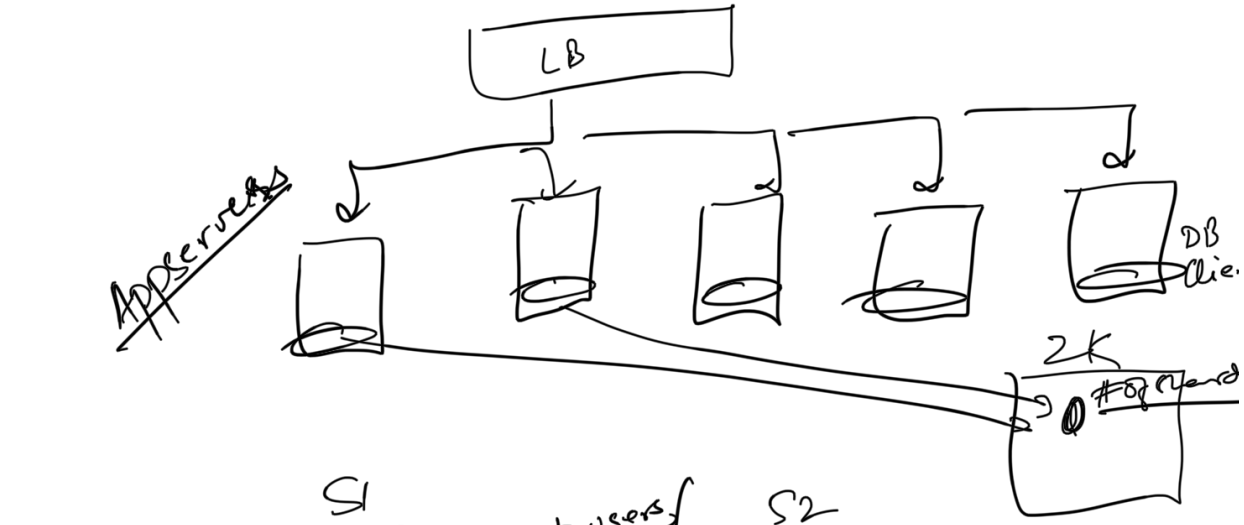
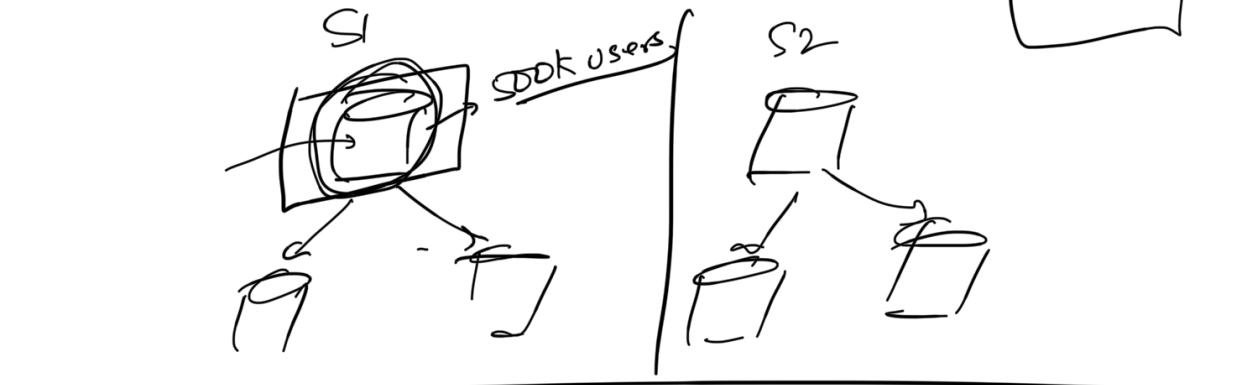
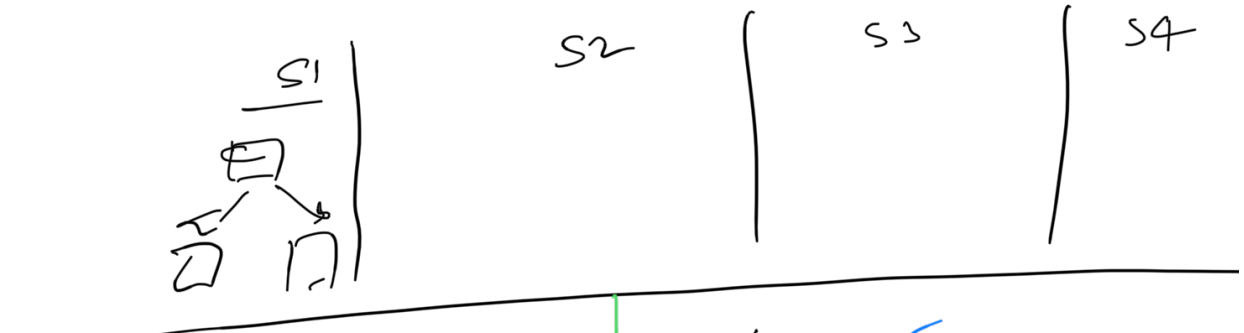
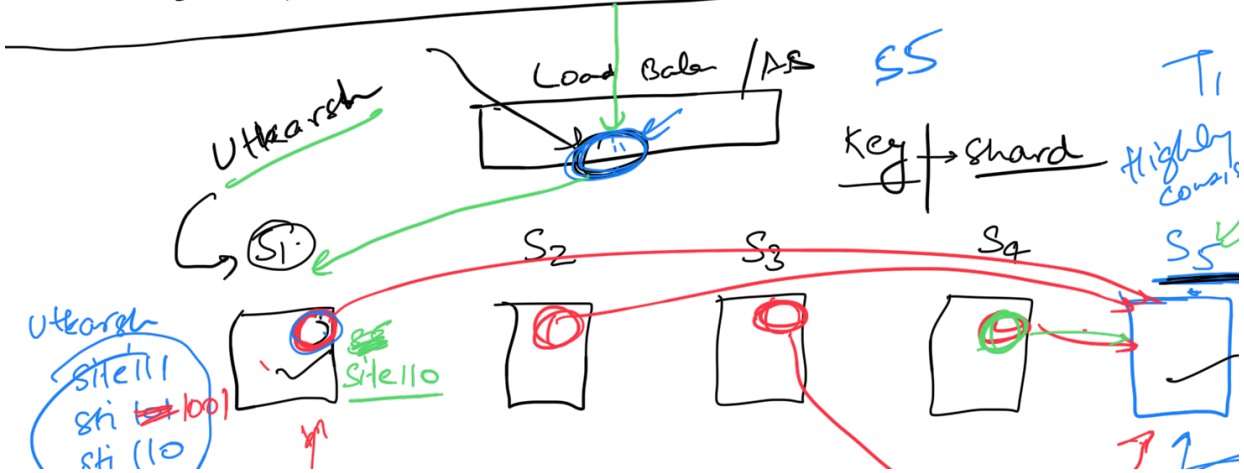
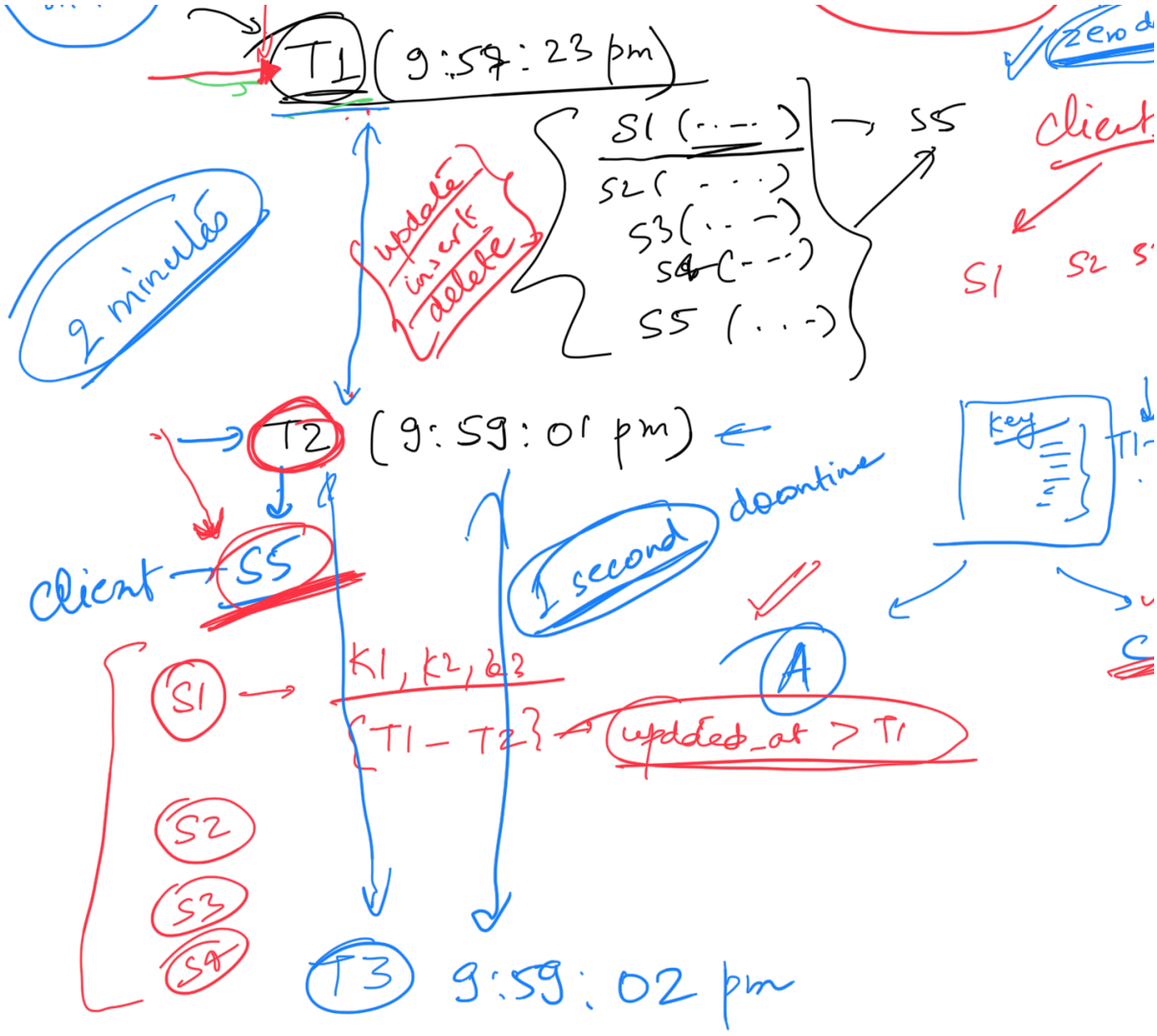
NoSQL


LB
DB
2K


S1
500K users
S2


S1
S2
S3
S4


Load Balancer / AS
SS
T1
Key → Shard
Uttarakh
S1
S2
S3
S4
S5
Sitelll
Sitello



T1 → don't ~~eta~~ tell client about S5.

But start data transfer.

T2 → Data Transfer complete. Tell clie about S5

High A → Inconsistent

T3: Catch up

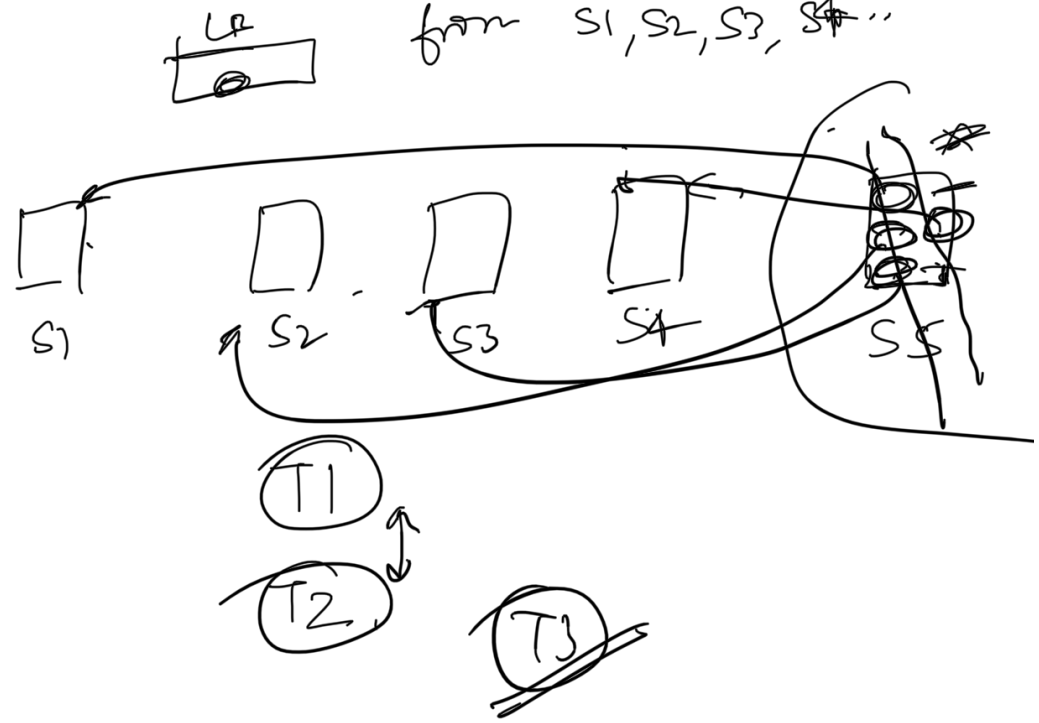
High C

Downtime till

T3: Catch up on updates post T1 a before T2

Downtime. Pro

Post T3, delete S5 keys from S1, S2, S3, S4 ... T2 – T3



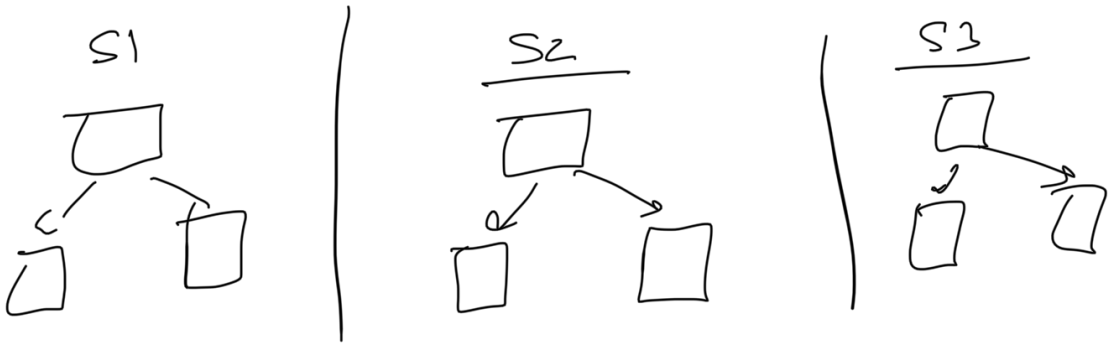
---

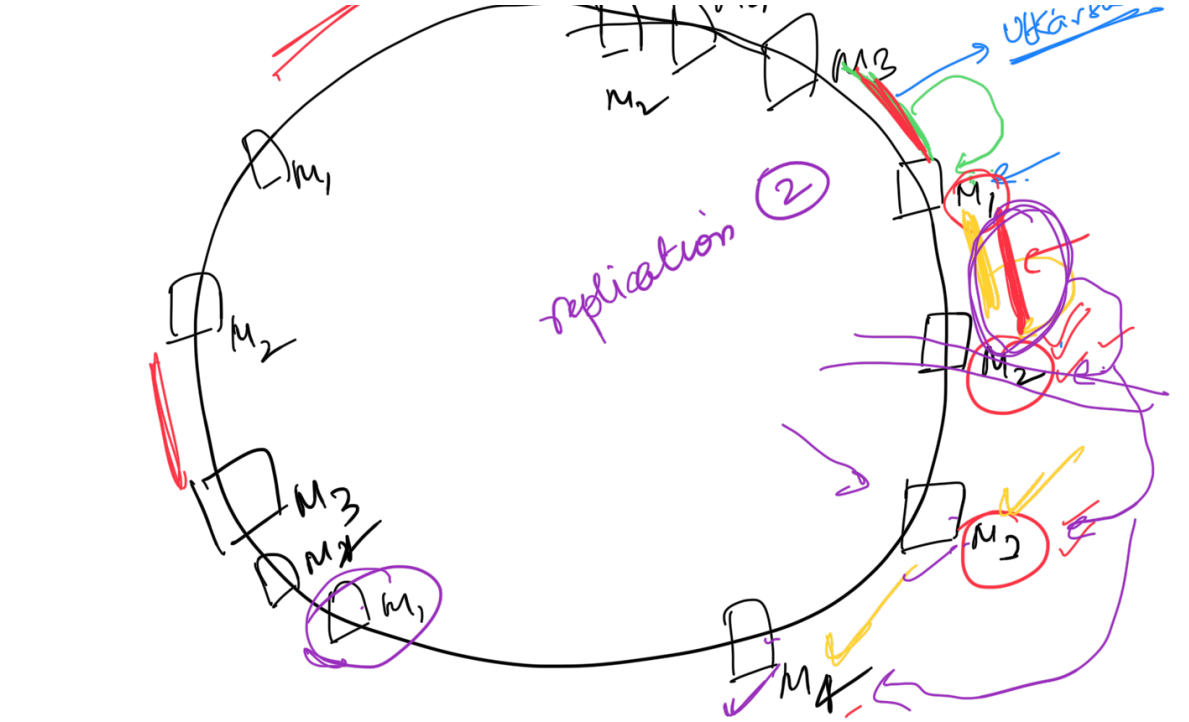
① 1 way of creating a NoSQL DB

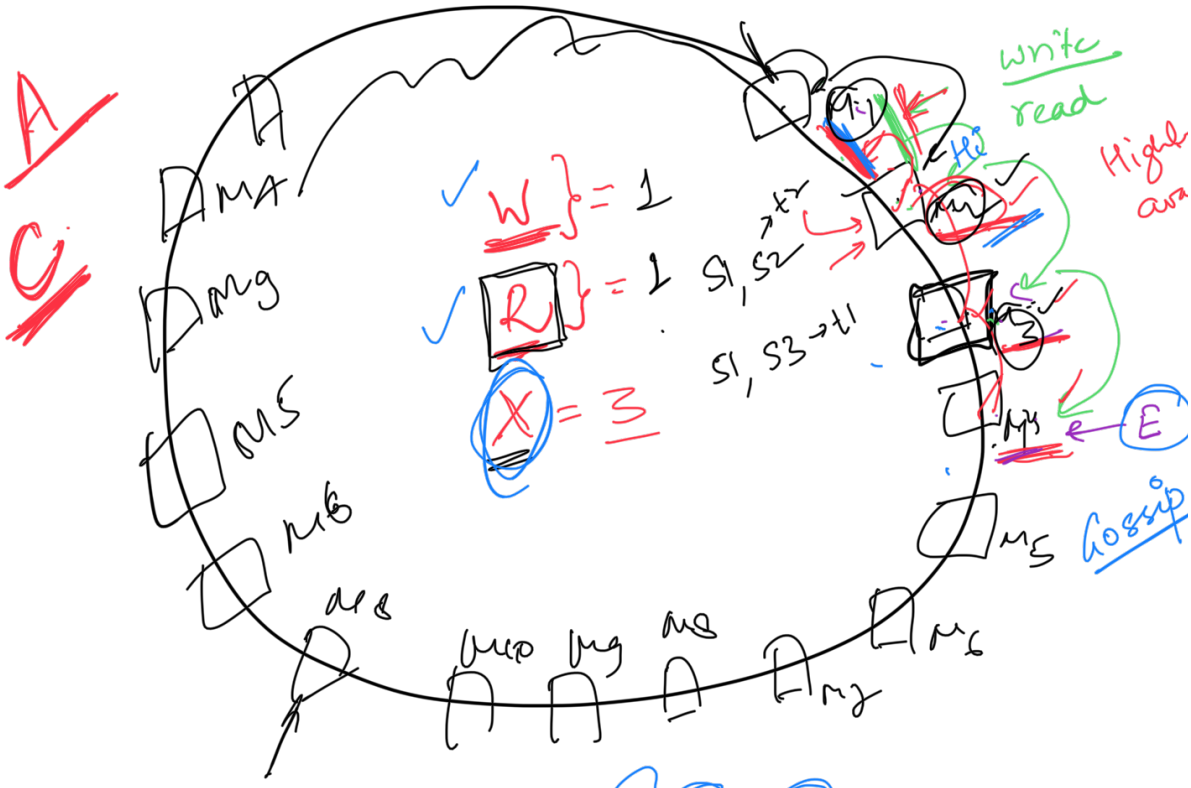
↳ ① Scaling up or down by adding more nodes

② Ensures replication level

③ Sharding

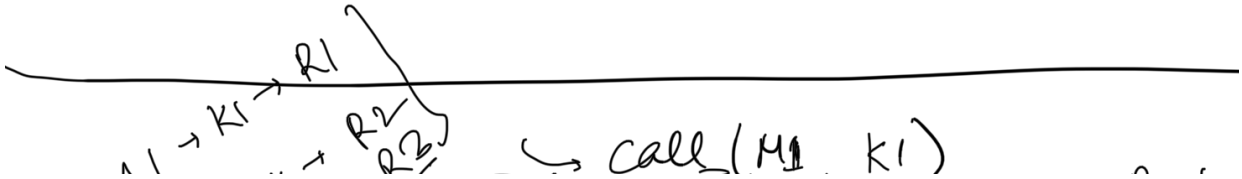


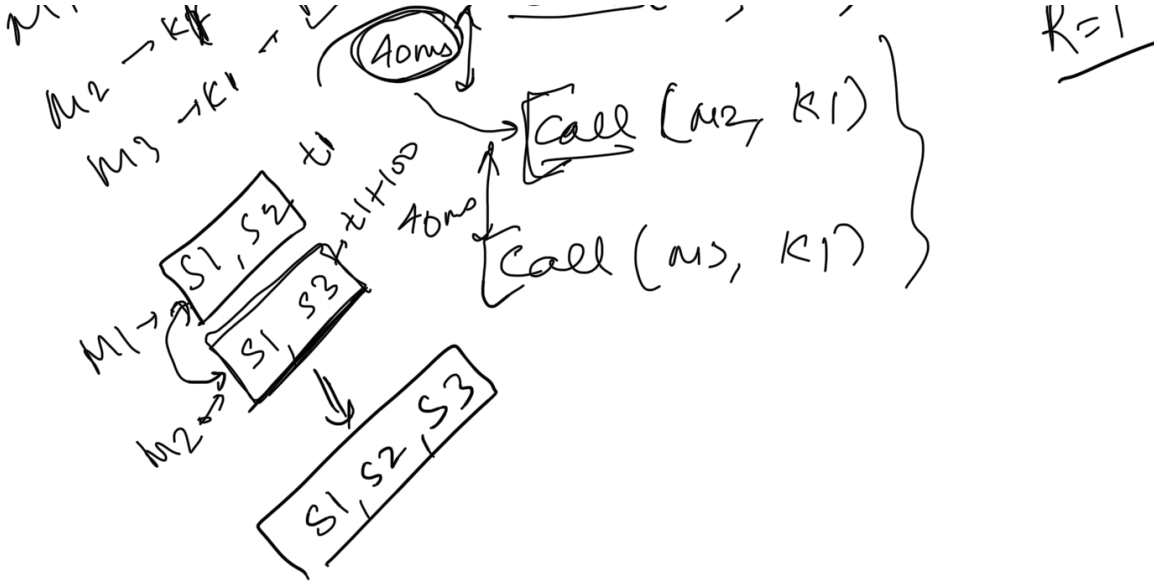




Tunable consistency ←

- R + W > X → Highly C
- R + W ≤ X → Highly A





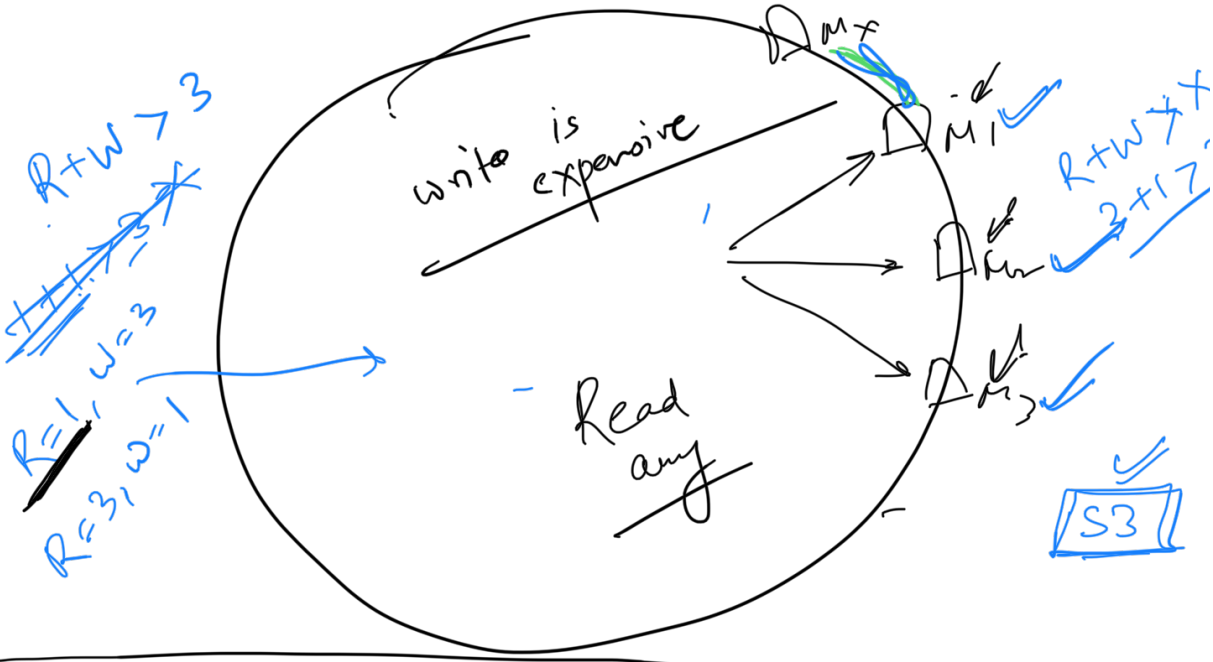


Table
users

id	name	role	updated at	...
		0/1/2		
4 Byte	50	1 byte	8	8

① Add machines
↳ new shard
↳ Automatic data transfer

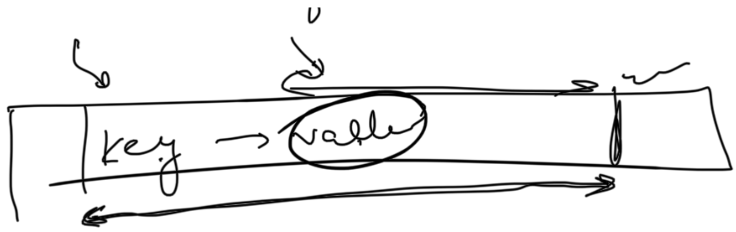
② Replication level is maintained

① non-schema driver

key → value

↳ fixed size

a json



② SQL → B+ trees

write O(log N) → lightweight

read O(log N)

X —loans→ Y

T1 — STEP 1: Check X's bal
X = X − 100

T2 — STEP 2: Y = Y + 100

STEP 3: ON FAILURE, ROLLBAC T1