SYMANTEC SOLUTIONS ASSESSMENT GUIDES

Symantec Storage Foundation and High Availability Solutions SmartIO Assessment Guide for Linux Platforms

Ritesh Shah

September, 2014

Table of Contents

Introduction

This document discusses how to analyze the impact of Symantec's SmartIO feature. It also includes performance statistics.

In today's world of higher performance Service Level Agreements (SLAs) and shrinking budgets, it becomes very important to make better use of existing resources. With SmartIO, administrators can improve performance by introducing onboard Peripheral Component Interconnect (PCI) solid-state drives (SSDs). By introducing SmartIO in the Storage Area Network (SAN), the administrator can ease data congestion and improve I/O response times across the SAN.

SmartIO uses fast SSD storage devices for increasing application speed by caching the hot data on the accessing node. In line with Symantec's support for heterogeneous environments, SmartIO is SSD agnostic, and you can use it on PCI SSD cards from a number of vendors.

SmartIO terminology

This document uses the following terminology:

Cache area A dynamic storage region reserved for caching. For the current release, there can be only one cache area of each type on a system.

Caching modes The caching mode used for SmartIO. SmartIO has three caching modes; however, the caching mode can be either at the volume level or file system level. The caching modes are:

Volume level read cache: This block-based read cache can be used for applications that store their data on raw volumes. For Storage Foundation and High Availability release 6.1, this mode is not supported for shared volumes.

File system level read cache: This is a file-level read cache for applications that store their data in files on a file system. This configuration supports very granular data caching, and you can also use it in a clustered file system (CFS) environment. Currently, there is no metadata caching support for file system level caching.

Write-back cache: This is a super set of file system read cache, and allows writes to be buffered on SSD devices. Using this caching mode, the application can improve the performance of latency sensitive I/Os, such as Oracle redo log writes.

Data object The object for which caching is enabled, either the Symantec file system or a Symantec raw volume.

Persistent read cache A read cache in which cache data survives a system power cycle. Also known as warm cache.

Read cache A caching solution designed to accelerate read I/Os. The solution caches the reads on a system and any re-reads are served from the cache.

Volatile read cache A read cache in which cached data is lost after a system power cycle.

Write-through The data is updated on both the cache and the disks. This means that if the system crashes, the cache does not contain any dirty data.

Caching basics

SmartIO is an I/O caching solution that reduces the amount of I/O that goes to the storage. This in turn improves the performance of other applications and also increases the number of applications that the existing infrastructure can serve. For these benefits to be evident, you need to take several factors into account. The factors listed below relate to caching and SSD in general and are not specific to SmartIO.

- **The existing environment**: SmartIO reduces the I/O bottleneck on the SAN. SmartIO improves I/O latency. This is achieved because of the difference in I/O response times of SAN and SSD used. If the SAN is very fast (for example, SSD storage), using SSD for caching may not show a very evident benefit in performance, although the amount of data going down to the SAN is still reduced.

- **Your I/O requirements**: If read latencies are not critical for the application, improving read performance by introducing SmartIO may not show an increase in performance. However, the reads going down to the array are reduced, so you may see benefits for other applications using the array.

- **The mount of cached data that gets reused**: If the data being cached is not reused, then SmartIO may not show a benefit. The amount of data reused depends on the application. For example, an application that does sequential reads and does not reference the data again does not see a benefit from SmartIO. On the other hand, an online transaction processing (OLTP) environment, which has a limited dataset and reuses the data, is expected to benefit from SmartIO.

- **The “write cliff”**: For some SSD cards, a “write cliff” occurs under a sustained heavy write load and card performance drops. The write cliff is related to SSD technology, and different cards behave differently under a write cliff scenario.

- **Cache overflow**: If the active dataset is much larger than the size of the cache configured, the cache overflows. The data is cached and then ejected before it can be used. While SmartIO has built in intelligence to handle this scenario, increasing the cache area size helps.

- **CPU**: Regardless of how conducive a workload is for caching, if the server does not have enough spare CPU, using SmartIO may not increase throughput.

The following sections discuss ways to analyze the efficiency of SmartIO.

Analyzing SmartIO

You can measure the effectiveness of SmartIO using three sets of statistical tools:

- **Operating system statistical tools**: These tools, such as the iostat command, are commonly available and let you measure performance without involving the Symantec stack.

- **Symantec tools:** These tools include the Symantec vxstat and sfcache commands and let you evaluate SmartIO through the Symantec stack.

- **Application tools:** These are application-specific tools, such as the Automated Workload Repository (AWR) for Oracle databases, which show the impact of SmartIO in the environment.

As with all performance comparisons, you need a baseline to effectively compare the benefits of SmartIO. Also for SmartIO to make an impact, you should study the effect after SmartIO has been enabled for a reasonable time to allow cache population; otherwise, the conclusions will be misleading. The following sections describe how to collectively use the tools to get a better understanding of SmartIO.

Analyzing SmartIO using operating system tools

You can use tools like iostat and sar to study disk I/O as the operating system sees it. The most important metrics here are read operations per second, reads per second (KB/s), and disk response times (ms). As a secondary metric, you can also compare the write operations and writes (KB/s) to see the benefits of easing the I/O bottleneck. You can use these tools to check the performance of data disks as well as the SSD device.

Table 1 iostat output before introducing SmartIO

Device	r/s	w/s	rsec/s	wsec/s	await
sdc	1220.57	468.03	20175.47	11473.6	6.64

Table 2 iostat output after introducing SmartIO

Device	r/s	w/s	rsec/s	wsec/s	await
sdc	62.17	966.9	1288	21322.13	2.86
fioa	32565.8	16882.5	534785.6	300848.93	2.14

As the data above shows, the reads on data disks have decreased dramatically, which has also improved the disk response times. In addition to the decrease in reads, the writes increased. In this example, SmartIO has made a positive impact on the environment.

Table 3 iostat output before introducing SmartIO

Device	r/s	w/s	rsec/s	wsec/s	await
sdc	1220.57	468.03	20175.47	11473.6	6.64

Table 4 Iostat output after introducing SmartIO

Device	r/s	w/s	rsec/s	wsec/s	await
sdc	1200.57	468.03	20001.47	11473.6	6.64
fioa	32565.8	16882.5	534785.6	300848.93	2.14

In the example above, although there are significant writes to the SSD device, there is no reduction in the data read from the SAN. This seems to indicate that the application is not reusing the data it has read. To confirm this, we must use the sfcache command described in the Symantec tools section.

Analyzing SmartIO using Symantec tools

Similar to iostat, vxstat can be used to study the impact of enabling SmartIO. The important metrics are the same for both commands: read operations, blocks read, and service times.

To administer SmartIO, Symantec has introduced a new command, sfcache. This command lets you create, enable, and destroy the SmartIO cache area. sfcache also lets you collect statistics for SmartIO. The important metrics here are cache used, data read, data written, and cache hit ratio. If write-back cache is enabled, write back data also becomes important.

Table 5 Good cache hit ratio

Read cache hit ratio	**Data read**	**Data written**	**Write-back hit ratio**	**Data written**
/data_disks:				
90.84%	1.49 TB	100 GB	0.00%	0 kB

As the data above shows, the cache hit ratio is quite good. The application uses a lot of data that it reads in. Also, the writes are low. This means there are very few data updates happening. The above scenario happens when a mostly read operation is evicting data from the application cache. In this case, SSD significantly boosts performance.

Table 6 Low cache hit ratio

Read cache hit ratio	**Data read**	**Data written**	**Write-back hit ratio**	**Data written**
/data_disks:				
10%	2 TB	100 GB	0.00%	0 kB

In the scenario above, the cache hit ratio is quite low. This means that the application does not reuse the data it reads. In such a scenario, it is useful to look at disk statistics to see if the response times have improved.

Table 7 High write rate, high cache hit ratio

Read cache hit ratio	**Data read**	**Data written**	**Write-back hit ratio**	**Data written**
/data_disks:				
90.84%	1.49 TB	2.5 TB	0.00%	0 kB

This data indicates an update-intensive scenario, which means that write performance of the backend storage is still important. For such a scenario it is better to check application statistics to see if we have achieved a performance boost.

Application specific optimization

Any optimizations we bring into the environment should improve application performance. The final metric used to measure performance improvement is environment-specific; however, each application has a set of counters which you can use to measure the impact of SmartIO.

Oracle database: The final performance metric for each environment is different; however, there are some indications in the Oracle statistics. The wait events section in the AWR report shows the wait details for major event classes. Because SmartIO speeds up reads, the User I/O class shows the impact of SmartIO.

Table 8 shows the wait events for a properly-tuned database. Here, I/O wait consumes 94% of the wait. Table 9 shows the wait events for the same workload after SmartIO is introduced.

Table 8 Wait Events for a normally running Oracle database

Wait class	Waits	% Time outs	Total wait time(s)	Avg wait (ms)	% DB time
User I/O	90,446,123	0	684,590	8	94

Table 9 Wait Events for a database with SmartIO enabled

Wait class	Waits	% Time outs	Total wait time(s)	Avg wait (ms)	% DB time
User I/O	163,376,889	0	485,588	3	67

As the data shows, the time for user I/O (sequential, scattered, and so on) has decreased significantly. This means that even though number of waits has increased due to a higher number of transactions served, the total time spent waiting is reduced.

Conclusion

SmartIO is an I/O acceleration tool. It does this by caching the hot data on the host where it is required. This reduces the latency of reads by removing SAN overhead and also providing more predictable throughput by eliminating the impact of bursts within the SAN.

Although you can configure SmartIO on any device that is supported by VxVM, for maximum impact it is better to configure it on a device that is close to the CPU, such SSD PCIe cards.

21340929 10/14