RAIDIX

White paper

SSD Cache in Hybrid Data Storage: How It Works

Contents

About SSD Caching

Data that we store can be divided into two main types, depending on how often we access it. The first type is “cold” data: we request it rather rarely, though it occupies the most of the space on the hard drives of personal computers or company file shares. The other one is “hot” or “warm” data: it is mainly service information used during application loading or running standard procedures and it has a relatively high rate of repeated requests.

SSD-caching is a data processing technology which uses solid state drives as a buffer space for frequently used data. Read and write operations are faster, since the system detects “warmth” of the data and, if it is “hot” enough, puts it to high speed drives.

SSD caching is quite popular where it comes to traditional data storage systems with HDD arrays. HDDs are a good choice for sequential workloads, but they have a physical limit for working with random patterns. Paired with HDD arrays, SSD cache optimizes processing of random requests and improves total storage performance. For this purpose, caching space usually takes up 5-10% of the main storage space.

Storage systems with SSD cache and HDD arrays are traditionally called hybrid. They are very popular, because they are much more affordable than all-flash solutions, but still effective with a wide range of tasks and services.

When SSD Cache is Useful

SSD cache is the most useful in the storage systems that receive not only sequential workload, but also random requests. Moreover, efficiency of SSD cache is higher when “hot” data is kept in a particular address space.

For instance, SSD cache significantly increases productivity when the storage is processing streams in video surveillance environment. This workload pattern receives mostly sequential requests with seldom random write and read operations. Without SSD cache, the storage makes attempts to moderate these peaks by HDDs, which consequently cause total performance degradation.

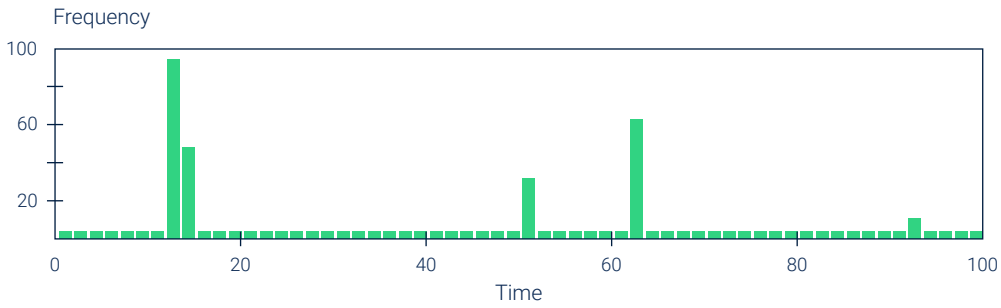


Figure 1. Irregular time gaps and unpredictable frequency of the requests

In reality, random requests often emerge within steady sequential workloads. For example, during simultaneous work of different applications, the one with the higher priority generates a sequential pattern, while the others send random requests to the storage from time to time. Another source of random requests is I/O blender effect, when relatively sequential requests are blended.

At the same time, SSD caching is ineffective in processing highly frequent random requests to non-repeatable data.

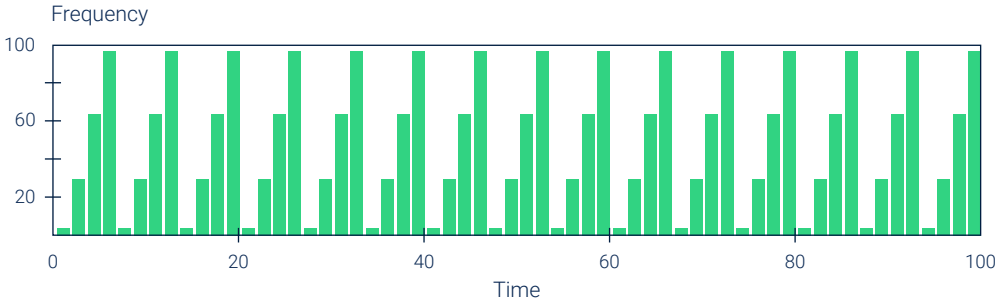


Figure 2. Regular time gaps and predictable frequency of the requests

With a high number of non-repeatable requests, flash drives will run out of space very fast, and performance of the storage system will lower to the speed of HDD array.

Remember, that SSD cache is a very situational tool and its productivity strongly depends on the context. Generally, it is recommended to be used, if the workload has:

- low intensity and irregular time gaps of random read or write requests;
- much more read IO than write IO;
- less "hot" data than usable SSD cache capacity.

SSD Cache in Hybrid Storage

Main purpose of cache is to accelerate operations by placing frequently used data blocks on the fast drive space. RAM memory is used for the "hottest" data — it is called first level cache (L1 cache).

L1 cache can be extended by slower flash drives — in this case we have a second level cache (L2 cache). This approach is the most common way to implement SSD caching in traditional hybrid storage.

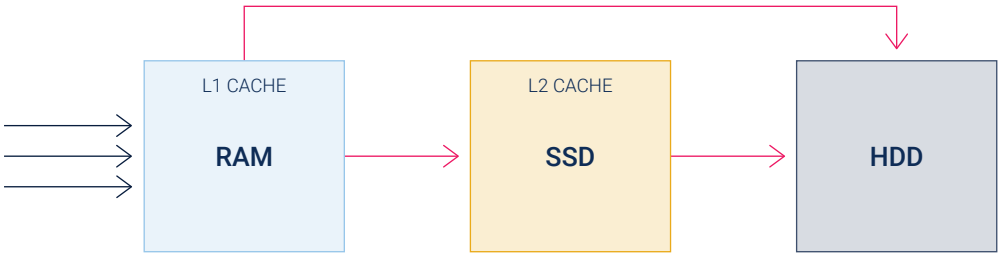


Figure 3. Traditional second-level SSD cache

Traditional scheme of second level cache is the following: all requests come to RAM first and move to the SSD buffer afterwards (Figure 3).

How SSD Cache Works for Read Operations

When read request comes to the storage, the system finds the required data block on the HDD and performs the read operation. If requests to these data blocks repeat, the system makes temporary copies on the SSD drives. The following read operations will be faster, because they are performed from the flash drives.

How SSD Cache Works for Write Operations

When a write request comes to the storage, the system writes data blocks to the SSD drives. As the flash drives are fast, write operation and status response are quick and have low latency. With cache filling, the system gradually moves less "warm" data to the primary HDD storage.

Cache Replacement Algorithms

The main issue in SSD caching is selecting data blocks for replacement in the cache buffer. The space on the flash drives is strictly limited, and one has to decide which block and why will be replaced during the next iteration.

There are several cache replacement algorithms that are used in storage systems to solve this issue.

FIFO (First In, First Out) — the blocks that enter the cache first are gone out first without any regard of request frequency.

LRU (Least Recently Used) — the blocks with the oldest request date leave the cache first.

LARC (Lazy Adaptive Replacement Cache) — the data block comes to the cache if it has at least two requests for a particular time frame. Replacement candidates are noted in the LRU-queue, which is stored in RAM.

SNLRU (Segmented LRU) — data blocks leave the cache according to the LRU, but at the same time they are also segmented by attributes like “cold”, “warm”, “hot”, etc. The grade is defined by the frequency of requests.

LFU (Least Frequently Used) — the blocks leave the cache first if they have less requests than others.

LRFU (Least Recently/Frequently Used) — this algorithm combines LRU and LFU, primarily replacing the blocks that have the calculated parameter of the date and request frequency.

Total productivity of SSD caching depends on the type of replacement algorithm and quality of its implementation.

Parallel SSD Cache in RAIDIX Storage

SSD cache in RAIDIX has parallel architecture with two unique features:

1. All incoming requests are sorted into RRC (Random Read Cache) and RWC (Random Write Cache) categories.

2. Log-structured writing improves proprietary replacement algorithms.

1. RRC and RWC

RAIDIX cache space is separated according to the functional categories: RRC for random read requests and RWC for random write requests. Each group has its own rule to get and replace data blocks.

Special Detector qualifies all incoming requests and sorts them into the right category.

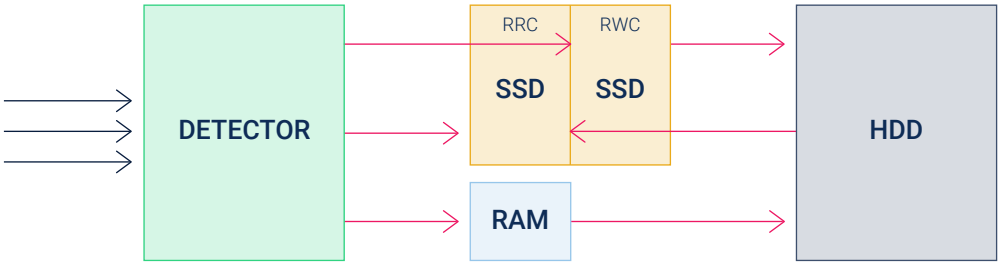


Figure 4. Parallel SSD caching in RAIDIX storage

RRC Caching Policy

RRC category gets only random data requests with frequency more than two (“ghost” queue).

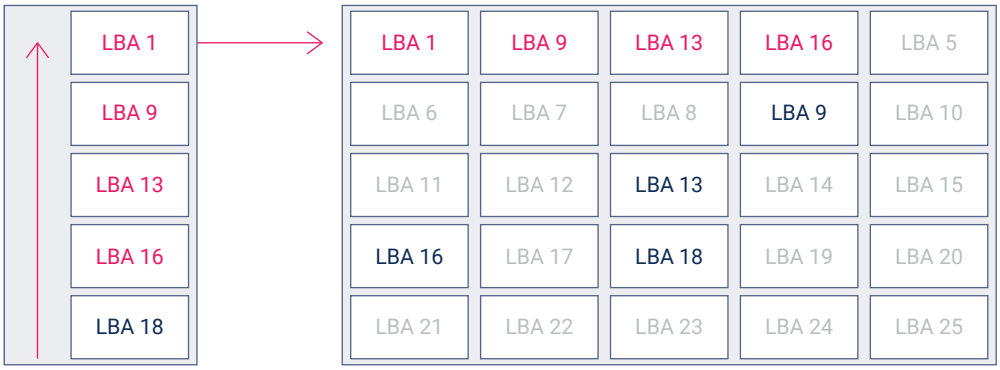
RWC Caching Policy

RWC accepts all random write requests if their block size is less than setting parameter (default value is 32KB).

2. Log-Structured Writing and Cache Replacement

Log-structured writing is a logic of sequential data writing disregard logical block addressing (LBA).

Figure 5. Visualization of Log-structured writing



RAIDIX storage employs log-structured writing for filling dedicated data areas (with 1GB default size) within RRC and RWC. These log-structured areas give additional ranging options to the cache replacement process.

RRC Data Replacement

System identifies the “coldest” log-structured areas in RRC category, and then puts there new blocks from the “ghost” queue (with frequency more than two).

RWC Data Replacement

System selects data log-structured areas using FIFO algorithm, and then sequentially removes data blocks according to their logical block address (LBA).

How to Benefit with SSD Caching

SSD cache in RAIDIX is not a simple buffer for the “hot” data. Implemented parallel architecture moves it beyond the traditional approach: RAIDIX caching works as a smart distributor of the workload to a primary device storage. With request allocation and unique replacement algorithms, SSD cache effectively mitigates random I/O peak impact, reducing negative effect to the total storage performance.

Replacement algorithms use log-structured writing for more effective data allocation in the cache space. It reduces the total amount of requests to the flash drives and significantly decreases their wearing-out.

Reducing Wearing-Out of Flash Drives

Implemented detector and proprietary cache algorithms successfully reduce total write hits to the dedicated SSD array. Conventional 2L cache with LRU algorithms shows 10.8 write hits, while RAIDIX shows only 1.8.

It means that RAIDIX caching demands 6 times less flash memory write cycles than caching in traditional hybrid storage. Therefore, this approach can make your flash drives stay in the caching services up to 6 times longer.

Improving Performance for Different Workloads

Looking at the mixed workload patterns, we can see a chronological range of different states where each one has either a sequential or random type of request. Storage should be ready to deal with each state despite how convenient they are.

We've made a lab performance test to reveal RAIDIX SSD cache productivity in different workload states. We checked the difference between storage with active SSD caching and storage without SSD caching. Both of them were under the same emulated workloads.

Test system configuration:

SSD cache: RAID 10, 4 SAS SSD, 372GB

Main storage: RAID 6i, 13 HDD, 3072GB

Workload	Request Type	SSD Cache is On	SSD Cache is Off	Performance Growth
Random read (100% cache hit)	random read 100%	85.5K IOps	2.5K IOps	34 times
Random write (100% cache hit)	random write 100%	23K IOps	500 IOps	46 times
Random read (80% cache hit, 20% HDD hit)	random read 100%	16.5K IOps	2.5K IOps	6.5 times
Random read from cache, seq write on HDD	random read 50% sequential write 50%	40K IOps 870 Mbps	180 IOps 411 Mbps	222 times 2 times
Random read and write (100% cache hit)	random read 50% random write 50%	30K IOps 19K IOps	224 IOps 800 IOps	133 times 23 times
Sequential requests with large block size, all random requests has 100% cache hit	random read 25% random write 25% sequential read 25% sequential write 25%	2438 IOps 1918 IOps 668 Mbps 382 Mbps	56 IOps 82 IOps 120 Mbps 76.7 Mbps	43 times 24 times 5.5 times 5 times

Certainly, every real case has its own unique workload specifics. But this lab test intends to highlight how SSD caching tool can support your hybrid storage struggling with emergent random requests.

Conclusion

SSD caching technology allows you to accelerate storage performance when working with mixed workload. It is a simple and affordable way to prepare storage system for the cases where HDD arrays are not good enough operating on their own.

In the era of diversity and complexity of server tasks and applications, SSD cache in hybrid storage is more and more appealing. But still important to remember, that this technology is very sensitive to the context, and it cannot be the universal solution for all storage performance issues.

Parallel SSD cache in RAIDIX storage has a bundle of unique features that not only boost performance, but also saves you flash drive durability.