[$LFP_READ_1024/4]
rd=TR24-s$[$LFP_READ_1024/8]_SPC-2-FP,streams=$[$LFP_READ_1024/8]
rd=TR25-s1_SPC-2-FP,streams=1

* Test Run Sequence 6
rd=default,xfersize=256k
rd=TR26-s"$LFP_READ_256"_SPC-2-FP,streams=$LFP_READ_256
rd=TR27-s$[$LFP_READ_256/2]_SPC-2-FP,streams=$[$LFP_READ_256/2]
rd=TR28-s$[$LFP_READ_256/4]_SPC-2-FP,streams=$[$LFP_READ_256/4]
```

```
rd=TR29-s$[$LFP_READ_256/8]_SPC-2-FP,streams=$[$LFP_READ_256/8]
rd=TR30-s1_SPC-2-FP,streams=1
" >> $LFP

#LARGE DATABASE QUERY
echo "* Large Database Query Test (LDQ)" > $LDQ
cat $SPC2_HOSTS >> $LDQ
echo "" >> $LDQ
cat $SPC2_DISKS >> $LDQ

echo "
maxlatestart=0
reportinginterval=5
segmentlength=512m
rd=default,rdpct=99,rampup=180,periods=90,measurement=180,runout=45,rampdown=15

* LDQ, 1024 KiB Test Phase
* Test Run Sequence 1
rd=default,xfersize=1024k,buffers=4
rd=TR1-s"$LDQ_1024_4"_SPC-2-DQ,streams=$LDQ_1024_4
rd=TR2-s$[$LDQ_1024_4/2]_SPC-2-DQ,streams=$[$LDQ_1024_4/2]
rd=TR3-s$[$LDQ_1024_4/4]_SPC-2-DQ,streams=$[$LDQ_1024_4/4]
rd=TR4-s$[$LDQ_1024_4/8]_SPC-2-DQ,streams=$[$LDQ_1024_4/8]
rd=TR5-s1_SPC-2-DQ,streams=1
* Test Run Sequence 2
rd=default,buffers=1
rd=TR6-s"$LDQ_1024_1"_SPC-2-DQ,streams=$LDQ_1024_1
rd=TR7-s$[$LDQ_1024_1/2]_SPC-2-DQ,streams=$[$LDQ_1024_1/2]
rd=TR8-s$[$LDQ_1024_1/4]_SPC-2-DQ,streams=$[$LDQ_1024_1/4]
rd=TR9-s$[$LDQ_1024_1/8]_SPC-2-DQ,streams=$[$LDQ_1024_1/8]
rd=TR10-s1_SPC-2-DQ,streams=1

* LDQ, 64 KiB Test Phase
* Test Run Sequence 3
rd=default,xfersize=64k,buffers=4
rd=TR11-s"$LDQ_64_4"_SPC-2-DQ,streams=$LDQ_64_4
rd=TR12-s$[$LDQ_64_4/2]_SPC-2-DQ,streams=$[$LDQ_64_4/2]
rd=TR13-s$[$LDQ_64_4/4]_SPC-2-DQ,streams=$[$LDQ_64_4/4]
rd=TR14-s$[$LDQ_64_4/8]_SPC-2-DQ,streams=$[$LDQ_64_4/8]
rd=TR15-s1_SPC-2-DQ,streams=1
* Test Run Sequence 4
rd=default,buffers=1
rd=TR16-s"$LDQ_64_1"_SPC-2-DQ,streams=$LDQ_64_1
rd=TR17-s$[$LDQ_64_1/2]_SPC-2-DQ,streams=$[$LDQ_64_1/2]
rd=TR18-s$[$LDQ_64_1/4]_SPC-2-DQ,streams=$[$LDQ_64_1/4]
rd=TR19-s$[$LDQ_64_1/8]_SPC-2-DQ,streams=$[$LDQ_64_1/8]
rd=TR20-s1_SPC-2-DQ,streams=1
" >> $LDQ

#VIDEO ON DEMAND
echo "* Video on Demand Test (VOD)" > $VOD
cat $SPC2_HOSTS  >> $VOD
echo "" >> $VOD
cat $SPC2_DISKS >> $VOD

echo "
maxlatestart=0
videosegmentduration=1200
maxlatevod=0
reportinginterval=5
rd=default,rampup=1200,periods=600,measurement=7200,runout=45,rampdown=15,buffers=8
rd=TR1-s"$VOD_STREAMS"_SPC-2-VOD,streams=$VOD_STREAMS
" >> $VOD
```

all_killslaves

Invokes the **killslaves** script to terminate all Slave JVMs.

```
#!/bin/bash

source hosts
for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     ssh -n -f $NAME "CLASSPATH='c:\spc\spc1' /cygdrive/c/spc/run/killslaves >
/dev/null 2>/dev/null" &
done
wait
echo "Done killing all slaves"
```

killslaves

```
ps -efW | grep -e java -e spc1 | awk '{print "/bin/kill -f " $2}' | bash
```

all_runslaves

Invokes the **runslaves** script to start all the SPC-1 Slave JVMs

```
#!/bin/bash

source hosts
for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     SLAVES=${host_slave#*:}
     if [ $SLAVES -gt 0 ]; then
            ssh -n -f $NAME "/cygdrive/c/spc/run/killslaves; CLASSPATH='c:\spc\spc1'
/cygdrive/c/spc/run/runslaves $SLAVES 2> /tmp/slaves.err " >/tmp/slaves.$NAME &
     fi
done
wait
echo "Done running all slaves"
```

runslaves

```
#!/bin/bash

if [ $# -gt 0 ]; then
     NUMOFSLAVES=$1
else
     echo "Missing number of slaves to run"
     exit
fi
FILENAME=`hostname|  tr '[A-Z]' '[a-z]' | tr -d '\n' | tr -d '\r'`"_"
ROOTDIR='/cygdrive/c/spc/run/slaves/'
JAVA='/cygdrive/c/spc/run/java7 -Xmx12000m -Xms3000m'

SCRIPT_PATH=`pwd`
CNTR=0

slaves_list=`ls -l $ROOTDIR |  grep -o $FILENAME[0-9]*`

for SLAVE in ${slaves_list}; do

        if [ $CNTR -ge $NUMOFSLAVES ]; then
                exit 0
        fi
        cd $ROOTDIR$SLAVE > /dev/null
        #/cygdrive/c/Program\ Files/Java/jre7/bin/java.exe spc1 -f$SLAVE.txt && echo
"$SLAVE finished with success. Time:`date`" || echo "$SLAVE finished with rc $?.
Time: `date`"&
            $JAVA spc1 -f$SLAVE.txt && echo "$SLAVE finished with success.
Time:`date`" || echo "$SLAVE finished with rc $?. Time: `date`"&
        cd $SCRIPT_PATH > /dev/null
        CNTR=$[$CNTR + 1]
done
```

diskpart_script.bat

To overcome permission issues when connecting ssh.

```
diskpart /s C:\spc\run\diskpart_script > C:\spc\run\task.out
```

diskpart_script.xml

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.3" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>2012-03-10T01:06:47.4655841</Date>
    <Author>BO20\Administrator</Author>
  </RegistrationInfo>
  <Triggers>
    <TimeTrigger>
      <StartBoundary>2012-03-10T01:05:31.4106845</StartBoundary>
      <Enabled>true</Enabled>
    </TimeTrigger>
  </Triggers>
  <Principals>
    <Principal id="Author">
      <UserId>BO20\Administrator</UserId>
      <LogonType>Password</LogonType>
      <RunLevel>HighestAvailable</RunLevel>
    </Principal>
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <DisallowStartOnRemoteAppSession>false</DisallowStartOnRemoteAppSession>
    <UseUnifiedSchedulingEngine>false</UseUnifiedSchedulingEngine>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>P3D</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>C:\spc\run\run_diskpart_script.bat</Command>
    </Exec>
  </Actions>
```

```
</Task>
```

The following scripts are used to start the **RemoteStart** agent on the Host Systems for the SPC-2 Test Run:

- ***all_start_remote*** – start the agent on all the hosts.
- ***startremote***- start the agent on a single host.
- ***run_remotestart.xml*** – A scheduled task used to overcome ssh permission issues
- ***import_run_remotestart_task*** – import the scheduled task.

all_start_remote

```
#!/bin/bash

source hosts
for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     SLAVES=${host_slave#*:}
     if [ $NAME != "win01" -a $SLAVES -gt 0 ]; then
            #ssh -n -f $NAME "\'schtasks.exe /Run /TN run_remotestart 2>&1 \'"
>/tmp/startremote.$NAME  &
            ssh -n -f $NAME '\schtasks.exe /Run /TN run_remotestart \' &
     fi
done
echo "Done running all RemoteStart"
```

startremote

```
#!/bin/bash
java -cp 'c:\spc\spc2\' RemoteStart
```

run_remotestart.xml

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>2013-09-02T17:43:05.8856994</Date>
    <Author>WIN11\Administrator</Author>
  </RegistrationInfo>
  <Triggers />
  <Principals>
    <Principal id="Author">
      <UserId>WIN11\Administrator</UserId>
      <LogonType>Password</LogonType>
      <RunLevel>HighestAvailable</RunLevel>
    </Principal>
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>P3D</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>java</Command>
      <Arguments>-cp c:\spc\spc2\ RemoteStart</Arguments>
      <WorkingDirectory>C:\Program Files (x86)\Java\jre7\bin</WorkingDirectory>
    </Exec>
  </Actions>
</Task>
```

import_run_remotestart_task

```
schtasks.exe /delete /f /tn run_remotestart
schtasks.exe /create /xml "C:\spc\run\run_remotestart.xml" /RU
Administrator /RP Passw0rd /TN run_remotestart
```

all_prepssd

Run the ASU pre-fill on all the hosts. Each host is invoked with the config file created when running the **get_disks** script and thus pre-fill different volumes. The script waits for 10 minutes at the end to ensure all the pre-fill processes concluded. After the pre-fill is done all the parameter and output files are copied to the master server with the **all_copy_ssdprep** script.

```
#!/bin/bash

rm -rf /cygdrive/c/spc/run/ssdprep;

source hosts
for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     SLAVES=${host_slave#*:}
     if [ $SLAVES -gt 0 ]; then
            ssh -n -f $NAME
"/cygdrive/c/Users/Administrator/Desktop/vdbench503rc11/vdbench.bat -f
'C:\spc\run\prepssd.txt' -o 'C:\spc\run\ssdprep' 2> /tmp/prepssd.err "
>/tmp/prepssd.$NAME &
     fi
done
sleep 10

NUM_HOSTS=`echo $hosts_slaves | wc -w`
i="0"
while [ `tail -n 1 /tmp/prepssd.* | grep success | wc -l` -lt $NUM_HOSTS ]
do
sleep 60
i=$[$i+1]
date
echo "$i - still running prepssd."
done

echo "sleeping to make sure IO is complete"
sleep 600
wait

./all_copy_ssdprep
```

```
echo "Done running prepssd"
```

all_copy_ssdprep

```
#!/bin/bash

source hosts

mkdir -p /cygdrive/c/spc/run/ssdprep/$MASTER/ssdprep
mv  /cygdrive/c/spc/run/ssdprep/*  /cygdrive/c/spc/run/ssdprep/$MASTER/ssdprep
>/tmp/copy_ssd_prep.$MASTER &
cp -p /cygdrive/c/spc/run/prepssd.txt  /cygdrive/c/spc/run/ssdprep/$MASTER/  >>
/tmp/copy_ssd_prep.$MASTER  &

for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     SLAVES=${host_slave#*:}
     if [ $NAME != "$MASTER" -a $SLAVES -gt 0 ]; then
            mkdir /cygdrive/c/spc/run/ssdprep/$NAME
            scp -rp $NAME:/cygdrive/c/spc/run/ssdprep
/cygdrive/c/spc/run/ssdprep/$NAME  >/tmp/copy_ssd_prep.$NAME &
            scp -p $NAME:/cygdrive/c/spc/run/prepssd.txt
/cygdrive/c/spc/run/ssdprep/$NAME/  >> /tmp/copy_ssd_prep.$NAME &

     fi
done
wait
echo "Done copying ssd prep from all hosts"
```

all_run_cmd

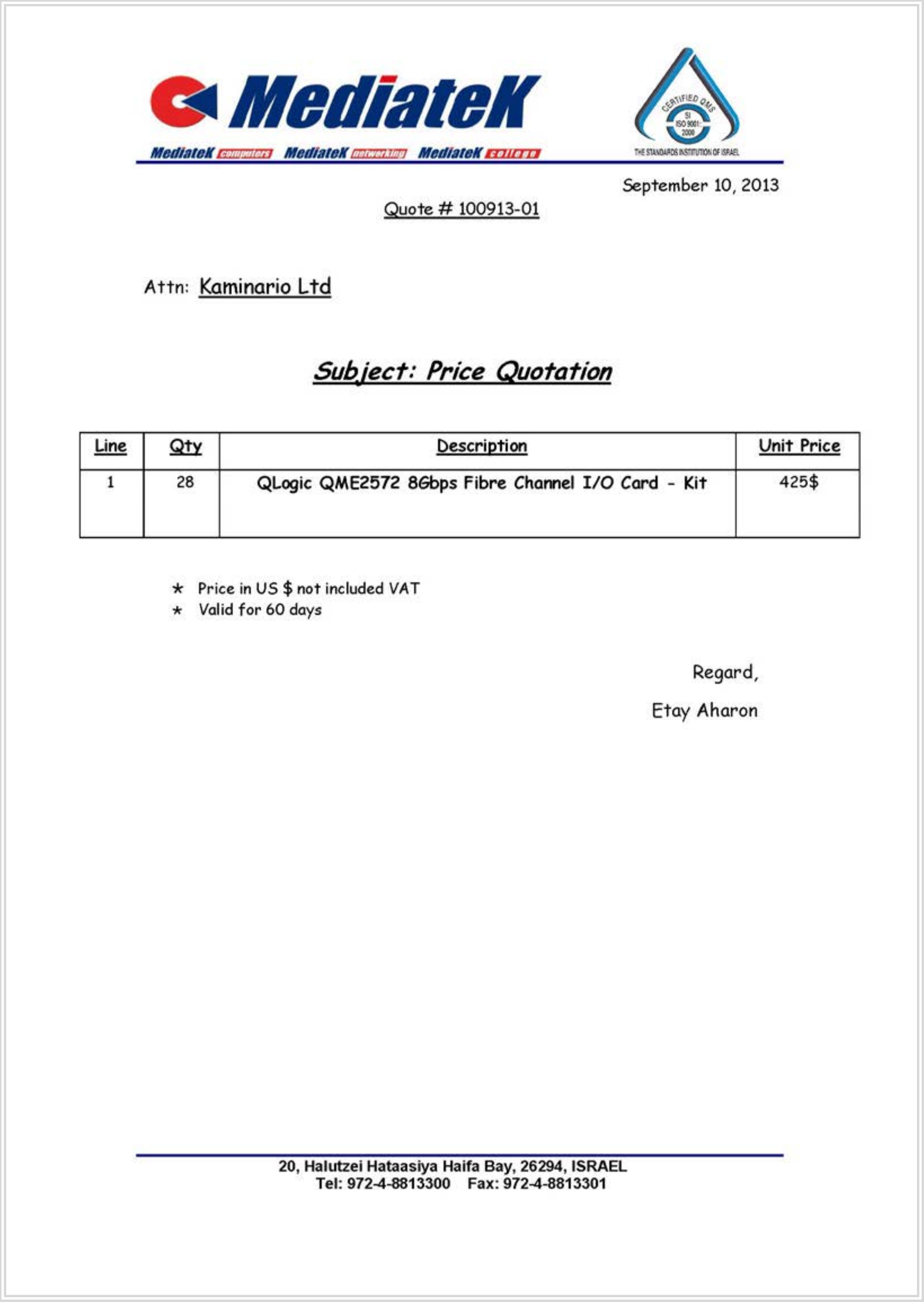
This script is used to run any command on all workload hosts. For example, move a directory or delete on as part of a cleanup process.

```
#!/bin/bash

source hosts
for host_slave in $hosts_slaves; do
     NAME=${host_slave%:*}
     SLAVES=${host_slave#*:}
     if [ $SLAVES -gt 0 ]; then
            ssh -n -f $NAME "$1 2>&1 " >/tmp/cmd.$NAME  &
            #ssh -n -f $NAME "$1 2>&1 "
     fi
done
wait
echo "Done running cmd on all hosts"
```

APPENDIX F: THIRD-PARTY QUOTATIONS

QLogic QME2572 8Gbps FC I/O Card



MediateK

MediateK computers MediateK networking MediateK college

CERTIFIED QMS SI ISO 9001:2000
THE STANDARDS INSTITUTION OF ISRAEL

September 10, 2013

Quote # 100913-01

Attn: Kaminario Ltd

Subject: Price Quotation

Line	Qty	Description	Unit Price
1	28	QLogic QME2572 8Gbps Fibre Channel I/O Card - Kit	425$

- ⋆ Price in US $ not included VAT
- ⋆ Valid for 60 days

Regard,

Etay Aharon

20, Halutzei Hataasiya Haifa Bay, 26294, ISRAEL
Tel: 972-4-8813300 Fax: 972-4-8813301