

Performance Q4

$$\text{Performance} = \frac{\text{Potential}}{(\text{Scaln})} - \text{Interference}$$

(Your responsibility)

`support@scaln.com`

- Computer Networks - }
→ DBMS

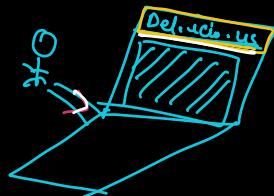
Joshua
2005/2006

♀ → Create an account
♀ → Some info

DEL. ICIO. 4.S



→ Create Account
→ add Bookmark (userId, url),
get All Bookmarks (userId)



To be continued --

Computer Network : DNS

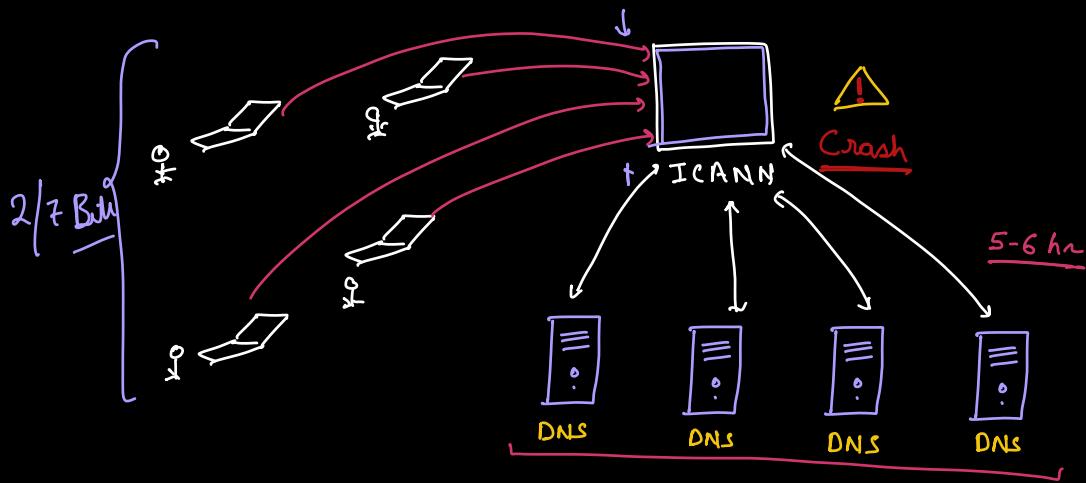
Naming System (MP)

49700L

Ambikapur (CJ)

Host Name	IP
www.scaler.com	121.56.41.47
fb.com	58.56.120.11
...	...

ICANN (Source of truth
(Non profit org) for the mappings)

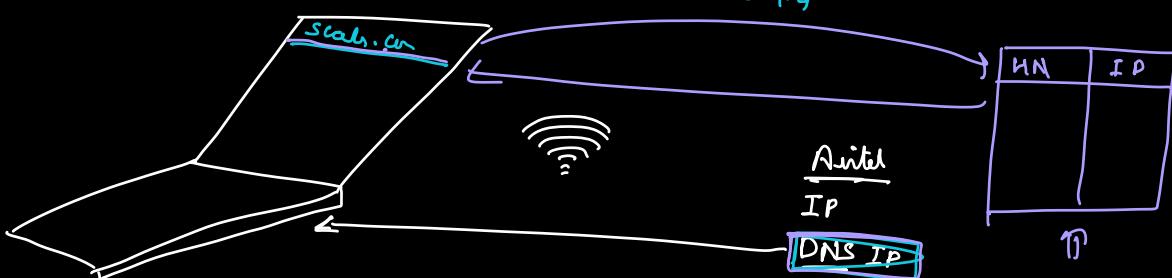


Who maintains DNS?

① ISP (Airtel, Tata Docomo, Jio ...)

③ Google, Facebook

→ 8.8.8.8
→ 8.8.4.4



Joshua
(2003/2006)

User Id	url
10	Scalr.com $\approx 1KB$
20201	google.com
10	geeksfgeek.com



RAM : 512MB
Hard Disk : 40GB
Clock Cycle : 1.6 GHz ≈

{ 1 cycle
Database :
↳ Other pros

1 record takes $\rightarrow 1000 \text{ Bytes} \approx 1 \text{ KB}$

& he gets 1 million requests per day.
(Add BookMark)

Memory required to save records created per day

$$1 \text{ KB} \times 1 \text{ million}$$

$$1000 \text{ B} \times 10^6$$

$$10^9 \text{ B}$$

$$1 \text{ GB}$$

If 1 GB records are getting saved per day
40GB $\longrightarrow \leq 40 \text{ days}$

Scaling challenge

Solution

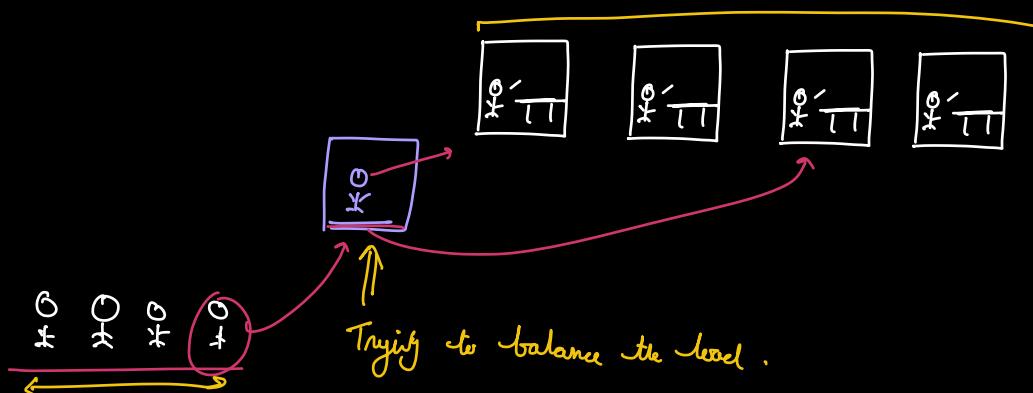
① Buy a better system with high config.

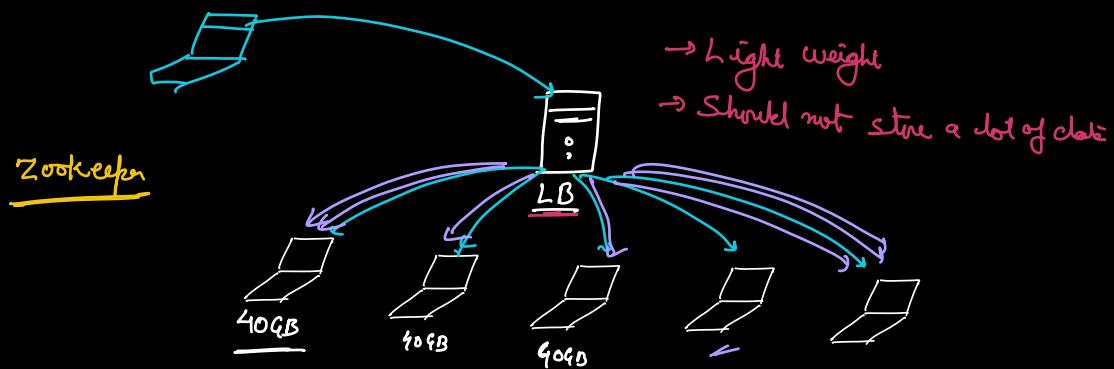
($\frac{500 \text{GB}}{50 \text{day}}$, 4 GB RAM)

Vertical Scoring

② Buy more laptops (of same config)]

Horizontal Scaling

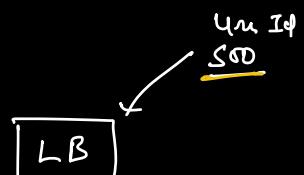




⇒ Round Robin
⇒ Hash connects first.
⇒ Hash response time first

Data Partitioning / Sharding

Data of a specific user
is stored in a single server



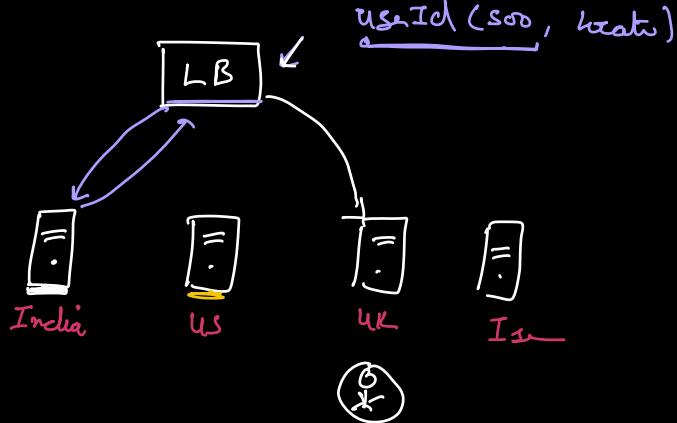
→ Region / Geography ✗

→ Round - Robin ✗

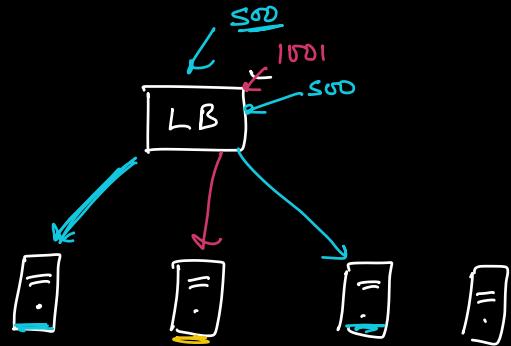
→ Hash function / Mod

→ id Ranges ✗

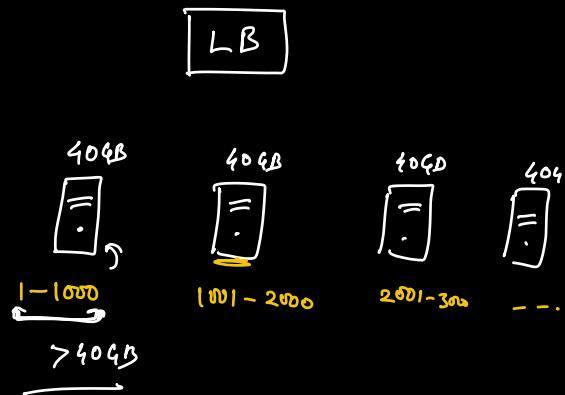
Geography X



Round Robin X



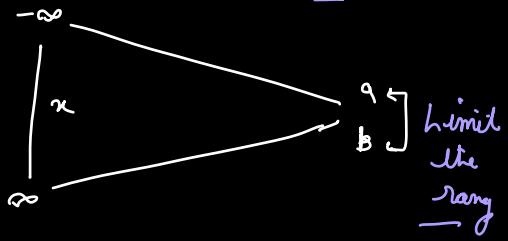
Range Based X



Hash Function

$$f(x) = \underline{y}$$

$\%$ \Rightarrow Mod



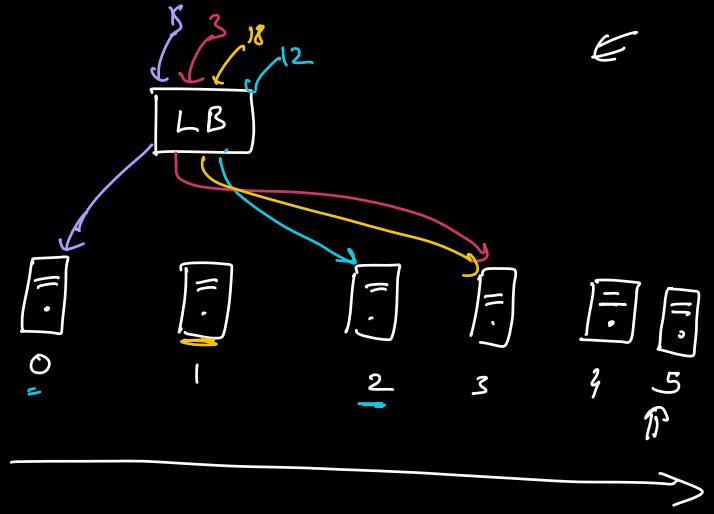
$a \% b \Rightarrow$ Remainder of $a \div b$

$$\begin{aligned} 113 \% 10 &\Rightarrow 2 \\ 105 \% 10 &\Rightarrow 5 \\ 110 \% 10 &\Rightarrow 0 \end{aligned} \quad [0-9]$$

$$A \% n \Rightarrow [0, n-1]$$

[Using %, N]

User Id	<u>N=5</u>	<u>N=6</u>
12	2	0
13	3	1
18	3	0
15	0	3



Break till 10:55 pm

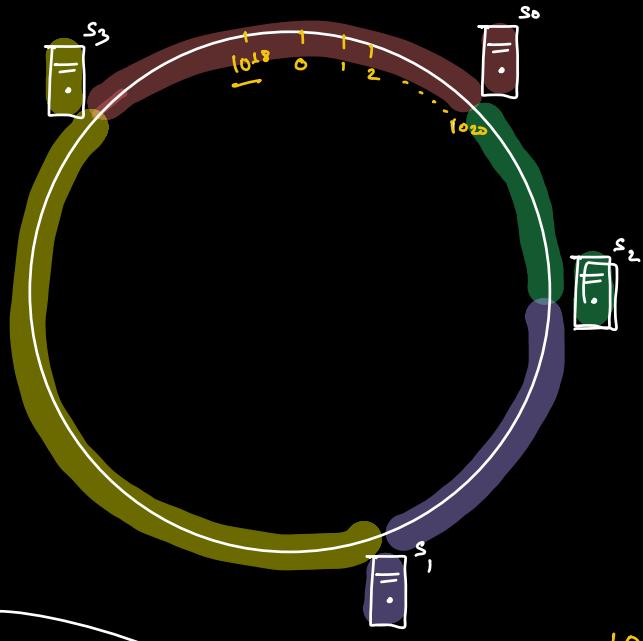
Consistent Hashing

Hash fn

$$h_u(U_{\text{Id}}) \Rightarrow [0, 10^8]$$

$$\underline{h_u(s_0) \Rightarrow 10,111}$$

$$h_u(101) \Rightarrow 10^8$$



Hash fn

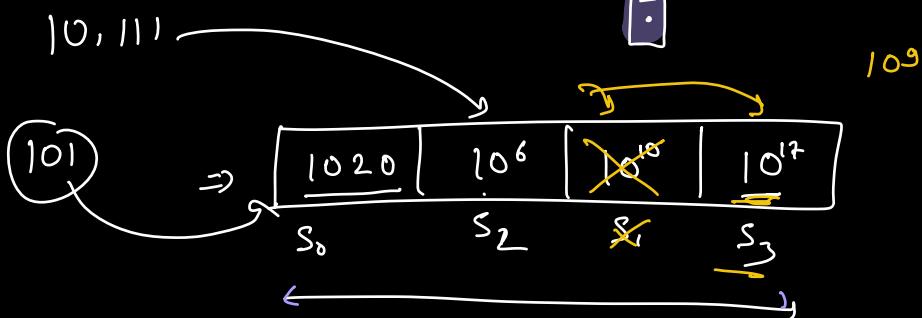
$$h_s(S_{\text{Id}}) \Rightarrow [0, 10^8]$$

$$h_s(s_0) = 10^{20} \quad \}$$

$$h_s(s_2) = 10^6$$

$$h_s(s_1) = 10^{10}$$

$$h_s(s_3) = 10^{17}$$



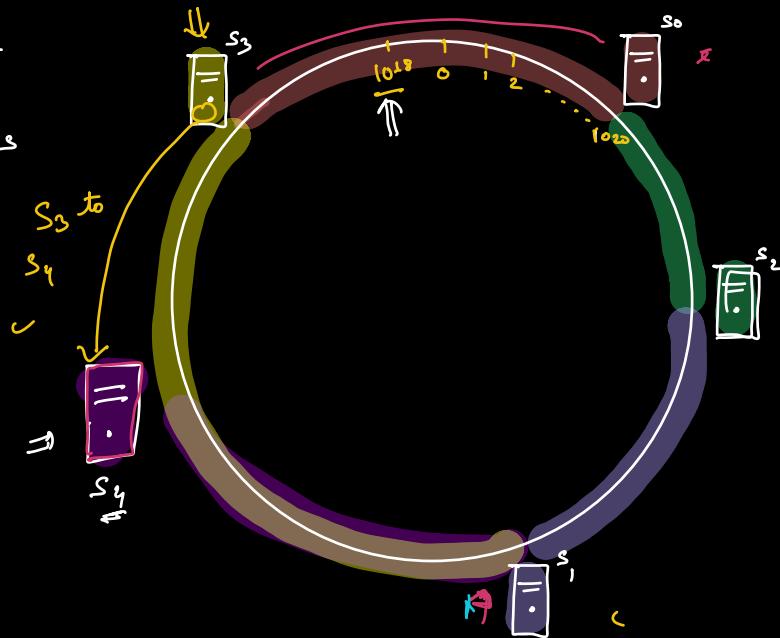
$$h(101) \Rightarrow 10$$

$$h(\underline{301}) \Rightarrow \underline{100}$$

Add 1 more

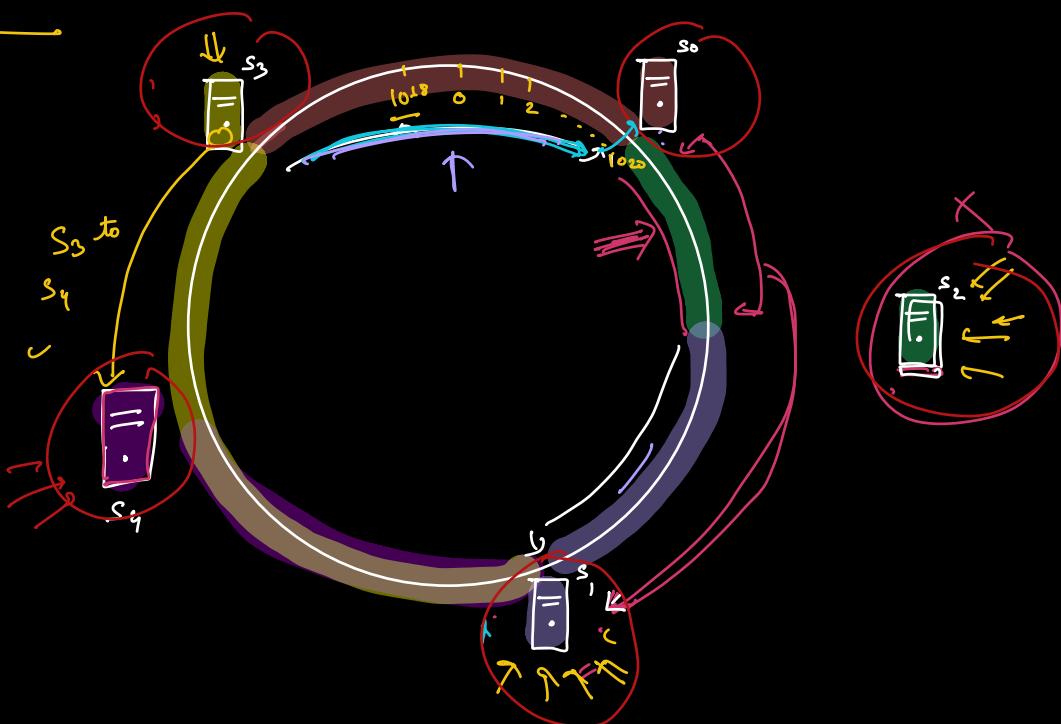
Server //

$$h_s(s_7) = 10^{18}$$



$$P(x) = \frac{1}{10^{18}} = 1 \times 10^{-18} \approx 0$$

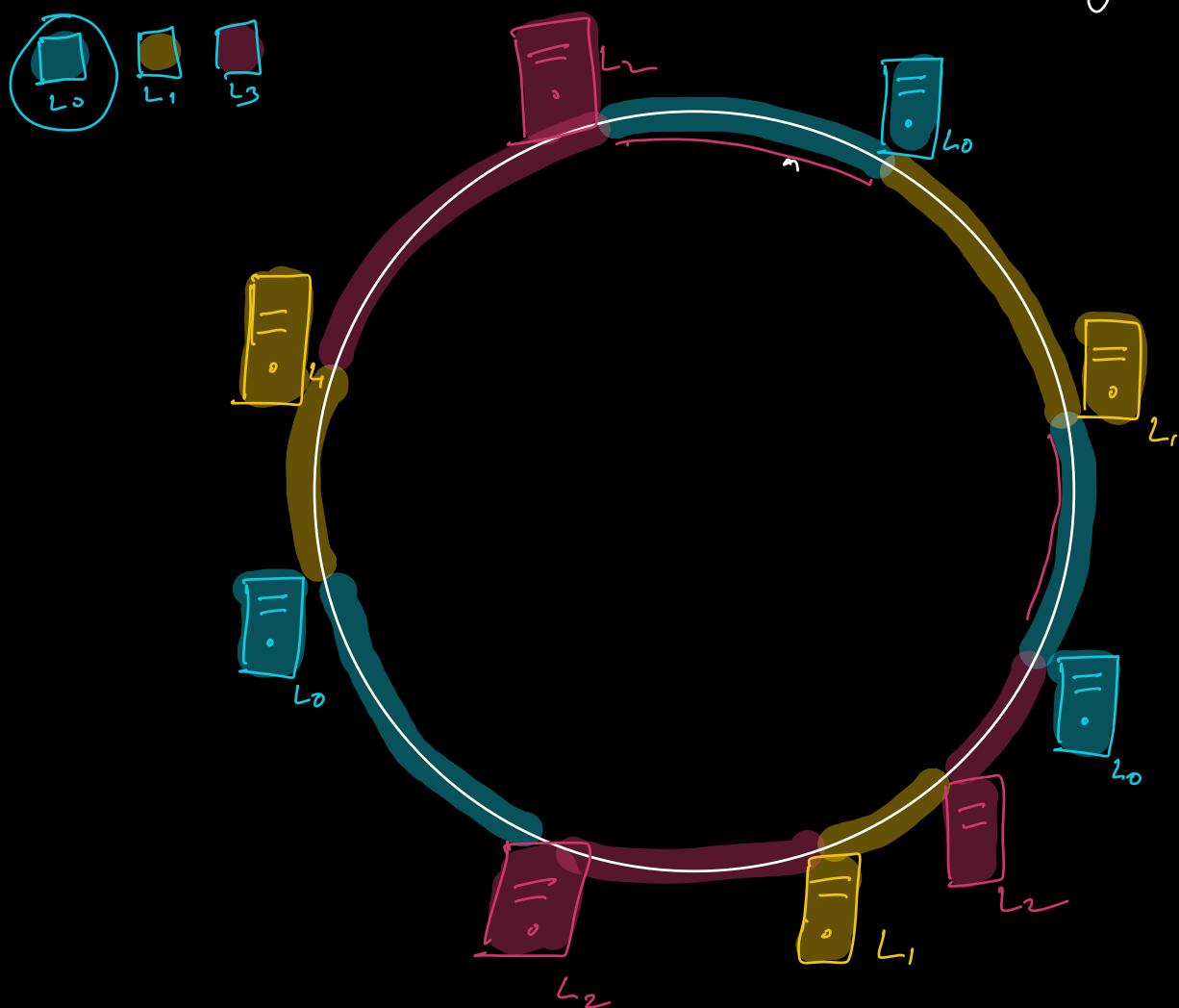
Server goes down



Ideally

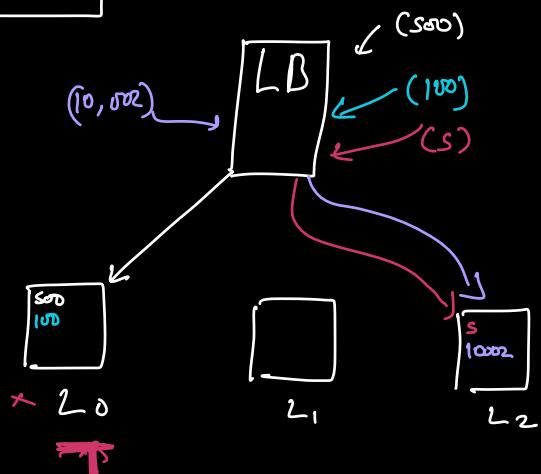
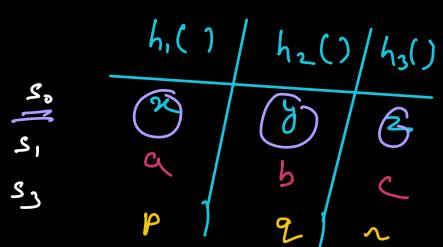
① Adding a new server should reduce the load of all the servers equally.

② When a server goes down \Rightarrow the load should get distributed to all other servers equally.

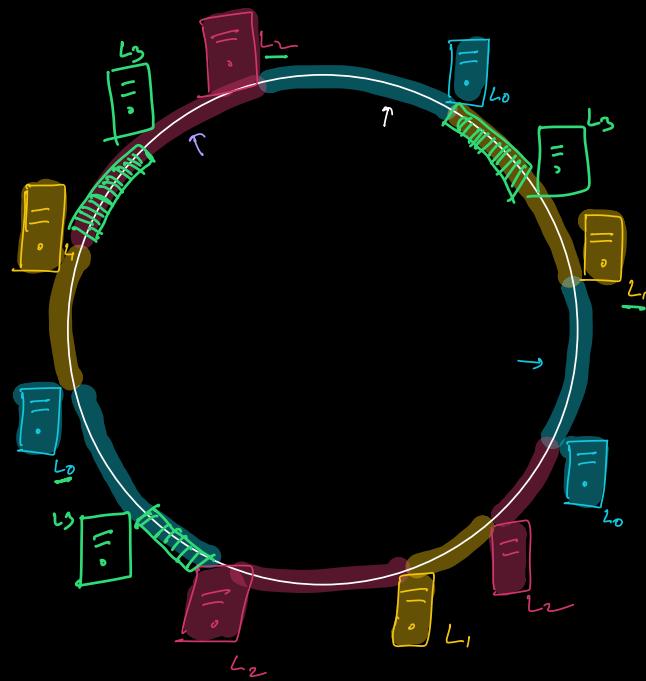


User Id	Hardcard	Server
s00	12000	L0
100	10 ⁸	L0
s	10 ¹⁶	L2
10,002	8x10 ¹⁵	L2

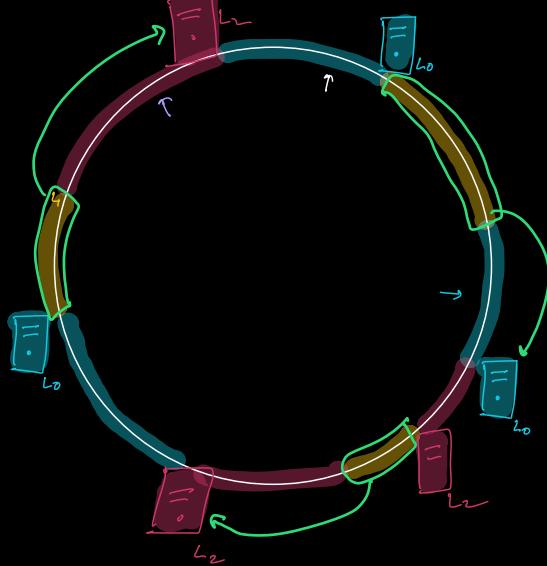
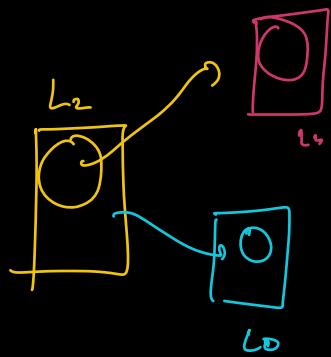
10	1000	s0000	10 ⁴	10 ⁶	8x10 ⁶	10 ¹⁰	10 ¹⁵	10 ¹⁷
L0	L1	L0	L2	L1	L2	L0	L1	L2

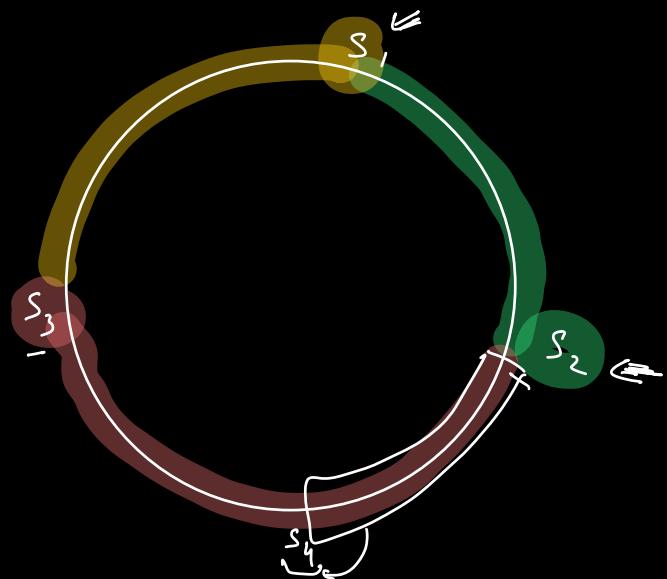
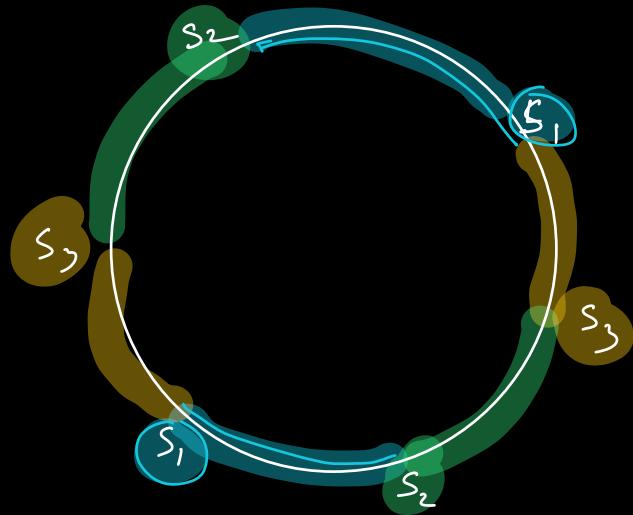


① Add a new Server



② If a server dies





Statis IP

20. 99. 240. 11