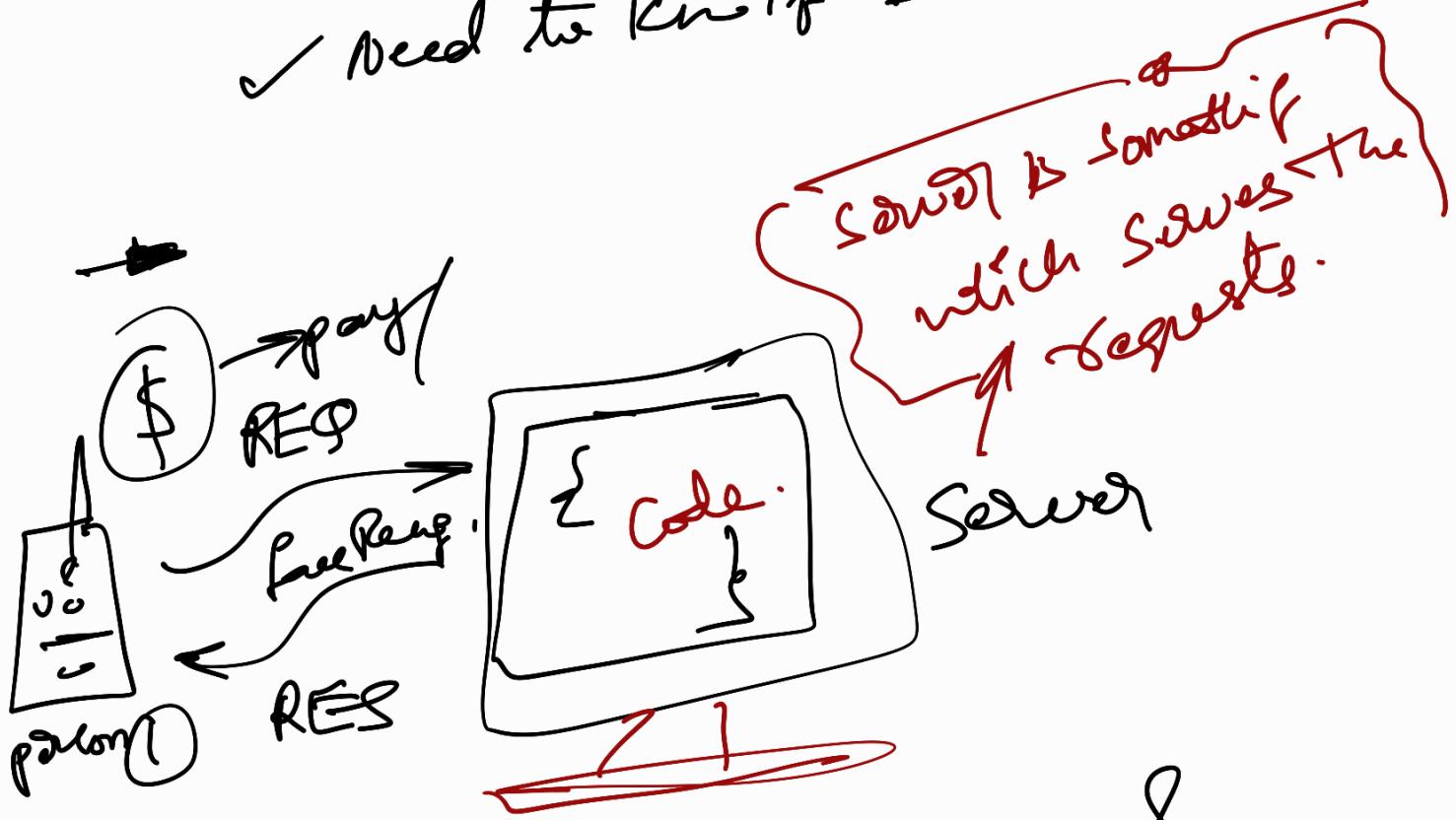


3. What is load balancing?

Consistent Hashing →

✓ Need to know if building Systems -



if more clients increased.

$P_2, P_3, P_4, \dots, P_{100}$.

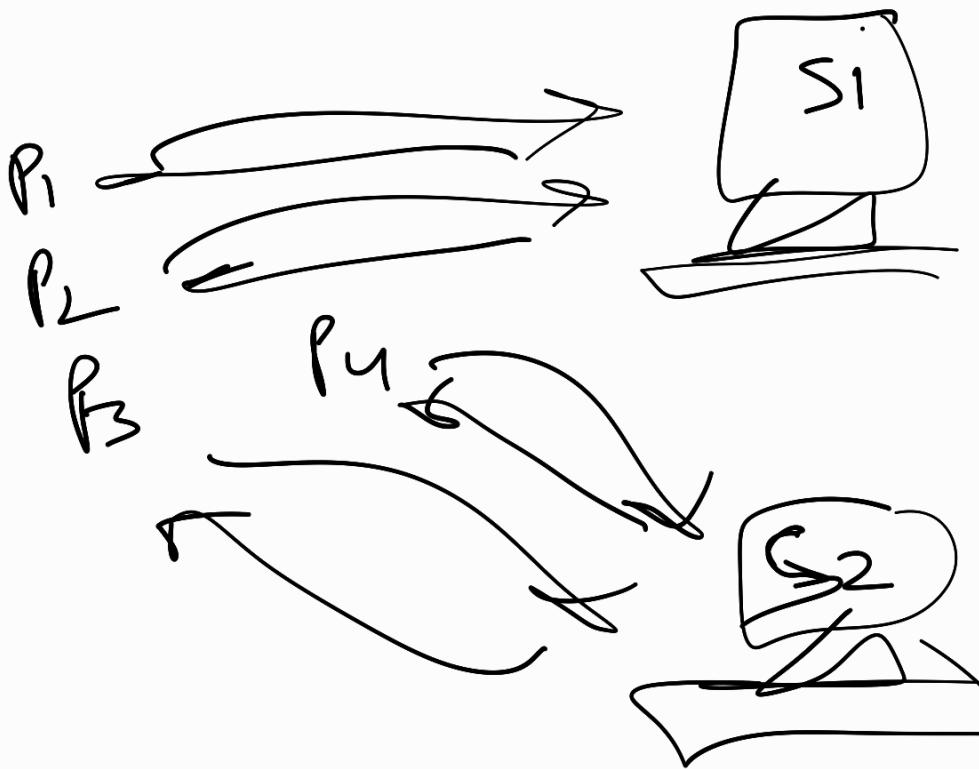
server can't handle.



Buy new computer server.



problem → where do you send
Req's → $S_1 \rightarrow S_2$

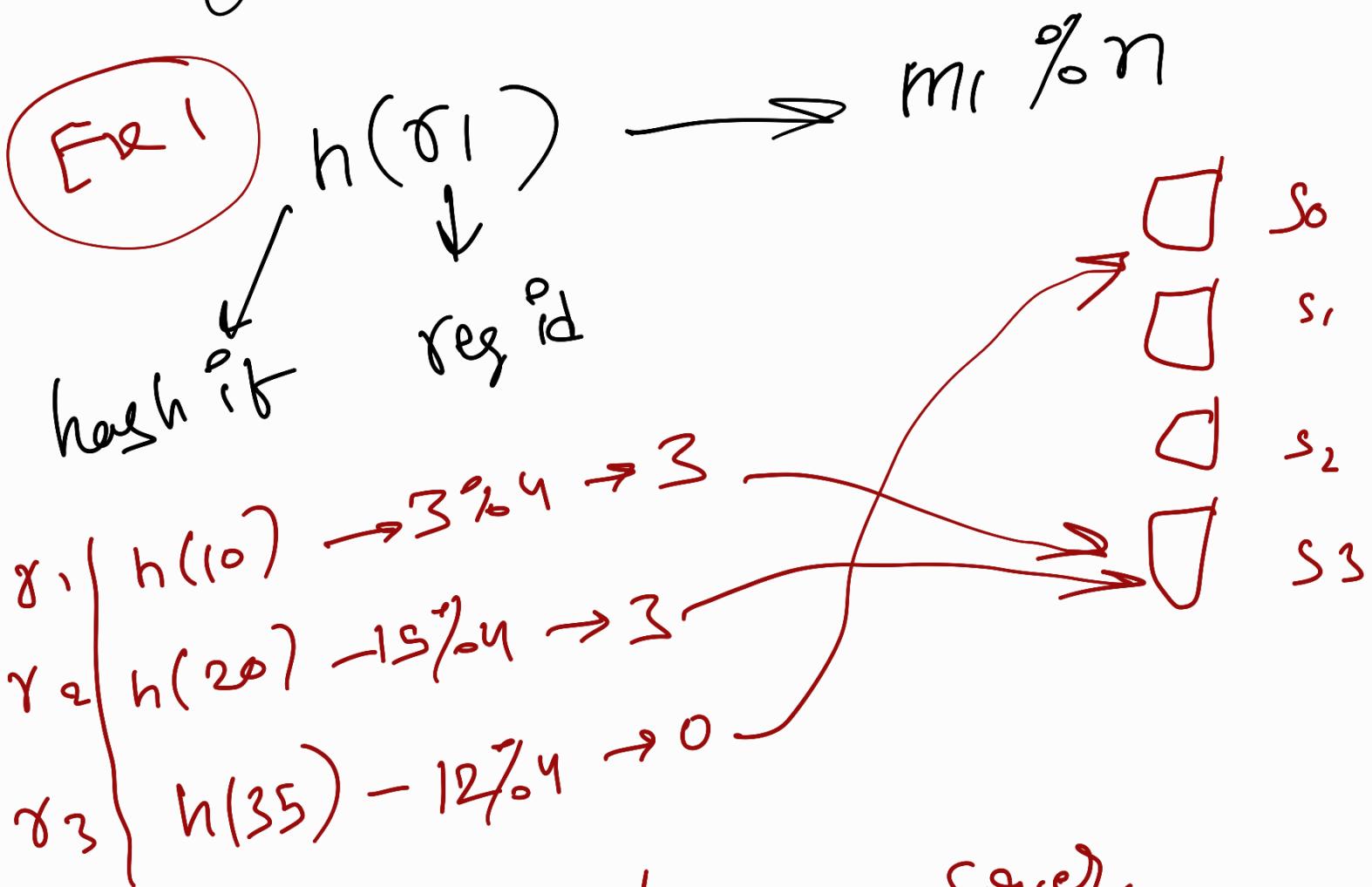


- if N servers \rightarrow to balance the load on free servers
- ✓ yes's are need to process.
- ✓ servers has load.
- ✓ Concept of taking N-servers & trying to balance the load on all of them \rightarrow is called load balancing.

✓ To achieve this concept of Concept Hashing is required.

Reg-ID \rightarrow (Uniform Random Number)

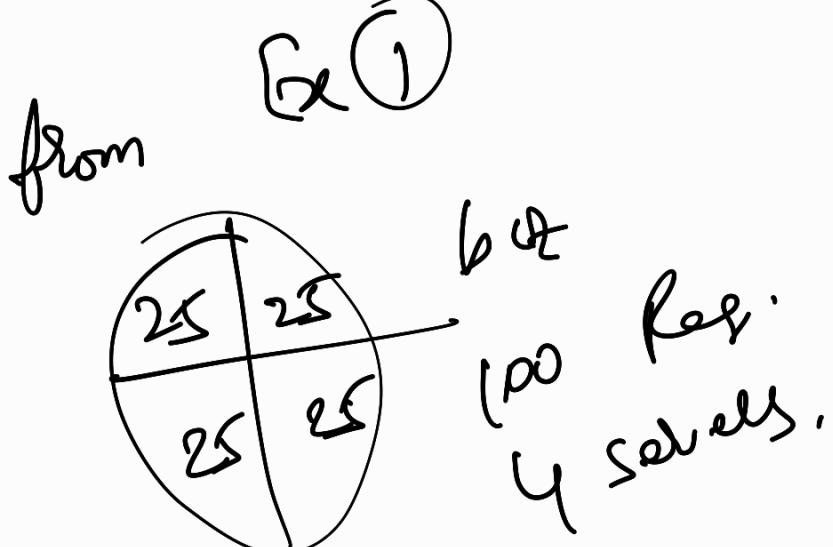
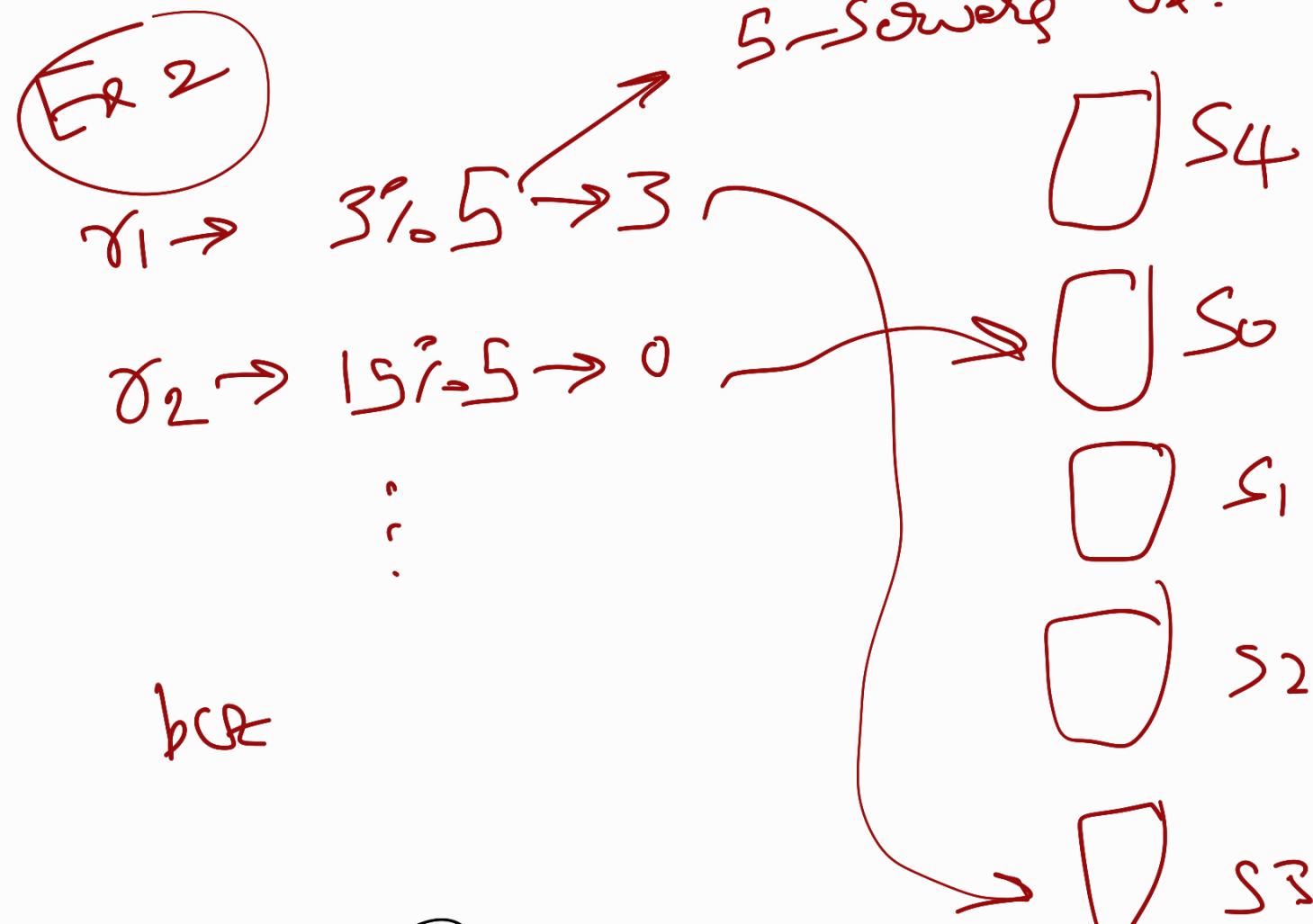
(b) M servers $0 \rightarrow M-1$



$f_1 \rightarrow$ maps to s_3 server.
 If all servers to uniform load.

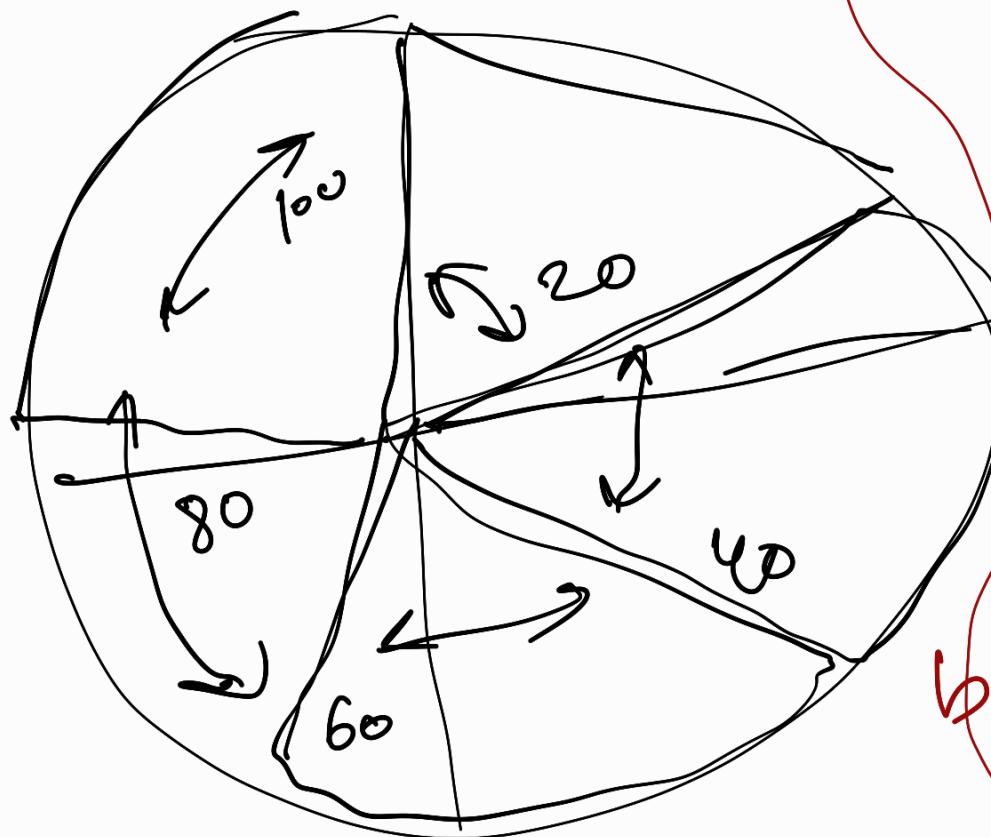
$X/n \rightarrow Y_n \rightarrow$ load factor.

if Resutly are huge \rightarrow modify current.



f3)

Ex (2).



Entire system chose
+ Cache dump no use
but server chose

Cost of the operation if 1-node
increases with Rep id \rightarrow

$$5+5+(0+10+15+15+20+20 = 100)$$

Cost of operation

↑
Entire Search space.

req_id is never be random

→ usually unique

→ generates $\text{UserID}(\text{Client})$
for ex USERID .
 $\rightarrow \text{UserID}$

if $h(\text{nagendra}) \rightarrow h(u_1)$

hash of "name" → all times
same

$h(\text{gaurav}) \rightarrow h(u_2)$
 $\rightarrow \text{USERID}$

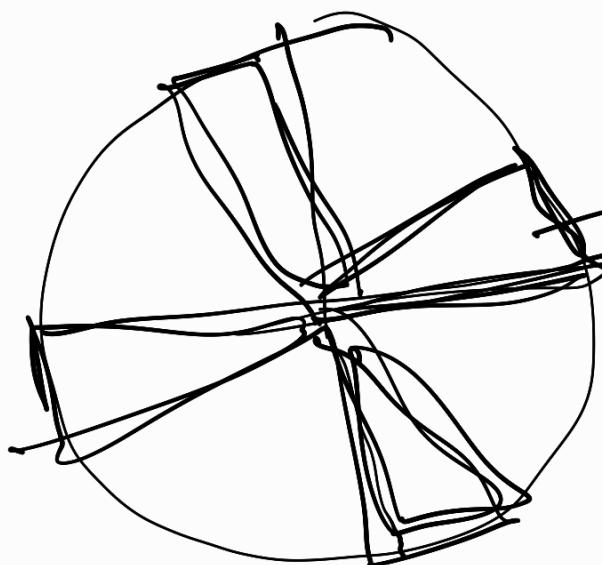
hash of "gaurav" all times same.

if hash is same → remainder
Same

→ it sends info to same
server again & again.
✓ use or use this logic.

- if Req is to same Server or profile fetch \rightarrow can keep in Cache & can fetch fast.
- ✓ if new Server every time diff hit will same Server \rightarrow Cache

Depending on User ID \rightarrow we can send Req to same Server \rightarrow we can store relevant info of user in the same Server Cache.



Sum of 4 small areas is 20%
Sum of 4 small areas is 20%
Sum of 4 small areas is 20%

Overall char. should
be min.

i.e Advanced approaches

↓
Consistent Hashing

~~= & ≠~~