Big Data Platforms (5 ECTS)
DATA14003
Lecture 5

Keijo Heljanko

Department of Computer Science
University of Helsinki
keijo.heljanko@helsinki.fi

21.9-2021

Exercise: Fault tolerance of RAID 0

Assume a mean time to failure (MTTF) of a hard disk to be 300000 hours. When using 4 TB hard disks in RAID 0 configuration (striping data over all disks), compute the mean time to failure in years of a RAID 0 array that consist of 64 TB of storage?

Hint: The array fails if any one of the hard disk fails in RAID 0 configuration.

Solution: Fault tolerance of RAID 0

Since the system of hard disks is set up in a RAID 0 configuration, the first error that occurs in any of the disks will break the disk array. We know that the error process for each of the K hard disks can be modeled as an Poisson process, where the transition frequency (failing of hard drive) of each drive is λ.

In our case we have $K = (64TB/4TB) = 16$ hard drives.

We also have single drive failure frequency of:
$\lambda = \frac{1}{300000h} = \frac{1}{300000h/(365.244days/year \cdot 24h/day)} \approx \frac{1}{34.224year}$

Solution: Fault tolerance of RAID 0 (cnt.)

The failure frequency F of the array is just:
$F = K \cdot \lambda = \frac{16}{34.224 year} \approx 0.46751/year$.

Thus the mean time to failure is:
$MTTF = \frac{1}{F} = \frac{1}{0.4675} years \approx 2.14 years$

Exercise: URE during RAID 5 rebuild

We do some calculations of the expected number of hard disk read failures (URE, unrecoverable read error) during RAID 5 array rebuilds. Basically if an URE happens, the hard disk gives out an error saying that it can not read a particular data item stored on the hard disk. This in turn will result in a rebuild failure of the RAID 5 array.

When using consumer hard disks in RAID 5 configuration, compute the expected number of URE errors during RAID 5 array rebuild. Use the typical consumer URE rate of 1 bit error per 10^{15} bits read. Assume the RAID 5 array is full of data and consist of 16 TB following amounts of storage space

Hint: The RAID 5 rebuild reads in as much data as the storage array has storage space.

Solution: URE during RAID 5 rebuild

We need to read 16 TB of data during the RAID 5 rebuild. This is $8 bits/byte \cdot 16 \cdot 10^{12} bytes = 128 \cdot 10^{12} bits$.

The expected number of URE errors during RAID 5 rebuild of 16 TB array is $128 \cdot 10^{12} bits \cdot \frac{1 URE\ error}{10^{15} bits} = 0.128 URE\ errors$.

Storage Technologies for the Cloud

The following storage technologies are widely used in the cloud:

- RAM (Random Access Memory)
- Flash (also called SSD - Solid State Drive)
- Hard Disk
- Tape

Flash Storage

- Currently one of the trends is the introduction of Flash memory SSDs (solid state disks) are replacing hard disks in many applications
- Flash capacity per Euro is increasing faster than hard disk capacity per Euro
- Currently Flash capacity is around 10% of all Enterprise storage capacity but is growing quickly
- Random SSD read (and often also write) IOPS can be more than 1000 $\times$ those of high end hard disk read and write IOPS

Flash Storage (cnt.)

- Sequential read and write speeds are much faster than those of the best hard disks
- As SSDs have no moving parts they are fail quite a bit less frequently than hard disks
- As such SSDs are a very good match for typical client laptop and desktop usage patterns

Flash Storage

- Flash has both strengths and limitations compared to hard disks, they are not a direct hard disk replacement for all server workloads
- Especially the write endurance of SSDs (the amount of data that the SSD is specified to write reliably during its lifetime) is quite often very small compared to hard disks
- For write intensive server workloads hard disks might still be the only economically viable option due to SSDs having to be replaced once write endurance is exhausted
- Apart from limited write endurance, Flash drives have typically less failures than hard drives

Flash Organization

- A flash chip is organized in pages of some KBs in size (e.g., 2KB)
- A page read can load the data of a page into buffers that can be quickly randomly accessed
- A page write can only flip bits of the page from 0 to 1
- In order to change the bits from 1 to 0 an erase operation operating on a large number of blocks needs to be performed
- An erase operation works on blocks that consist of several flash pages at the time (e.g., 128KB of data)
- If the blocks to be erased contain some data, that data must be written elsewhere before the block is erased

Flash Types

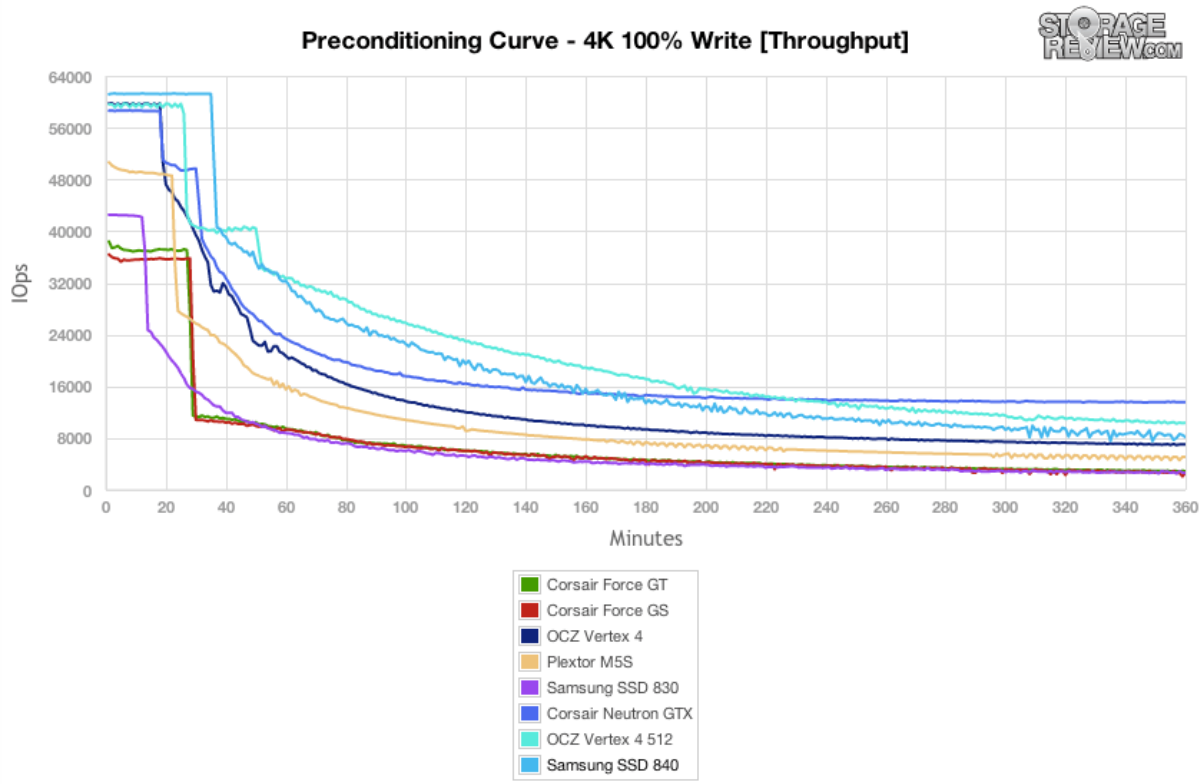
- Flash memory comes in several flavours, the main ones being NAND flash in varieties: quad-level cell QLC (cheapest), triple-level cell TLC (second cheapest), multi-level cell MLC (cheap), and single-level cell SLC (expensive)
- Flash blocks can be erased: around 1000 times (TLC flash), 1000-10000 times (MLC flash), or 100000-1000000 times (SLC flash)

Flash Wear Levelling Algorithms

- Highly sophisticated algorithms are used for wear leveling, where the goal is to write each flash cell as evenly as possible
- When flash disk becomes almost full, the free space can become fragmented which can result in performance loss on writes as the SSD moves blocks around to make space for the written data
- Flash disks often use “spare capacity” set aside to make the fragmentation problems less severe

Sustained Flash Write Performance

- Because of the RAM write caches and the write leveling algoritms, the sustained Flash memory write speeds can sometimes drop for long workloads:
  `http://www.storagereview.com/samsung_ssd_840_pro_review`



Optimizing for Flash

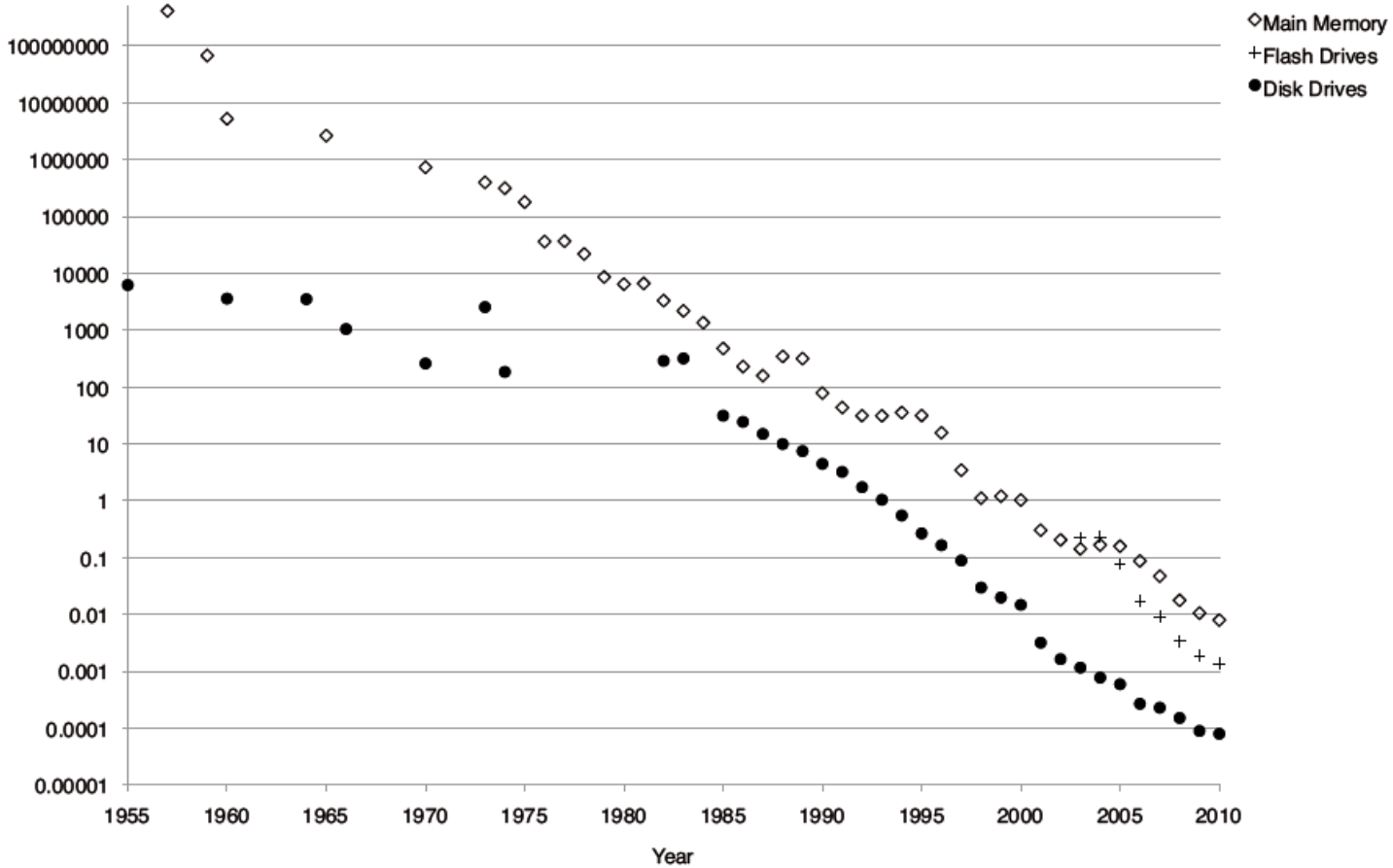
- One should minimize the number of bytes written to Flash
  - Less bytes written means less wear and longer lifetime expectancy of Flash
  - Less bytes written means less frequent slow block erases, and this improves the overall Flash IOPS
- Heavy sequential write workloads (e.g., database logs) can be sometimes be problematic for Flash drives due to write endurance limiting the lifetime of Flash
- Operating systems support such as the TRIM command which tells to the SSD which data blocks can be safely discarded is very useful

Some Rule of Thumb Numbers for Flash

For a single random access read, the following rules of thumb numbers apply:

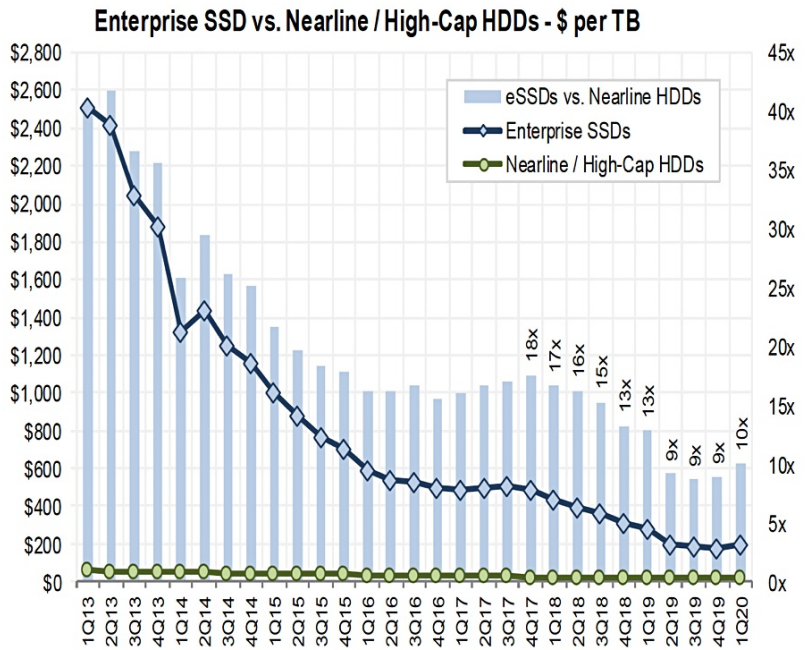
- RAM memory reference: 100 ns
- Flash drive access: 100 000 ns (1000x slower than memory)
- Disk seek: 10 000 000 ns (100 000x slower than memory)

Storage Price Trends



- Trends of RAM, Flash, and HDD prices. From: H. Plattner and A. Zeier: In-Memory Data Management: An Inflection Point for Enterprise Applications

Enterprise Hard Disk vs Flash Prices



Source: TrendFocus; Wells Fargo Securities, LLC

- Currently the Enterprise Flash price premium per TB of storage over high capacity hard drives is still significant
- For small client drives SSDs are already very price competitive with hard disks

Storage Usage Scenarios

- Tapes are still being used for backup purposes as they are cheap per Terabyte
- RAM (and Flash) are radically faster than HDDs: One should use RAM/Flash whenever possible
- RAM is roughly the same price as HDDs were a decade earlier
  - Workloads that were viable with hard disks a decade ago are now viable in RAM
  - One should only use hard disk based storage for datasets that are not yet economically viable in RAM (or Flash)
  - The Big Data applications (HDD based massive storage) should consist of applications that were not economically feasible a decade ago using HDDs

Cost of Storage: RAM vs Flash vs Disk

The paper: John K. Ousterhout et. al: The case for RAMCloud. Commun. ACM 54(7): 121-130 (2011) summarizes the Total Cost of Ownership for Storage Technologies:

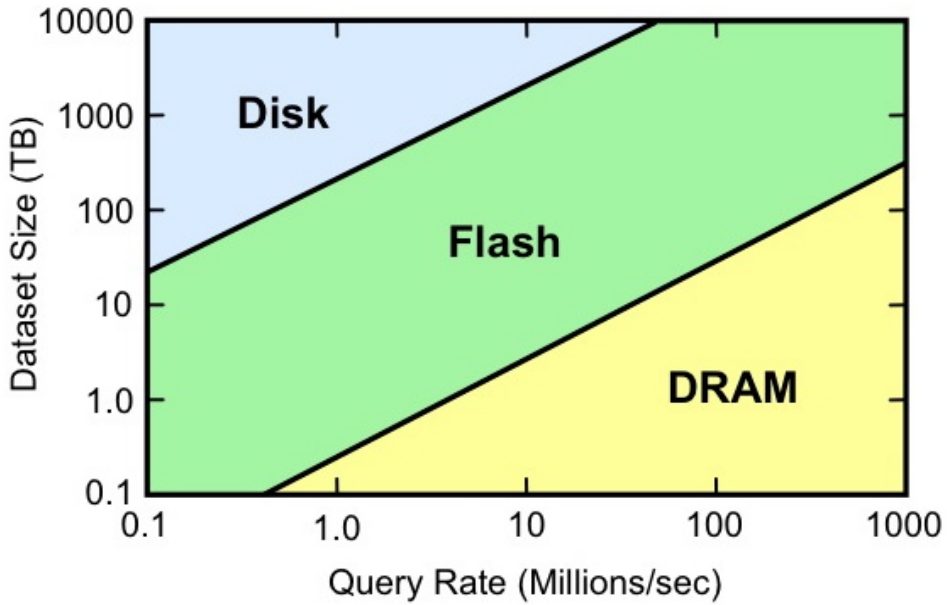


Figure 2. This figure (reproduced from Andersen et al. [1]) indicates which storage technology has the lowest total cost of ownership over 3 years, including server cost and energy usage, given the required dataset size and query rate for an application (assumes random access workloads).

Cost of Storage: RAM vs Flash vs Disk

- Most practical system contain data sets with different “temperatures”
- Hot Storage: When a large number of IOPS/TB is needed, RAM is the cheapest way to obtain lots of IOPS
- Cold Storage: When a large amount of TB/IOPS is needed, Hard Disks are the cheapest way to obtain lots of storage
- Warm Storage: Flash is a compromise between cost per IOPS and cost per TB
- When the pricing of RAM/Flash/HDD changes, the exact optimal selection of best storage technology changes

Example: Facebook Storage

- Facebook has done research on the warmth of their datasets in paper: Subramanian Muralidhar et. al: f4: Facebook's Warm BLOB Storage System

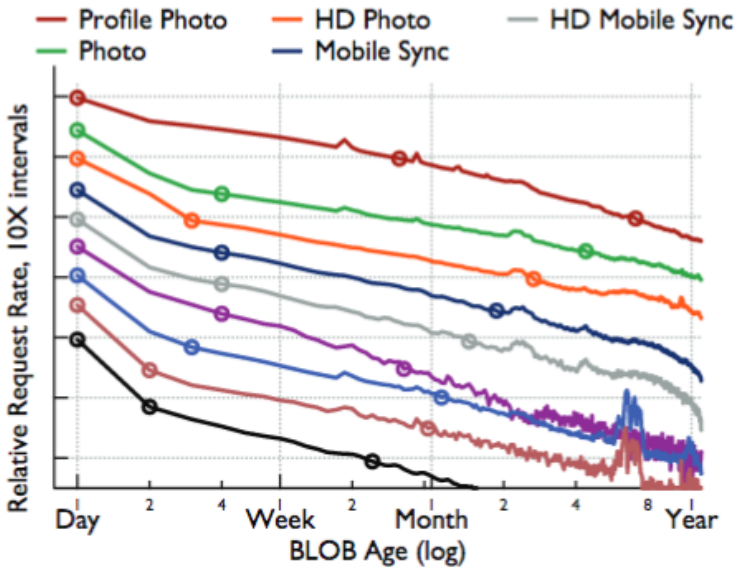


Figure 3: Relative request rates by age. Each line is relative to only itself, absolute values have been denormalized to increase readability, and points mark an order-of-magnitude decrease in request rate.

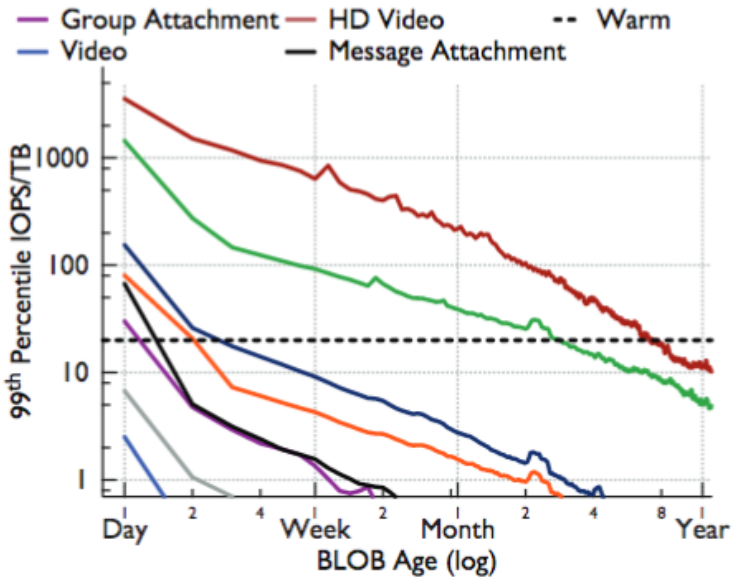


Figure 4: 99th percentile load in IOPS/TB of data for different BLOB types for BLOBs of various ages.

Example: Facebook Storage

- The data stored at Facebook cools as it gets older: The optimal way to store the data is different for new and old files at Facebook
- Facebook has separate storage solutions for Hot, Warm, and Cold data storage
- As data gets older and thus colder, it is migrated from one storage system to another to save costs

Storage Technologies - Jim Gray view

A mantra from Turing Award Winner Jim Gray of Microsoft in his very visionary (from year 2006!) presentation: `http://research.microsoft.com/en-us/um/people/gray/talks/Flash_is_Good.ppt`:

- Tape is Dead
- Disk is Tape
- Flash is Disk
- RAM Locality is King

Jim Gray on Disks (in 2006)

- Disk are cheap per capacity
- Sequential access full disk reads of take hours
- Random access full disk reads take weeks
- Thus most of disk should be treated as a cold-storage archive

Jim Gray on Flash (in 2006)

- Lots of IOPS
- Expensive compared to disks (but improving)
- Limited write endurance
- Slow to write (compared to reading)

Jim Gray on RAM (2006 numbers)

- Flash/disk is 100000-1000000 cycles away from the CPU
- RAM is 100 cycles away from the CPU
- Thus Jim Gray concludes that main memory databases are going to be common

Caching to Improve Hard Disk Performance

- Caching the hot part of the dataset in RAM is a well known technique for improving the performance of a hard disk based storage system
- The large difference in RAM and hard disk random access latencies makes this sometimes difficult, as caches need to have extremely good hit rates for the hard disk accesses not to dominate

RAMCloud

John K. Ousterhout, Parag Agrawal, David Erickson, Christos Kozyrakis, Jacob Leverich, David Mazières, Subhasish Mitra, Aravind Narayanan, Diego Ongaro, Guru M. Parulkar, Mendel Rosenblum, Stephen M. Rumble, Eric Stratmann, Ryan Stutsman: The case for RAMCloud. Commun. ACM 54(7): 121-130 (2011).

- Facebook in 2009 has been running in 2009 with enough RAM to fit 75% of their dataset (not counting images or video) in RAM
- If they would use enough RAM cache to hit 99% of the dataset, the random disk seek latency would still kill their average latency performance: (99% $\times$ 100 ns + 1% $\times$ 10 000 000 ns) = 100099 ns, which is way more that 100ns!

RAMCloud - Discussing the Numbers

- With Flash disks and 99% cache hit rate we get latencies: (99% $\times$ 100 ns + 1% $\times$ 100 000 ns) = 1099 ns, which is still 10x more that 100ns.
- The above concerns only average latency, not throughput, but similar reasoning also applies to aggregate IOPS numbers of RAM, Flash, and Disk
- If a system can have enough memory to have 100% of the working set in RAM, instead of Flash / Hard disk, way better latency can be obtained, and more IOPS can be served with the same number of servers.

RAMCloud - Limiting Factors

- The main problem in RAMcloud style designs to guarantee data persistence in case of power failure + quick recovery in case of server crash / power outage
- New non-volatile memory technologies such as Intel/Micron 3D-Xpoint memories (`https://en.wikipedia.org/wiki/3D_XPoint`) that promise even much faster performance than Flash might be able to help realize this
- The second problem is to have enough low latency networking hardware and OS to keep RAM busy
- For discussion, see: Luiz Barroso et. al: Attack of the killer microseconds, Communications of the ACM, Volume 60 Issue 4, April 2017, Pages 48-54 `https://doi.org/10.1145/3015146`

Improving Disk Based Database Performance

- Sometimes we need to create databases that are much larger than can fit the RAM
- RAM is a very expensive resource, and thus should be utilized as well as possible
- Using a normal cache of hot data in RAM can improve query latency for items that are in the database
- RAM can also be used to improve performance for queries of items not in the database
  - RAM can hold an index of all the items on hard disk. However, sometimes the index size can be larger than RAM
  - Probabilistic data structures use RAM even more efficiently than normal indexes - Example: Bloom filters