https://www.youtube.com/watch?app=desktop&v=iuqZvajTOyA&ab_channel=SystemDesignInterview

Probabilistic protocols for data replication

- gossip
- epidemic broadcast trees
- facour eventual consist

Consensus protocols

- paxos
- 2pc

Helps deal with the hot shard replica and increase availability

Consistency

- can introduce consistency through synchronous data replication
- but at the cost of latency and the complexity of the system

Consistent hashing issues

- domino effect
  - when a server dies and all the load goes to the next server, which goes to the next... and so on
- nodes dont split the circle evenly

JumpHash or Professional Hash?

https://www.youtube.com/watch?v=U3RkDLtS7uY&ab_channel=GauravSen

Use different nodes for cache? or use db server memory for cache?

- prefer the former typically
- can scale the cache layer independently
- makes cache fault tolerance independent of db ie if the db node goes down the corresponding shard of the dist cache does not go down with it

Write through vs write-back cache

- write through: edits update cache first if the key exists in the cache, then updates db
- write back: edits update db first, then write the new state to the cache
- feels like this depends on how much you value consistency?
  - critical data eg passwords you have to write to persistent db first
  - but maybe for less critical data you can go through the cache first