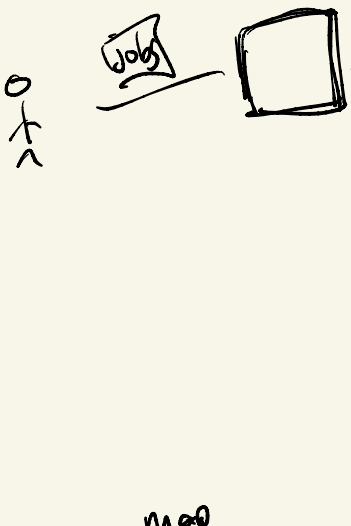
Massively parallel framework (MapReduce / Liteminer)

Speed up some computation by distributing work across servers



Heartbeat

progress report

failures / leave the cluster

slow workers

incorrect output

quarantine

restart the task

hedging

redistribute

proportional

map

reduce

filter

shard 1

filter

shard 2

aggregate

find subset of shard that meets a criteria

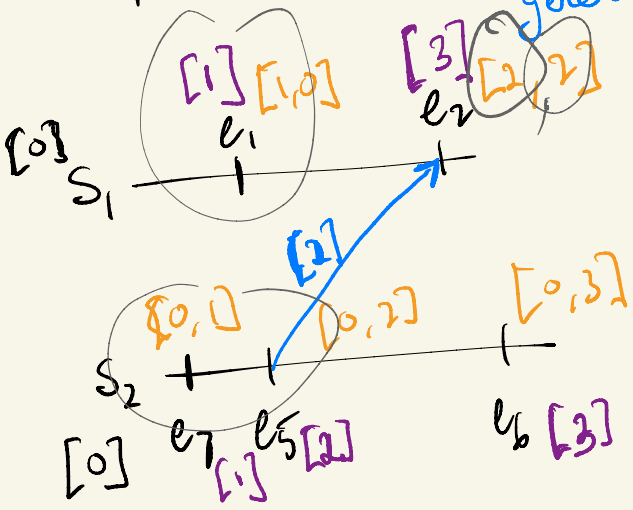
Shards are not all the same size

Data imbalance

Time & consistency

Total order : all servers are able to impose the same ordering on all events

FIFO : ordering is relative to the server which generates the event



S_1 : e_1 e_2 e_5 e_6

S_2 : e_1 e_2 e_5 e_6

} Total

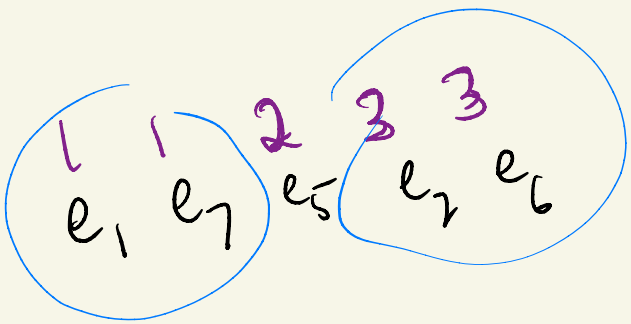
S_1 : e_6 e_5 e_1 e_2

S_2 : e_5 e_6 e_1 e_2

} Causal

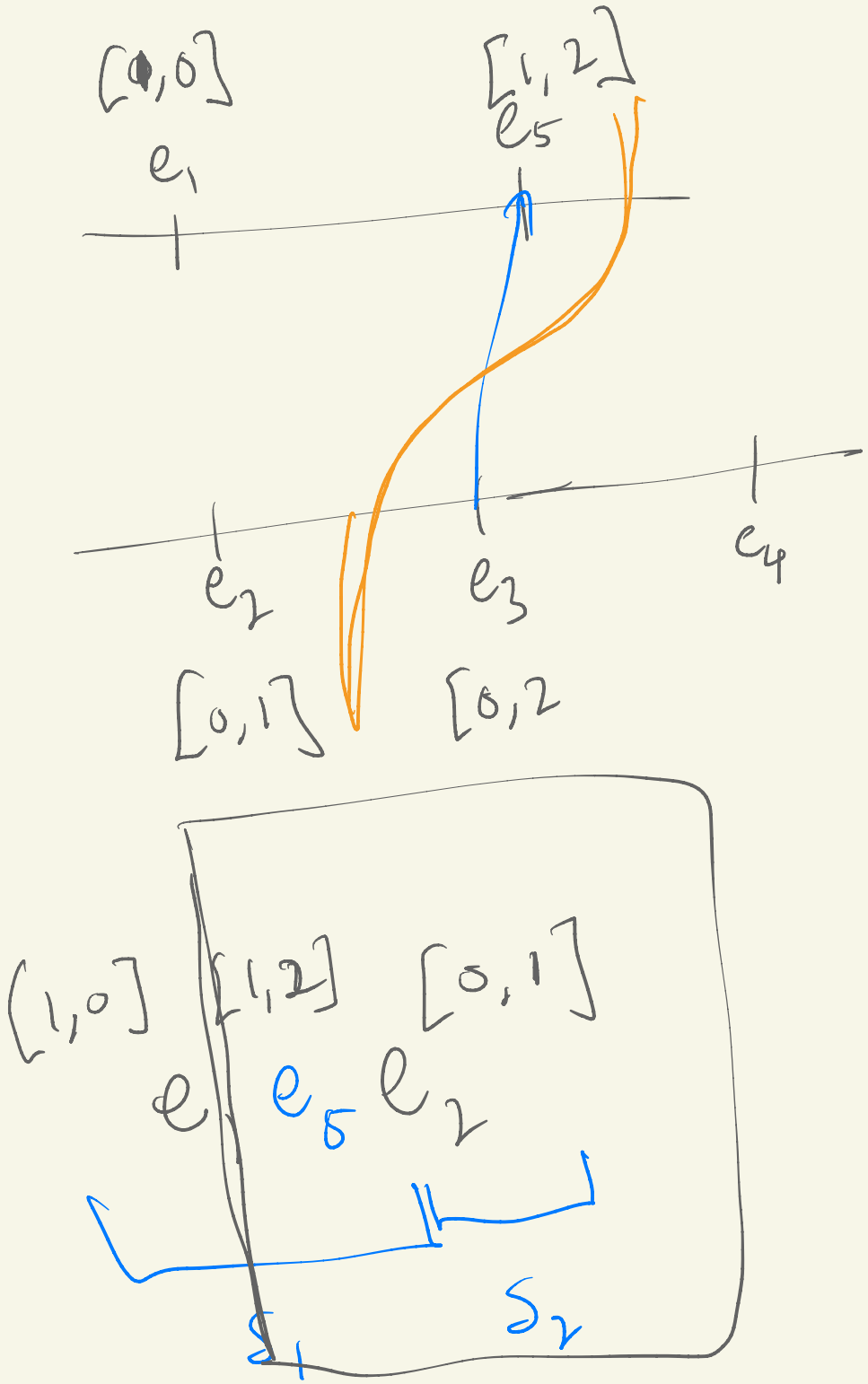
events
- send, local — [for each event increment clock]
- recv — max(msg.clock, local clock) ++

e_1 e_7 e_5 e_2 e_6



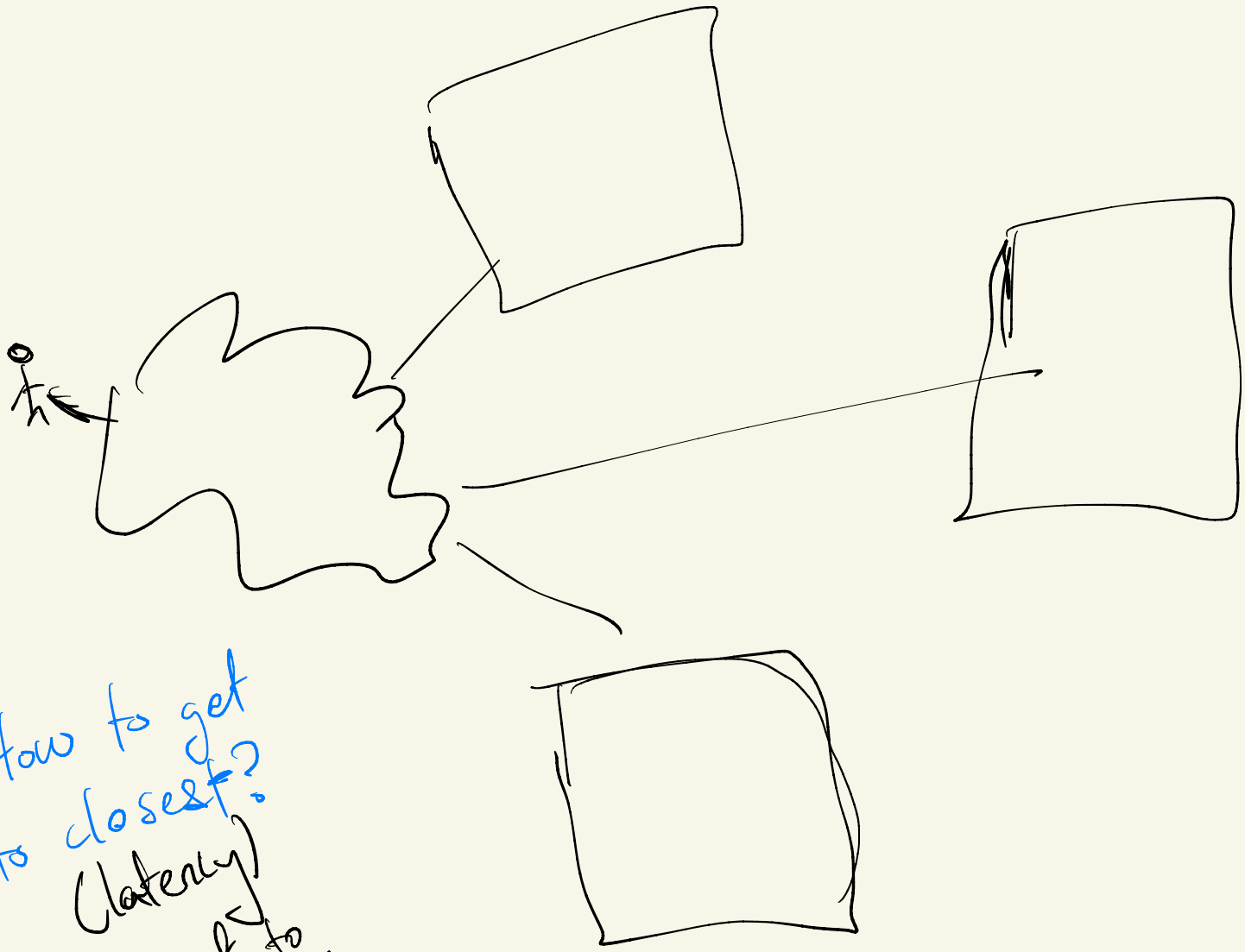
S1 e_1 e_7 e_5 e_2 e_6

S2 e_7 e_1 e_5 e_2 e_6


[0,0]
e1
[1,2]
e5
e2
e3
e4
[0,1]
[0,2
[1,0]
[1,2]
[0,1]
e1
e5
e2
S1
S2

Global

Local

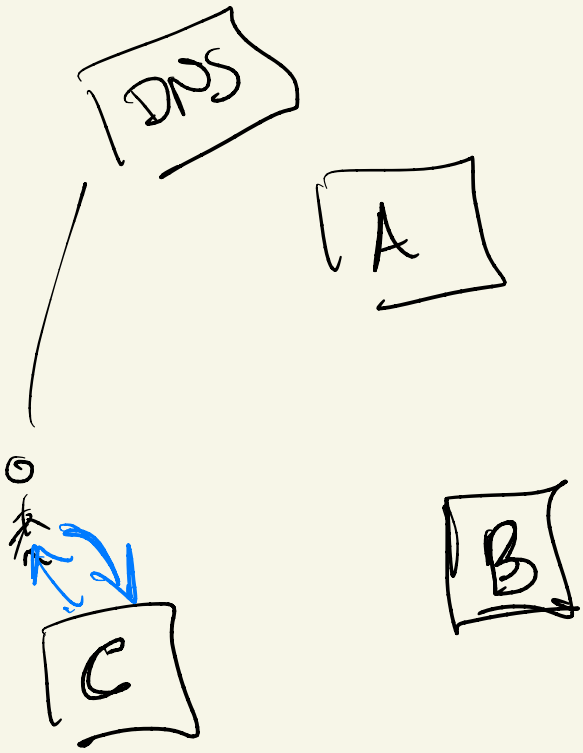


How to get to closest?

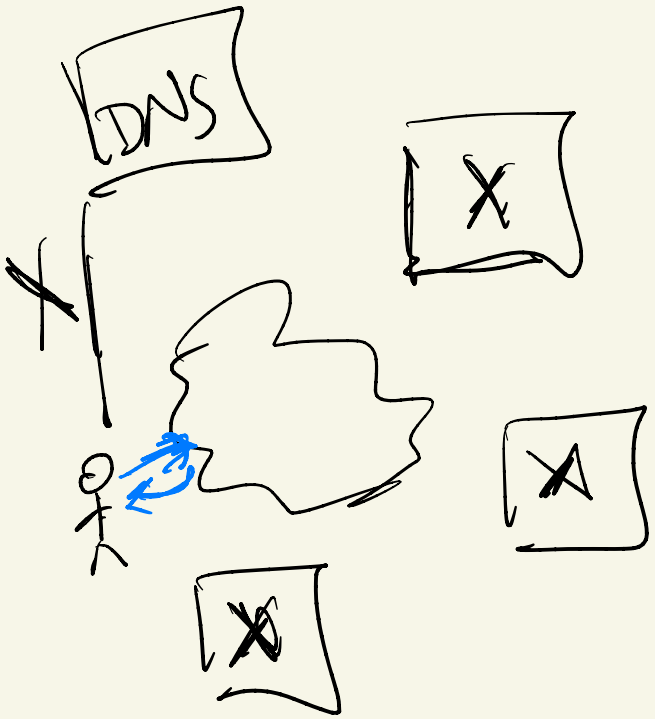
(Latency)

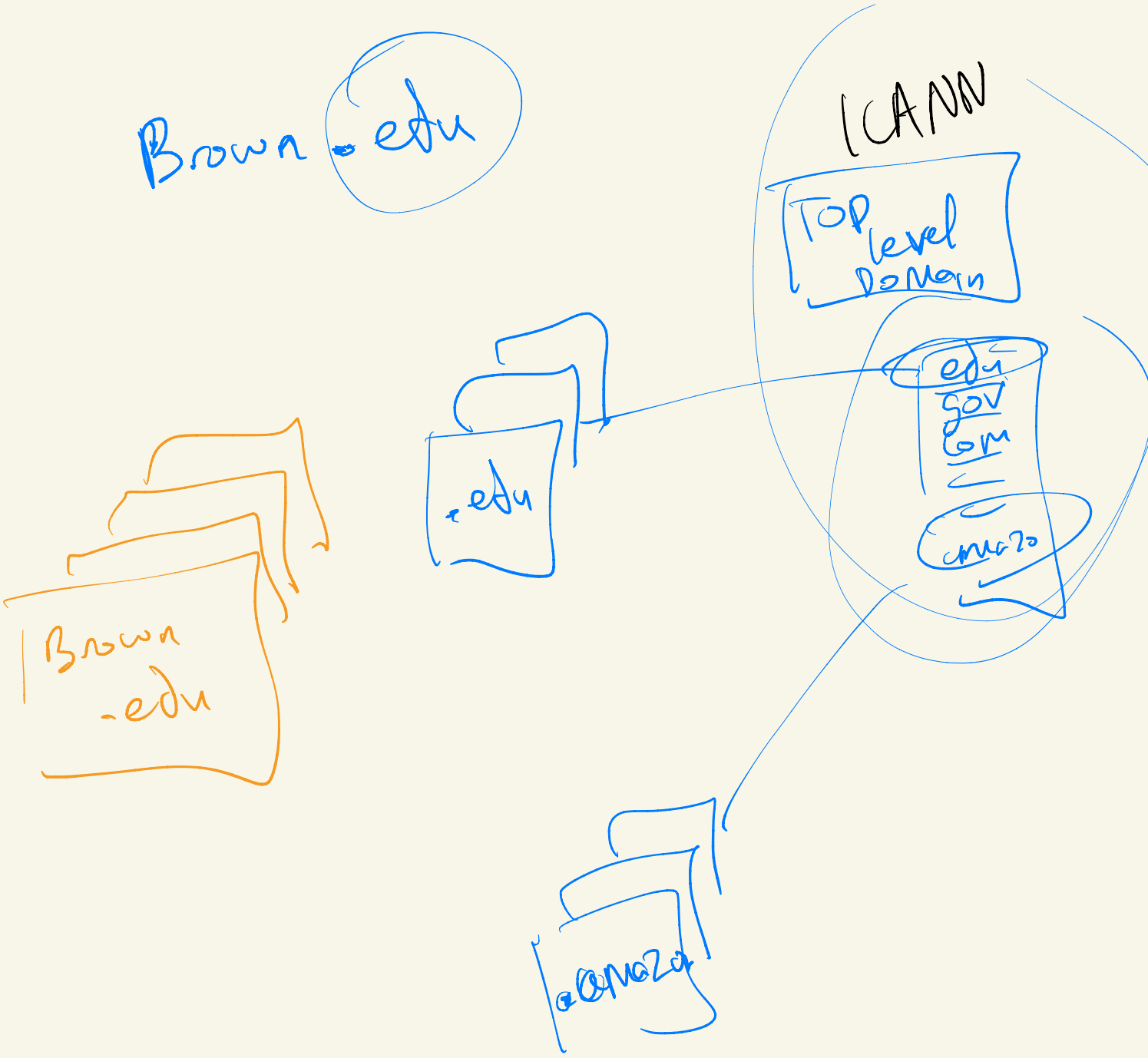
How to get to a policy compliant clusters.

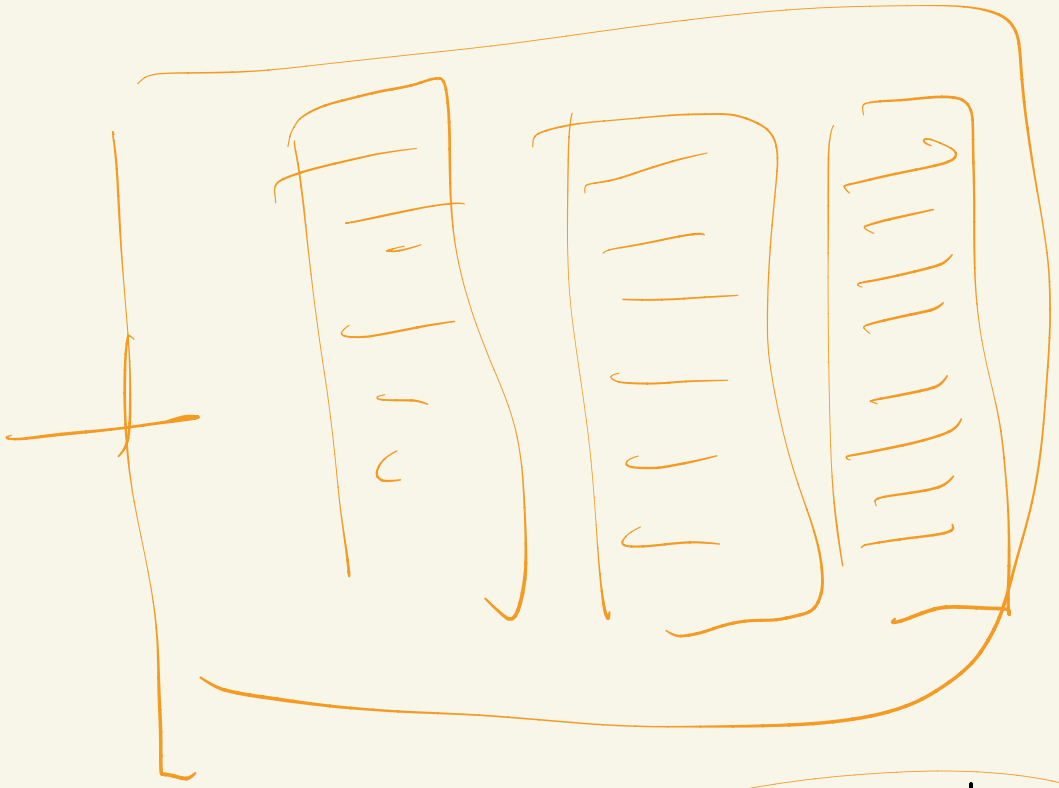
Pure DNS



ANY CAST




Brown.edu
ICANN
Top level Domain
edu
gov
com
.edu
Brown
.edu
.com



main objective = even utilization across servers

randomly spreads clients to servers

+

Consistently

+ want a user to consistently go to same server

Global LB

* purely DNS / ANYCAST
  - ↳ closest cluster (reduce latency)
  - or
  - some corporate policy

Local LB

* Consistent hashing
  - ↳ even util
  - ↳ consistently map client to server

Assumptions

msgs are FIFO

no msgs are ever dropped

no server ever fails

only one snapshot at a time

FIFO

easy

C L

@ initiating server

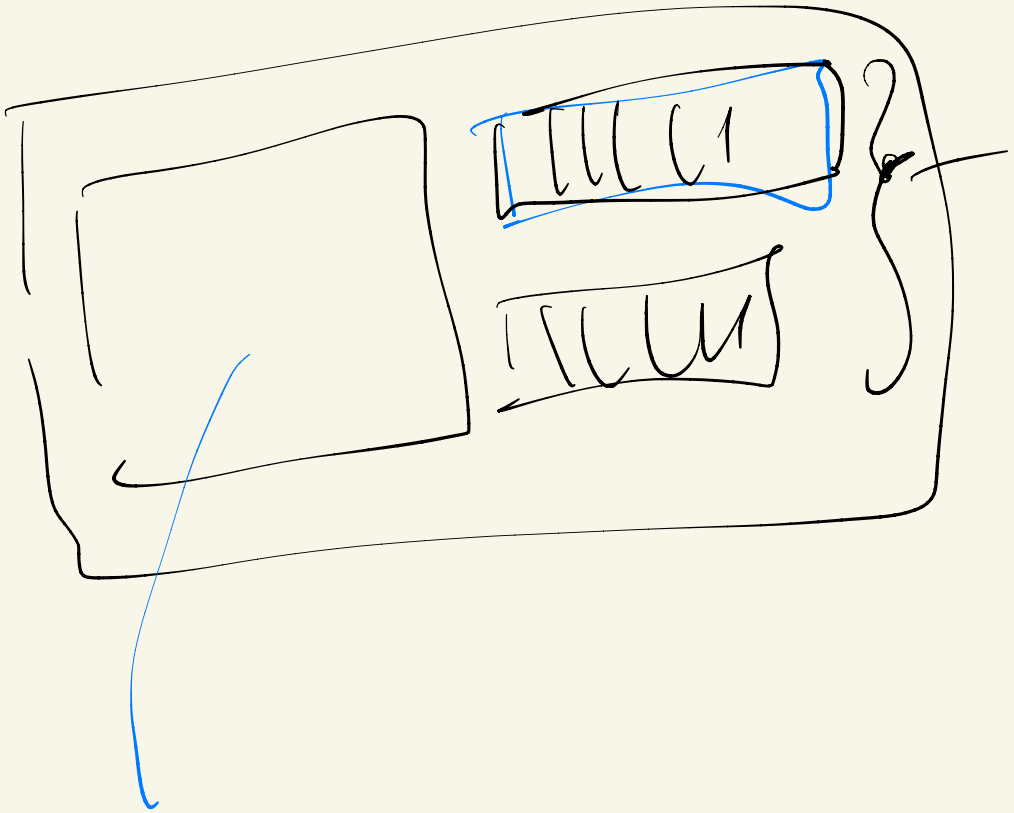
takes a checkpoint of memory

Sends out markers to every server in system

and wait for everyone else to send a marker

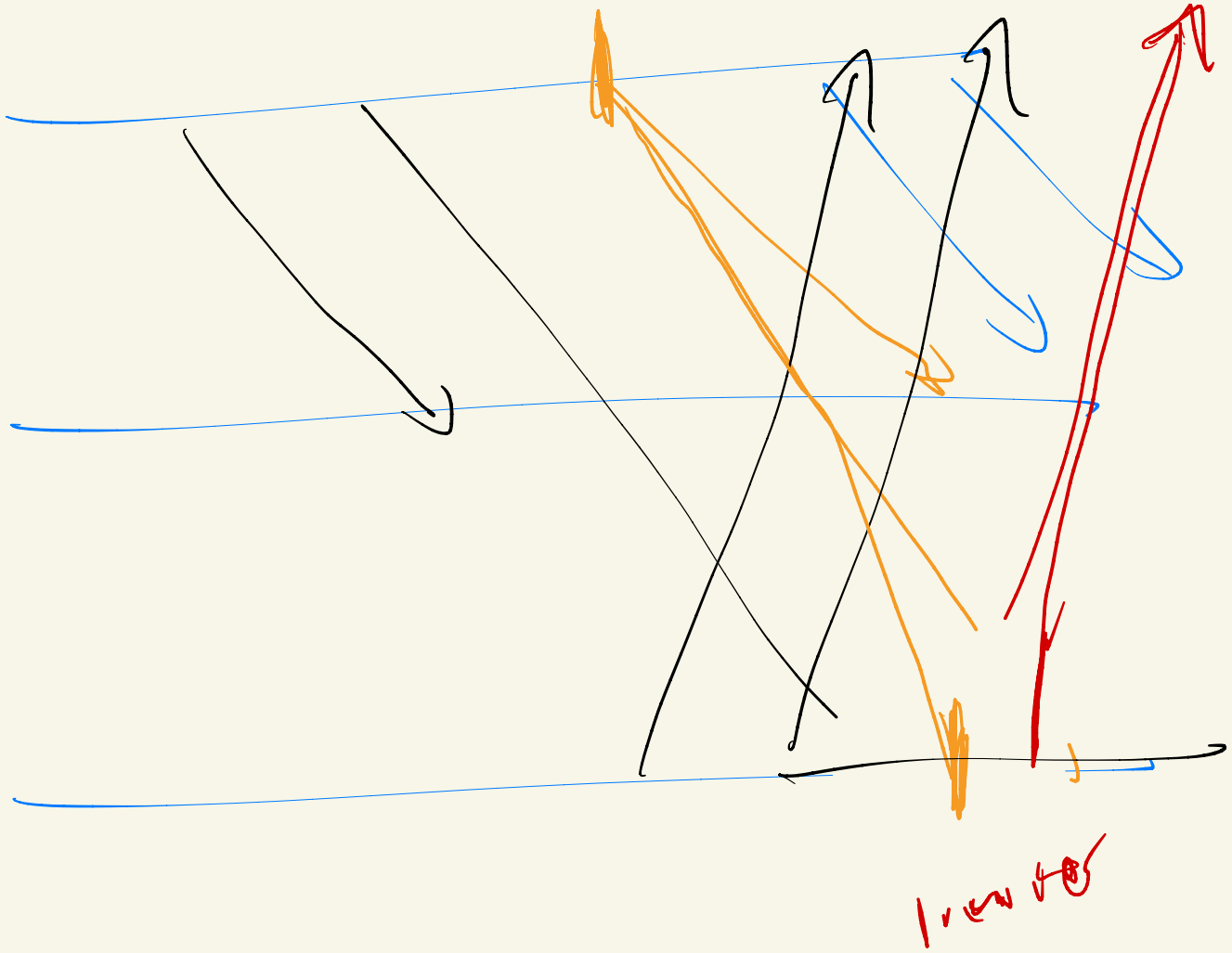
a queue for each server

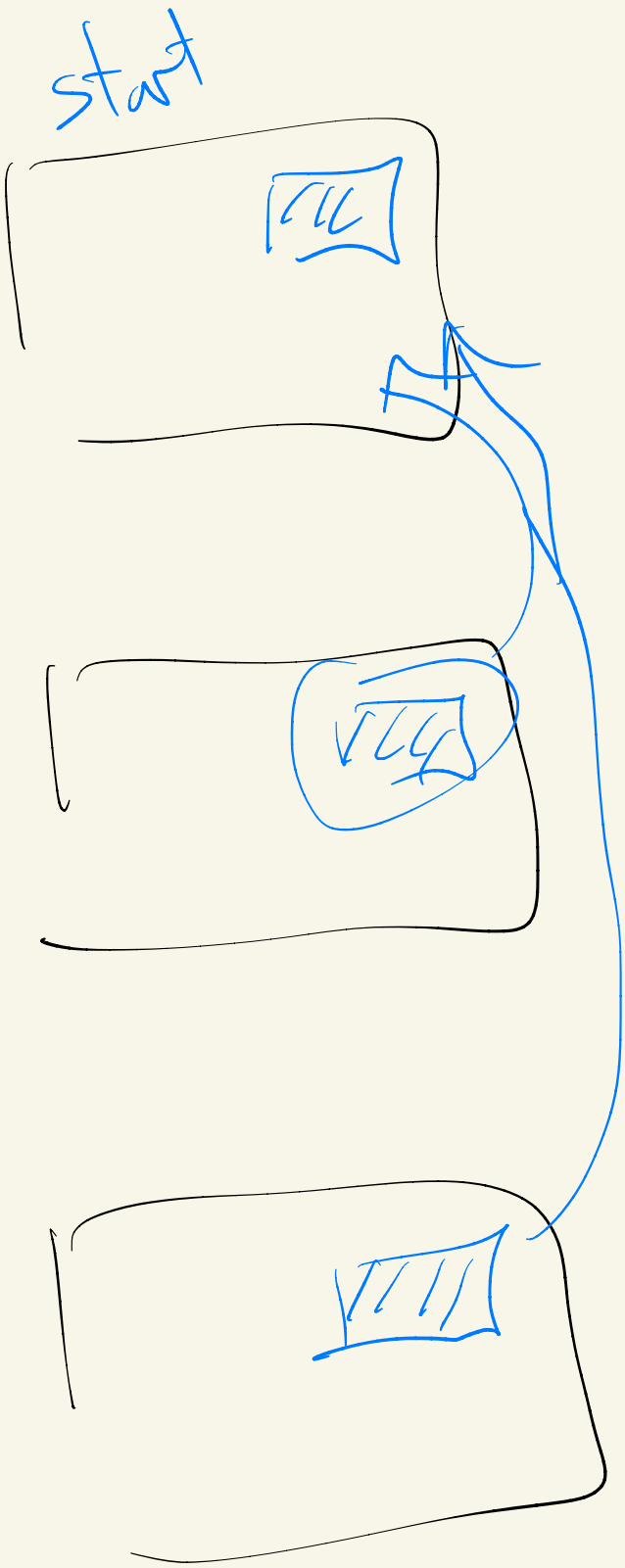
every msg from server goes in queue until you get a marker from the server



n-1 servers

memory checkpoint




start
III
IIII
VIII

FIFO
- logical clock
- Vector clock

Total : FIFO + Tie Breaker