

Replication And Sharding

Replication

The act of having a **secondary database** (which is a replica of the main primary database) in order to make our systems more **tolerant** to **regional failures** of databases.

The replica database is used only when the main database is down. The **main database updates** the **replica database** in a **synchronous** way (ie data is written in the main database and the replica database at the same time. This leads to longer writes) or an **asynchronous** way (Replica DB gets updated every 5 mins / 1 hour / 1 min ...).

The asynchronous way of updating the replica database is useful when longer writes cannot be afforded. The asynchronous way can also be used when a feature which is deployed to the prod is to be experienced

only by a certain set of users. If the feature works as expected then the data will be copied to the replica databases.

Replication can also be used to (horizontal scaling database) handle extra load, decreasing the latency of accessing data (This is done by replicating the data in a data center which is closer to the clients)

Sharding

Sharding is also called as data partitioning. This is the act of splitting the main DB into two or more sections called as shards and is typically done to increase the throughput of the database.

Popular sharding strategies:

- ① Sharding based on the client's region
- ② Sharding based on the type of the

data being stored (Eg: User data gets stored in one shard and payments data gets stored in another shard)

The main database should be sharded in such a way that hot spots are prevented

Hot Spot

When distributing a workload across a set of servers, that workload might be spread unevenly. This can happen if the sharding key of the hashing function are sub optimal, or if the workload is naturally skewed: some servers will receive lot more traffic than others thus creating a hot spot

Eg:



