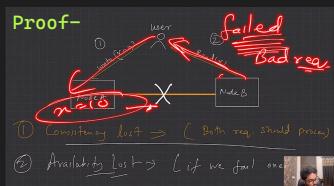


Lecture 20

CAP theorem

C → Consistency (data in each node must be consistent even wrote only one node)
A → Availability (system must be available for requests)
P → Partition tolerance (partition is break in communication bw nodes if it done but system should not fail, is partition tolerance)

CAP theorem → In any distributed system all 3 properties cannot be applicable simultaneously, only one or two at most can be at a time. (only distributed system cause single node will never have partitioning)



CAP theorem (No SQL db) →

- i) CA databases → Consistency and availability, it is not possible practically in distributed system because of partitioning (only possible in not distributed system)
- ii) CP databases → Consistency and partitioning, it is practically possible above proof image is example, and MongoDB was the example for this. It has only one primary node and all nodes are replicas.
Example → Banking system, it requires consistency more than availability.
- iii) AP databases → Availability and partitioning, it is possible refer above image, "these databases are eventually consistent" when partition even all nodes are available but not consistent(updated)
Apache Cassandra is an ap database.

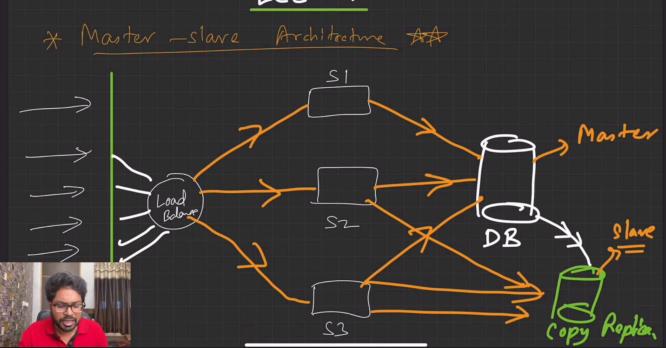
Example → Facebook, Availability >>> consistency

HW → ACID vs BASE properties

Lecture 21

Master Slave Architecture

Lec-21



Master db(node) will handle all write operations cause it is original or primary or latest db
Slaves db(replica nodes) → read operation and replicate from master db

It is completely lec 19 pattern 3

How replication happen?

- i) Asynchronous → slaves replicate at some time T from the Master db
ex. we saw cab booking eg in lec 19 now worry of late data updation of book history
- ii) Synchronous → if we update master db then at same time we replicate slaves db when they return success to master db then will show succeed to user.
ex. banking, it needs real time updation

Q. If slave got update query?

- Never allow slave can have write query.
- If allowed then need to handle it like master db, and it will be Master master architecture
- Ex. pattern 4 lec 19 multi primary pattern

Advantages →

- i) Backup
- ii) Scale out for read operations
- iii) Availability increases
- iv) Reduce latency
- v) Request distribution (parallelism)

