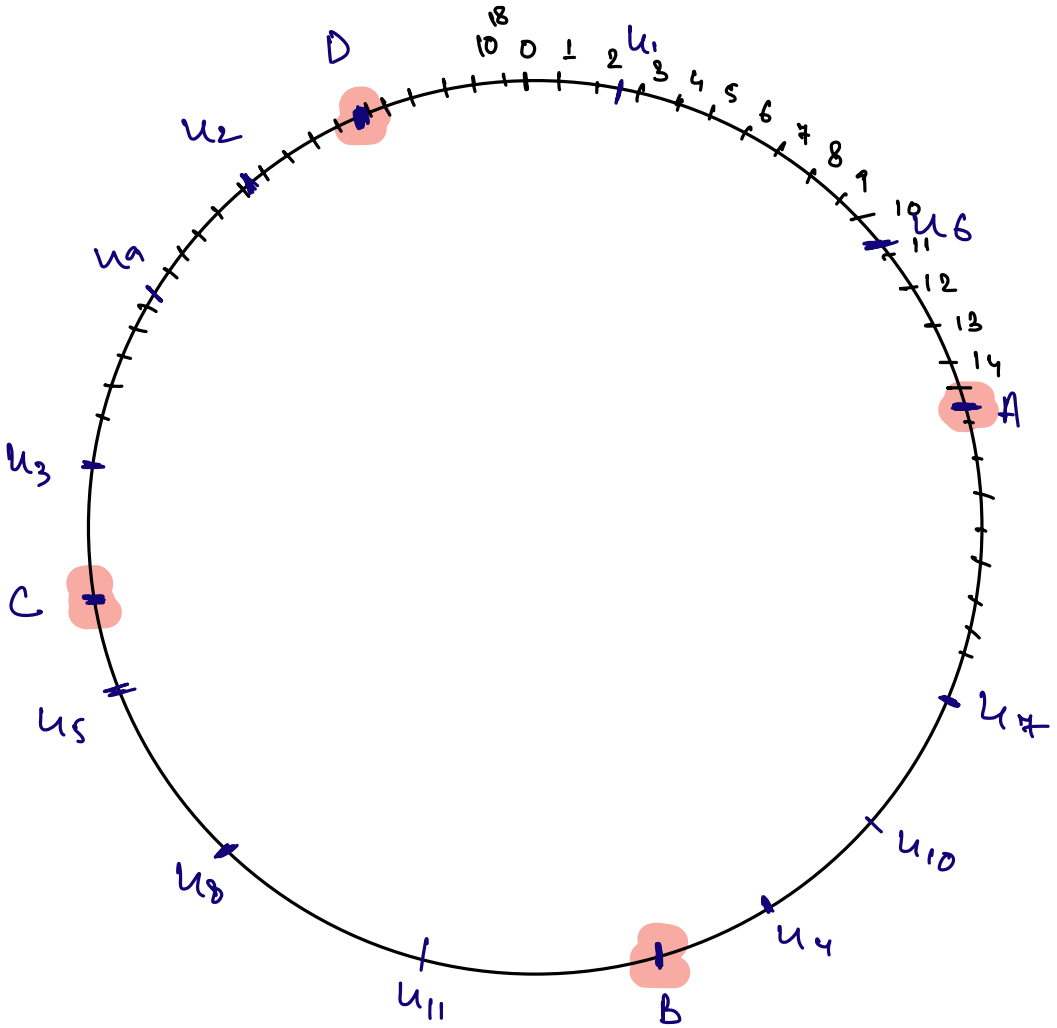
Agenda.

→ Consistent Hashing

→ Caching

Consistent Hashing

⇒ 10^{18}



Logical Ring

4 servers

$H_s(s_1) = A$

$H_s(s_2) = B$

$H_s(s_3) = C$

$H_s(s_4) = D$

↖ server id

$H_s(x) \in [0, 10^{18}]$

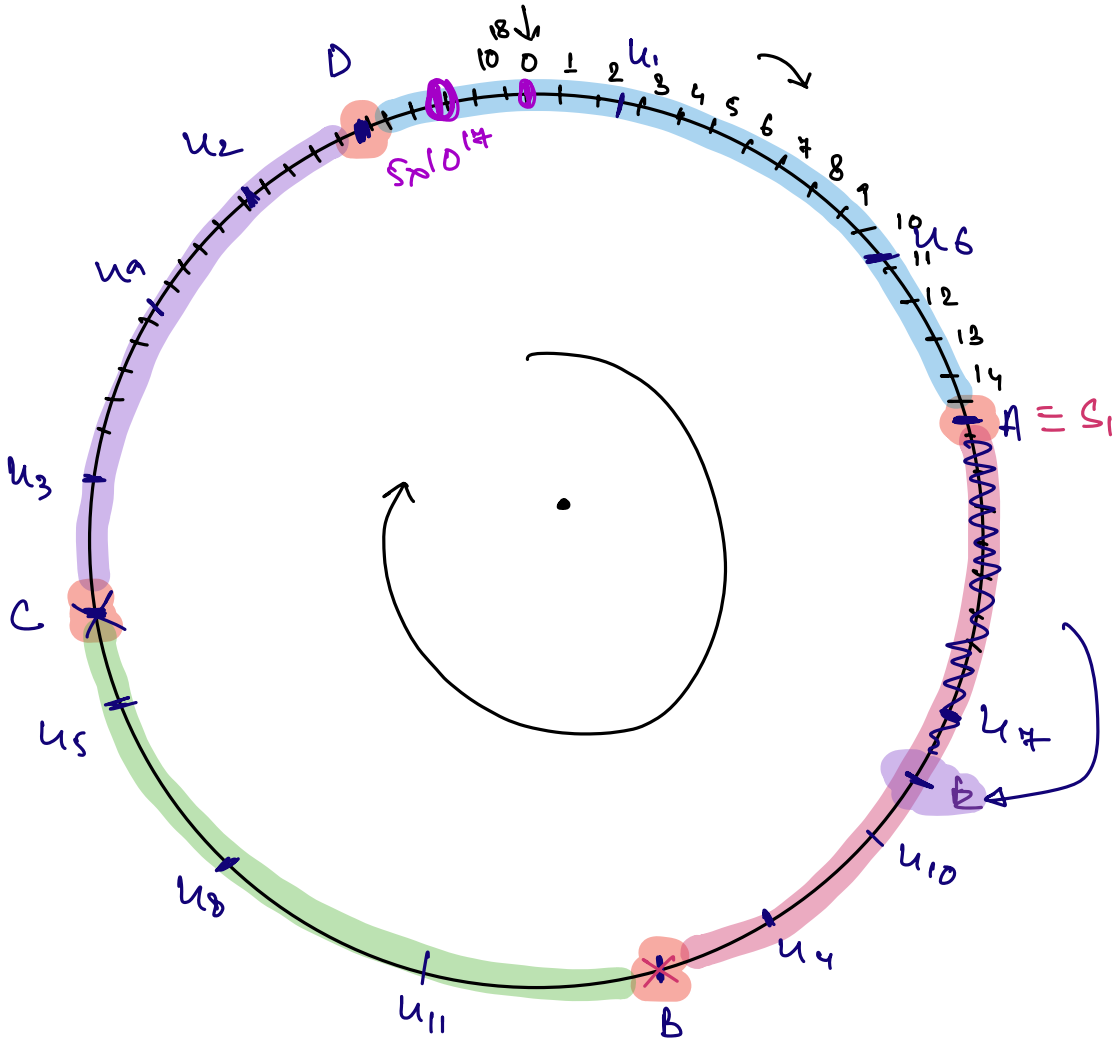
$H_u(u_1) =$

$H_u(u_2)$

$H_u(u_3)$

$H_u(u_4)$

$\equiv$ ← neutral



$A \rightarrow u_1, u_6$

$B \rightarrow u_4, u_{10}, u_7$

$C \rightarrow u_5, u_8, u_{11}$

$D \rightarrow u_2, u_9, u_3$

Implementation

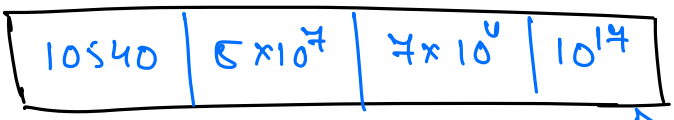
A = 10540

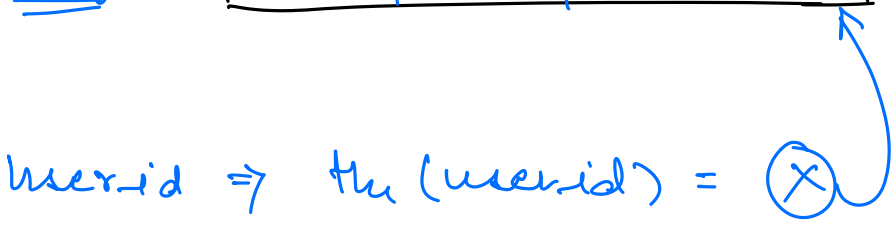
B = 5×10^7

C = 7×10^{11}

D = 10^{17}

Array :-





userid with hash value x will get mapped to the machine with hash value just greater than x.

⇒ O(log N)

If m/c (B) goes down, then its load will be transferred to the next m/c in the clockwise direction on the Ring.

⇒ That means the load on the next m/c will get doubled up.

Scale-Up.

$M_S(S_5) = 2$

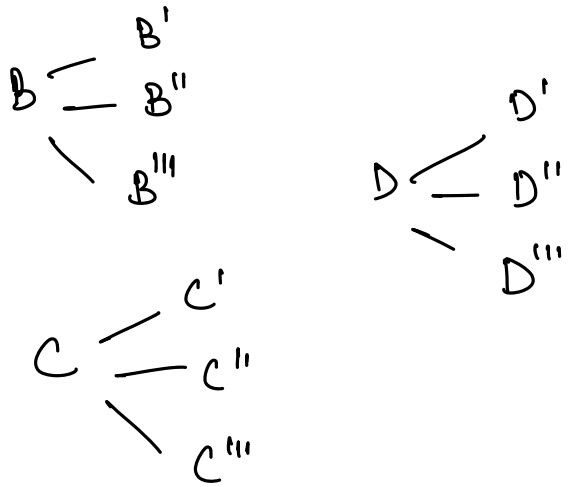
Cascading failure.

⇒ Soln

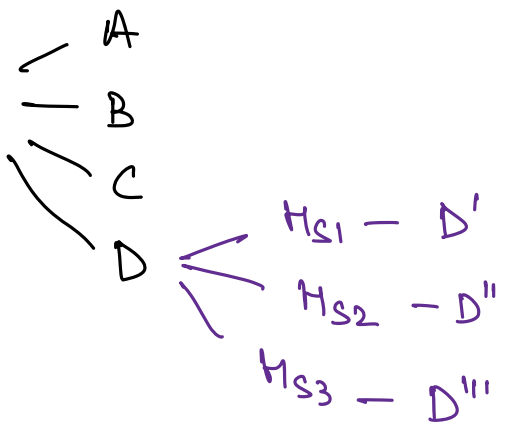
$M_{S1}(A) = A'$

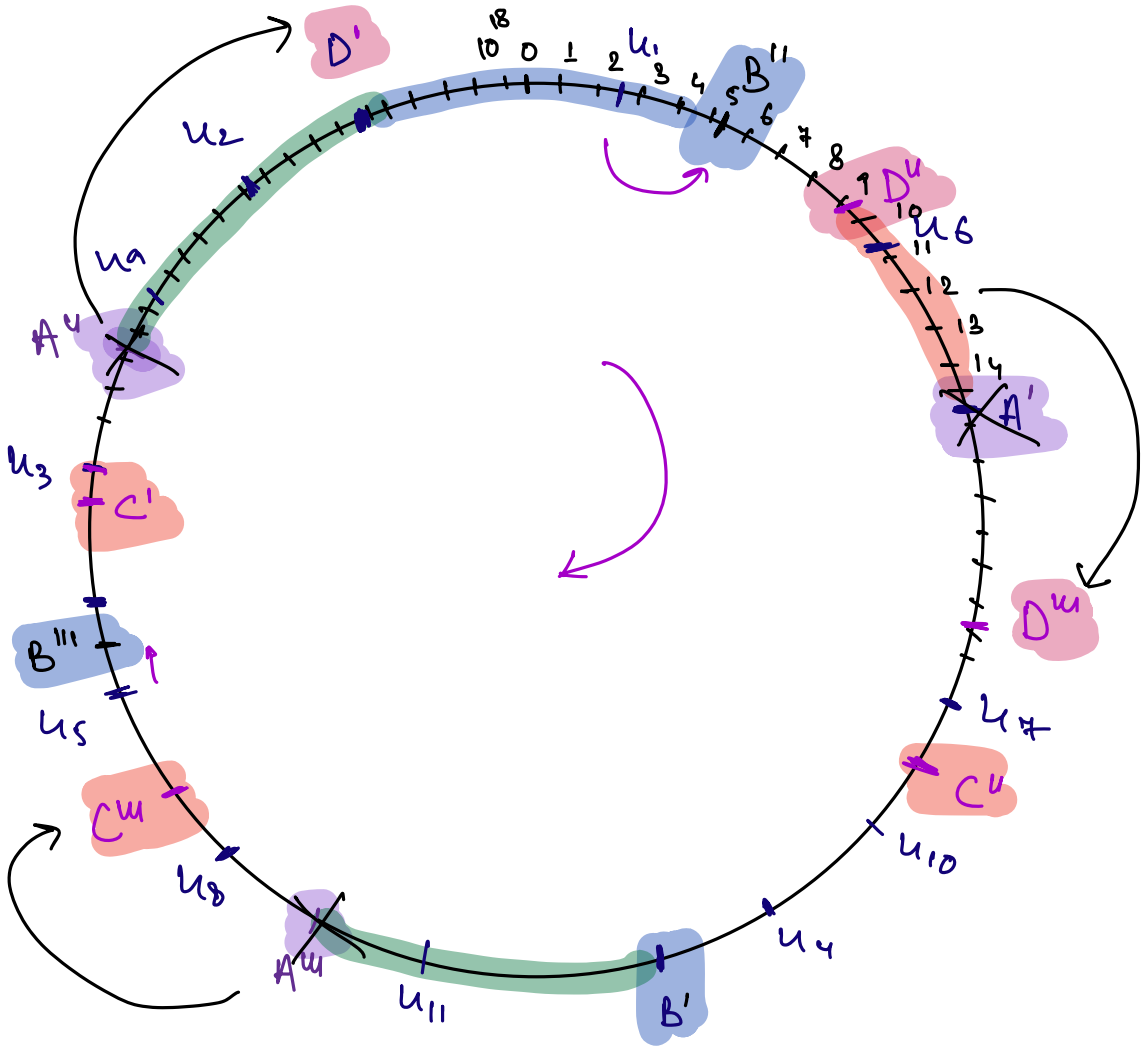
$M_{S2}(A) = A''$

$M_{S3}(A) = A'''$



⇒ 4 m/c





$u_1 \rightarrow$ (B)

$u_{11}, u_6 \rightarrow$ (A)

$u_5 \rightarrow$ (B)

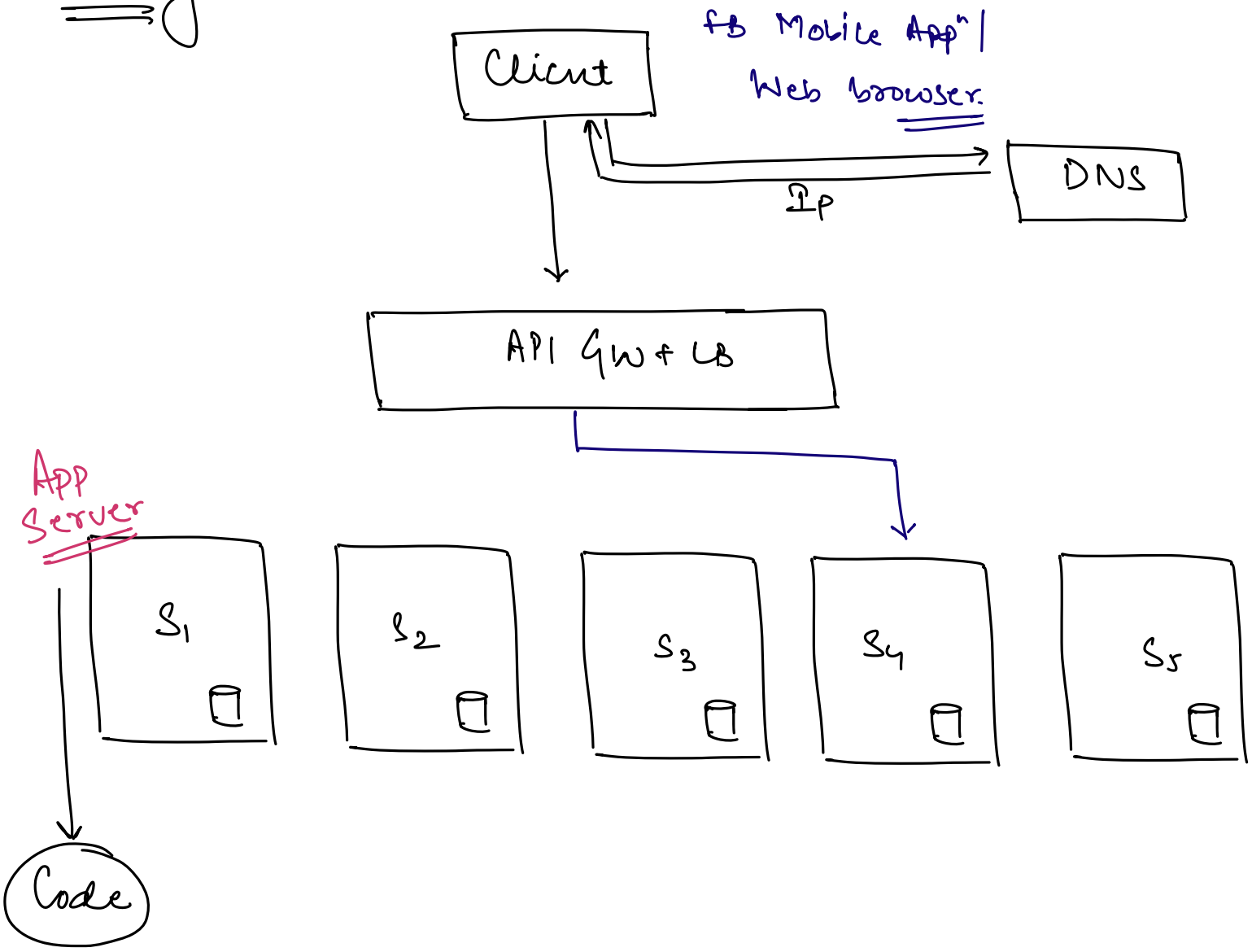
$\Rightarrow$ Cascading failures ↓

RBST

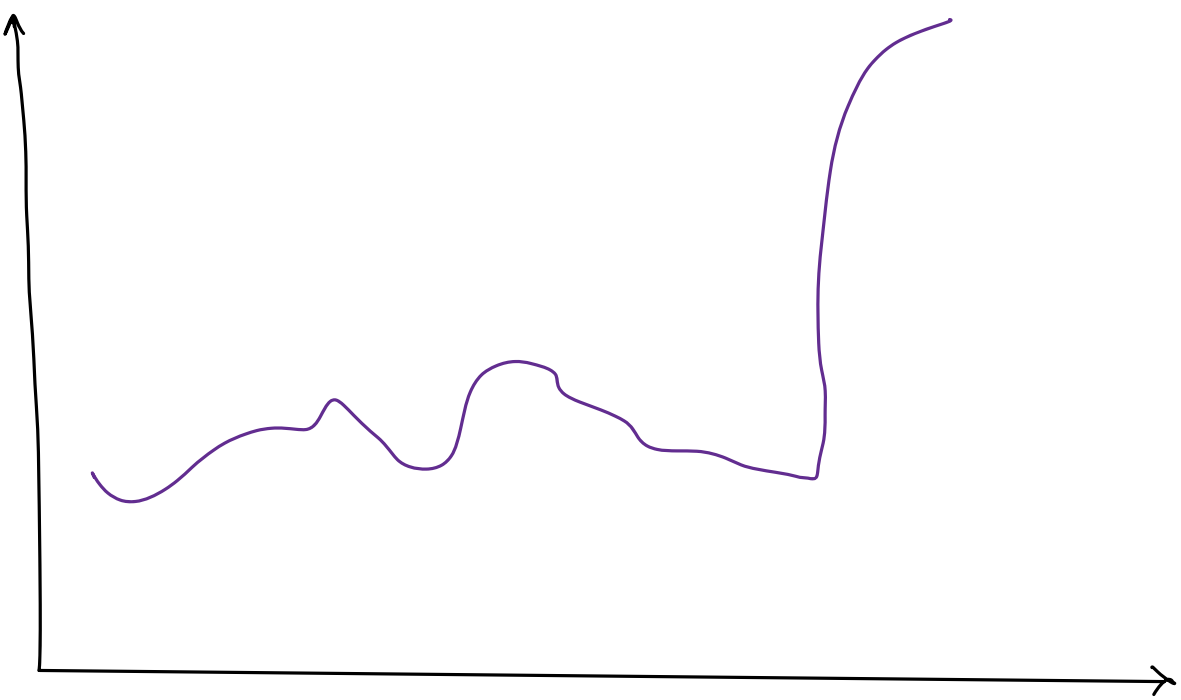
↓

Stateless

Caching



Hotstar : 2019 Semi-final (Ind vs NZ)



⇒ Ideally, App Server layer & Storage layer should be decoupled from each other.

I) Continous deployments will make App servers restart again & again but we need not to restart DB frequently.

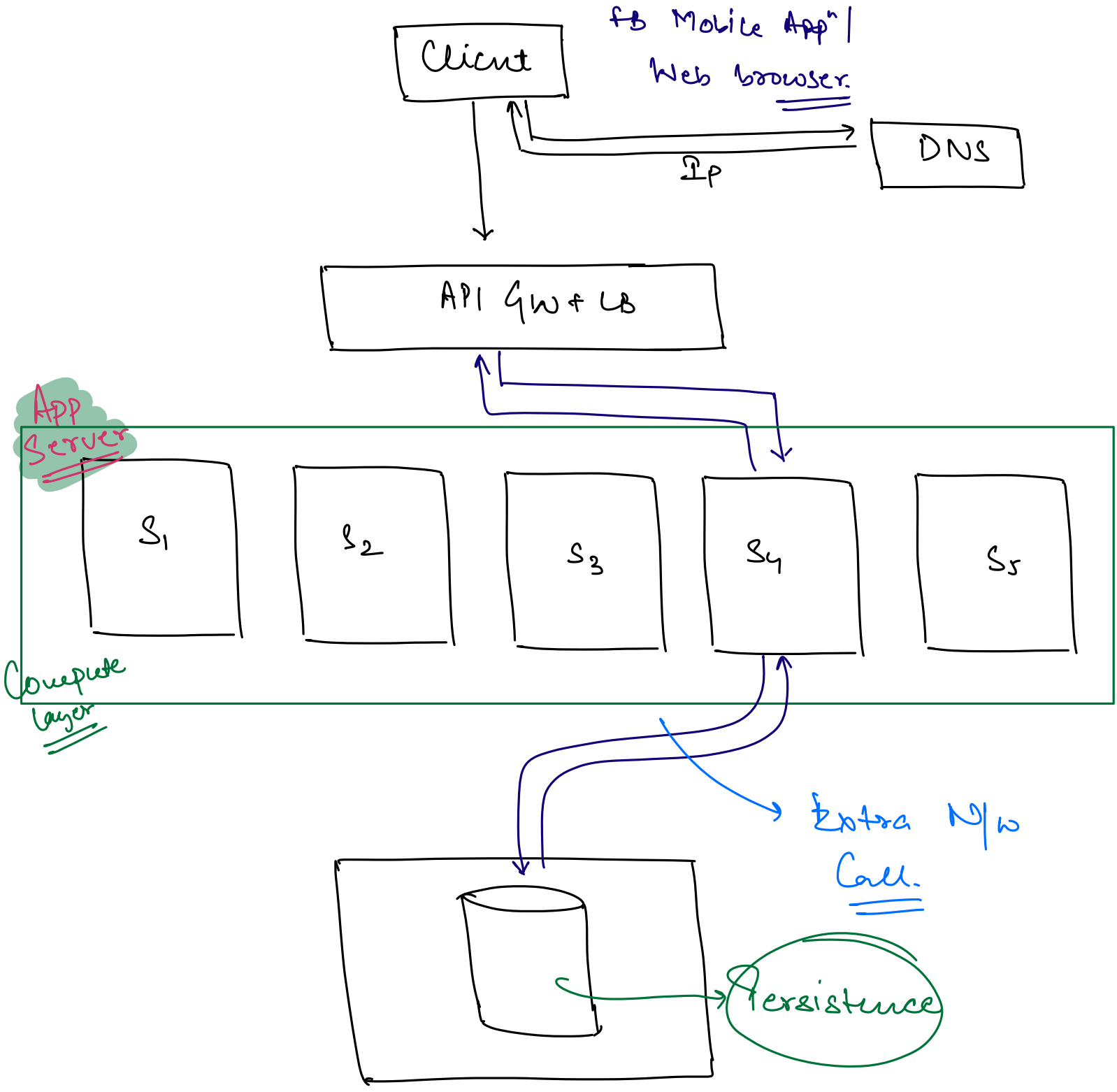
II) Scaling up App server would also lead to addition of more Storage even if it is not required.

App Server ≡ Business Logic

↳ Good CPU + RAM.

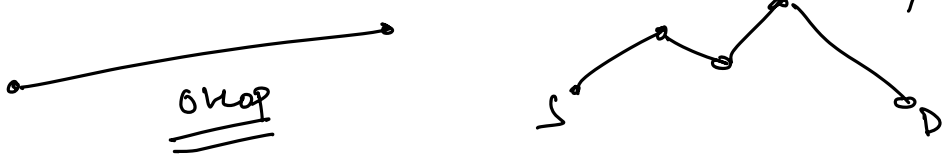
DB Server ≡ More Space

↳ HDD/SSD. → less CPU + RAM



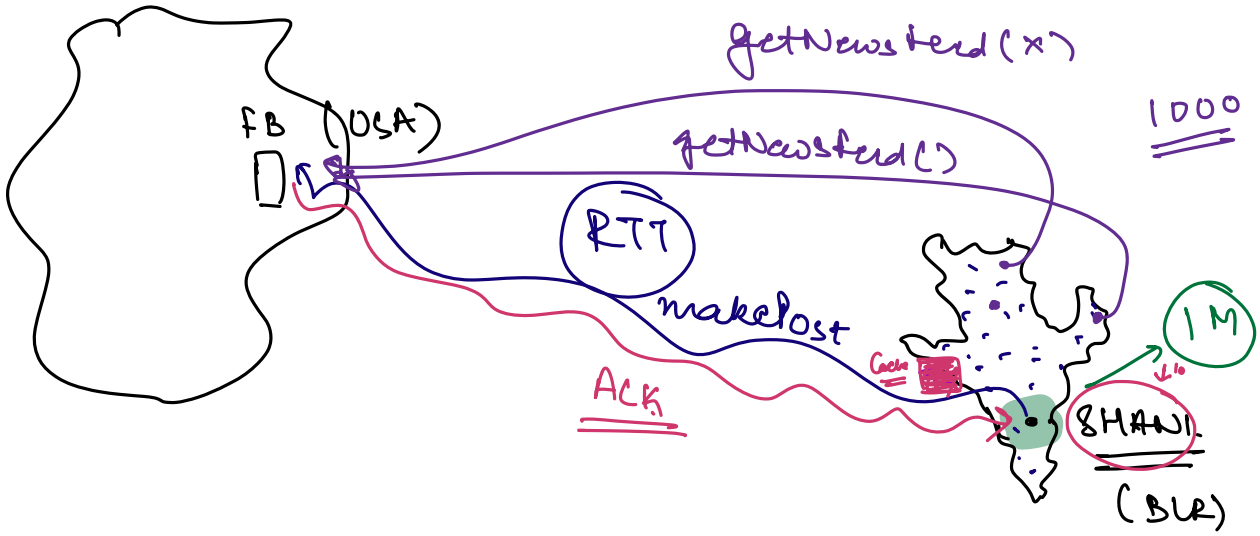
RTT : Round trip time.

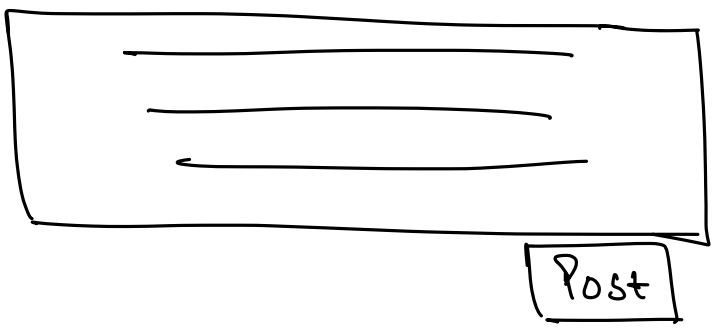
⇒ Trade-off.

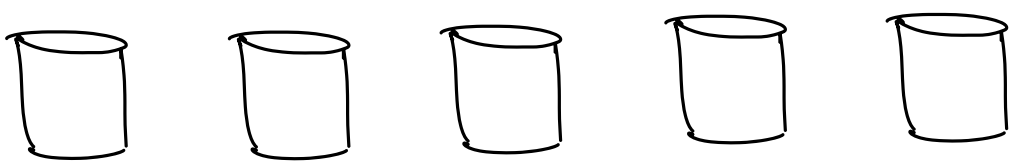


Latency ∝ N/w Hops.

FB.







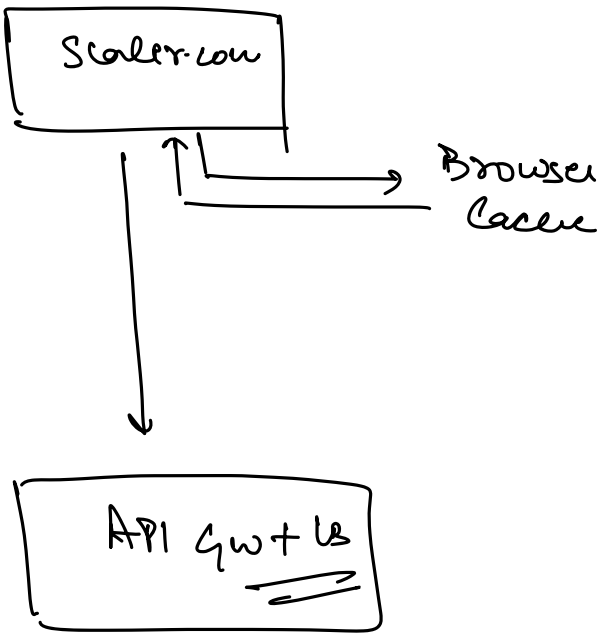
Latency ∝ Distance

Cache ⇒ Volatile

Caching is a process of storing copy of some frequently accessed data near to the end user, so that latency can be improved.

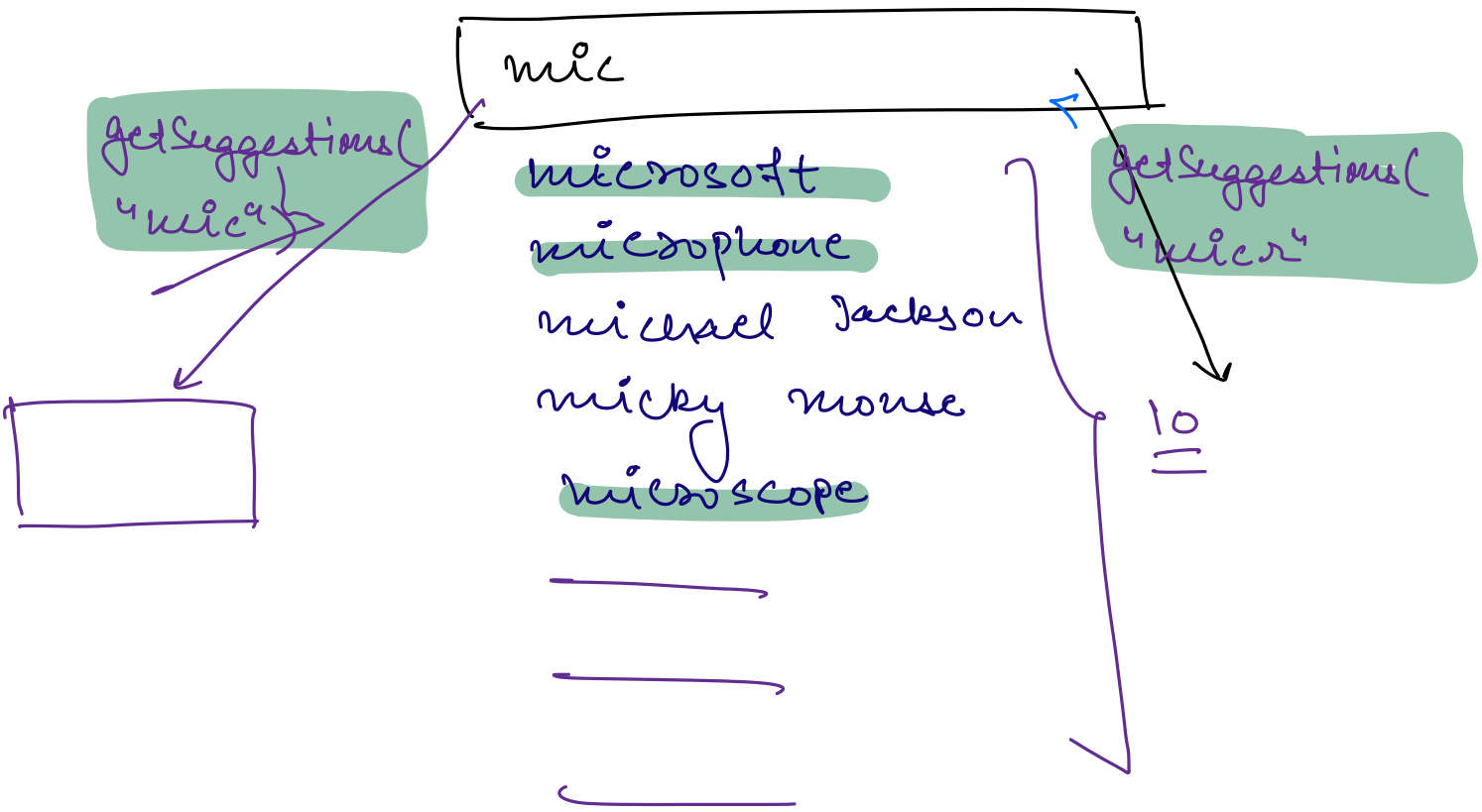
Cache is very expensive.

Client Side | Browser | App Side

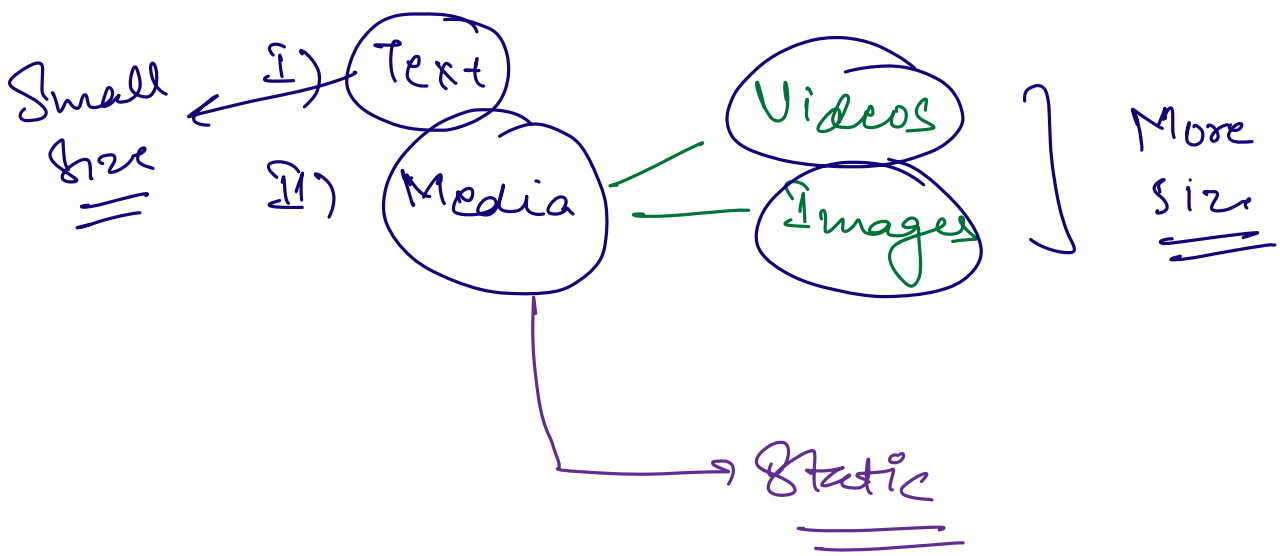


Google Search.

Type ahead Search



⇒ Types of data we access over the n/w.



CDN.

↳ Content Delivery Network ⇒ Next Class

$0 - 10^{18}$

10k

$S_1 \Rightarrow 10^3$

$S_2 \Rightarrow 10^6$

$S_3 \Rightarrow$

$S_4 \Rightarrow$

$S_5 \rightarrow$