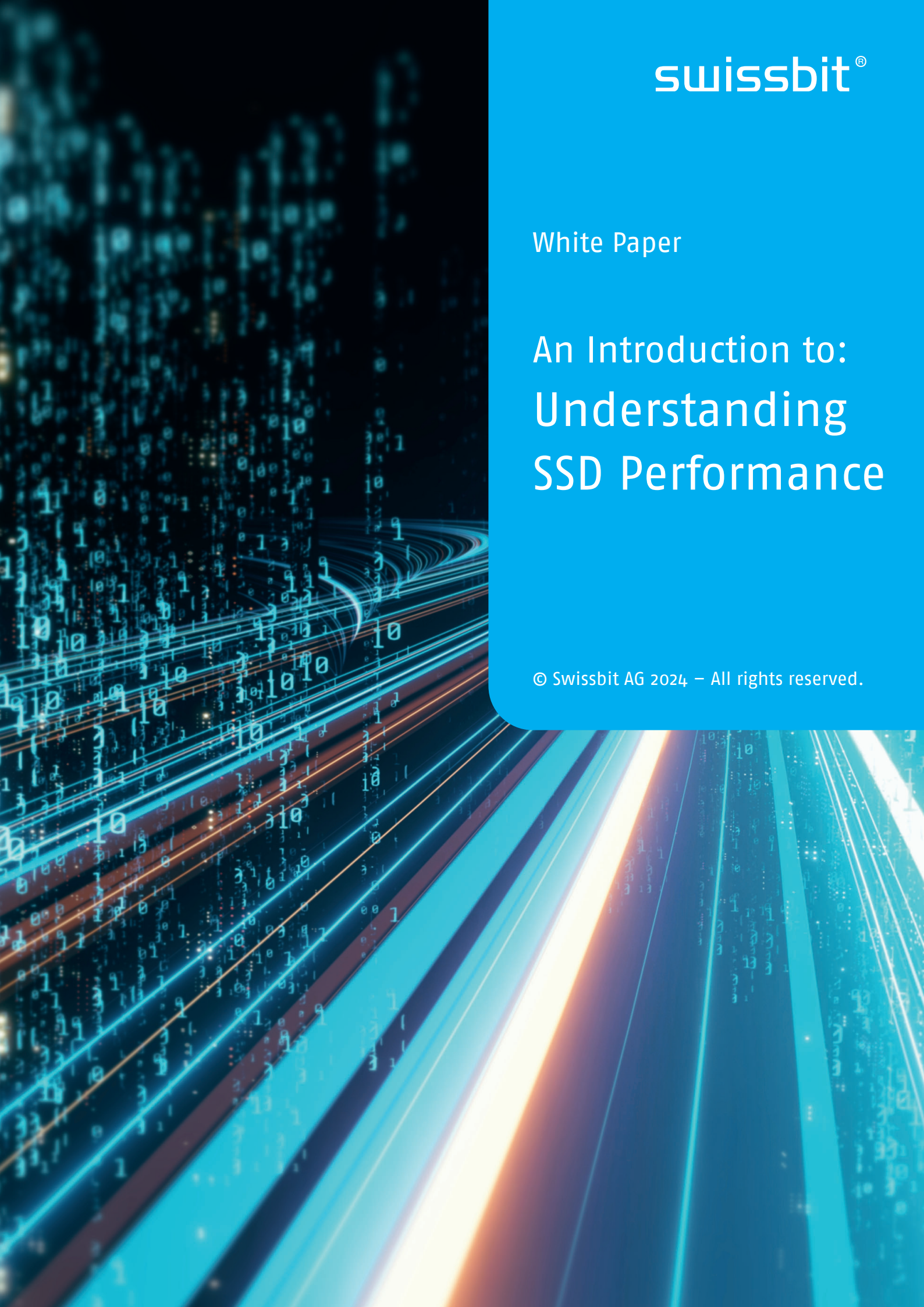


swissbit®

White Paper

An Introduction to: Understanding SSD Performance

swissbit®

Table of Contents

1. Introduction

Performance of storage systems such as flash memory cards or SSDs refers to the amount of data that can be transferred in a given amount of time.

It is often assumed that performance can be expressed in a single number. Product flyers frequently indicate the performance for a storage product valid under optimal conditions.

Understandably, when comparing two products, this one number is generally used as a reference. However, this is far from sufficient. When examining data transfers more closely, the difference between write and read transfers becomes obvious. to measure and compare the performance of one drive to another.

Furthermore, the size of the chunks and the randomness of the allocation units (LBAs) in which data is transferred is a relevant factor.

In addition to that and unlike Hard-Disk Drives, SSDs vary in their performance over their lifetime.

Consequently, they require unique performance measurement techniques along with an analysis of the specific use-cases and environmental conditions, such as temperature, to measure and compare the performance of one drive to another.

2. Write Speed

Sequential and Random Write Performance

The most common performance characteristics measured are sequential and random operations. If the performance is indicated with a single number, e.g. 550 MB/s, it usually refers to the maximum interface transfer rate or a sequential read performance, which is often the highest of all performance values. If a data sheet indicates two performance numbers, a read- and a write-performance, these usually refer to the sequential-read- and sequential-write-performance. Sequential operations access locations on the storage device in a contiguous manner and are generally associated with large data transfer sizes (e.g. 128 kB or more) while random operations access locations on the storage device in a non-contiguous manner and are generally associated with small data transfer sizes (e.g. 4kB). Fig. 1 visualizes the access patterns.

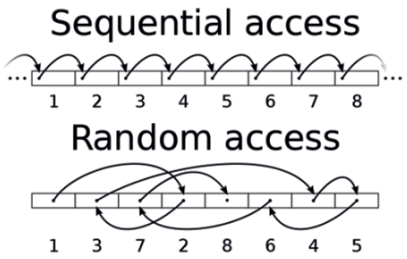


Figure 1: Sequential vs random write access

In the case of sequential access, the data throughput is much higher. The reason for this is that a lot of data is moved with very few commands. Therefore, the overhead in the flash memory controller is rather small. On the other hand, in case of random access, a high number of commands transfer a small amount of data per command. This results in a lot of overhead, i.e. management of the incoming commands, queuing and other tasks, which in return leads to a lower data throughput.

Use cases with purely sequential write performance usually experience higher performance. Random write performance is a more challenging for the flash translation layer to manage.

Fresh out of the Box

A first misconception about performance is that it is invariant over time. Repeated measurements, even over a longer period of time, should lead to the same results – given the fact that the measurement is executed in the same manner. For flash memory, this is not the case. In fact, the performance of SSDs can vary over time due to fragmentation and the need for maintenance operations such as garbage collection or wear-leveling. The drives are sensitive to how much of their capacity is used and to their write-history. The first time in use, the so-called "fresh out of the box" scenario describes a state, where all memory cells are fresh or unused. All cells are in an erased state. Since cells have to be erased before they can be programmed, the "fresh out of the box" case is a scenario, where the incoming data can be written to the flash immediately. Consequently, the performance – to be precise, the write performance – is the highest for sequential and random write scenarios. Yet this level of write performance will only be achieved for a small fraction of the lifetime of a flash memory product – usually the first few hours or the first day of use. Therefore, this is a poor indication of the real performance of a flash memory drive. Once a certain amount of data has been written onto the drive, the performance drops – and the drive never in its lifetime regains the initial write performance.

The reason for this behavior lies within the architecture of the flash memory. Memory cells are arranged in strings, which are then grouped into pages, which make up a block. While data is written by a multiple of pages, it is only possible to delete a whole block. Furthermore, flash memory must be erased before it can be re-programmed. During the vast majority of the lifetime of a drive, when data is re-written (meaning: at any point in time, data has been written to this location before), the original data is marked as "invalid" and the new data is written to a different location. After all blocks have been written to once, the drive must read all the good data around the invalid data and move it to another location, where it is re-ordered and consolidated. The old or invalid blocks are then erased and new data is written to the newly freed-up blocks. This process is called "garbage collection" – an ongoing activity of re-organizing the stored data that is causing large volumes of traffic on the flash bus and limiting the performance of writing new data from the host. This also explains why, unlike for hard-disks, when writing to a SSD

the sequential or random nature of the writes will affect future performance. Sequential writes will generally leave few large blocks of free space which results in less effort for the garbage collection. Random writes will generally leave many small blocks of free space which results in a higher effort for the garbage collection.

The variation between "fresh out of the box" and steady-state performance reveals that performance-testing for short periods of time will almost certainly not disclose the real performance over the lifetime. The meaningful measurement of the steady state performance requires a certain preconditioning that can be achieved by making sure that each page of the flash has been written at least once before performing the test.

The speed of your SSD or storage media is best when it is fresh. As soon as physical flash blocks contain a mix of valid and invalid data, garbage collection kicks-in reducing the user-experienced write performance.

SLC Cache

NAND flash memory stores data in memory cells, which are made of floating-gate transistors. At first, each cell has two possible states, so one bit of data is stored in each cell, a so-called single-level cell (SLC) flash memory. A multi-level cell (MLC) is a memory element capable of storing more than a single bit of information, usually two. Triple-level cells (TLC) and quad-level cells (QLC) are memory cells, which can store 3 and 4 bits per cell, respectively.

With increasing bits per memory cell, write speeds to the cells significantly decrease because the additional bits require more signal processing and error correction during writing (programming). The decrease in write speed to the cells results in a decrease in overall write performance of storage systems using flash memory cells with higher per-cell storage capacity.

To overcome this, a high-performance write buffer (cache) can be created within the flash memory that simulates high-performance SLC. This is done by specifically configuring an area of cells in MLC, TLC or QLC flash to only store one bit of data. This is called pseudo-SLC (pSLC) mode and immensely increases write speed to these cells. During write operations, data from the host system is first written to the high-performance buffer at accelerated speeds and then during the idle periods, the data is relocated from the buffer to the other (slower) memory cells. This mechanism notably improves the perceived speed of a flash memory storage device, since an amount of data up to the size of the buffer can be transferred at a very high speed. A typical buffer size is shown in Table 1 – it varies based on the capacity of the SSD, increasing with larger capacities.

Drive Capacity	Buffer Size
120 GB	5 GB
250 GB	5 GB
500 GB	9 GB
750 GB	14 GB
1000 GB	20 GB

Table 1: Typical buffer sizes

Under consecutive write operations with no idle time, the buffer will eventually fill up. The buffer size therefore determines the maximum duration of continuous write operations at accelerated speeds.

At this point, the transfer will continue and the data will then be written directly to the slower memory cells not operating in pSLC mode. Depending on the technology of the cells and many other factors, the write speed of a whole SSD can plunge from 400 or 500 MB/s down to below 100 MB/s. It will continue to write at this speed until the sequential transfer is finished and until it had time to clear the buffer by relocating all data to the other storage cells. This usually takes around 30 to 60 seconds depending on the buffer size. The numbers in Table 2 shows that a measurement of 100 MB/s is still not the lowest. Certain consumer SSDs with QLC technology show sustained sequential write performance of around 80 MB/s only.

Capacity		1TB	2TB
Sequential Write	SLC Cache	520MB/s	520MB/s
	QLC	80MB/s	160MB/s

Table 2: Specification of one of the first consumer SSD with QLC NAND flash memory

Thermal Throttling

Performance can also depend on temperature. The temperature of the chip inside the package is determined by two factors – the ambient temperature and the heat produced by the chip, which is mainly due to switching activity. This generally means the chip is active. Since the chip might be damaged by high temperatures, almost all the latest flash memory controllers have an internal sensor that measures the temperature. If it rises over a defined threshold, the chip reduces its performance to not heat up further and risk damage. This is called thermal throttling. While most flash controllers have such functionality, the threshold is often set at a different level. Depending on the power consumption and therefore the heat dissipation of the chip itself, the reduced activity after reaching the temperature threshold leads to different performances.

In Fig. 4 the behavior of a competitor SSD can be seen. The drive is continuously filled with data in a sustained sequential write transfer. The internal temperature measurement reads 28 °C in the beginning. After a transfer time of a little more than 8 minutes, a temperature of 72 °C is reached, which triggers the thermal throttling to become active. As a result, the write speed frequently drops down to as low as 50 MB/s. In average, the write speed is close to 250 MB/s over the whole period in which the drive´s internal temperature is 72 °C. While this already indicates a significant drop in performance at this temperature, there are other drives on which the impact of thermal throttling is even worse. In Fig. 5 the measurement of another drive is depicted. The initial write speed is around 420 MB/s. It can be sustained for about 13 minutes, after which the internal temperature reaches 67 °C and the write speed decreases to roughly 105 MB/s.

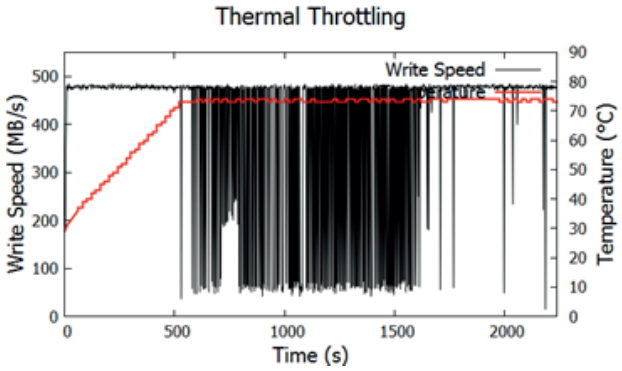


Figure 2: Write speed over time over time of a competitor SSD (SATA) in a scenario of sustained sequential write

When looking at the graphs, it is a valid argument to say that a sequential write transfer of dozens of gigabytes is not a typical workload. Nevertheless, the issue with temperature is still the same. In this test, the drive was connected and then tested immediately.

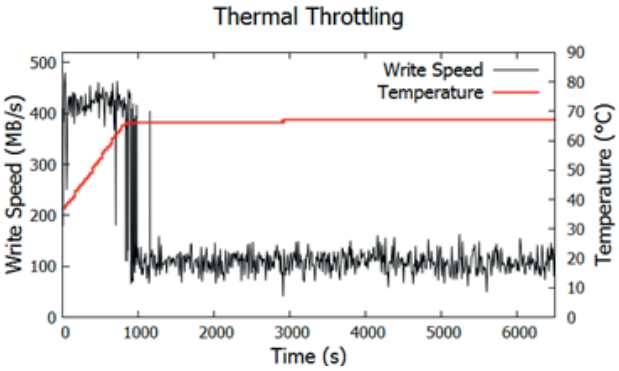


Figure 3: Write speed over time of a competitor SSD (SATA) in a scenario of sustained sequential write

This means that the drive was at room temperature before beginning this test. If it had been running in an industrial system for hours, days or months, the internal temperature would have risen to 40 or 50 °C before starting the test.

Considering the industrial temperature range from -40 °C to +85 °C it becomes clear that both drives would only deliver a fraction of their advertised performance in the upper operational temperature range from above 72 °C to 85 °C.

Thermal throttling is used especially in "System-in-package"products to manage the controller temperature and its effect on the flash reliability. Depending on the controller´s design, this can have a significant impact on performance when operating an SSD in high temperature environments.

Detailed Example of Write Speed

For a better understanding of the effects of repeated filling of a storage medium and the associated occupancy of flash blocks, fragmentation and garbage collection, the change in the write speed of an SSD without a buffer during the execution of various workloads is considered below. It becomes clear that the write speed depends not only on the current workload but also on the previous workload and therefore on the state of the SSD.

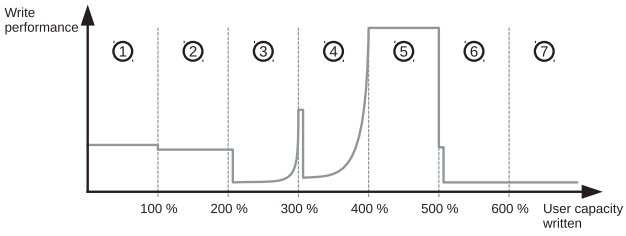


Figure 4: Broken down sections of write operations

Section 1: 4KiB Random Write

At the beginning of section 1 the storage medium is empty. It is fresh-out-of-the-box, is completely trimmed or has been com- pletely deleted ("secure-erased"). Accordingly, the physical addresses are not assigned to any logical addresses the mapping tables contain no entries. As the storage medium begins to be filled with random write accesses (4 KiB Random-Write) each logical address is written exactly once. Due to the small amount of data per write access and the high administrative overhead when tracking the allocation tables, the speed is severely limited. The speed stays constant over the entire time.

Section 2: 4KiB Random Write (Same Address Sequence)

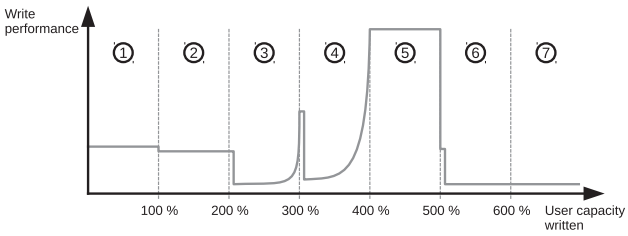
In section 2, the same write accesses from section 1 are repeated: the sequence of addresses written is identical. A slight reduction in write speed can be observed. This is due to the mapping tables being completely filled and the old entry having to be invalidated for each newly written address. Because of the identical address sequence, there is no pressure on the garbage collector, because with each newly filled flash block another block is automatically released again. When a block is completely filled, a second block exists with the same logical addresses, which are now obsolete, whereby the second block can be deleted.

Within a datasheet the speed of this section is often shown as "sustained random write". However, this can be misleading by suggesting this is the minimum achievable write speed. The minimum write speed is only reached at maximum load on the garbage collector. This is covered in the following sections. Careful consideration must be given to the minimal speed requirements of the respective application.

Section 3: 4KiB Random Write

After filling the storage medium twice with the same sequence of random write accesses, the address sequence is changed at the start of section 3. Even after writing a small amount of data, the speed drops significantly: "over-provisioning" has been used up and the previously inactive garbage collector, now works under full load. By changing the address sequence, blocks are no longer automatically freed, and the garbage collector must constantly copy data to gain free blocks for new data.

Since each address is still written exactly once in each section, the write speed increases exponentially shortly before the end of the section, as with the shrinking, as yet missing amount of addresses with each newly filled block, blocks are freed up again automatically. When the last block of this section is written to, all blocks from the over-provisioning are automatically freed-up again.



Section 4: 128 KiB Sequential Write

At the beginning of section 4, a random address sequence is now switched to a sequential address sequence. In addition, the amount of data per access is increased from 4 KiB to 128 KiB. This increase immediately results in higher throughput through greater efficiency of transfer between host and storage medium. After over-provisioning is used up, the write speed drops to almost the same value as in section 3 because all the old data was written with random addresses and now the load on the garbage collector is just as high.

Since the garbage collector sorts the addresses as far as possible when moving data, the speed increase occurs earlier with decreasing address quantity and when blocks automatically become free again. Consequently, write-speed increases at a faster rate than in section 3 because of the greater access size of 128 KiB.

Section 5: 128 KiB Sequential Write (Second Run)

In the previous section, the storage medium was filled sequentially. The addresses of the data in the flash blocks show a strictly monotonous increase in each block. All blocks from the over-provisioning are freed up again, since again each address has been written exactly once. In section 5, the storage medium is filled sequentially just as in section 4.

Since the address sequence is the same as in the previous section, with each flash block written, another block is freed up automatically. The large amount of data per host write access makes the transfer between host and storage medium very efficient. The garbage collector is load free, and owing to the sequential write accesses, the tracking of the allocation tables also generates little additional load. Here, the maximum write speed of the storage medium is reached. Only an empty storage medium (after trim or secure erase) could still achieve a slightly higher speed.

Section 6 and 7: 4 KiB Random-Write (Complete Random Address)

At the beginning of section 6, random address sequences are chosen again. In contrast to sections 1–3, where each address has been written exactly once, the entire address space is now open for each new address. Accordingly, in both sections, on the one hand, not all logical addresses are necessarily rewritten, and on the other hand, several addresses are written several times. After filling the initial over-provisioning, the garbage collector is under full load. Since a flash block is never automatically freed, the garbage collector remains permanently under full load. This operating state is now the true sustained random write, which achieves the minimum write speed. However, this is unlikely to reflect a practical application, especially as the write amplification factor is so high that the storage medium would very quickly reach its specified number of write and erase cycles.

3. Read Speed

NAND flash can be read much faster than it can be written to, almost independent of technology. The write performance depends on program times of different flash technologies, caches, garbage collection and basic readiness of the flash to be written to. Read performance on the other hand mainly depends on the ability of the controller or flash translation layer to identify the data location and on the quality level of the data representation in the flash arrays.

Sequential and random read performance

Somewhat analog to the sequential and random write transfers, in the case of random read there is additional effort for the flash translation layer to locate the data. When the host system requests the data of a certain address, the memory controller has to look up in its mapping table where this address is physically stored in the flash memory. In case of sequential reading, the next address that will be requested is already known and its physical location can already be determined. Another factor is the lookup of the physical location in the mapping table itself. The table is built up in a tree-shape, starting at the trunk. The location of data with similar addresses would be located on one branch. Searching for further locations from there on the same branch is a lot faster than searching for completely different addresses for which it is necessary to start at the trunk again. In addition to that, for sequential operations, concepts like prefetching to get the data of the next address can be utilized.

Sequential and random preconditioning

If data has been written sequentially and is read in the same sequence, the locating effort is minimal. Hence this is the best case for read performance. When data that has been written randomly is read sequentially and when data that has been written sequentially but is read randomly, both cases alike, the system shows much lower read performance compared to the pure sequential case.

Cross Temperature Read Performance

When data is read at a significantly different temperature than it has been written at, the likelihood of bit errors and the need to calibrate read voltage levels increases. As an effect, the read performance deteriorates dramatically.

Performance over lifetime

Very much like the write performance, read performance changes over the lifetime of the flash memory as high temperatures and data retention time comes into play. The major reason is the errors during the read-out of the flash that have to be corrected by the flash memory controller. Naturally, the number of errors increases as the memory reaches its end of life. In the beginning, when there are few errors, the data can be transferred without any further processing. As time progresses, the data from the memory contains more errors which are corrected by the error correction module. The necessary calculations for the correction take time and introduce latency. With increasing amount of errors, the complexity for the correction increases, as does the time needed.

At some point, the error correction is not capable of correcting all errors in the data anymore. In this case, most controllers perform a so-called "read-retry". As the name implies, the data is read again from the flash with adjusted threshold voltage levels to compensate for the aging of the storage cells. This has a significant impact on performance. Compared with the optimal case, where data is read and can be transferred directly without any further processing, in this case the data has to be read, processed in the error correction with maximum processing length, then read again and probably also processed rather long, since after a read-retry it is likely that the amount of errors is close to what can be corrected. In total, the read process for a single block of data in this case takes up about four times as long, which in turn leads to a drop of the read performance to 25% for the block of data.

In the latest generations of 3D flash, where errors are more frequent than in older generations, the error correction uses a mechanism called soft-decoding for data that cannot be corrected instantly by the hardware engine. Instead of reading data just as zeros and ones, it also reads out the likelihood of it being a zero or one to aid in the correction process. The soft-decoding achieves correction capabilities far superior to pure hardware correction but takes

up to ten times of a normal read process. Consequently, one soft-decoding process heavily impacts the overall read performance.

Read Performance and Reliability

Another factor that influences read performance is read disturb management. During the read process of flash memory cells, neighboring cells are "disturbed". Reading one cell frequently will at some point compromise the data in surrounding cells. To prevent this, read-counters keep track of the number of times each cell is read. When one cell is read continuously over a long period of time, it is needed to write the read-counters of cells onto the flash. Obviously, as the flash memory can´t be read and written to at the same time, this will briefly interrupt the prolonged read and therefore slightly reduce performance. This is not a mandatory process. To reach a maximum performance, it is possible to not perform this at all. This however, highlights that a maximum performance can sometimes be associated with a reduced reliability and safety of the data.

Similar to write performance, the read performance is dependent on the use case. The sequential read performance is usually much higher than the random performance. The variation of the read performance over the lifetime is mostly due to the increasing amount of errors in the data, which need to be corrected. The highest performance numbers are therefore also expected when the SSD is fresh.

Detailed example of Read Speed

Read performance is not dependent on as many factors as write speed. It depends only on the type of access used to write and how it is read.

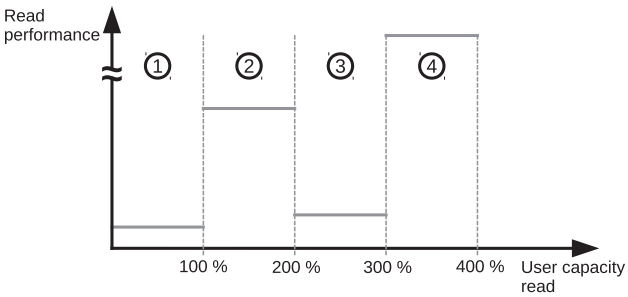


Figure 5: Broken down sections of Read Speed

The data in section 1 was randomly written and read again at random, but in a different order. Here, the lowest read performance is achieved because the search effort in the allocation tables is very high and the efficiency of the data transfer between storage medium and host is low because of the packet size of 4 KiB.

In section 2, the randomly written data is sequentially read. The read performance is much higher, since 128 KiB are now transferred to the host per access, which increases efficiency. The search effort in the allocation tables is still very high.

Before section 3 the storage medium was written sequentially. The read accesses in this section were random. The read performance is correspondingly low due to the package size of 4 KiB. However, sequential write makes searching in mapping tables easier and faster than in section 1.

In section 4, the storage medium reaches the maximum read performance. The sequentially written data is now also read sequentially in 128 KiB packets. The search in the allocation tables is very simple, the transfer to the host very efficient and internal access to the NAND flash can be done using "read ahead". For a storage medium with pSLC cache, a higher read performance would occur for some of the memory addresses in the four sections

4. Conclusion

There is a level of complexity when measuring and assessing performance of SSDs. This paper explains the difference between the "fresh out of the box" performance and the steady-state performance and that only the latter should be considered as indicator of what to expect from a SSD. In addition to that, the issue of MLC, TLC and QLC NAND flash delivering lower write performance was discussed and how this can somewhat be ameliorated by using a pSLC cache. The limitations of such an approach in case of sustained write transfers were outlined. At last, the possibly strong effects of thermal throttling on performance were examined.

In conclusion, all these aspects contribute to the fact that performance and its measurement is a complex subject and that it is essential to understand exactly what has been measured under which conditions to be able to compare different drives. Numerous indications in advertisements and data sheets are under conditions far from reality and for use-cases that are very different from the target.

swissbit®

Do you have any questions? Get in touch!

Swissbit Europe (HQ)
Tel. +41 71 913 03 00
sales@swissbit.com

Swissbit North America
Tel. +1 978-490-3252
salesna@swissbit.com

Swissbit Japan
Tel. +81 3 6258 0521
sales-japan@swissbit.com

Swissbit Asia
Tel. +886 912 059 197
salesasia@swissbit.com

About Swissbit

Swissbit AG is the leading European manufacturer of storage, security and embedded IoT solutions for demanding applications. As trusted partner, Swissbit empowers the digital and connected world by reliably storing and protecting data in industrial, security and IoT applications.

www.swissbit.com

Subject to change without prior notice.
Visit www.swissbit.com for latest information.