MAKING DATA SMART

Qsan Document - White Paper

SSD Caching

Version 1.3
January 2015

Copyright

Trademarks

Qsan Technology, Inc.

4F., No.103, Ruihu St.,
Neihu Dist., Taipei City 114,
Taiwan (R.O.C.)

Tel: +886-2-7720-2118
Fax: +886-2-7720-0295

Email: Sales@Qsan.com
Website: www.Qsan.com

Introduction

Traditionally, data are stored on the HDDs (Hard Disk Drives) and SSDs (Solid-State Drives) are mainly used for mission-critical applications that demand high-speed storage systems. In recent years, the capacity of HDDs has increased, but their random input/output (I/O) has not kept pace. For some applications such as web commerce, clouds, and virtualization that require both high capacity and performance, HDDs, though capacious, simply are not fast enough.

SSD caching technology leverages the strengths of both HDDs and SSDs, to cost-effectively meet the capacity and performance requirements of enterprise applications. Data are stored on HDDs while SSDs serve as an extended cache for many I/O operations. A single chassis, therefore, can provide both the capacity and economy of HDDs and the blistering performance of SSDs.

Generally, SSD caching is particularly effective when:

1. Reads are far more common than writes in the production environment.
2. The inferior speeds of HDD reads cause performance bottlenecks.
3. The size of repeatedly accessed data is smaller than the capacity of the SSD cache.

The Solution

SSD caching is secondary cache that improves performance by keeping frequently accessed data on SSDs where they are read far more quickly than from HDDs. One or more SSDs can be assigned to a single virtual disk to provide the SSD cache. Note that the capacity allocated to the cache is not available for regular data storage. Currently, the maximum SSD cache size in a system is 2.4TB.

SSD Cache Implementation

When reads or writes are performed, the data from the HDDs are copied into the SSD cache. Any subsequent reads of the same logical block addresses will be accessed directly from the SSD cache. Therefore, response times are much lower, increasing overall performance. In the very unlikely event the SSDs fail, there is no data loss because the cached data are copies of the original data residing on the HDDs.

An SSD cache is divided into a group of sectors of equal sizes. Each group is called a cache block; each block is divided into sub-blocks. The cache block size can be adjusted to suit a specific application - such as a database or web server.

Populating the Cache

The actions that read data from the HDD and then write to the SSD are called populating the cache. Typically, this is a background operation that immediately follows a host read or write operation. As the goal of the cache is to store frequently accessed data, not every I/O operation should trigger a cache population, but only ones that pass a certain threshold, implemented as a counter. There are both a populate-on-read threshold, and a populate-on-write threshold.

1. Populate-on-read threshold: When the same data block to be read over the threshold, it is called hot data and populated to the SSD cache. The threshold must be great than zero. If it is zero, no action is performed for a read cache.
2. Populate-on-write threshold: When the same data block to be written over the threshold, it is called hot data and populated to the SSD cache. The threshold must be great than zero. If it is zero, no action is performed for a write cache.

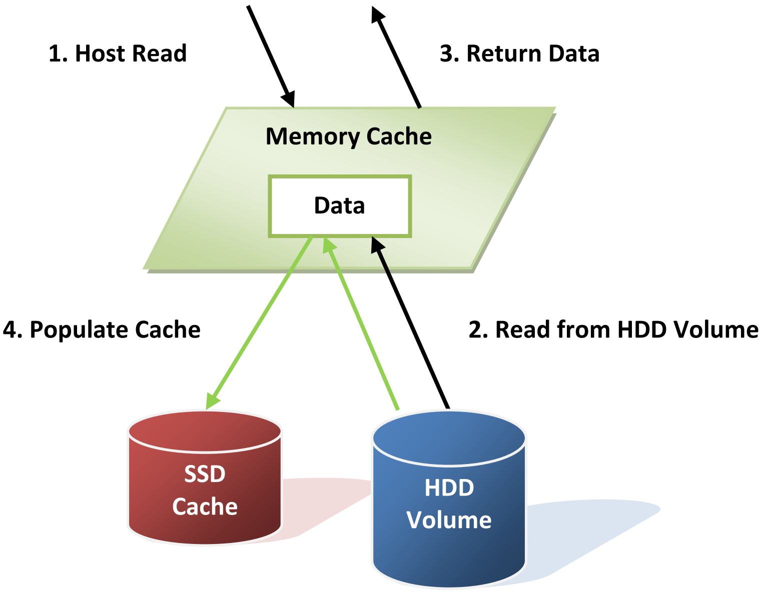
Each cache block on HDD volume has read and write counter associated. When a host requests to read data located in that cache block, the read count is increased. If the data is not found in the cache already and the read count is greater than or equal to the populate-on-read threshold, then a cache-populate operation is performed concurrently with the host read operation. If a cache hit occurs, the data is immediately returned from the SSD cache and a populate operation is not performed. If the read count is smaller than the threshold, a populate operation is not performed.

Write cases are the same scenario as read. Below, we provide figures to further describe read and write caching.

Read/Write Cache Cases

- Read Data with Cache Miss

  The following figure shows how the controller handles a host read request when some or all of the data are not in the SSD cache.

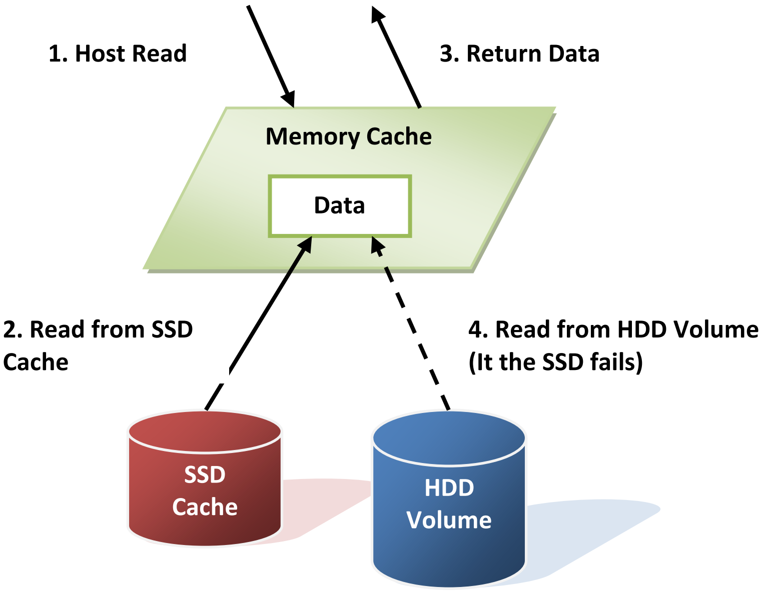


These steps are:

1. A host requests to read data. The system will check if the requested data is in memory cache or SSD cache. If not, it is called cache miss.
2. Data is read from the HDD volume because of cache miss.
3. The requested data is returned to the host. And the system will check whether the requested data is hot data.
4. If it is, the SSD cache is populated.

- Read Data with Cache Hit

  The following figure shows how the controller handles a host read request when the data is in the SSD cache.

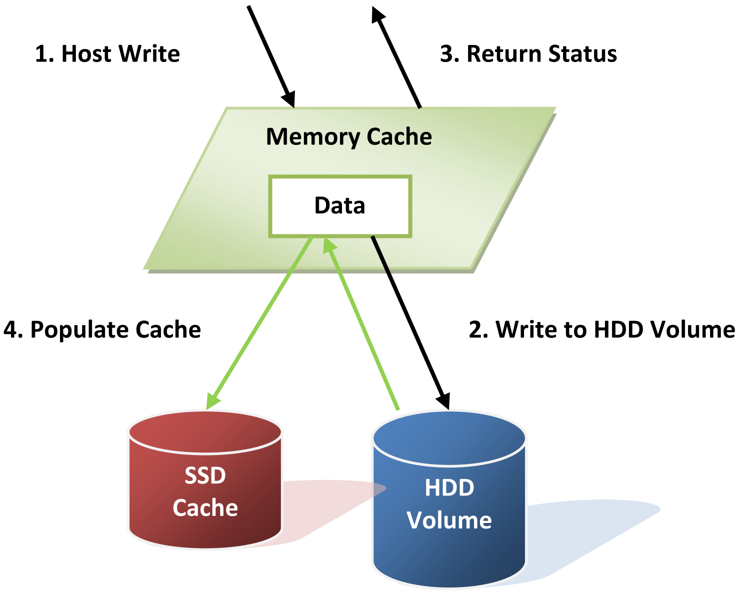


These steps are:

1. A host requests a read data. The system finds that the data is in SSD cache, so it is called cache hit.
2. Data is read from the SSD cache.
3. The requested data is returned to the host.
4. If there is an SSD cache error, data is read from the HDD volume.

- Write Data

The following figure shows how the controller handles a host write request.



These steps are:

1. A host requests to write data.
2. Data is written to the HDD volume.
3. The status is returned to the host.
4. The SSD cache is populated if the write threshold is reached.

SSD Cache Tuning

The SSD cache can be tuned to maximize its efficiency base on application usage. Cache block size, populate-on-read threshold and populate-on-write-threshold are the main parameters.

- Cache Block Size

A large cache block suits applications where frequently accessed data is close to each other, known as a high locality of reference. A large cache block will also fill up the SSD cache quickly - this is known as the warm-up time. After the cache is warmed up, the performance would be quite good for applications with high locality of reference. Such as the file system or web service usage, the frequently accessed data are based on some concentrated files which are usually in large block size. However large cache blocks will also generate larger I/O overhead, increasing response time, especially for cache misses.

A smaller cache block size suits applications with data that is less localized, meaning the data is accessed more randomly, such as database usage. The SSD cache will fill up slower, but with more cache blocks, there is greater chance of a cache hit, especially for data with less locality of reference. With a smaller cache block size, cache usage is usually less than with a larger cache block size, but overhead is less, so the penalty for cache misses is less severe.

- Cache Sub-block Size

  The cache block is divided into many sub-blocks. When any of the sub-blocks inside a cache block is accessed, the counter associated with the cache block will be increased. Once the read or write counter reaches the population threshold, the system will copy the entire cache block into the SSD cache. The cache sub-block size is used to decide how many sub-blocks are divided from the cache block.

- Population Threshold

  The population threshold is the number of accesses at which point that cache block is copied to the SSD Cache. A higher number ensures that the cache only stores frequently accessed data so there will not be much cache turnover however it also means the cache will take longer to warm up and be fully effective. A lower number means the cache is warmed up quickly, but may cause excessive cache populations. A populate on read threshold of 2 is sufficient for many applications. Populate-on-write is useful when data that is written to is often read soon after. This is often the case in file systems. Other applications, such as database software, may not have this tendency so populate on write may sometimes even be disabled.

I/O Type	Block Size (Sectors)	Sub-block Size (Sectors)	Populate-on-Read Threshold	Populate-on-Write Threshold
Database	1MB (2,048)	8KB (16)	2	0
File System	2MB (4,096)	16KB (32)	2	2
Web Service	4MB (8,192)	64KB (128)	2	0
Customization	1MB/2MB/4MB	8KB/16KB/64KB	≥ 0	≥ 0

As you can see there are tradeoffs for increasing or decreasing each parameter. Understanding the data locality of the application is essential and it can be useful to do some field testing to see what works best.

The Formula

We provide a formula that can estimate the warm-up time. The concept is that the warn-up time should be the period of reading/writing data from HDD to SSD. We define that:

- T: Warm up time; seconds required.
- I: Average random IOPS of one HDD.
- S: I/O Size.
- D: Number of HDDs.
- C: Total SSD caching capacity.
- P: Populate-on-read or Populate-on-write threshold

We assume that how fast the reading/writing data from the HDD to optimize the capacity of the SSD. If the hot data is full in SSD, the SSD caching should be effective. So the total SSD caching capacity times the populate threshold equals to the IOPS of HDD times I/O size and warn-up time. The formula looks like on the following.

C * P = I * S *D * T

We can estimate the warm up time, at least.

T = (C * P) / (I * S * D)

Real cases may be longer than the estimated time. Here is an example.

- I: 250 IOPS (Random IOPS per HDD in 64KB I/O pattern)
- S: 64KB (I/O Size)
- D: 16 (Total 16 HDDs for a RAID group)
- C: 480GB (1 SSD)
- P: 2 (Populate-on-read threshold, the data hits twice then go into SSD)

Warm-up time T = (480GB * 2) / (250 * 64KB * 16) = 3932.16 seconds = 65.536 minutes

We assume that after 65 minutes, the frequency used data will be read twice (populate-on-read threshold is 2), and all of them are populated to SSD. At the next step, if the same data is requested, it will be read from SSD. Although it is an ideal formula for reference, the warn-up time is not very accurate because the I/O pattern may be very random. Sometime, the warm up time may be a large deviation.

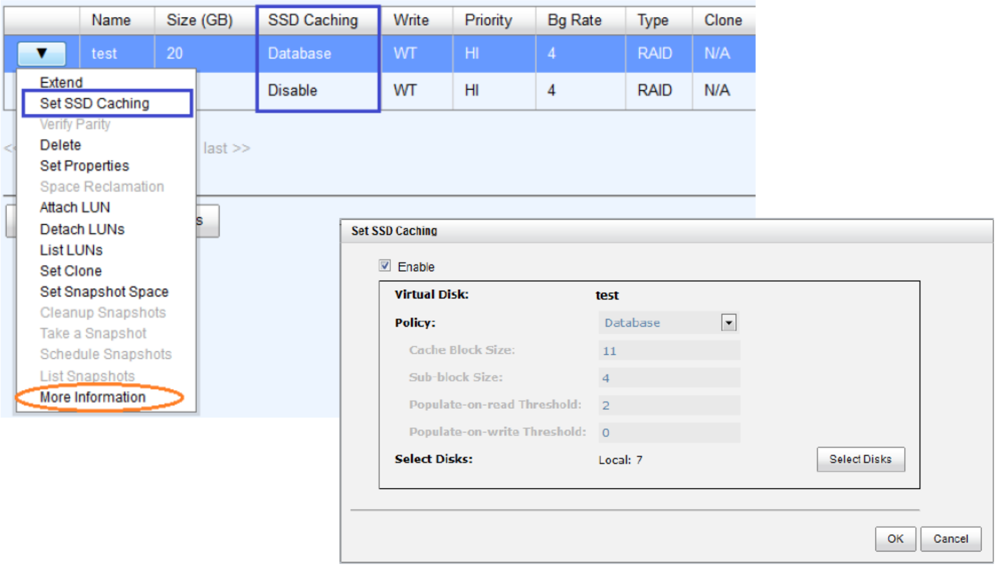
Configuration

Activate the license key

User needs to obtain a license key and downloads it to the system to activate the SSD caching function in **System Maintenance -> Upgrade -> SSD Caching License**. Each license key is unique and dedicated to a specific system. To obtain the license key, please contact sales for assistance.

Here is an example of enabling SSD caching.

1. Create a RAID group and then create a virtual disk.
2. Click▼ **-> Set SSD Caching** of the selected virtual disk.
3. Check the **Enable** box.
4. Select the policy by drop down menu.
5. Click **Select Disks**, and then check the SSDs that are provided for SSD caching.
6. Click **OK** button to enable SSD caching.



	Name	Size (GB)	SSD Caching	Write	Priority	Bg Rate	Type	Clone
▼	test	20	Database	WT	HI	4	RAID	N/A
			Disable	WT	HI	4	RAID	N/A

Set SSD Caching

☑ Enable

Virtual Disk: test
Policy: Database
Cache Block Size: 11
Sub-block Size: 4
Populate-on-read Threshold: 2
Populate-on-write Threshold: 0
Select Disks: Local: 7

Constraints

The following are some constraints regarding SSD caching.

- Only SSDs can be used for the SSD caching space of a virtual disk.
- A SSD can be assigned to one and only one virtual disk as its caching space.
- Up to 8 SSDs can be used as a SSD cache of a virtual disk.
- There is support for up to 2.4TB of SSD caching space in one system.

Test Results

Test Case 1

This test verifies the dramatic performance gains offered by SSD caching. We set the populate-on-read threshold to 1 which means the data hits once and is populated to SSD. This test used the following equipment and settings:

- Server: ASUS RS700-X7/PS4 (x2)
- Storage: AegisSAN LX F630Q-D316 FW 3.4.0
- HDD: Seagate Constellation ES, ST500NM0001, 500GB, SAS 6Gb/s (x12)
- SSD: Seagate 1200 SSD, ST400FM0053, 400GB, SAS 12Gb/s (x1)
- RAID Group: RAID 0
- I/O Type: Customization
  - Cache Block Size: 1MB
  - Sub-block Size: 8KB
  - Populate-on-read Threshold: 1
- I/O Pattern:
  - Tool: IOmeter V1.1.0
  - Workers: 2
  - Outstanding: 64
  - Access Specifications: 4KB, 100% Read, 100% Random

According to the formula, we assume the warm-up time is

T = (C * P) / (I * S * D) = (10GB * 1) / (430 * 4KB * 12) = 508.03 seconds = 8.47 minutes

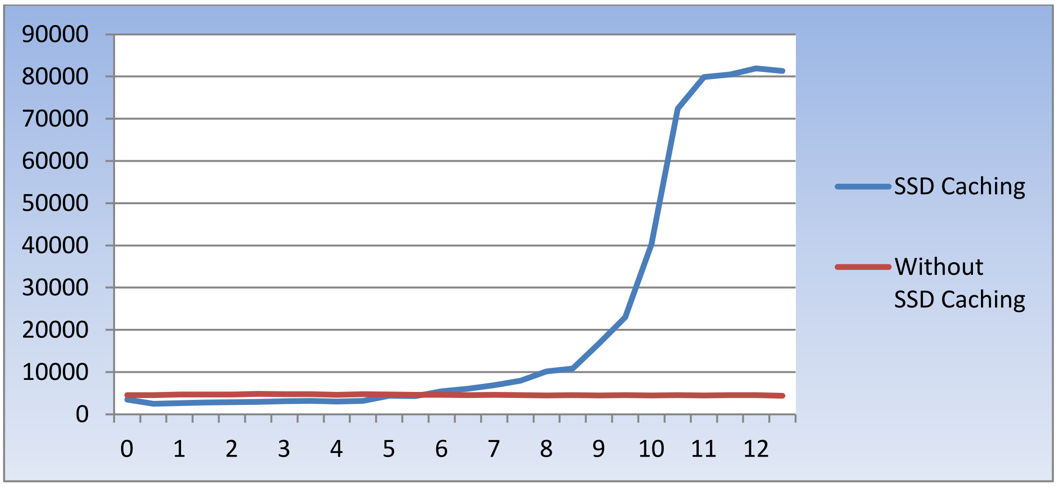
Here is the test result:

Virtual Disk Size	Disable SSD Caching (IOPS)	Enable SSD Caching			Improved
		SSD Capacity	Warm up Time	After Populating (IOPS)	
10GB	4,516	400GB	11.3 min.	81,892	1,713%

Without SSD caching, the average of IOPS is 4,516. Enable SSD caching, IOPS increases to 81,892. It improves (81,892 – 4,516) / 4,516 = 17.134 == 1,713%. Before the test, we assume the warm-up time is 8.47 minutes by formula. In this case, we run 11.3 minutes and watch the IOPS is almost at the top and stable.

In addition, we provide a chart for the relationship between IOPS and warm-up time. The red line is the IOPS without SSD caching. And the blue one is enabled SSD caching. At the beginning 6 minutes, the blue line is lower than the red one because the SSD is on populating. At the period of 6 to 11 minutes, the data is hit and read from the SSD. If it is not hit, the data is read from HDD and populated to the SSD. After 11.3 minutes, the IOPS climbs to the top and stable. It means that the data are all in the SSD.

If you are interesting the changes, we provide a tip to watch the performance monitor on Web UI by enabling the HDD and SSD slots. You can see the I/O changes via the diagrams. According to the test, we are so excited that the result is very good.



Test Case 2

Assume that the real world I/O pattern is random read 90% plus random write 10%. Testing verifies the performance gains offered by SSD caching. One test used the following equipment and settings.

- Storage: AegisSAN Q500-D316 FW 1.2.0
- HDD: Seagate Constellation ES, ST1000NM0011, 1TB, SATA 6Gb/s (x8)
- SSD: Intel SSD DC 3500, SSDSC2BB480G4, 480GB, SATA 6Gb/s (x5)
- RAID Group: RAID 5
- I/O Type: Database Service
  - Cache Block Size: 1MB
  - Sub-block Size: 8KB
  - Populate-on-read Threshold: 2
- I/O Pattern:
  - Tool: IOmeter V1.1.0
  - Workers: 2

- Outstanding: 64
- Access Specifications: 8KB, Random Read 90% + Write 10%

According to the formula, we assume the warm-up time is:

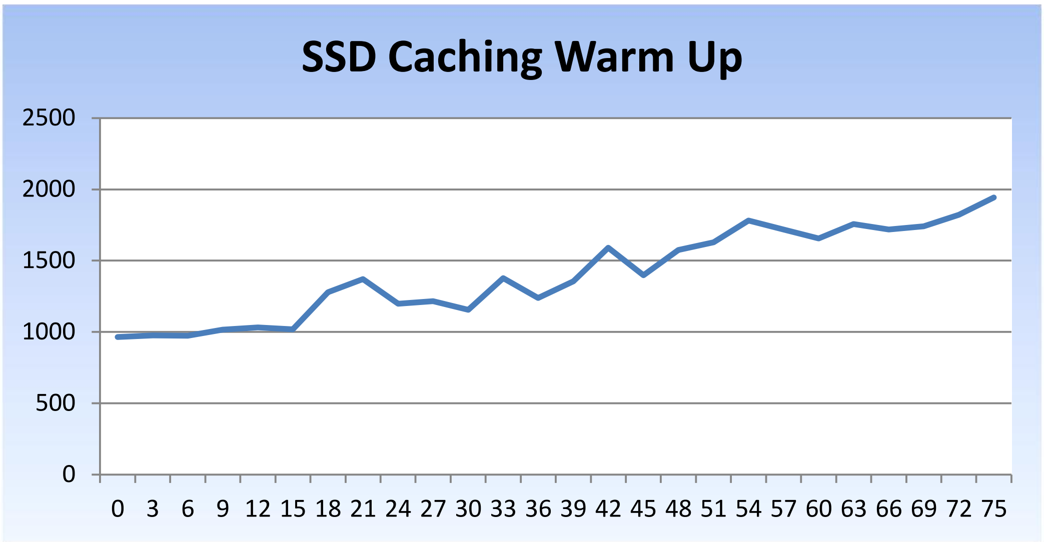
T = (C * P) / (I * S * D) = (2TB * 2) / (244 * 8KB * 8) = 275036.33 seconds = 76.40 hours

Here is the test result:

Virtual Disk Size	Disable SSD Caching (IOPS)	Enable SSD Caching			Improved
		SSD Capacity	Warm up Time	After Populating (IOPS)	
2TB	962	2.4TB	75hr.	1,942	102%

Without SSD caching, the average of IOPS is 962. Enable SSD caching, IOPS increases to 1,942. It improves (1,942 – 962) / 962 = 1.019 == 102%. Before the test, we assume the warm-up time is 76.40 hours by formula. In this case, we run 75 hours and watch the IOPS is almost at the top and stable.

In addition, we provide a chart for the relationship between IOPS and warm-up time. It looks like the IOPS climbs steadily.



Conclusion

The hybrid storage concept of storage acceleration uses the idea of hot data to accelerate I/O performance of an entire storage system. When hardware and IT administration costs are taken into consideration, it turns out that SSD caching as available in modern SAN systems such the AegisSAN Q500 or AegisSAN LX series, are generally the best way for most businesses to gain the benefits of the faster performance from flash based storage without sacrificing the reliability of their data.

Applies To

- AegisSAN Q500: FW 1.3.0
- AegisSAN LX FW 3.4.0