Technology Insight Paper

Measuring Performance of Solid State Storage Arrays

Using Data Set and Data Stream Pattern Modeling

By Leah Schoeb

June 2014

Evaluator Group

Enabling you to make the best technology decisions

Evaluator Group

Introduction

The advent of datacenter consolidation with server and desktop virtualization has created highly demanding and randomized, latency-sensitive workloads that rival the demands of mission critical database environments. Solid state storage (SSS) arrays are an increasingly important solution within modern IT infrastructures to accelerate specifically Tier-1 applications, virtual infrastructures, and the next generation of computing with modern applications, such as Hadoop and powerful analytics applications.

The amount of data generated in today's datacenters, coupled with the needed online performance and availability, are costly to IT departments in terms of both equipment and personnel. SSS arrays represent a way to keep up with performance demands of modern business-critical applications while reducing operational costs. When coupled with capacity optimization technologies like deduplication and compression, modern SSS-based arrays can greatly increase storage efficiency when compared to traditional Hard Disk Drive (HDD) arrays, delivering previously unavailable space and cost savings. Data reduction technologies have therefore become a mandatory part of the modern storage infrastructure and are an essential part of new SSS array designs. This means that, when measuring performance with these new arrays, new methodologies and tool capabilities are required to accurately model and benchmark application workloads that realistically characterize storage performance. At the very least, the deduplication and compression characteristics of the data sets and data streams used in the tests must be carefully validated.

The performance of a storage array with built-in data reduction technologies is dependent upon the breadth of the data content, metadata characteristics, and access patterns presented to an array. Today, most data sets and data streams produced by current I/O generation tools to measure data reduction storage arrays are inadequate to demonstrate actual performance levels. But that is changing and there are a small number I/O generation tools that have added the appropriate level of complexity to measure the performance of modern SSS arrays. This document first presents an approach to the steps needed to create appropriate data sets and data streams and then how to measure performance once they are presented to the storage array.

The performance methodology developed in this document addresses the specific requirements to measure the performance of enterprise SSS arrays using advanced capacity optimization technologies. The SNIA has done a great job in publishing a specification to measure raw performance for individual solid state devices (SSD), but a way is still needed to accurately measure SSS arrays at a system level supporting advanced inline data services, like data reduction technologies.

The tests listed in this document are designed to allow test sponsors to create repeatable, verifiable, and comparable performance results for SSS arrays that offer advanced data reduction features. This

document is intended to be vendor and platform independent and provide objective, relevant, and verifiable data to vendors and purchasers of SSS arrays. Measuring real world workloads is very important and this methodology should be considered when measuring production workloads. Dealing with the special needs and care of measuring production workloads versus synthetic workloads will not be discussed in this document.

This paper also discusses the differences in characterizing the performance of traditional HDD and hybrid HDD/SSS array versus an all-SSS array. The process presented in this document should be followed whether for running basic feeds and speeds measurements (commonly called four corner testing) or measuring the performance of a sophisticated application workload.

Unique Behavior of SSS Array vs. HDD Array Technologies

Traditional storage controllers were designed to handle HDD behaviors, including rotational latency, throttling to handle disk speeds, and de-staging algorithms from cache to disk, These behaviors perform magnitudes more slowly than solid state devices SSS. HDD performance is in the hundreds of IOPS with millisecond latencies; compare that to an solid state device (SSD), which performs thousands of IOPS with latencies in the microseconds.

SSS has unique performance characteristics that differ from what consumers are used to with HDD systems. Some of these HDD metrics are rotational latency, seek time, data density and other metrics associated with the workings of mechanical rotational media. Unlike HDD, SSS has no moving parts, so performance penalties associated with mechanical arms and rotating media no longer need to be considered. With the elimination of these rotational metrics, access times are much faster, typically in microseconds rather than milliseconds, and transfer rates are much higher. This results in much higher performance (IOPS) and read throughput (MB/sec).

The unique performance and behavior of solid state technology makes it necessary to design storage controllers differently than for HDDs. SSS controllers are basically managing persistent memory, usually with a translation layer to make them appear as HDD. Therefore they have similar management tasks as managing memory, for example, metadata management and wear leveling algorithms.

Inline deduplication can have additional performance impacts on HDD and SSS arrays. Some SSS array vendors have implemented a form of data reduction that performs some post-process data reduction in conjunction with real time in-line deduplication. Other implementations that use log-structured data layouts look to minimize fragmentation by grooming free space when data has been either overwritten or unmapped, which can cause holes in usable free space. In this case, system-level garbage collection is often used to defragment usable free space and make it easier to write new data using full stripe writes. With HDD arrays there are concerns about reducing fragmentation and amortizing seek times that are

caused by deduplication, which increase latency. These concerns have been optimized and managed well with backup and archival systems to HDD but still have not been optimized enough to significantly impact traditional HDD primary storage systems.

Factors Impacting SSS Array Performance

Inline Deduplication and Compression

Inline deduplication and compression mean that data is analyzed in-stream, to ensure data is stored only once. The hash calculations for the metadata are created on a target as the data enters the storage array. Inline compression reduces the size real time, If the algorithm spots a duplicate block, it creates a reference to the existing block instead of storing it again. One advantage of doing this task up front instead of during post-processing is that the technique uses less storage capacity. However, the deduplication overhead can have a negative impact on performance. The more efficient the algorithm, the lower the performance impact is on the system. When required to maintain the latency-sensitive application performance, vendors with advanced algorithms spend much time minimizing this impact. They reduce deduplication by managing the required metadata and lookup tables in a DRAM cache, along with optimizing how duplicates are fingerprinted and located. In addition, how metadata is managed by using optimized writes and by minimizing read-performance penalties can be very important.

Accurately measuring the performance of deduplication and compression depends on generating data content patterns sufficient to stress a storage array. A statistical model that is too small or lacks complexity cannot be used to measure solid state storage arrays with deduplication and compression.

This requires testing with algorithms that consider both spatial (duplicated data) and temporal (access patterns) locality. Few publicly available data sets for measuring data reduction technologies are available due to enterprise privacy concerns. Also, generating a challenging data set can be difficult since some deduplication implementations are more advanced than others and require much larger and more complex data sets to properly measure their efficiency.

Building a data set to successfully measure SSS arrays with deduplication and compression can be quite difficult if the resources are not available to carefully to construct the right statistical model. When measuring SSS arrays from multiple vendors it is good to always have a test scenario that measures 100% unique and incompressible data so that performance results from vendors not supporting in-line data reduction can show comparable results. This will be discussed later.

Protocols

FC is still the highest performing storage protocol but gains can also be made from SAS, iSCSI and SATA-based SSS solutions. This may change in the future as solid state technologies move towards communicating more like memory and move away from HDD communication protocols. This means that solutions to use PCIe for longer distances for external array use and better ways to use direct memory access are on the way.

Garbage Collection

SSS arrays do not change data in place; rather, when data changes, the block containing the data is re-written elsewhere and the stale block is marked for deletion, called garbage collection. Garbage collection locates and manages available free space by finding and removing blocks that are eligible for deletion. These algorithms are not designed equally and some algorithms could cause a negative performance impact. Measuring steady state performance with garbage collection activated is an important part of the measurement in order to get accurate performance results. If extended measurements are being taken, sometimes it is useful to track when garbage collection turns on and off and at what degree (i.e. 5%, 50%). This can be a useful measure of behavioral predictability.

Some SSS array vendors have designed their products to mitigate the need for garbage collection by over-provisioning the storage. For these SSS arrays, production environments may take longer to see the performance impact of garbage collection. In this case it may take much longer to get to a steady state. This area is a bit controversial since some vendors claim their solution makes garbage collection a very rare occurrence and others are seeing their customers reach this state on average anywhere from 4 to 18 months depending on the workload.

Metadata Management

Metadata management controls the logical and physical locations of and is a critical factor in SSS arrays. Not all metadata management algorithms are designed the same; some algorithms are more efficient with less overhead and offer better performance than others. How metadata management is architected is very important to how data from virtual infrastructures, HPC, and tier 1 applications with large datasets are handled. Some vendors keep metadata in DRAM, SSS array capacity, or both:

- DRAM – Metadata stored in DRAM is very fast, but limits the total addressable virtual capacity of an array and can limit the maximum data reduction achievable if total capacity of the array is used.

- SSD capacity – Metadata storage in SSS array capacity is not quite as fast as DRAM but does allow for unlimited virtual capacity when data reduction is very high.
- DRAM and SSD capacity - Metadata storage in SSS array capacity with active metadata cached in DRAM can take advantage of the performance of DRAM while not limiting virtual capacity.

Wear Leveling Algorithms

For NAND Flash technology, wear leveling algorithms are critical to device longevity because they ensure that flash cells are written evenly across a LUN. These algorithms ensure that cells don't prematurely burn out, extending the life of a solid state device. This task is usually performed at the controller level and manages not only post-processing to help clean up physical locations, but also to manage the logical and physical abstraction of data.

Write Amplification

If an array performs writes in large block sizes, small batch writes will be bundled up into those larger block sizes before being written out to media. In SSS arrays this can in some cases improve performance, but depending on how well it is working with garbage collection this may or may not result in an increase in performance. Some solid state devices (SSDs) are unable to perform reads and writes simultaneously. Reads may be delayed if writes are hitting the same device or if the devices are performing a parity rebuild.

Block Alignment

Storage Misalignment has been a performance issue that has been around for a long time. It can contribute to as much as a 30% loss in performance if not resolved. In addition, this increase in the amount of writes produced in the array by having a misalignment over time can negatively affect the lifespan of an SSD.

Not all SSS arrays are created equal

Performance of a storage system is only as fast as its slowest part. Knowing this, the bottleneck should be the maximum performance of the back-end storage. In order to easily add tiered storage products to market, some vendors have chosen to add SSD support to their traditional HDD controller based storage. Unfortunately some of these controllers cannot realize maximum SSS performance. So the high price paid to add SSDs only allows partial performance gains because the system bottlenecks at the storage controller. Vendors that do have this issue are working hard to re-design their controllers to handle the higher performance of SSDs.

There are also a variety of protocols (FC, SAS, iSCSI) that vendors have used to design their SSS arrays, and each protocol will yield different levels of performance. In addition, some are block-based arrays and some products are file-based arrays, which is another performance consideration.

The performance of an SSS array performance is gated by the performance of its system design, unlike the performance of a single solid state device. Some SSS arrays are designed to perform like JBOD. Others are designed with advanced features built-in like data reduction technologies that can increase efficiencies in capacity and preserve the life of flash technology. The most popular data reduction technologies used are deduplication, compression, pattern removal, and thin provisioning. As SSS arrays offering data services continue to mature more advanced features will be made available.

SSS flash media comes in different forms, each with its own performance and reliability levels. Beyond dimensions like MLC, SLC, and so on, device and module design can be part of a vendor's 'secret sauce' for performance and reliability.

A variety of SSS array controller designs

There are many flash arrays on the market today, with each vendor promoting performance numbers with results that seem to be all over the place. Additionally, most vendors are calling their flash arrays 'all flash arrays,' which can be misleading because multiple flash arrays designs have been released. Effective use of this high performance technology is great for accelerating everything from a single application to a whole IT infrastructure, potentially reducing the amount of infrastructure needed to meet SLAs and offering CAPEX and OPEX savings if the infrastructure is well architected.

SSS Arrays based on 2.5" form factor devices

Some vendors use SSDs in an HDD 2.5" form factor and use standard interfaces like SAS or SATA to appear as storage on the backend. These 2.5" form factor SSDs can be used just like a disk drive in an existing traditional storage system or can be used in a SSS designed controller as a commodity part. This means any housekeeping activities unique to SSDs, like wear leveling, are left up to the internal features of the supported SSD. Other vendors have chosen to rely on garbage collection built in to the individual SSDs, called eMLC, and implement garbage collection by using log structuring and it's associated system level garbage collection. In some cases this may cause some performance consistencies and latency spikes. This can mean when the write cliff caused by break-in (when read-erase-write operation begin to occur because all blocks have been written often enough to force read-write-erase operations), performance degradation can be more severe than in flash arrays that have specifically managed around this type of event. Controller-based control can offer some increase in performance with a lower response time, but the performance will bottleneck if the storage controller has not been re-architected or architected from the ground up to handle the performance behavior of solid state technology.

Re-architected Controller

Some vendors with existing traditional storage controllers designed to support advanced features, such as copy (i.e. replication), data reduction technologies (i.e. deduplication), storage efficiency (i.e. thin provisioning), and other features supporting business continuity, have chosen to re-architect their controllers and customize their ASICs to support the unique behavior and performance of solid state technology while being able to continue offering the advanced features their customers have grown accustomed to.

SSS arrays based on a custom module design

Some vendors have taken their re-architected traditional storage array one step further by creating a custom flash module design that incorporates a controller to handle the housekeeping tasks, like metadata management and garbage collection, for a flash technology device. These flash module devices (FMDs) use a custom hardware design with a custom ASICs. This type of custom design may yield better performance than SSD implementations because of its tight integration. FMDs have an embedded controller on each device that handles flash storage efficiency and management, such as wear leveling, data refresh and error correction, block/page mapping, inline data reduction, an endurance manager and a performance manager.

Controller Designed Specifically for SSS

These arrays are architected from the ground up specifically for flash technology. They are especially aggressive on metadata management and buffering algorithms that have an impact on how garbage collection is efficiently handled. These flash controllers have been designed from the ground up to optimize metadata management, garbage collection, wear leveling, PCIe support, and optimizing writes with write coalescing. All the critical features needed for good memory management and persistence may be supported natively.

These arrays are currently limited on offering general data services but do offer advanced data reduction technologies, such as deduplication, compression, and thin provisioning. They use these efficiency technologies inline to reduce the number of writes to get the most efficient use and longevity of flash storage capacity. For further efficiency, some vendors are also including thin provisioning technologies to achieve the maximum use of capacity. Performance is not only dependent on the efficiency of the deduplication or compression algorithm but also on the amount of CPU cycles that are available for processing (especially for rehydration) and how well the array can handle frequent read-modify-writes (i.e. active database data sets). For both performance and efficiency, some vendors use these data reduction features globally either across the array or in cluster implementation.

Some vendors offer scale-out clustering architectures as opposed to a scale up architecture that could be more attractive to in-memory databases when global scale-out capacity is present as a single source of SSS. Other vendors offer other cluster technologies that are compatible with being clustered with other storage products within their storage product family.

The internal fabric design of these arrays have the capability to handle memory speed by using PCIe internally and DMA is used to handle DRAM backup. There are newer self-healing techniques that use cell sparing to handle failures and rebuilds. Other designs use an algorithm to find available pages instead of using physical dedicated spares.

Measuring Performance of SSS arrays

The basic overall process that the SNIA SSSI has defined for measuring solid state devices can also be applied to measuring SSS arrays. The item that is missing for solid state arrays is consideration for advanced features, such as data reduction technologies that all-flash array vendors are now designing into their systems. Most of the usual I/O generators used in measuring SSS performance are not designed to properly stress technologies like deduplication or compression because they lack the ability to generate the data sets and data streams required to stress a SSS array. Not all data reduction technologies are created equal and some designs are more sophisticated and complex than others. This difference makes it more difficult to create data sets and data streams that are acceptable to measure performance is the presence of advanced features.

SSS array measurement methodology considerations and guidelines

Measuring consistent performance for solid state arrays can be challenging depending on the architecture of the storage products. Some arrays are built with consistent predictable performance in mind; others require you to be aware of the nature of solid state technology. In either case, being aware that there is a 'write cliff' at some point during the lifetime of the array is important. If an array has not been pre-conditioned before use, the write cliff usually happens after using the array at a steady production rate anywhere from three to 18 months depending on the workload presented to the array. This means when all of the cells have been written to at least once, a read-erase-write operation starts happening every time a write request comes to the device. This will cause degradation in performance and when this transition is measured it looks like a cliff when the data is plotted on a performance curve. Some array vendors have planned for this event to offer more predictable performance. If performance measurements are to be performed on the array, the act of pre-conditioning gets this write cliff event out of the way so that the real long-term steady state measurement can be made.

Data reduction MUST be part of the measurement

Many SSS arrays are designed with inline data reduction services to help maximize the performance, lifespan and efficiency of the array. To be fair, performance and throughput comparisons must be enabled to ensure accurate performance results. Measuring performance without the features normally used in production does not produce accurate results. If the data services are architected efficiently there should be little to no performance degradation and in some cases there may be a performance improvement.

Deduplication benefits depend on factors like data type, the size of data being processed and the algorithm used to process the data. The performance of the algorithm is dependent on how efficiently duplicate data chunks are stored and restored when retrieved. This is where vendors can vary in the performance impact of an array and is why data reduction must be part of any performance measurement to get accurate results.

Pre-conditioning

Pre-conditioning can be a challenge in filling up an array when inline data services like deduplication and compression are enabled. Therefore, tools that can generate unique uncompressible and un-dupable data are key to the success of filling up an array. The process to measure data reduction technologies like deduplication and compression is highly dependent on the data content pattern. This means that measurements need data sets to be created that are compressible, non-compressible, repeatable, and non-repeatable to measure the effectiveness of deduplication and compression. Make sure when performing pre-conditioning that each LUN has been written to multiple times to ensure that both the primary and reserve capacity has been written to at least once to ensure garbage collection has started.

There are some array products designed to mitigate garbage collection in such a way that it would be rare to see this operation materially affecting the performance of the array under normal use. This means that pre-conditioning to the point of activating garbage collection can become quite difficult. Therefore pre-conditioning the full usable capacity, including spare capacity, of the array may be sufficient to run a steady state measurement. It is also recommended by some to pre-condition by write over the full plus spare capacity multiple times. Using a heavy write workload, like a 100% random write or a 60% write 40% read type work could be used to pre-condition the array.

Capacity utilization and data set Size

Performance based on capacity utilization is a factor to consider when setting up a data set to measure against. In the case of measuring performance with rotating media, some use ‘short stroking’ as a means of getting better performance. Short stroking is when the outside third of the disk is used to shorten the stroke of the mechanical arm to read the data. This cuts down on seek times and increases performance and response times. Most industry-standard storage benchmarks either require or

encourage higher capacity utilization to demonstrate performance closer to what would be observed in a production environment. There is a similar situation with solid state arrays when storage usage exceeds 80% of capacity. Even though there are no mechanical arms, the amount for free capacity available to write means the array does not have to work as hard if a large percentage of the capacity is not utilized. This could be looked at as a form of short stroking.

To achieve inflated unrealistic performance results, some vendors use data sets so small that most if not all of the I/O never make it out of RAM memory and, therefore, never touches the persistent media. Unless the data set was designed to work entirely from memory, like an in-memory database, these results are not a fair representation of the array's performance. This information should not be used for performance planning or any representation of how well the array will perform in a production IT environment.

Measuring Steady State

Measuring steady state performance for SSS arrays can vary. A longer steady state measurement can give the test sponsor an opportunity to look at the true behavior of the array over time. Since some SSDs behave better than others, this can be shown through performing longer (hours, days, a week, etc.) measurements. Many tend to take shorter (minutes) measurements and will get a valid result but the opportunity to show steady state performance over the course of a day or a week is lost.

Types of performance measurements

Some measurement tests characterize performance; other measurement tests demonstrate the maximum performance and throughput the array can produce under certain workload response time constraints based on average industry application response time tolerances.

Basic Characterization

A basic performance characterization of a storage array gives a picture of how well the array performs and behaves. The basic factors that affect performance are access pattern (random vs. sequential), read/write mixture, I/O transfer size, number of LUNs, outstanding I/Os, number of threads to drive the system under test, and queue depth. There are other factors like cache hit ratio, skew, etc. but we are going to stick with the basics in this section.

- **%read/%write performance curve** – Varying the percentage mix from 100% read to 100% write can give indicators that some workload mixes may not behave or perform well which could warrant more investigation. For example, a six point read/write curve would be 100/0, 80/20, 60/40, 50/50, 40/60 20/80, 0/100. Most curves are usually four to six points, but if time is no object a 10-point curve can really show the best and worst points of the storage array at certain

read/write ratios. This is great information for engineers in development to expose where improvement could be made in the development of the array's design. The curves can also expose how well garbage collection is performing.

- **I/O transfer size** – Most random workloads, like OLTP or email, usually use small transfers (8k, 32k, etc.) for better performance. Most sequential workloads use larger block sizes (128k, 256k, etc.) for better throughput. There are a few customer workloads that have what they call large block random workloads (32k or 64k).
    - Random performance – use a small transfer size usually less than 64K
    - Sequential performance – use a reference transfer size 64k or greater
- **Random-ness/Sequential-ness** - Solid state storage arrays are known for handling random performance very well, so the majority of interesting measurements taken are with 100% random I/O. But applications such as databases, where there is a component of sequential-ness in the workload, are also useful. Redo logs usually take up about 10% of the load in an email/messaging environment so there may be a need to make the workload more complex by introducing some measure of sequential-ness to characterize what happens to performance when this variable is introduced like it would in a production environment.
- **Outstanding I/Os** – This number can become very important and is the dominant intensity parameter for most benchmarks. Make sure there is the right balance of outstanding I/Os so that all queues are comfortably full to maximize driving the system under test (the storage) to its maximum performance, without having I/Os waiting in the queue so long that it starts affecting overall response times. Sometimes this can be a delicate balance. For example, in a typical OLTP database workload outstanding I/Os in most cases are much lower than most other applications to get maximum performance.
- **Metrics** – In a performance characterization, there are a few metrics to collect for analysis to examine the behavior of the storage system under test.
    - **Performance (IOPS)** – This metric is important for measuring random workloads and is what reports the measure of performance.
    - **Throughput (MB/sec or GB/sec)** – This metric is necessary for measuring sequential workload and is what reports the measure of bandwidth or throughput.
    - **Latency (micro- or milli-seconds)** – The latency reports the time it takes for an I/O request to be serviced within the storage
    - **Overall response time (milli- or seconds)** – The response time is the amount of time it takes for an I/O request to be serviced round-trip from the workload generator or application.

There are several sets of performance curves that are beneficial to produce to understand the basic performance and behavior of a solid state array. Once results have been collected, there are a couple of interesting charts that help visualize the behavior of the solid state array. One is the Read/Write

percentage curve mentioned near the beginning of this section; the others are a performance/response time curve and throughput/response time curve. These curves are useful not only for looking at the overall behavior of the array but also for exposing performance at various response times. There may be a data point that shows great performance but it may be at a response time that would not be tolerable to an end user.

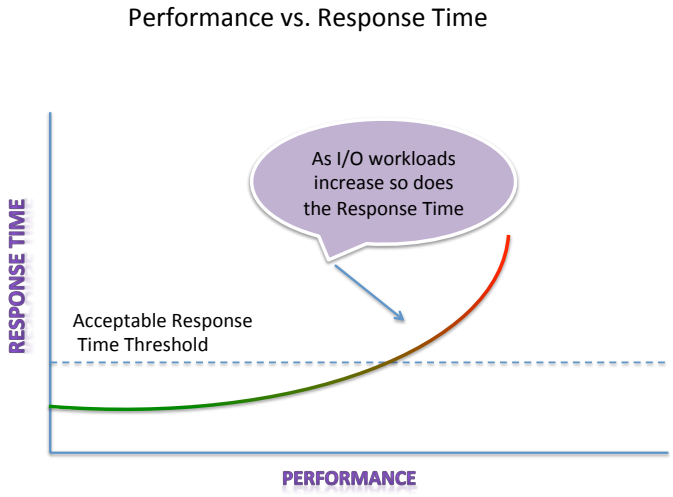


Figure 1 – performance/response time cure

There are other dimensions to measure that are not discussed in this paper that are beneficial performance factors to measure. These include:

- scaling up the number of devices on a controller
- scaling out the number of controller nodes
- software and efficiency upgrades for comparison of old versus new
- changing solid state media
- Single-node versus clustered (performance and throughput per node)
- Number of LUNs
- Degraded performance due to failure or interruption

Benchmarking

Unlike basic characterization, benchmarking is a type of measurement that shows maximum performance under a particular workload with fixed variables. Industry standard benchmarks are usually the most popular in this area because it is a fixed workload that most vendors have agreed to call a standard of measure. The tool comes in the form of a fixed benchmark kit that everyone agrees to use.

Coupled with a certified independent audit, these results can be used in product comparisons. The amount of information in Full Disclosure Reports (FDRs) has good supporting information that not only reveals how well the array performs but also how well it behaves.

Testing and Validation

Testing the limits of an array or validating its performance and functionality is another reason to use a workload generator. This activity can use I/O workload generators, as in the case of basic characterization, but instead of performing measurements to understand the performance behavior and design of the array, it is used to test (pass/fail) or validate claimed performance. The guidelines for measuring basic characterization can be used for test and validation but the focus of the information is different. These results are not good candidates for benchmark results but can be used for product comparison within the company if the variables are fixed.

Tools

It is very important to make the right tool selection and most tools today are not equipped to handle accurately measuring today's SSS array performance. Load generators like IOmeter were specifically designed to measure the performance of a single HDD internal to a server. There are other workload I/O generators that have not developed the additional capabilities to measure features like deduplication and compression. There are some tools like Vdbench, FIO, BTEST, and WorkloadWisdom, that have come up with ways to support the additional needs of deduplication and compression offering unique data sets to run from and offer knobs to control deduplication ratios. Virtual Instruments has been very interesting because of the complexity of the generated data sets and data streams. These data sets and data streams have been created to challenge the more advanced deduplication and compression algorithms.

In taking peak performance measurements, remember the bottleneck must be at the storage media. This means the neither the server running the workload generator nor application nor the network should ever be the bottleneck. The performance of these components needs to be monitored and tuned to ensure that performance is maximized at the backend storage media.

Performance Under Failure Scenarios

Even though failure scenarios are not part of the scope of this document, it is important to consider additional tests to demonstrate performance under failure conditions in proof of concept (POC) projects. The process discussed in the paper still applies in performing the actual measurement once a failure scenario is set up. The value of this data is crucial to understanding the real value and expected performance of SSS arrays under failure conditions. These scenarios are mainly measuring performance of Non-Disruptive Operations (NDO), which can include continued system operation in the face of:

- Failures – storage controllers, SSDs, I/O ports, I/O cards, and cables
- Upgrades – storage controllers, firmware, and software

Measurement Scenarios

To measure the System Under Test (SUT), in this case the SSS array, there are a few items that need to be considered in order to have reliable repeatable performance:

- All data reduction services, such as deduplication and compression, must remain enabled for the duration of the measurement period.
- A consistent environment must be maintained. Only one variable can be changed at a time from test run to test run. Changing more than one variable at a time will lose the accuracy of the results.
- Result reported should be an average of a steady state time period.

Performance results will largely depend on:

- Pre-conditioning stage and write history
- Workload characteristics
- Data content pattern

When executing a test scenario to measure performance, there are six main steps that need to be followed in order to get accurate performance information.

1. **Data Content Pattern** – A data set with appropriate data content patterns to stress deduplication and compression must be created. The characteristics of both the data set and data streams can also impact how the array performs.
2. **Common starting point (Fresh out of the box (FOB))** - Start the test by first placing the storage into a known, repeatable state.

3. **Pre-conditioning** – Conditioning SSS puts the storage in a "used" state. Attempting to measure performance without conditioning will result in artificially high performance results that are not sustainable. Depending on the SSS array, cells may have to be written multiple times in order to ensure all cells in the reserve capacity is written to at least once.
4. **Transition period** – Brand new devices, in particular, will report inflated performance through what some call a transient state of temporary inflated performance. Once that period of time has passed, the real performance of SSS begins. That is why pre-conditioning is so important.
5. **Steady State** – Measurements are taken only when key performance metrics are relatively time invariant and variability has stayed within a defined acceptable range representing a steady state. At this state a measurement interval can be taken to report the real performance of the array
6. **Ramp down** – Once the measurement period is complete from steady state the load activity can be brought down to an idle state.

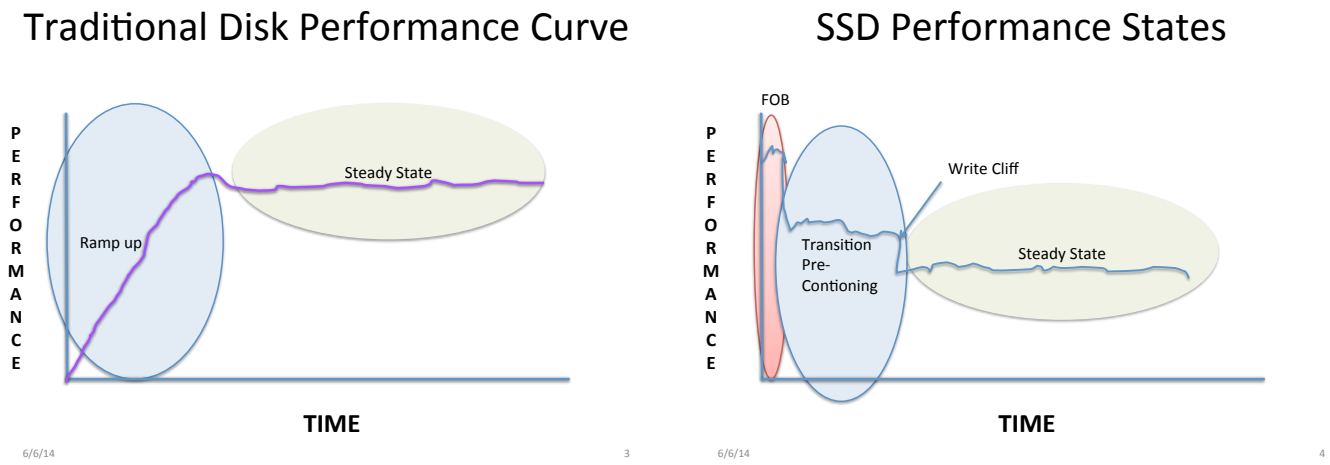


Figure 2 – traditional disk performance curve vs. SSD performance curve

Data Content Patterns

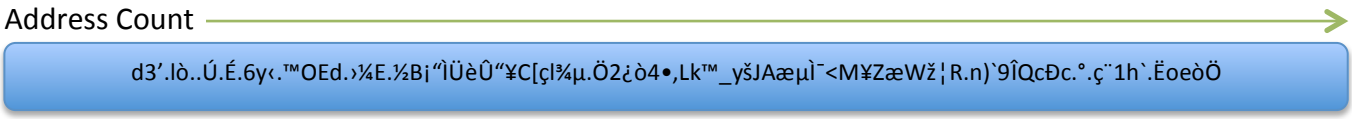
If a real production data pattern is not being used for the measurement, a synthetic data content pattern and data stream needs to be generated. Working in conjunction with Virtual Instruments, four data content patterns along with composite patterns made up of the four data content patterns were developed and validated for use with synthetic workload performance measurements. These data patterns are either one data type or two data types with interleaving sequences of blocks. Available data pattern types are:

1. Non-duplicable and non-compressible random data
2. Duplicable data (non-compressible) using seeded random data
3. Duplicable data and compressible data patterns using repeated patterns 0x00h or 0x11h
4. The degree of compressibility (inserted zero hex characters (0x0h)) can be merged into pattern 1 and pattern 2 below. Some arrays now treat zeros as special, so a different byte may have to be used to test compressibility.

The following data patterns shown below are examples of the kind of data that can be used to create data patterns to cover a whole dataset used over the whole array.

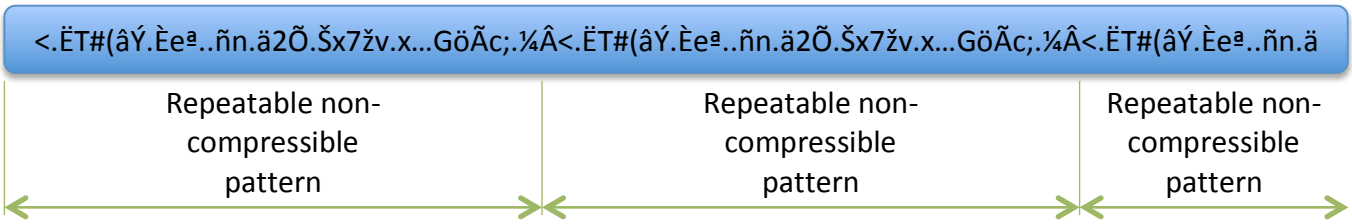
Data Pattern 1

This random data pattern contains non-duplicable, non-compressible random data. This data pattern has one data type and each block sequence is unique along the whole data pattern.



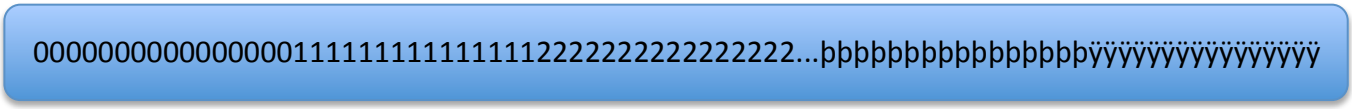
Data Pattern 2

This data pattern contains de-duplicable and non-compressible random data. A seeded random data content pattern is used for duplicable data but not compressible data. The length of the repeatable patterns can vary from 512 byte to 100s of GBs.



Data Pattern 3

This data pattern contains de-duplicable and compressible random data and the length of the content patterns can vary. Block size is also a factor that should be considered.



Data Pattern 4

This data pattern is a generalization of Data Pattern 1 and 2 that allows for predictable degree of compressibility. It is achieved by inserting sequences of zeroes of variable length ranging from 0 to 1K bytes in the otherwise random and uncompressible data.

©:.¼Â#ûaí™‘L`“··èSÈÛ4000000000”º½¡QPÃqrÛÞÐ(¨á8.koh00000000¢‡¥ËPHl.éÏÒe($6.Qoz 0000000

Composite Patterns

A composite pattern is formed by selecting data from 4 data patterns and writing them at random or pre-defined locations with relative weights W_1, W_2, W_3 for data patterns 1, 2, and 3 respectively. There are two composite patterns that were tested and validated by WorkloadWisdom.

- **Composite Pattern 1** – This is data pattern 2 with α ≈ 1. This means a random non-duplicable and non-compressible byte stream. It is used for both pre-conditioning and testing.

  α = Used capacity after reduction / used capacity before reduction

- **Composite Pattern 2** – This a composite of data patterns 1, 2, and 3 chosen for preconditioning (see below) are 0.2, 0.6, 0.2 and the weights chosen for write operations testing are W_1=1/3, W_2=1/3, and W_3=1/3. The spatial distribution is uniform within allocated logical address space.

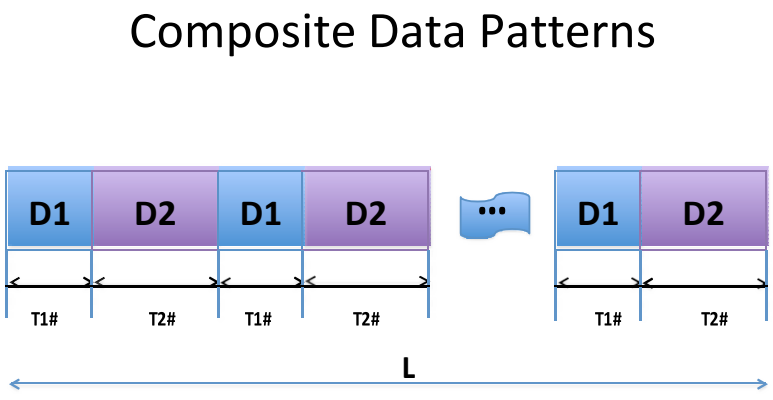


Figure 3
Mixed data pattern with D1 and D2 representing Data types; T1 and T2 is the length of each data type Segment and L is the full length of the dataset

Measurement Tests

The performance measurements are performed for each of the three data patterns discussed above. Each test has the following steps:

1. Common Starting Point - Purge the storage on the array
2. Pre-conditioning – 128K sequential writes written to the entire LBA space twice. In the SNIA SSSI measurement specification they call this the Workload Independent Pre Condition (WIPC) stage.
3. Steady state – Different workload data points can be collected varying the read/write mix, the block size, etc. This stage is where the measurement interval is taken.

The series of measurements that were run with WorkloadWisdom were based off of the SNIA SSSI measurement specification. The Workload Dependent Pre Condition (WDPC) consists of operation loops. The measurement guidelines below are a series of 25 measurements each having a duration of five minute measurements in steady state. Block size and the read/write ratio varies for each measurement.

Performance (IOPS) Measurements

For (Round = 25)
  For (R/W Mix % = 100/0, 95/5, 65/35, 50/50, 35/65, 5/95, 0/100)
    For (Block Size = 512B, 4KB, 32KB, 64KB, 1024KB)
      -Execute random IO, for the (R/W Mix %, Block Size), for 5 minutes
      -Record measurement

Throughput (MB/sec) Measurements

For (Round = 25)
  For (R/W Mix % = 100/0, 50/50, 0/100)
    For (Block Size = 128KB)
      - Execute sequential IO, for the (R/W Mix %, Block Size), for 5 minute
      - Record measurement

Latency Measurements

For (Round = 25)
  For (R/W Mix % = 100/0, 65/35, 0/100)
    For (Block Size = 512B, 4KB, 8KB, 32KB, 64KB)
      -Execute random IO, for the (R/W Mix %, Block Size), for 5 minute
      -Record measurement

Summary and looking forward

Measuring SSS array performance differs from the way we measure HDD arrays. More advanced arrays have implemented real-time data reduction as a part of processing a data stream and this must be reflected in measuring the performance of these arrays. In addition, advances to mitigate overhead like garbage collection are also unique factors that have to be considered when performing a measurement. But as SSS arrays make advances to be more efficient and higher performing, the basic process to measure performance still remains the same.

SSS arrays are still evolving and maturing and are becoming the new enterprise primary storage for the datacenter. Data reduction technologies like deduplication, compression, and thin provisioning offered in solid state arrays are a critical part of lowering the total cost of ownership (TCO) in the areas of capacity optimization and energy efficiency. The real value of these all-flash arrays is based on price/performance balanced with advanced storage technology features, better efficiency, consistency, and reliability. This high performance technology is growing in importance for mission critical applications, as well as virtualized and cloud infrastructures.

Thank You Reviewers

Thanks so much for taking the time to review the document

- Rob Commins, Tegile
- Steven Johnson, Oracle
- Miroslav Klivansky, EMC XtremIO
- Lou Lydiksen, Pure Storage
- Peter Murray, Virtual Instruments
- Chuck Paridon, HP
- Carlos Pratt, IBM

References

http://www.snia.org/tech_activities/standards/curr_standards/pts

www.evaluatorgroup.com

About Evaluator Group

*Evaluator Group Inc. is dedicated to helping **IT professionals** and vendors create and implement strategies that make the most of the value of their storage and digital information. Evaluator Group services deliver **in-depth, unbiased analysis** on storage architectures, infrastructures and management for IT professionals. Since 1997 Evaluator Group has provided services for thousands of end users and vendor professionals through product and market evaluations, competitive analysis and **education**. **www.evaluatorgroup.com** Follow us on Twitter @evaluator_group*