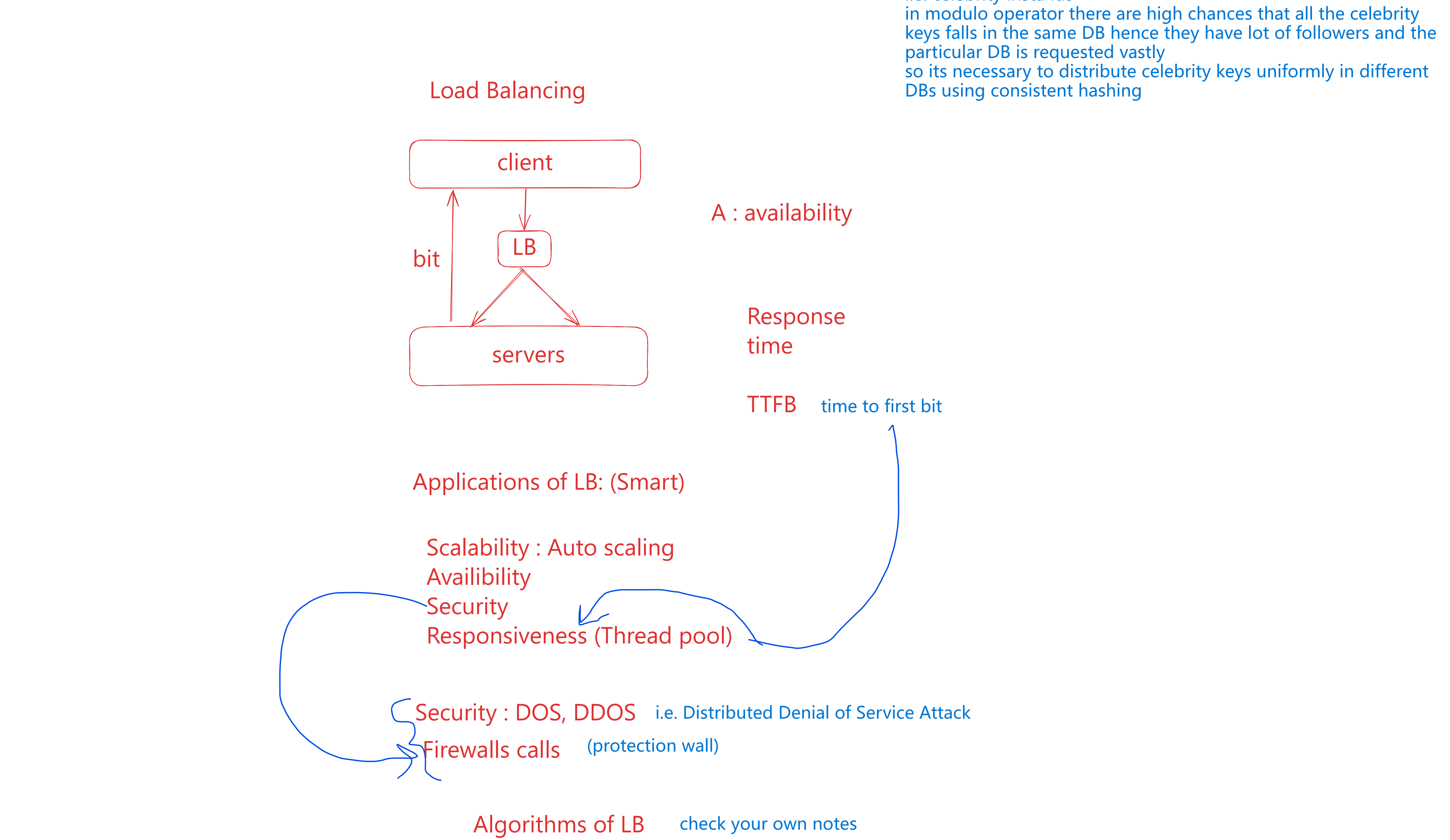
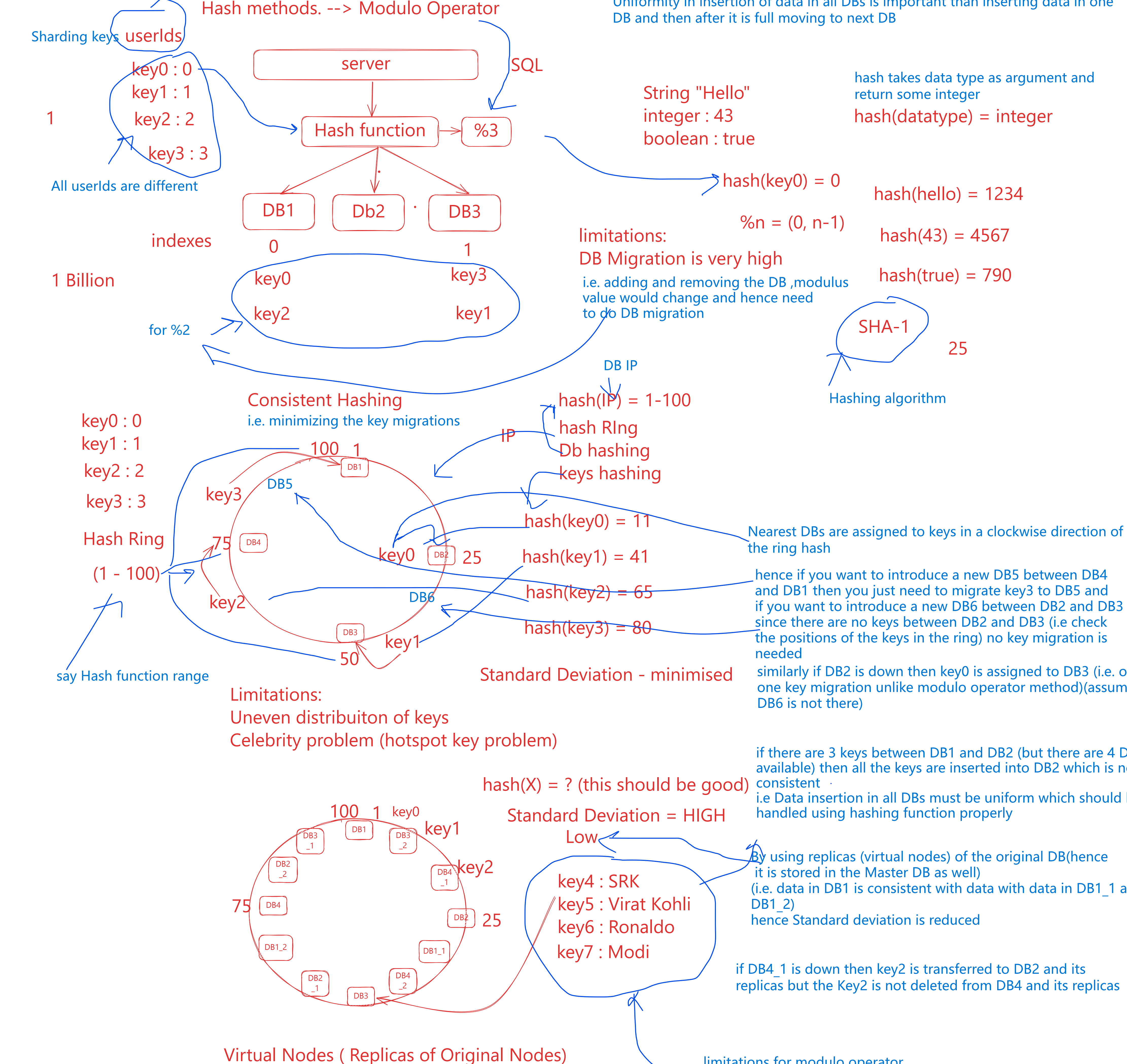
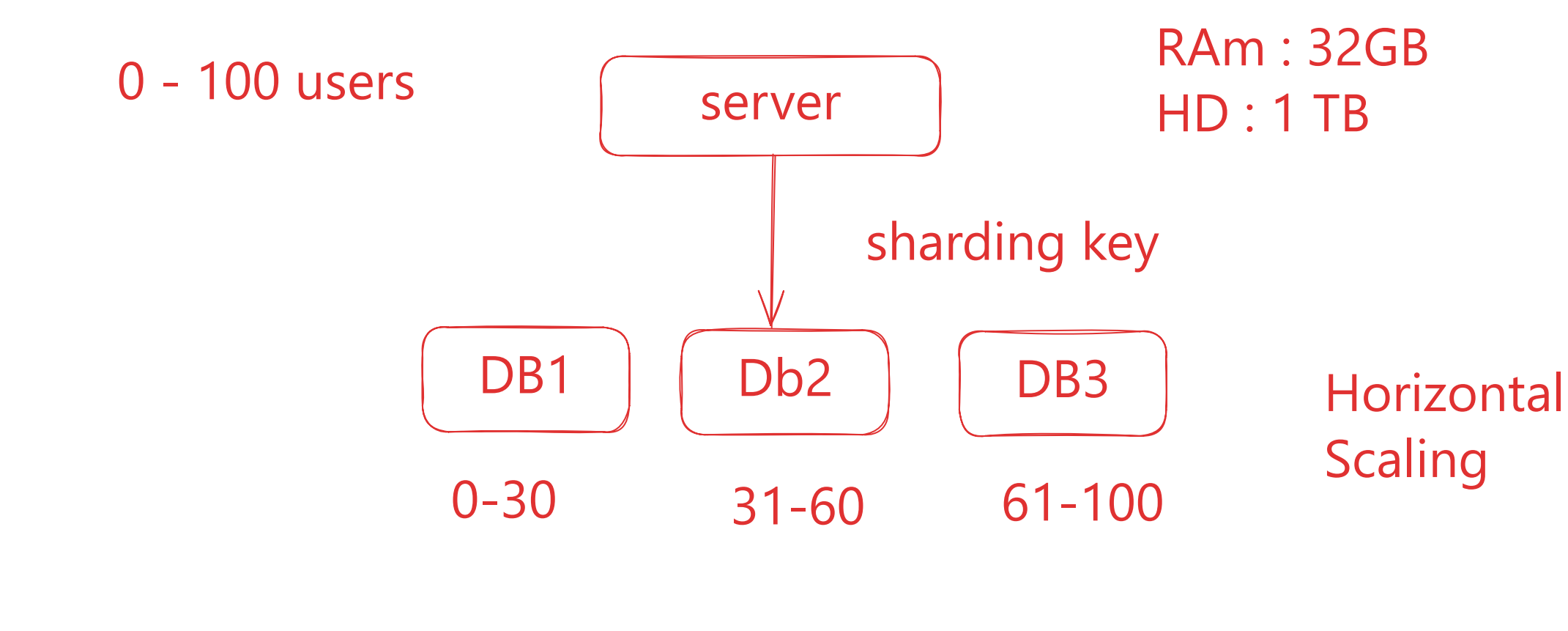
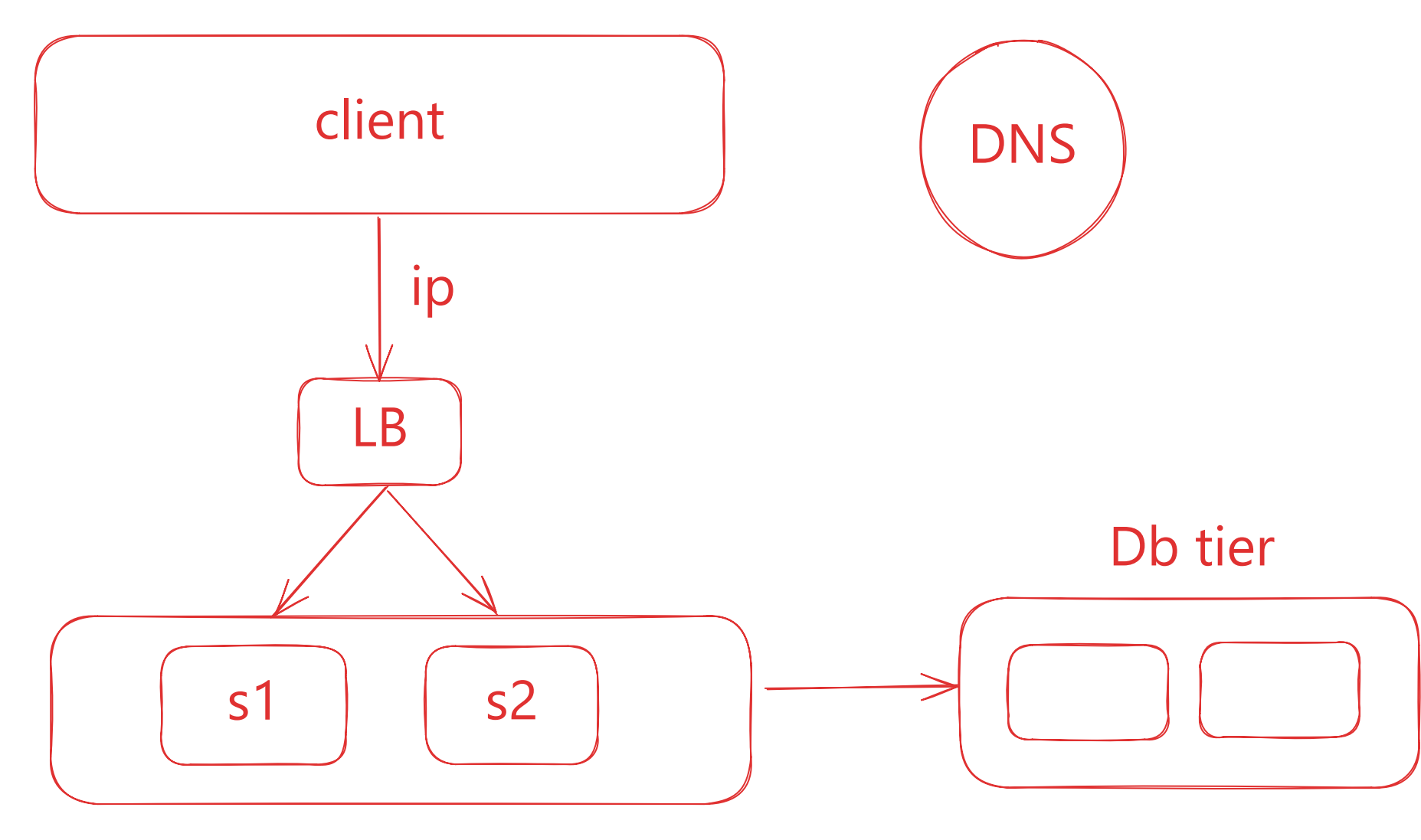


Sharding -- Consistent Hashing

Load balancers



Static LB's Algorithms:

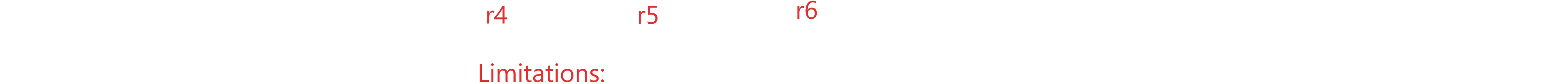
1. Round Robin Algorithm



Limitations:

- Do not consider load on any server

2. Weighted Round Robin Algorithm



wt : RAM, CPU

wt : 32 GB

Limitation

- Does not manage load on server

3. IP Hashing



r1 : IP : hash(X) = ? (MOD)

r2 L IP : hash(X) = ?

Tightly coupled (stateful Architecture)

if I am requesting from my machine since my machine has a same IP for every request the output of the hash function will be same for every request and load hits the same server every time (this is the issue with IP hashing)

Limitation : Each user will always get same server

Dynamic LB Algorithms

1. Least Connection method



connection

HTTP : Stateless

u1 --> s1

2. Weighted Least connection method



server:

- No. of connections
- weight

no. of conn / weight

3. Least Response time method



Response Time

No. of active conn * RT

s1 = 30

s2 = 44

s3 = 75

application - where bulk users hit (ex- Tatkal booking)

4. Resources Method



agents are algorithm

agents inform LB if server is up or down

agents does the health check

capacity/load decision factors

1. no. of active conn

2. response time

3. latency

4. health check

HTTP (empty) /health/check

Health Check 200 (ok)

agents:

- To check if server is up
- At a particular moment what is the capacity or load in that server

Load Balancer (BASED ON LAYERS)

- Application level LB (HTTP/HTTPS)
- Network layer LB (Ip addresses)
- Global server Load balancing (data centers) (Geo,DNS)
- DNS load balancer



Types of LB



LB is also a computer/server

for very big applications but not scalable

Data center

GBs of request