# SPC Benchmark 1™

# Full Disclosure Report

## MacroSAN Technologies Co., Ltd
## MacroSAN MS5580G2

## SPC-1 V3.8.0

**Submission Identifier: A31022**

**Submitted For Review: July 16, 2019**

## First Edition – July 2019

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 MacroSAN 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 People's Republic of China. MacroSAN 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 MacroSAN representative for information on products and services available in your area.

## Trademarks

SPC Benchmark 1, SPC-1, SPC-1 IOPS, SPC-1 LRT and SPC-1 Price-Performance are trademarks of the Storage Performance Council.

MacroSAN, the MacroSAN logo and MS5580G2 are trademarks or registered trademarks of MacroSAN in the People's Republic of China and other countries. All other brands, trademarks, and product names are the property of their respective owners.

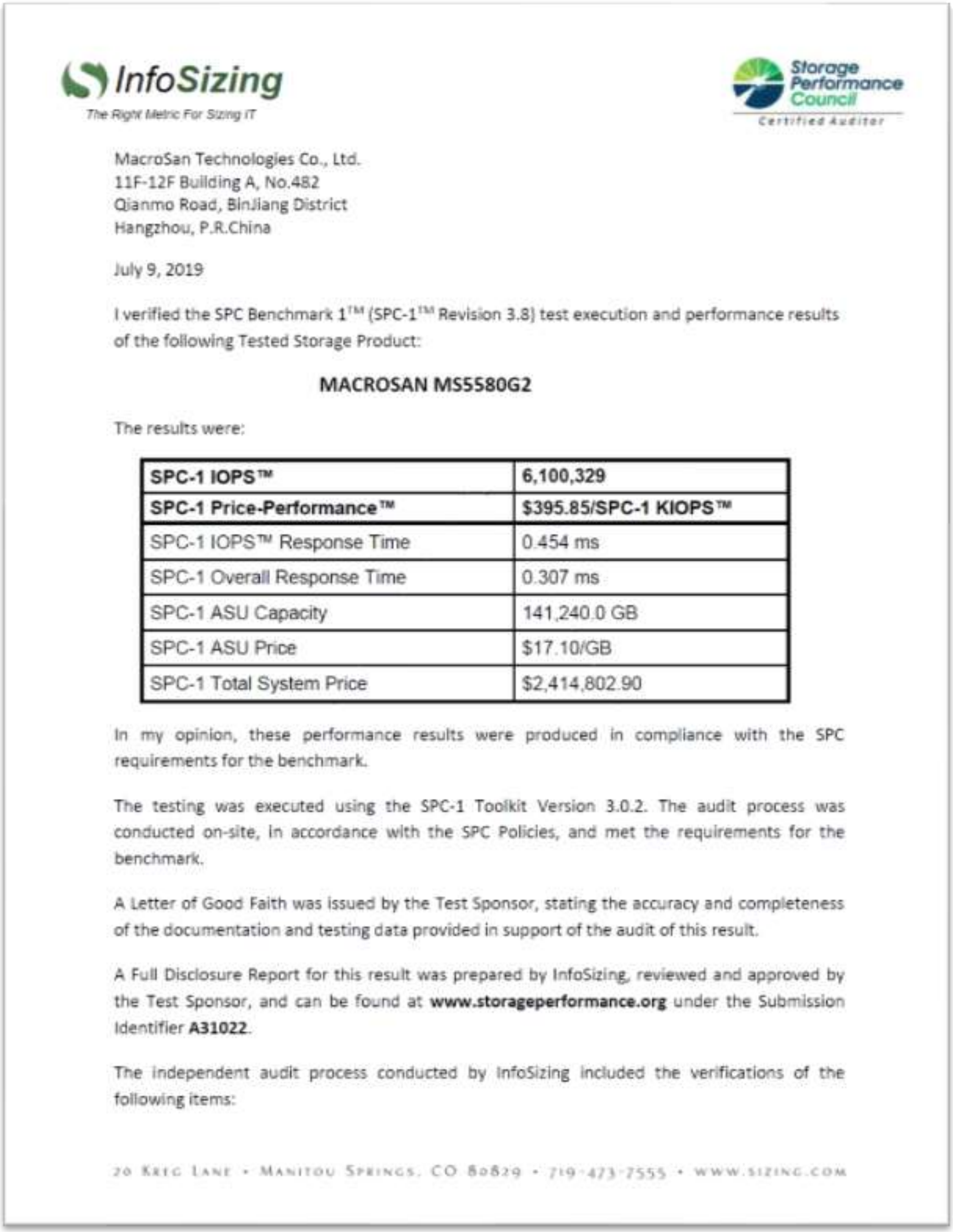
## Benchmark Specification and Glossary

The official SPC Benchmark 1™ (SPC-1™) specification is available on the website of the Storage Performance Council (SPC) at www.spcresults.org.

The SPC-1™ specification contains a glossary of the SPC-1™ terms used in this publication.

# Table of Contents

# AUDIT CERTIFICATION



InfoSizing
The Right Metric For Sizing IT

Storage Performance Council
Certified Auditor

MacroSan Technologies Co., Ltd.
11F-12F Building A, No.482
Qianmo Road, BinJiang District
Hangzhou, P.R.China

July 9, 2019

I verified the SPC Benchmark 1™ (SPC-1™ Revision 3.8) test execution and performance results of the following Tested Storage Product:

**MACROSAN MS5580G2**

The results were:

| | |
|---|---|
| **SPC-1 IOPS™** | **6,100,329** |
| **SPC-1 Price-Performance™** | **$395.85/SPC-1 KIOPS™** |
| SPC-1 IOPS™ Response Time | 0.454 ms |
| SPC-1 Overall Response Time | 0.307 ms |
| SPC-1 ASU Capacity | 141,240.0 GB |
| SPC-1 ASU Price | $17.10/GB |
| SPC-1 Total System Price | $2,414,802.90 |

In my opinion, these performance results were produced in compliance with the SPC requirements for the benchmark.

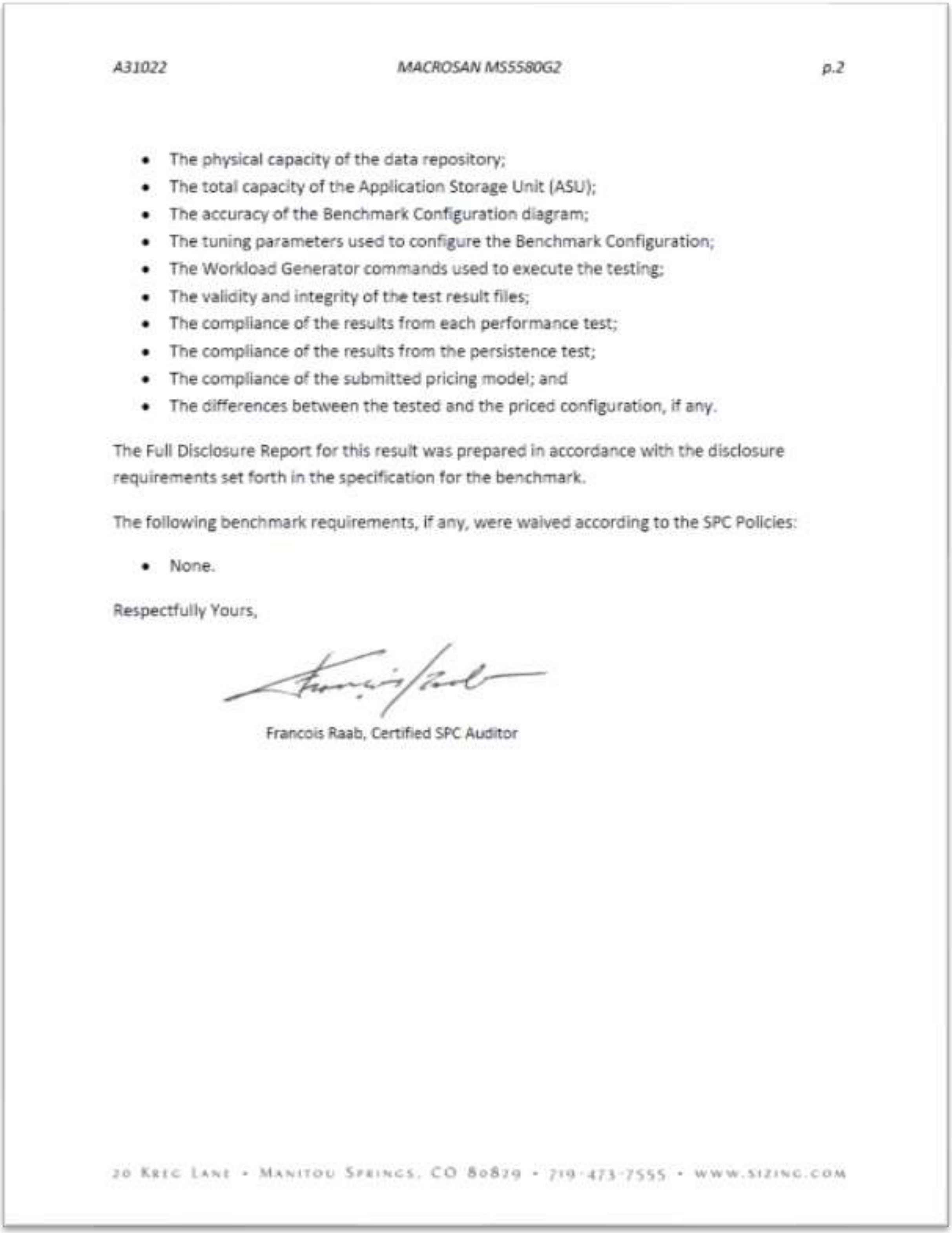
The testing was executed using the SPC-1 Toolkit Version 3.0.2. The audit process was conducted on-site, in accordance with the SPC Policies, and met the requirements for the benchmark.

A Letter of Good Faith was issued by the Test Sponsor, stating the accuracy and completeness of the documentation and testing data provided in support of the audit of this result.

A Full Disclosure Report for this result was prepared by InfoSizing, reviewed and approved by the Test Sponsor, and can be found at **www.storageperformance.org** under the Submission Identifier **A31022**.

The independent audit process conducted by InfoSizing included the verifications of the following items:

20 Kreg Lane • Manitou Springs, CO 80829 • 719-473-7555 • www.sizing.com



- The physical capacity of the data repository;
- The total capacity of the Application Storage Unit (ASU);
- The accuracy of the Benchmark Configuration diagram;
- The tuning parameters used to configure the Benchmark Configuration;
- The Workload Generator commands used to execute the testing;
- The validity and integrity of the test result files;
- The compliance of the results from each performance test;
- The compliance of the results from the persistence test;
- The compliance of the submitted pricing model; and
- The differences between the tested and the priced configuration, if any.

The Full Disclosure Report for this result was prepared in accordance with the disclosure requirements set forth in the specification for the benchmark.

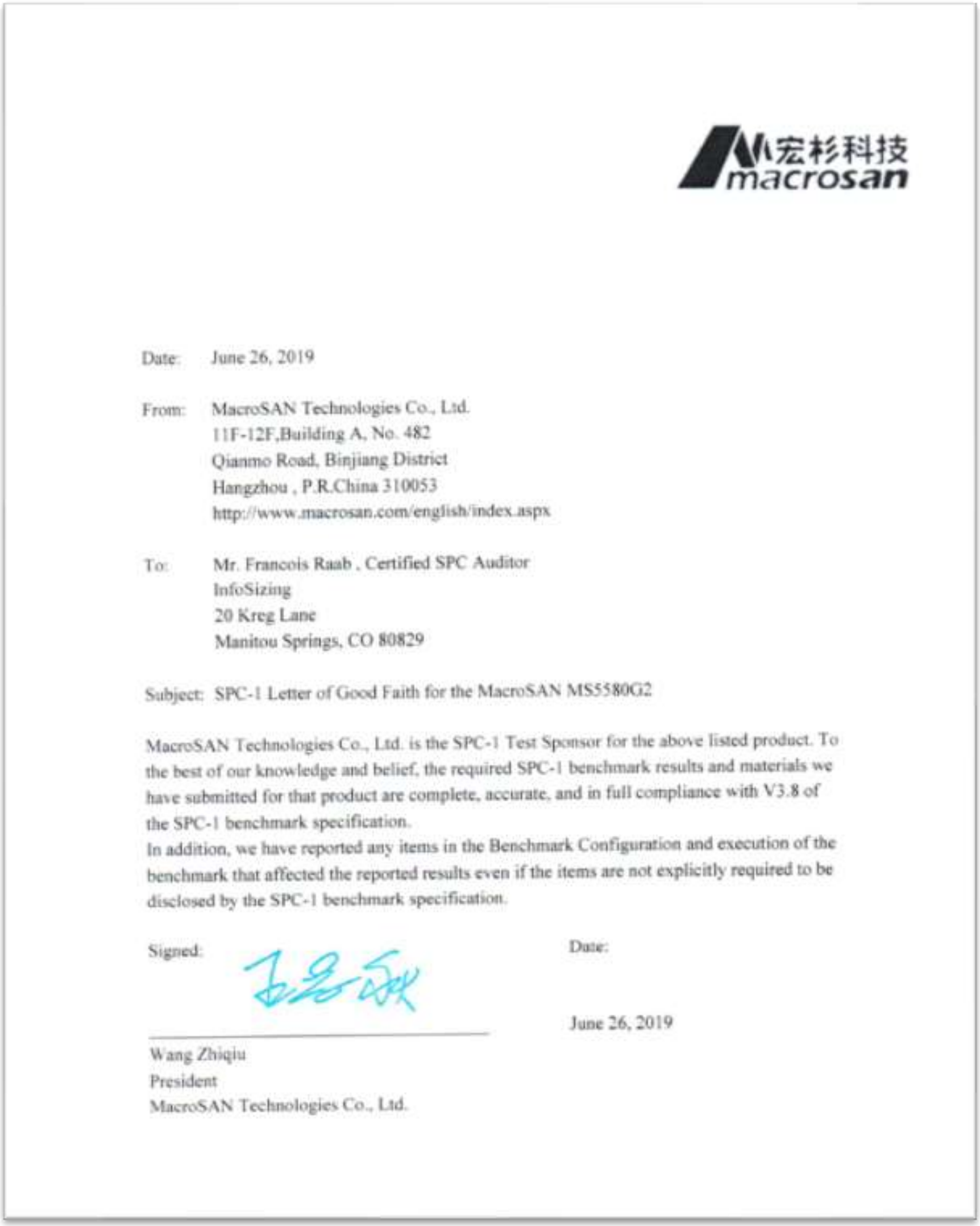
The following benchmark requirements, if any, were waived according to the SPC Policies:

- None.

Respectfully Yours,

Francois Raab, Certified SPC Auditor

# LETTER OF GOOD FAITH



宏杉科技
macrosan

Date: June 26, 2019

From: MacroSAN Technologies Co., Ltd.
11F-12F,Building A, No. 482
Qianmo Road, Binjiang District
Hangzhou , P.R.China 310053
http://www.macrosan.com/english/index.aspx

To: Mr. Francois Raab , Certified SPC Auditor
InfoSizing
20 Kreg Lane
Manitou Springs, CO 80829

Subject: SPC-1 Letter of Good Faith for the MacroSAN MS5580G2

MacroSAN Technologies Co., Ltd. is the SPC-1 Test Sponsor for the above listed product. To the best of our knowledge and belief, the required SPC-1 benchmark results and materials we have submitted for that product are complete, accurate, and in full compliance with V3.8 of the SPC-1 benchmark specification.
In addition, we have reported any items in the Benchmark Configuration and execution of the benchmark that affected the reported results even if the items are not explicitly required to be disclosed by the SPC-1 benchmark specification.

Signed:

Wang Zhiqiu
President
MacroSAN Technologies Co., Ltd.

Date:

June 26, 2019

# SPC Benchmark 1™

# Executive Summary

## MacroSAN Technologies Co., Ltd
## MacroSAN MS5580G2

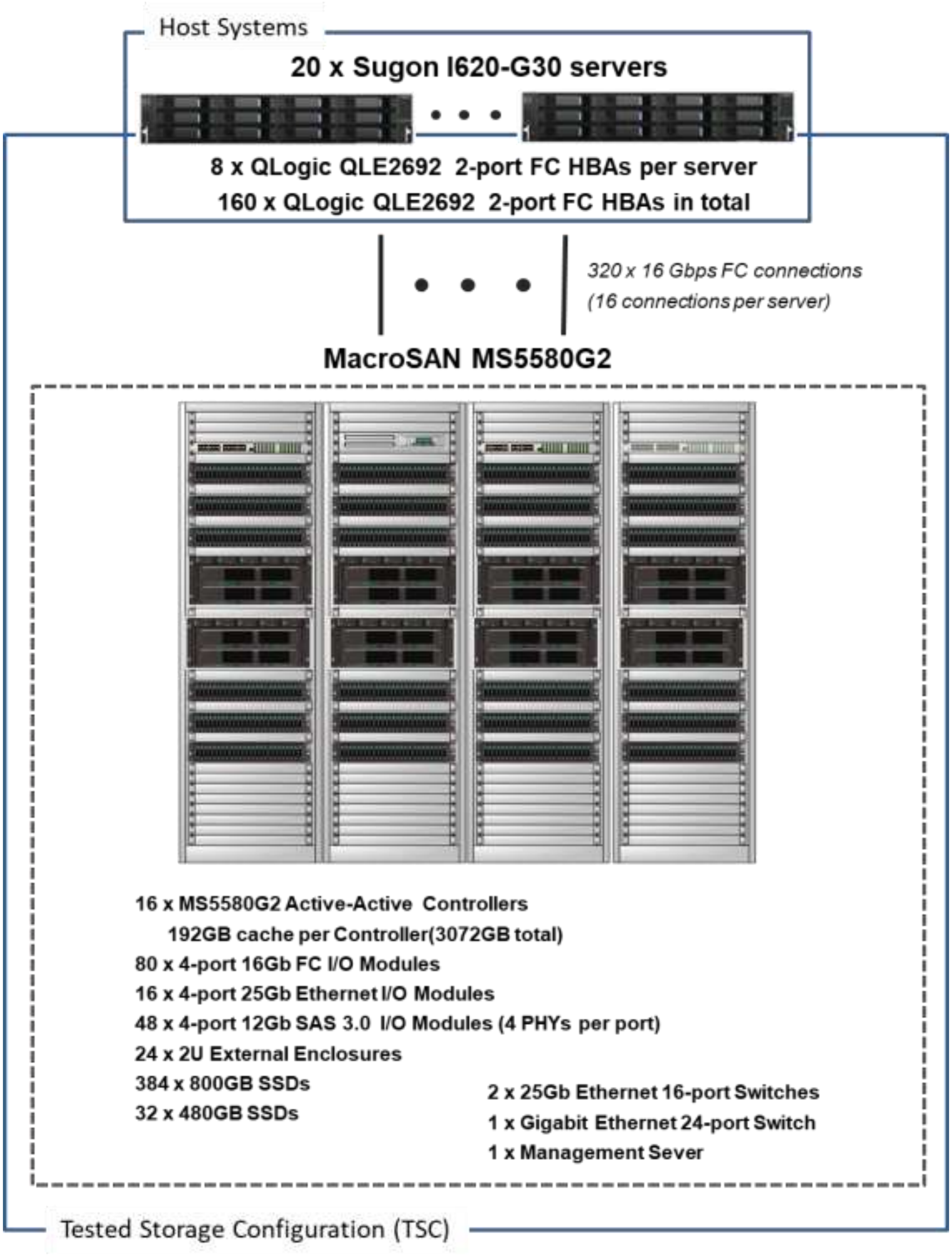
| | |
|---|---|
| **SPC-1 IOPS™** | **6,100,329** |
| **SPC-1 Price-Performance™** | **$395.85/SPC-1 KIOPS™** |
| SPC-1 IOPS™ Response Time | 0.454 ms |
| SPC-1 Overall Response Time | 0.307 ms |
| SPC-1 ASU Capacity | 141,240.0 GB |
| SPC-1 ASU Price | $17.10/GB |
| SPC-1 Total System Price | $2,414,802.90 |
| Data Protection Level | Protected 2 (RAID-10 and full redundancy) |
| Physical Storage Capacity | 299,968 GB |
| Pricing Currency / Target Country | U.S. Dollars / People's Republic of China |

**SPC-1 V3.8.0**

**Submission Identifier: A31022**

**Submitted For Review: July 16, 2019**

## Benchmark Configuration Diagram


Host Systems
20 x Sugon I620-G30 servers
8 x QLogic QLE2692 2-port FC HBAs per server
160 x QLogic QLE2692 2-port FC HBAs in total
320 x 16 Gbps FC connections
(16 connections per server)
MacroSAN MS5580G2
16 x MS5580G2 Active-Active Controllers
192GB cache per Controller(3072GB total)
80 x 4-port 16Gb FC I/O Modules
16 x 4-port 25Gb Ethernet I/O Modules
48 x 4-port 12Gb SAS 3.0 I/O Modules (4 PHYs per port)
24 x 2U External Enclosures
384 x 800GB SSDs
32 x 480GB SSDs
2 x 25Gb Ethernet 16-port Switches
1 x Gigabit Ethernet 24-port Switch
1 x Management Sever
Tested Storage Configuration (TSC)

## Tested Storage Product Description

MacroSAN MS5580G2 is a storage product which is aimed at large and medium size data centers. It integrates many advanced design concept and architecture technology to provide a safe, reliable and flexible storage platform. The MS5580G2 supports a maximum of 32 controllers with a maximum of 1,536GB cache per dual controllers.

The MS5580G2 can guarantee data safety by symmetric dual-active, replication, mirror image and snapshots. Furthermore, based on open data storage platform (ODSP), the MS5580G2 can realize interface development, customized function development and function migration. The MS5580G2 delivers leadership levels of reliability, functionality and manageability.

For more details, visit:

http://www.macrosan.com/en/product_detail.aspx?k1=2&k2=10&id=207

## Priced Storage Configuration Components

| 160 x QLogic QLE2692 2-port FC HBAs |
|---|
| 16 x MS5580G2 Active-Active Controllers, each with:<br>192GB cache (3072GB total)<br>80 x 4-port 16Gb FC I/O modules<br>16 x 4-port 25Gb Ethernet I/O modules<br>48 x 4-port 12Gb SAS 3.0 I/O modules (4 PHYs per port)<br>8 x 2U External Enclosures, each with:<br>4 x 480GB SSDs<br>16 x 800GB SSDs<br>16 x 2U External Enclosures, each with:<br>16 x 800GB SSDs<br>2 x 25Gb Ethernet 16-port switches<br>1 x Gigabit Ethernet 24-port switch<br>1 x Management Sever |

# Storage Configuration Pricing

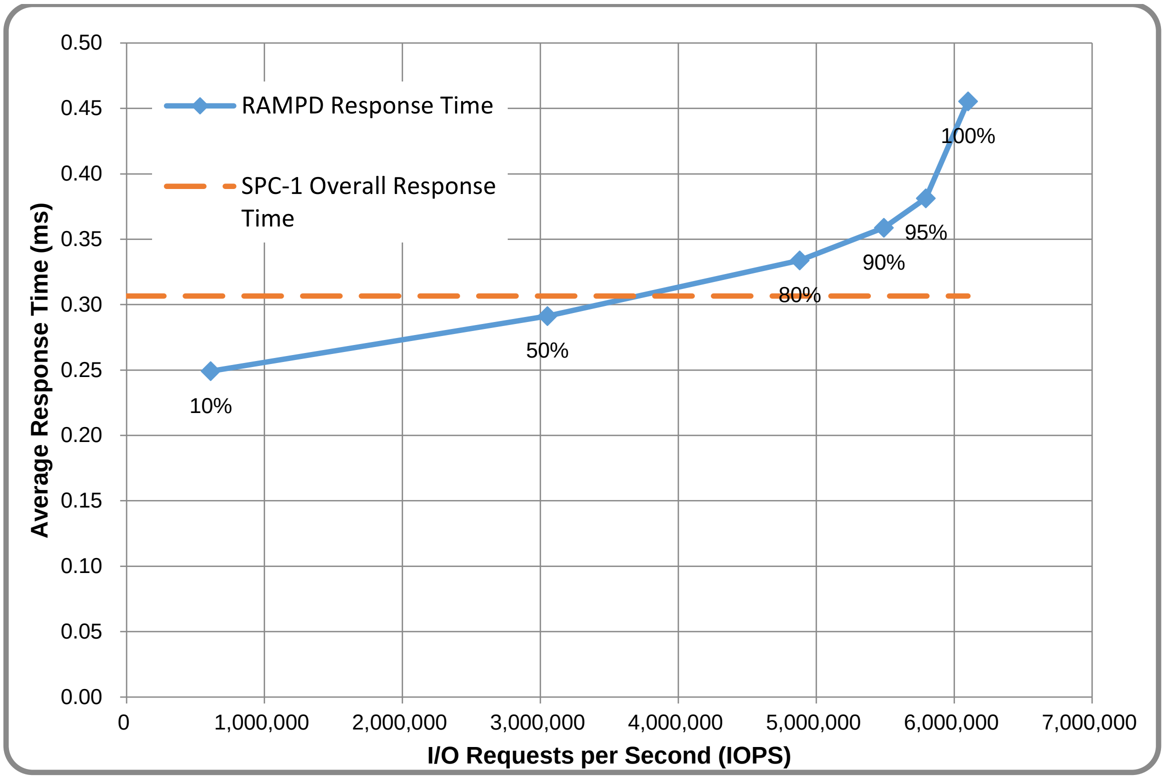
| | Description | Qty | Unit Price | Ext. Price | Disc. | Disc. Price |
|---|---|---|---|---|---|---|
| | **Hardware & Software** | | | | | |
| MS5580G2-384GB | MS5580G2 Storage Controllers Unit (Dual Controllers, 5U, 384GB cache, 3Y 7x9xND Basic Svc&Warranty) | 8 | 77,826.09 | 622,608.70 | 50% | 311,304.35 |
| SSU2225B | SSU2225B SAS Switch Unit (Support 25 2.5 inch Disks, SAS 3.0) | 8 | 6,130.43 | 49,043.48 | 50% | 24,521.74 |
| SSU2225 | SSU2225 SAS Switch Unit (Support 25 2.5 inch Disks, SAS 3.0) | 16 | 5,217.39 | 83,478.26 | 50% | 41,739.13 |
| MS2M1SD480G0M2AEB | 480GB MLC SSD Drive(2.5") for SSU2225B | 32 | 1,623.19 | 51,942.03 | 50% | 25,971.01 |
| MS2M2SD800GAM2AEA | 800GB MLC SSD Drive(2.5") for SSU2225 | 384 | 7,826.09 | 3,005,217.39 | 50% | 1,502,608.70 |
| IOA6014A | 4*48Gb SAS 3.0 I/O Module | 48 | 1,507.25 | 72,347.83 | 50% | 36,173.91 |
| IOA4014A | 4*16Gb FC IO Module | 80 | 3,478.26 | 278,260.87 | 50% | 139,130.43 |
| IOAC014A | 4*25Gb Ethernet I/O Module | 16 | 1,855.07 | 29,681.16 | 50% | 14,840.58 |
| LIS_MS5580_BASE_STD | MacroSAN ODSP for Multi Control - MS5580G2 Storage Management Software (Basic Storage Management, CRAID, support FC&iSCSI, System Monitoring and Warning)-English Version | 1 | 9,275.36 | 9,275.36 | 50% | 4,637.68 |
| LIS_MS5580_2C_EXPAND | MacroSAN ODSP-MS5580G2 Storage System Management Software - 2 Controllers EXPAND | 7 | 9,971.01 | 69,797.10 | 50% | 34,898.55 |
| LC10-LC10-MI2-L5 | 10M Fiber Cable (Multimode, LC-LC) | 320 | 20.29 | 6,492.75 | 50% | 3,246.38 |
| 05020024 | 2*25Gb Multimode Optical Module QSFP+, AOC Fiber Cable-5M | 32 | 753.62 | 24,115.94 | 50% | 12,057.97 |
| 05020012 | SAS Cable (1M, miniSAS HD-miniSAS HD) | 192 | 0.00 | 0.00 | 0% | 0.00 |
| Qlogic QLE2692 | Qlogic QLE2692 HBA Card, PCIE, 16Gbps 4-Ports, Fiber Channel<br>Multimode LC Optic Interface | 160 | 2,347.83 | 375,652.17 | 50% | 187,826.09 |
| Management Suite | Management Suite (1Gb Ethernet 24-port Switch*1, Management Sever*1, Network Cable 1Gbps-3M*17) | 1 | 5,362.32 | 5,362.32 | 50% | 2,681.16 |
| Ethernet Switch | 25Gb Ethernet 16-Port Switch | 2 | 7,669.57 | 15,339.13 | 50% | 7,669.57 |
| | | | | **Hardware & Software Subtotal** | | **2,349,307.25** |
| | **Support & Maintenance** | | | | | |
| MS-5000-SV-BS-SPU | MS5580G2 Installation Service - Per Node | 8 | 1,793.48 | 14,347.83 | 20% | 11,478.26 |
| MS-5000-SV-BS-DSU | MS5580G2 Installation Service - Per SSU | 24 | 543.48 | 13,043.48 | 20% | 10,434.78 |
| MS-5000-SV-SS-1Y-SPU | MS5580G2 Svc&Warranty Upgrade - 7x24x4 Premium - Per Year Per Node | 8 | 3,043.48 | 24,347.83 | 20% | 19,478.26 |
| MS-5000-SV-SS-1Y-DSU | MS5580G2 Svc&Warranty Upgrade - 7x24x4 Premium - Per Year Per SSU | 24 | 1,255.43 | 30,130.43 | 20% | 24,104.35 |
| | | | | **Support & Maintenance Subtotal** | | **65,495.65** |
| | | | | | | |
| | **SPC-1 Total System Price** | | | | | **2,414,802.90** |
| | SPC-1 IOPS™ | | | | | 6,100,329 |
| | **SPC-1 Price-Performance™ ($/SPC-1 KIOPS™)** | | | | | **395.85** |
| | SPC-1 ASU Capacity (GB) | | | | | 141,240 |
| | **SPC-1 ASU Price ($/GB)** | | | | | **17.10** |

**Discount Details**: The discounts shown are generally available, and based on the capacity and total price of the storage configuration purchased.

**Warranty**: Pricing includes Gold-Level Service with: 24x7 online support, unlimited software upgrades and bug fixes, and on-site presence of a qualified maintenance engineer within 4 hours of a problem acknowledgement, inside the Target Market.

**Availability Date**: Currently available.

## Response Time and Throughput Graph



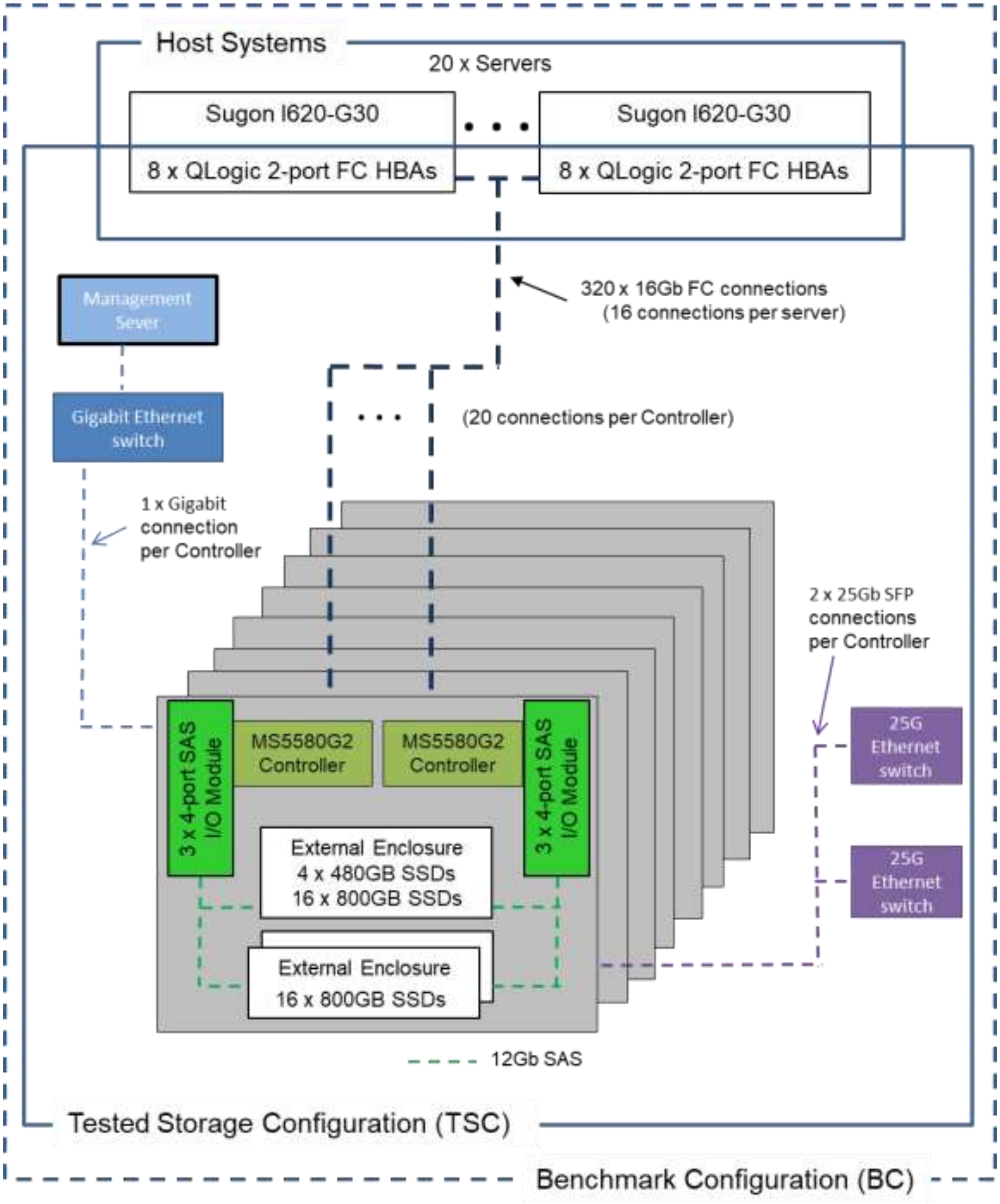
| Contact Information | |
|---|---|
| **Test Sponsor Primary Contact** | MacroSAN Technologies Co ., Ltd.– http://www.macrosan.com/<br>Yi Shen – shenyi@macrosan.com |
| **SPC Auditor** | InfoSizing – www.sizing.com<br>Francois Raab – francois@sizing.com |

| Revision Information | |
|---|---|
| **SPC Benchmark 1™ Revision** | V3.8.0 |
| **SPC-1 Workload Generator Revision** | V3.0.2 |
| **Publication Revision History** | None |

# CONFIGURATION INFORMATION

## Benchmark Configuration and Tested Storage Configuration

The following diagram illustrates the Benchmark Configuration (BC), including the Tested Storage Configuration (TSC) and the Host System(s).


Host Systems
20 x Servers
Sugon I620-G30
8 x QLogic 2-port FC HBAs
Sugon I620-G30
8 x QLogic 2-port FC HBAs
320 x 16Gb FC connections
(16 connections per server)
Management Sever
Gigabit Ethernet switch
(20 connections per Controller)
1 x Gigabit connection per Controller
2 x 25Gb SFP connections per Controller
3 x 4-port SAS I/O Module
MS5580G2 Controller
MS5580G2 Controller
3 x 4-port SAS I/O Module
25G Ethernet switch
25G Ethernet switch
External Enclosure
4 x 480GB SSDs
16 x 800GB SSDs
External Enclosure
16 x 800GB SSDs
12Gb SAS
Tested Storage Configuration (TSC)
Benchmark Configuration (BC)

## Storage Network Configuration

The Tested Storage Configuration (TSC) involved an external storage subsystem made of 8 MacroSAN MS5580G2 Nodes, each with two Storage Controllers. They were driven by 20 host systems (Sugon I620-G30). Each host connected one-to-one to each Storage Controller. Each one-to-one connection was established using a port from one of the eight dual-port HBAs on each host; and a port from each of the five 4-port I/O Modules on each Storage Controller. These were 16Gbps Fibre Chanel paths.

## Host System and Tested Storage Configuration Components

The following table lists the components of the Host System(s) and the Tested Storage Configuration (TSC).

| Host Systems |
|---|
| 20 x Sugon I620-G30, each with:<br>2 x Intel Xeon Silver 4116 (2.1 GHz, 12-Core, 16.5 MB L3)<br>192GB Main Memory<br>CentOS Linux release 7.4.1708 |
| **Priced Storage Configuration** |
| 160 x QLogic QLE2692 2-port FC HBAs |
| 16 x MS5580G2 Active-Active Controllers, each with:<br>192GB cache (3072GB total)<br>80 x 4-port 16Gb FC I/O modules<br>16 x 4-port 25Gb Ethernet I/O modules<br>48 x 4-port 12Gb SAS 3.0 I/O modules (4 PHYs per port)<br>8 x 2U External Enclosures, each with:<br>4 x 480GB SSDs<br>16 x 800GB SSDs<br>16 x 2U External Enclosures, each with:<br>16 x 800GB SSDs<br>2 x 25Gb Ethernet 16-port switches<br>1 x Gigabit Ethernet 24-port switch<br>1 x Management Sever |

## Differences Between Tested and Priced Storage Configurations

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

## Component Changes in Revised Full Disclosure Report

The following table outlines component changes that were made in revisions to this Full Disclosure Report.

| Original Component | Revised Component | Description of Change |
|---|---|---|
| n/a | n/a | Initial submission |

# Benchmark Configuration Creation Process

## Customer Tuning Parameters and Options

All the customer tuning parameters and options that have been altered from their default values for this benchmark are included in Appendix C and in the Supporting Files (see Appendix A).

## Tested Storage Configuration Creation

A detailed description of how the logical representation of the TSC was created is included in Appendix D and in the Supporting Files (see Appendix A).

## Tested Storage Configuration Inventory

An inventory of the components in the TSC, as seen by the Benchmark Configuration, is included in Appendix E and in the Supporting Files (see Appendix A).

## Workload Generator Storage Configuration

The SPC-1 Workload Generator storage configuration commands and parameters used to invoke the execution of the tests are included in Appendix F and in the Supporting Files (see Appendix A).

## Logical Volume Capacity and ASU Mapping

The following table details the capacity of each ASU and how they are mapped to logical volumes (LV).

| | LV per ASU | LV Capacity | Used per LV | Total per ASU | % ASU Capacity |
|---|---|---|---|---|---|
| **ASU-1** | 3 | 21,186.0 | 21,186.0 | 63,558.0 | 45.00% |
| **ASU-2** | 3 | 21,186.0 | 21,186.0 | 63,558.0 | 45.00% |
| **ASU-3** | 3 | 4,708.0 | 4,708.0 | 14,124.0 | 10.00% |
| | | | **SPC-1 ASU Capacity** | **141,240.0** | |

## Physical Storage Capacity and Utilization

The following table details the Physical Capacity of the storage devices and the Physical Capacity Utilization (percentage of Total Physical Capacity used) in support of hosting the ASUs.

| Devices | Count | Physical Capacity | Total Capacity |
|---|---|---|---|
| 480GB SSD | 32 | 446.0 | 14,272.0 |
| 800GB SSD | 384 | 744.0 | 285,696.0 |
| | | **Total Physical Capacity** | **299,968.0** |
| | | **Physical Capacity Utilization** | **47.09%** |

## Data Protection

The data protection level used for all logical volumes was **Protected 2**, which was accomplished by configuring 8 pools, each 48 drives, into 64 RAID 10 arrays. All components and access paths from the Host Systems to the Storage Devices were redundant.

# BENCHMARK EXECUTION RESULTS

This portion of the Full Disclosure Report documents the results of the various SPC-1 Tests, Test Phases, and Test Runs.

## Benchmark Execution Overview

### Workload Generator Input Parameters

The SPC-1 Workload Generator commands and input parameters for the Test Phases are presented in the Supporting Files (see Appendix A).

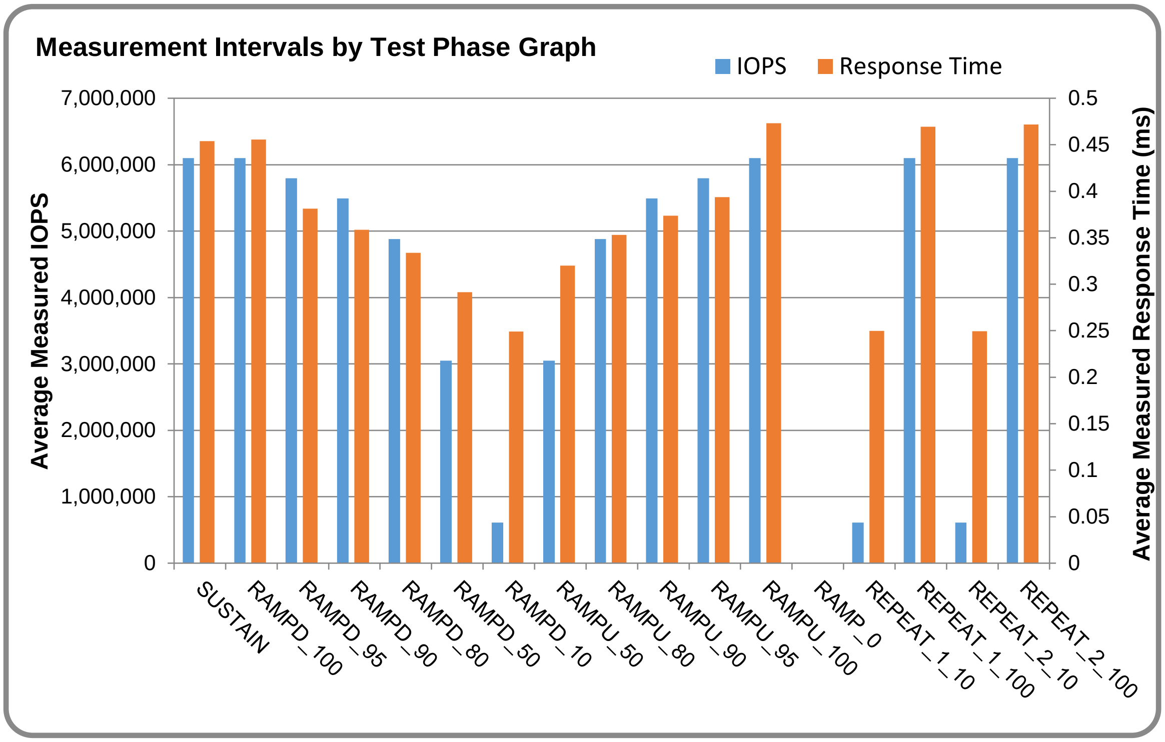
### Primary Metrics Test Phases

The benchmark execution consists of the Primary Metrics Test Phases, including the Test Phases SUSTAIN, RAMPD_100 to RAMPD_10, RAMPU_50 to RAMPU_100, RAMP_0, REPEAT_1 and REPEAT_2.

Each Test Phase starts with a transition period followed by a Measurement Interval.

### Measurement Intervals by Test Phase Graph

The following graph presents the average IOPS and the average Response Times measured over the Measurement Interval (MI) of each Test Phase.



### Exception and Waiver

None.

# SUSTAIN Test Phase

## SUSTAIN – Results File

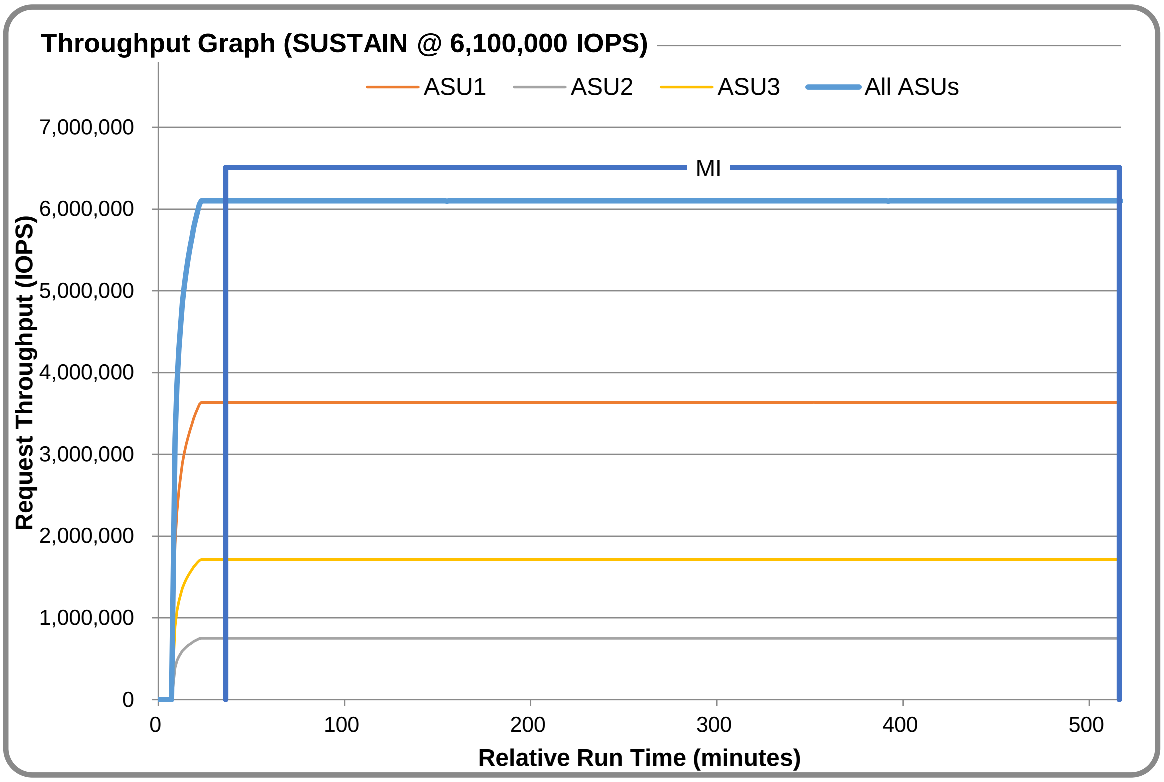
The results file generated during the execution of the SUSTAIN Test Phase is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

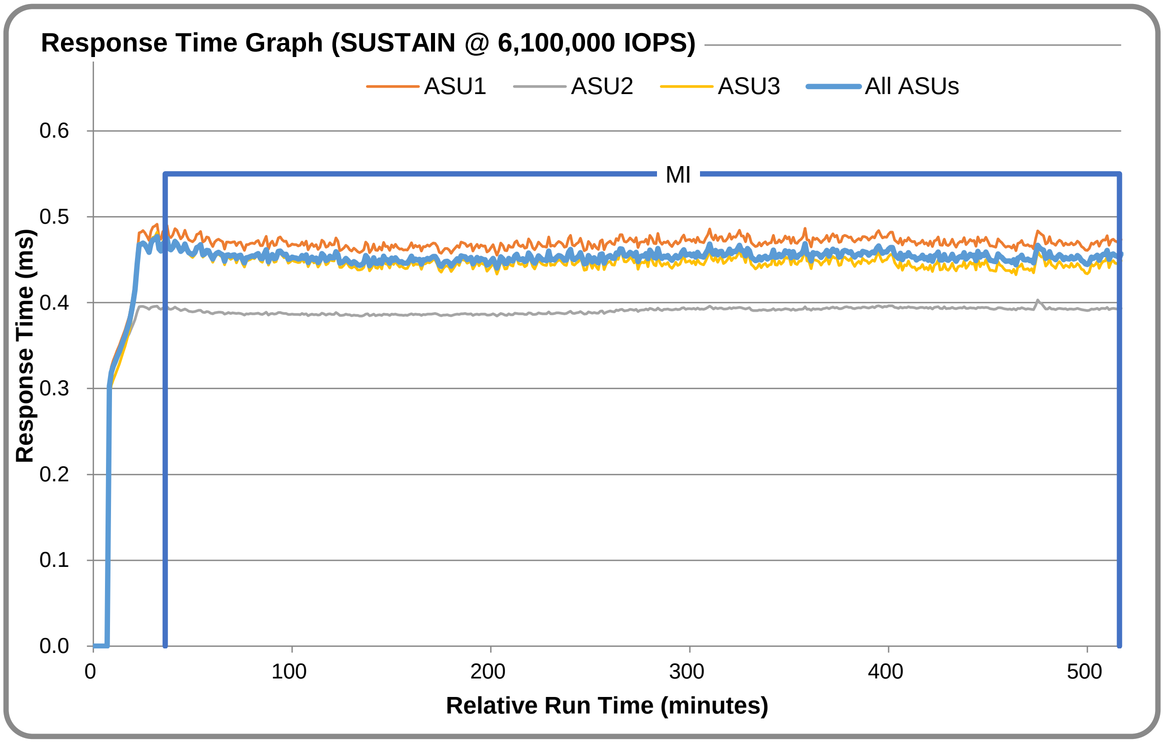
## SUSTAIN – Execution Times

| Interval | Start Date & Time | End Date &Time | Duration |
|---|---|---|---|
| Transition Period | 24-Jun-19 19:14:00 | 24-Jun-19 19:43:58 | 0:29:57 |
| Measurement Interval | 24-Jun-19 19:43:58 | 25-Jun-19 03:43:59 | 8:00:01 |

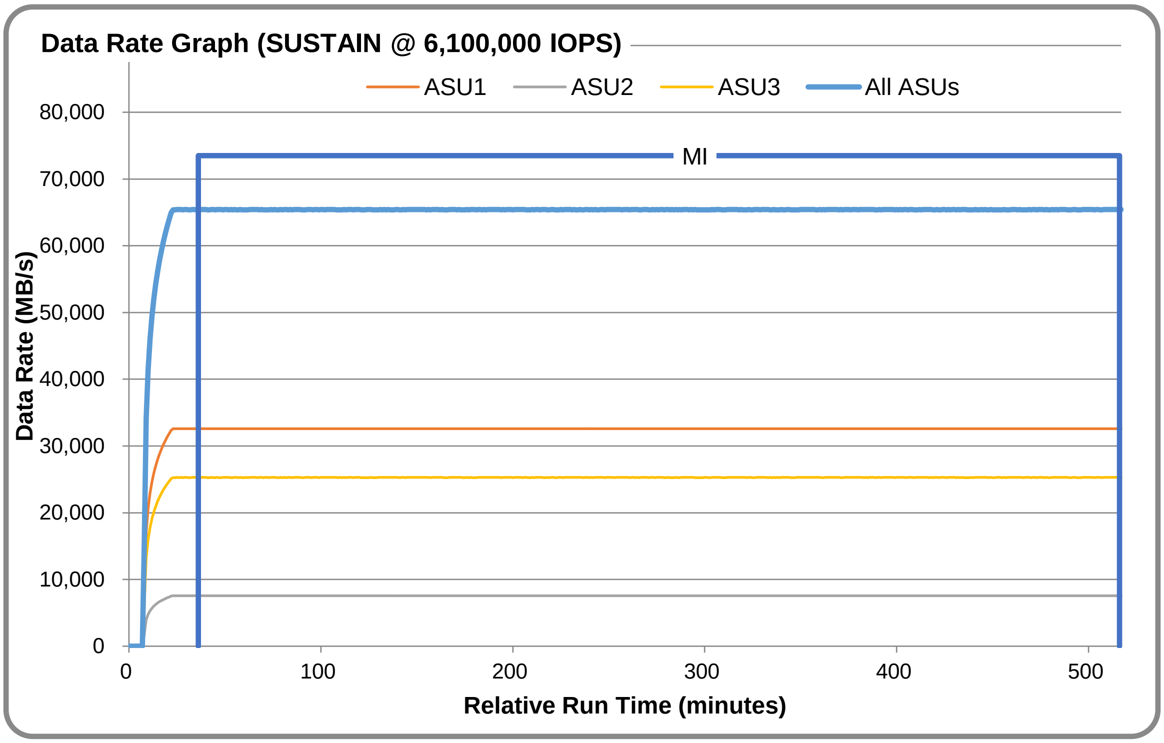
## SUSTAIN – Throughput Graph



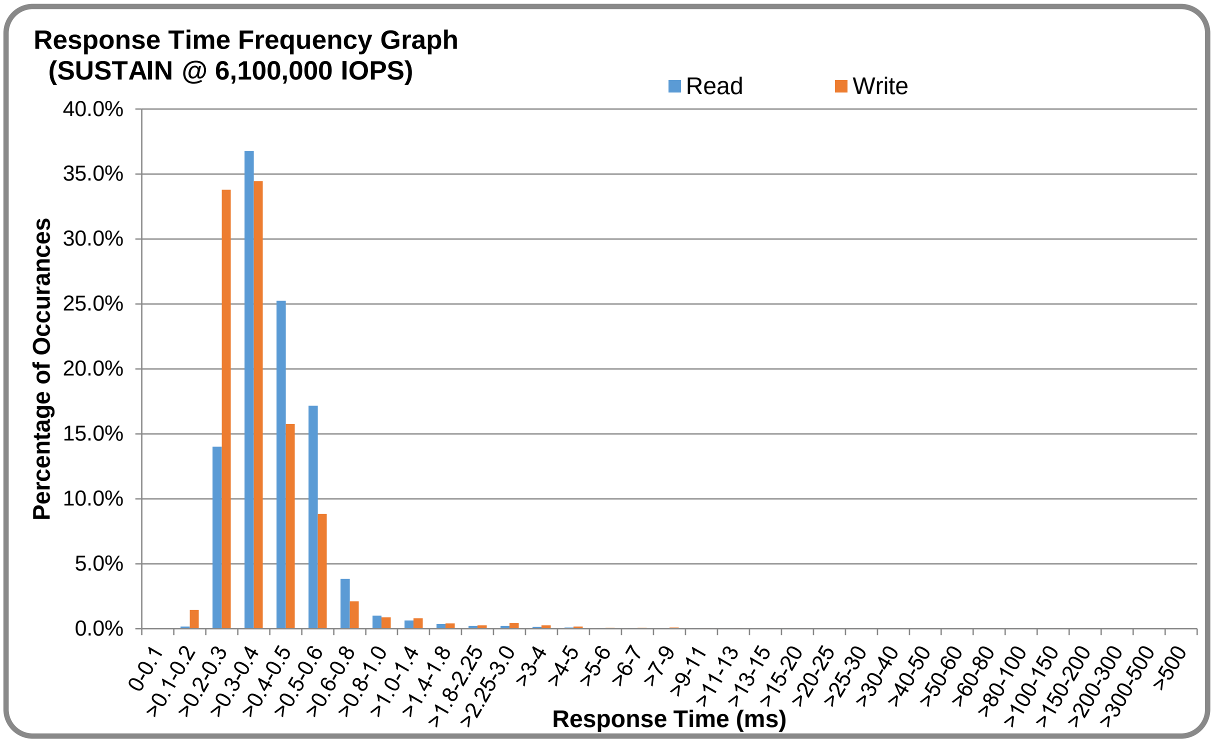
## SUSTAIN – Response Time Graph



## SUSTAIN – Data Rate Graph



## SUSTAIN – Response Time Frequency Graph



## SUSTAIN – Intensity Multiplier

The following table lists the targeted intensity multiplier (Defined), the measured intensity multiplier (Measured) for each I/O STREAM, its coefficient of variation (Variation) and the percentage of difference (Difference) between Target and Measured.

| | ASU1-1 | ASU1-2 | ASU1-3 | ASU1-4 | ASU2-1 | ASU2-2 | ASU2-3 | ASU3-1 |
|---|---|---|---|---|---|---|---|---|
| Defined | 0.0350 | 0.2810 | 0.0700 | 0.2100 | 0.0180 | 0.0700 | 0.0350 | 0.2810 |
| Measured | 0.0350 | 0.2810 | 0.0700 | 0.2100 | 0.0180 | 0.0700 | 0.0350 | 0.2810 |
| Variation | 0.0003 | 0.0001 | 0.0002 | 0.0001 | 0.0004 | 0.0002 | 0.0003 | 0.0001 |
| Difference | 0.004% | 0.001% | 0.006% | 0.000% | 0.004% | 0.002% | 0.004% | 0.002% |

# RAMPD_100 Test Phase

## RAMPD_100 – Results File

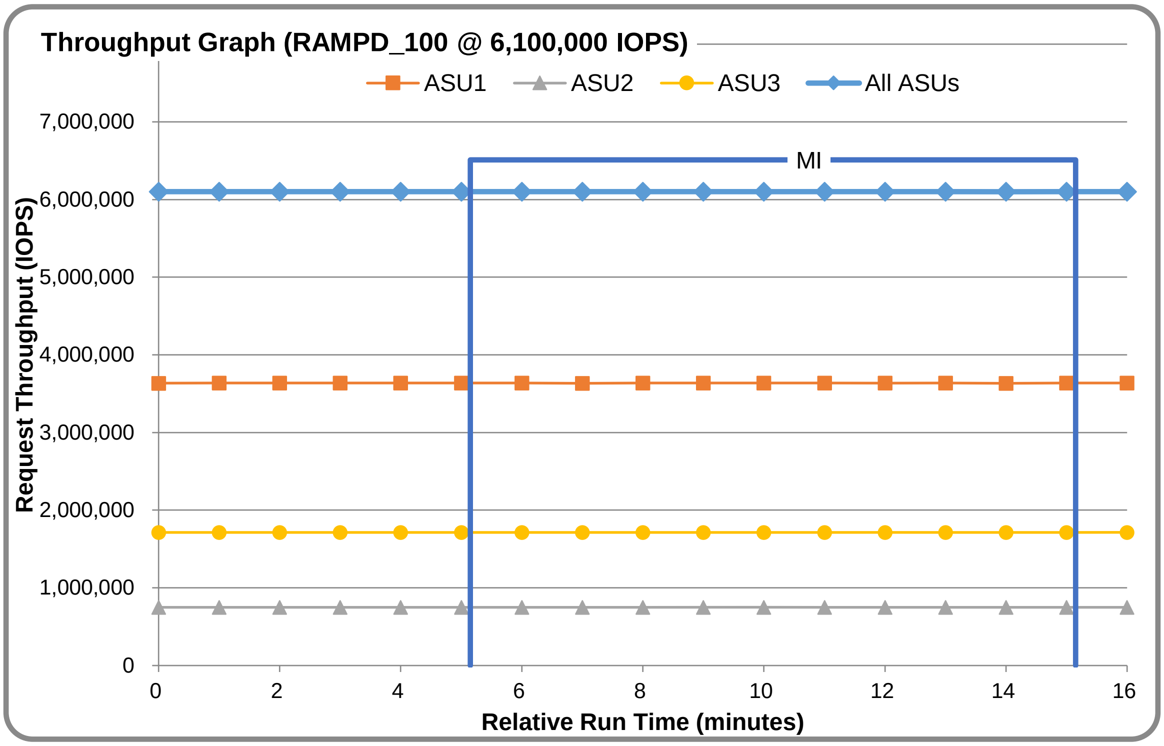
The results file generated during the execution of the RAMPD_100 Test Phase is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

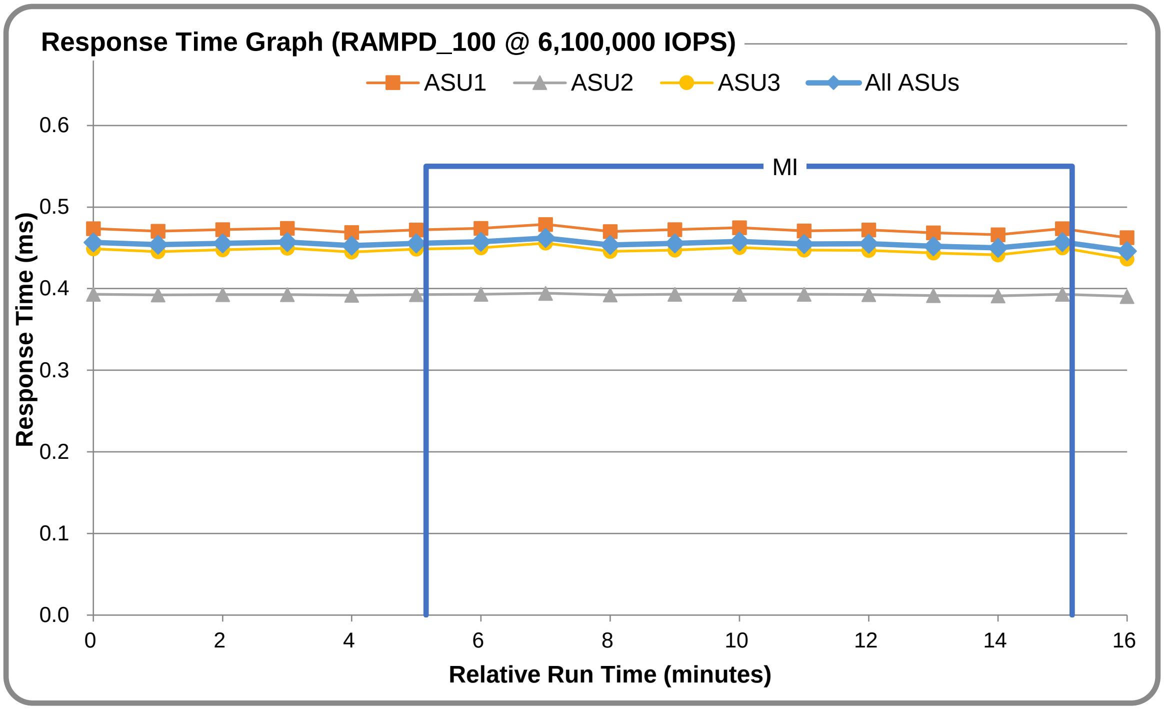
## RAMPD_100 – Execution Times

| Interval | Start Date & Time | End Date &Time | Duration |
|---|---|---|---|
| **Transition Period** | 25-Jun-19 03:44:58 | 25-Jun-19 03:49:59 | 0:05:01 |
| **Measurement Interval** | 25-Jun-19 03:49:59 | 25-Jun-19 03:59:59 | 0:10:00 |

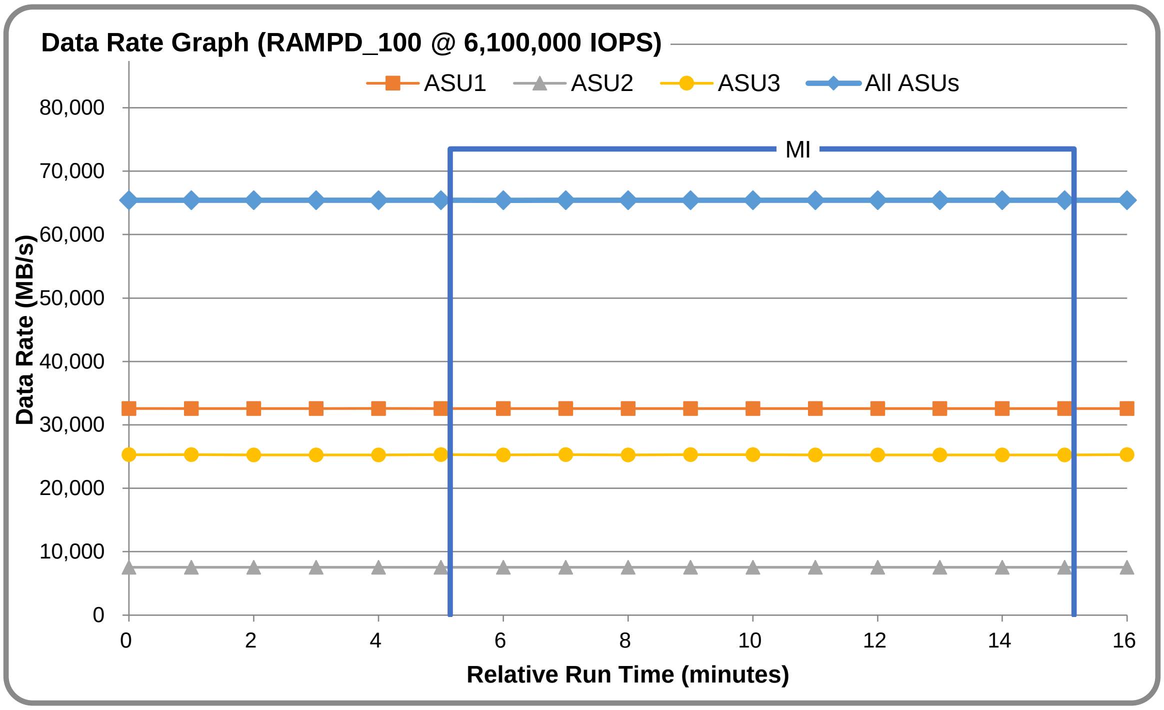
## RAMPD_100 – Throughput Graph



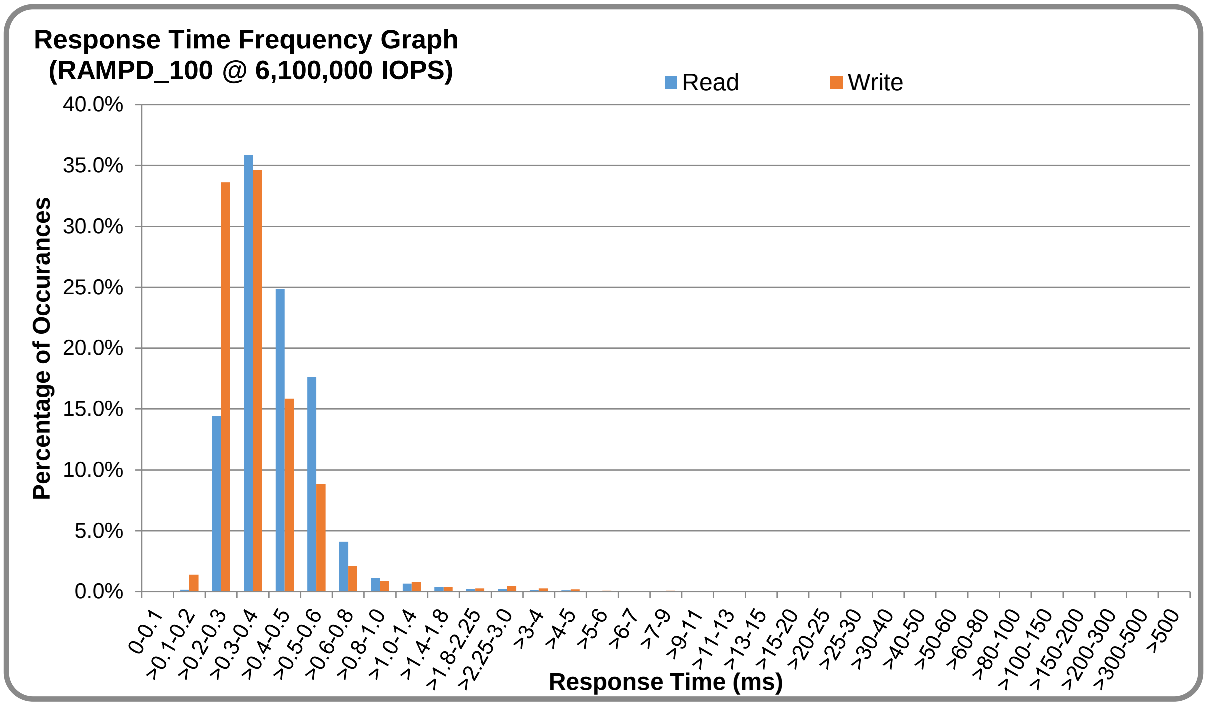
## RAMPD_100 – Response Time Graph



## RAMPD_100 – Data Rate Graph



## RAMPD_100 – Response Time Frequency Graph



## RAMPD_100 – Intensity Multiplier

The following table lists the targeted intensity multiplier (Defined), the measured intensity multiplier (Measured) for each I/O STREAM, its coefficient of variation (Variation) and the percentage of difference (Difference) between Target and Measured.

| | ASU1-1 | ASU1-2 | ASU1-3 | ASU1-4 | ASU2-1 | ASU2-2 | ASU2-3 | ASU3-1 |
|---|---|---|---|---|---|---|---|---|
| **Defined** | 0.0350 | 0.2810 | 0.0700 | 0.2100 | 0.0180 | 0.0700 | 0.0350 | 0.2810 |
| **Measured** | 0.0350 | 0.2810 | 0.0700 | 0.2100 | 0.0180 | 0.0700 | 0.0350 | 0.2810 |
| **Variation** | 0.0002 | 0.0001 | 0.0002 | 0.0001 | 0.0005 | 0.0002 | 0.0003 | 0.0001 |
| **Difference** | 0.001% | 0.002% | 0.001% | 0.004% | 0.012% | 0.003% | 0.006% | 0.002% |

## RAMPD_100 – I/O Request Summary

| | |
|---|---|
| **I/O Requests Completed in the Measurement Interval** | 3,660,086,101 |
| **I/O Requests Completed with Response Time <= 30 ms** | 3,659,881,511 |
| **I/O Requests Completed with Response Time > 30 ms** | 204,590 |

# Response Time Ramp Test

## Response Time Ramp Test – Results File

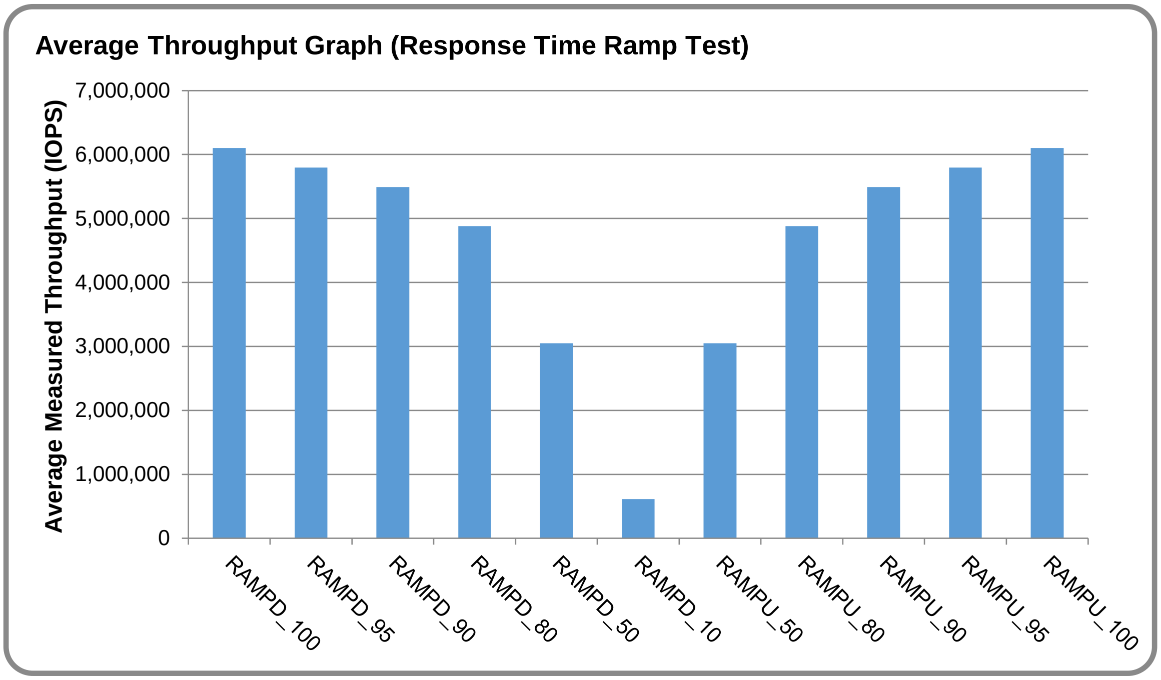
The results file generated during the execution of the Response Time Ramp Test is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_METRICS_0_Raw_Results.xlsx**

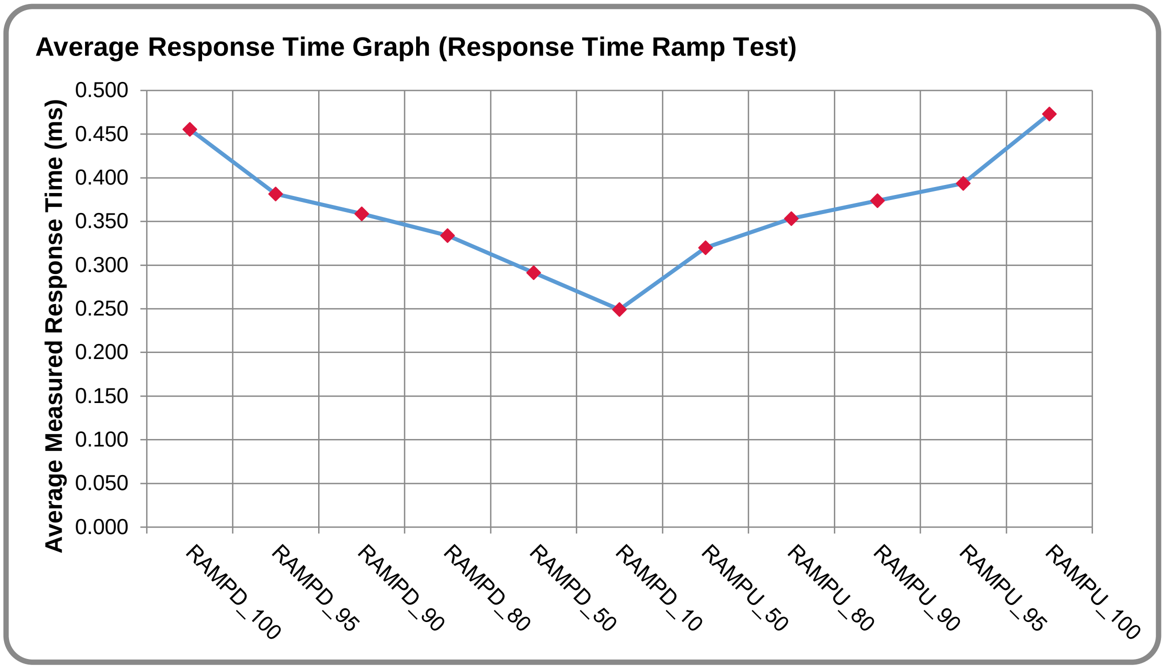
## Response Time Ramp Test – Phases

The Response Time Ramp Test is comprised of 11 Test Phases, including six Ramp-Down Phases (executed at 100%, 95%, 90%, 80%, 50%, and 10% of the Business Scaling Unit) and five Ramp-Up Phases (executed at 50%, 80%, 90%, 95%, and 100% of the Business Scaling Unit).

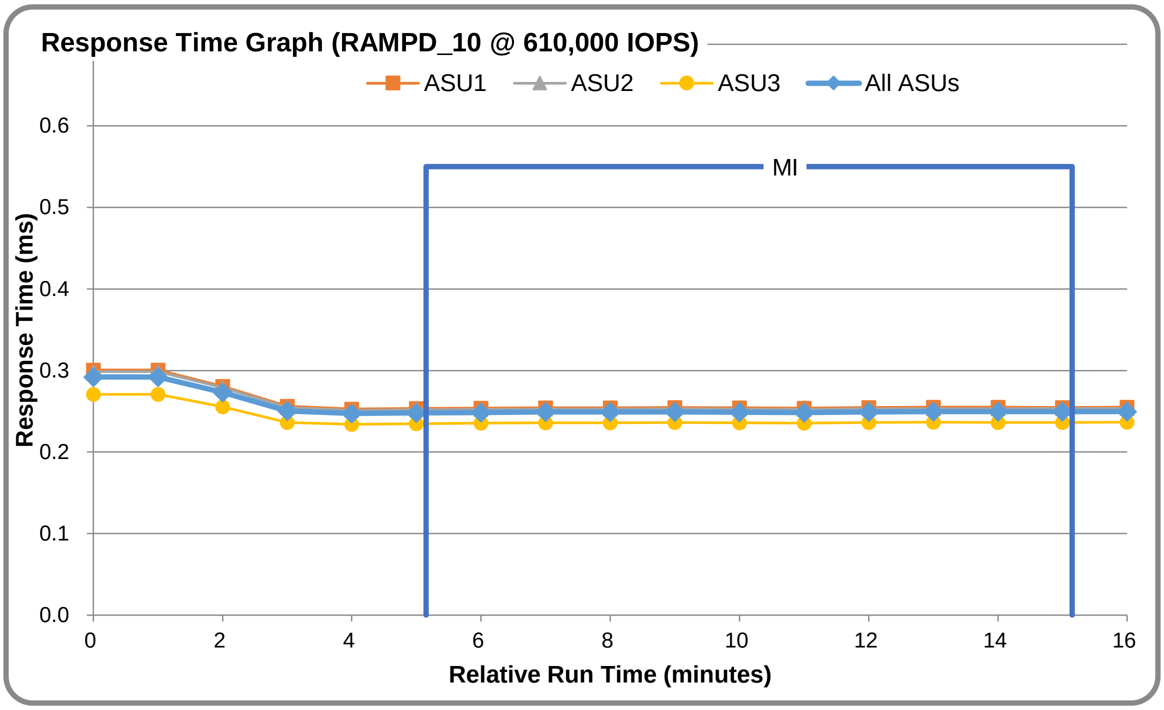
## Response Time Ramp Test – Average Throughput Graph



## Response Time Ramp Test – Average Response Time Graph



## Response Time Ramp Test – RAMPD_10 Response Time Graph



# Repeatability Test

## Repeatability Test Results File

The results file generated during the execution of the Repeatability Test is included in the Supporting Files (see Appendix A) as follows:

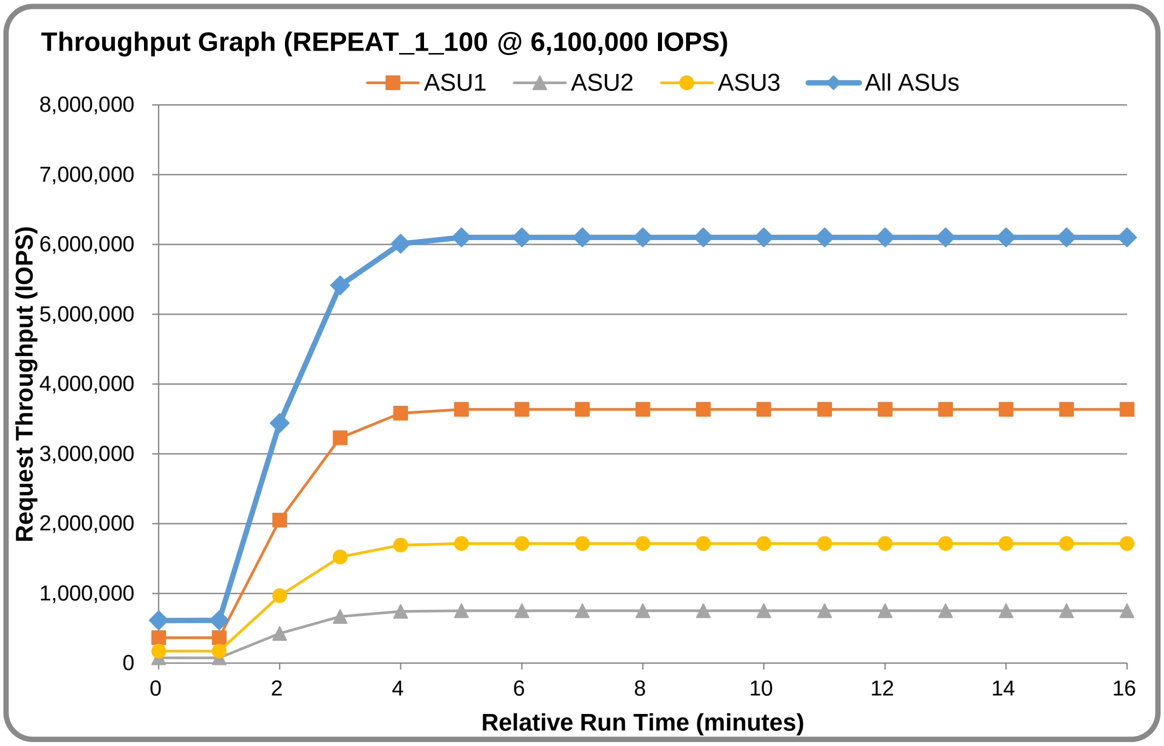
- **SPC1_METRICS_0_Raw_Results.xlsx**

## Repeatability Test Results

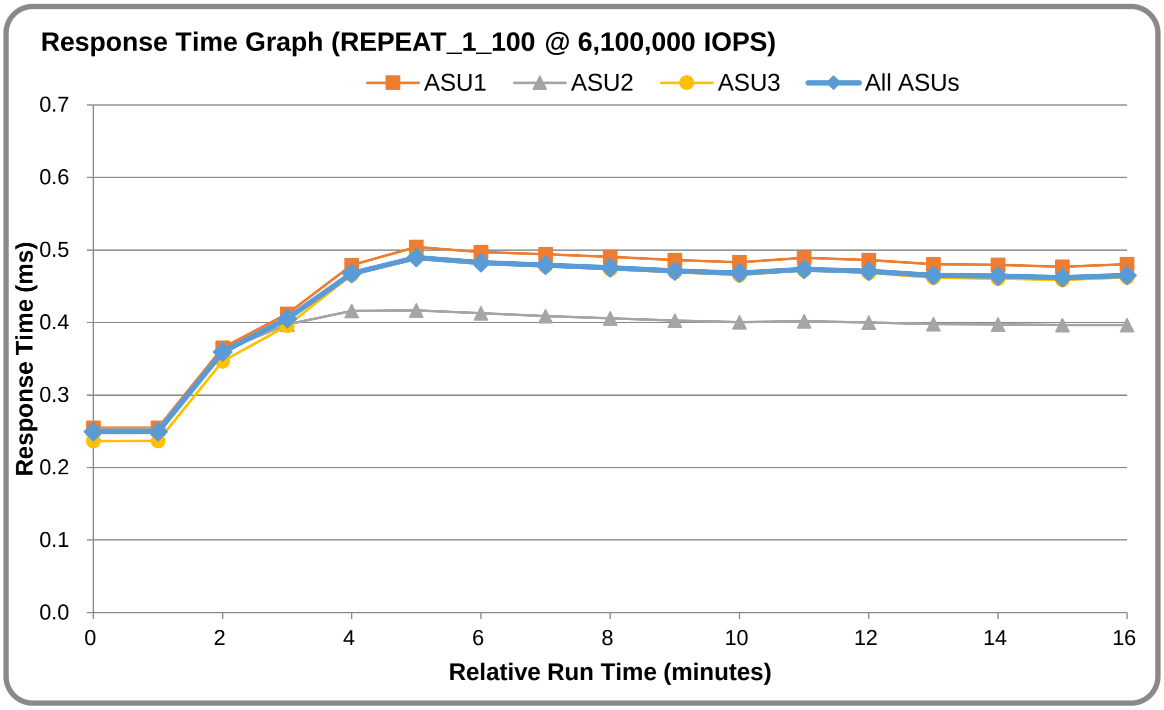
The throughput measurements for the Response Time Ramp Test (RAMPD) and the Repeatability Test Phases (REPEAT_1 and REPEAT_2) are listed in the tables below.

| Test Phase | 100% IOPS | 10% IOPS |
|---|---|---|
| RAMPD | 6,100,137.2 | 610,031.8 |
| REPEAT_1 | 6,100,508.0 | 610,060.3 |
| REPEAT_2 | 6,100,324.2 | 610,047.4 |

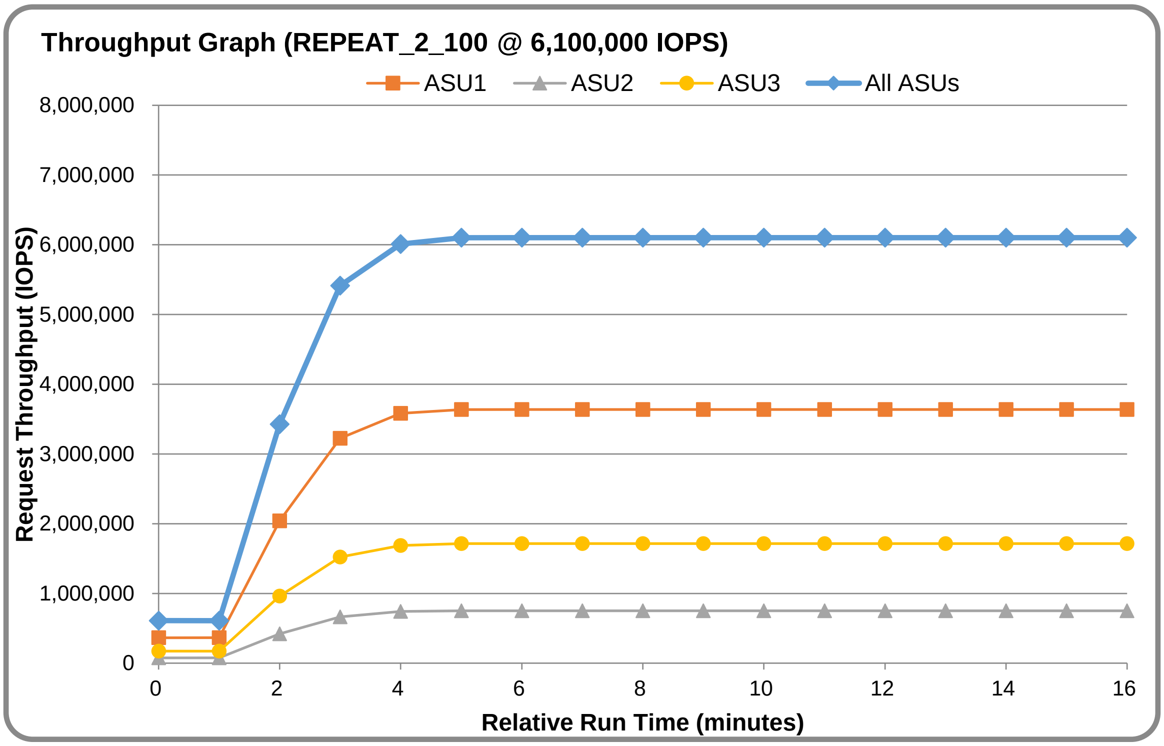
## REPEAT_1_100 – Throughput Graph



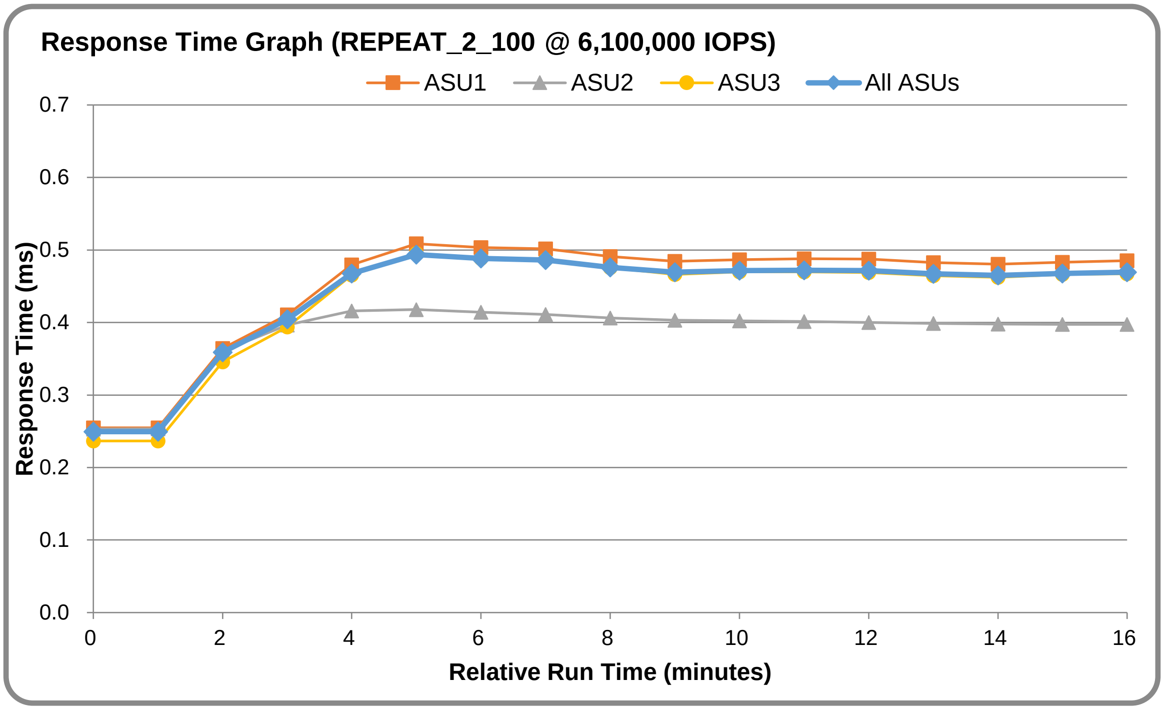
## REPEAT_1_100 – Response Time Graph



## REPEAT_2_100 – Throughput Graph



## REPEAT_2_100 – Response Time Graph



## Repeatability Test – Intensity Multiplier

The following tables lists the targeted intensity multiplier (Defined), the measured intensity multiplier (Measured) for each I/O STREAM, its coefficient of variation (Variation) and the percent of difference (Difference) between Target and Measured.

### REPEAT_1_100 Test Phase

| | ASU1-1 | ASU1-2 | ASU1-3 | ASU1-4 | ASU2-1 | ASU2-2 | ASU2-3 | ASU3-1 |
|---|---|---|---|---|---|---|---|---|
| Defined | 0.0350 | 0.2810 | 0.0700 | 0.2100 | 0.0180 | 0.0700 | 0.0350 | 0.2810 |
| Measured | 0.0350 | 0.2810 | 0.0700 | 0.2100 | 0.0180 | 0.0700 | 0.0350 | 0.2810 |
| Variation | 0.0002 | 0.0001 | 0.0001 | 0.0001 | 0.0004 | 0.0002 | 0.0003 | 0.0001 |
| Difference | 0.003% | 0.001% | 0.007% | 0.001% | 0.006% | 0.003% | 0.014% | 0.002% |

### REPEAT_2_100 Test Phase

| | ASU1-1 | ASU1-2 | ASU1-3 | ASU1-4 | ASU2-1 | ASU2-2 | ASU2-3 | ASU3-1 |
|---|---|---|---|---|---|---|---|---|
| Defined | 0.0350 | 0.2810 | 0.0700 | 0.2100 | 0.0180 | 0.0700 | 0.0350 | 0.2810 |
| Measured | 0.0350 | 0.2810 | 0.0700 | 0.2100 | 0.0180 | 0.0700 | 0.0350 | 0.2810 |
| Variation | 0.0002 | 0.0001 | 0.0002 | 0.0000 | 0.0005 | 0.0002 | 0.0001 | 0.0001 |
| Difference | 0.006% | 0.004% | 0.009% | 0.000% | 0.002% | 0.012% | 0.005% | 0.003% |

# Space Optimization Reporting

## Description of Utilized Techniques

No space optimization was used for this SPC-1 result.

## Physical Free Space Measurements

The following table lists the Physical Free Space as measured at each of the required points during test execution. If space optimization techniques were not used, "NA" is reported.

| Physical Free Space Measurement | Free Space (GB) |
|---|---|
| **After Logical Volume Creation** | NA |
| **After ASU Pre-Fill** | NA |
| **After Repeatability Test Phase** | NA |

## Space Optimization Metrics

The following table lists the required space optimization metrics. If space optimization techniques were not used, "NA" is reported.

| Space Optimization Metric | Value |
|---|---|
| **SPC-1 Space Optimization Ratio** | NA |
| **SPC-1 Space Effectiveness Ratio** | NA |

# Data Persistence Test

## Data Persistence Test Result files

The results files generated during the execution of the Data Persistence Test is included in the Supporting Files (see Appendix A) as follows:

- **SPC1_PERSIST_1_0_Raw_Results.xlsx**
- **SPC1_PERSIST_2_0_Raw_Results.xlsx**

## Data Persistence Test Execution

The Data Persistence Test was executed using the following sequence of steps:

- The PERSIST_1_0 Test Phase was executed to completion.
- The Benchmark Configuration was taken through an orderly shutdown process and powered off.
- The Benchmark Configuration was powered on and taken through an orderly startup process.
- The PERSIST_2_0 Test Phase was executed to completion.

## Data Persistence Test Results

| Data Persistence Test Phase: Persist1 | |
|---|---|
| **Total Number of Logical Blocks Written** | 754,020,246 |
| **Total Number of Logical Blocks Verified** | 370,365,473 |
| **Total Number of Logical Blocks Overwritten** | 383,654,773 |
| **Total Number of Logical Blocks that Failed Verification** | 0 |
| **Time Duration for Writing Test Logical Blocks (sec.)** | 301 |
| **Size in Bytes of each Logical Block** | 8,192 |
| **Number of Failed I/O Requests During the Test** | 0 |

## Committed Data Persistence Implementation

The TSC uses cache protection technology. Each MS5580G2 Node has four system disks, an internal set of five batteries, and an external set of two batteries in the enclosure holding the system disks. When a Node loses power, the two sets of batteries allow the content of the cache to be flushed to the system disks for permanent storage. When the Node is restarted, the cache content is automatically recovered from the system disks.

# APPENDIX A: SUPPORTING FILES

The following table details the content of the Supporting Files provided as part of this Full Disclosure Report.

| File Name | Description | Location |
|---|---|---|
| **/SPC1_RESULTS** | **Data reduction worksheets** | **root** |
| SPC1_INIT_0_Raw_Results.xlsx | Raw results for INIT Test Phase | /SPC1_RESULTS |
| SPC1_METRICS_0_Quick_Look.xlsx | Quick Look Test Run Overview | /SPC1_RESULTS |
| SPC1_METRICS_0_Raw_Results.xlsx | Raw results for Primary Metrics Test | /SPC1_RESULTS |
| SPC1_METRICS_0_Summary_Results.xlsx | Primary Metrics Summary | /SPC1_RESULTS |
| SPC1_PERSIST_1_0_Raw_Results.xlsx | Raw results for PERSIST1 Test Phase | /SPC1_RESULTS |
| SPC1_PERSIST_2_0_Raw_Results.xlsx | Raw results for PERSIST2 Test Phase | /SPC1_RESULTS |
| SPC1_Run_Set_Overview.xlsx | Run Set Overview Worksheet | /SPC1_RESULTS |
| SPC1_VERIFY_0_Raw_Results.xlsx | Raw results for first VERIFY Test Phase | /SPC1_RESULTS |
| SPC1_VERIFY_1_Raw_Results.xlsx | Raw results for second VERIFY Test Phase | /SPC1_RESULTS |
| **/C_Tuning** | **Tuning parameters and options** | **root** |
| aio-max-nr.sh | Set maximum asynchronous I/O | /C_Tuning |
| nr_request.sh | Increase disk queue depth | /C_Tuning |
| scheduler.sh | Change the I/O scheduler | /C_Tuning |
| **/D_Creation** | **Storage configuration creation** | **root** |
| init_ms5580g2.sh | Create the storage environment | /D_Creation |
| mklvm_ms5580g2.sh | Create the Logical Volumes | /D_Creation |
| lv_scan.sh | Scan the Logical Volumes | /D_Creation |
| **/E_Inventory** | **Configuration inventory** | **root** |
| volume_list.sh | Captures list of Logical Volumes | /E_Inventory |
| profile_ms5580g2.sh | Captures storage devices profiles | /E_Inventory |
| volume_listing_start.txt | List of logical volumes before INIT | /E_Inventory |
| profile_start_ms5580g2.txt | List of storage devices before INIT | /E_Inventory |
| volume_listing_end.txt | List of logical volumes after restart | /E_Inventory |
| profile_end_ms5580g2.txt | List of storage devices after restart | /E_Inventory |
| **/F_Generator** | **Workload generator** | **root** |
| slave_asu.asu | Defining LUNs hosting the ASUs | /F_generator |
| 20.HST | Host configuration file | /F_generator |
| Ms5580g2_test_phase1.sh | Executing test phases up to VERIFY_1 | /F_generator |
| Ms5580g2_test_persist2.sh | Executing test phase PERSIST_1 | /F_generator |
| Ms5580g2_test_phase3.sh | Executing test phase PERSIST_2 | /F_generator |

## APPENDIX B: THIRD PARTY QUOTATION

All components are directly available through the Test Sponsor.

# APPENDIX C: TUNING PARAMETERS AND OPTIONS

The following scripts, listed below, were used to set tuning parameters and options:

- ***aio-max-nr.sh*** to set the maximum asynchronous I/O
- ***nr_request.sh*** to increase the disk queue depth
- ***scheduler.sh*** to change the I/O scheduler

The scripts described above are included in the Supporting Files (see Appendix A) and listed below.

### *aio-max-nr.sh*

```
echo 1048576 > /proc/sys/fs/aio-max-nr
cat   /proc/sys/fs/aio-max-nr
```

### *nr_request.sh*

```
#!/bin/bash

#default:
for i in sdaa sdab sdac sdad sdae sdaf sdag sdah sdai sdaj sdak sdal sdam sdan
   sdao sdap sdaq sdar sdas sdat sdau sdav sdaw sdax sday sdaz sdb sdba sdbb sdbc
   sdbd sdbe sdbf sdbg sdbh sdbi sdbj sdbk sdbl sdbm sdbn sdbo sdbp sdbq sdbr
   sdbs sdbt sdbu sdbv sdbw sdbx sdby sdbz sdc sdca sdcb sdcc sdcd sdce sdcf sdcg
   sdch sdci sdcj sdck sdcl sdcm sdcn sdco sdcp sdcq sdcr sdcs sdct sdcu sdcv
   sdcw sdcx sdcy sdcz sdd sdda sddb sddc sddd sdde sddf sddg sddh sddi sddj sddk
   sddl sddm sddn sddo sddp sddq sddr sdds sddt sddu sddv sddw sddx sddy sddz sde
   sdea sdeb sdec sded sdee sdef sdeg sdeh sdei sdej sdek sdel sdem sden sdeo
   sdep sdeq sder sdes sdet sdeu sdev sdew sdex sdey sdez sdf sdfa sdfb sdfc sdfd
   sdfe sdff sdfg sdfh sdfi sdfj sdfk sdfl sdfm sdfn sdfo sdfp sdfq sdfr sdfs
   sdft sdfu sdfv sdfw sdfx sdfy sdfz sdg sdga sdgb sdgc sdgd sdge sdgf sdgg sdgh
   sdgi sdgj sdgk sdgl sdgm sdgn sdgo sdgp sdgq sdgr sdgs sdgt sdgu sdgv sdgw
   sdgx sdgy sdgz sdh sdha sdhb sdhc sdhd sdhe sdhf sdhg sdhh sdhi sdhj sdhk sdhl
   sdhm sdhn sdho sdhp sdhq sdhr sdhs sdht sdhu sdhv sdhw sdhx sdhy sdhz sdi sdia
   sdib sdic sdid sdie sdif sdig sdih sdii sdij sdik sdil sdim sdin sdio sdip
   sdiq sdir sdis sdit sdiu sdiv sdiw sdix sdiy sdiz sdj sdja sdjb sdjc sdjd sdje
   sdjf sdjg sdjh sdji sdjj sdjk sdjl sdjm sdjn sdjo sdjp sdjq sdjr sdjs sdjt
   sdju sdjv sdjw sdjx sdjy sdjz sdk sdka sdkb sdkc sdkd sdke sdkf sdkg sdkh sdki
   sdkj sdkk sdkl sdkm sdkn sdko sdkp sdkq sdkr sdks sdkt sdku sdkv sdkw sdkx
   sdky sdkz sdl sdla sdlb sdlc sdld sdle sdlf sdlg sdlh sdli sdlj sdlk sdll sdlm
   sdln sdlo sdlp sdlq sdlr sdls sdlt sdlu sdlv sdlw sdlx sdly sdlz sdm sdma sdmb
   sdmc sdmd sdme sdmf sdmg sdmh sdmi sdmj sdmk sdml sdmm sdmn sdmo sdmp sdmq
   sdmr sdms sdmt sdmu sdmv sdmw sdmx sdmy sdmz sdn sdna sdnb sdnc sdnd sdne sdnf
   sdng sdnh sdni sdnj sdnk sdnl sdnm sdnn sdno sdnp sdnq sdnr sdns sdnt sdnu sdo
   sdp sdq sdr sds sdt sdu sdv sdw sdx sdy sdz
do
   echo 1024 > /sys/block/$i/queue/nr_requests
done

#show
for j in sdaa sdab sdac sdad sdae sdaf sdag sdah sdai sdaj sdak sdal sdam sdan
   sdao sdap sdaq sdar sdas sdat sdau sdav sdaw sdax sday sdaz sdb sdba sdbb sdbc
   sdbd sdbe sdbf sdbg sdbh sdbi sdbj sdbk sdbl sdbm sdbn sdbo sdbp sdbq sdbr
   sdbs sdbt sdbu sdbv sdbw sdbx sdby sdbz sdc sdca sdcb sdcc sdcd sdce sdcf sdcg
```

```
      sdch sdci sdcj sdck sdcl sdcm sdcn sdco sdcp sdcq sdcr sdcs sdct sdcu sdcv
      sdcw sdcx sdcy sdcz sdd sdda sddb sddc sddd sdde sddf sddg sddh sddi sddj sddk
      sddl sddm sddn sddo sddp sddq sddr sdds sddt sddu sddv sddw sddx sddy sddz sde
      sdea sdeb sdec sded sdee sdef sdeg sdeh sdei sdej sdek sdel sdem sden sdeo
      sdep sdeq sder sdes sdet sdeu sdev sdew sdex sdey sdez sdf sdfa sdfb sdfc sdfd
      sdfe sdff sdfg sdfh sdfi sdfj sdfk sdfl sdfm sdfn sdfo sdfp sdfq sdfr sdfs
      sdft sdfu sdfv sdfw sdfx sdfy sdfz sdg sdga sdgb sdgc sdgd sdge sdgf sdgg sdgh
      sdgi sdgj sdgk sdgl sdgm sdgn sdgo sdgp sdgq sdgr sdgs sdgt sdgu sdgv sdgw
      sdgx sdgy sdgz sdh sdha sdhb sdhc sdhd sdhe sdhf sdhg sdhh sdhi sdhj sdhk sdhl
      sdhm sdhn sdho sdhp sdhq sdhr sdhs sdht sdhu sdhv sdhw sdhx sdhy sdhz sdi sdia
      sdib sdic sdid sdie sdif sdig sdih sdii sdij sdik sdil sdim sdin sdio sdip
      sdiq sdir sdis sdit sdiu sdiv sdiw sdix sdiy sdiz sdj sdja sdjb sdjc sdjd sdje
      sdjf sdjg sdjh sdji sdjj sdjk sdjl sdjm sdjn sdjo sdjp sdjq sdjr sdjs sdjt
      sdju sdjv sdjw sdjx sdjy sdjz sdk sdka sdkb sdkc sdkd sdke sdkf sdkg sdkh sdki
      sdkj sdkk sdkl sdkm sdkn sdko sdkp sdkq sdkr sdks sdkt sdku sdkv sdkw sdkx
      sdky sdkz sdl sdla sdlb sdlc sdld sdle sdlf sdlg sdlh sdli sdlj sdlk sdll sdlm
      sdln sdlo sdlp sdlq sdlr sdls sdlt sdlu sdlv sdlw sdlx sdly sdlz sdm sdma sdmb
      sdmc sdmd sdme sdmf sdmg sdmh sdmi sdmj sdmk sdml sdmm sdmn sdmo sdmp sdmq
      sdmr sdms sdmt sdmu sdmv sdmw sdmx sdmy sdmz sdn sdna sdnb sdnc sdnd sdne sdnf
      sdng sdnh sdni sdnj sdnk sdnl sdnm sdnn sdno sdnp sdnq sdnr sdns sdnt sdnu sdo
      sdp sdq sdr sds sdt sdu sdv sdw sdx sdy sdz
    do
      cat -n /sys/block/$j/queue/nr_requests
    done

    #cat -n /sys/block/sda[a-z]/queue/nr_requests
```

### ***scheduler.sh***

```
#!/bin/bash


#default: noop anticipatory deadline [cfq]

for i in sdaa sdab sdac sdad sdae sdaf sdag sdah sdai sdaj sdak sdal sdam sdan sdao sdap
sdaq sdar sdas sdat sdau sdav sdaw sdax sday sdaz sdb sdba sdbb sdbc sdbd sdbe sdbf sdbg
sdbh sdbi sdbj sdbk sdbl sdbm sdbn sdbo sdbp sdbq sdbr sdbs sdbt sdbu sdbv sdbw sdbx
sdby sdbz sdc sdca sdcb sdcc sdcd sdce sdcf sdcg sdch sdci sdcj sdck sdcl sdcm sdcn sdco
sdcp sdcq sdcr sdcs sdct sdcu sdcv sdcw sdcx sdcy sdcz sdd sdda sddb sddc sddd sdde sddf
sddg sddh sddi sddj sddk sddl sddm sddn sddo sddp sddq sddr sdds sddt sddu sddv sddw
sddx sddy sddz sde sdea sdeb sdec sded sdee sdef sdeg sdeh sdei sdej sdek sdel sdem sden
sdeo sdep sdeq sder sdes sdet sdeu sdev sdew sdex sdey sdez sdf sdfa sdfb sdfc sdfd sdfe
sdff sdfg sdfh sdfi sdfj sdfk sdfl sdfm sdfn sdfo sdfp sdfq sdfr sdfs sdft sdfu sdfv
sdfw sdfx sdfy sdfz sdg sdga sdgb sdgc sdgd sdge sdgf sdgg sdgh sdgi sdgj sdgk sdgl sdgm
sdgn sdgo sdgp sdgq sdgr sdgs sdgt sdgu sdgv sdgw sdgx sdgy sdgz sdh sdha sdhb sdhc sdhd
sdhe sdhf sdhg sdhh sdhi sdhj sdhk sdhl sdhm sdhn sdho sdhp sdhq sdhr sdhs sdht sdhu
sdhv sdhw sdhx sdhy sdhz sdi sdia sdib sdic sdid sdie sdif sdig sdih sdii sdij sdik sdil
sdim sdin sdio sdip sdiq sdir sdis sdit sdiu sdiv sdiw sdix sdiy sdiz sdj sdja sdjb sdjc
sdjd sdje sdjf sdjg sdjh sdji sdjj sdjk sdjl sdjm sdjn sdjo sdjp sdjq sdjr sdjs sdjt
sdju sdjv sdjw sdjx sdjy sdjz sdk sdka sdkb sdkc sdkd sdke sdkf sdkg sdkh sdki sdkj sdkk
sdkl sdkm sdkn sdko sdkp sdkq sdkr sdks sdkt sdku sdkv sdkw sdkx sdky sdkz sdl sdla sdlb
sdlc sdld sdle sdlf sdlg sdlh sdli sdlj sdlk sdll sdlm sdln sdlo sdlp sdlq sdlr sdls
sdlt sdlu sdlv sdlw sdlx sdly sdlz sdm sdma sdmb sdmc sdmd sdme sdmf sdmg sdmh sdmi sdmj
sdmk sdml sdmm sdmn sdmo sdmp sdmq sdmr sdms sdmt sdmu sdmv sdmw sdmx sdmy sdmz sdn sdna
sdnb sdnc sdnd sdne sdnf sdng sdnh sdni sdnj sdnk sdnl sdnm sdnn sdno sdnp sdnq sdnr
sdns sdnt sdnu sdo sdp sdq sdr sds sdt sdu sdv sdw sdx sdy sdz

do

echo noop > /sys/block/$i/queue/scheduler

done
```

```
#show

for j in sdaa sdab sdac sdad sdae sdaf sdag sdah sdai sdaj sdak sdal sdam sdan sdao sdap
sdaq sdar sdas sdat sdau sdav sdaw sdax sday sdaz sdb sdba sdbb sdbc sdbd sdbe sdbf sdbg
sdbh sdbi sdbj sdbk sdbl sdbm sdbn sdbo sdbp sdbq sdbr sdbs sdbt sdbu sdbv sdbw sdbx
sdby sdbz sdc sdca sdcb sdcc sdcd sdce sdcf sdcg sdch sdci sdcj sdck sdcl sdcm sdcn sdco
sdcp sdcq sdcr sdcs sdct sdcu sdcv sdcw sdcx sdcy sdcz sdd sdda sddb sddc sddd sdde sddf
sddg sddh sddi sddj sddk sddl sddm sddn sddo sddp sddq sddr sdds sddt sddu sddv sddw
sddx sddy sddz sde sdea sdeb sdec sded sdee sdef sdeg sdeh sdei sdej sdek sdel sdem sden
sdeo sdep sdeq sder sdes sdet sdeu sdev sdew sdex sdey sdez sdf sdfa sdfb sdfc sdfd sdfe
sdff sdfg sdfh sdfi sdfj sdfk sdfl sdfm sdfn sdfo sdfp sdfq sdfr sdfs sdft sdfu sdfv
sdfw sdfx sdfy sdfz sdg sdga sdgb sdgc sdgd sdge sdgf sdgg sdgh sdgi sdgj sdgk sdgl sdgm
sdgn sdgo sdgp sdgq sdgr sdgs sdgt sdgu sdgv sdgw sdgx sdgy sdgz sdh sdha sdhb sdhc sdhd
sdhe sdhf sdhg sdhh sdhi sdhj sdhk sdhl sdhm sdhn sdho sdhp sdhq sdhr sdhs sdht sdhu
sdhv sdhw sdhx sdhy sdhz sdi sdia sdib sdic sdid sdie sdif sdig sdih sdii sdij sdik sdil
sdim sdin sdio sdip sdiq sdir sdis sdit sdiu sdiv sdiw sdix sdiy sdiz sdj sdja sdjb sdjc
sdjd sdje sdjf sdjg sdjh sdji sdjj sdjk sdjl sdjm sdjn sdjo sdjp sdjq sdjr sdjs sdjt
sdju sdjv sdjw sdjx sdjy sdjz sdk sdka sdkb sdkc sdkd sdke sdkf sdkg sdkh sdki sdkj sdkk
sdkl sdkm sdkn sdko sdkp sdkq sdkr sdks sdkt sdku sdkv sdkw sdkx sdky sdkz sdl sdla sdlb
sdlc sdld sdle sdlf sdlg sdlh sdli sdlj sdlk sdll sdlm sdln sdlo sdlp sdlq sdlr sdls
sdlt sdlu sdlv sdlw sdlx sdly sdlz sdm sdma sdmb sdmc sdmd sdme sdmf sdmg sdmh sdmi sdmj
sdmk sdml sdmm sdmn sdmo sdmp sdmq sdmr sdms sdmt sdmu sdmv sdmw sdmx sdmy sdmz sdn sdna
sdnb sdnc sdnd sdne sdnf sdng sdnh sdni sdnj sdnk sdnl sdnm sdnn sdno sdnp sdnq sdnr
sdns sdnt sdnu sdo sdp sdq sdr sds sdt sdu sdv sdw sdx sdy sdz

do

cat -n /sys/block/$j/queue/scheduler

done

#cat -n /sys/block/sda[a-z]/queue/scheduler
```

# APPENDIX D: STORAGE CONFIGURATION CREATION

## Step 1 - Create Storage Pool, RAIDs, LUNs, Clients, Mapping

The ***init_ms5580g2.sh*** script, listed below, is executed on one of the Storage Controllers via a CLI session to perform the following actions:

1. Create a storage pool for each Node: Pool-A-1, Pool-B-1, Pool-C-1, Pool-D-1, Pool-E-1, Pool-F-1, Pool-G-1 and Pool-H-1
2. Create 64 RAID 10 arrays (8 RAID 10 for each storage node)
3. Create 384 LUNs (6 LUNs per RAID 10, 128 LUNs of 370GB, 128 LUNs of 371GB and 128 LUNs of 372GB)
4. Create 320 Clients (40 Clients for each storage node)
5. Add Host System FC Initiators for the Clients (1 Initiator per Client)
6. Create Storage Targets for the Clients (1 Target per Client and each Target is associated with 1 FC port)
7. Add LUNs to the Targets (24 LUNs per Target)

The script file described above is included in the Supporting Files (see Appendix A) and listed below.

### *init_ms5580g2.sh*

```
#!/bin/bash

node=("NodeA" "NodeB" "NodeC" "NodeD" "NodeE" "NodeF" "NodeG" "NodeH")
noli=("A" "B" "C" "D" "E" "F" "G" "H")
#Create     8*Pool:Pool-A-1/Pool-B-1/Pool-C-1/Pool-D-1/Pool-E-1/Pool-F-1/Pool-G-
   1/Pool-H-1/
echo "Waiting for create 8*Pool"
for((i=0;i<8;i++))
do
   ms-cli ${node[$i]} "pool mgt create -n Pool-${noli[$i]}-1 -o CRAID-P"
done

#Create RAID: 8 *8* RAID10
echo "Waiting for create 64 RAID"
for((j=0;j<8;j++))
do
   ms-cli ${node[$j]} "raid mgt create -n RAID-${noli[$j]}-6-001 -p Pool-
   ${noli[$j]}-1 -l RAID10 -d 6:1:1:1,6:1:1:2,6:1:1:3,6:1:1:4,6:1:1:7,6:1:1:8 -
   o CRAID-P -t SSD -i SAS -s 128KB -C 512"
   ms-cli ${node[$j]} "raid mgt create -n RAID-${noli[$j]}-6-002 -p Pool-
   ${noli[$j]}-1                -l                 RAID10                  -d
   6:1:1:9,6:1:1:10,6:1:1:13,6:1:1:14,6:1:1:15,6:1:1:16 -o CRAID-P -t SSD -i SAS
   -s 128KB -C 512"
   ms-cli ${node[$j]} "raid mgt create -n RAID-${noli[$j]}-6-003 -p Pool-
   ${noli[$j]}-1                -l                 RAID10                  -d
   6:1:1:19,6:1:1:20,6:1:1:21,6:1:1:22,10:1:1:19,10:1:1:20 -o CRAID-P -t SSD -i
   SAS -s 128KB -C 512"
   ms-cli ${node[$j]} "raid mgt create -n RAID-${noli[$j]}-9-001 -p Pool-
   ${noli[$j]}-1 -l RAID10 -d 9:1:1:1,9:1:1:2,9:1:1:3,9:1:1:4,9:1:1:7,9:1:1:8 -
   o CRAID-P -t SSD -i SAS -s 128KB -C 512"
```

```
    ms-cli  ${node[$j]}  "raid  mgt  create  -n  RAID-${noli[$j]}-9-002  -p  Pool-
     ${noli[$j]}-1                 -l                 RAID10                  -d
     9:1:1:9,9:1:1:10,9:1:1:13,9:1:1:14,9:1:1:15,9:1:1:16 -o CRAID-P -t SSD -i SAS
     -s 128KB -C 512"
    ms-cli  ${node[$j]}  "raid  mgt  create  -n  RAID-${noli[$j]}-9-003  -p  Pool-
     ${noli[$j]}-1                 -l                 RAID10                  -d
     9:1:1:19,9:1:1:20,9:1:1:21,9:1:1:22,10:1:1:21,10:1:1:22 -o CRAID-P -t SSD -i
     SAS -s 128KB -C 512"
    ms-cli  ${node[$j]}  "raid  mgt  create  -n  RAID-${noli[$j]}-10-001  -p  Pool-
     ${noli[$j]}-1                 -l                 RAID10                  -d
     10:1:1:5,10:1:1:6,10:1:1:7,10:1:1:8,10:1:1:9,10:1:1:10  -o  CRAID-P  -t  SSD  -i
     SAS -s 128KB -C 512"
    ms-cli  ${node[$j]}  "raid  mgt  create  -n  RAID-${noli[$j]}-10-002  -p  Pool-
     ${noli[$j]}-1                 -l                 RAID10                  -d
     10:1:1:11,10:1:1:12,10:1:1:13,10:1:1:14,10:1:1:15,10:1:1:16 -o CRAID-P -t SSD
     -i SAS -s 128KB -C 512"

done

#Create LUN: 48*8 LUN
echo "Waiting for create LUN"
for((k=0;k<8;k++))
do
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0001 -o SP1 -p
     Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-6-001 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0002 -o SP1 -p
     Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-6-001 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0003 -o SP1 -p
     Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-6-001 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0004 -o SP1 -p
     Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-6-002 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0005 -o SP1 -p
     Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-6-002 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0006 -o SP1 -p
     Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-6-002 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0007 -o SP1 -p
     Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-6-003 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0008 -o SP1 -p
     Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-6-003 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-0009 -o SP1 -p
     Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-6-003 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00010 -o SP1 -p
     Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-9-001 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00011 -o SP1 -p
     Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-9-001 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00012 -o SP1 -p
     Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-9-001 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00013 -o SP1 -p
     Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-9-002 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00014 -o SP1 -p
     Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-9-002 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00015 -o SP1 -p
     Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-9-002 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00016 -o SP1 -p
     Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-9-003 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00017 -o SP1 -p
     Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-9-003 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00018 -o SP1 -p
     Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-9-003 -r disable -w disable"
    ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00019 -o SP1 -p
     Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-10-001 -r disable -w disable"
```

```
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00020 -o SP1 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-10-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00021 -o SP1 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-10-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00022 -o SP1 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-10-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00023 -o SP1 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-10-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00024 -o SP1 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-10-002 -r disable -w disable"

   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00025 -o SP2 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-6-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00026 -o SP2 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-6-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00027 -o SP2 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-6-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00028 -o SP2 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-6-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00029 -o SP2 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-6-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00030 -o SP2 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-6-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00031 -o SP2 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-6-003 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00032 -o SP2 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-6-003 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00033 -o SP2 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-6-003 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00034 -o SP2 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-9-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00035 -o SP2 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-9-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00036 -o SP2 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-9-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00037 -o SP2 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-9-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00038 -o SP2 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-9-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00039 -o SP2 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-9-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00040 -o SP2 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-9-003 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00041 -o SP2 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-9-003 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00042 -o SP2 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-9-003 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00043 -o SP2 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-10-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00044 -o SP2 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-10-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00045 -o SP2 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-10-001 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00046 -o SP2 -p
    Pool-${noli[$k]}-1 -s 370GB -R RAID-${noli[$k]}-10-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00047 -o SP2 -p
    Pool-${noli[$k]}-1 -s 371GB -R RAID-${noli[$k]}-10-002 -r disable -w disable"
   ms-cli ${node[$k]} "lun mgt createthicklun -n LUN-${noli[$k]}-00048 -o SP2 -p
    Pool-${noli[$k]}-1 -s 372GB -R RAID-${noli[$k]}-10-002 -r disable -w disable"

done
```

```
#create itl
for((k1=0;k1<8;k1++))
do
    targetvalue=(`ms-cli ${node[$k1]} "client target queryportlist" |grep 'fc
    port' |sed -e 's/^.*> //g ; s/fc port-./FC-Target/g'`)
    initiatorvalue=(`ms-cli ${node[$k1]} "client target queryportlist" |grep
    'initiator\[0\]' |sed  's/^initiator\[0\]        : //g'`)
    targetna=(`ms-cli ${node[$k1]} "client target queryportlist" |grep  'fc port'
    |sed -e 's/^.*> //g ; s/fc port-./FC/g'`)
    targetport=(`ms-cli ${node[$k1]} "client target queryportlist "|grep 'fc
    port' |sed -e 's/^.*> //g ; s/fc port-..//g'`)
    itll=${#initiatorvalue[*]}
    for((k2=0;k2<$itll;k2++))
    do
        #Create client
        echo "Waiting for create client"
        ms-cli ${node[$k1]} "client mgt create -n ${targetport[$k2]}"
        #Create initiator
        echo "Waiting for create initiator"
        ms-cli ${node[$k1]} "client initiator modifyos -w ${initiatorvalue[$k2]} -
    o Linux"
        ms-cli ${node[$k1]} "client initiator modifyalias -w
    ${initiatorvalue[$k2]} -a ${targetport[$k2]}"
        #Add Target: 16
        echo "Waiting for add Target"
        ms-cli ${node[$k1]} "client target create -t fc -p ${targetna[$k2]}"
        #map-client
        echo "Waiting for map-client"
        ms-cli ${node[$k1]} "client mgt mapinitiator -c ${targetport[$k2]} -i
    ${initiatorvalue[$k2]}"
        #map-target
        echo "Waiting for map-target"
        ms-cli ${node[$k1]} "client itl maptarget -i ${initiatorvalue[$k2]} -t
    ${targetvalue[$k2]}"
        #map LUN
        echo "Waiting for map LUN"
        if [[ $k2 -lt 20 ]]
        then
                for((k3=1;k3<25;k3++))
                do
                        ms-cli ${node[$k1]} "client itl maplun -i
    ${initiatorvalue[$k2]} -t ${targetvalue[$k2]} -l LUN-${noli[$k1]}-000$k3"
                done
        else
                for((k3=25;k3<49;k3++))
                do
                  ms-cli ${node[$k1]} "client itl maplun -i ${initiatorvalue[$k2]}
    -t ${targetvalue[$k2]} -l LUN-${noli[$k1]}-000$k3"
                done
        fi
    done
done


echo "complete "
exit 0
```

## Step 2 - Create Volumes on the Master Host System

The script ***mklvm_ms5580g2.sh*** is executed on the Master Host System to create 9 logical volumes as follows:

1. Create Physical Volume:

   Create 384 physical volumes using the pvcreate command.

2. Create Volumes Groups:

   Create 3 volume groups (vg11, vg22 and vg33) using the vgcreate command over the 384 physical volumes.

3. Create Logical Volumes:

   Create 1 logical volume with a volume capacity of 21,186GB on each volume group for ASU-1.

   Create 1 logical volume with a volume capacity of 21,186GB on each volume group for ASU-2.

   Create 1 logical volume with a volume capacity of 4,708GB on each volume group for ASU-3.

4. Scan Physical Volumes, Volume Groups, and Logical Volumes; and activate each Logical Volume:

   Execute the ***lv_scan.sh*** on the Slave Host Systems to scan the physical volumes, volume group and logical volumes; and make each logical volume available (activate).

The script files described above are included in the Supporting Files (see Appendix A) and listed below.

### ***mklvm_ms5580g2.sh***

```
#!/bin/bash
#create_pv

pvcreate /dev/sdaa
pvcreate /dev/sdab
pvcreate /dev/sdac
pvcreate /dev/sdad
pvcreate /dev/sdae
pvcreate /dev/sdaf
pvcreate /dev/sdag
pvcreate /dev/sdah
pvcreate /dev/sdai
pvcreate /dev/sdaj
pvcreate /dev/sdak
pvcreate /dev/sdal
pvcreate /dev/sdam
pvcreate /dev/sdan
pvcreate /dev/sdao
pvcreate /dev/sdap
pvcreate /dev/sdaq
pvcreate /dev/sdar
pvcreate /dev/sdas
pvcreate /dev/sdat
```

```
       pvcreate /dev/sdau
       pvcreate /dev/sdav
       pvcreate /dev/sdaw
       pvcreate /dev/sdax
       pvcreate /dev/sday
       pvcreate /dev/sdaz
       pvcreate /dev/sdb
       pvcreate /dev/sdba
       pvcreate /dev/sdbb
       pvcreate /dev/sdbc
       pvcreate /dev/sdbd
       pvcreate /dev/sdbe
       pvcreate /dev/sdbf
       pvcreate /dev/sdbg
       pvcreate /dev/sdbh
       pvcreate /dev/sdbi
       pvcreate /dev/sdbj
       pvcreate /dev/sdbk
       pvcreate /dev/sdbl
       pvcreate /dev/sdbm
       pvcreate /dev/sdbn
       pvcreate /dev/sdbo
       pvcreate /dev/sdbp
       pvcreate /dev/sdbq
       pvcreate /dev/sdbr
       pvcreate /dev/sdbs
       pvcreate /dev/sdbt
       pvcreate /dev/sdbu
       pvcreate /dev/sdbv
       pvcreate /dev/sdbw
       pvcreate /dev/sdbx
       pvcreate /dev/sdby
       pvcreate /dev/sdbz
       pvcreate /dev/sdc
       pvcreate /dev/sdca
       pvcreate /dev/sdcb
       pvcreate /dev/sdcc
       pvcreate /dev/sdcd
       pvcreate /dev/sdce
       pvcreate /dev/sdcf
       pvcreate /dev/sdcg
       pvcreate /dev/sdch
       pvcreate /dev/sdci
       pvcreate /dev/sdcj
       pvcreate /dev/sdck
       pvcreate /dev/sdcl
       pvcreate /dev/sdcm
       pvcreate /dev/sdcn
       pvcreate /dev/sdco
       pvcreate /dev/sdcp
       pvcreate /dev/sdcq
       pvcreate /dev/sdcr
       pvcreate /dev/sdcs
       pvcreate /dev/sdct
       pvcreate /dev/sdcu
       pvcreate /dev/sdcv
       pvcreate /dev/sdcw
       pvcreate /dev/sdcx
       pvcreate /dev/sdcy
       pvcreate /dev/sdcz
       pvcreate /dev/sdd
       pvcreate /dev/sdda
```

```
        pvcreate /dev/sddb
        pvcreate /dev/sddc
        pvcreate /dev/sddd
        pvcreate /dev/sdde
        pvcreate /dev/sddf
        pvcreate /dev/sddg
        pvcreate /dev/sddh
        pvcreate /dev/sddi
        pvcreate /dev/sddj
        pvcreate /dev/sddk
        pvcreate /dev/sddl
        pvcreate /dev/sddm
        pvcreate /dev/sddn
        pvcreate /dev/sddo
        pvcreate /dev/sddp
        pvcreate /dev/sddq
        pvcreate /dev/sddr
        pvcreate /dev/sdds
        pvcreate /dev/sddt
        pvcreate /dev/sddu
        pvcreate /dev/sddv
        pvcreate /dev/sddw
        pvcreate /dev/sddx
        pvcreate /dev/sddy
        pvcreate /dev/sddz
        pvcreate /dev/sde
        pvcreate /dev/sdea
        pvcreate /dev/sdeb
        pvcreate /dev/sdec
        pvcreate /dev/sded
        pvcreate /dev/sdee
        pvcreate /dev/sdef
        pvcreate /dev/sdeg
        pvcreate /dev/sdeh
        pvcreate /dev/sdei
        pvcreate /dev/sdej
        pvcreate /dev/sdek
        pvcreate /dev/sdel
        pvcreate /dev/sdem
        pvcreate /dev/sden
        pvcreate /dev/sdeo
        pvcreate /dev/sdep
        pvcreate /dev/sdeq
        pvcreate /dev/sder
        pvcreate /dev/sdes
        pvcreate /dev/sdet
        pvcreate /dev/sdeu
        pvcreate /dev/sdev
        pvcreate /dev/sdew
        pvcreate /dev/sdex
        pvcreate /dev/sdey
        pvcreate /dev/sdez
        pvcreate /dev/sdf
        pvcreate /dev/sdfa
        pvcreate /dev/sdfb
        pvcreate /dev/sdfc
        pvcreate /dev/sdfd
        pvcreate /dev/sdfe
        pvcreate /dev/sdff
        pvcreate /dev/sdfg
        pvcreate /dev/sdfh
        pvcreate /dev/sdfi
        pvcreate /dev/sdfj
        pvcreate /dev/sdfk
        pvcreate /dev/sdfl
        pvcreate /dev/sdfm
        pvcreate /dev/sdfn
        pvcreate /dev/sdfo
        pvcreate /dev/sdfp
        pvcreate /dev/sdfq
        pvcreate /dev/sdfr
        pvcreate /dev/sdfs
        pvcreate /dev/sdft
        pvcreate /dev/sdfu
        pvcreate /dev/sdfv
        pvcreate /dev/sdfw
        pvcreate /dev/sdfx
        pvcreate /dev/sdfy
        pvcreate /dev/sdfz
        pvcreate /dev/sdg
        pvcreate /dev/sdga
        pvcreate /dev/sdgb
        pvcreate /dev/sdgc
        pvcreate /dev/sdgd
        pvcreate /dev/sdge
        pvcreate /dev/sdgf
        pvcreate /dev/sdgg
        pvcreate /dev/sdgh
        pvcreate /dev/sdgi
        pvcreate /dev/sdgj
        pvcreate /dev/sdgk
        pvcreate /dev/sdgl
        pvcreate /dev/sdgm
        pvcreate /dev/sdgn
        pvcreate /dev/sdgo
        pvcreate /dev/sdgp
        pvcreate /dev/sdgq
        pvcreate /dev/sdgr
        pvcreate /dev/sdgs
        pvcreate /dev/sdgt
        pvcreate /dev/sdgu
        pvcreate /dev/sdgv
        pvcreate /dev/sdgw
        pvcreate /dev/sdgx
        pvcreate /dev/sdgy
        pvcreate /dev/sdgz
        pvcreate /dev/sdh
        pvcreate /dev/sdha
        pvcreate /dev/sdhb
        pvcreate /dev/sdhc
        pvcreate /dev/sdhd
        pvcreate /dev/sdhe
        pvcreate /dev/sdhf
        pvcreate /dev/sdhg
        pvcreate /dev/sdhh
        pvcreate /dev/sdhi
        pvcreate /dev/sdhj
        pvcreate /dev/sdhk
        pvcreate /dev/sdhl
        pvcreate /dev/sdhm
        pvcreate /dev/sdhn
        pvcreate /dev/sdho
        pvcreate /dev/sdhp
        pvcreate /dev/sdhq
        pvcreate /dev/sdhr
        pvcreate /dev/sdhs
        pvcreate /dev/sdht
        pvcreate /dev/sdhu
        pvcreate /dev/sdhv
        pvcreate /dev/sdhw
        pvcreate /dev/sdhx
        pvcreate /dev/sdhy
        pvcreate /dev/sdhz
        pvcreate /dev/sdi
        pvcreate /dev/sdia
        pvcreate /dev/sdib
        pvcreate /dev/sdic
        pvcreate /dev/sdid
        pvcreate /dev/sdie
        pvcreate /dev/sdif
        pvcreate /dev/sdig
        pvcreate /dev/sdih
        pvcreate /dev/sdii
        pvcreate /dev/sdij
        pvcreate /dev/sdik
        pvcreate /dev/sdil
        pvcreate /dev/sdim
        pvcreate /dev/sdin
        pvcreate /dev/sdio
        pvcreate /dev/sdip
        pvcreate /dev/sdiq
        pvcreate /dev/sdir
        pvcreate /dev/sdis
        pvcreate /dev/sdit
        pvcreate /dev/sdiu
        pvcreate /dev/sdiv
        pvcreate /dev/sdiw
        pvcreate /dev/sdix
        pvcreate /dev/sdiy
        pvcreate /dev/sdiz
        pvcreate /dev/sdj
        pvcreate /dev/sdja
        pvcreate /dev/sdjb
        pvcreate /dev/sdjc
        pvcreate /dev/sdjd
        pvcreate /dev/sdje
        pvcreate /dev/sdjf
        pvcreate /dev/sdjg
        pvcreate /dev/sdjh
        pvcreate /dev/sdji
        pvcreate /dev/sdjj
        pvcreate /dev/sdjk
        pvcreate /dev/sdjl
        pvcreate /dev/sdjm
        pvcreate /dev/sdjn
        pvcreate /dev/sdjo
        pvcreate /dev/sdjp
        pvcreate /dev/sdjq
        pvcreate /dev/sdjr
        pvcreate /dev/sdjs
        pvcreate /dev/sdjt
        pvcreate /dev/sdju
        pvcreate /dev/sdjv
        pvcreate /dev/sdjw
        pvcreate /dev/sdjx
        pvcreate /dev/sdjy
        pvcreate /dev/sdjz
        pvcreate /dev/sdk
        pvcreate /dev/sdka
        pvcreate /dev/sdkb
        pvcreate /dev/sdkc
        pvcreate /dev/sdkd
        pvcreate /dev/sdke
        pvcreate /dev/sdkf
        pvcreate /dev/sdkg
        pvcreate /dev/sdkh
        pvcreate /dev/sdki
        pvcreate /dev/sdkj
        pvcreate /dev/sdkk
        pvcreate /dev/sdkl
        pvcreate /dev/sdkm
        pvcreate /dev/sdkn
        pvcreate /dev/sdko
        pvcreate /dev/sdkp
        pvcreate /dev/sdkq
        pvcreate /dev/sdkr
        pvcreate /dev/sdks
        pvcreate /dev/sdkt
        pvcreate /dev/sdku
        pvcreate /dev/sdkv
        pvcreate /dev/sdkw
        pvcreate /dev/sdkx
        pvcreate /dev/sdky
        pvcreate /dev/sdkz
        pvcreate /dev/sdl
        pvcreate /dev/sdla
        pvcreate /dev/sdlb
        pvcreate /dev/sdlc
        pvcreate /dev/sdld
        pvcreate /dev/sdle
        pvcreate /dev/sdlf
        pvcreate /dev/sdlg
        pvcreate /dev/sdlh
        pvcreate /dev/sdli
        pvcreate /dev/sdlj
        pvcreate /dev/sdlk
        pvcreate /dev/sdll
        pvcreate /dev/sdlm
        pvcreate /dev/sdln
        pvcreate /dev/sdlo
        pvcreate /dev/sdlp
        pvcreate /dev/sdlq
        pvcreate /dev/sdlr
        pvcreate /dev/sdls
        pvcreate /dev/sdlt
        pvcreate /dev/sdlu
        pvcreate /dev/sdlv
        pvcreate /dev/sdlw
        pvcreate /dev/sdlx
        pvcreate /dev/sdly
        pvcreate /dev/sdlz
        pvcreate /dev/sdm
        pvcreate /dev/sdma
        pvcreate /dev/sdmb
        pvcreate /dev/sdmc
        pvcreate /dev/sdmd
        pvcreate /dev/sdme
        pvcreate /dev/sdmf
```

```
pvcreate /dev/sdmg
pvcreate /dev/sdmh
pvcreate /dev/sdmi
pvcreate /dev/sdmj
pvcreate /dev/sdmk
pvcreate /dev/sdml
pvcreate /dev/sdmm
pvcreate /dev/sdmn
pvcreate /dev/sdmo
pvcreate /dev/sdmp
pvcreate /dev/sdmq
pvcreate /dev/sdmr
pvcreate /dev/sdms
pvcreate /dev/sdmt
pvcreate /dev/sdmu
pvcreate /dev/sdmv
pvcreate /dev/sdmw
pvcreate /dev/sdmx
pvcreate /dev/sdmy
pvcreate /dev/sdmz
pvcreate /dev/sdn
pvcreate /dev/sdna
pvcreate /dev/sdnb
pvcreate /dev/sdnc
pvcreate /dev/sdnd
pvcreate /dev/sdne
pvcreate /dev/sdnf
pvcreate /dev/sdng
pvcreate /dev/sdnh
pvcreate /dev/sdni
pvcreate /dev/sdnj
pvcreate /dev/sdnk
pvcreate /dev/sdnl
pvcreate /dev/sdnm
pvcreate /dev/sdnn
pvcreate /dev/sdno
pvcreate /dev/sdnp
pvcreate /dev/sdnq
pvcreate /dev/sdnr
pvcreate /dev/sdns
pvcreate /dev/sdnt
pvcreate /dev/sdnu
pvcreate /dev/sdo
pvcreate /dev/sdp
pvcreate /dev/sdq
pvcreate /dev/sdr
pvcreate /dev/sds
pvcreate /dev/sdt
pvcreate /dev/sdu
pvcreate /dev/sdv
pvcreate /dev/sdw
pvcreate /dev/sdx
pvcreate /dev/sdy
pvcreate /dev/sdz

# create vg
vgcreate vg11 /dev/sdac /dev/sdaf /dev/sdai /dev/sdal /dev/sdao /dev/sdar
    /dev/sdau /dev/sdax /dev/sdb /dev/sdba /dev/sdbd /dev/sdbg /dev/sdbj /dev/sdbm
    /dev/sdbp /dev/sdbs /dev/sdbv /dev/sdby /dev/sdcb /dev/sdce /dev/sdch
    /dev/sdck /dev/sdcn /dev/sdcq /dev/sdct /dev/sdcw /dev/sdcz /dev/sddc
    /dev/sddf /dev/sddi /dev/sddl /dev/sddo /dev/sddr /dev/sddu /dev/sddx /dev/sde
    /dev/sdea /dev/sded /dev/sdeg /dev/sdej /dev/sdem /dev/sdep /dev/sdes
    /dev/sdev  /dev/sdey  /dev/sdfb  /dev/sdfe  /dev/sdfh  /dev/sdfk  /dev/sdfn
    /dev/sdfq  /dev/sdft  /dev/sdfw  /dev/sdfz  /dev/sdgc  /dev/sdgf  /dev/sdgi
    /dev/sdgl /dev/sdgo /dev/sdgr /dev/sdgu /dev/sdgx /dev/sdh /dev/sdha /dev/sdhd
    /dev/sdhg  /dev/sdhj  /dev/sdhm  /dev/sdhp  /dev/sdhs  /dev/sdhv  /dev/sdhy
    /dev/sdib  /dev/sdie  /dev/sdih  /dev/sdik  /dev/sdin  /dev/sdiq  /dev/sdit
    /dev/sdiw  /dev/sdiz  /dev/sdjc  /dev/sdjf  /dev/sdji  /dev/sdjl  /dev/sdjo
    /dev/sdjr /dev/sdju /dev/sdjx /dev/sdk /dev/sdka /dev/sdkd /dev/sdkg /dev/sdkj
    /dev/sdkm  /dev/sdkp  /dev/sdks  /dev/sdkv  /dev/sdky  /dev/sdlb  /dev/sdle
    /dev/sdlh  /dev/sdlk  /dev/sdln  /dev/sdlq  /dev/sdlt  /dev/sdlw  /dev/sdlz
    /dev/sdmc  /dev/sdmf  /dev/sdmi  /dev/sdml  /dev/sdmo  /dev/sdmr  /dev/sdmu
    /dev/sdmx /dev/sdn /dev/sdna /dev/sdnd /dev/sdng /dev/sdnj /dev/sdnm /dev/sdnp
    /dev/sdns /dev/sdq /dev/sdt /dev/sdw /dev/sdz
vgcreate  vg22  /dev/sdaa  /dev/sdad  /dev/sdag  /dev/sdaj  /dev/sdam  /dev/sdap
    /dev/sdas  /dev/sdav  /dev/sday  /dev/sdbb  /dev/sdbe  /dev/sdbh  /dev/sdbk
    /dev/sdbn /dev/sdbq /dev/sdbt /dev/sdbw /dev/sdbz /dev/sdc /dev/sdcc /dev/sdcf
    /dev/sdci  /dev/sdcl  /dev/sdco  /dev/sdcr  /dev/sdcu  /dev/sdcx  /dev/sdda
    /dev/sddd  /dev/sddg  /dev/sddj  /dev/sddm  /dev/sddp  /dev/sdds  /dev/sddv
    /dev/sddy  /dev/sdeb  /dev/sdee  /dev/sdeh  /dev/sdek  /dev/sden  /dev/sdeq
    /dev/sdet /dev/sdew /dev/sdez /dev/sdf /dev/sdfc /dev/sdff /dev/sdfi /dev/sdfl
    /dev/sdfo  /dev/sdfr  /dev/sdfu  /dev/sdfx  /dev/sdga  /dev/sdgd  /dev/sdgg
    /dev/sdgj  /dev/sdgm  /dev/sdgp  /dev/sdgs  /dev/sdgv  /dev/sdgy  /dev/sdhb
    /dev/sdhe  /dev/sdhh  /dev/sdhk  /dev/sdhn  /dev/sdhq  /dev/sdht  /dev/sdhw
    /dev/sdhz /dev/sdi /dev/sdic /dev/sdif /dev/sdii /dev/sdil /dev/sdio /dev/sdir
    /dev/sdiu  /dev/sdix  /dev/sdja  /dev/sdjd  /dev/sdjg  /dev/sdjj  /dev/sdjm
    /dev/sdjp  /dev/sdjs  /dev/sdjv  /dev/sdjy  /dev/sdkb  /dev/sdke  /dev/sdkh
    /dev/sdkk /dev/sdkn /dev/sdkq /dev/sdkt /dev/sdkw /dev/sdkz /dev/sdl /dev/sdlc
    /dev/sdlf  /dev/sdli  /dev/sdll  /dev/sdlo  /dev/sdlr  /dev/sdlu  /dev/sdlx
    /dev/sdma  /dev/sdmd  /dev/sdmg  /dev/sdmj  /dev/sdmm  /dev/sdmp  /dev/sdms
    /dev/sdmv  /dev/sdmy  /dev/sdnb  /dev/sdne  /dev/sdnh  /dev/sdnk  /dev/sdnn
    /dev/sdnq /dev/sdnt /dev/sdo /dev/sdr /dev/sdu /dev/sdx
vgcreate  vg33  /dev/sdab  /dev/sdae  /dev/sdah  /dev/sdak  /dev/sdan  /dev/sdaq
    /dev/sdat  /dev/sdaw  /dev/sdaz  /dev/sdbc  /dev/sdbf  /dev/sdbi  /dev/sdbl
    /dev/sdbo  /dev/sdbr  /dev/sdbu  /dev/sdbx  /dev/sdca  /dev/sdcd  /dev/sdcg
    /dev/sdcj /dev/sdcm /dev/sdcp /dev/sdcs /dev/sdcv /dev/sdcy /dev/sdd /dev/sddb
    /dev/sdde  /dev/sddh  /dev/sddk  /dev/sddn  /dev/sddq  /dev/sddt  /dev/sddw
    /dev/sddz  /dev/sdec  /dev/sdef  /dev/sdei  /dev/sdel  /dev/sdeo  /dev/sder
    /dev/sdeu  /dev/sdex  /dev/sdfa  /dev/sdfd  /dev/sdfg  /dev/sdfj  /dev/sdfm
    /dev/sdfp /dev/sdfs /dev/sdfv /dev/sdfy /dev/sdg /dev/sdgb /dev/sdge /dev/sdgh
    /dev/sdgk  /dev/sdgn  /dev/sdgq  /dev/sdgt  /dev/sdgw  /dev/sdgz  /dev/sdhc
    /dev/sdhf  /dev/sdhi  /dev/sdhl  /dev/sdho  /dev/sdhr  /dev/sdhu  /dev/sdhx
    /dev/sdia  /dev/sdid  /dev/sdig  /dev/sdij  /dev/sdim  /dev/sdip  /dev/sdis
    /dev/sdiv /dev/sdiy /dev/sdj /dev/sdjb /dev/sdje /dev/sdjh /dev/sdjk /dev/sdjn
    /dev/sdjq  /dev/sdjt  /dev/sdjw  /dev/sdjz  /dev/sdkc  /dev/sdkf  /dev/sdki
    /dev/sdkl  /dev/sdko  /dev/sdkr  /dev/sdku  /dev/sdkx  /dev/sdla  /dev/sdld
    /dev/sdlg /dev/sdlj /dev/sdlm /dev/sdlp /dev/sdls /dev/sdlv /dev/sdly /dev/sdm
    /dev/sdmb  /dev/sdme  /dev/sdmh  /dev/sdmk  /dev/sdmn  /dev/sdmq  /dev/sdmt
    /dev/sdmw  /dev/sdmz  /dev/sdnc  /dev/sdnf  /dev/sdni  /dev/sdnl  /dev/sdno
    /dev/sdnr /dev/sdnu /dev/sdp /dev/sds /dev/sdv /dev/sdy

#create lv
lvcreate -n asu11  -i 128 -I 1024 -C y -L 21186g vg11
lvcreate -n asu110 -i 128 -I 1024 -C y -L 21186g vg22
lvcreate -n asu120 -i 128 -I 1024 -C y -L 21186g vg33

lvcreate -n asu21  -i 128 -I 1024 -C y -L 21186g vg11
lvcreate -n asu210 -i 128 -I 1024 -C y -L 21186g vg22
lvcreate -n asu220 -i 128 -I 1024 -C y -L 21186g vg33

lvcreate -n asu31  -i 128 -I 1024 -C y -L 4708g vg11
lvcreate -n asu32  -i 128 -I 1024 -C y -L 4708g vg22
lvcreate -n asu33  -i 128 -I 1024 -C y -L 4708g vg33
```

### ***lv_scan.sh***

```
#!/bin/bash

echo '---------------pvscan-----------------'
pvscan
echo '---------------vgscan-----------------'
vgscan
echo '---------------lvscan-----------------'
lvscan

lvchange -ay /dev/vg11/asu11
lvchange -ay /dev/vg11/asu21
lvchange -ay /dev/vg11/asu31
lvchange -ay /dev/vg22/asu110
lvchange -ay /dev/vg22/asu210
lvchange -ay /dev/vg22/asu32
lvchange -ay /dev/vg33/asu120
lvchange -ay /dev/vg33/asu220
lvchange -ay /dev/vg33/asu33
```

# APPENDIX E: CONFIGURATION INVENTORY

An inventory of the Tested Storage Configuration was collected during the execution of the scripts ***ms5580g2_test_phase1.sh*** and ***ms5580g2_test_phase3.sh***. by invoking the script ***profile_ms5580g2.sh***. The following log files were generated:

| | |
|---|---|
| • ***profile_start_ms5580g2.txt*** | List of configured storage before the INIT Phase. |
| • ***volume_listing_start.txt*** | List of configured volumes before the INIT Phase. |
| • ***profile_end_ms5580g2.txt*** | List of configured storage after TSC restart. |
| • ***volume_listing_end.txt*** | List of configured volumes after TSC restart. |

The above log files are included in the Supporting Files (see Appendix A).

The profile collection script described above is included in the Supporting Files (see Appendix A) and listed below.

### ***profile_ms5580g2.sh***

```
#!/bin/sh

var=$1
node=("NodeA" "NodeB" "NodeC" "NodeD" "NodeE" "NodeF" "NodeG" "NodeH")
iplist1=("10.2.2.101"   "10.2.2.103"   "10.2.2.105"   "10.2.2.107"   "10.2.2.109"
   "10.2.2.111" "10.2.2.113" "10.2.2.115" )
iplist2=("10.2.2.102"   "10.2.2.104"   "10.2.2.106"   "10.2.2.108"   "10.2.2.110"
   "10.2.2.112" "10.2.2.114" "10.2.2.116" )

# configuration
echo "---- Start Timestamp: `date` ----" > profile_$1_ms5580g2.txt
echo "<<< Configuration Information >>>" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt
echo "1.System Parameter" >> profile_$1_ms5580g2.txt
echo "2.Controller Info" >> profile_$1_ms5580g2.txt
echo "3.Frontend/Backend Connection" >> profile_$1_ms5580g2.txt
echo "4.DSU List/Disk List" >> profile_$1_ms5580g2.txt
echo "5.DSU Details" >> profile_$1_ms5580g2.txt
echo "6.Disk Details" >> profile_$1_ms5580g2.txt
echo "7.RAID Details" >> profile_$1_ms5580g2.txt
echo "8.LUN List" >> profile_$1_ms5580g2.txt
echo "9.LUN Details" >> profile_$1_ms5580g2.txt
echo "10.FC Module" >> profile_$1_ms5580g2.txt
echo "11.Email Notification" >> profile_$1_ms5580g2.txt
echo "12.SNMP Notification" >> profile_$1_ms5580g2.txt
echo "13.HotCache" >> profile_$1_ms5580g2.txt
echo "14.Private Network" >> profile_$1_ms5580g2.txt

# 1.system parameter
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 1.System Parameter *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

ms-cli ALL "HWManager -g -all" >> profile_$1_ms5580g2.txt

# 2.controller firmware
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 2.Controller Info *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

echo "== Controller Version =======================" >> profile_$1_ms5580g2.txt
ms-cli ALL "system mgt getversion" >> profile_$1_ms5580g2.txt

echo "== Cache Info ===============================" >> profile_$1_ms5580g2.txt

# [modify]
for ((h=0;h<8;h++))
do
   echo "${node[$h]} SP1 Cache:" >> profile_$1_ms5580g2.txt
   ssh  root@${iplist1[$h]}  "sh  /home/driver/chk_mem/localmeminfo  |sed  's/^/
   Cache Size:/' |sed 's/$/GB/'" >> profile_$1_ms5580g2.txt
   echo "${node[$h]} SP2 Cache:" >> profile_$1_ms5580g2.txt
   ssh  root@${iplist2[$h]}  "sh  /home/driver/chk_mem/localmeminfo  |sed  's/^/
   Cache Size:/' |sed 's/$/GB/'" >> profile_$1_ms5580g2.txt
done
echo "" >> profile_$1_ms5580g2.txt

# 3.Frontend/Backend Connection
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 3.Frontend/Backend Connection *****" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

echo "== Frontend Connection ======================" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

ms-cli ALL "client target queryhbalist" |sed "s/velocity   : 16/velocity   : 16
   Gbps/g" -  >> profile_$1_ms5580g2.txt
# [modify]
echo "== Backend Connection ======================" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt
for ((i=0;i<8;i++))
do
    echo "${node[$i]} SP1:" >> profile_$1_ms5580g2.txt
    ms-cli ${node[$i]} "system sas getportlist -c 1:6" >> profile_$1_ms5580g2.txt
    echo "" >> profile_$1_ms5580g2.txt

    ms-cli ${node[$i]} "system sas getportlist -c 1:9" >> profile_$1_ms5580g2.txt
    echo "" >> profile_$1_ms5580g2.txt

    ms-cli    ${node[$i]}    "system    sas    getportlist    -c    1:10"    >>
   profile_$1_ms5580g2.txt
    echo "" >> profile_$1_ms5580g2.txt

    echo "${node[$i]} SP2:" >> profile_$1_ms5580g2.txt
    ms-cli ${node[$i]} "system sas getportlist -c 2:6" >> profile_$1_ms5580g2.txt
    echo "" >> profile_$1_ms5580g2.txt

    ms-cli ${node[$i]} "system sas getportlist -c 2:9" >> profile_$1_ms5580g2.txt
    echo "" >> profile_$1_ms5580g2.txt
```

```
     ms-cli ${node[$i]} "system sas getportlist -c 2:10" >>
    profile_$1_ms5580g2.txt
     echo "" >> profile_$1_ms5580g2.txt

     echo "======================" >> profile_$1_ms5580g2.txt
done

# 4.dsu/disk List
echo "" >> profile_$1_ms5580g2.txt
echo "**********************************************" >> profile_$1_ms5580g2.txt
echo "********* 4.DSU/Disk List *********" >> profile_$1_ms5580g2.txt
echo "**********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

echo "== DSU List ==================================" >> profile_$1_ms5580g2.txt
ms-cli ALL "dsu mgt getlist" >> profile_$1_ms5580g2.txt

echo "== Disk List =================================" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt
# [modify]
for ((j=0;j<8;j++))
do
    echo  "${node[$j]}:" >> profile_$1_ms5580g2.txt
    for dsu in `ms-cli ${node[$j]} "dsu mgt getlist"| grep Name | awk 'BEGIN {FS="
    "}{print$2}'`
    do
        echo "--$dsu ----------------------------------" >> profile_$1_ms5580g2.txt

        ms-cli ${node[$j]} "disk mgt getlist -d $dsu"  >> profile_$1_ms5580g2.txt
    done
done

#5.DSU Details
echo "" >> profile_$1_ms5580g2.txt
echo "**********************************************" >> profile_$1_ms5580g2.txt
echo "********* 5.DSU Details *********" >> profile_$1_ms5580g2.txt
echo "**********************************************" >> profile_$1_ms5580g2.txt

echo "" >> profile_$1_ms5580g2.txt
# [modify]
for ((q=0;q<8;q++))
do
    echo  "${node[$q]}:" >> profile_$1_ms5580g2.txt
    echo "--DSU battery ----------------------------------" >>
    profile_$1_ms5580g2.txt
    ms-cli ${node[$q]} "dsu battery query -d 10:1:1"  >> profile_$1_ms5580g2.txt
    for dsu in `ms-cli ${node[$q]} "dsu mgt getlist"| grep Name | awk 'BEGIN {FS="
    "}{print$2}'`
    do
        echo "--$dsu ----------------------------------" >> profile_$1_ms5580g2.txt
        ms-cli ${node[$q]} "dsu mgt query -d $dsu"  >> profile_$1_ms5580g2.txt
    done
done

# 6.disk details
echo "" >> profile_$1_ms5580g2.txt
echo "**********************************************" >> profile_$1_ms5580g2.txt
echo "********* 6.Disk Details *********" >> profile_$1_ms5580g2.txt
echo "**********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt
# [modify]
```

```
for ((k=0;k<8;k++))
do
    echo "${node[$k]}:" >> profile_$1_ms5580g2.txt
    for dsu in `ms-cli ${node[$k]} "dsu mgt getlist"| grep Name | awk 'BEGIN {FS="
    "}{print$2}'`
    do
         echo "==$dsu ==================================" >>
    profile_$1_ms5580g2.txt
         for disk in `ms-cli ${node[$k]} "disk mgt getlist -d $dsu" | grep Name
    |cut -d : -f 5|sort -n`
         do
                ms-cli ${node[$k]} "disk mgt query -d $dsu:$disk" >>
    profile_$1_ms5580g2.txt
         done
    done
    echo "" >> profile_$1_ms5580g2.txt
done

# 7.raid details
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 7.RAID Details *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

# [modify]
for ((l=0;l<8;l++))
do
    echo "${node[$l]}:" >> profile_$1_ms5580g2.txt
    for pool in `ms-cli ${node[$l]} "pool mgt getlist"| grep Name |awk 'BEGIN
    {FS=" "}{print$2}'`
    do
       echo "== $pool ====================================" >>
    profile_$1_ms5580g2.txt
       for raid in `ms-cli ${node[$l]} "raid mgt getlist -p $pool"| grep Name|awk
    'BEGIN {FS=" "}{print$2}'`
       do
              echo "-- $raid -----------------------------------" >>
    profile_$1_ms5580g2.txt
              ms-cli ${node[$l]} "raid mgt query -n $raid" >>
    profile_$1_ms5580g2.txt
              echo "" >> profile_$1_ms5580g2.txt
              done
    done
    echo "" >> profile_$1_ms5580g2.txt
done

# 8.lun list
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "*********     8.LUN List *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt
for ((n=0;n<8;n++))
do
    echo "${node[$n]}:" >> profile_$1_ms5580g2.txt
    for pool in `ms-cli ${node[$n]} "pool mgt getlist"| grep Name |awk 'BEGIN
    {FS=" "}{print$2}'`
    do
       echo "== $pool ====================================" >>
    profile_$1_ms5580g2.txt
       ms-cli ${node[$n]} "lun mgt getlist -p $pool" >> profile_$1_ms5580g2.txt
```

```
   done
done
echo "" >> profile_$1_ms5580g2.txt

# 9.lun details
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 9.LUN Details *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt
for ((m=0;m<8;m++))
do
   echo "${node[$m]}:" >> profile_$1_ms5580g2.txt
   for pool in `ms-cli ${node[$m]} "pool mgt getlist"| grep Name |awk 'BEGIN
   {FS=" "}{print$2}'`
   do
      echo "== $pool ====================================" >>
   profile_$1_ms5580g2.txt
      for lun in `ms-cli ${node[$m]} "lun mgt getlist -p $pool"| grep Name|awk
   'BEGIN {FS=" "}{print$2}'`
      do
             echo "-- $lun ----------------------------------" >>
   profile_$1_ms5580g2.txt
             ms-cli ${node[$m]} "lun mgt query -n $lun" >>
   profile_$1_ms5580g2.txt
             echo "" >> profile_$1_ms5580g2.txt
             done
   done
   echo "" >> profile_$1_ms5580g2.txt
done

# 10.fc module
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 10.FC Module *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

ms-cli ALL "client target queryportlist"  >> profile_$1_ms5580g2.txt

# 11.email notification
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 11.Email Notification *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

ms-cli ALL "alarm email query" >> profile_$1_ms5580g2.txt

# 12.snmp notification
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 12.SNMP Notification *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

ms-cli ALL "alarm snmp query" >> profile_$1_ms5580g2.txt

# 13.hotcache
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 13.HotCache *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt

ms-cli ALL "hotcache mgt query" >> profile_$1_ms5580g2.txt

#14.Private Network information
echo "" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "********* 14.Private Network *********" >> profile_$1_ms5580g2.txt
echo "*********************************************" >> profile_$1_ms5580g2.txt
echo "" >> profile_$1_ms5580g2.txt
for ((p=0;p<8;p++))
do
   echo "" >> profile_$1_ms5580g2.txt
   echo "---- ${node[$p]} SP1 Private Network ----"   >> profile_$1_ms5580g2.txt
   ssh root@${iplist1[$p]} ifconfig eth_1_5_1 |grep broadcast|awk -F ' ' '{print
   $1,$2}'|awk  -F  '  '  '{print  "'${node[$p]}'_SP1  eth_1_5_1  addr:",$2}'  >>
   profile_$1_ms5580g2.txt
   ssh root@${iplist1[$p]} ethtool eth_1_5_1 |grep -i speed |awk -F ':' '{print
   "'${node[$p]}'_SP1 eth_1_5_1 speed:",$2}'>> profile_$1_ms5580g2.txt
   ssh root@${iplist1[$p]} ifconfig eth_1_5_2 |grep broadcast|awk -F ' ' '{print
   $1,$2}'|awk  -F  '  '  '{print  "'${node[$p]}'_SP1  eth_1_5_2  addr:",$2}'  >>
   profile_$1_ms5580g2.txt
   ssh root@${iplist1[$p]} ethtool eth_1_5_2 |grep -i speed |awk -F ':' '{print
   "'${node[$p]}'_SP1 eth_1_5_2 speed:",$2}'>> profile_$1_ms5580g2.txt
   echo "" >> profile_$1_ms5580g2.txt

   echo "" >> profile_$1_ms5580g2.txt
   echo "---- ${node[$p]} SP2 Private Network ----"   >> profile_$1_ms5580g2.txt
   ssh root@${iplist2[$p]} ifconfig eth_2_5_1 |grep broadcast|awk -F ' ' '{print
   $1,$2}'|awk  -F  '  '  '{print  "'${node[$p]}'_SP2  eth_2_5_1  addr:",$2}'  >>
   profile_$1_ms5580g2.txt
   ssh root@${iplist2[$p]} ethtool eth_2_5_1 |grep -i speed |awk -F ':' '{print
   "'${node[$p]}'_SP2 eth_2_5_1 speed:",$2}'>> profile_$1_ms5580g2.txt
   ssh root@${iplist2[$p]} ifconfig eth_2_5_2 |grep broadcast|awk -F ' ' '{print
   $1,$2}'|awk  -F  '  '  '{print  "'${node[$p]}'_SP2  eth_2_5_2  addr:",$2}'  >>
   profile_$1_ms5580g2.txt
   ssh root@${iplist2[$p]} ethtool eth_2_5_2 |grep -i speed |awk -F ':' '{print
   "'${node[$p]}'_SP2 eth_2_5_2 speed:",$2}'>> profile_$1_ms5580g2.txt
   echo "" >> profile_$1_ms5580g2.txt
done
echo "---- End Timestamp: `date` ----" >> profile_$1_ms5580g2.txt
```

# APPENDIX F: WORKLOAD GENERATOR

The host parameters for the SPC-1 workload generator were defined using the script ***20.HST***.

The ASUs accessed by the SPC-1 workload generator are defined using the script ***slave_asu.asu***.

The initial test phases of the benchmark are executed using the scripts ***ms5580g2_test_phase1.sh***. The PERSIST_1 phase is invoked by the script ***ms5580g2_test_phase2.sh***. This is followed by a full shutdown and power down of the TSC. Once the TSC has been restarted, the PERSIST_2 test phase is executed using the script ***ms5580g2_test_phase3.sh***.

The above scripts are included in the Supporting Files (see Appendix A) and listed below.

### *20.HST*

```
-- SMALL_HOST definition
LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost
PORT=7001
WINDOWS=N
```

```
WEIGHT=1
STORAGE=slave_asu.asu
HOST=localhost



LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.158
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.158
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.158
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.158
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.158
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.158
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.158

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.171
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.171
```

```
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.171
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.171
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.171
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.171
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.171


LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.154
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.154
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.154
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.154
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.154
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.154
```

```
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.154

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.155
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.155
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.155
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.155
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.155
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.155
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.155

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.156
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.156
```

```
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.156
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.156
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.156
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.156
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.156


LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.159
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.159
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.159
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.159
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.159
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.159
```

```
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.159

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.172
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.172
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.172
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.172
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.172
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.172
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.172

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.161
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.161
```

```
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.161
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.161
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.161
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.161
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.161

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.162
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.162
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.162
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.162
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.162
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.162
PORT=7001
```

```
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.162

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.163
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.163
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.163
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.163
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.163
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.163
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.163

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.164
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.164
PORT=3001
```

```
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.164
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.164
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.164
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.164
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.164

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.177
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.177
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.177
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.177
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.177
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.177
PORT=7001
WINDOWS=N
```

```
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.177

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.166
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.166
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.166
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.166
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.166
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.166
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.166

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.167
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.167
PORT=3001
WINDOWS=N
```

```
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.167
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.167
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.167
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.167
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.167

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.168
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.168
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.168
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.168
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.168
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.168
PORT=7001
WINDOWS=N
WEIGHT=1
```

```
STORAGE=slave_asu.asu
HOST=10.2.2.168

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.169
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.169
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.169
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.169
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.169
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.169
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.169

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.178
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.178
PORT=3001
WINDOWS=N
WEIGHT=1
```

```
STORAGE=slave_asu.asu
HOST=10.2.2.178
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.178
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.178
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.178
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.178



LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.173
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.173
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.173
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.173
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.173
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.173
PORT=7001
WINDOWS=N
```

```
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.173

LOGIN=root
PASSWORD=passwd
CONFIG=/root/v302
EXEC=Supported/RHEL/7_2/spc1_optimized_v3.0.2
OUTPUT=/root/v302/Output/Output
PORT=1001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.174
PORT=2001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.174
PORT=3001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.174
PORT=4001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.174
PORT=5001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.174
PORT=6001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.174
PORT=7001
WINDOWS=N
WEIGHT=1
STORAGE=slave_asu.asu
HOST=10.2.2.174
```

### ***slave_asu.asu***

```
ASU=1
OFFSET=0
SIZE=0
DEVICE=/dev/vg11/asu11
DEVICE=/dev/vg22/asu110
DEVICE=/dev/vg33/asu120

--
ASU=2
OFFSET=0
SIZE=0
DEVICE=/dev/vg11/asu21
DEVICE=/dev/vg22/asu210
DEVICE=/dev/vg33/asu220
```

```
--
ASU=3
OFFSET=0
SIZE=0
DEVICE=/dev/vg11/asu31
DEVICE=/dev/vg22/asu32
DEVICE=/dev/vg33/asu33
```

## ***ms5580g2_test_phase1.sh***

```
#!/bin/bash

date

echo "Collect_MS5580G2_info"
ssh root@10.2.2.188 >/dev/null 2>&1 <<eeooff
/root/profile_ms5580g2.sh start
exit
eeooff
echo "---------------"

echo "Collect_volume_info"
/root/v302/sh/volume_list.sh start
echo "---------------"


date

echo "Init tests"
/root/v302/Supported/RHEL/7_2/spc1_optimized_v3.0.2 -run SPC1_INIT  -iops 16000
   -storage slave_asu.asu -output /root/5580g2/0624 -master 20.HST
echo "---------------"

echo "Verify"
/root/v302/Supported/RHEL/7_2/spc1_optimized_v3.0.2 -run SPC1_VERIFY  -iops 100
   -storage slave_asu.asu -output /root/5580g2/0624
echo "---------------"

echo "Start metrics test"
/root/v302/Supported/RHEL/7_2/spc1_optimized_v3.0.2  -run    SPC1_METRICS  -iops
   6100000 -storage slave_asu.asu -output /root/5580g2/0624 -master 20.HST
echo "metrics test over"
echo "---------------"

echo "Start test verify"
/root/v302/Supported/RHEL/7_2/spc1_optimized_v3.0.2 -run SPC1_VERIFY  -iops 100
   -storage slave_asu.asu -output /root/5580g2/0624
echo "---------------"

echo "test_phase1 test over"

date
```

## ***ms5580g2_test_phase2.sh***

```
#!/bin/bash

date

echo "Start test persist1"
```

```
/root/v302/Supported/RHEL/7_2/spc1_optimized_v3.0.2  -run  SPC1_PERSIST_1   -iops
   1525000  -storage slave_asu.asu -output /root/5580g2/0624 -master 20.HST
echo "---------------"
echo "test_persist1 test over"

date
```

### ms5580g2_test_phase3.sh

```
#!/bin/bash

date

echo "Collect_MS5580G2_info"
ssh root@10.2.2.188 >/dev/null 2>&1 <<eeooff
/root/profile_ms5580g2.sh end
exit
eeooff
echo "---------------"

echo "Collect_volume_info"
/root/v302/sh/volume_list.sh end
echo "---------------"

echo "start test persist2"
/root/v302/Supported/RHEL/7_2/spc1_optimized_v3.0.2  -run  SPC1_PERSIST_2   -iops
   1525000  -storage slave_asu.asu -output /root/5580g2/0624 -master 20.HST
echo "---------------"
echo "test_persist2 test over"

date
```