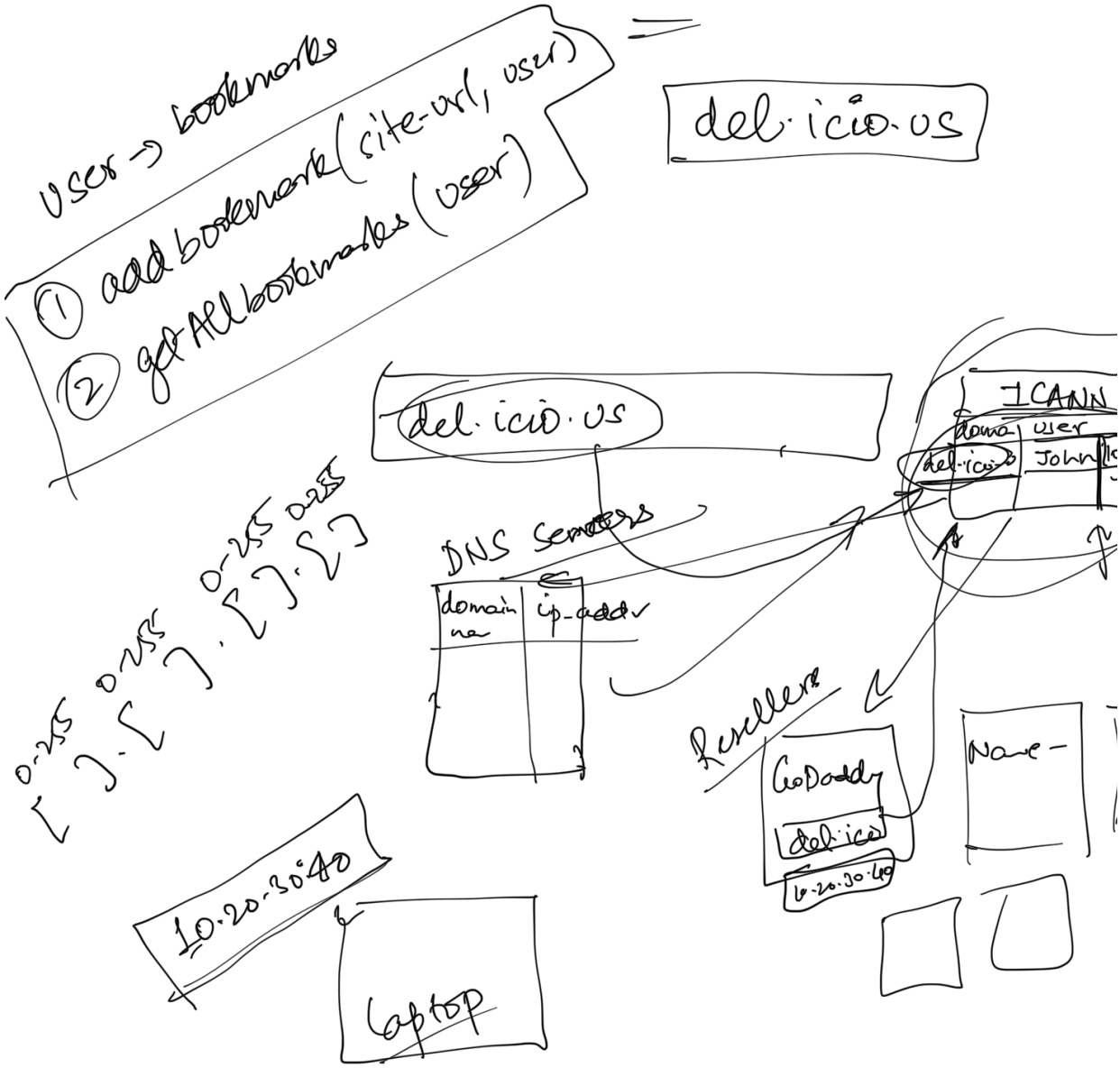
SYSTEM DESIGN - 101



① Who does DNS server belong to?

② How does browser know?

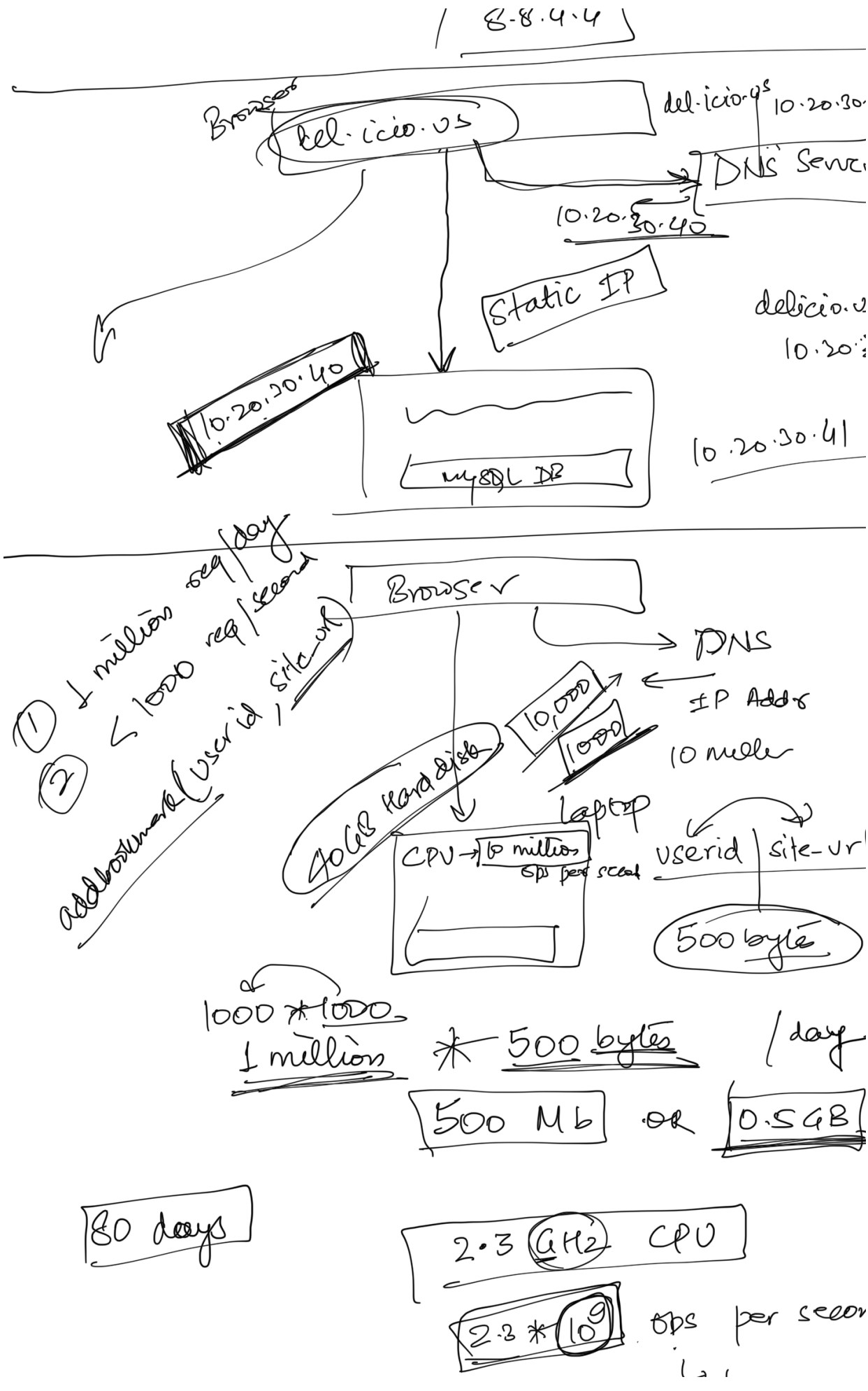
↳ DNS lookup

① ISP

② Google / FB

↳ 8.8.8.8

Google's DNS server

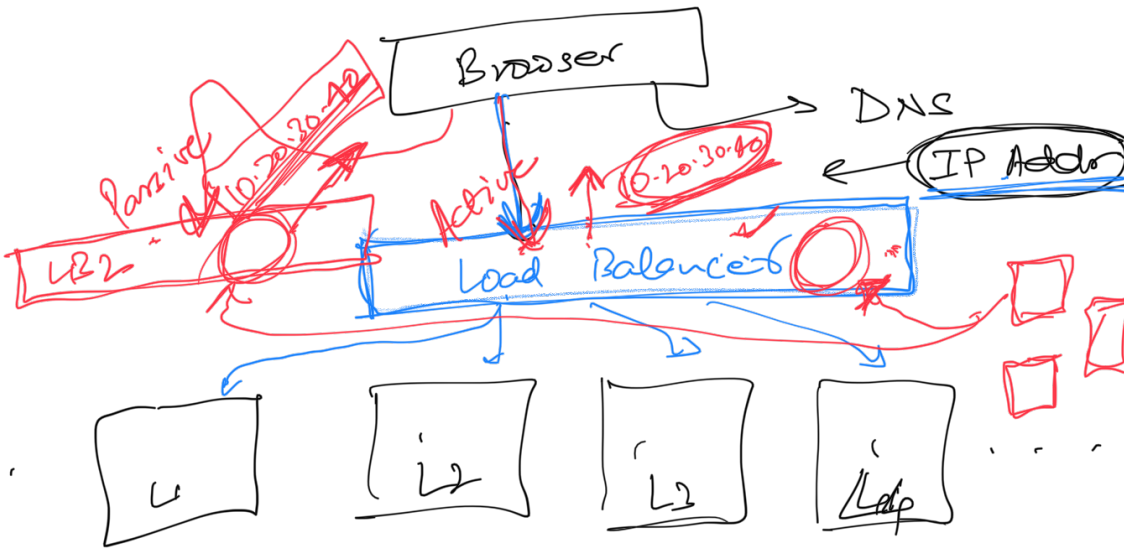

8.8.4.4
Browser
del.icio.us
10.20.30.40
Static IP
10.20.30.41
MySQL DB
1 million req/day
< 1000 req/second
addbookmark(userid, site_url)
Browser
DNS
IP Addr
10,000
1000
10 mille
40GB Hard disk
CPU → 10 million
Laptop
userid | site-url
500 bytes
1000 * 1000
1 million * 500 bytes / day
500 MB or 0.5GB
30 days
2.3 GHz CPU
2.3 * 10^9

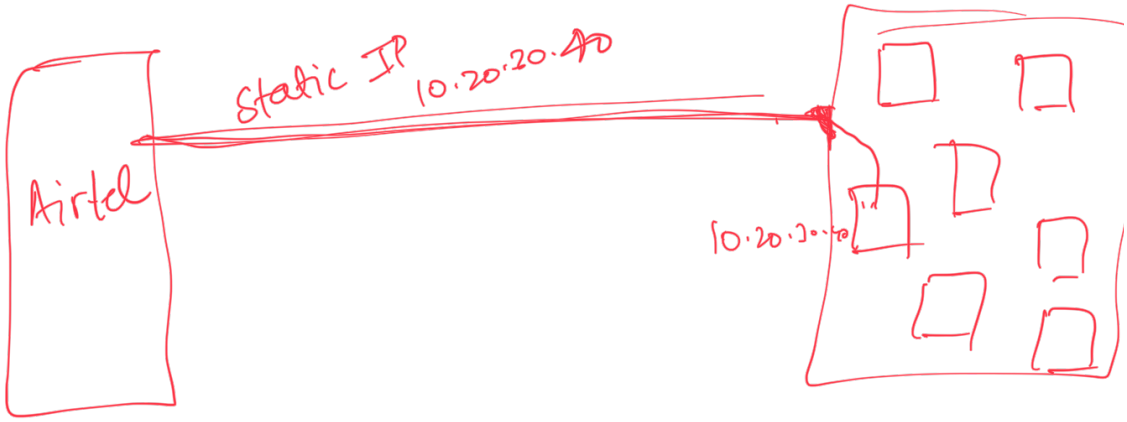
(1) Buy a better laptop

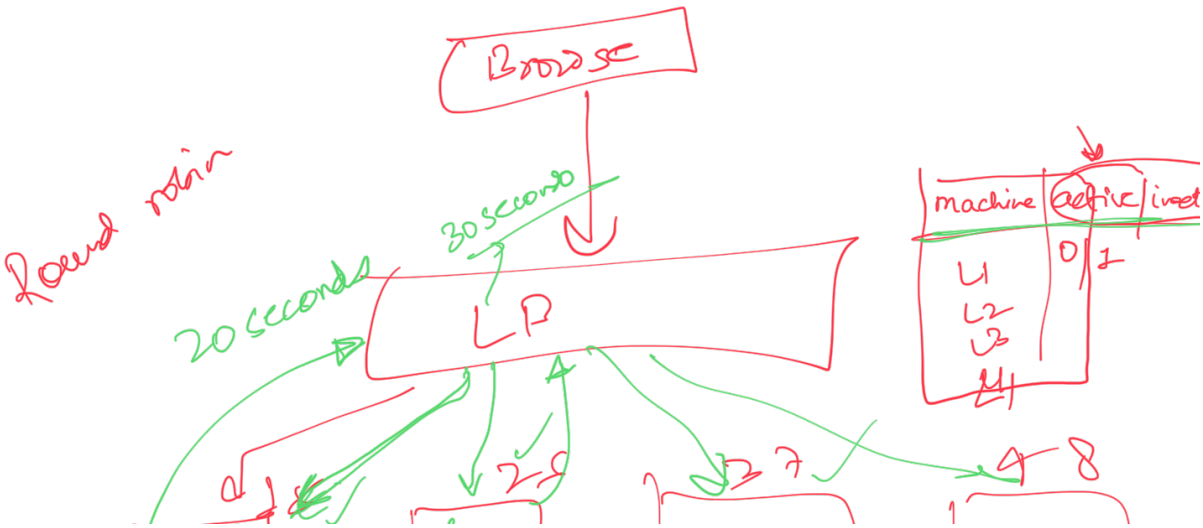
Vertical scaling

80 GB

(2) Buy more laptops



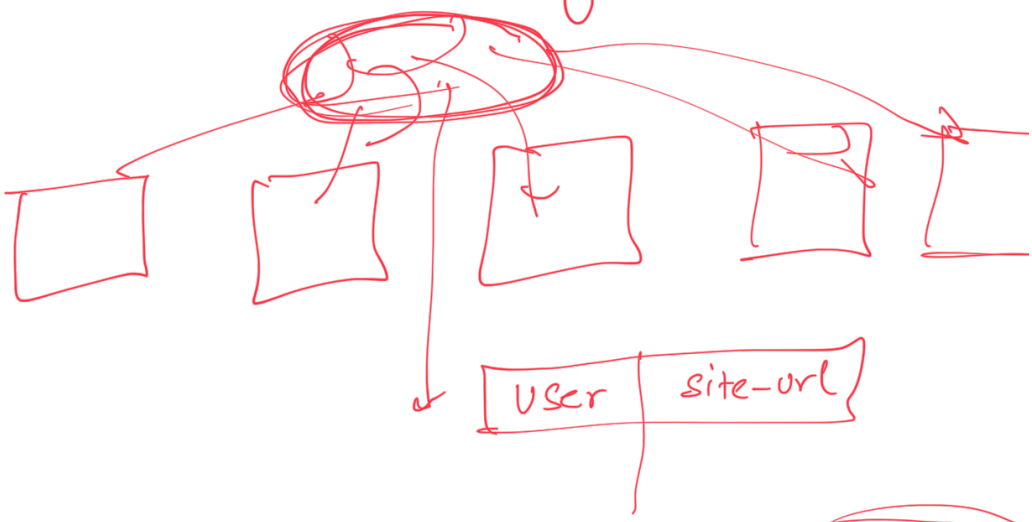


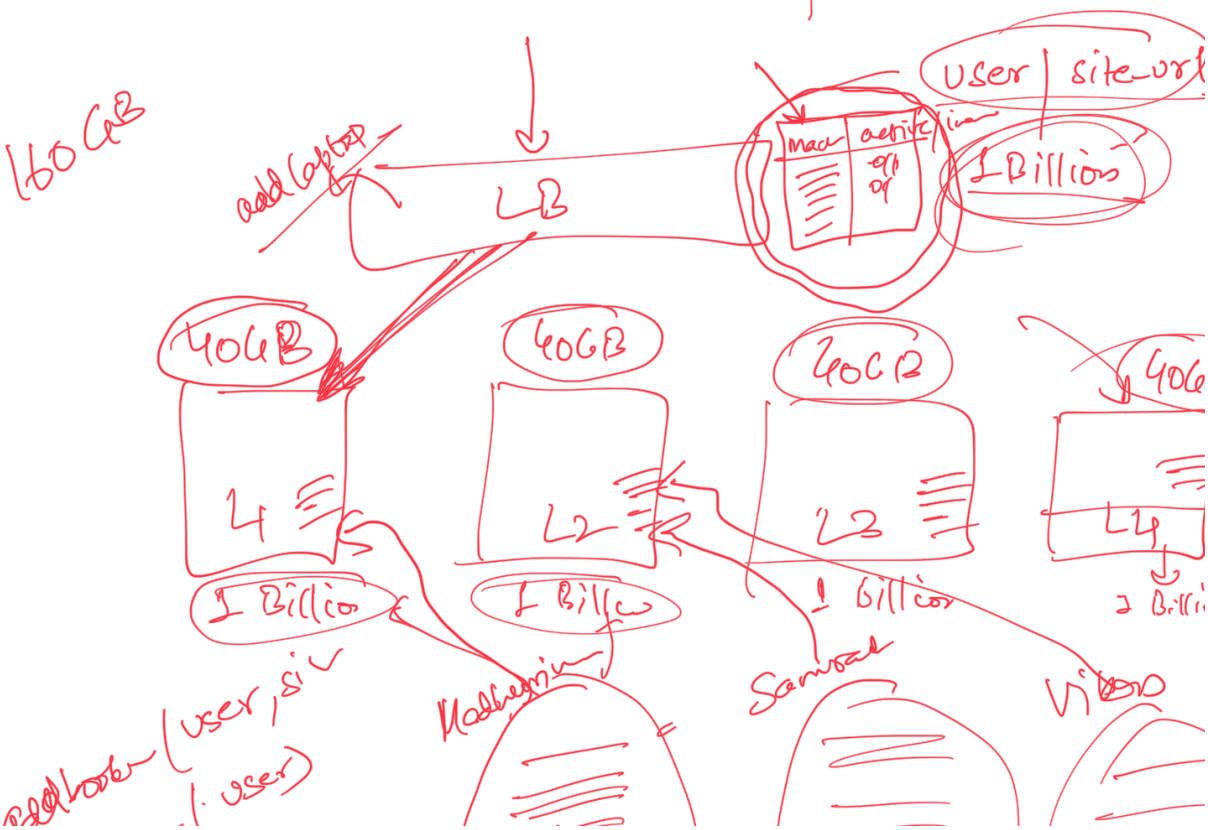


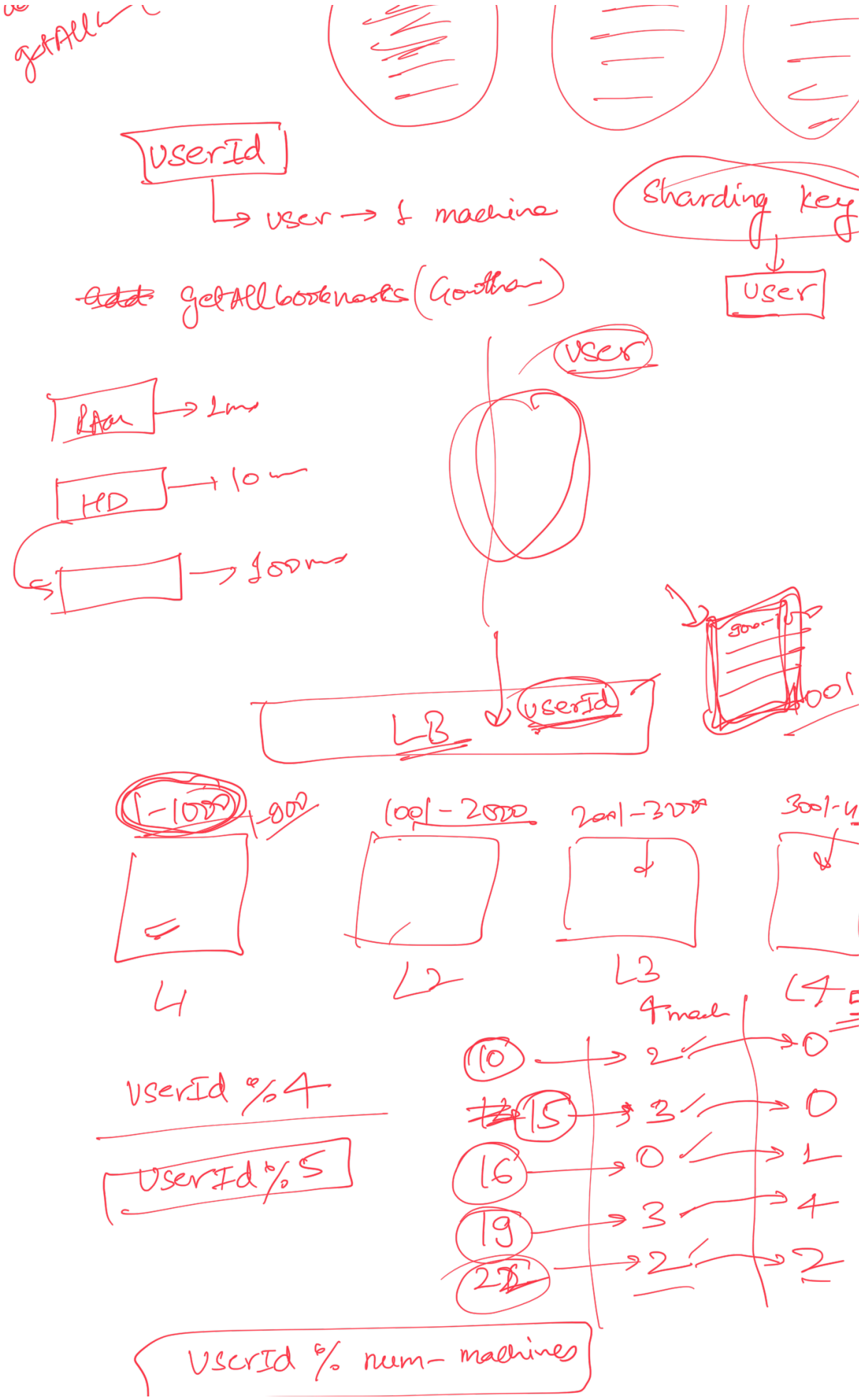
- Health check (LB → machines)
- Heartbeat

data Sharding

① LB knows which data is wh



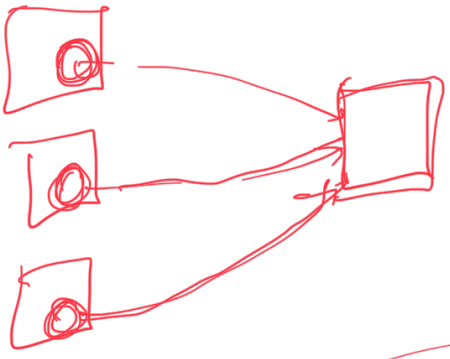



getAll
UserId
→ user → & machine
Sharding key
User
getAllBookmarks(Godthan)
RAM → 2ms
HD → 10ms
→ 100ms
User
LB
UserId
1-1000
1001-2000
2001-3000
3001-4
L1
L2
L3
L4
4 machines
UserId %4
UserId %5
10
15
16
19
20
UserId % num-machines

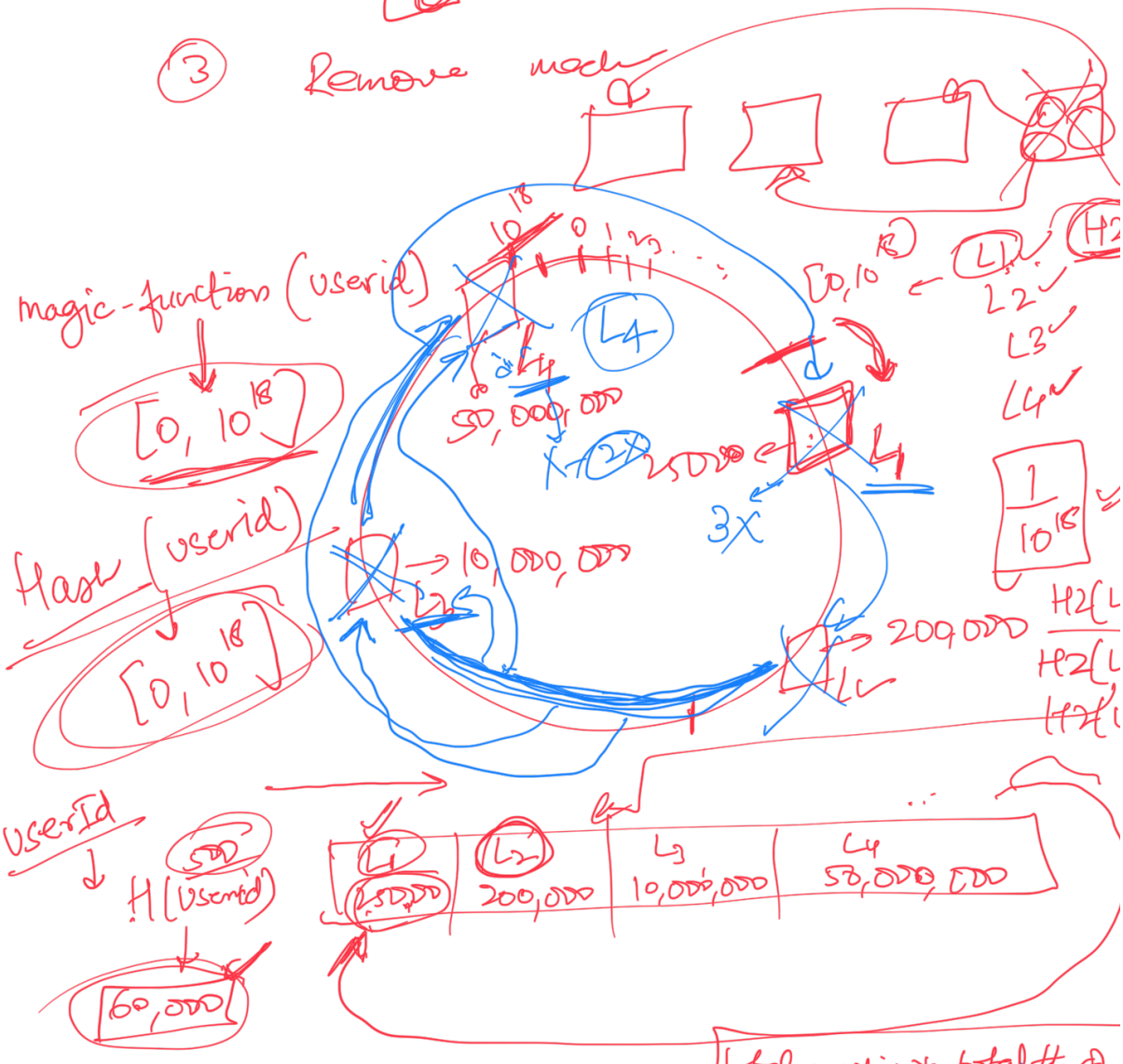
1

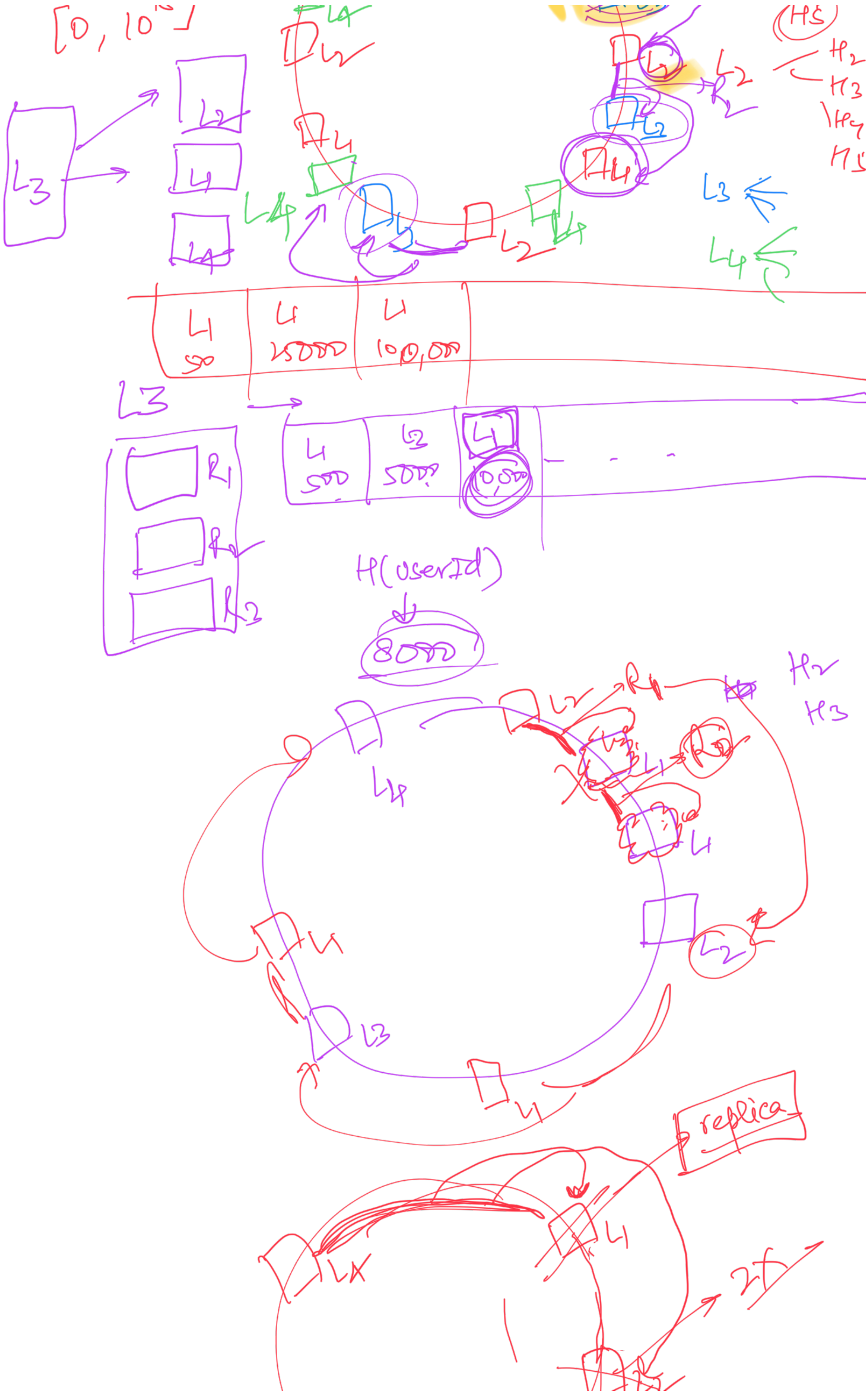
① LB should find user → machine quickly

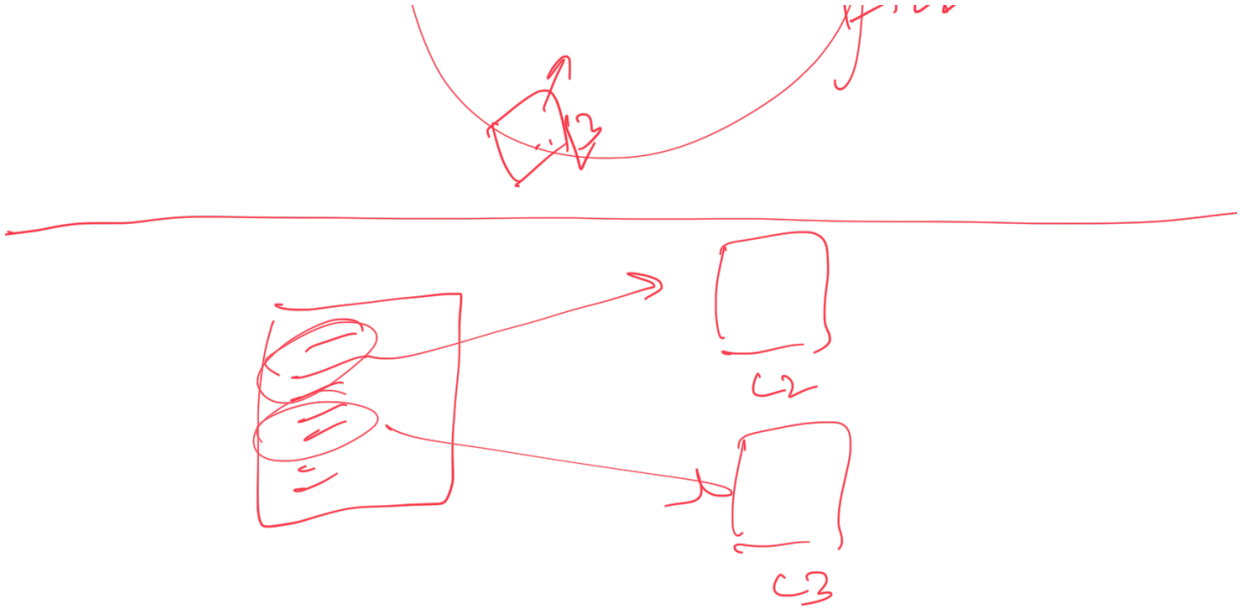
② Add machine

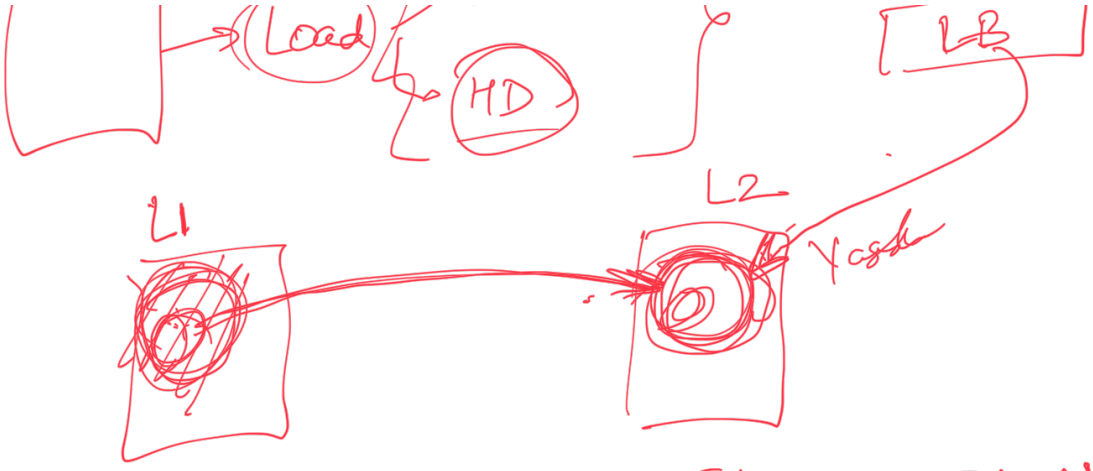


③ Remove machine



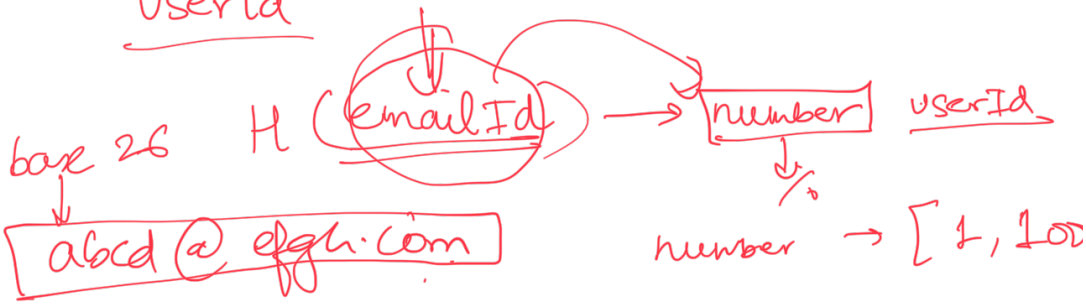

L3
L2
L1
L4
L1 50
L1 25000
L1 100,000
L1 500
L2 5000
L1 10,000
R1
R2
H(userId)
8000
H2
H3
H5
replica


L2
L3
L3 → users that have hash in range
X - Y
shift them to
int hash1(int userId){
L3
H(100)
H(100)
rand(arg)
1000
1000
userid % num-machines
500 bytes
40 GB
80 millions
die
reboot
crash → HD
RAM
CPU, RAM



UserId → User Identifi

Userid



('m' * 26^6 + 'b' * 26^1 + ...) % 1000

12