SNIA
Solid State Storage Initiative

Storage Performance Benchmarking Guidelines - Part I: Workload Design

May 2010

SSSI Member and Author:
Chuck Paridon, Hewlett Packard Corporation

The Solid State Storage Initiative

About SNIA

The Storage Networking Industry Association (SNIA) is a not-for-profit global organization made up of some 400-member companies and 7,000 individuals 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, 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 http://www.snia.org.

Introduction/Background

One of the main dilemmas facing IT customers and sales teams is the correct sizing of an enterprise storage solution to meet a customer's performance and capacity requirements. This task is especially unwieldy if there is no previous storage installation in place that can be used as a model or scaling element to predict the full solution's fitness to meet the customer's needs. In this case, a model workload or benchmark is either invented or selected from an existing collection to represent the operation of the actual application. The behavior of this benchmark is then related to the application and corresponding scaling factors are applied to the resources used in the test to determine the actual needs of the installation. Lastly, the deployment of emerging technology, such as Solid State Storage (SSS) with much more robust throughput capabilities, will have a profound effect on accurate storage sizing.

The purpose of this paper is to cite and recommend methods to be used for the determination of an appropriate storage performance focused benchmark. The implicit assumption throughout is that the remainder of the benchmarking infrastructure (servers, network, etc.) is sufficiently robust to force the storage element to be the bottleneck or performance limiting resource.

> ***Editor's Note:*** *This paper deals with the design aspects of a workload suitable for providing the stimulus in a storage performance benchmarking environment. There will be two complementary papers to follow. Part 2 will delve into the benchmarking data collection and reduction processes. Part 3 will concentrate on the presentation of benchmark results.*

Benchmarking Fundamentals

As cited above, the goal of benchmarking is to determine the suitability of a solution (in this case the storage portion) to solve a need. A benchmark must be defensibly representative of the actual application under consideration. The remainder of this section is devoted to the enumeration of attributes to be considered when determining the fitness of a benchmark.

Workload Parameters

Workload parameters are the collection of input/output (IO) stimulus attributes that must be chosen such that they truly represent the IO transactions presented to the storage by the real application. These are presented in no particular order, as each can have a profound effect.

Block Size

The block size is the quantum of data for each IO transaction requested by the host. The correct choice of this parameter will expose the storage subsystem's ability to process the required number of transactions to support either an IO per Second (IOPS) load or the corresponding throughput in MB per sec (MB/s). Some typical examples of block sizes are 8KB (this is an Oracle and/or file system transfer quantum), 64KB (this is a typical transfer quantum for a backup/restore process), and 256KB (this is a streaming video application transfer quantum).

Errors that could be introduced into the benchmark through the incorrect selection of this parameter include an optimistic estimate of throughput (MB/s) if the block size chosen is too large, or an optimistic estimate of the IOPS rate if chosen too small.

Read/Write Ratio

Every byte of data read from a storage device has to be previously written, and in order to achieve any value from writing data to the subsystem, it must be subsequently read. The point is that these operations coexist on the storage and must be modeled accordingly. In general, writes are more expensive than reads due to the level of protection offered to new data by the subsystem and the fact that writes to SSS devices are in general slower than reads because of programming and wear leveling operations. Because these two operations treat storage resources differently, the proper ratio of these in the stimulus is important. A few typical values are 100 % read (emulating a backup operation), 100% write (emulating a restore operation or a database log file) and 60% read, 40% write (database transaction mix).

Errors that could be introduced into the benchmark through the incorrect selection of this parameter include an optimistic estimate of application throughput (MB/s) by choosing an inappropriate heavily biased read ratio, or pessimistic throughput estimates by choosing too heavy of a write component.

Access Patterns

In practice, IO access patterns are as numerous as the stars. The most accurate way to determine an application's access pattern is to measure it. However, one of the underlying premises of this paper is that we do NOT have an instance of the relevant application currently in operation; therefore simplifying assumptions are in order. For the sake of tractability, we'll consider only two access patterns: random and sequential. Random access is exemplified by online database transactions where data reads and modifications are made in a uniformly scattered manner across the entire dataset. Sequential access is typical during back-up and restore operations as well as event logging.

Errors that could be introduced into the benchmark through the incorrect selection of this parameter include an optimistic estimate of random application throughput (MB/s) if sequential access is incorrectly chosen for the workload, because most storage subsystems perform optimizations (such as pre-fetching or full page writes when using SSS) on sequential access streams.

Working Set Size

The working set of an application is defined as the total address space that is traversed (either written and/or read) in some finite, short period of time during its operation. The importance of this parameter is amplified if the storage subsystem has sufficient cache capacity to hold a significant portion of the working set. An incorrect choice of this parameter can result in an order of magnitude difference in performance on the same hardware. Cache access times differ from rotating media access times by at least an order of magnitude. Choosing a working set size that is insufficient will surely result in an overly optimistic application performance estimation. Of all the parameters discussed thus far, this is the most difficult to estimate, in addition to having the most profound effect on performance. More on how to deal with this in a conservative manner later.

IO Demand Rate

The IO demand rate parameter can take on one of two units of measure depending on the transfer block size. One can be derived from the other.

In general, for IO transfer sizes less than 64KB, the IO processing rate (IOP) is the unit of choice. This is so because smaller transfers are limited by the transaction processing rate of the storage rather than the raw bandwidth. The demand of OLTP workloads, due to their relatively small block sizes (8-16KB), is expressed in these units. One can think of this parameter as the analog to a toll booth on a freeway. Regardless of the speed limit (bandwidth), unless enough vehicles can be processed by the toll booths, there will be performance problems.

The other unit of measure is MB/s. In general for IO transfer sizes greater than 64KB this is the measure of choice because larger transfers are limited by the bandwidth of the storage rather than the IO processing capability. Backup and restore workloads are expressed using this unit of measure.

Outstanding Request Population or Demand Intensity

This parameter represents the degree of IO demand parallelism presented to the storage by the application. The influence of this item is quite profound, especially when using a random, small block workload. Unless there is sufficient demand on the LUNs, the addition of more spindles or SSS devices has no effect on performance. When storage systems are sized for a particular application, this term is often overlooked. This oversight generally results in observed performance being much lower than estimated, because the storage is sized for the device's capability, not the host's level of demand. Like the working set size, the estimate for this parameter is best gleaned from the application vendor or the system administrator. Some applications, such as Oracle, allow the Database Administrator (DBA) to select the degree of parallelism employed during table scans, etc., so the DBA may be the best source of this data. A great deal of Fear, Uncertainty, and Doubt (FUD) can be generated by choosing too small a value for this item during the execution of a benchmark.

Assembling a Representative Workload

Working form an Existing Installation

The most accurate way to determine the choice and combination of the above parameters is to measure a running installation and choose values from it. Once this has been done for all of the application IO demands to be satisfied by the storage, a defensible aggregate workload can be developed. Some of the parameters are straightforward to measure. For example, the UNIX SAR utility can be used to measure the I/O block sizes and the rate of demand, whether expressed in IOPS or MB/s. In addition, some versions of SAR split out the reads from writes so the ratio can be determined.

Access patterns are generally known by the nature of the application. OLTP causes random access while backups and restores are sequential. Oftentimes there are "hot spots" or small data regions with frequent accesses in applications. These patterns can generally only be determined through a precise trace and analysis taken from the running application. The determination of the working set size is also only achievable through a detailed trace and analysis.

Additional and sometimes more comprehensive data collection facilities reside on the storage subsystem. For example, Performance Advisor on the HP XP family of products can be used to capture all of the aforementioned parameters, with the exception of the working set size. The challenge of using such a tool is to ensure that one is capturing the attributes of a workload in isolation. This is generally not possible on storage subsystems accommodating consolidated, simultaneous applications.

After the collection of the parameters for all of the simultaneously executing applications is complete, like workloads are grouped together and their demands summed to determine the new aggregate workload description. For those workloads that are disjoint (IOs are different enough that they cannot be consolidated), a separate IO stimulus process can be created and run in parallel.

An example of developing a small synthetic benchmark

The following sample data was collected from four hosts, each running a single application. The objective is the development of a synthetic workload that can be used as a stimulus to assess the performance of proposed storage solutions.

	APPLICATIONS			
	#1 OLTP	**#2 BACKUP**	**#3 DSS**	**#4 RESTORE**
Block Size	8 KB	64 KB	8 KB	64 KB
Read Write Ratio	60/40	100/0	80/20	0/100
Access Pattern	Random	Sequential	Random	Sequential
IO Demand Rate	2,000 IOPS	25 MB/s	3,000 IOPS	50 MB/s
Working Set Size	TBD	TBD	TBD	TBD

Assumptions**:**

1. The DSS and OLTP operate on the same space simultaneously
2. The restore and backup are two different address spaces

The consolidation of this workload will result in 3 stimulus threads: One for the backup, one for the restore, and one for the combination of the OLTP/DSS workloads.

	WORKLOADS		
	#1 OLTP + DSS	**#2 BACKUP**	**#3 RESTORE**
Block Size	8KB	64KB	64KB
Read Write Ratio	72/28	100/0	0/100
Access Pattern	Random	Sequential	Sequential
IO Demand Rate	5,000 IOPS	25 MB/s	50 MB/s
Working Set Size	TBD	TBD	TBD

Benchmarking Pitfalls, Tricks and Lies

In general, the goal of benchmarking is to determine the suitability of a solution (in this case the storage portion) to solve a need.There are certainly exceptions to this rule. Oftentimes, a storage vendor knows well the strong/weak points of his/her product and those of the competition. The unscrupulous vendor will attempt to convince the customer of the applicability of the strong points of his product and the weak points of the competition for the customer's needs.

This type of behavior must be held in check by the competition. It's intuitively obvious then that we must be on guard for those types of stimuli that are not representative of the application(s) under consideration. The remainder of this section is devoted to the enumeration of some of the more blatant violations of tailoring a benchmark to fit a particular vendor's product, rather than solving a customer's need.

We've learned the definitions of some of the key benchmarking parameters and how they play a part in product assessment. Now let's consider how they can be exploited to show a better result from an inferior product

Workload Parameters

Block Size

The block size is the quantum of data for each IO transaction requested by the host. Let's assume the competitive vendor's product has an IO processing rate (IOPS) deficiency, but a sufficient throughput rate (MB/s). In this case, an erroneous larger block size can be chosen and the throughput in MB/s used as the index of merit in the results. When the customer deploys this, he sees performance much lower than what was actually expected. The blocks are smaller and the IO transaction limit is choking the throughput.

In a like manner, a product having internal transfer bottlenecks can be presented in a much better light by using very small blocks and reporting the results in terms of IOPS. Again, the customer will be quite disappointed when this solution is actually deployed.

Read/Write Ratio

As mentioned above, different storage subsystems handle reads and writes very differently. This discrepancy can be used to the advantage of the unscrupulous benchmarker. In general, a saturated write stimulus creates much more housekeeping than a saturated read workload. By choosing a larger read/write ratio, the storage can be coerced to perform better, per the benchmark, but this may not represent real performance under customer workload.

Alternately, some storage subsystems cache all writes and if a hit is generated, return the acknowledgement to the host at almost solid state response times. If some of the read hot spots in an application are modeled as writes, optimistically high throughput can be obtained.

Access Patterns

As mentioned earlier, the variability in access patterns is infinite. The consolidation of applications onto a single storage frame makes these patterns even more unwieldy. As a result, the legitimacy of the access patterns used in a synthetic benchmark may be very hard to determine. An unscrupulous benchmarker may take unfair advantage of cache assistance by artificially creating hot spots or regions of spatial locality of reference.

Working Set Size

As cited earlier…"Of all the parameters discussed thus far, this is the most difficult to estimate, in addition to having the most profound effect on performance." For those who have seen the marketing numbers for many high end storage arrays, this is the explanation of how those very high figures were obtained. If we choose a very small working set size, let's say a 32KB address space, the entire working set will fit within the host facing processor cache on these devices. Now, taken in combination with a very small block size (1KB, lets say), this working set will result in a stimulus capable of producing over 2 million IOPS on a typical high end storage subsystem. In terms of competitive comparison, these figures are stellar. In terms of value to the customer, they're worthless.

Benchmarking Guidelines: Conclusion

In conclusion, the goal of benchmarking is to determine the suitability of a solution (in this case the storage portion) to solve a need. A benchmark must be defensibly representative of the actual application under consideration. In order to achieve this, the benchmark must take a number of workload parameters into account, such as Block Size, Read/Write Ratio, Access Patterns, Working Set Size, IO Rate Unit of Measure, and Demand Intensity. Having established a parameter model, benchmark designers and reviewers must be aware that there are many "tricks" that can be played to artificially inflate the performance of storage subsystems through workload manipulation. The best way to safeguard against having to compete under these circumstances is to ensure compliance of the stimulus to that of the actual application in all of the categories cited above. must be on guard for those types of stimuli that are not representative of the application(s) under consideration. The remainder of this section is devoted to the enumeration of some of the more blatant violations of tailoring a benchmark to fit a particular vendor's product, rather than solving a customer's need.

SNIA

Storage Networking Industry Association
425 Market Street, Suite 1020 • San Francisco, CA 94105 • Phone: 415-402-0006 • Fax: 415-402-0009 • www.snia.org