

Dive Into Databases

SYSTEM DESIGN DATABASES



Talha Rizwan
@Toosterr



Diving into Databases

One of the main and the most important part of a system is the database. And every part of the software is dependent on it. could be writing on database or reading from it. But with billions of users performing operations on your system could be chaotic. Slowing your already slow system a lot slower (Both read and write). So, the system designer should design such system which can uphold petabytes of data.



Reads Vs Writes

Study says that most of the systems with billion of user base make 80% reads and 20% writes surprising isn't it? Yes, the data reads are way more frequent than writes. So, our primary goal is handling such amount of reads without ignoring writes also.

This discussion led us to the point where we have to make decision of where to store the data?



Hard Drives as Databases in Depth:

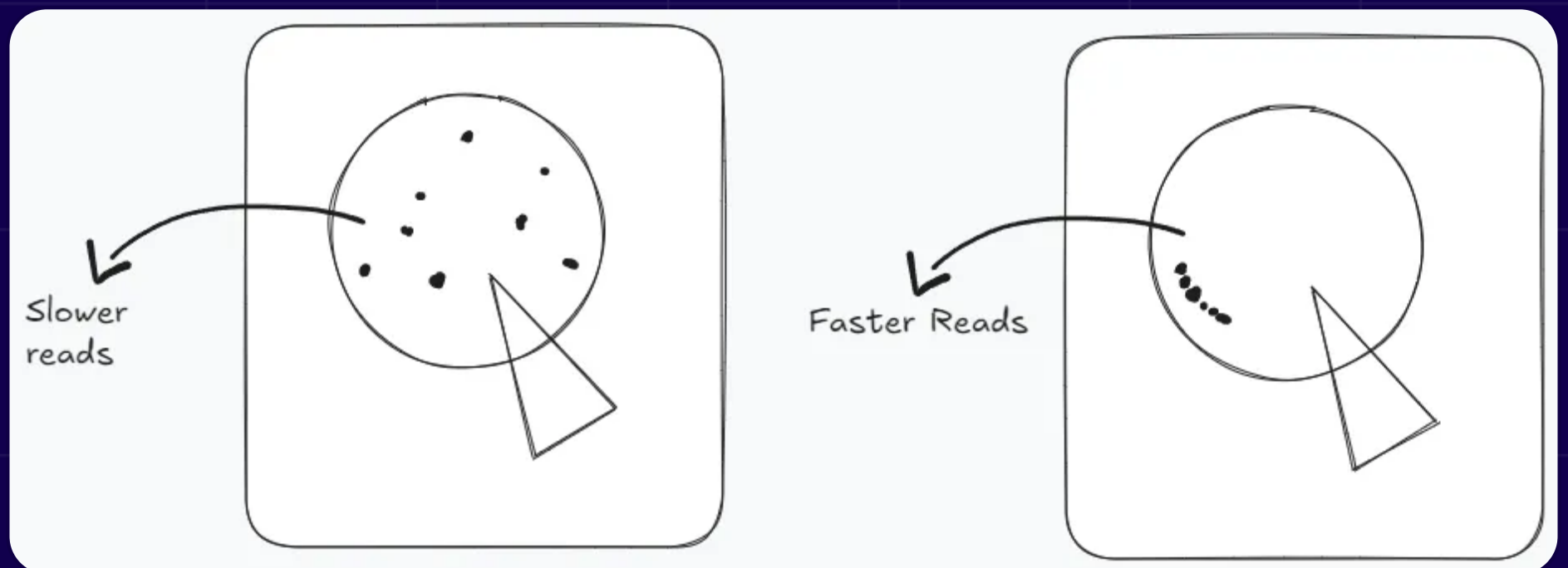
Hard drive stores data on metal disks and is divided into parts which can be accessed by the addresses. It also consists of a read/write head (A Pointer) which moves on the disk and reads data from it.



Reading From Hard drive

Data on hard drive could be at different addresses so while reading it, the pointer needs to move frequently to get the desired information which could be expensive.

- So, we need our data to be closer so that our pointer does find the data fast. e.g. storing data in array.



Now the next part is finding data. Let's take an example where we have people and their ages

Person 1	20
Person 2	24
Person 3	30



1	Person 1	20
2	Person 2	24
3	Person 3	30

Found

Time Complexity: $O(n)$

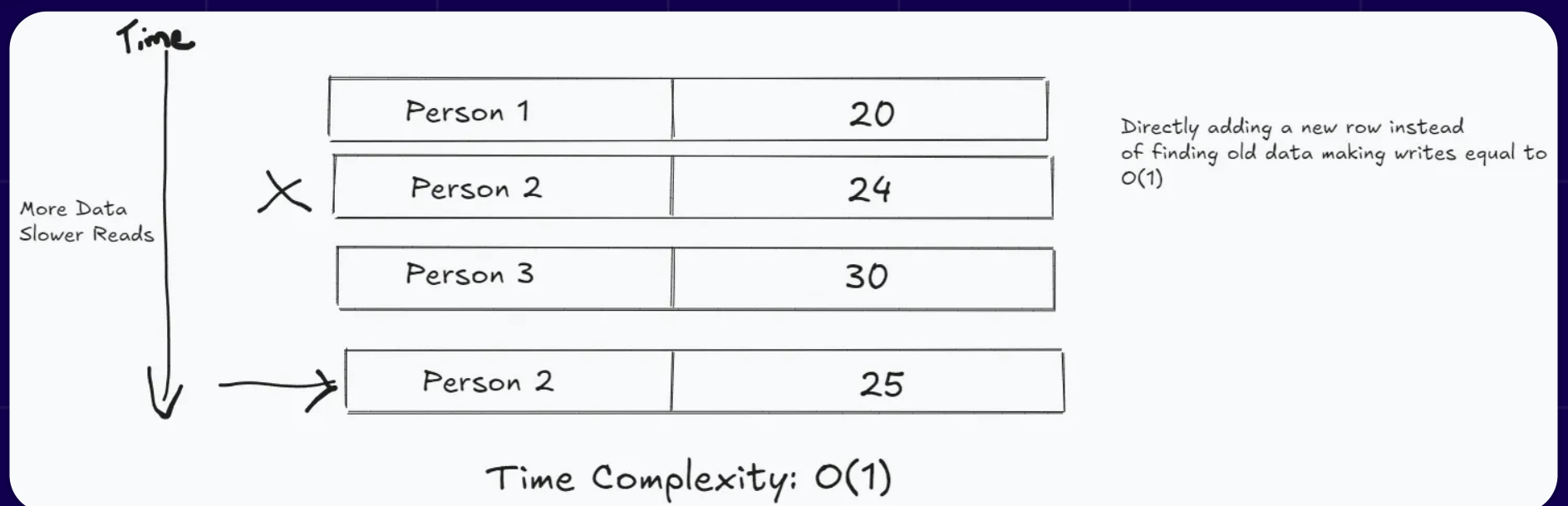
If we want to find the age of person 3, we actually don't know the address of the data where it resides but rather it will go over all the rows and find the desired data.

And in case we want to write the data on some row we still have to find that row and change the data making the time complexity of it equals to $O(n)$ also.



Making Write Faster

If we store new data every time, we want to write like appending new data, it will give us $O(1)$ write speed, but we are compromising on storage and the read speed. Explained in below figure.



Need of Indexes

This is where we need indexes. We will be diving deep into Indexes in next learning. Will discuss the following Indexes:

- Hash Index
- B-Tree Index
- LSM and SS Table Index



Conclusion

Designing a database system that can handle billions of users while maintaining high read and write performance is crucial. The core takeaway is that reads tend to dominate system operations, making up about 80% of the workload, while writes comprise 20%. This imbalance highlights the need for efficient read handling without neglecting write performance.

Hard drives, as the foundation for data storage, present challenges due to their mechanical nature.

Persistency is also a key aspect of database systems. Ensuring data remains intact after system failures or restarts is crucial, and hard drives fulfill this requirement by storing data long-term.



Thanks for
reading



Talha Rizwan
@Toosterr

