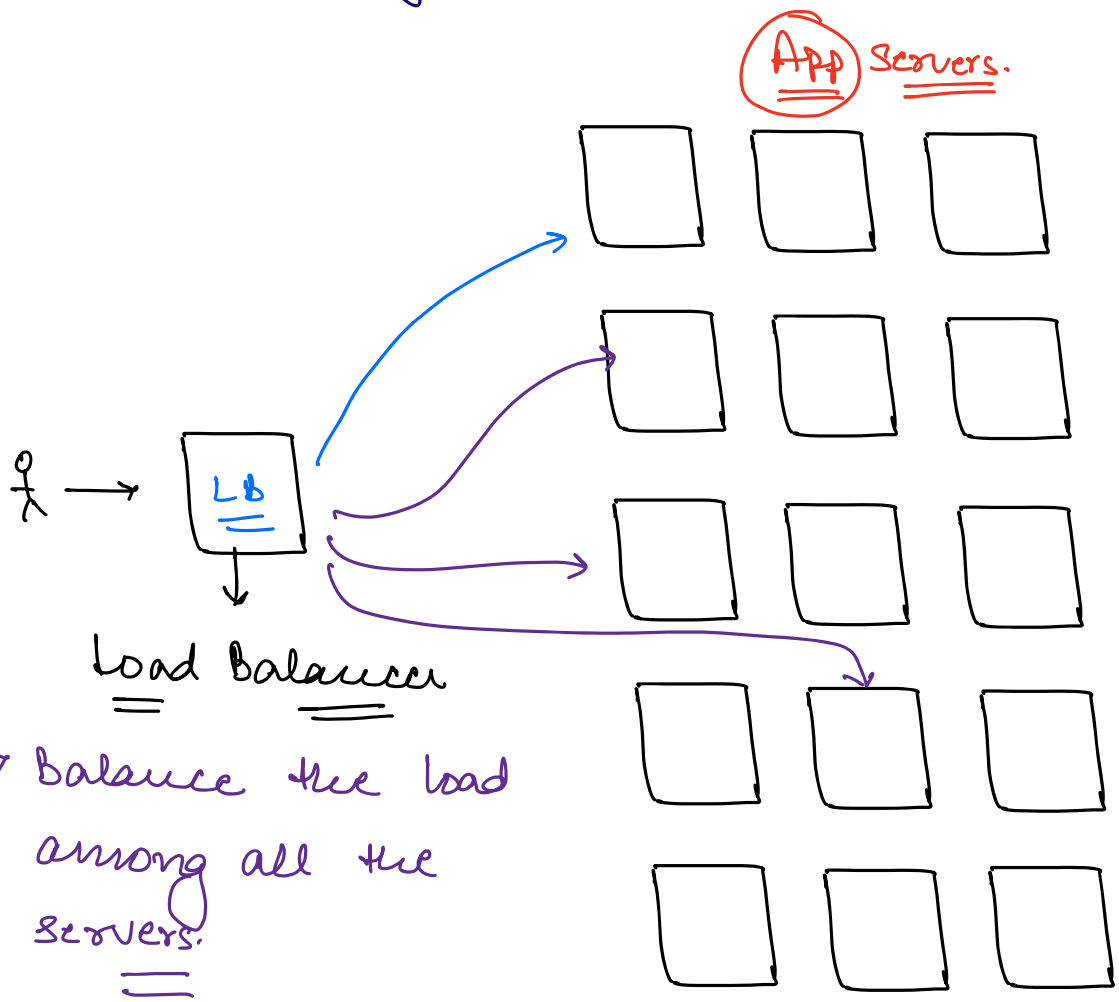
Agenda.

- → Load Balancing
- → Properties of a good Load balancer.
- → Stateful (vs) Stateless Load Balancing
- → Algorithms

⇒ Vertical Scaling (vs) Horizontal Scaling



⇒ Balance the load among all the servers.

⇒ Servers can crash at any time.

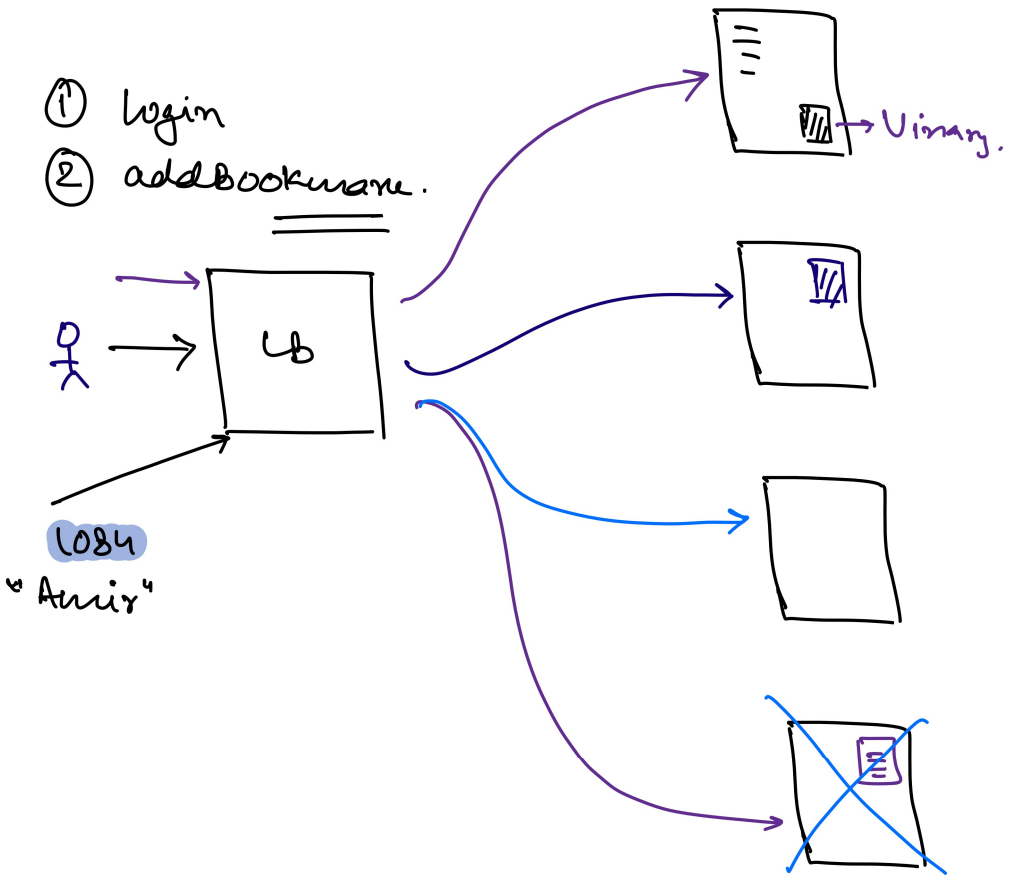
→ Heart Beat.

→ Health Check.

Load Balancer can become a SPOF.

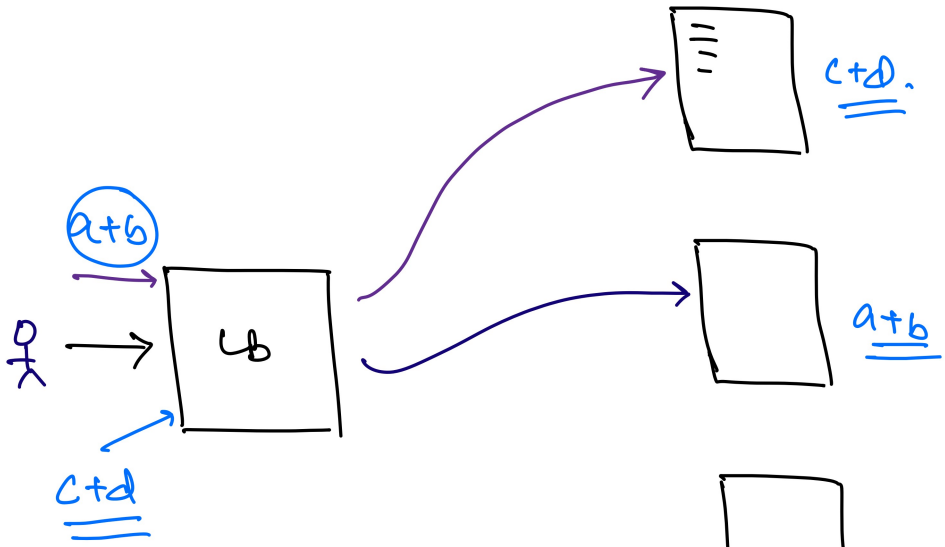
We can have multiple load balancers and in DNS, we configure multiple Ip addresses instead of only one 1 Ip.

Bookmarking.



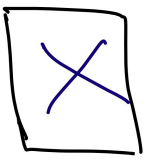
⇒ Stateful Load Balancing

Calculator.

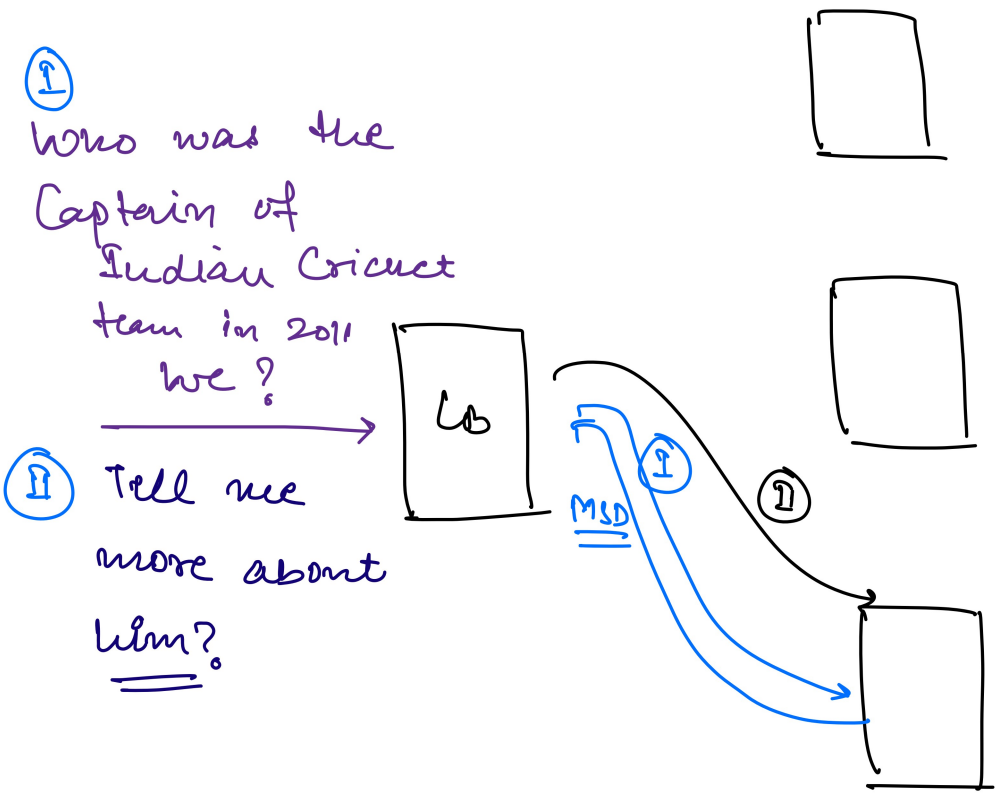


⇒ Stateless

Load Balancing



ChatGPT.



Load Balancing / Routing Algorithm

→ Properties.

I) fast

II) Equal Distribution.

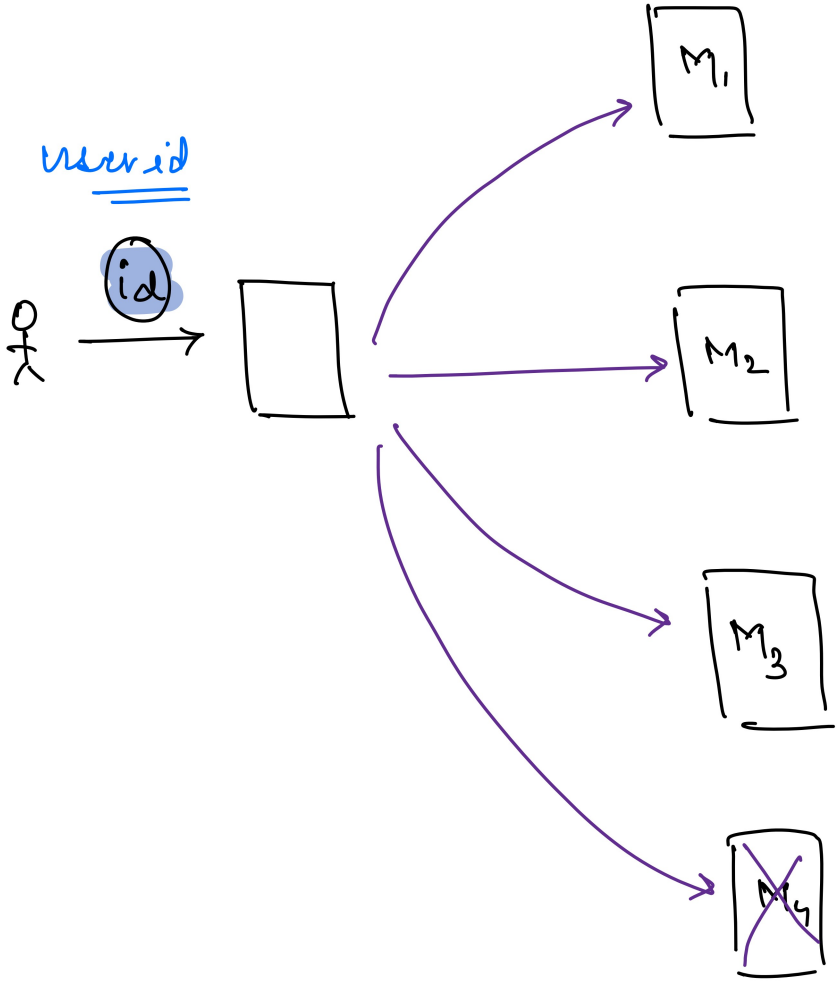
III) Easy to add / remove m/c.

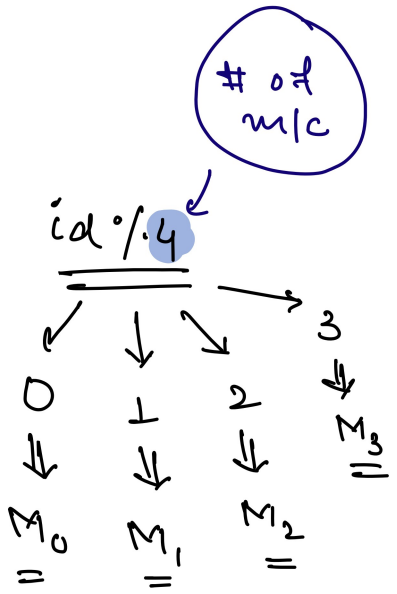
IV) Minimal Data transfer in case m/c is getting added / removed.

Round Robin Algorithm.

user id			
M_1	0	4	8
M_2	1	5	9
M_3	2	6	10
M_4	3	7	11

Stateful





→ Fast & Easy to implement.

→ Equal distribution of load.

→

user_id

M_1	0	4	8
M_2	1	5	9
M_3	2	6	10
~~M_4~~	~~3~~	~~7~~	~~11~~

If M4 goes down.

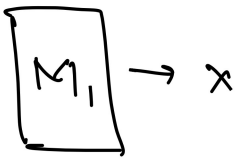
user-id % 3 $\begin{cases} 0 \\ 1 \\ 2 \end{cases}$

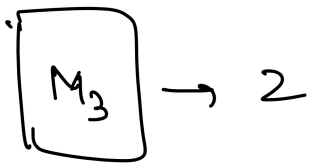
There can be lot of unnecessary data migration.

M_1	0	3	6	9
M_2	1	4	7	10
M_3	2	5	8	11

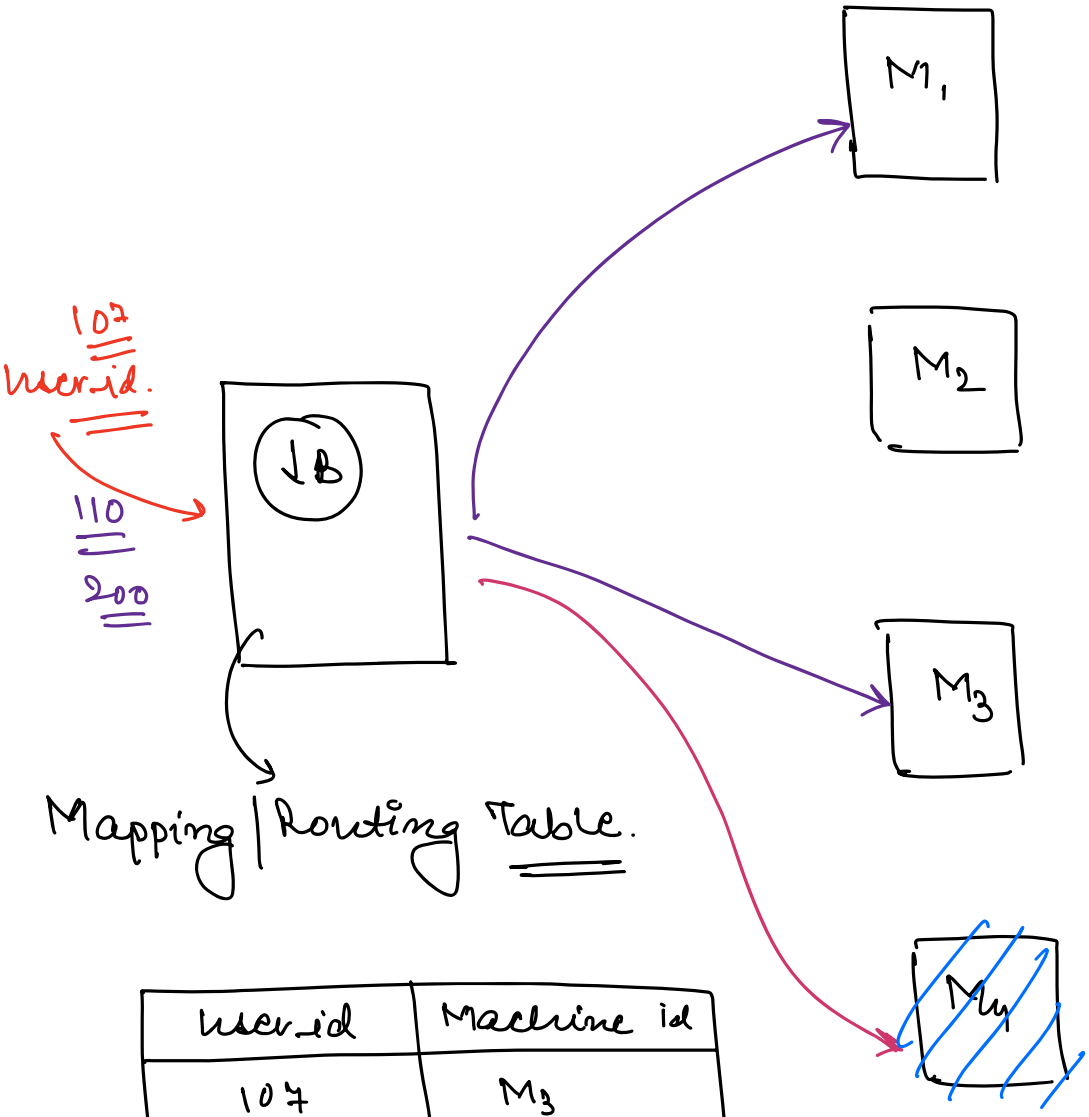
Least Connection

⇒





least Response Time



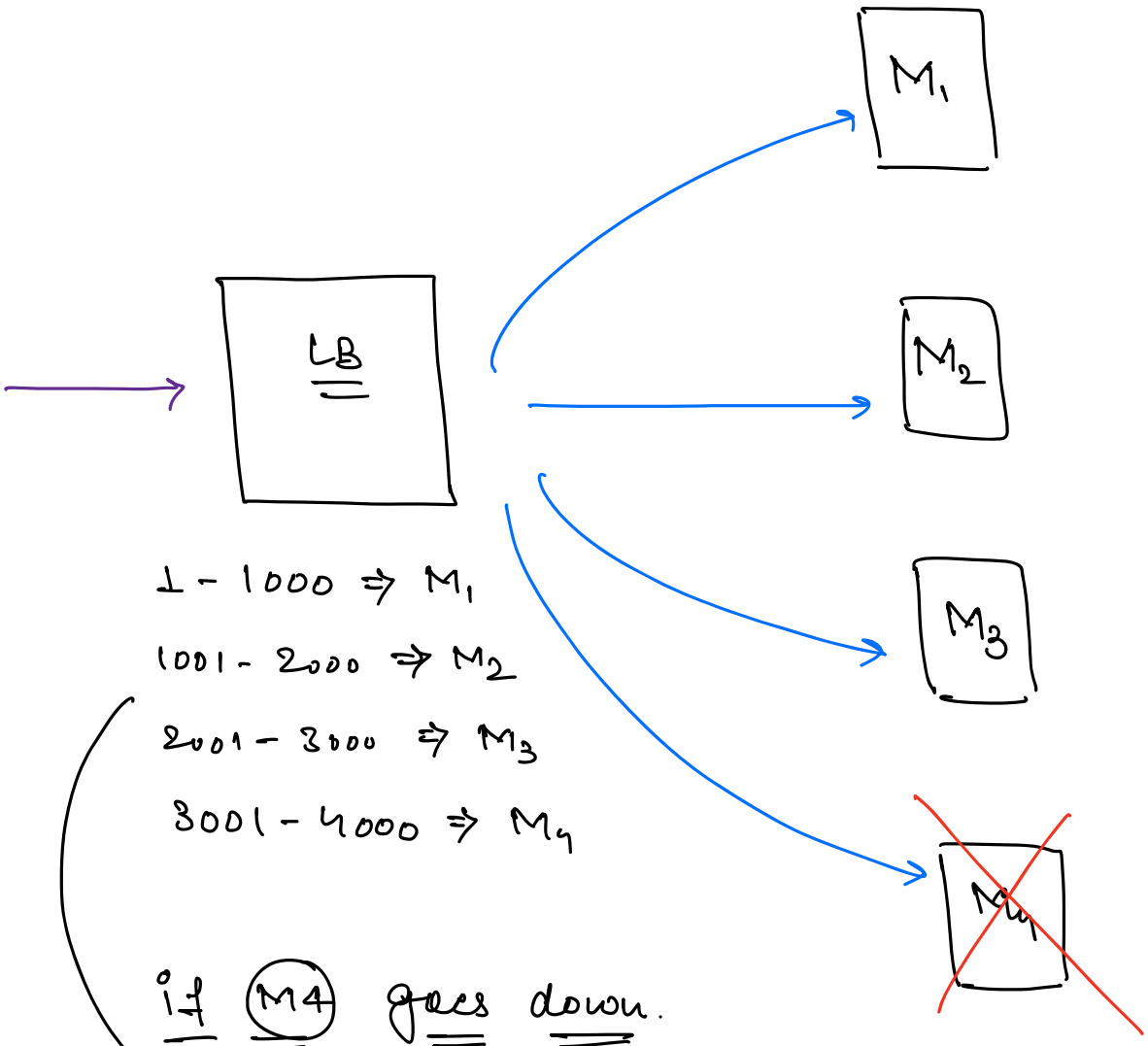
user id	Machine id
107	M_3
104	M_2
110	M_1
200	M_4 ✗
305	M_4 ✗

→ Stateful LB.

→ Easy to implement fast.

→ No impact on the requests which are going to the m/c other than the one which is going down.

Range Based Distribution.



1 - 1000 ⇒ M_1

1001 - 2000 ⇒ M_2

2001 - 3000 ⇒ M_3

3001 - 4000 ⇒ M_4

if M4 goes down.

1 — 1333 → M_1

1334 - 2666 → M_2

2667 - 4000 → M_3.

⇒ Lot of Data Migration.

Consistent Hashing

↳ Stateful