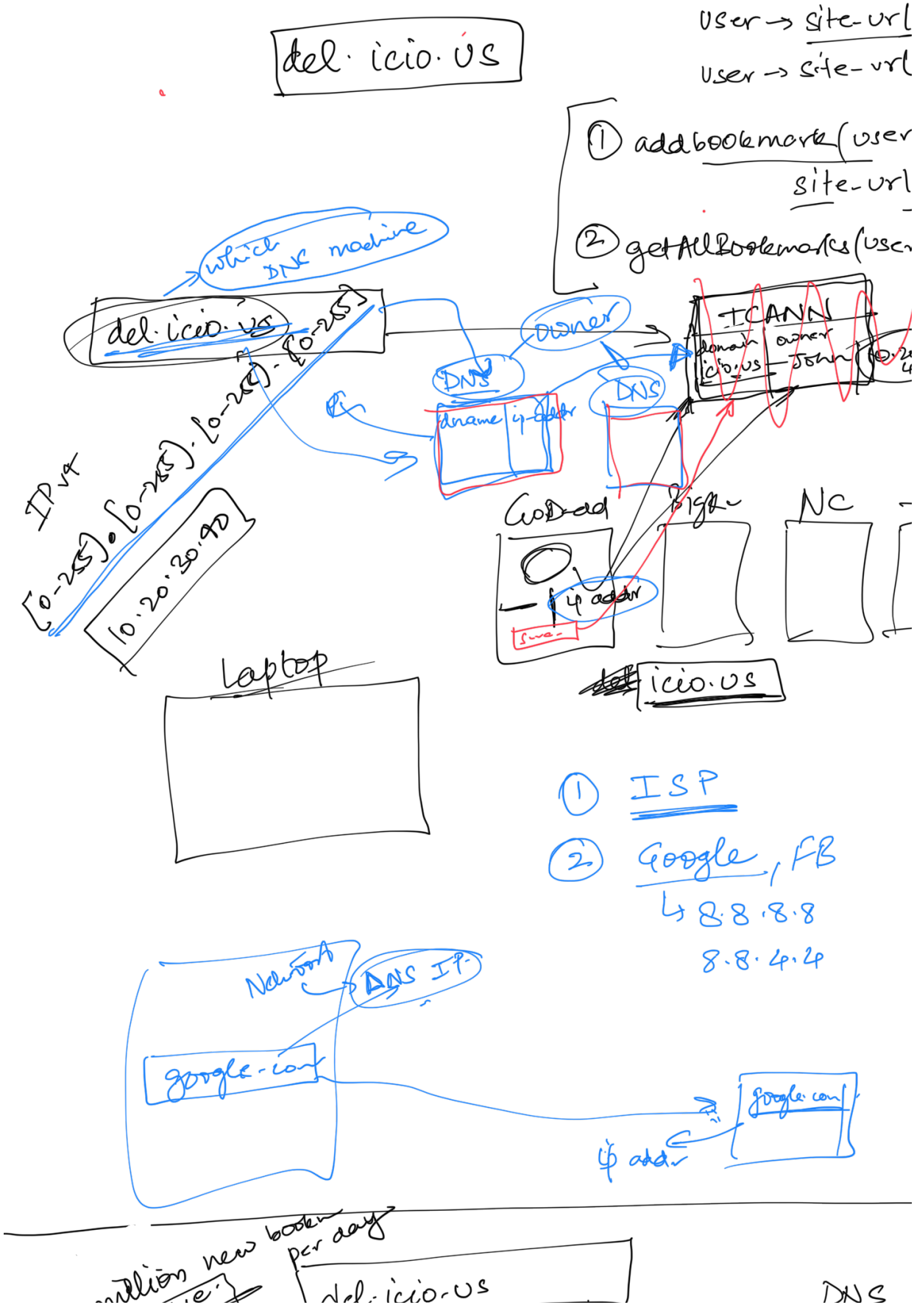
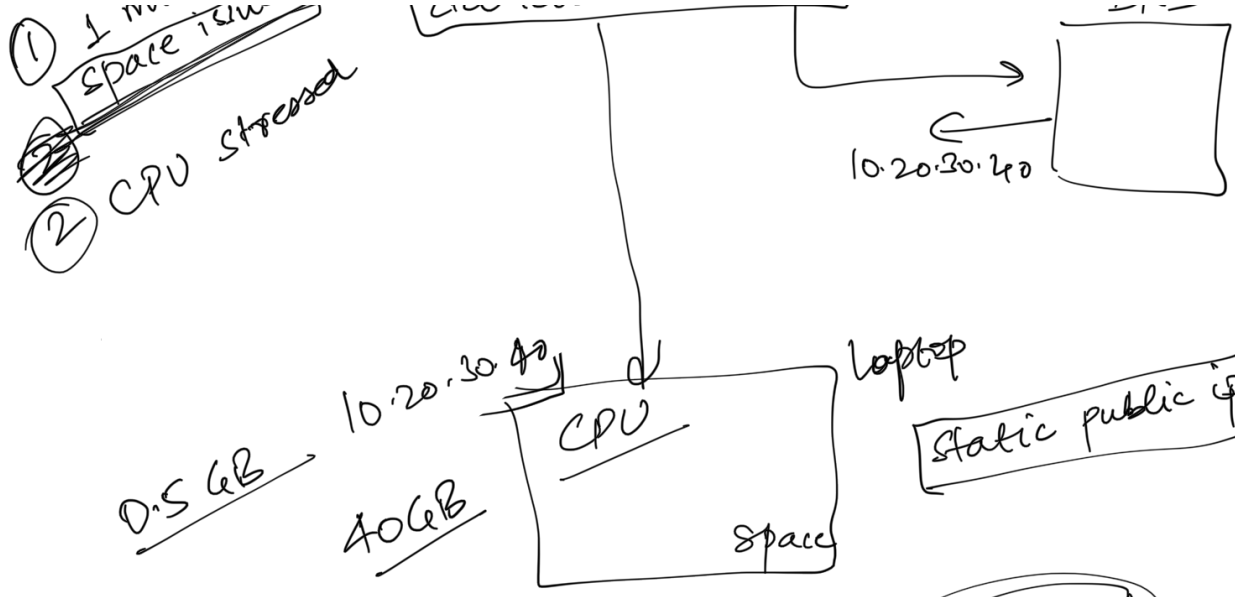
SYSTEM DESIGN - 101



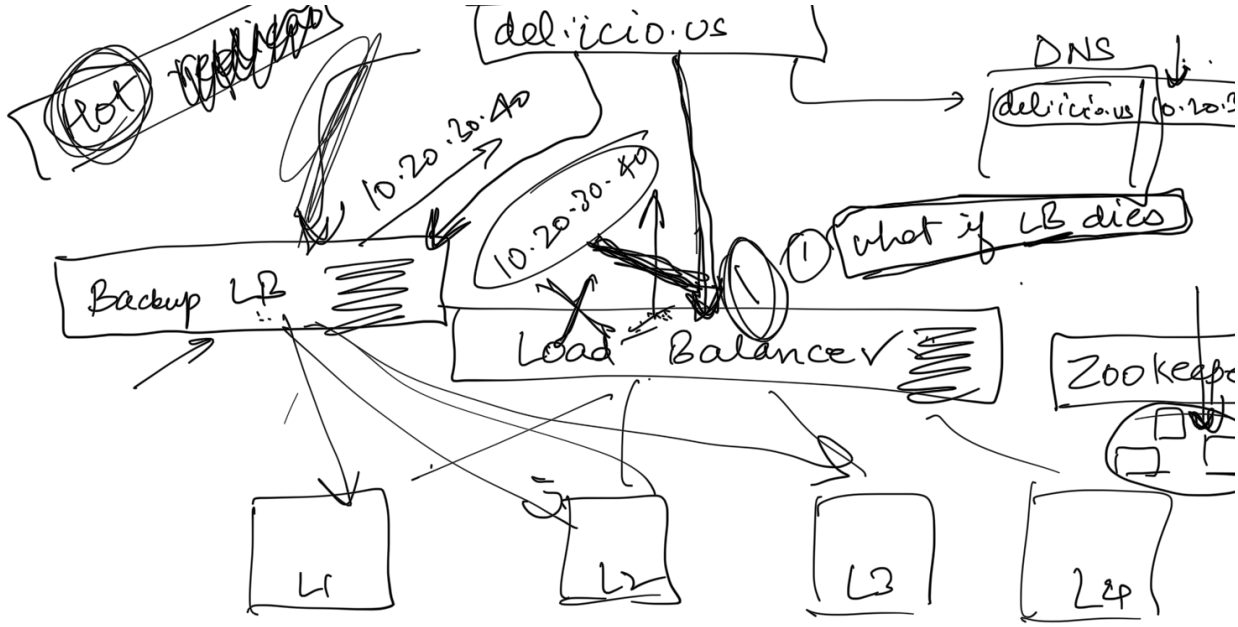


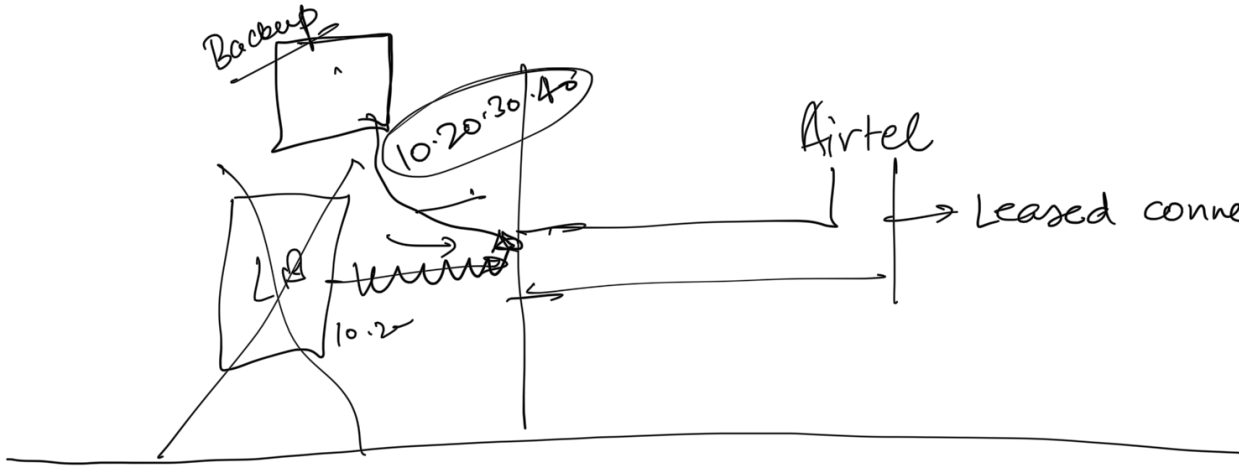
---

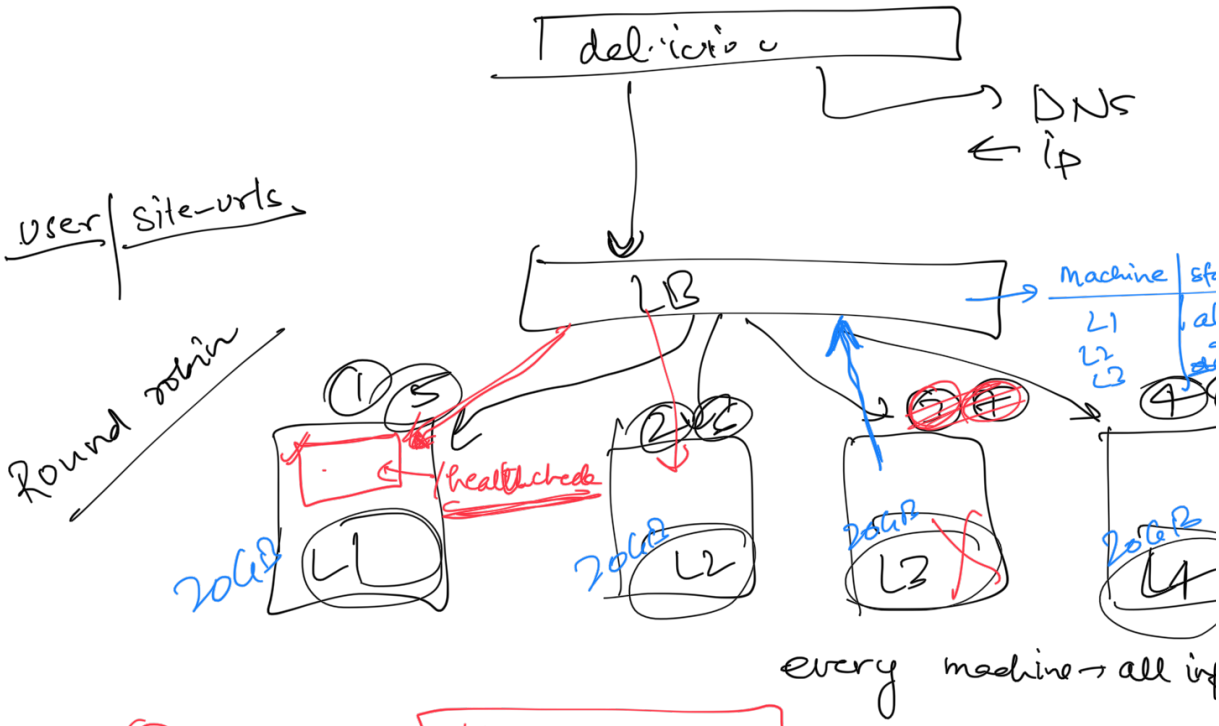
(1) Better machine / increase storage / compre
vertical scaling

60 days ⟺ 80GB

(2) More machines (Horizontal scaling)



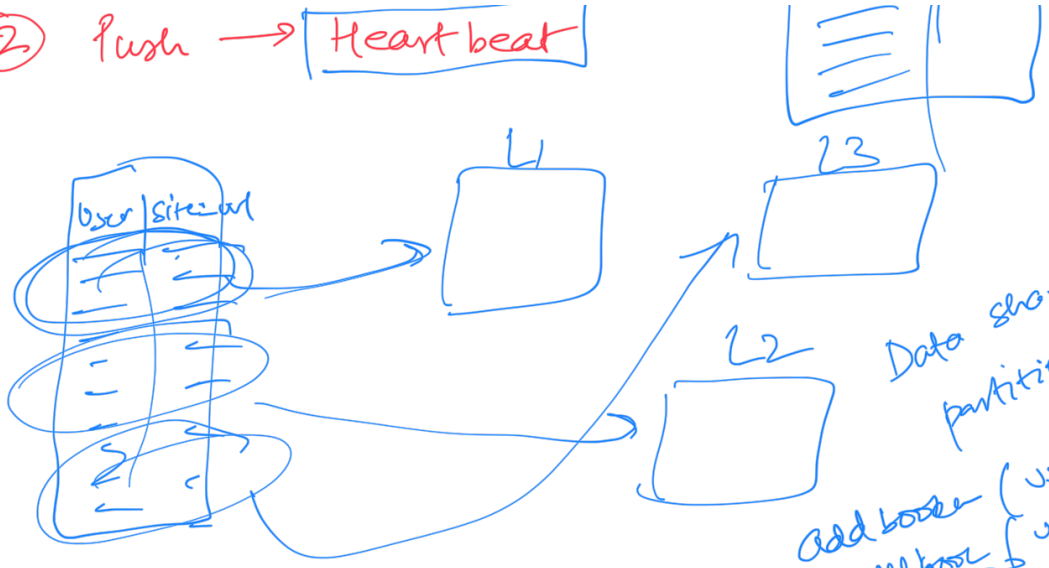




① Pull → Health check

machine | status

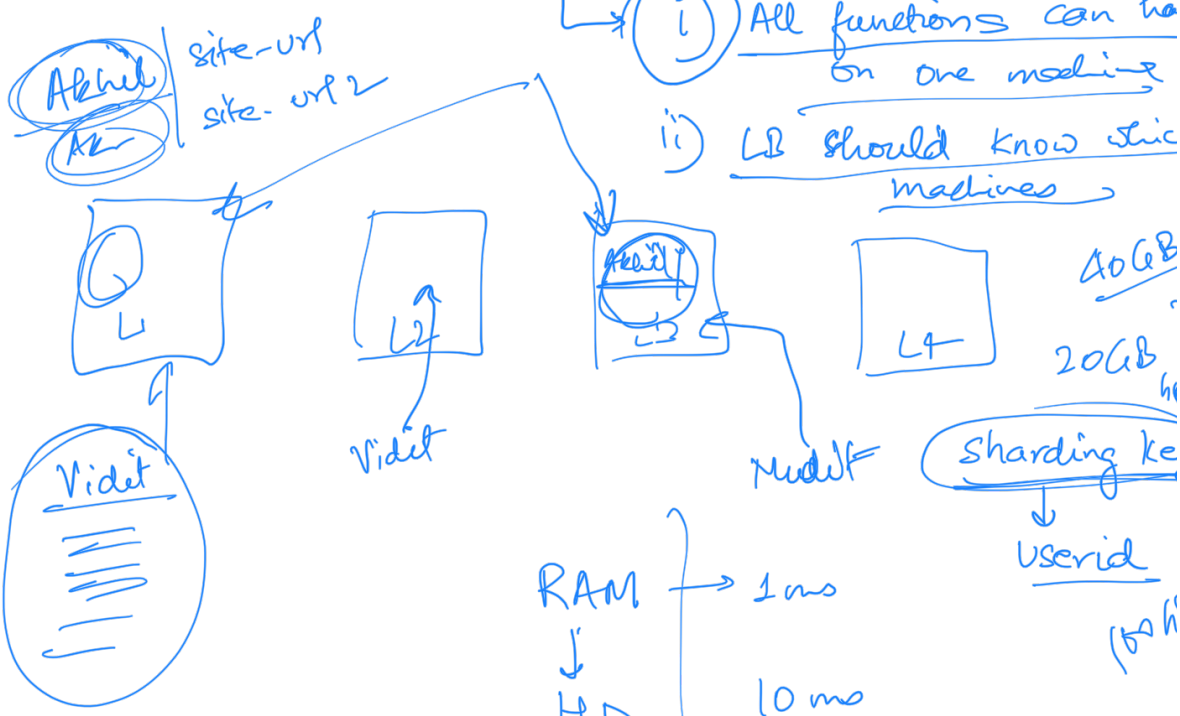
② Push → Heart beat



① How to split data

i) All functions can ha
on one machine

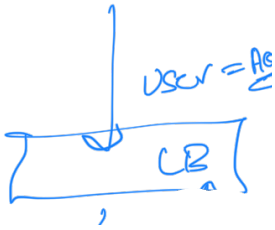
ii) LB should know stic
machines

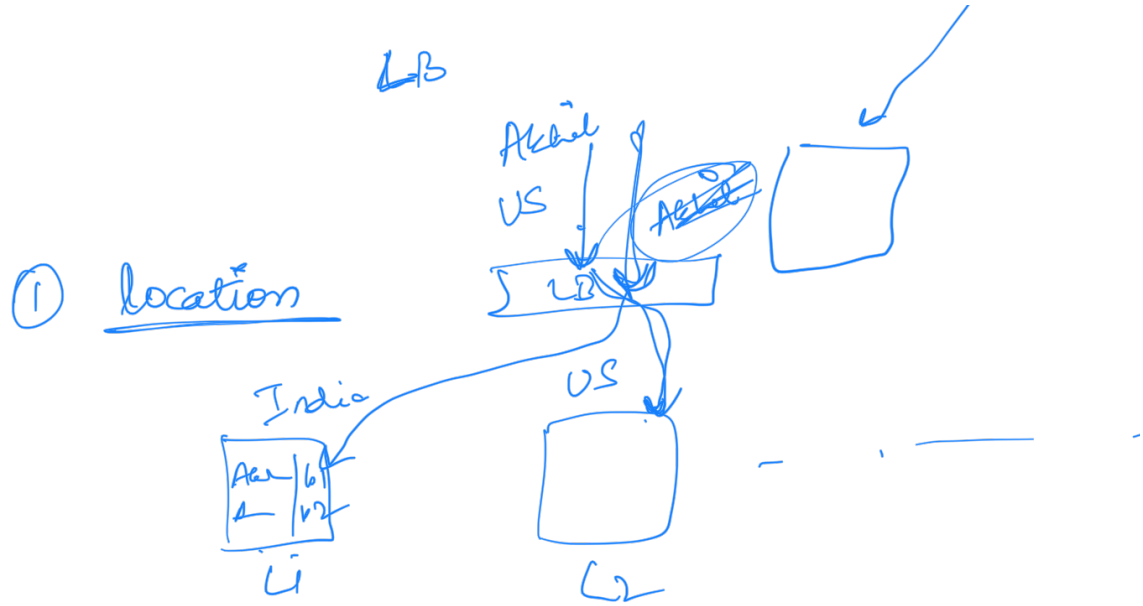


RAM → 1 ms
HD → 10 ms
Another n → 100 ms

① Sharding key = user

→ user → L1, L2, L3, L4





① location

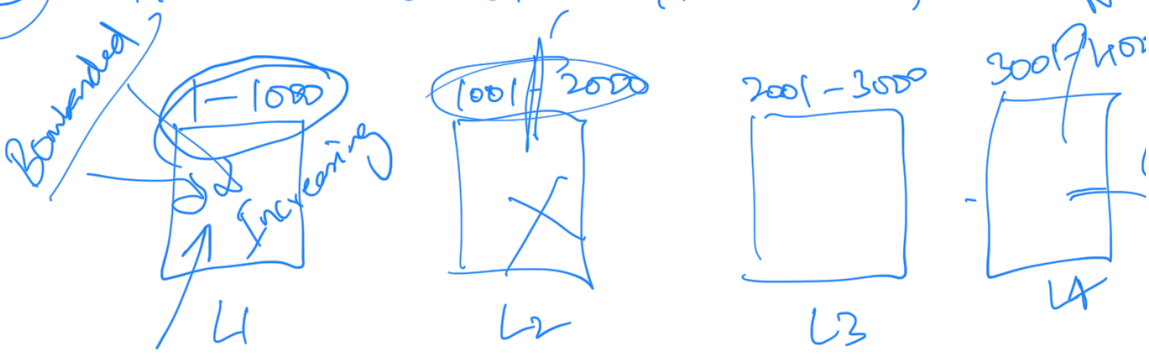
② userid % num_machines

	Machine	Machine
14	2	4
18	2	3
20	0	0
~~27~~	3	2
31	3	1
33	1	3

④ L4, L1

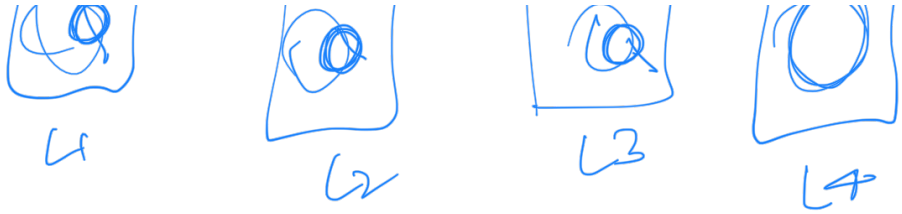
⑤ Add machine

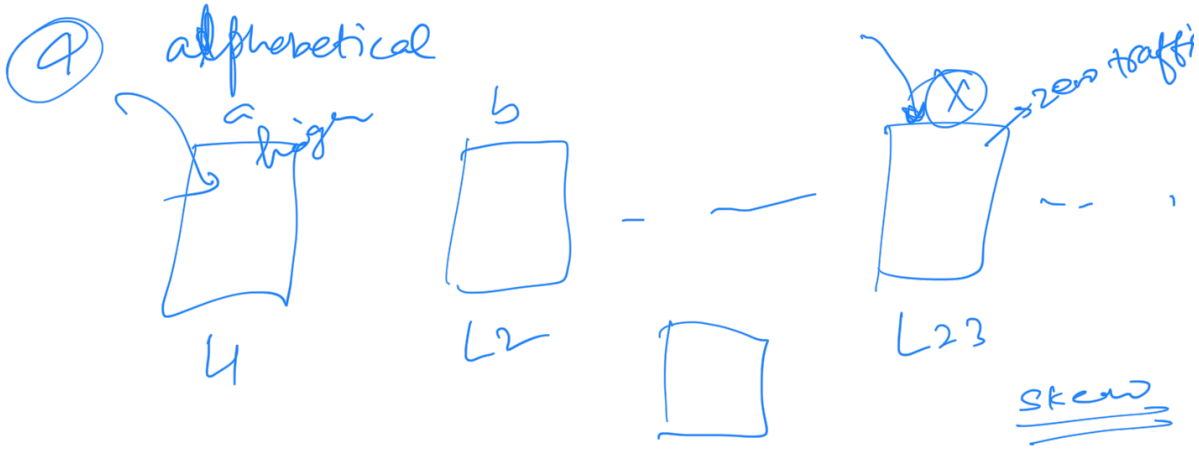
③ First 1000 to L1 Next 1000 to next, ... No



① Skew of load

②

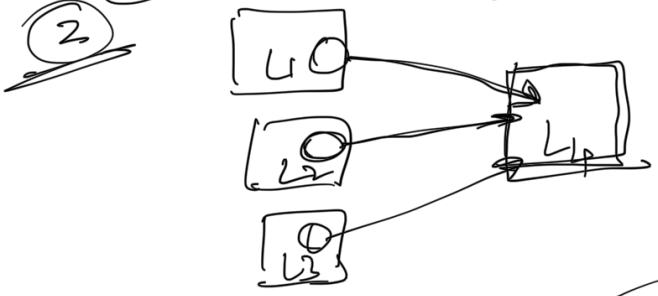




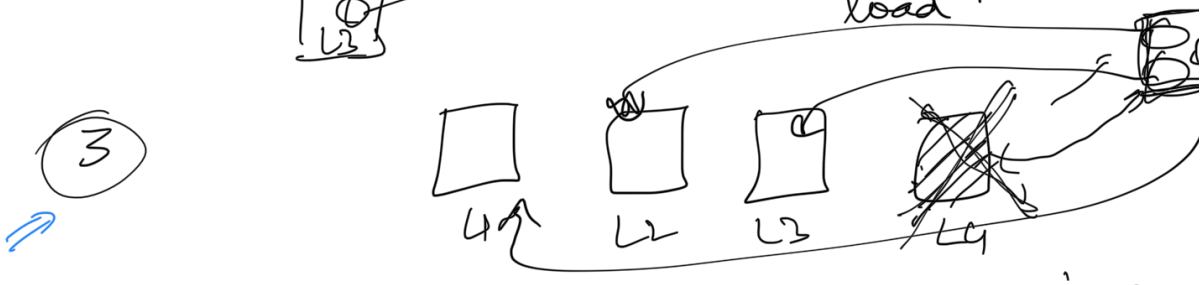
Consistent Hashing

Requirements -

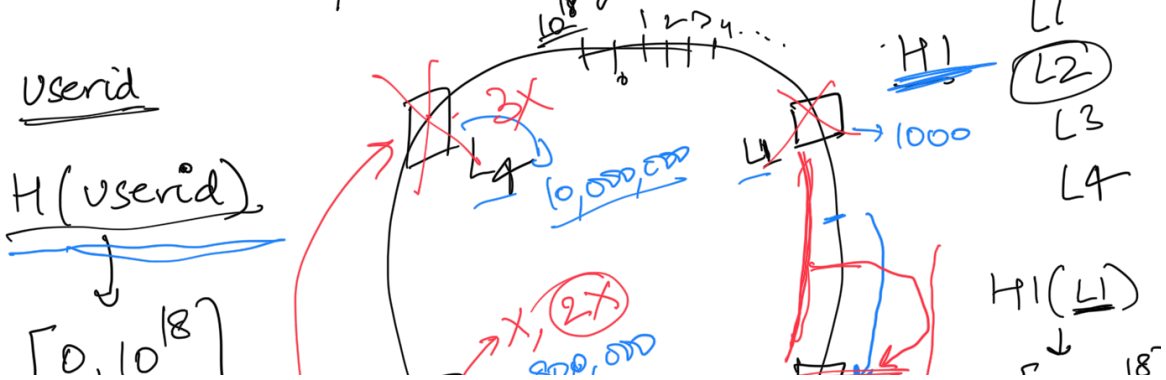
1. userid → machine quickly on L3
   1.1 userid is assigned to only 1 machine

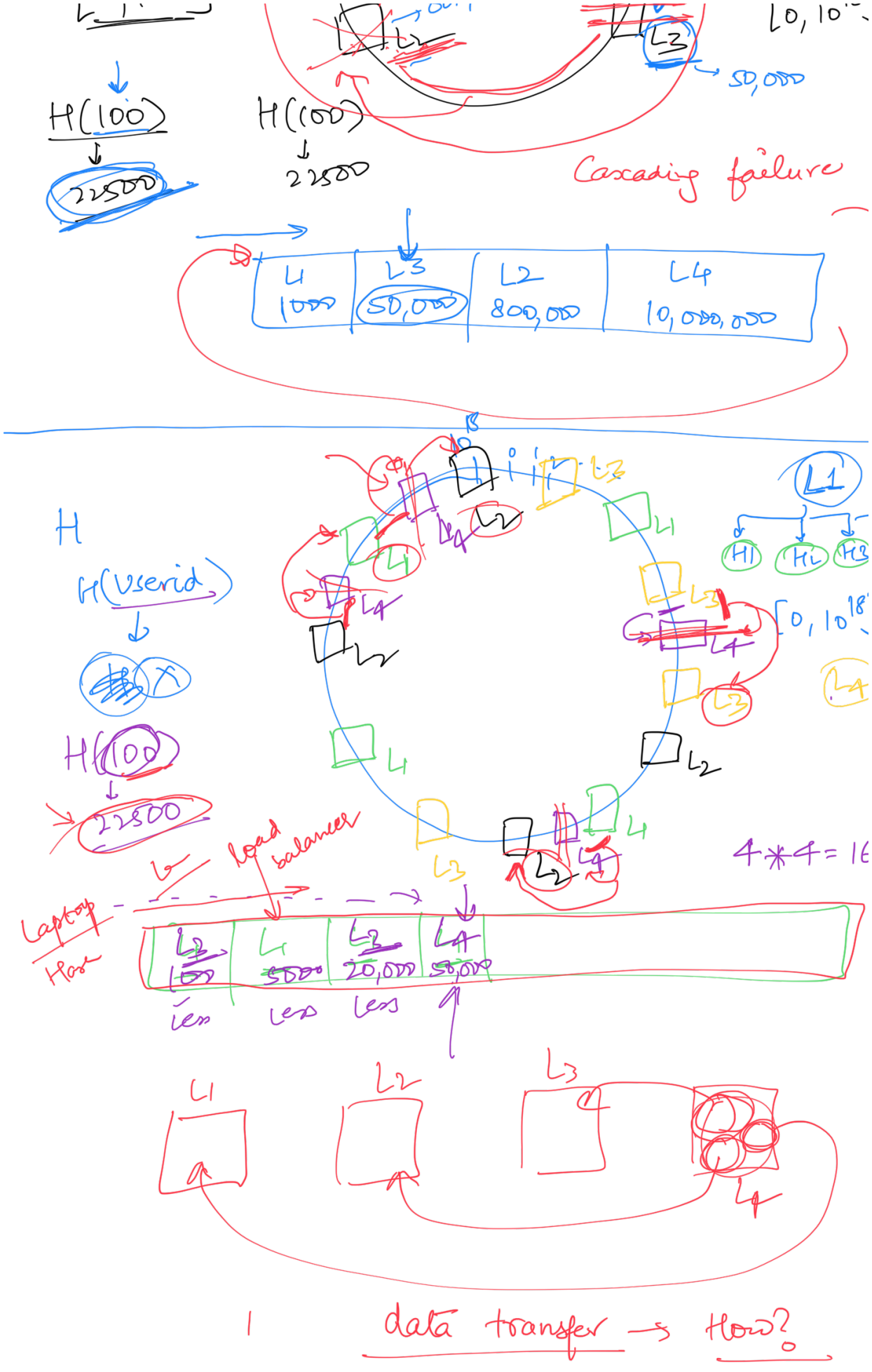


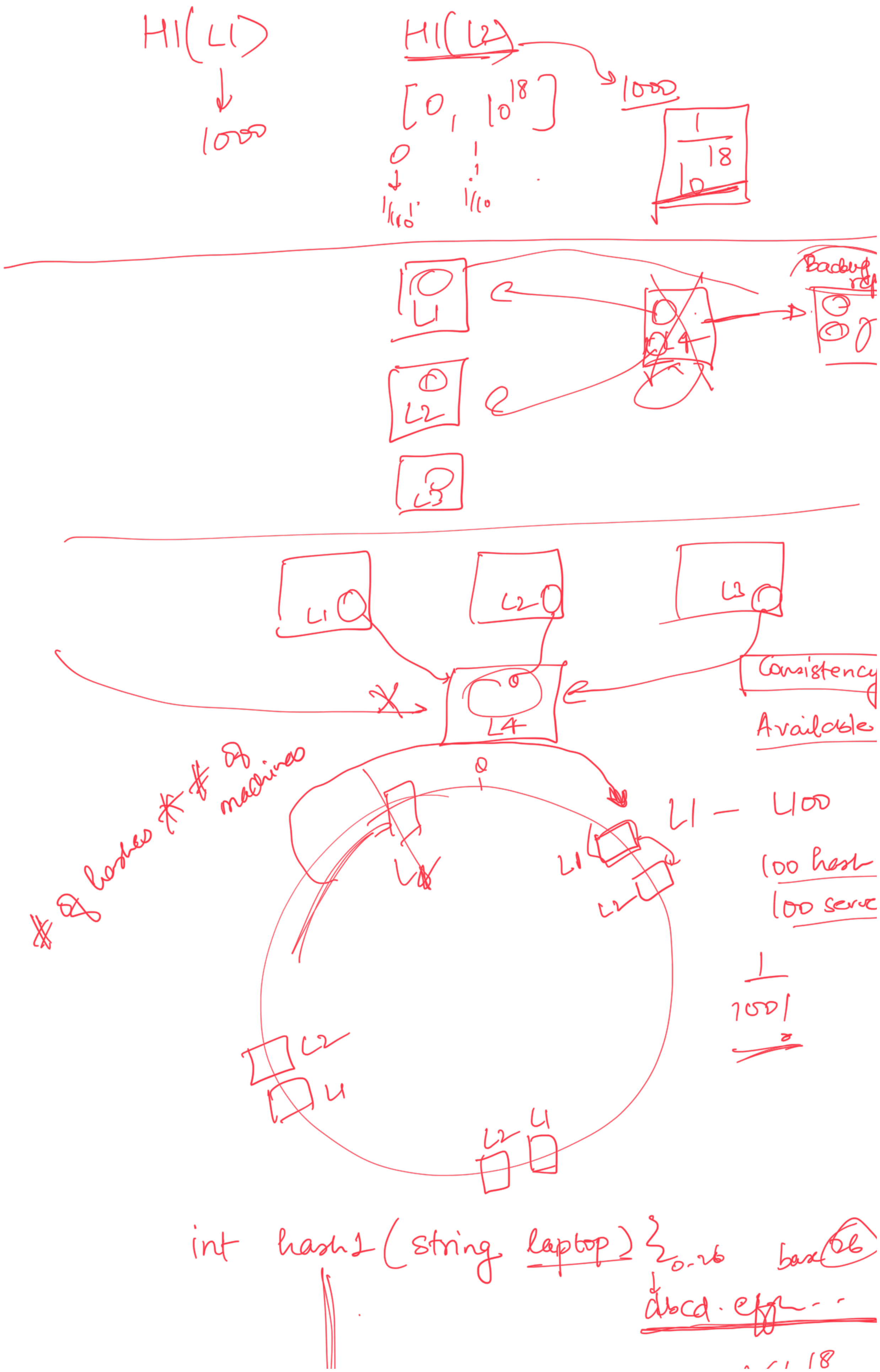
Add machine should reduce every existing mach load.



When a machine dies, it's load is equ
Split amongst remaining.




H(100)
22500
H(100)
22500
50,000
Cascading failure
L1
1000
L3
50,000
L2
800,000
L4
10,000,000
H
H(userid)
H(100)
22500
load balancer
Laptop
Host
L2
100
L1
5000
L3
20,000
L4
50,000
less
less
less
4*4=16
L1
L2
L3
L4
data transfer → How?



H1(L1)
1000
H1(L2)
1000
[0, 10^18]
Backup
L1
L2
L3
L4
Consistency
Available
# of hashes # # of machines
L1 — L100
100 hash
100 server
1/1001
int hash1(string laptop)
0-26
abcd.efgh

return true ;

% 10

$[0, 10^{18}]$