

CAP → the most important topic

↳ what can we do, and what **can't we do**

PACELC → extension to CAP

CAP

Master - Slave Replication

CAP Theorem

Atomicity
Consistency
Isolation
Durability

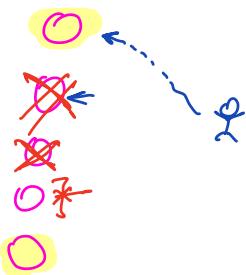
Consistency → there should be **no stale reads**
(Not the ACID consistency) from the client's perspective, there is just 1 copy of the data

No matter what nodes are down, whenever the client reads, they should get most recent data

$a = 10 \rightarrow 20 \leftarrow$
 $a = 10 \leftarrow \times$
 $a = 10 \rightarrow 20 \leftarrow$

Availability → no matter what nodes are down, if client sends request to a working node, they should get a valid response.

It is okay for the response to use old/stale data



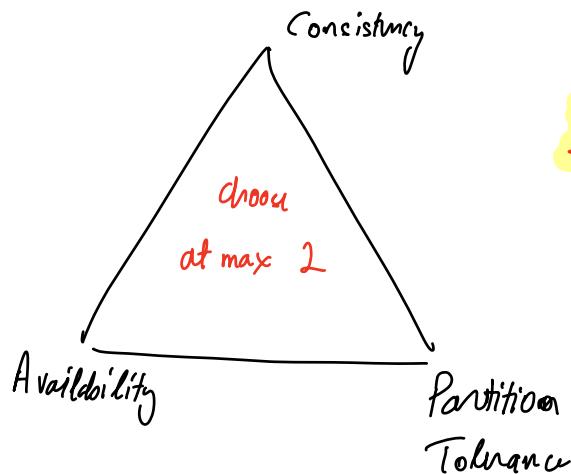
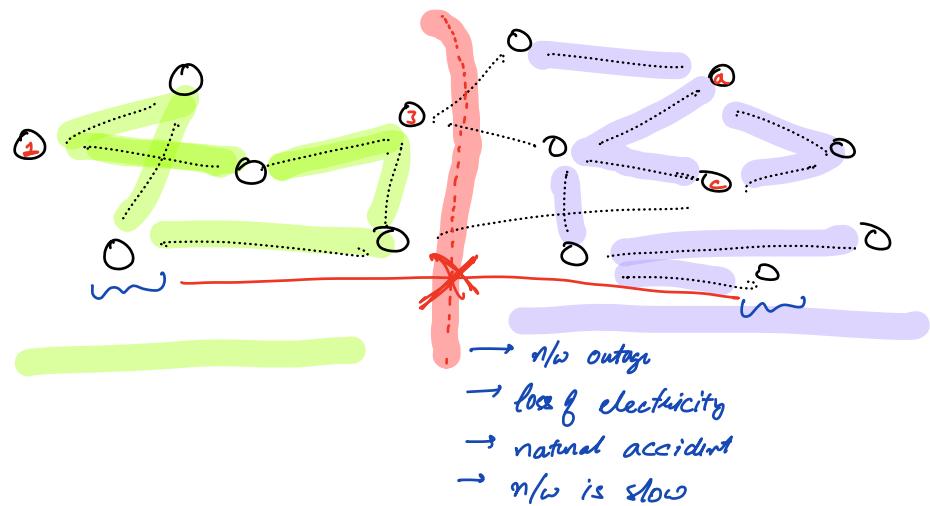
Partition Tolerance → In case of **n/o partitions** → can be arbitrarily large does our system continue to function as expected?

Partition

- ↳ grain from rice
- ↳ concepts
- ↳ humans from aliens
- ↳ n/w partitions



facebook
100 servers



CAP

You cannot have all 3
at the same time !!

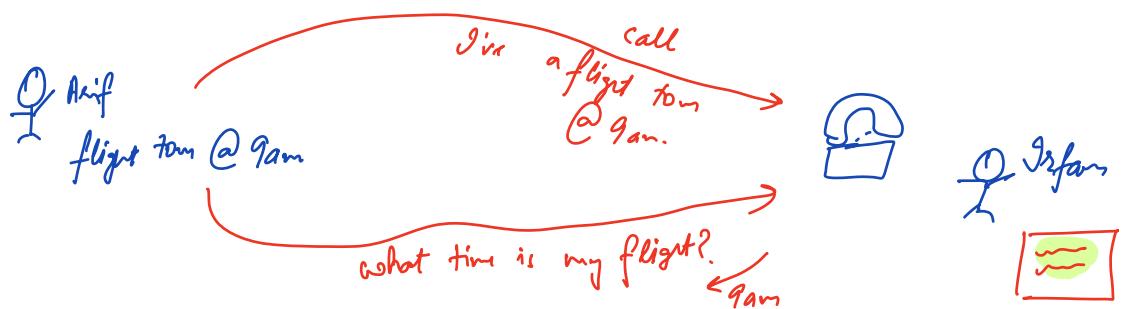
AC → 3 cannot tolerate Partition

AP → Eventual Consistency

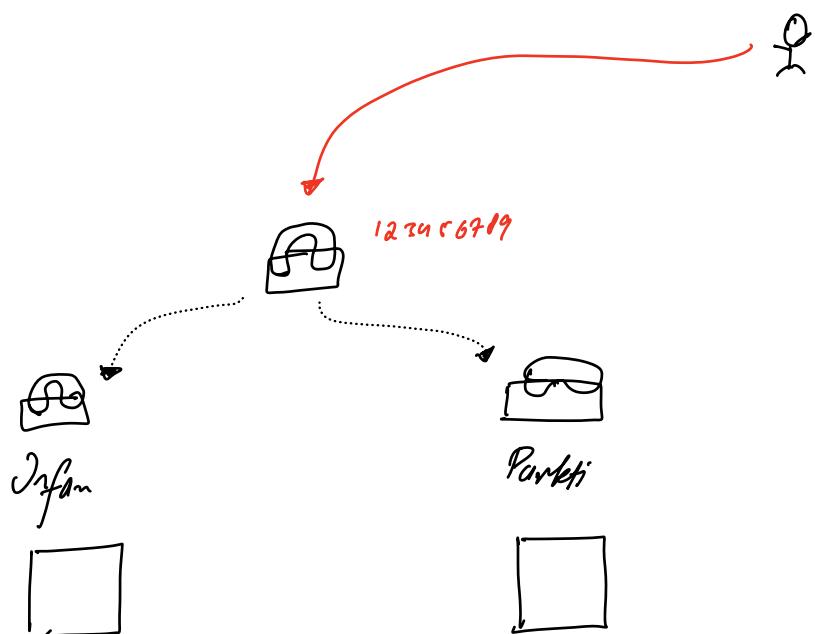
CP → Availability will suffer

Irfan → Reminder Service (Phone)

Phone: 123456789



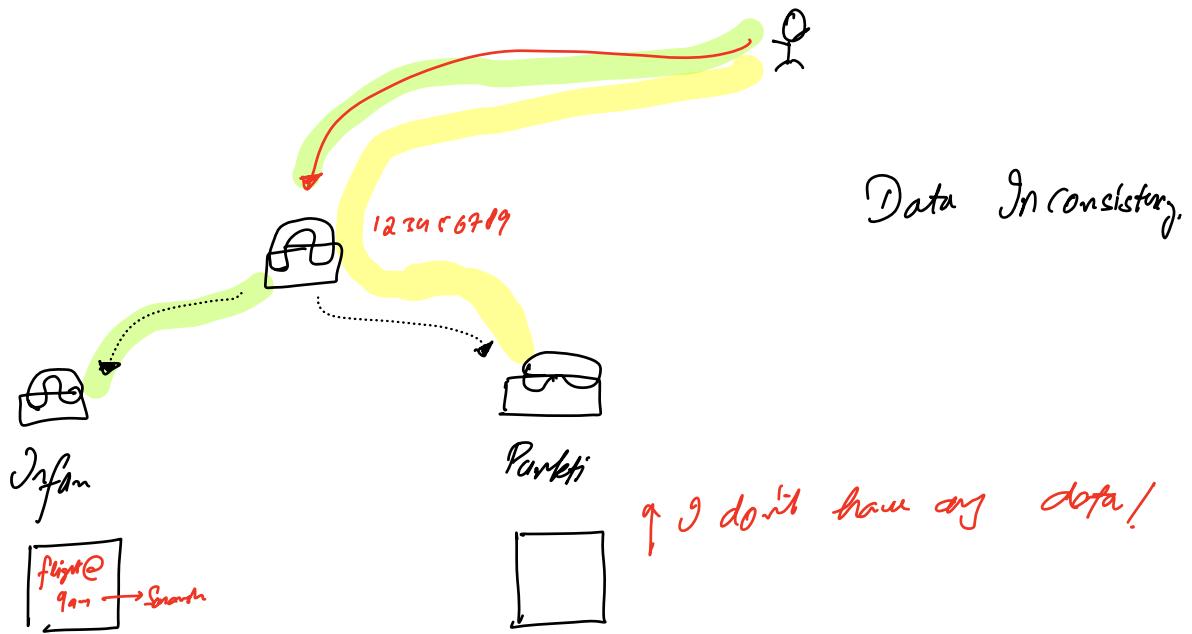
Irfan hires Pankti to help him



Catastrophe → Saranya calls Pankti & is v.v.v. angry
@ Pankti

Sorannah → I told you that my flight was from
Today I called you & you had forgotten

Panketi → You never called



Solutions

① use a single diary → flight for resource
diary is the bottleneck.

High write throughput → storage
is bottleneck.

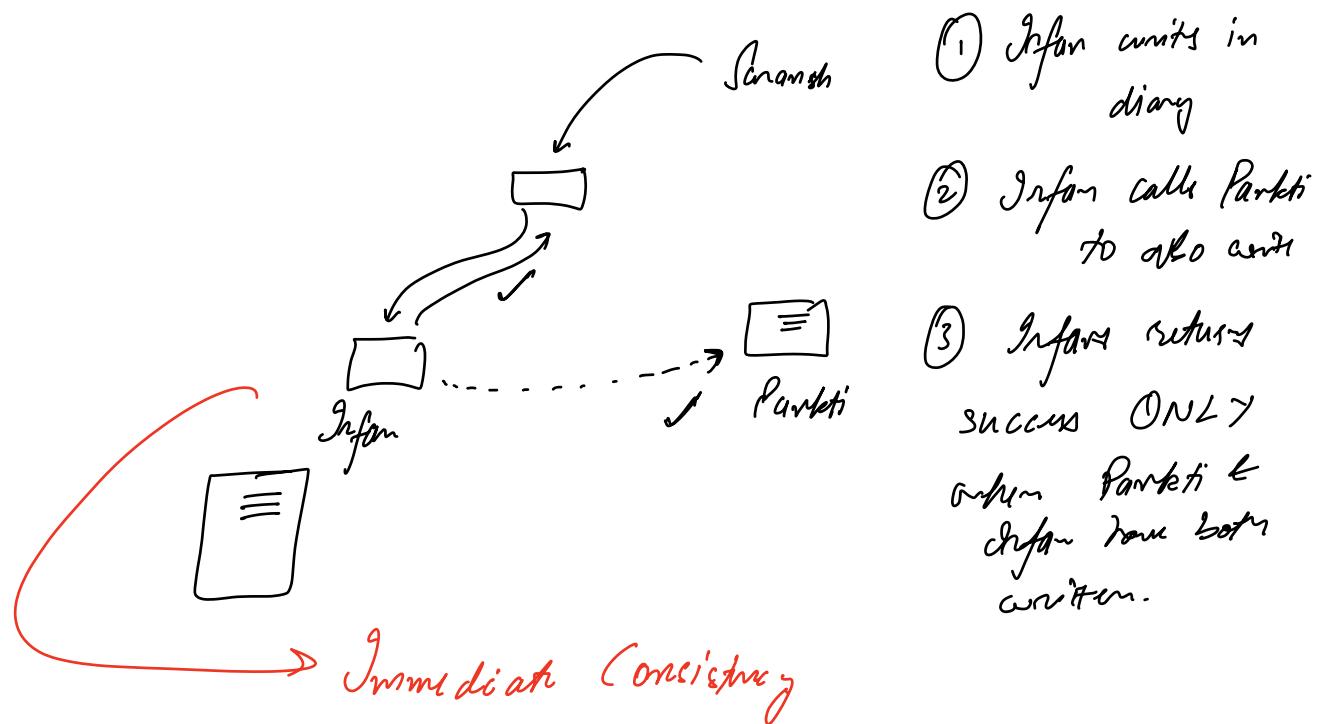
② Consistent hashing → what if Infan goes on
vacation?

③ We need a way to sync-up data

Protocol 1 → Periodically both Parkiti & Infor will sync up their diary

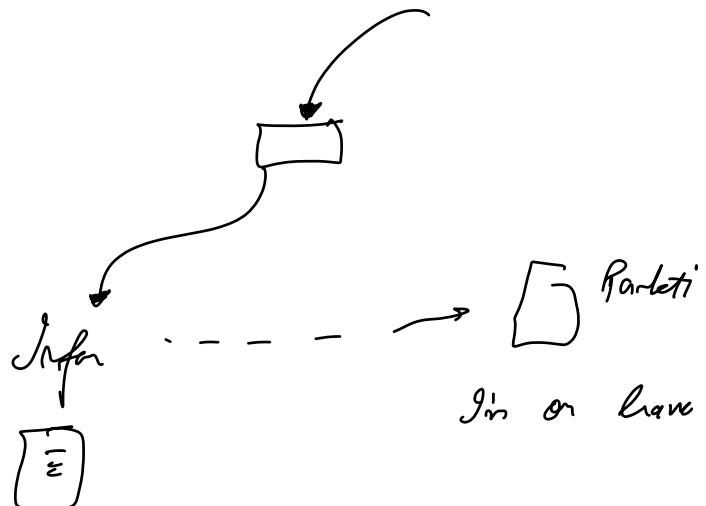
- ① Before sync-up → can be 2 versions of data
Temporary inconsistency
 - ② After sync-up → guaranteed to be consistent
Eventually it will be consistent
- *Eventual Consistency.*

Protocol 2 → Every time a call comes, the operator must call the other operator as well



Pankti goes on a vacation

↳ Server is temporarily down → no n/w partition



for now, only Infan will work.

When Pankti comes back from vacation, before she takes any calls,

she has to copy data to her diary.

Protocol → Immediate Consistency → writing to both diaries

Pankti was on leave (Server was down)

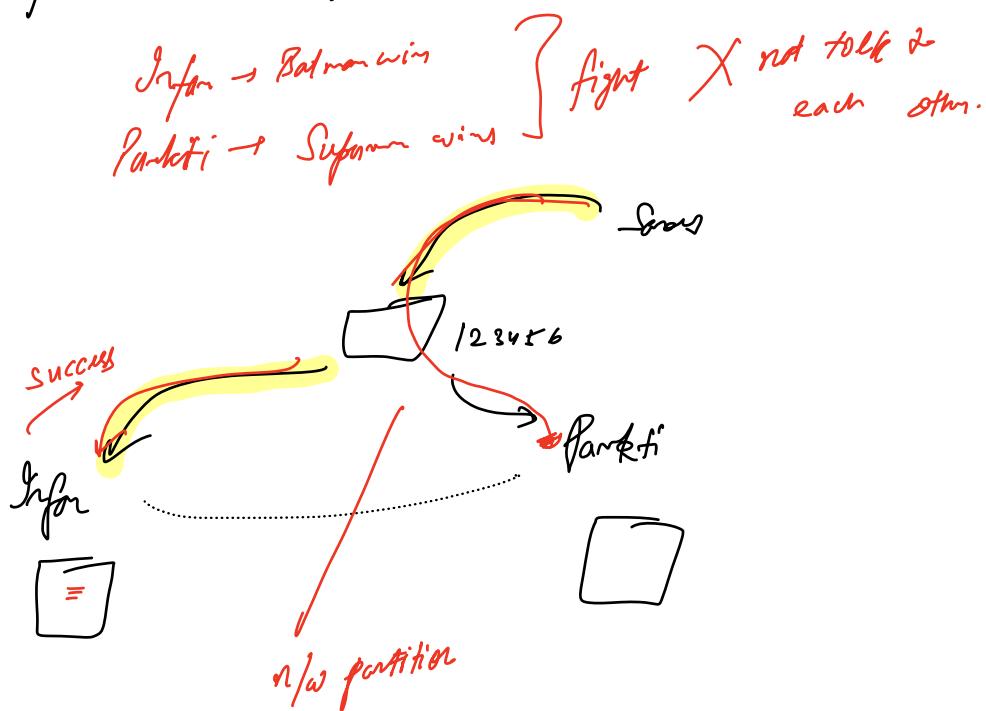
N/w was fine (Infan & Pankti could still communicate)

Consistency + Availability ← No n/w partition

Yifan & Pandeti still talking? Yes \rightarrow no n/w position

Pandeti is on leave \rightarrow she's not serving any req.

Pandeti & Yifan have a big fight!!



Can we get both consistency & availability \rightarrow

No!!

Available \rightarrow No Consistency.

~~A~~ P

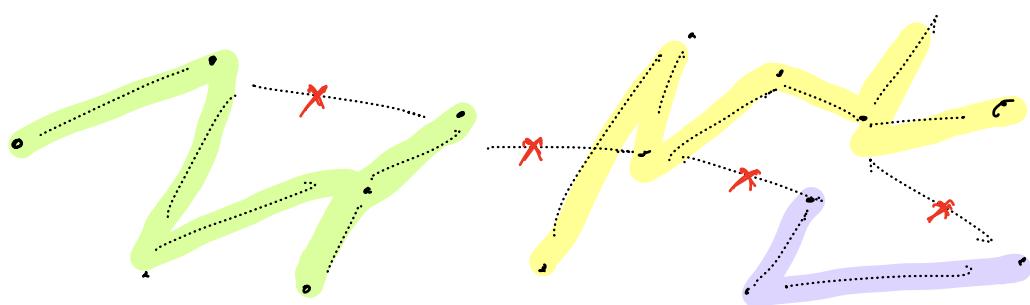
Consistency \rightarrow Not available

C ~~A~~ P

CAP → choose 2 out of C, A, & P

L In case of n/w partition you must
either choose Consistency
or choose Availability.

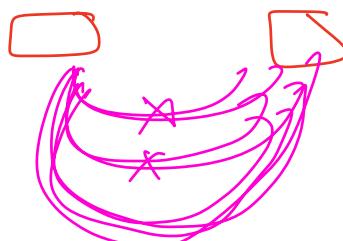
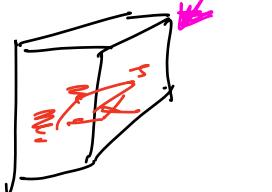
when we have
distributed system
Partitions are
UNAVOIDABLE



Stock Exchange → NSE / BSE / NYSE / --

v.v.v important systems

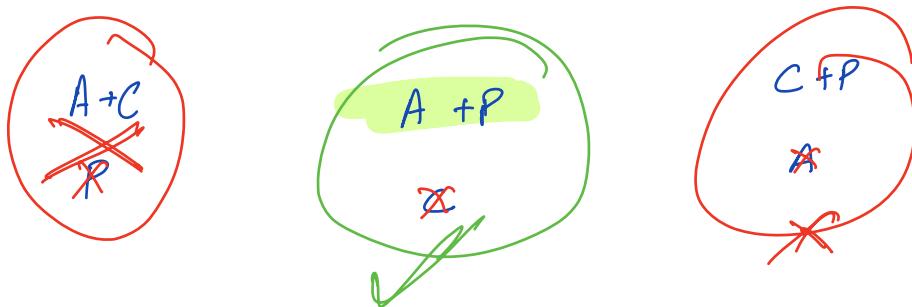
A + C



LinkedIn / Facebook / Instagram / Social Media

Available → always make posts
always read posts

Consistent → if your friend has posted / updated something
you immediately see the changes



Financial Services → ATM machines

Availability → if you go to ATM

① check your balance

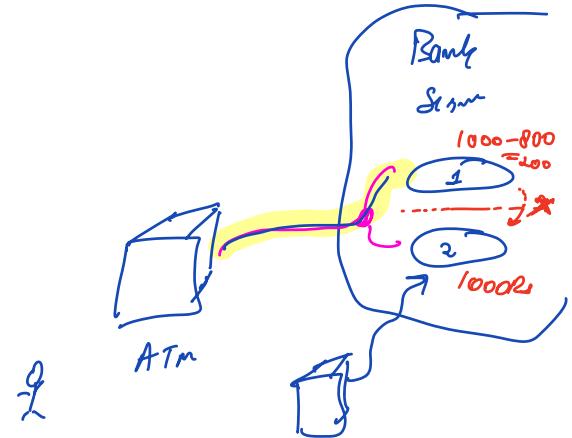
② withdraw money

Consistency → ① if you have 1000Rs in your account

you should not be able to withdraw 2000Rs

② your balance shown should be up-to-date
always

ATM → 1000 Rs balance



① check balance
⇒ 1000Rs

② withdraw money (800Rs)
+ 800Rs

③ go to another ATM that is connected to server 2
& receive 1000 Rs more!!

+ 1000Rs

ATM → should care Consistency (CP)

X

In reality ATMs are AP

① Money can be time-critical
Δ n/w positions or not to client's fault
Δ client has sufficient balance!!!

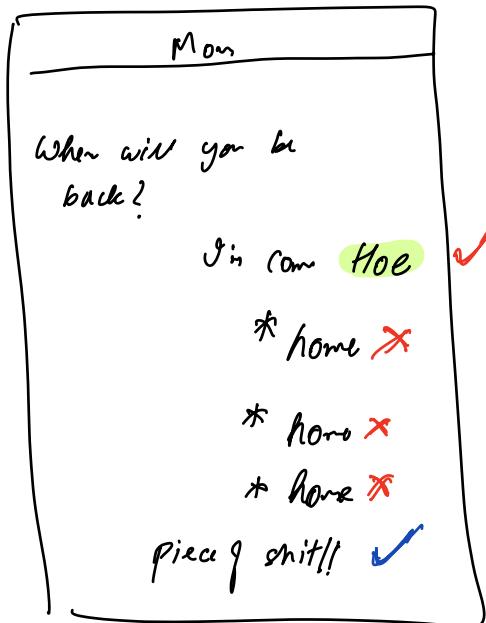
② Your balance will be -800Rs
over draft fee + Interest

WhatsApp / slack / ...

CP

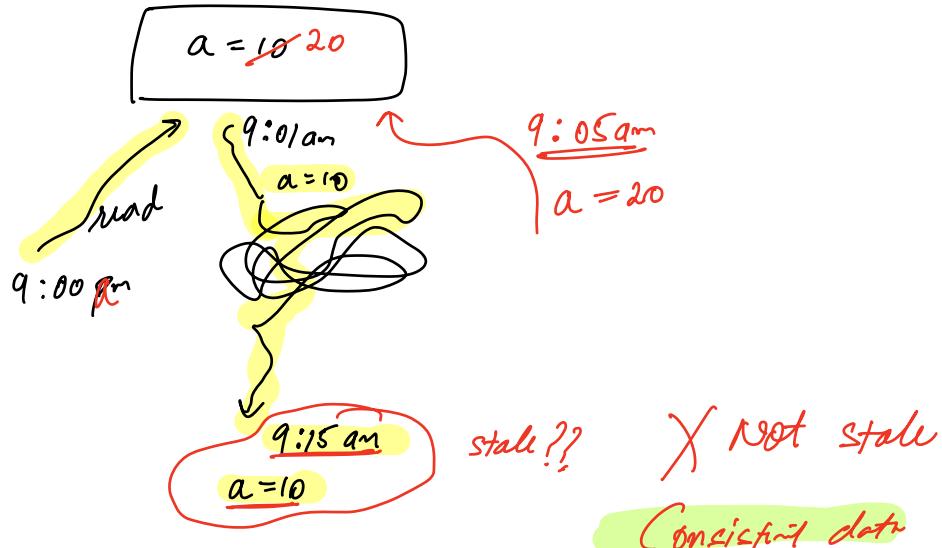
Consistent → any message should be sent in real time
any changes should be reflected immediately.
messages always show up in the correct order

Available → able to always send & receive messages.



Consistency \neq lack of delay

Consistency = No stalls made.



$10:28 \xrightarrow{w_0} 10:50$

PACELC / Master-Slave Replication

PACELC

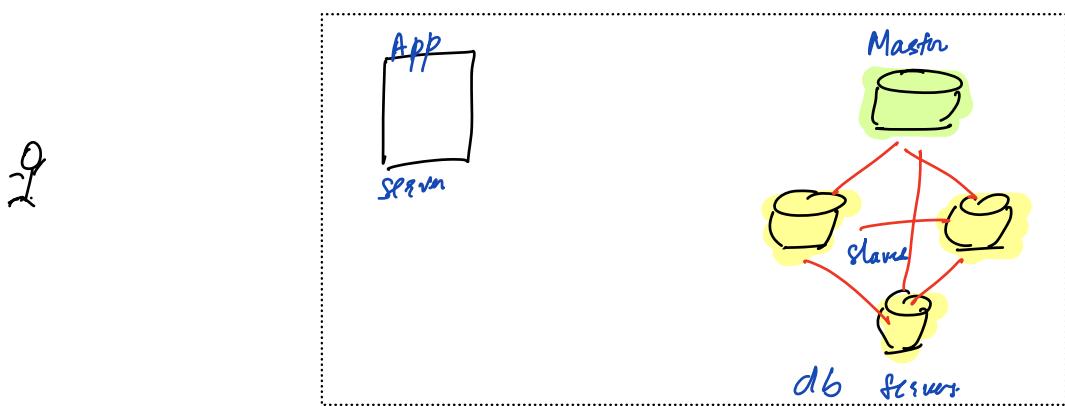
In case of n/w Partition
trade-off b/w Availability & Consistency

but even Else

trade-off b/w Latency & Consistency
↓
service will be slow

Master - Slave Replication → data → ∵ we want to NEVER lose our data.

- ↳ Leader - Follower
- ↳ Active - Passive
- ↳ Primary - Secondary



① what if master goes down?

↳ election → one of the slaves will be promoted to master

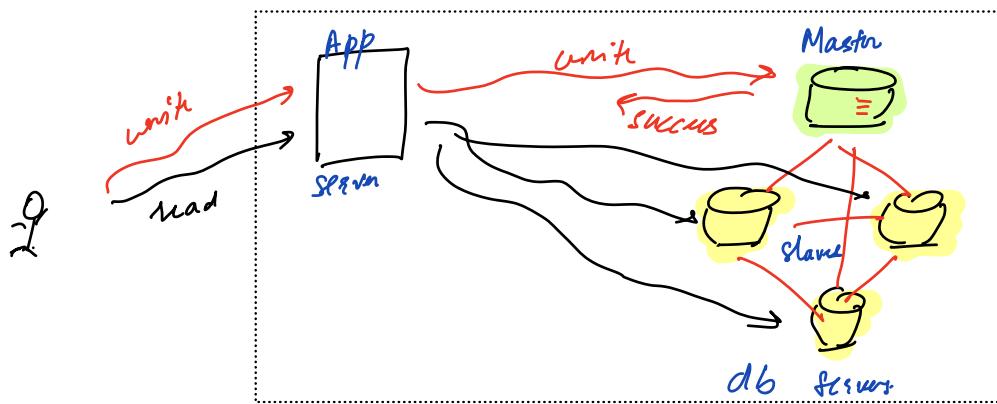
② what if master comes back online?

↳ continues working as a slave

100,000 GRs of data
↳ shard this ::
a single server can't store even 100 GRs of data
↳ replicate
copy from same data on multiple servers

Master-Slave Replication

No Consistency (data loss is possible!)



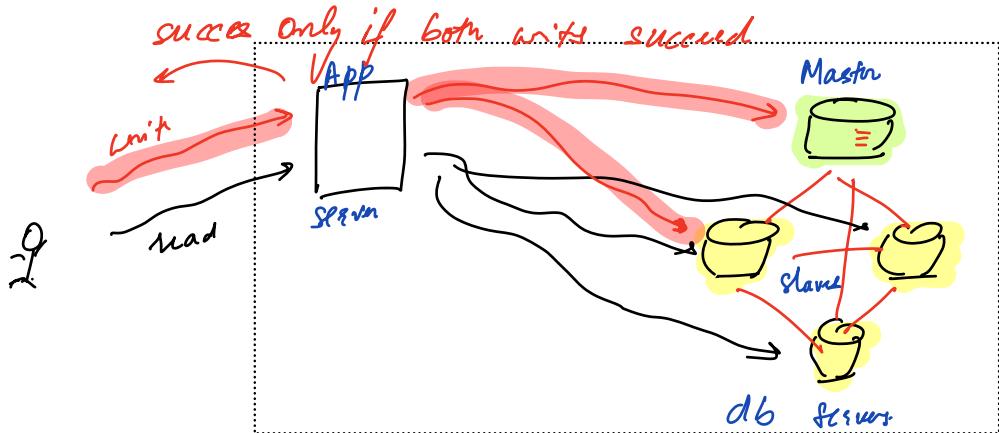
write → only at master (sync)
async → slave will sync-up with master

read → at any 1 replica

① Stale data? Possible ∵ 3 sync delay.

② Lose data? if master HDD fails → any unsync'd data
is lost FOREVER!!!

Eventual Consistency



read → any 1 slave

write → the master + at least 1 slave (any)

App Server is responsible for making this atomic → if any write fails, roll back & return failure
if both succeed return success

Atomic how??

2-phase Commit Protocol (Sync)

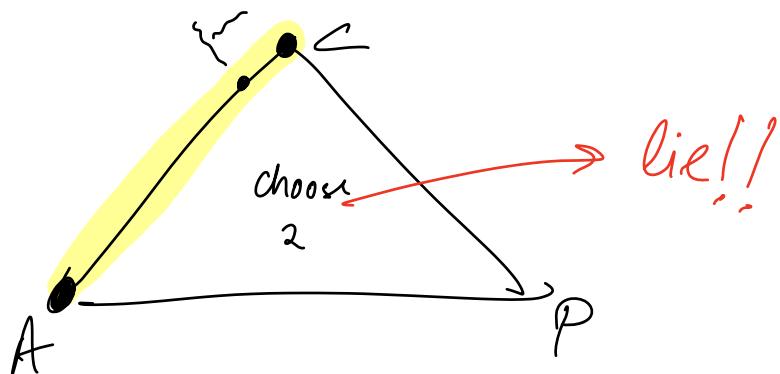
① stale reads? Possible

② data loss? X

① Immediat Consistency

② Immediat Consistency via Quorum diff

③ Tunable Consistency!



In case of Partition trade-off b/w A & C

20% consistent + 80% Avai

$x\%$ Consist + $(100-x)\%$ Avai

Caesandra

~~MUGF~~ → extra class → to cover Master-Slave
to an fb caching case study.

Mind-Wiki session → 9pm tomorrow/Tuesday)

✓ 'Optional.'

call me +91 7851769231

AC → stock exchange

→ distributed training for ML → chartPT training

CPI → AP System

