Large scale deduplication (Episode 2)

ALEX XU

APR 9, 2022

How to avoid crawling duplicate URLs at Google scale?

Option 1: Use a Set data structure to check if a URL already exists or not. Set is fast, but it is not space-efficient.

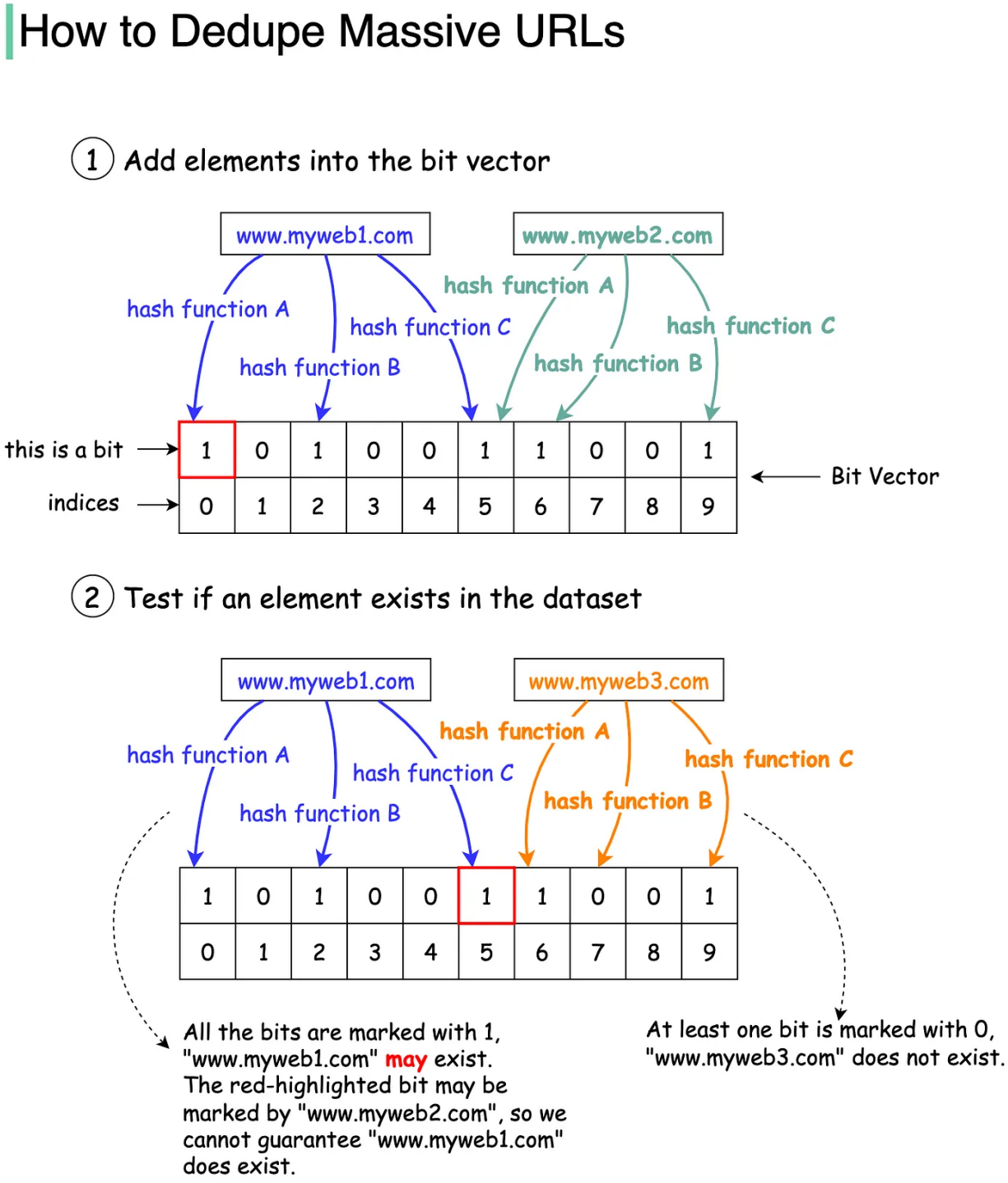
Option 2: Store URLs in a database and check if a new URL is in the database. This can work but the load to the database will be very high.

Option 3: **Bloom filter**. This option is preferred. Bloom filter was proposed by Burton Howard Bloom in 1970. It is a probabilistic data structure, that is used to test whether an element is a member of a set.

- false: the element is definitely not in the set.
- true: the element is probably in the set.

False-positive matches are possible, but false negatives are not.

The diagram below illustrates how the Bloom filter works. The basic data structure for the Bloom filter is Bit Vector. Each bit represents a hashed value.



Step 1: To add an element to the bloom filter, we feed it to 3 different hash functions (A, B, and C) and set the bits at the resulting positions. Note that both "www.myweb1.com" and "www.myweb2.com" mark the same bit with 1 at index 5. False positives are possible because a bit might be set by another element.

Step 2: When testing the existence of a URL string, the same hash functions A, B, and C are applied to the URL string. If all three bits are 1, then the URL may exist in the dataset; if any of the bits is 0, then the URL definitely does not exist in the dataset.

Hash function choices are important. They must be uniformly distributed and fast. For example, RedisBloom and Apache Spark use murmur, and InfluxDB uses xxhash.

Question - In our example, we used three hash functions. How many hash functions should we use in reality? What are the trade-offs?

System design interview tip: Don't optimize prematurely.

In our 4-step system design interview framework, the second step is "Propose High-level Design and Get Buy-in". The objective of this step is to come up with a high-level design diagram for the problem at hand and establish a common ground for further exploration.

It is a red flag to get carried away this early in the session with premature optimizations. For example, some candidates love to talk about caching and sharding in this step.

Optimizations are complicated and expensive. They require solid justifications for the added complexity. At this early stage, the justifications are often hand-wavy. The added complexity distracts the interviewer and prevents them from understanding the high-level design.

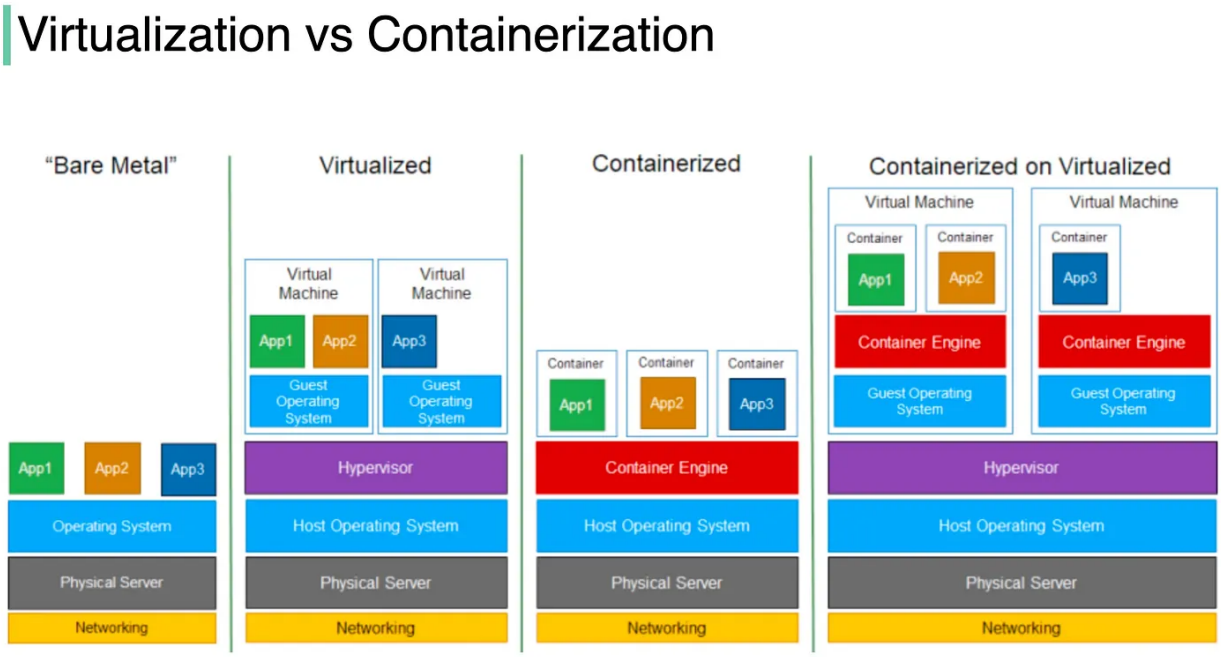
Focus on the task at hand. If you find yourself getting distracted by optimization ideas, table them. Make a list of ideas to revisit in the deep dive section.

Keep the high-level design simple. This step should take about 15 minutes.

What are some other premature optimizations that can be tabled?

What are the differences between Virtualization (VMware) and Containerization (Docker)?

The diagram below illustrates the layered architecture of virtualization and containerization.



"Virtualization is a technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system" [1].

"Containerization is the packaging together of software code with all its necessary components like libraries, frameworks, and other dependencies so that they are isolated in their own "container" [2].

The major differences are:

- In virtualization, the hypervisor creates an abstraction layer over hardware, so that multiple operating systems can run alongside each other. This technique is considered to be the first generation of cloud computing.

- Containerization is considered to be a lightweight version of virtualization, which virtualizes the operating system instead of hardware. Without the hypervisor, the containers enjoy faster resource provisioning. All the resources (including code, dependencies) that are needed to run the application or microservice are packaged together, so that the applications can run anywhere.

Question: how much performance differences have you observed in production between virtualization, containerization, and bare-metal?

Image Source: https://lnkd.in/gaPYcGTz

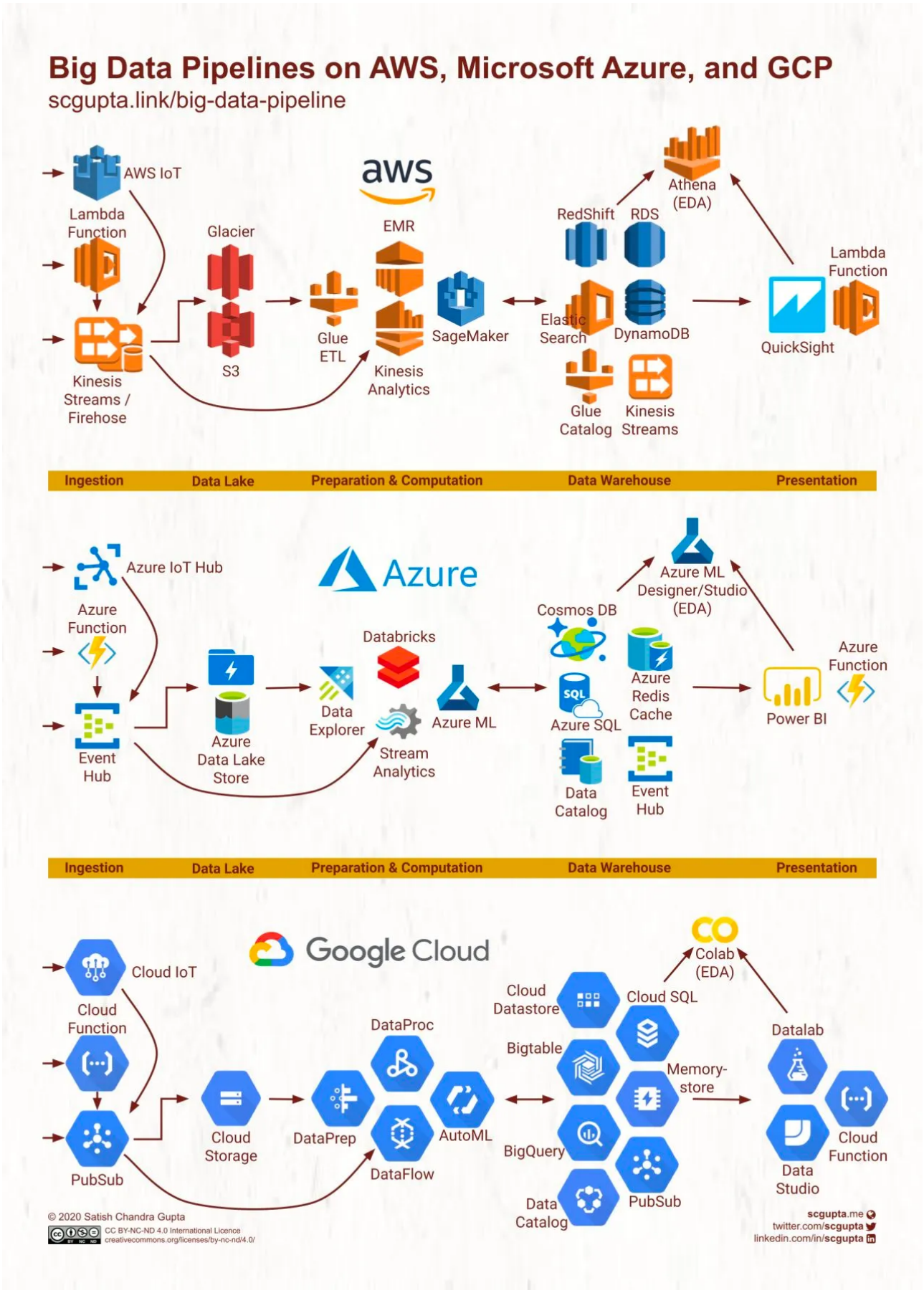
Sources:

[1] Understanding virtualization: https://lnkd.in/gtQY9gkx

[2] What is containerization?: https://lnkd.in/gm4Qv_x2

Which cloud provider should be used when building a big data solution?

The diagram below illustrates the detailed comparison of AWS, Google Cloud, and Microsoft Azure.



The common parts of the solutions:

1. Data ingestion of structured or unstructured data.

2. Raw data storage.

3. Data processing, including filtering, transformation, normalization, etc.

4. Data warehouse, including key-value storage, relational database, OLAP database, etc.

5. Presentation layer with dashboards and real-time notifications.

It is interesting to see different cloud vendors have different names for the same type of products.

For example, the first step and the last step both use the serverless product. The product is called “lambda” in AWS, and “function” in Azure and Google Cloud.

Question - which products have you used in production? What kind of application did you use it for?

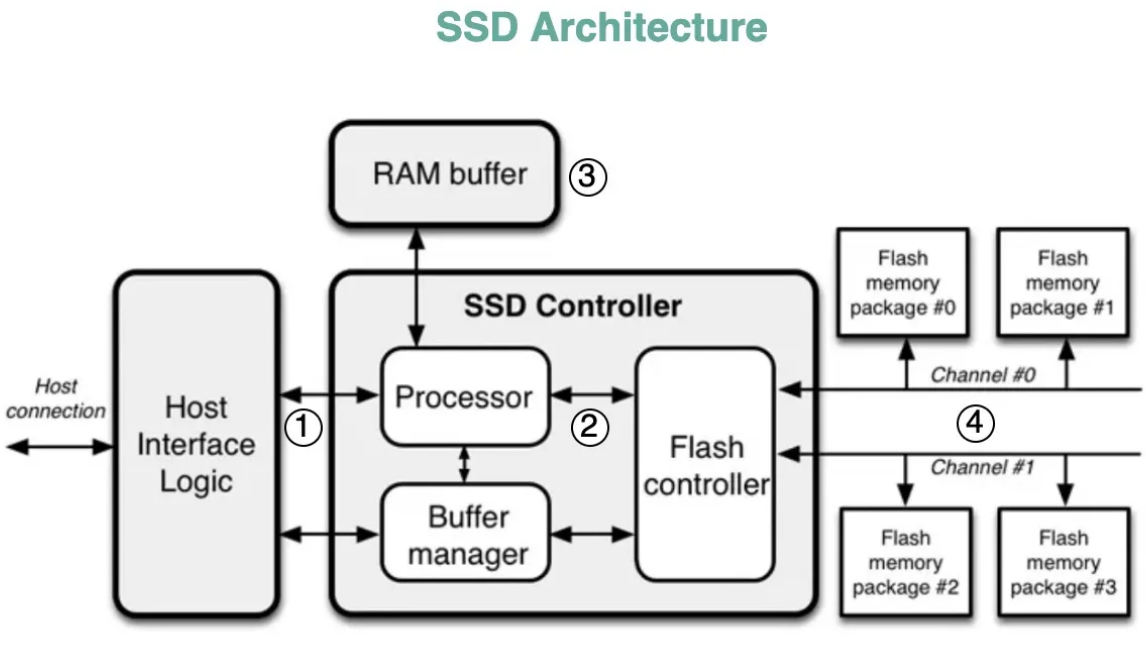
Why is a solid-state drive (SSD) fast?

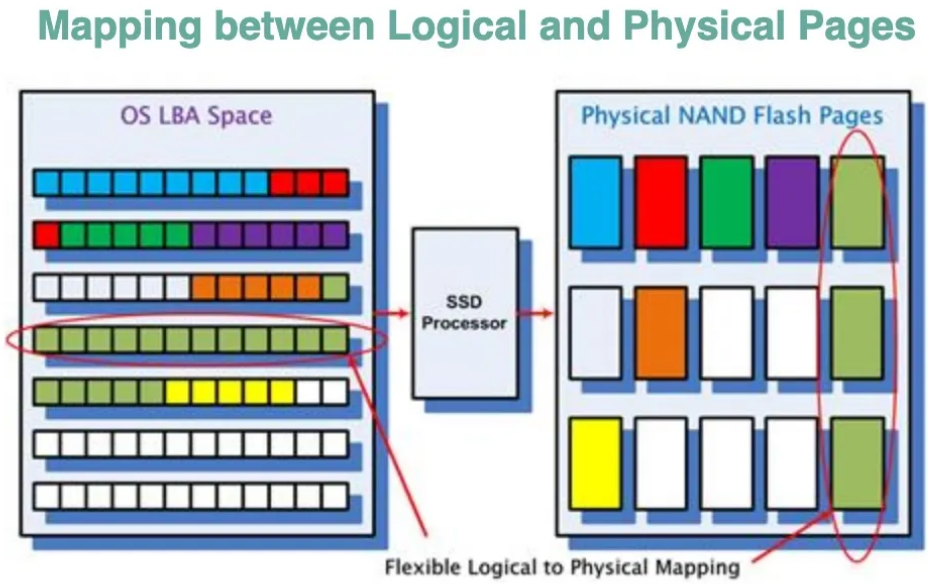
“A solid state drive reads up to 10 times faster and writes up to 20 times faster than a hard disk drive.” [1].

“An SSD is a flash-memory based data storage device. Bits are stored into cells, which are made of floating-gate transistors. SSDs are made entirely of electronic components, there are no moving or mechanical parts like in hard drives (HDD)” [2].

The diagram below illustrates the SSD architecture.

Why is SSD(Solid State Drive) Fast?





Step 1: “Commands come from the user through the host interface” [2]. The interface can be Serial ATA (SATA) or PCI Express (PCIe).

Step 2: “The processor in the SSD controller takes the commands and passes them to the flash controller” [2].

Step 3: “SSDs also have embedded RAM memory, generally for caching purposes and to store mapping information” [2].

Step 4: “The packages of NAND flash memory are organized in gangs, over multiple channels” [2].

The second diagram illustrates how the logical and physical pages are mapped, and why this architecture is fast.

SSD controller operates multiple FLASH particles in parallel, greatly improving the underlying bandwidth. When we need to write more than one page, the SSD controller can write them in parallel [3], whereas the HDD has a single head and it can only read from one head at a time.

Every time a HOST Page is written, the SSD controller finds a Physical Page to write the data and this mapping is recorded. With this mapping, the next time HOST reads a HOST Page, the SSD knows where to read the data from FLASH [3].

Sources:

[1] SSD or HDD: Which Is Right for You?: https://www.avg.com/en/signal/ssd-hdd-which-is-best

[2] Coding for SSDs: https://codecapsule.com/2014/02/12/coding-for-ssds-part-1-introduction-and-table-of-contents/

[3] Overview of SSD Structure and Basic Working Principle: https://www.elinfor.com/knowledge/overview-of-ssd-structure-and-basic-working-principle1-p-11203

Comments