Storage Systems: Organization, Performance, Coding, Reliability, and Their Data Processing, 1st Edition, October 13, 2021, Author: Alexander Thomasian

A. Thomasian
Thomasian and Associates
Pleasantville, NY
alexthomasian@gmail.com

Why This Book?

Textbooks on computer organization and architecture, operating systems, and databases do not sufficiently cover storage systems. This is especially so in view of rapid advances in storage technology in the form of flash and *Storage Class Memories - SCMs* and storage organizations such as data-centric and the DisAggregated Shared Everything - DASE.

Garth Gibson's PhD thesis provides the RAID classification, but mainly deals with reliability modeling [1]. RAID is also a proposal to replace expensive, large form factor, high capacity disks associated with mainframe computers known as Direct Access Storage Devices - DASD with an array of low cost, small form factor and capacity disks used in personal computers. At this time all high capacity *Hard Disk Drives - HDDs* have a small form factor with *Fixed Block Architecture - FBA* with 4 KB sectors. The last DASD was the IBM 3390 was last produced in May 1993.

Only 6 out of 105 pages on storage systems are dedicated to RAID in the latest 6th edition of Hennessey and Patterson Computer Architecture book [2]. A collection of 45 "influential" papers on storage systems is presented in [3], but given that most readers have access to digital libraries about 1700 references are provided in the reviewed book [4].

Storage systems are reviewed from different viewpoints as reflected by the book's title. Given the rapid evolution in the field readers are encouraged to follow storage conferences such as USENIX's File and Storage Technologies - FAST and Mass Storage and Technology - MSST:
`https://www.usenix.org/conference/fast23/storage/conference.us` and trade publications such as:
`https://blocksandfiles.com/2023/07/05/legacy-external-storage/`
`https://www.theregister.com/2020/10/09/key_role_of_storage/`

Desirable attributes for storage systems are high capacity, low latency, high data access rates (randomly placed data blocks), high data transfer bandwidth (sequential data), no hiccups (write amplification, writing to flash delayed by garbage collection, writing onto HDDs with *Shingled Magnetic Recording - SMR*), *Flash Translation Layer - FTL* to balance wear and picking blocks, nonvolatility (achieved by Uninterruptible Power Supply - UPS), shock resistance, absence of mechanical parts, low cost per GB, and low power consumption. durability (improved by RAID level), lifespan (HDDs and NAND flash SSD, 3-5 years, M-disc 1000 years).

Low latency and high access rates are important in *On-Line Transaction Processing - OLTP* and e-commerce applications, which require high access rates to randomly placed disk blocks. Data mining requires high transfer rates to access large volumes of data, as in the case of Birch data clustering method [4]. Very high archival storage capacity is achievable by robotic libraries of magnetic tapes.

While hyperscalers (large disk farms) are based on HDDs, a second paradigm shift is due to flash *Solid-State Drives - SSDs* replacing HDDs due to lower access time and high bandwidth. Lower power consumption and higher reliability resulting in a lower TCO.

Flash SSDs have lower access times and bandwidths: DRAM: 75 ns, 2.4 GB/s; SCM 179 ns, 2.4 GB/s; SSD 85 μs (microseconds), 1.6 GB/s; HDD mean seek time plus T_{rot} = [3600/RPM]/2 divided by two, 100-200 MB/s. The transfer time of small blocks tends to be negligibly short. SCMs serve as cache frontends to SSDs reducing flash wearout.

One intent of the text is to familiarize the reader with storage technologies used by computers. As the volume of data is increasing exponentially storage efficiency can be increased via lossless data compression, which depends on data category: text (Huffman, Lempel-Ziv), audio, images (JPEG), video (MPEG). Data deduplication is used to reduce the volume of data for archiving. Data encryption is used for higher data security and privacy.

The RAID paradigm utilizes replication or erasure coding to deal with failures. Data is replicated at multiple distributed sites to deal with disasters: earthquakes, floods, fires, power outages, but additional protection is required against ransomware, computer viruses.

Most data processing used to be carried out using disk-resident files and later databases. Given complex SQL queries higher processing efficiency was achievable by selecting appropriate indexing and applying sophisticated query optimization methods. *Log Structured Merge Trees - LSMTs* are more appropriate than B+ trees in SSDs. New algorithms have been developed for NoSQL, multimodel, document, graph, key-value stores. Current research is now focused on data processing on flash and other storage technologies.

Maximum Disk Separable - MDS RAID$(4+k), k \geq 1$ disk arrays use the minimum redundancy level, k redundant disks to tolerate k disk failures. RAID6 utilizes RS codes, but IBM's EVENODD and NetApp;s Row-Diagonal Parity - RDP are two parity–based implementations incurring lower computational cost than RS. Both have been extended to 3DFTs. Similarly RAIDIX RAID7.3 uses $k = 3$ parity disks

to protect 45 data HDDs.

When $j \leq k$ disks fail in RAID(4+K) with $N+k$ disks fail their contents can be reconstructed on demand by accessing the corresponding blocks on N disks. Noting that with j failed disks the read load is increased (j+1)-fold so provided sufficient spare space is available rebuild processing should be started right away. With $j = k$ disk failures the system is in a critical state, since more failure will lead to data loss.

A common reason for data loss during rebuilds are *Latent Sector Errors - LSEs*, which can be dealt with disk scrubbing and IntraDisk Redundancy - IDR. In [8] we review techniques based on ML which predict disk failures.

In the case of RAID6 preferably two failed disks are rebuilt simultaneously, rather than one disk at a time. Rebuild processing in multi-TB HDDs can takes many hours and there is the possibility of data loss due to additional disk failures, especially if they are correlated.

Pyramid codes use Local Redundancy Codes - LRC for low cost recovery and they were later utilized by Windows Azure distributed storage system to reduce communication cost.

Data Base Machines - DBMs use parallel processing to achieve high-performance DB processing. Tandem/HP NonStop SQL and Teradata DBC/1012 both initially used mirrored disks for higher reliability.

Two chapters are dedicated to author's research dealing with Heterogeneous Disk Arrays - HDAs, Hierarchical RAID, and related studies.

Intended Audience and Background Requirements

The book is intended for upper division undergraduate and master level computer science and engineering, electrical engineering, and data and information science and technology students and professionals. The book is interspersed with discussion of performance and reliability analysis.

At this time computing permeates all areas of engineering, science, technology, and business hence knowledge of computer hardware and software and associated companies is necessary for all professionals even students choosing a laptop for high school.

Background in basic computer science, mathematics (discrete, probability, and Markov chains), Brief tutorial material is provided on reliability modeling and queueing theory.

In addition to references in the bibliography a list of relevant books and various sources is provided in the Appendix.

Book Chapters

The book TOC has a full list of book topics.

Chapter 1: Introduction. We start with history of companies and their computers in commercial, scientific, engineering, and military applications after WW II. IBM introduced the S/360 computer family in 1964 which led in 1967 for to it to become the most highly valued company S/360 computers now called IBM Zsystems were extended with caches, virtual memory, 64 bit addressing, data compression, encryption. Companies in the BUNCH (Burroughs, Univac, NCR, CDC, Honeywell) plus DEC and HP had their own proprietary computers and software, which has since disappeared while Wintel - Microsoft Windows and Intel are dominant. HDDs are produced by Seagate, Western Digital etc. and disk arrays: Dell/EMC, HPE, NetApp, etc.

Chapter 2: Storage Technologies. Paper allowed writing and later printing. Punched cards were used to store census data, company records and simple data processing. Magnetic tapes and disks (also optical) held data, images, audio and video. Tapes and disks were used for batch and disks with random access capabilities for online data processing. Tapes only accessible sequentially are used for batch processing, while disks in allow random access via hashing or indexing useful for OLTP Extensible and linear hashing allows dynamic additions and deletion of records and so do B+trees/VSAM. For high performance flash memories and other SCMs - Storage Class Memories such as PCM - Phase Change Memory are expected to replace magnetic disks. Data compression and deduplication can be used in preserving storage space and network bandwidth and encryption for data security and privacy.

Chapter 3: Disk drive organization, data Placement, and scheduling. Commodity disks with fixed size sectors replaced CKD DASD associated with IBM mainframes. For mainframes running legacy application under z/OS CKD DASD are simulated by the disk array controllers running z/OS on IBM mainframes. Zoned bit recording - ZBR, Shingled Magnetic Recording - SMR and Energy Assisted Magnetic Recording - EAMR provide higher storage capacity. Disk scheduling policies and data placement to minimize seek and access time. Shortest Access Time First - SATF is extended with lookahead and two priority classes. Queueing analysis of disk performance with FCFS (with Zoned Bit Recording - ZBR), SCAN, and SATF policies is discussed. A review RAID performance analyses using queueing theory and simulation is presented.

Chapter 4: Mirrored and hybrid arrays. Mirrored disks, classified as RAID1, protect against disk failures by storing data twice. In a hybrid disk array with half of the disks hold data and the other half *eXclusive ORed - XORed* data: $\text{Disk}_{2n} = \text{Disk}_{2n-1} \oplus \text{Disk}_{(2n+1)\text{mod}(12)}, 1 \leq n \leq 6$. This allows all two disk failures and half of consecutive three disk failures to be tolerated, while there s data loss with basic Mirroring when a disk pair fails. RAID1 doubles disk access time bandwidth, but when a disk fails the load on the surviving disk is doubled. RAID1 configurations which distribute the load of a failed disk over multiple disks achieve a more balanced load. Shortcut reliability analysis shows that BM is most reliable RAID1, but less reliable than hybrid disk arrays Analytic and simulation methods to estimate RAID reliability are presented.

Chapter 5: Variation of RAID with emphasis on RAID(4+k), k=1,2,3. Operation in normal, degraded, and rebuild modes is discussed and their performance analyzed. Clustered RAID5 has a parity group size $G < N$, where N is the number of disks. *Balanced incomplete Block Designs - BIBD*, Thorp shuffle, *Nearly Random Permutations - NRP*, and row shifting are four methods to place parities balance updating of disk loads. The Vacationing Server Model - VSM for RAID5 is analyzed/ It processes rebuilds requests only when the queue of external requests is empty. VSM outperforms the Permanent Customer Model - PCM in rebuild time and external request response time degradation. IntraDisk redundancy - IDR and disk scrubbing methods to deal with Latent Sector Errors - LSEs, which may lead to unsuccessful rebuilds are presented. IDR is a low cost method in terms of space ad performance which obviates the need for higher redundancy levels than RAID5.

Chapter 6: Coding Methods. EVENODD, RDP, and X-code are Maximum Distance Separable - MDS codes in that they incur the minimum level of redundancy to tolerate two disk failures with parity coding. RDP is shown to outperform EVENODD in the number of XORs required as shown by M. Blaum.

Chapter 7: Power Reduction in Storage Systems and Servers. Power consumption is a major problem for server and storage clouds. Server energy can be saved by lowering voltage and in the case of disks by lowering RPMs and spindown. In the case of laptops there is a switch to SSDs which use less power. Massive Arrays of Independent Disks - MAID paradigm leaves a few disk spinning for caching recently accessed data and for new updates. Hybrid disks use a flash memory or NVRAM cache to hold updates while the disk is spundown. Power consumption can be reduced by temporal and spatial clustering of disk accesses. Submerging servers into liquid baths is a new development.

Chapter 8: Database parallelism, big data, analytics, and deep learning. Early Data Base Machines - DBMs mainframes running database management systems, such as IMS. Data mining such as Association Rule Mining on mainframes was considered too costly, since results such as diapers and beer bought together was of questionable value. The advent of low-cost powerful microprocessors allowed the building of highly parallel DBMs for dealing with big data.

Active disks processed high priority disk accesses for OLTP, while processing data mining requests at no cost by freeblock scheduling. Processor per track disks such as the Relational Associative Processor - RAP, are no longer feasible because of high track densities, but this paradigm has been applied to FAWN - FAST Array of Wimpy (flash) Nodes and RamCloud which attains durability by keeping backup copies on secondary storage. It was predicted by Gibson in [1] that DRAM will replace HDDs after 2027.

Extracting information from data for decision making utilizes CPUs, GPUs, FPGAs, Application Specific Integrated Circuits - ASICs, and Tensor Google's Processing Units - TPUs, which is Google's ASIC for *Machine Learning - ML.* with Deep Neural Networks. DPUs - Data Processing Units combine a CPU, GPU, and a network interface.

Chapter 9: Structured, semi-structured, and unstructured data. These include network, hierarchical, and relational databases, big data, Hadoop technology ecosphere, NoSQL databases, key-value stores, document stores, times series, graph, and object-oriented databases, web search engines, Resource Description Framework - RDF, wide column stores, native XML and JSON databases, realtime database, event stores, content store, multimodel databases, main memory databases. Novel applications such as streaming analytics and business intelligence platforms.

Chapter 10: Heterogeneous Disk Arrays - HDAs. allow multiple Virtual Arrays - VAs to share disk space using controllers for multiple RAID levels. The RAID level and organization of VAs is determined by reliability and workload requirements, e.g., RAID1 suited for OLTP with high access rates to small data blocks, while RAID5 suited for parallel accesses to large files. Allowing variable RAID levels conserves disk bandwidth, because not all VAs need to be allocated with the highest redundancy level required by a subset of datasets.

Chapter 11: Hierarchical RAID. HRAID is motivated by the storage bricks paradigm. There are N Storage Node - SN whose controllers implement the intraSN and interSN code with k and ℓ check codes. Thus HRAID uses a multilevel RAID paradigm with interSN erasure coding.

Disk failures are handled by intraSN redundancy firstly and more costly interSN redundancy secondly. An baseline array with no interdisk redundancy is considered for comparison purposes. Operations with and without interSN rebuild processing is considered in simulations, which are also used to determine the effect of varying HRAID parameters on the MTTDL - Mean Time to Data Loss. We disagree on how redundancy is apportioned in IBM's Intelligent Bricks project in [6].

Chapter 12: Some Further Topics. This is to encourage readers to pursue them on their own.

An extension of the seminal RAID tutorial [7] appears in [8] extending some discussions in [5].

References

[1] G. A. Gibson. Redundant Disk Arrays: Reliable Parallel Secondary Storage. The MIT Press 1992.

[2] J. L. Hennessey and D, Patterson. Computer Architecture: A Quantitative Approach, 6th edition. Elsevier 2019.

[3] H. Jin, B. Rajkumar and T. Cortes. High Performance Mass Storage and Parallel I/O: Technologies and Applications IEEE & Wiley Interscience, 2002

[4] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: A new data clustering algorithm and its applications. J. Data Mining and Knowledge Discovery, 1, 2 (1997), 141182.

[5] A. Thomasian. Storage Systems: Organization, Performance, Coding, Reliability, and Their Data Processing, 1st Edition, Oct. 2021, 712 pages
Paperback ISBN: 9780323907965, eBook ISBN: 9780323908092
`https://shop.elsevier.com/books/storage-systems/`
Available chapter by chapter from ScienceDirect.
`https://www.sciencedirect.com/book/9780323907965/storage-systems`
Google books preview, 151 pages (about 20% of the book).
`https://www.google.com/books/edition/Storage_Systems/t8wnEAAAQBAJ`

[6] A. Thomasian. Optimizing Apportionment of Redundancies in Hierarchical RAID. `https://arxiv.org/pdf/2205.06330.pdf`, May 2022.

[7] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz and D. A. Patterson. RAID: High-Performance, Reliable Secondary Storage. ACM Computing Surveys 26, 2 (1994), 145-185.

[8] A. Thomasian. RAID Organizations for Improved Reliability and Performance: A Not Entirely Unbiased Tutorial. `https://arxiv.org/abs/2306.08763`, June 2023.

Alexander Thomasian, Life Fellow, IEEE

With a BSEE Degree from Univ. of Tehran I joined IBM WTC as a Systems Engineer. At Tehran's electric utility developed the billing application for 1/2 million customers. With a CS PhD from UCLA I taught at Case Western Reserve and then Univ. Southern Calif. At Almaden Research Center I analyzed the performance of IBM's RAID5. Back in Yorktown I contributed to a NASA sponsored project for indexing satellite images. At NJIT my research was funded by NSF, Hitachi, and AT&T's Virtual Univ. Research Inst. I was at `siat.cas.cn` (2010-11) and `aua.am` as a Fulbright Fellow (2015). I have four patents, over 150 papers, a book republished by Springer on concurrency control, and supervised ten PhD students.