SNIA™
Green Storage Initiative

User Guide

for the *SNIA Emerald™ Power Efficiency Measurement Specification*

Version V.1.0

SNIA Emerald™

October 7, 2011

About the SNIA

The Storage Networking Industry Association (SNIA) is a not–for–profit global organization, made up of some 400 member companies spanning virtually the entire storage industry. SNIA's mission is to lead the storage industry worldwide in developing and promoting standards, technologies, and educational services to empower organizations in the management of information. To this end, the SNIA is uniquely committed to delivering standards, education, and services that will propel open storage networking solutions into the broader market. For additional information, visit the SNIA web site at www.snia.org.

About the SNIA Green Storage Initiative

SNIA's Green Storage Initiative (GSI) is dedicated to advancing energy efficiency and conservation in all networked storage technologies in an effort to minimize the environmental impact of data storage operations. SNIA's Green Storage activities take place in two separate working bodies, the SNIA Green Storage Technical Working Group (TWG) and the Green Storage Initiative. The TWG is focused on developing test metrics by which energy consumption and efficiency can be measured. The Green Storage Initiative is focused on creating and publicizing best practices for energy efficient storage networking, educating the IT community, and promoting storage–centric applications that reduce storage footprint and associated power requirements. The members of the GSI, as of October 2011, include EMC, HDS, HP, IBM, LSI, Oracle, NetApp, Q-logic, Seagate, VMware, and Xyratex. For more information about SNIA's Green Storage Initiative, visit www.snia.org/green

About the SNIA Emerald™ Program

The SNIA Emerald™ Program provides a publicly accessible repository of vendor storage system power efficiency measurement and related data. The measurement data is generated through the use of well-defined testing procedures prescribed in the *SNIA Emerald™ Power Efficiency Measurement Specification*. This data quantifies storage system power efficiency for several types of workloads.

The Emerald™ Program repository maintains downloadable test data reports for each vendor opting to participate in the SNIA Emerald™ Program. The report includes product measurement data as well as other information related to system power efficiency including system configuration details such as storage device types, RAS features and their configuration, and power supply types. The test data reports can help IT professionals make storage platform selections as part of an overall Green IT and Sustainability objective. The program is open to the industry at large, including non-members of SNIA. For additional information, visit the SNIA Emerald™ Program website at http://www.sniaemerald.com.

The information contained in this publication is subject to change without notice. This guide represents a "best effort" attempt by the SNIA Green Storage Technical Working Group to provide guidance to those implementing the *SNIA Emerald™ Power Efficiency Measurement Specification Version 1.0*, and the guide may be updated or replaced at any time. The SNIA shall not be liable for errors contained herein.

Suggestions for revisions to this guide and questions concerning implementation of the *SNIA Emerald™ Power Efficiency Measurement Specification Version 1.0* can be directed (via email) to greentwg-chair@snia.org.

Contents

1 Introduction

This document is intended to be used with the *SNIA Emerald™ Power Efficiency Measurement Specification* (referred to within this document as simply the "Measurement Specification"), developed as part of the SNIA Emerald™ Program. The SNIA Emerald™ Program was set up to provide a consistent and credible way for storage system vendors to demonstrate product power efficiency. In order to facilitate this, the SNIA Green Storage Initiative (GSI) and Green Technical Working Group (TWG) has developed a simple and standard method to measure storage system power efficiency along with a mechanism via the SNIA Emerald™ Program to store results accessible for information and comparison. This method is not intended to demonstrate the true power efficiency of a storage system at a customer site, but instead provide a general understanding of expected power efficiency while in use.

1.1 Audience

The target audience of this document includes the individual(s) planning for and implementing the Measurement Specification. This document provides advice on how to:

- develop a product family definition beyond the Measurement Specification taxonomy
- determine appropriate test configurations
- set up and complete the measurement sequence
- use a sample Vdbench script for workload generation
- avoid problems
- submit results

1.2 References

This guide is designed to be used with this document:

- *SNIA Emerald™ Power Efficiency Measurement Specification Version 1.0*. Copyright 2011 SNIA. Referred to in this document as simply the "Measurement Specification."
- *SNIA Emerald™ Test Data Report*, available on the Emerald™ website

Additional information about the SNIA Emerald™ Program and associated *SNIA Emerald™ Power Efficiency Measurement Specification* is available at these websites:

- http://www.sniaemerald.com, the SNIA Emerald™ website
- http://www.snia.org/forums/green, the SNIA GSI website

2 Scope

2.1 General

SNIA developed the *Measurement Specification* so that vendors and consumers of storage systems would have a reliable and consistent way to observe and evaluate storage power efficiency among different storage solutions and systems. Metrics of IO/s/Watt, MiB/s/Watt and GB/Watt were agreed upon by the member companies of SNIA to be proxies for these comparisons. The IO/s/Watt and MiB/s/Watt metrics are active metrics, which represent the power efficiency of moving the data to and from a host. The GB/Watt metric represents the power efficiency of storing and protecting the data on the storage system.

2.2 Taxonomy and product-family Structures

Due to the wide spectrum of storage-oriented products, a taxonomy structure was created. The taxonomy presently has categories of storage including Online, Near Online, Removable Media Library, and Virtual Media Library. Each category is then divided into classifications based on different characteristics. Since each of these categories provides its own unique testing criteria, it is critical for valid measurement to correctly identify the category and classification. The taxonomy is defined in the Measurement Specification.

While the taxonomy is useful in differentiating storage systems, vendors may still have a wide range of products within a single category and classification. To further aid in test configuration develop, this document also includes advice on differentiating products and families along with suggestions on limiting test configurations.

2.3 Basic Testing Criteria

The testing criteria for all storage solutions are basically the same, as follows:

1. The System Under Test (SUT) is run through a SUT Conditioning Test to get it into a known and stable state.
2. The SUT Conditioning Test is followed by a defined series of Active Test phases that collect data for the active metrics, each with a method to assure stability of each metric value.
3. The Active Test is followed by the so-called *Ready Idle* Test that collects data for the capacity metric.
4. Lastly, the Capacity Optimization Test phases are executed which demonstrate the storage system's ability to perform defined capacity optimization methods.

For each of the categories of storage, there will be different workloads and run times depending on the category characteristics.

2.4 Using this Guide

This document provides guidance on implementing the Measurement Specification. Areas covered include:

- Defining a product family. (Section 3)
- Finding the most appropriate system configuration(s) to use for testing. (Section 4)
- Setting up and performing the power efficiency measurement. (Section 5)
- Determining the metric data to be collected and generated for submission. (Section 6)
- Additional advisory notes. (Section 7)

3 Defining the Product Family

For storage system vendors, selecting which configuration(s) to test per the Measurement Specification can be a challenging task. Even the smallest systems may have a significant number of configuration options, and each configuration test requires significant execution effort. Similarly, customers wish to have a clear and easily interpreted yet comprehensive set of results. This section defines the concept of products and product families as an aid to selecting a reduced—yet comprehensive—number of test configurations.

3.1 Overview and Goal

Several aspects come into play when considering which storage system configurations to test for energy efficiency. In particular, customers want a clear and reasonably complete method to gauge and evaluate efficiencies of particular product candidates. At the same time, storage system vendors wish to minimize efficiency test variations for lowest cost and widest coverage from a potentially large set of product configurations and use cases.

The vendor-side goal of the product/family definition is to provide a method to define the minimum number of reasonable proxy test configurations, each with the widest possible applicable and acceptable scope. A key aspect is to minimize variables to the greatest extent possible.

The customer-side goal is to present customers with useable/comparable product efficiency results via the SNIA Emerald™ Program.

3.2 Product/Family Definition

The Measurement Specification includes a taxonomy that divides storage products into relatively coarse categories and classifications. Once a product is aligned with a taxonomy category/classification, the question remains of how the product and its possible variations are actually tested per the goals listed in Section 3.1.

The concept of products and product families is presented here to help further define actual storage system test configurations. While vendors vary in the manner in which they define and sell their products, this product/family approach is believed to be generally applicable.

A product has different aspects depending on the observer. To the customer, a product represents a particular purchased and installed configuration. To the vendor, it can be a base (possibly entry) unit with a bounded set of configuration options. A product family can also have many interpretations.

In this document, a product and product family are arbitrarily defined as follows:

- A *product* represents a fundamental performance capability space that separates it from any other potentially related products.
- A *product family* represents the full *range* space of configuration variables and options for a particular *product.*

The terms *family* and *range* are used interchangeably within this section and may include such aspects as number and type of storage device (spinning or solid state drive), availability levels, etc.

Figure 1 depicts a simplified but possible product/family (range) differentiation depiction. Note that this figure could apply to any storage system architecture, e.g., monolithic, scale-up, or scale-out (with *scale-up* generally referring to a system of a limited number of controllers with scalable back-end storage and *scale-out* referring to systems constructed of interconnected compute-storage nodes, real or virtual).

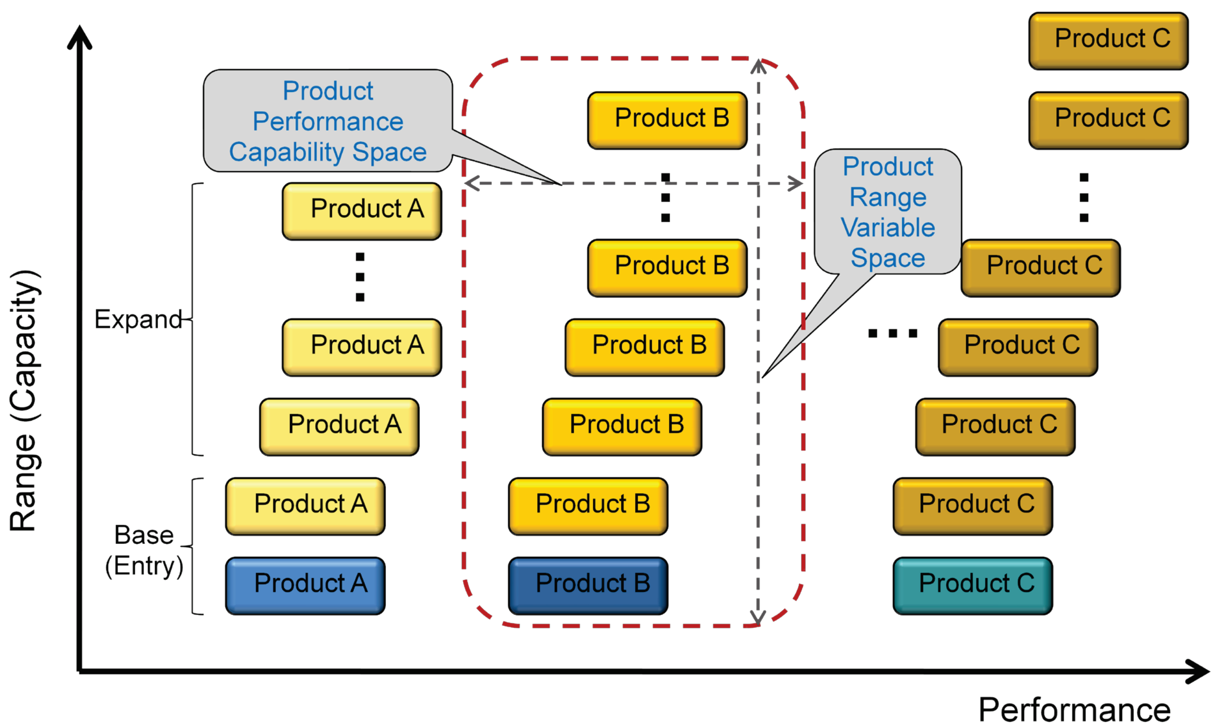

Figure 1 Possible Product/Family (Range) Depiction

The range variable space shown in Figure 1 focuses on capacity but can also imply storage device type or other variables. Note that some products illustrated may increase performance with added capacity and some may not, e.g., roll off, after a certain capacity/variable point.

3.3 Range Variable Discussion

As noted in the product/family discussion in Section 3.2, a full family range encompasses many variables both in number and type, of which SNIA has defined at least 25. The following list highlights those considered to have the highest potential energy consumption impact:

- Controller or related compute element - Typically defines the product performance space.
- Cache functions - May not always be aligned with the controller but not considered part of the user addressable space.
- Number and type of persistent storage device - Defines the user addressable space consisting of hard disk drives (HDD), solid state drives (SSD), etc.
- RAS items - Energy consuming functions necessary to meet requirements for reliability, availability, and serviceability.
- Capacity optimization - Functionality (usually software) that more effectively utilizes physical storage space, such as compression, deduplication, and thin provisioning.

Other items such as power supplies, I/O ports, cooling components, interconnect ports, etc., are not being ignored but are considered to be aligned and scale with performance and the items defined in this section.

Reduction of the variable space to the five items listed in this section still leaves vendors with a potentially very large set of test requirements and cases, each with significant set-up and execution times. Even configurations in which the number and type of HDDs and SSDs are the only variables can be too difficult to support. Maximum system size tests are expensive and cumbersome to manage. Customers would have similar issues in attempting to parse through a large number of test results and make effective vendor product comparisons. Rather than attempt to reduce this variable set further, a different method is proposed, the "best foot forward" (a.k.a. "sweet spot") approach defined in Section 3.4.

3.4 Best Foot Forward Test Methodology

The Best Foot Forward (BFF) approach looks at a storage system product holistically. It allows the storage vendor to select and test one or more specific product/family configurations at operating points determined to be at or near Measurement Specification metric peak values, i.e., the "sweet spots." This results in a reduced test result set representative of the entire product family, which is both easier and less expensive for vendors to test and produces results simpler to understand and therefore more useful to customers.

The approach is based on the idea that the Measurement Specification active metrics have “peak” value points located within smaller— and hence more easily testable—product/family configurations. The vendor selects one or more appropriately representative configurations and locates these Measurement Specification metric peak points. Key to this method is the avoidance of maximum configuration testing and other complex methods such as extrapolation and interpolation. (Note that in some cases of smaller systems, the maximum configuration may in fact be the BFF).

The diagram in Figure 2 shows an example of a hypothetical storage system in which system scaling is by capacity and performance tends to roll off at scale.

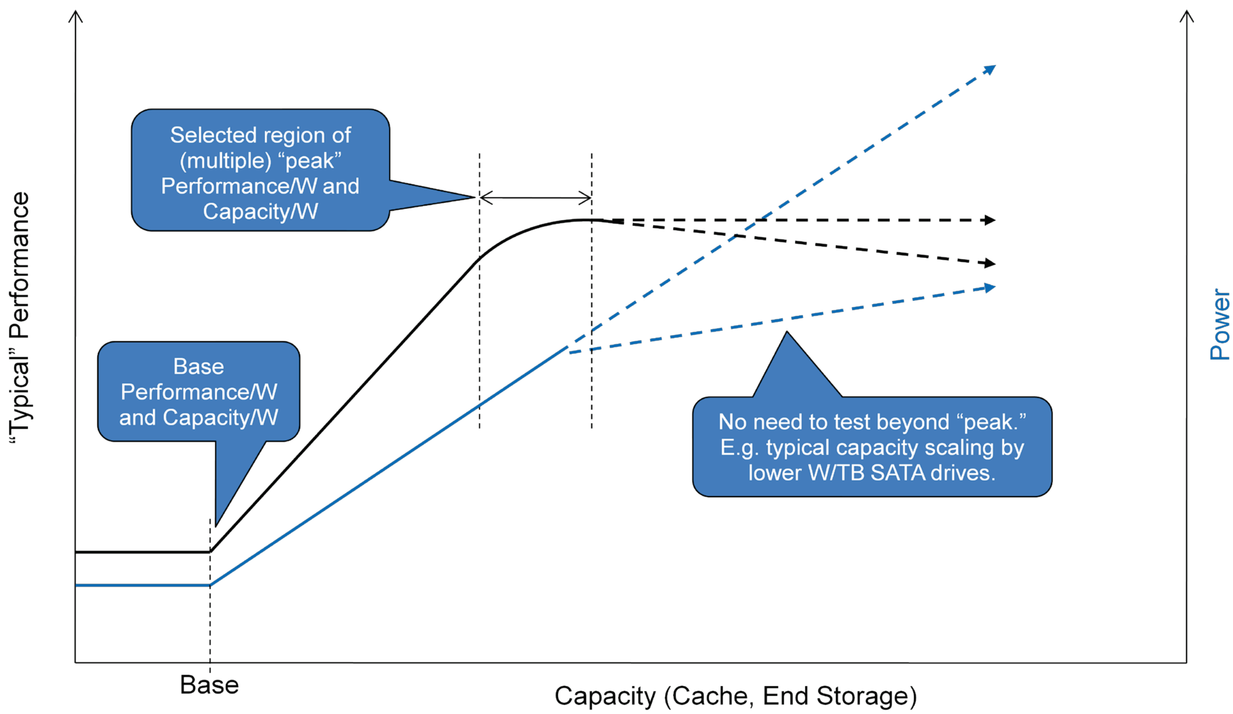

Figure 2 Hypothetical Storage System Performance/Power Example

The lines in Figure 2 represent highly simplified pictorial approximations and will vary with real systems (in scale-out systems the performance line may not roll off as extensively). Regardless, the example attempts to depict how a smaller representative system can be selected and tested at its vendor-determined peak Measurement Specification metric value points. One could also test at the base (entry point). However, there is no requirement to test beyond the peak points. In fact, many systems already scale capacity using high GB/Watt, high capacity HDDs. Similarly, scale-out systems can scale performance and capacity by step-and-repeat instantiation of the same devices as those tested per the BFF method.

4 Finding the Best Foot Forward

4.1 Overview and Goal

The Best Foot Forward (a.k.a." sweet spot") as a methodology for testing product/family configurations at the peak values of the power efficiency metrics was introduced in Section 3.4. The stated benefit of this approach is to reduce the testable sets from a large variable range to fewer in number (potentially just one) with the test results representative of the entire product family.

This section describes one method for finding the Best Foot Forward configuration by using prediction tools; it also provides characteristics of the approach. By using the described tools, a large range of configuration variables can be evaluated and the predicted sweet spots arrived at relatively quickly.

4.2 A Step-wise Approach

To determine the Best Foot Forward, a vendor can follow these steps:

1. Start with a product offering that fits within a taxonomy definition. If the product can be configured to fit into several taxonomies, then the vendor needs to consider a separate data submission for each applicable taxonomy category and classification.
2. Considering all possible (and valid, i.e., saleable) product SKU's (Stock Keeping Unit), identify the optimized configurations that will give the peak power efficiency metrics.

   Since there are seven different SNIA Emerald™ Program test profiles (six active and one idle), it is expected that there can be up to seven different optimized (tuned) configurations that achieve peak metrics:

   - 4 x Random [IO/s/Watt], 2 x Sequential [MiB/s/Watt], and 1 x Ready-Idle [raw capacity, GB/Watt]
3. Use estimator tools to predict the peak metrics. The alternative is to develop educated-guess derivations, which could potentially lead to a significant amount of labor- and resource-intensive testing. As long as the simulated results are reasonably accurate, the physical configuration selected to identify (by measurement) the peak value can be quite limited in range.
4. Set up, test, and measure the peak metric values for your first sweet-spot:
   - Run through the complete sequence of SNIA Emerald™ test profiles.
   - Test, validate and data correlate the predicted results.
5. Re-configure and re-test for each additional sweet spot of interest.

For each sweet spot, there is a tuned configuration that will produce a peak metric for a specific test profile. However, a single tuned configuration may, in fact, generate multiple peak metrics for related workloads (i.e., random or sequential). When submitting sweet-spot data, it may be advantageous to identify the SUT as *optimized to perform best at specific test profile "X."*

4.3 Discussion of Estimator/Simulation Tools

When faced with the task of finding the peak metric values of full product/family range of configurations, estimator tools can be an invaluable aid. Storage vendors may have a variety of power calculator and performance estimator tools for their storage products. Some may even have tools that can predict a limited set of power efficiency metrics. These tools can be based on complex simulation methods and/or grounded on some data points with interpolation and extrapolation. The accuracy of prediction is always in question, and thus the predicted results will always need to identify completed data correlations before accuracy claims can be made.

The *Test Data Report* may contain spec sheet data that allows customers to perform these calculations, as well.

4.4 Example Exercises

Using power calculator and performance estimator tools for a representative Online-3 array, some characteristic plots of performance, power and the power efficiency metrics were generated for the SNIA Emerald™ Program test profiles. The array controller performance options were fixed to a high level, and then the configuration variables in the drive type and drive count were evaluated (note: SSDs were not evaluated in this set of exercises). The maximum configuration size of this array is 240 LFF drives and/or 450 SFF drives. The objective of the prediction exercises is to find the peak metrics for power efficiency for each test profile. Several illustrative plots are shown in Figure 3 , Figure 4 , and Figure 5 To arrive at the absolute "sweet spot" for each SNIA Emerald™ Program test profile, some additional variable tuning may be required.

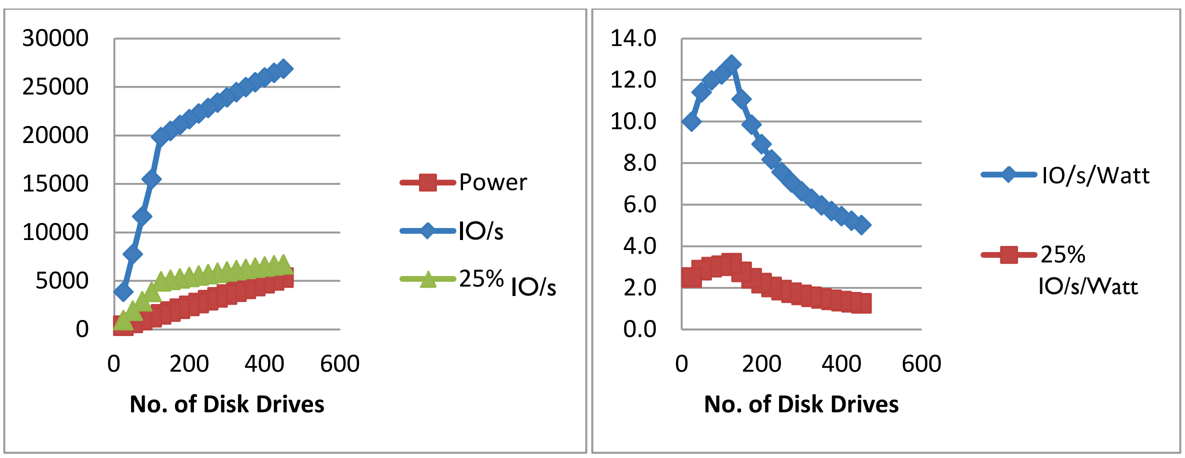

Figure 3 Performance, Power, and Power Efficiency Metric vs. Drive Count [Mixed Workload 8K Random 70/30 R/W, SFF 15K rpm SAS drives, RAID 5]

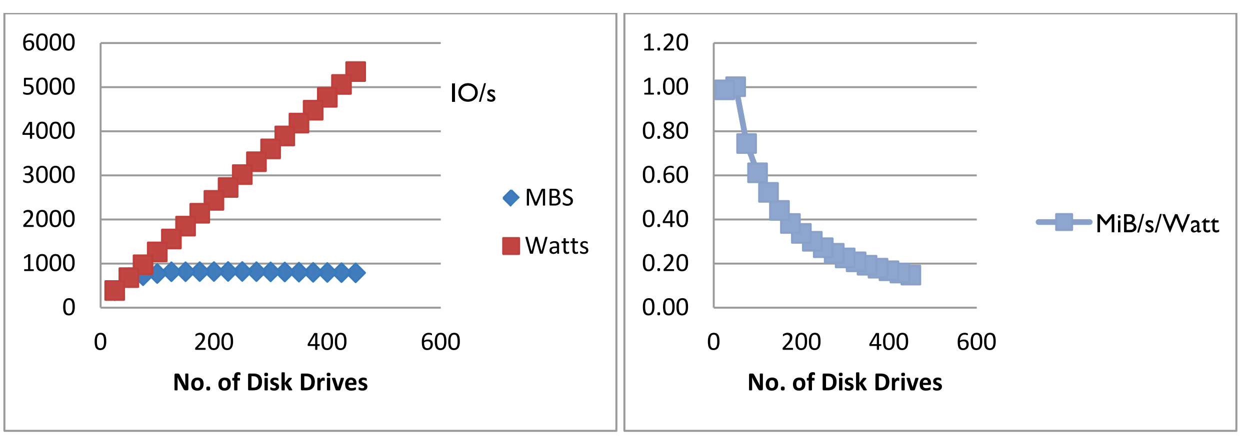

Figure 4 Performance, Power, and Power Efficiency Metric vs. Drive Count [128K Sequential Write, SFF 15K rpm SAS drives, RAID 5]

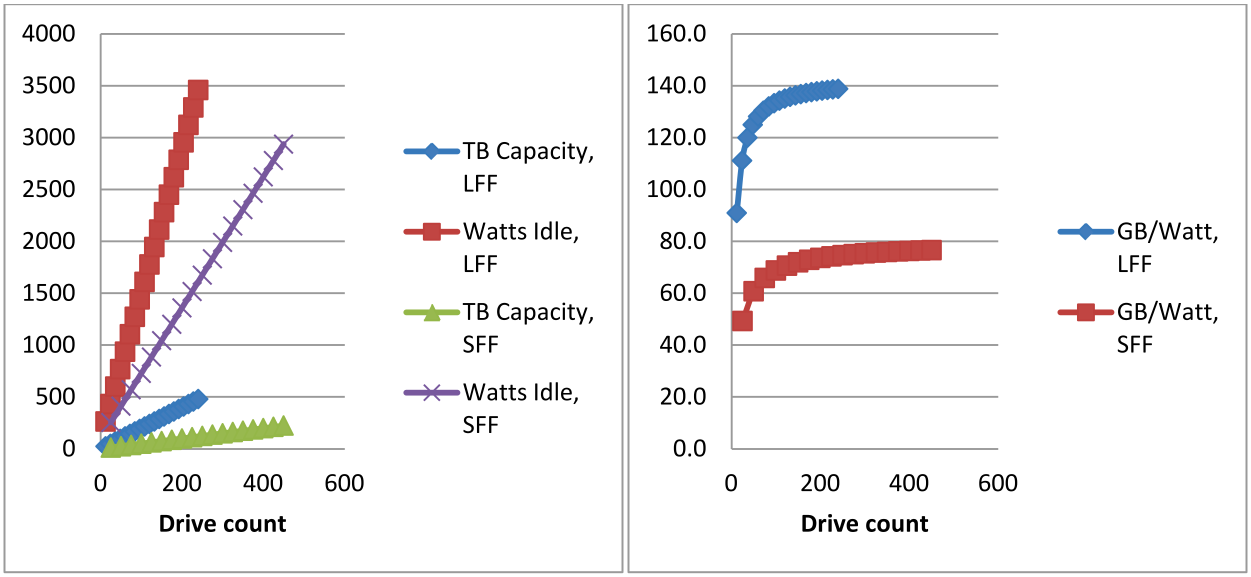

Figure 5 Idle Capacity, Power, and Idle Efficiency Metric vs. Drive Count [LFF 2TB 7.2K rpm and SFF 500GB 7.2K rpm SAS drives]

Obviously, based on the controller performance, bandwidth, and hardware efficiency, the slopes and shapes of these curves will vary. However, these observations can be made from this example:

- For all cases, the power steadily and regularly increases as the configuration size increases.
- For all active cases, the performance reaches a peak at a configuration considerably smaller than the largest drive count; then it levels out or goes down slightly.
- For all active cases, the peak metric [performance/power] is also reached at relatively small configurations.
- For random and for sequential workloads, the peak metrics were achieved with the SFF, 15K rpm spinning drive.
- For the ready idle case, the peak metric continues to rise with drive count (as the controller electronics power is amortized over increasing numbers of drives).

5 Setting up the Test

The Measurement Specification includes the procedures used to derive the storage power efficiency metrics. The test procedure used by the Measurement Specification for all storage taxonomy categories follows the same basic test flow of:

- SUT conditioning
- Active measurements
- Ready Idle measurements
- Capacity optimization method (COM) verification.

These tests are required to be run in an uninterrupted sequence. An I/O generator (i.e., Vdbench, IOMETER) is used to provide a simulated workload to the storage system during the SUT Conditioning Test and Active Test phases. The Ready Idle Test requires no external I/Os (I/Os from or to a host) but the system should still be connected to the network or host and ready to support an I/O request.

To generate a correct representation of the storage system power efficiency of a storage system, it must be properly pre-conditioned. This is the goal of the SUT Conditioning Test, which is designed to get the system to a known and stable state before the active measurements commence. All storage devices need to be touched during the SUT Conditioning Test. Any cache must be filled up so that the system is accessing end storage devices to obtain data. This process will ensure that the active measurement will be representative of an operational storage system at a customer site.

The active metric data is collected in minute-by-minute averages, which are used to calculate the overall average for the metric test window. During the SUT Conditioning, Active, and Ready Idle Tests, the power drawn by the storage system is measured and captured by a power meter in 5-second intervals. This data then is used to calculate an average power value. These values are then used to generate the metrics reported to the SNIA Emerald™ Program, where the performance or capacity is the numerator and the average power is the denominator. See section 6 for further information on required data for a valid storage system power efficiency submission to the SNIA Emerald™ Program.

5.1 Instrumentation

5.1.1 Recommend Power Meters

The power meter requirements are listed in Table 1. A list of recommended power meters is located in appendix A of the Measurement Specification.

Table 1: Power Meter Requirements

Power Consumption (p)	Minimum Accuracy
$p \leq 10$ W	± 0.01 W
$10 < p \leq 100$ W	± 0.1 W
$p > 100$ W	± 1.0 W

5.1.2 Setup

The storage system power efficiency measurements are intended to take place in a location indicative of a data center environment. The input power source for the storage system must meet the voltage requirements listed in the Measurement Specification. The temperature and humidity should follow the *ASHREA 2008 Thermal Guidelines Class 1* specifications. The data collection requirements are listed in **Table 2**.

Table 2: Data Collection Summary

Test	Collection Interval (seconds)		Minimum Benchmark Driver Data Collection		Minimum Test Duration (minutes)	
	Power Meter	Temp Meter	Online/ Near Online	Removable/ Virtual	Online/ Near Online	Removable/ Virtual
Conditioning	5	60	Response Time (per 1m interval)	Throughput (MB/s)	720	7
Active	5	60	Response Time (per 1m interval)	Throughput (MB/s)	30	30
Idle	5	60	N/A	N/A	120	120

The general setup is shown in Figure 6

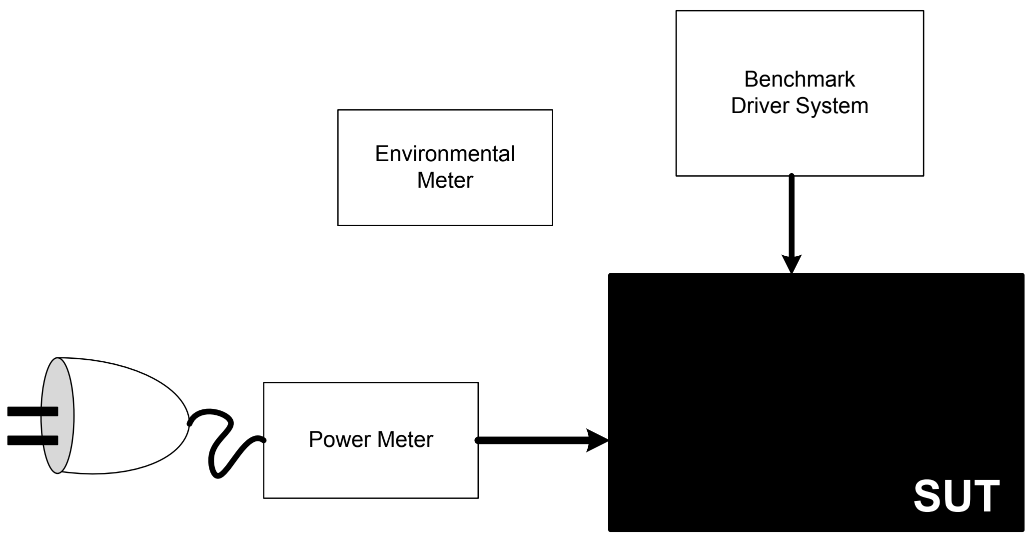

Figure 6 Basic power measurement setup

5.2 *SUT Conditioning, Active, and Idle Tests*

The SUT Conditioning Test is used in the Measurement Specification to demonstrate the SUT's ability to satisfy an I/O request, ensure the storage devices of the system is fully operational, achieve operational temperature, and place the storage system in a stable and known state. The minimum SUT Conditioning Test time period is defined by each storage category but may be increased in length as necessary. The goal of the SUT Conditioning Test is to get the system up to a stable operational state such that data must be transferred to or from the end-user storage devices (i.e., disk drive, tape drives) and not just cache. All storage devices must be accessed at some time during the test.

The workload and time requirements for each test for each category of storage are specified in the Measurement Specification. Sections 5.2.1, 5.2.2, and 5.2.3 include requirement highlights particular to each taxonomy category.

5.2.1 Online and Near Online

The Online and Near Online taxonomy categories are run through the same test sequence and measurement test windows, but have different response time requirements.

- Online systems must be able to obtain first data within 80mS, For Near Online, it can be longer than 80mS.
- Online systems must have an average response time of less than 30 mS for the last four hours of the SUT Conditioning Test and show system stability. Near Online systems do not have an average response time requirement but should show stability.
- For Online systems, the minute average response time must not be over 80mS for the active tests. For Near Online systems, there is no minute average response time constraint.
- Online systems must have an average response time of 30mS over the full active measurement test window. For Near Online systems, there is no average response time constraint.

5.2.2 Removable Media Library

The Measurement Specification calls for the power efficiency measurement of these systems to be within 80% of the published data throughput due to their sequential I/O nature. The published throughput of a system may have to be calculated by determining the throughput average of the various media devices.

5.2.3 Virtual Media Library

The Measurement Specification calls for the power efficiency measurement of these systems to be within 90% of the published data throughput due to their sequential I/O nature.

5.3 Sample Vdbench Script

Section 5.3.1 provides sample Vdbench scripts for providing test workload generation. Vdbench is an open source storage I/O workload generator that can be downloaded from sourceforge.net: http://sourceforge.net/projects/vdbench/.

SNIA members have used Vdbench as the workload generator for the development of the Measurement Specification. However, any benchmark driver on the recommended list of the Measurement Specification can be used for the workload generator in the measurement of a storage system.

> **NOTE**: The current list of recommended benchmark drivers is in Appendix B of the Measurement Specification. This list will be updated by the SNIA Green Technical Working Group (TWG) over time. Any party that would like to recommend a benchmark driver can contact the SNIA Green TWG by sending email to greentwg-chair@snia.org. Please state in the subject matter that you are

requesting acceptance of a new benchmark driver for the Measurement Specification. The Measurement Specification lists the requirements for a valid workload generator.

Some recommendations before running the Vdbench scripts include:

- If you are not knowledgeable of Vdbench, it is strongly suggested that you read the user guide included with the Vdbench package.
- Use any available tools that may help you to configure your system for optimal use of energy based on your storage needs.
- If possible, pre-fill the disk with data. For small and some medium-sized storage systems, you can do this by running the conditioning part for a longer time. For medium-sized to large systems, it may not be possible to complete this process in a reasonable amount of time (i.e., less than 24 hours).
- Attempt, as much as possible, to do a single host run. It will make completing the measurement much easier.
- Note that the next configuration can all be done in a single file. For ease of use, the configuration has been separated it into two files named ***exploratory.cfg*** and ***emerald.cfg***.

5.3.1 Example script for finding the 30mS response

File: ***exploratory.cfg***

This section shows the full text of the file in *italics.* Following this text is the explanation on how it works.

```
#
hd=localhost,system=GRENNNAME.Green.green.com
#hd=GRENNAME,system= GRENNNAME.Green.green.com,shell=vdbench
#
sd=sd85_1,lun=\\.\PhysicalDrive13
sd=sd85_2,lun=\\.\PhysicalDrive14
sd=sd85_3,lun=\\.\PhysicalDrive15
sd=sd85_4,lun=\\.\PhysicalDrive16
sd=sd_1,lun=\\.\PhysicalDrive1
sd=sd_2,lun=\\.\PhysicalDrive2
sd=sd_3,lun=\\.\PhysicalDrive3
sd=sd_4,lun=\\.\PhysicalDrive4
sd=sd_5,lun=\\.\PhysicalDrive5
sd=sd_6,lun=\\.\PhysicalDrive6
```

```
sd=sd_7,lun=\\.\PhysicalDrive7

sd=sd_8,lun=\\.\PhysicalDrive8

sd=sd_9,lun=\\.\PhysicalDrive9

sd=sd_10,lun=\\.\PhysicalDrive10

sd=sd_11,lun=\\.\PhysicalDrive11

sd=sd_12,lun=\\.\PhysicalDrive12

#wd=encs302,sd=(sd302_1,sd302_2,sd302_3,sd302_4,sd302_5,sd302_6,sd302_7,sd30
2_8,sd302_9,sd302_10,sd302_11,sd302_12)

wd=enc85,sd=(sd85_1,sd85_2,sd85_3,sd85_4)

wd=encall,sd=(sd85_1,sd85_2,sd85_3,sd85_4,sd_1,sd_2,sd_3,sd_4,sd_5,sd_6,sd_7
,sd_8,sd_9,sd_10,sd_11,sd_12)

#rd=snia_mix_curve,wd=encall,iorate=curve,curve=(100,90,50,25),elapsed=30m,i
nterval=1m,pause=00,forxfersize=(8k),forrdpct=(70),forseekpct=(100),forrhpct
=(0),forwhpct=(0)

#rd=snia_mix_curve,wd=encall,iorate=(3500,3550,3600),elapsed=30m,interval=1m
,pause=00,forxfersize=(8k),forrdpct=(70),forseekpct=(100),forrhpct=(0),forwh
pct=(0)

#rd=snia_write_curve,wd=encall,iorate=curve,curve=(100,90,80),elapsed=30m,in
terval=1m,pause=00,forxfersize=(8k),forrdpct=(0),forseekpct=(100),forrhpct=(
0),forwhpct=(0)

#rd=snia_write_curve,wd=encall,iorate=(1990,2000,2020),elapsed=40m,interval=
1m,pause=00,forxfersize=(8k),forrdpct=(0),forseekpct=(100),forrhpct=(0),forw
hpct=(0)

#rd=snia_read_curve,wd=encall,iorate=curve,curve=(100,90),elapsed=30m,interv
al=1m,pause=00,forxfersize=(8k),forrdpct=(100),forseekpct=(100),forrhpct=(0)
,forwhpct=(0)

#rd=snia_read_curve,wd=encall,iorate=(4150,4200),elapsed=40m,interval=1m,pau
se=00,forxfersize=(8k),forrdpct=(100),forseekpct=(100),forrhpct=(0),forwhpct
=(0)

#rd=snia_seqwrite_curve,wd=encall,iorate=(660,670,680),elapsed=30m,interval=
1m,pause=00,forxfersize=(256k),forrdpct=(0),forseekpct=(0),forrhpct=(0),forw
hpct=(0)

#rd=snia_seqread_curve,wd=encall,iorate=(2000,2050,2100),elapsed=30m,interva
l=1m,pause=00,forxfersize=(256k),forrdpct=(100),forseekpct=(0),forrhpct=(0),
forwhpct=(0)

#rd=snia_seqwrite_curve1,wd=encall,iorate=curve,curve=(100),elapsed=30m,inte
rval=1m,pause=00,forxfersize=(256k),forrdpct=(0),forseekpct=(0),forrhpct=(0)
,forwhpct=(0),forthreads=(2,4)

#rd=snia_seqread_curve1,wd=encall,iorate=curve,curve=(100),elapsed=30m,inter
val=1m,pause=00,forxfersize=(256k),forrdpct=(100),forseekpct=(0),forrhpct=(0
),forwhpct=(0),forthreads=(2,4)
```

Explanation of the *exploratory.cfg* file

On the first two lines, the host(s) running Vdbench are defined. For this example, the test is run on a single host.

NOTE: The commented-out host definition is just an example; you should replace this with your host names.

```
#
hd=localhost,system=GRENNNAME.Green.green.com
#hd=GRENNAME,system= GRENNNAME.Green.green.com,shell=vdbench
#
```

Storage Definition (SD). These are the volumes you defined on your system, as they are presented to the host(s). The volumes as configured for this example are:

```
sd=sd85_1,lun=\\.\PhysicalDrive13
sd=sd85_2,lun=\\.\PhysicalDrive14
sd=sd85_3,lun=\\.\PhysicalDrive15
sd=sd85_4,lun=\\.\PhysicalDrive16
sd=sd_1,lun=\\.\PhysicalDrive1
sd=sd_2,lun=\\.\PhysicalDrive2
sd=sd_3,lun=\\.\PhysicalDrive3
sd=sd_4,lun=\\.\PhysicalDrive4
sd=sd_5,lun=\\.\PhysicalDrive5
sd=sd_6,lun=\\.\PhysicalDrive6
sd=sd_7,lun=\\.\PhysicalDrive7
sd=sd_8,lun=\\.\PhysicalDrive8
sd=sd_9,lun=\\.\PhysicalDrive9
sd=sd_10,lun=\\.\PhysicalDrive10
sd=sd_11,lun=\\.\PhysicalDrive11
sd=sd_12,lun=\\.\PhysicalDrive12
```

Work Definitions (WD).

The work definitions include the volumes defined previously in this section, as shown in this example. For preparation of the final test, you may want to have several WDs.

```
#wd=encs302,sd=(sd302_1,sd302_2,sd302_3,sd302_4,sd302_5,sd302_6,sd302_7,sd30
2_8,sd302_9,sd302_10,sd302_11,sd302_12)

wd=enc85,sd=(sd85_1,sd85_2,sd85_3,sd85_4)

wd=encall,sd=(sd85_1,sd85_2,sd85_3,sd85_4,sd_1,sd_2,sd_3,sd_4,sd_5,sd_6,sd_7
,sd_8,sd_9,sd_10,sd_11,sd_12)
```

Run Definitions (RD)

This is the section used to find the maximum IO/s and MiB/s. From the numbers found, the fix values for the *emerald.cfg* file were established .

It is strongly suggested that only one curve be run at a time. To do so, just remove the comment character of the line containing the run definition for the curve you are interested in testing.

```
#rd=snia_mix_curve,wd=encall,iorate=curve,curve=(100,90,50,25),elapsed=30m,i
nterval=1m,pause=00,forxfersize=(8k),forrdpct=(70),forseekpct=(100),forrhpct
=(0),forwhpct=(0)

#rd=snia_mix_curve,wd=encall,iorate=(3500,3550,3600),elapsed=30m,interval=1m
,pause=00,forxfersize=(8k),forrdpct=(70),forseekpct=(100),forrhpct=(0),forwh
pct=(0)

#rd=snia_write_curve,wd=encall,iorate=curve,curve=(100,90,80),elapsed=30m,in
terval=1m,pause=00,forxfersize=(8k),forrdpct=(0),forseekpct=(100),forrhpct=(
0),forwhpct=(0)

#rd=snia_write_curve,wd=encall,iorate=(1990,2000,2020),elapsed=40m,interval=
1m,pause=00,forxfersize=(8k),forrdpct=(0),forseekpct=(100),forrhpct=(0),forw
hpct=(0)

#rd=snia_read_curve,wd=encall,iorate=curve,curve=(100,90),elapsed=30m,interv
al=1m,pause=00,forxfersize=(8k),forrdpct=(100),forseekpct=(100),forrhpct=(0)
,forwhpct=(0)

#rd=snia_read_curve,wd=encall,iorate=(4150,4200),elapsed=40m,interval=1m,pau
se=00,forxfersize=(8k),forrdpct=(100),forseekpct=(100),forrhpct=(0),forwhpct
=(0)

#rd=snia_seqwrite_curve,wd=encall,iorate=(660,670,680),elapsed=30m,interval=
1m,pause=00,forxfersize=(256k),forrdpct=(0),forseekpct=(0),forrhpct=(0),forw
hpct=(0)

#rd=snia_seqread_curve,wd=encall,iorate=(2000,2050,2100),elapsed=30m,interva
l=1m,pause=00,forxfersize=(256k),forrdpct=(100),forseekpct=(0),forrhpct=(0),
forwhpct=(0)
```

```
#rd=snia_seqwrite_curve1,wd=encall,iorate=curve,curve=(100),elapsed=30m,interval=1m,pause=00,forxfersize=(256k),forrdpct=(0),forseekpct=(0),forrhpct=(0),forwhpct=(0),forthreads=(2,4)

#rd=snia_seqread_curve1,wd=encall,iorate=curve,curve=(100),elapsed=30m,interval=1m,pause=00,forxfersize=(256k),forrdpct=(100),forseekpct=(0),forrhpct=(0),forwhpct=(0),forthreads=(2,4)
```

After obtaining the optimal results for each curve, save the values of IO/s (and make note of any other adjustments you have made to the runs); you will need them for the **emerald.cfg.**

For this example of the *exploratory.cfg* file, the IO/s values found for each of the run definitions were:

For snia_mix_curve: IO/s=3550

For snia_write_curve: IO/s=2000

For snia_read_curve: IO/s=4200

For snia_seqwrite_curve: IO/s=660

For snia_seqread_curve: IO/s=2000

NOTE: These five numbers show up as bold in Section 5.3.2.

5.3.2 Sample script for the full test results

File: ***emerald.cfg***

Once the optimal values are found using the *exploratory.cfg* file, use them on the script in this section, and execute it without interruption for the final SNIA Emerald™ Program run.

The values found are shown here in **bold** and are the only ones that need to be changed.

It is strongly suggest that you complete a verification run before starting the official run. There are often adjustments needed after the variable runs.

```
#

hd=localhost,system=GRENNNAME.Green.green.com

#

sd=sd85_1,lun=\\.\PhysicalDrive13

sd=sd85_2,lun=\\.\PhysicalDrive14

sd=sd85_3,lun=\\.\PhysicalDrive15

sd=sd85_4,lun=\\.\PhysicalDrive16

sd=sd_1,lun=\\.\PhysicalDrive1

sd=sd_2,lun=\\.\PhysicalDrive2

sd=sd_3,lun=\\.\PhysicalDrive3
```

```
sd=sd_4,lun=\\.\PhysicalDrive4
sd=sd_5,lun=\\.\PhysicalDrive5
sd=sd_6,lun=\\.\PhysicalDrive6
sd=sd_7,lun=\\.\PhysicalDrive7
sd=sd_8,lun=\\.\PhysicalDrive8
sd=sd_9,lun=\\.\PhysicalDrive9
sd=sd_10,lun=\\.\PhysicalDrive10
sd=sd_11,lun=\\.\PhysicalDrive11
sd=sd_12,lun=\\.\PhysicalDrive12
wd=encall,sd=(sd85_1,sd85_2,sd85_3,sd85_4,sd_1,sd_2,sd_3,sd_4,sd_5,sd_6,sd_7
,sd_8,sd_9,sd_10,sd_11,sd_12)
rd=conditioning,wd=encall,iorate=3550,elapsed=12h,interval=1m,pause=00,forxf
ersize=(8k),forrdpct=(70),forseekpct=(100),forrhpct=(0),forwhpct=(0)
rd=mw1,wd=encall,iorate=3550,elapsed=40m,interval=1m,pause=00,forxfersize=(8k
),forrdpct=(70),forseekpct=(100),forrhpct=(0),forwhpct=(0)
rd=mw2,wd=encall,iorate=888,elapsed=40m,interval=1m,pause=00,forxfersize=(8k
),forrdpct=(70),forseekpct=(100),forrhpct=(0),forwhpct=(0)
rd=rw,wd=encall,iorate=2000,elapsed=40m,interval=1m,pause=00,forxfersize=(8k
),forrdpct=(0),forseekpct=(100),forrhpct=(0),forwhpct=(0)
rd=rr,wd=encall,iorate=4200,elapsed=40m,interval=1m,pause=00,forxfersize=(8k
),forrdpct=(100),forseekpct=(100),forrhpct=(0),forwhpct=(0)
rd=sw,wd=encall,iorate=660,elapsed=40m,interval=1m,pause=00,forxfersize=(256
k),forrdpct=(0),forseekpct=(0),forrhpct=(0),forwhpct=(0)
rd=sr,wd=encall,iorate=2000,elapsed=40m,interval=1m,pause=00,forxfersize=(25
6k),forrdpct=(100),forseekpct=(0),forrhpct=(0),forwhpct=(0),forthreads=(4)
```

5.4 Capacity Optimization

The COMs tests are effectively existence tests. They include:

- Delta Snapshots (read and write)
- Thin Provisioning
- Data De-duplication
- RAID groups
- Compression

5.4.1 Finding the c Program

The program that generates the required dataset for the deduplication test can be downloaded from sourceforge.net:

http://sourceforge.net/projects/sniadeduptest/files/

Detailed instructions for running it are contained in the c file itself.

5.4.2 Running Capacity Optimization Tests

Vendors must follow the given steps for each COM they wish to be given credit for on a given SUT. Each heuristic requires vendors to document how to perform a set of steps on said SUT. No media may be added or removed, nor changed in state (taken on- or off-line, made a spare, or incorporated, etc.). RAID groups may not be changed. In the event of an automated disk failure and subsequent RAID rebuild at any time during a test, the test must be restarted after the rebuild is completed, and the failed disk replaced per manufacturers guidelines for installed and working systems.

Prior to running the heuristics, up to three data sets must be generated, one that is uncompressible but deduplicatable, one that is compressible but undeduplicatible, and the third that is completely unreducible. The supplied data sets are generated by the C program sniadeduptest.c. Before testing, compile this program and load it on a host that will be used to write the data to the SUT. The data sets will be created in a directory named "snia_capop_data" and will be approximately 2GB in size. The names of the subdirectories and files are

- unreducible: unreducible.dat
- compressible: compressible.dat
- dedupable: dedupable.dat

5.4.2.1 Delta Snapshots

Delta snapshots in a storage system can be detected using a straightforward algorithm:

a) Query the free space before taking a snapshot.

b) Attempt to create a snapshot.

c) Write something to the snapshot in the case of a writeable one.

d) Query the free space after that snapshot to determine whether significant storage space has been used.

The goal of this heuristic is simply to document how an independent third party can verify that the system under test is indeed capable of supporting delta snapshots. As stated previously, read-only and writeable delta snapshots are treated separately so that systems that only do read-only snapshots may get credit for them.

5.4.2.2 Thin Provisioning

The goal of this heuristic is not to highlight differences of thin provisioning implementations between vendors; it is to be used simply to ensure that the product under test has some sort of thin provisioning capability.

5.4.2.3 Data deduplication

Data set size is not considered very important for the purposes of deduplication detection. The heuristic builds a 2GB dataset consisting of uncompressible blocks of data, and then uses it to detect deduplication capability. The initial dataset contains many duplicated files of various sizes and many duplicated blocks aligned on block boundaries. It also contains duplicated blocks of variable lengths that are not aligned on block boundaries. This allows detection of block-based schemes, variable-length schemes, and SIS schemes when used in place.

The data deduplication test is the only test provided in this version of the de-duplication heuristics, as this version is aimed solely at primary (online) storage, and temporal deduplication is almost always used with secondary storage.

To better understand deduplication, refer to "Understanding Data Deduplication ratios" -- DDSR SIG, located at this website:

http://www.snia.org/sites/default/files/Understanding_Data_Deduplication_Ratios-20080718.pdf

5.4.2.4 Advanced RAID

Capacity utilization and improvement relative to a comparable RAID-1 configuration—the relative storage efficiency ratio—is simple to calculate, given that RAID group sizes and parity requirements are simple and well known.

5.4.2.5 Compression

The heuristic is accomplished by provisioning an empty container and storing the reference data sets onto it, then determining the actual amount of space used for storing the data.

5.5 Avoiding Potential Pitfalls while Taking Measurements

With all the complexities of storage systems, not all potential pitfalls of taking measurements on the system can addressed by the Measurement Specification. This section lists issues that may need to be addressed by the individual taking the measurements. This is not intended to be an all inclusive list, but rather a list of general items that need to be considered and/or addressed before taking the measurement.

- Ensure that the measurement includes enough writing on the sequential write test to have enough written data for a stable sequential read test.
- If a system is to be tested for a particular Capacity Optimization Method, the COM feature must be enabled during all other measurements.

- All disclosed RAS features must be turned on during the measurement. Although certain RAS features should have enough time to complete before measurements are taken, tasks such as charging batteries should be completed before the measurement starts, as this is not a typical daily activity. Any RAS feature that is a typical daily activity should be captured in the measurement.
- Timing between the workload generator and the power/temperature meter should match. Any offset will cause the metric generation to be off. The time settings should be within one second of each other.
- The host providing the workload to the storage system should not be the bottleneck.

6 Notes on Collecting and Submitting Data

Each Measurement Specification Active Test phase is effectively comprised of two parts—metric stability verification followed by the actual metric measurement interval. The stability verification must successfully complete before the measurement interval can commence. Additionally, stability must be maintained during the subsequent measurement interval.

The stability of a system is defined by looking at the minute-by-minute ratio of performance per watt. As detailed in the Measurement Specification, performance data is collected in 1 minute averages. The power meter collects power measurements every 5 seconds. The average power on a 1 minute basis is then generated. Each 1 minute average of performance and power is then used to generate a ratio of performance per watt. This data point is further used to verify stability of the system by using the 10-point exponential moving average (refer to the Measurement Specification for the equation).

The first valid stability verification checkpoint is not reached until at least ten of the above data points have been calculated (i.e., at least 10 minutes). Stability is reached when the exponential moving average no longer deviates by more than 5%. A graph of each of these minute-by-minute data points and the corresponding exponential moving average is useful to see system stability and can optionally be added to the report provided to the SNIA Emerald™ Program. The measurement interval must last at least 30 minutes with stability maintained during the entire interval. The Active Test phase metric is determined at the end of the measurement interval and is the average performance divided by the average power over the last 30 minutes of the measurement interval. The minute-by-minute performance averages can be used to calculate the average performance, but the 5-second power readings should be used to calculate the average power.

Primary metrics, which are reported to the SNIA Emerald™ Program, are defined in Clause 8 of the Measurement Specification for each of the different taxonomy categories. With the primary metrics determined, the Test Data Report for SNIA can be generated for a submission to the SNIA Emerald™ Program. The Test Data Report input form can be found on the SNIA Emerald™ Program website: www.sniaemerald.com. This report provides entries for basic system information, test setup information, and the results for each of the test phases defined for the taxonomy category. Once the report is generated and verified for accuracy it can be submitted to the SNIA Emerald™ Program via the website.

7 Additional Advisory Notes

7.1 Tradeoffs

As shown in *Section 3 Defining the Product Family* and *Section 4 Finding the Best Foot Forward*, there are tradeoffs between capacity, performance, and power. These tradeoffs need to be evaluated by storage system vendors when promoting their products to specific markets. At this time it is nearly impossible to define a single storage power efficiency metric proxy for all capable system setups. As such, it is in the best interest for the vendors to submit multiple system configurations as appropriate to the SNIA Emerald™ Program to show the overall storage power efficiency. Consumers of storage systems should look at multiple storage power efficiency measurements of a target system to see the total picture.

7.2 Reliability/ Availability / Serviceability (RAS)

SNIA has identified several Reliability/Availability/Serviceability (RAS) features of storage systems with significant impacts on power consumption. These are listed in Table 10 of the Measurement Specification. These RAS features usually have some impact on power yet are a requirement of high availability and serviceability features in many of today's storage systems. The issue with such functions is that their existence may have no direct benefit on performance and hence may have a detrimental impact on certain Measurement Specification metrics.

All RAS features in a SUT must be active during measurement tests to provide an accurate and appropriate measurement. The taxonomy attempts to take into account most of these requirements in the related classification sections.