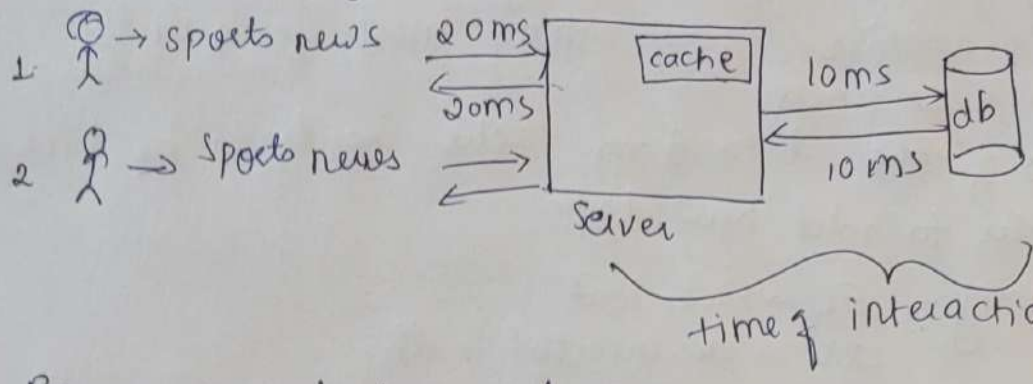
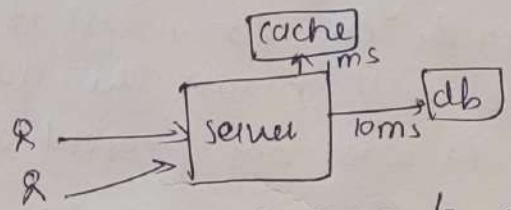
LECTURE 8 - CACHING

Process of storing frequently accessed data in temporary storage location so it can be accessed faster.
cache is a copy of database.



Person1 requests for sports news → 20ms
req goes to db, queries sql & gives results → 20ms (going from)
if person2 also asks for same data, then instead of querying same sql again, we can store the 1st result in our cache memory & when some new user (person2) asks for same request, we can directly send data that is stored in cache memory
→ this will reduce the time of interaction between server and db.



it is much fast and easy to access the data from cache rather than accessing / querying the req & access the data from db

Que Then why cannot we put our entire db in cache
Ans if db is small then its possible. But if db is large it is not possible cause our cache size is already small and limited
So, our cache would only store the frequently requested query results.

CACHE POLICY

least recent used → LRU
least frequently used → LFU

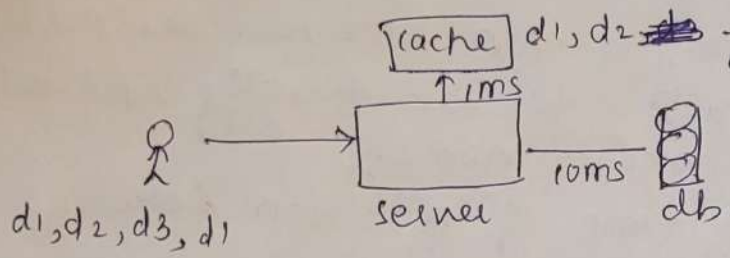
Suppose our cache is already filled with most frequently requested query.

→ Now there is another request coming most frequently ie movie release news
Then some of the data from cache needs to be kicked out in order to add new data.

by using - LFU → kicked out
LRU → inserted in db.

DRAWBACKS

(1) Trashing → when system spends more time on non productive task rather than on productive task



eg → cache is only a rec to store eg.

eg cache is already fulled with d1 & d2 data now user requests for d3, first it will check on cache then it will procss the query & give result (as d3 was not present in cache it took extra 1 ms) (E this was waste of time thrashy

→ insertion & removing & then inserting the prev data is time consumig

(2) Eventual consistency

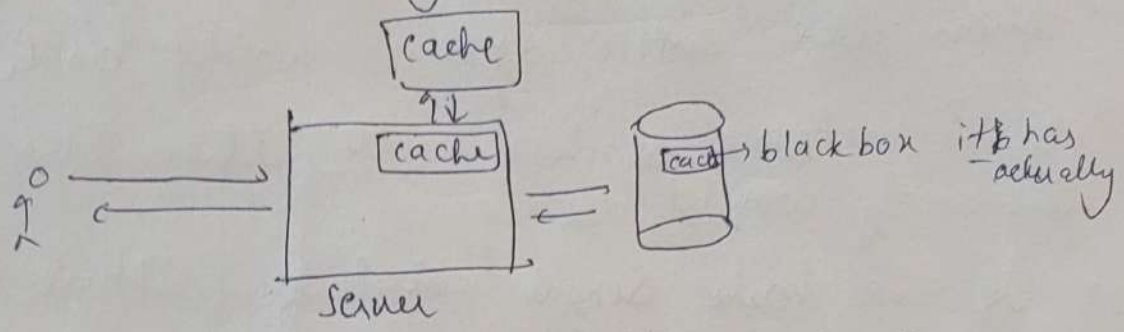
eg cache is update every one hour, our db & cache data are same at this pt eg (0 likes on yt)
→ now there is update in data within 5 min eg no of youtube likes got increased in 5 min
→ Now our db is having actual likes & cache is having the old data (ie data is not updated)

here we can see false data and true data is only seen after one hour.

in case of transaction user dont wants to see false data

→ This cache update is dependent on the policy that we use.

Where is cache memory stored.



ideally ~~cache~~ cache is present Globally, inside db, inside server

↓

independent in memory cache

ideal cache to choose is Global cache

→ as it can scale independently (kabhi bhi kitna bhi bada kar sakte hai)

→ if we change algo of cache, then there is no need to redeploy the code on servers. ie deployment is independent

→ Multiple servers can use the same cache (using same logic & executing query) that was executed earlier by some other server.

Eg brother asks 15 square one morning → it will take 15 sec

if he asks same in evening we could be able to response quickly