Datenbank Spektrum (2018) 18:121–127
https://doi.org/10.1007/s13222-018-0287-8

KURZ ERKLÄRT

On the Diversity of Memory and Storage Technologies

Ismail Oukid[1] · **Lucas Lersch**[1,2]

Published online: 4 June 2018

Abstract
The last decade has seen tremendous developments in memory and storage technologies, starting with Flash Memory and continuing with the upcoming Storage-Class Memories. Combined with an explosion of data processing, data analytics, and machine learning, this led to a segmentation of the memory and storage market. Consequently, the traditional storage hierarchy, as we know it today, might be replaced by a multitude of storage hierarchies, with potentially different depths, each tailored for specific workloads. In this context, we explore in this "Kurz Erklärt" the state of memory technologies and reflect on their future use with a focus on data management systems.

Keywords Storage · Main Memory · Flash · SSD · DRAM · Storage-Class Memory · Non-Volatile Memory

1 Introduction

The traditional storage hierarchy comprises several layers of memory technologies, ordered from the fastest and least dense to the slowest and densest: CPU caches (SRAM), main memory (DRAM), secondary memory (HDD), and potentially tertiary memory (Tape Drive). The rise of Flash memory, manufactured in the Solid-State Drive (SSD) form, has pushed HDDs another level down the storage hierarchy: SSDs have successfully superseded HDDs. However, the rise of novel memory technologies, such as Storage-Class Memories (SCM), and substantial hardware and software advances in existing ones, such as Open-Channel SSDs [3], force us to reconsider how we conceive storage hierarchies. Indeed, storage system designers are faced with an unprecedented diversity of memory and storage technologies, as illustrated in Fig. 1.

The performance of data-intensive applications is directly dependent on the performance of the underlying storage system. Depending on their host device (servers, smartphones, embedded devices, etc.), and depending on the nature of their data access patterns, these applications may be bound by memory bandwidth, by memory latency, or by energy consumption. In a real world scenario, a data-intensive application will be constrained by a combination of the three. This disparity in bandwidth, latency, and energy consumption constraints led hardware manufacturers to segment the market of memory and storage devices into several products, each of which exhibits a different instantiation of the aforementioned three-way trade-off (see more details in Sects. 2 and 3).

On the one hand, the segmentation and diversification of memory technologies bring the opportunity to build storage systems that are optimized for specific workloads. On the other hand, building such systems is complex and requires exposing traditionally hardware-managed parts of storage to the application layer (e.g., Open-Channel SSDs). In the cloud, this task is even more complex as memory and storage resources may be shared by multiple entities, thereby making quality of service a challenging task for cloud providers.

The first "Kurz Erklärt" of this series explored the diversity of computing units and the opportunities they bring to data management systems [30]. In this second "Kurz Erklärt", we give an overview of the state of advancement – from a systems developer point of view – of memory and storage technologies and their impact on data management systems. To do so, we organize the remainder of this paper as follows: Sects. 2 and 3 discuss recent developments in main memory and storage technologies, respectively. Thereafter, Sect. 4 introduces Storage-Class Memory, highlights its opportunities, and underlines its chal-

✉ Ismail Oukid
ismail.oukid@sap.com

Lucas Lersch
lucas.lersch@sap.com

[1] SAP SE, Walldorf, Germany

[2] TU Dresden, Dresden, Germany

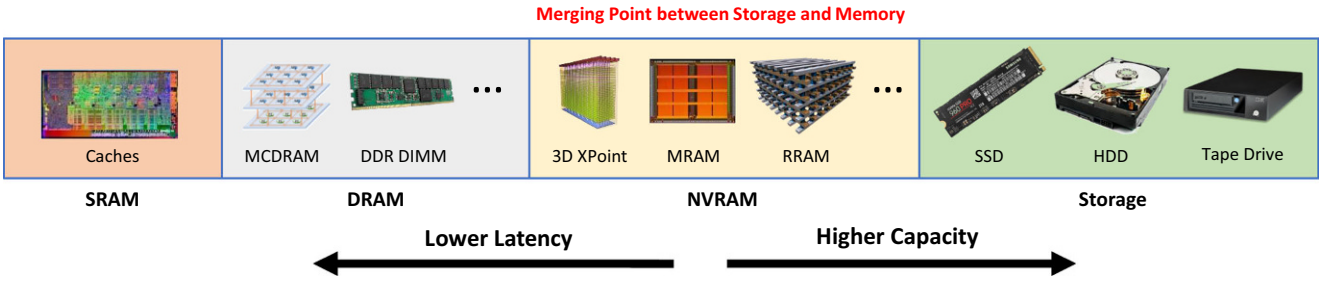

Fig. 1 Illustration of the diversity of memory and storage technologies (adapted from [29])

lenges. Finally, Sect. 5 summarizes the paper and outlines future breakthroughs lying ahead of us.

2 Random Access Memory

There are two main types of Random Access Memory (RAM): Static RAM (SRAM) and Dynamic RAM (DRAM). SRAM requires six transistors per memory cell and relies on changing the direction of the current to read and write memory cells. In contrast, DRAM requires only one transistor and one capacitor which is used to hold the charges. Therefore, DRAM is much simpler, denser, and cheaper (since it requires six times less transistors) than SRAM. However, since DRAM's capacitors produce current leakage, its memory cells must be constantly refreshed – hence the name "Dynamic" RAM. While DRAM is denser and simpler to produce, SRAM offers a much lower access latency and a much higher bandwidth. Therefore, SRAM is usually used for the smaller CPU caches whose performance is critical, while DRAM is used for the larger main memory. Since SRAM is embedded on-chip and inflexible, we focus in the remainder of this section on DRAM.

The DRAM market is currently dominated by Samsung, Micron, and SK Hynix; they own more than 95% of the market share [50]. Furthermore, the market is segmented into many categories, each of which is tailored for specific application domains:

- **Double Data Rate (DDR) DRAM** is targeted at Complex Instruction Set Computers (CISC), which can issue multiple instructions in a single cycle, such as CPUs found in desktops and servers. Therefore, it is optimized to handle parallel, small-sized memory requests using a typically 64-bit memory bus.
- **Low Power DDR (LPDDR)** offers very low power consumption and is targeted at smaller devices such as smartphones, tablets, and laptops.
- **Graphics DDR (GDDR)** is optimized for GPU workloads, or more generally, for Reduced Instruction Set Computers (RISC) that issue a single instruction per cycle. It differs significantly from DDR in that it has a wider memory bus (up to 256-bit wide) which allows it to provide much higher bandwidth. However, it does not handle well parallel non-adjacent memory requests.
- **High Bandwidth Memory (HBM)** is a variant of DRAM that provides much higher bandwidth than GDDR thanks to its 3D design: multiple layers of DRAM are stacked together and accessed through a very wide memory bus (typically 1024-bit wide). It is mainly targeted at high-end GPUs and servers.
- **Multi-Channel DRAM (MCDRAM)** is a type of HBM introduced by Intel in its second generation Xeon Phi processors[1] [36]. It is a high-bandwidth, low-capacity DRAM that can be used as a software-managed fast buffer between CPU caches and main memory to accelerate analytical workloads.
- **Hybrid Memory Cube (HMC)** is another promising high-bandwidth, low-capacity 3D-stacked DRAM and a competitor of MCDRAM. Its application domains include high-end computing and networking.

Each category, with the exception of MCDRAM and HMC, has improved over multiple generations, the latest being DDR5, GDDR6, LPDDR4X, and HBM2.

DDR DRAM is by far the category that offers the highest capacity. It is also the most relevant for database systems. While the cost per bit of DDR DRAM has steadily decreased over the years, the capacity and bandwidth per core have worsened [38]. As a matter of fact, it is intrinsically hard to further increase the density of DRAM [21]: The smaller the DRAM cell, the more it leaks energy which interferes with the state of neighboring cells, thereby exponentially increasing error rates. Another concern is that a significant share of data-centers energy consumption is attributed to DRAM [9], either directly or indirectly (e.g., through the cooling system). Consequently, DRAM no longer satisfies the demand for ever-increasing main-memory capacities.

[1] Intel has discontinued its Xeon Phi series, albeit some of its concepts have converged with the Xeon Scalable series.

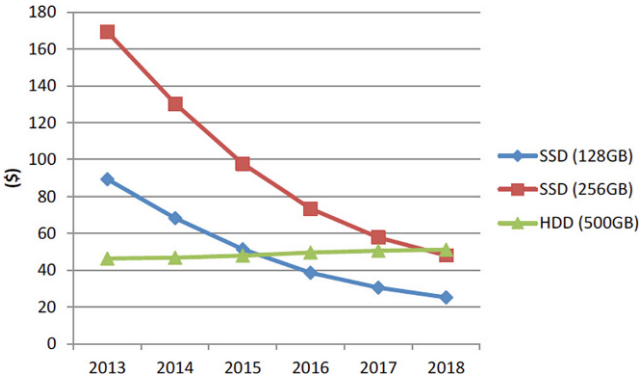

Fig. 2 Worldwide SSD and HDD average selling price [47]

3 Non-Volatile Storage

Flash-based solid-state drives (SSDs) were introduced in the early 90's and, while initially there were many challenges to be addressed, nowadays they have proven to be a usable and reliable technology. Similar to hard disk drives (HDDs), SSDs are non-volatile block-based devices. In contrast to HDDs, SSDs are purely electronic storage devices, i.e., without any moving parts, and with a much better performance. The purely electronic nature of SSDs enables a higher degree of parallelism, which also reduces the performance gap between sequential and random accesses, an important aspect that had to be considered when developing systems for HDDs. Furthermore, modern SSDs present an IO latency of tens of microseconds, while HDDs still have access latencies in the order of milliseconds.

Initial disadvantage of SSDs were higher costs and limited write endurance. However, today SSDs offer enough write endurance and on-chip wear-leveling that most application do not have to worry about such issues. A study [25] with 12 different SSDs showed that most of them only reach a wear-out scenario after about five years of regular usage (40 GB writes per day). The five years interval is similar to the warranty time offered by most HDD manufacturers. Figure 2 shows the average selling prices of SSDs and HDDs. While SSDs are still more expensive, the price is dropping significantly, while the price of HDDs has stabilized in the past years.

Another aspect worth noting is the power consumption. For consumer-level storage devices, SSDs tend to consume much less power than HDDs. However, this is not necessarily the case for high-capacity, enterprise-level storage devices. For example, according to their respective specifications, the Seagate Exos X12 HDD [49] consumes 9.3 Watts during operation, while the Intel DC P4510 SSD [20] consumes 16 Watts. Nevertheless, the power consumption relative to the offered performance makes SSDs a much more attractive option in terms of power savings. Finally, although different flash technologies exist, NAND flash memory became the standard in modern SSDs due to its higher density and better endurance, which translates into lower costs.

3.1 Density

Although most modern SSDs are based on NAND flash memory, they differ on the way these cells are organized internally. While the early designs stored a single bit per cell in a Single-Level Cell structure (SLC), later improvements were made to increase the density with Multi-Level Cell (MLC) and Triple-Level Cell (TLC), storing respectively two and three bits per cell. The increased density of MLC and TLC comes at the cost of reduced write performance, higher power consumption, and lower cell endurance. Most modern enterprise SSDs are either MLC or TLC, as the cost of SLC is prohibitive for systems handling large amounts of data.

Even with the higher densities enabled by MLC and TLC, it became harder to further scale capacity, since the device becomes much more error prone as more bits are packed in a single cell and fewer electrons trapped in the cell correspond to a bit. Manufacturers have solved this issue by stacking cells vertically, enabling more cells while maintaining the same surface on a single die. This technology is known as 3D NAND Flash.

In comparison to 2D geometries, the vertical stacking of cells can increase the capacity of SSDs by two orders of magnitude. The vast majority of modern SSDs are based on 3D flash memory. Furthermore, a recent study [48] based on data collected over 6 years in Google data centers showed that the reliability of modern MLC devices is comparable to that of SLC devices, reducing the range of use cases for SLC. The same study also shows that, while SSDs had a lower replacement rate than HDDs, the rate of uncorrectable errors was higher.

3.2 Performance

SSD flash cells are organized into packages of a certain capacity. While this capacity can be increased by adding more dies on a flash package, this has a negative impact on performance, as the access times of a single die increase. However, the decrease in performance is compensated by increasing the internal parallelism within and across packages. The SSD controller uses multiple individual channels to communicate to the packages. Thanks to this intrinsic parallelism, SSDs achieve a much higher bandwidth by striping data and interleaving accesses across packages.

The increase in performance made it necessary to adapt both hardware and software interfaces, as these became the bottleneck. Initially SSDs adopted the same Serial ATA (SATA) interface found in most HDDs. However, as

the SATA interface could not keep up with the potential bandwidth, many manufacturers started to offer SSDs with a more performant PCI Express (PCIe) interface. PCIe was later established with the standardization of the NVMe specification. While SATA SSDs are still sold on the market, the tendency is that these will be replaced by PCIe/NVMe in the near future. For instance, an Intel DC S4600 Series (SATA) [18] offers a bandwidth of up to 500 MB/s while the Intel DC P4510 (PCIe) [20] offers a bandwidth of up to 3000 MB/s.

Another alternative to SATA offered by some manufacturers is the Serial Attached SCSI (SAS) interface. While SAS performance is much better than SATA, but generally worse than PCIe, its key advantage over PCIe is the flexibility regarding extending a server's storage capacity simply by adding additional SAS devices to multi-port arrays. PCIe is not easily extensible: the devices have to be connected directly to the chipset.

While the underlying media (3D NAND flash memory) and hardware interface (PCIe) are the same for most SSDs, other aspects have critical impact on the performance characteristics of the final product. Some examples are the degree of parallelism, amount of cache, and the controller. The controller itself plays a major role, as it is responsible for multiple functionalities such as error detection and correction, bad block mapping, compression, wear leveling, and garbage collection. Therefore, it is common for a single manufacturer to offer a wide range of products aimed at specific scenarios (e.g., read/write optimized, mixed workloads, large capacities, and low latency).

Finally, software changes are also required to fully exploit the potential of modern SSDs. To that aim, efforts exist to allow the application to bypass the SSD controller and have full control over the behavior of the device. This is achieved by exposing inner SSD interfaces and allowing the application to optimize aspects such as parallelism and wear-leveling for specific use cases. This class of devices is known as Open-Channel SSDs and are already supported by Linux kernel through the LightNVM subsystem [4].

3.3 Recent and Future Developments

As the performance of SSDs keep improving, the bottleneck shifts to other components in the system's stack. The vast majority of modern storage devices still operate through a block-level interface. The mismatch between the application representation (e.g., objects) and the block representation requires data to be converted when reading or writing to the storage device. To lift the overhead introduced by this conversion, Samsung has announced a Key-Value SSD [46] which implements the usual logic present in key-value stores in the SSD firmware, allowing the application to interact with the device in a much simpler way. Moreover, Intel announced its Optane [17] line of products. Different than 3D NAND flash, Optane is based in the Intel 3D XPoint memory technology. The 3D XPoint technology does not only promise to offer 3–10 times lower latency, but also a much higher endurance than 3D NAND Flash.

While the performance of accessing local NVMe SSDs has greatly improved, leveraging these improvements over the network became a challenge. Recent work [24] has proposed a system with a tight coupling of network and storage in order to fully leverage NVMe SSDs performance and enable remote access latencies comparable to local ones. Many works such as Li et al. [33] and Levandoski et al. [32] have presented data structures designed to better exploit the characteristics of flash-based SSDs. More specific to the context of database systems, works such as Hardock et al. [14, 15] have exploited native flash management to reduce write amplification in presence of small updates.

3.4 HDD & Tape

For many decades, HDDs have been the default storage media for the vast majority of systems. Rapid advancements in SSD technologies raise the question whether HDDs will become obsolete in the near future. Even with lower performance characteristics, HDDs still offer a lower price per Gigabyte and potentially higher reliability than SSDs, which would make them a good alternative for archival and backup storage. In such a scenario, HDDs would compete directly with tape-based storage, which surprisingly-enough is still around.

A recent study [1] compared the characteristics of HDD and tape for archival and backup purposes. We highlight here three important observations. First, density of tape-based storage has been increasing in a higher rate than that of HDDs. Second, tape-based storage has a much lower idle power consumption. Third, the bandwidth of sequential access on tape can match that of modern HDDs. Based on these observations, one may wonder whether HDDs will still have a place in the storage hierarchy in the near future, as they are currently not fast enough to compete with SSDs, and not economic enough to compete with tape.

4 Storage-Class Memory

"The arrival of high-speed, non-volatile storage devices, typically referred to as storage-class memories (SCM), is likely the most significant architectural change that datacenter and software designers will face in the foreseeable future." [39].

Table 1 Comparison of the characteristics of SLC Flash and DRAM with those of several SCM candidates [37]

Parameter	SLC Flash	DRAM	PCM	STT-MRAM	RRAM
Read Latency	25 µs	50 ns	50 ns	10 ns	10 ns
Write Latency	500 µs	50 ns	500 ns	50 ns	50 ns
Byte-addressable	No	Yes	Yes	Yes	Yes
Endurance	10^4–10^5	$> 10^{15}$	10^8–10^9	$> 10^{15}$	10^{11}
Density	High	Low	Medium	Low	High

Storage-Class Memory[2] (SCM) is a class of novel memory technologies that exhibit characteristics of both storage and main memory: They combine the non-volatility, density, and economic characteristics of storage (e.g., flash) with the byte-addressability and a latency close to that of DRAM (albeit higher). Examples of such memory technologies include Resistive RAM [13] (researched by SK Hynix, SanDisk, Crossbar, Nantero), Magnetic RAM [10] (researched by IBM, Samsung, Everspin), and Phase-Change Memory [27] (researched by IBM, HGST, Micron/Intel). Table 1 summarizes the projected characteristics of these technologies and compares them to those of SLC Flash and DRAM. In particular, Intel and Micron announced an SCM technology, called 3D XPoint[3] [35], in the Dual Inline Memory Module (DIMM) form factor. SCM technologies are expected to exhibit asymmetric latencies, with writes being noticeably slower than reads, and limited write endurance (although SCM may be significantly more durable than flash memory, e.g., 3D XPoint by three orders of magnitude). Moreover, SCM will be denser than DRAM, yielding larger memory capacities. Finally, in contrast to DRAM that constantly consumes energy to refresh its state, idle SCM does not consume energy – only active cells do. Hence, SCM has the potential to lift the scalability issues of DRAM, both in terms of capacity and energy consumption.

Given its unique characteristics, SCM can serve as fast storage or as DRAM replacement. However, while SCM is projected to be cheaper than DRAM, it will be at first too expensive to replace flash. Additionally, it will be too slow at first to replace DRAM. Nevertheless, we foresee that SCM will be invaluable in extending main-memory capacity in large scale-up systems. Additionally, it can serve as a cheaper DRAM alternative when performance is not paramount. We argue, however, that these use cases do not harness the full potential of SCM: They do not exploit its non-volatility. A third option would be to use SCM as *persistent main memory*, i.e., as memory and storage at the same time. Given its byte-addressability and low latency, processors will be able to access SCM directly with load/store instructions. Both Microsoft Server [23] and Linux [34] already support this access method, called Direct Access (DAX), by offering zero-copy memory mapping that bypasses DRAM and grants the application layer direct access to SCM, as illustrated in Fig. 3.

While SCM brings unprecedented opportunities as a potential universal memory, it fulfills the *no free lunch* folklore conjecture and raises unprecedented challenges as well. To store data, software has traditionally assumed block-addressable devices, managed by a file system and accessed through main memory. The programmer holds full control over when data is persisted and the file system takes care of handling partial writes, leakage problems, and storage fragmentation. As a consequence, database developers are used to ordering operations at the logical level, e.g., writing an undo log before updating the database. SCM invalidates these assumptions: It becomes possible to access, read, modify, and persist data in SCM using load and store instructions at a CPU cache-line granularity. The journey from CPU registers to SCM is long and mostly volatile, including store buffers and CPU caches, leaving the pro-

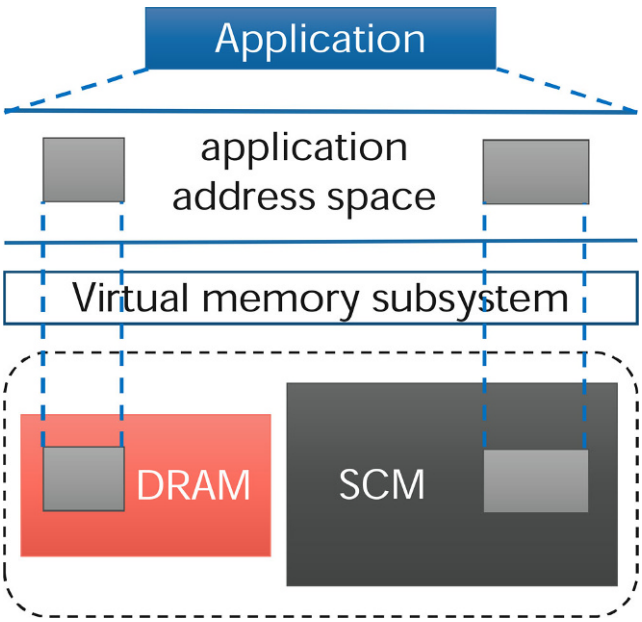

Fig. 3 SCM is mapped directly into the address space of the application, allowing direct access with load/store semantics

[2] SCM is also referred to as Persistent Memory, Non-Volatile RAM (NVRAM), or simply Non-Volatile Memory.

[3] Intel and Micron did not disclose so far the technology that 3D XPoint is based on, albeit it has been speculated that it is based on Phase-Change Memory [8].

grammer with little control over when data is persisted. Even worse, compilers and CPUs might speculatively reorder writes. Therefore, there is a need to enforce the order and durability of SCM writes at the system level (in contrast to the logical level) using persistence primitives, such as memory barriers and cache-line flushing instructions, often in a synchronous way. This, in turn, creates new failure scenarios, such as missing or misplaced persistence primitives, which can lead to data corruption in case of software or power failure. As a consequence, leveraging SCM as persistent main memory requires devising a novel programming model.

The last few years have seen a surge in research efforts that investigate how database systems can leverage SCM as persistent main memory. These research efforts can be categorized into: SCM memory management [19, 31, 44, 45], SCM-based persistent data structures [6, 28, 43, 53], optimizing database algorithms for SCM [7, 51], new testing frameworks for SCM-based software [26, 42], improving the database logging infrastructure [11, 16, 52], and finally exploring novel, SCM-enabled database storage architectures [2, 22, 41].

5 Summary and Outlook

In this "Kurz Erklärt", we briefly explored the increasing diversity in memory and storage technologies, and highlighted the rise of SCM as a potential universal memory. While this diversity brings opportunities for building workload-optimal memory and storage systems, it also pushes more complexity from the hardware layer to the software layer by exposing low-level hardware management features that were traditionally transparent to systems developers. Nevertheless, cloud providers can abstract away this complexity for cloud users through virtualization. Ideally, it should be possible to build custom memory hierarchies simply by provisioning the desired resources in the cloud. Ensuring quality of service will be the biggest challenge to achieve this vision.

In addition to SCM, more groundbreaking innovations await on the horizon. First, Processing In Memory (PIM), which requires embedding compute logic on memory devices, is becoming an attractive hardware acceleration method. For instance, Borumand et al. [5] showed that PIM can halve both energy consumption and execution time by reducing data movement in widely-used Google customer workloads. Second, two large consortia, Gen-Z [12] and OpenCAPI [40], proposed novel memory communication protocols that enable a scalable and flexible hardware topology accounting for heterogeneous memories and accelerators. These and other advancements promise to keep research in memory and storage systems an exciting field!

References

1. Appuswamy R, Borovica R, Graefe G, Ailamaki A (2017) The five minute rule thirty years later and its impact on the storage hierarchy. Proceedings of the 7th International Workshop on Accelerating Analytics and Data Management Systems Using Modern Processor and Storage Architectures.
2. Arulraj J, Perron M, Pavlo A (2016) Write-behind logging. Proceedings VLDB Endowment 10(4):337–348
3. Bjørling M (2018) Open-Channel Solid State Drives. https://openchannelssd.readthedocs.io/en/latest/. Accessed 25 Apr 2018
4. Bjørling M, González J, Bonnet P (2017) LightNVM: the Linux open-channel SSD subsystem. 15th USENIX Conference on File and Storage Technologies (FAST), USENIX Association, pp 359–374
5. Boroumand A, Ghose S, Kim Y, Ausavarungnirun R, Shiu E, Thakur R, Kim D, Kuusela A, Knies A, Ranganathan P et al (2018) Google workloads for consumer devices: mitigating data movement bottlenecks. Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems, ACM, pp 316–331
6. Chen S, Jin Q (2015) Persistent B+-trees in non-volatile main memory. Proceedings VLDB Endowment 8(7):786–797
7. Chen S, Gibbons PB, Nath S (2011) Rethinking database algorithms for phase change memory. Fifth Biennial Conference on Innovative Data Systems Research (CIDR), pp 21–31
8. Choe J (2017) Intel 3D XPoint Memory Die Removed from Intel Optane PCM (Phase Change Memory). http://techinsights.com/about-techinsights/overview/blog/intel-3D-xpoint-memory-die-removed-from-intel-optane-pcm/. Accessed 25 Apr 2018
9. Dayarathna M, Wen Y, Fan R (2016) Data center energy consumption modeling: a survey. IEEE Commun Surv Tutor 18(1):732–794
10. Dong X, Wu X, Sun G, Xie Y, Li H, Chen Y (2008) Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement. 45th ACM/IEEE Design Automation Conference., IEEE, pp 554–559
11. Fang R, Hsiao HI, He B, Mohan C, Wang Y (2011) High performance database logging using storage class memory. IEEE 27th International Conference on Data Engineering (ICDE). IEEE, Hannover, pp 1221–1231
12. Gen-Z Consortium (2018) Gen-Z Core Specification 1.0. https://genzconsortium.org/specifications/core-specification-1-0/. Accessed 25 Apr 2018
13. Govoreanu B, Kar G, Chen Y, Paraschiv V, Kubicek S, Fantini A, Radu I, Goux L, Clima S, Degraeve R et al (2011) 10x10nm 2 Hf/HfO x crossbar resistive RAM with excellent performance, reliability and low-energy operation. IEEE International Electron Devices Meeting (IEDM). IEEE, Washington, pp 31–36
14. Hardock S, Petrov I, Gottstein R, Buchmann A (2013) NoFTL: database systems on FTL-less flash storage. Proceedings VLDB Endowment 6(12):1278–1281
15. Hardock S, Petrov I, Gottstein R, Buchmann A (2017) From in-place updates to in-place appends: revisiting out-of-place updates on flash. Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD). ACM, Chicago, pp 1571–1586
16. Huang J, Schwan K, Qureshi MK (2014) NVRAM-aware logging in transaction systems. Proceedings VLDB Endowment 8(4):389–400
17. Intel (2017) Optane SSD DC P4800X Series. https://www.intel.com/content/www/us/en/products/memory-storage/solid-state-drives/data-center-ssds/optane-dc-p4800x-series/p4800x-750gb-2-5-inch.html. Accessed 25 Apr 2018

18. Intel (2017) SSD DC S4600 Series. https://www.intel.com/content/www/us/en/products/memory-storage/solid-state-drives/data-center-ssds/dc-s4600-series/dc-s4600-1-9tb-2-5inch-3d1.html. Accessed 25 Apr 2018
19. Intel (2018) Persistent Memory Development Kit. http://pmem.io/pmdk/libpmem/. Accessed 25 Apr 2018
20. Intel (2018) SSD DC P4510 Series. https://www.intel.com/content/www/us/en/products/memory-storage/solid-state-drives/data-center-ssds/dc-p4510-series/dc-p4510-8tb-2-5inch-3d2.html. Accessed 25 Apr 2018
21. ITRS (2015) International Technology Roadmap for Semiconductors 2.0, Beyond CMOS. http://www.semiconductors.org/clientuploads/Research_Technology/ITRS/2015/6_2015%20ITRS%202.0%20Beyond%20CMOS.pdf. Accessed 25 Apr 2018
22. Kimura H (2015) FOEDUS: OLTP engine for a thousand cores and NVRAM. Proceedings of the 2015 ACM International Conference on Management of Data (SIGMOD). Melbourne, ACM, pp 691–706
23. Klima T (2016) Using Non-volatile Memory (NVDIMM-N) as Byte-Addressable Storage in Windows Server 2016. https://channel9.msdn.com/events/build/2016/p470. Accessed 25 Apr 2018
24. Klimovic A, Litz H, Kozyrakis C (2017) Reflex: remote flash ≈ local flash. Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). ACM, Xi'an, pp 345–359
25. Kraft B (2017) SSD-Langzeittest beendet: Exitus bei 9,1 Petabyte. https://www.heise.de/newsticker/meldung/SSD-Langzeittest-beendet-Exitus-bei-9-1-Petabyte-3755009.html. Accessed 25 Apr 2018
26. Lantz P, Rao DS, Kumar S, Sankaran R, Jackson J (2014) Yat: a validation framework for persistent memory software. USENIX Annual Technical Conference, pp 433–438
27. Lee BC, Zhou P, Yang J, Zhang Y, Zhao B, Ipek E, Mutlu O, Burger D (2010) Phase-change technology and the future of main memory. IEEE Micro. https://doi.org/10.1109/MM.2010.24
28. Lee SK, Lim KH, Song H, Nam B, Noh SH (2017) Wort: write optimal radix tree for persistent memory storage systems. 15th USENIX Conference on File and Storage Technologies (FAST). USENIX Association, Santa Clara, pp 257–270
29. Lehner W (2017) The data center under your desk: how disruptive is modern hardware for DB system design? Proceedings VLDB Endowment 10(12):2018–2019
30. Lehner W, Ungethüm A, Habich D (2018) Diversity of processing units. Datenbank Spektrum 18(1):57–62
31. Lersch L, Oukid I, Schreter I, Lehner W (2017) Rethinking DRAM caching for LSMs in an NVRAM environment. 21th European Conference on Advances in Databases and Information Systems (ADBIS). Springer, Nicosia
32. Levandoski JJ, Lomet DB, Sengupta S (2013) The Bw-tree: a B-tree for new hardware platforms. IEEE 29th International Conference on Data Engineering (ICDE). IEEE, Brisbane, pp 302–313
33. Li Y, He B, Yang RJ, Luo Q, Yi K (2010) Tree indexing on solid state drives. Proceedings VLDB Endowment 3(1-2):1195–1206
34. Linux (2018) LIBNVDIMM Documentation. https://www.kernel.org/doc/Documentation/nvdimm/nvdimm.txt. Accessed 25 Apr 2018
35. Micron (2018) 3D XPoint Technology. https://www.micron.com/products/advanced-solutions/3d-xpoint-technology. Accessed 25 Apr 2018
36. Mike P, Intel (2016) An Intro to MCDRAM (High-Bandwidth Memory) on Knights Landing. https://software.intel.com/en-us/blogs/2016/01/20/an-intro-to-mcdram-high-bandwidth-memory-on-knights-landing. Accessed 25 Apr 2018
37. Mittal S, Vetter JS (2016) A survey of software techniques for using non-volatile memories for storage and main memory systems. IEEE Trans Parallel Distrib Syst 27(5):1537–1550
38. Mutlu O (2013) Memory scaling: a systems architecture perspective. Fifth IEEE International Memory Workshop (IMW). IEEE, Monterey, pp 21–25
39. Nanavati M et al (2016) Non-volatile storage: implications of the datacenter's shifting center. Commun Acm 50(1):58–63
40. OpenCAPI Consortium (2018) OpenCAPI Specifications. https://opencapi.org/technical/specifications/. Accessed 25 Apr 2018
41. Oukid I, Lehner W, Kissinger T, Willhalm T, Bumbulis P (2015) Instant recovery for main-memory databases. Seventh Biennial Conference on Innovative Data Systems Research (CIDR).
42. Oukid I, Booss D, Lespinasse A, Lehner W (2016) On testing persistent-memory-based software. Proceedings of the 12th International Workshop on Data Management on New Hardware. ACM, San Francisco, p 5
43. Oukid I, Lasperas J, Nica A, Willhalm T, Lehner W (2016) FPtree: a hybrid SCM-DRAM persistent and concurrent B-tree for storage class memory. Proceedings of the 2016 ACM International Conference on Management of Data (SIGMOD). ACM, San Francisco, pp 371–386
44. Oukid I, Booss D, Lespinasse A, Lehner W, Willhalm T, Gomes G (2017) Memory management techniques for large-scale persistent-main-memory systems. Proceedings VLDB Endowment 10(11):1166–1177
45. van Renen A, Leis V, Kemper A, Neumann T, Hashida T, Oe K et al (2018) Managing non-volatile memory in database systems. Proceedings of the 2018 ACM International Conference on Management of Data (SIGMOD). ACM, Houston. (to appear)
46. Samsung (2017) Key value SSD. https://www.samsung.com/us/labs/pdfs/collateral/Samsung_Key_Value_Technology_Brief_v7.pdf. Accessed 25 Apr 2018
47. SanDisk (2015) The SSD enabled PC total cost of ownership. https://www.sandisk.com/business/datacenter/resources/white-papers/the-ssd-enabled-pc-total-cost-of-ownership. Accessed 25 Apr 2018
48. Schroeder B, Lagisetty R, Merchant A (2016) Flash reliability in production: the expected and the unexpected. 14th USENIX FAST.
49. Seagate Technology (2017) HDD Exos X12. https://www.seagate.com/www-content/datasheets/pdfs/exos-x-12-DS1946-1-1709US-en_US.pdf. Accessed 25 Apr 2018
50. Statista (2017) Global market share held by DRAM chip vendors 2011-2017. https://www.statista.com/statistics/271726/global-market-share-held-by-dram-chip-vendors-since-2010/. Accessed 25 Apr 2018
51. Viglas SD (2014) Write-limited sorts and joins for persistent memory. Proceedings VLDB Endowment 7(5):413–424
52. Wang T, Johnson R (2014) Scalable logging through emerging non-volatile memory. Proceedings VLDB Endowment 7(10):865–876
53. Yang J, Wei Q, Wang C, Chen C, Yong K, He B (2015) NV-tree: a consistent and workload-adaptive tree structure for non-volatile memory. IEEE Trans Comput 65(7):2169–2183