

10/17 739

Last week:

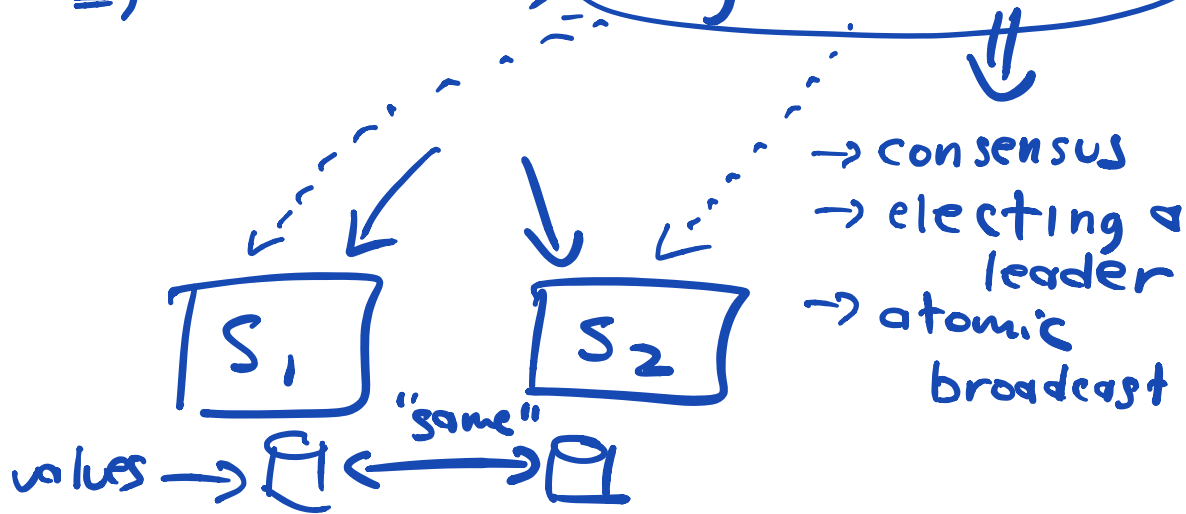
→ HA NFS

→ Friday: ML → Systems

Replication ⇒ Fault Tolerance

⇒ "hard"

⇒ agreement

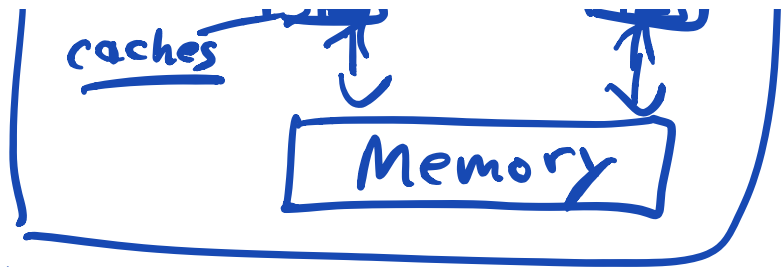


Why hard?

⇒ concurrency

⇒ failures (contrast to shared-memory multiprocessors)



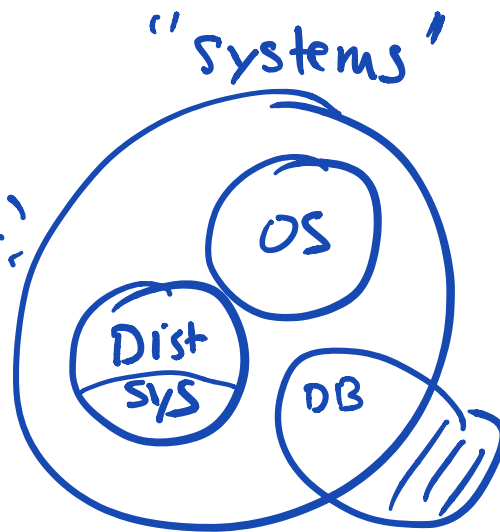


⇒ efficiency

Conceptually:

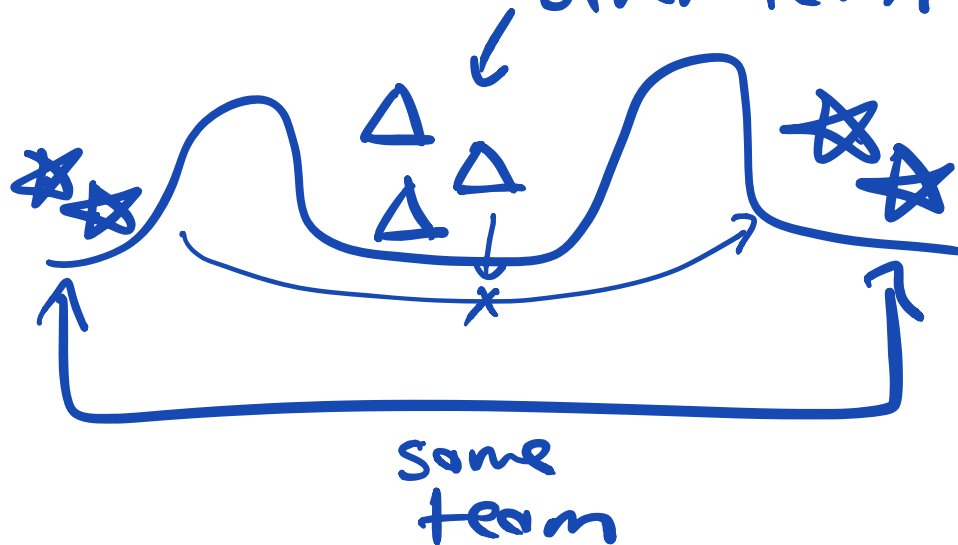
Dist Sys:
Practice ↔ Theory

possibility of
consensus

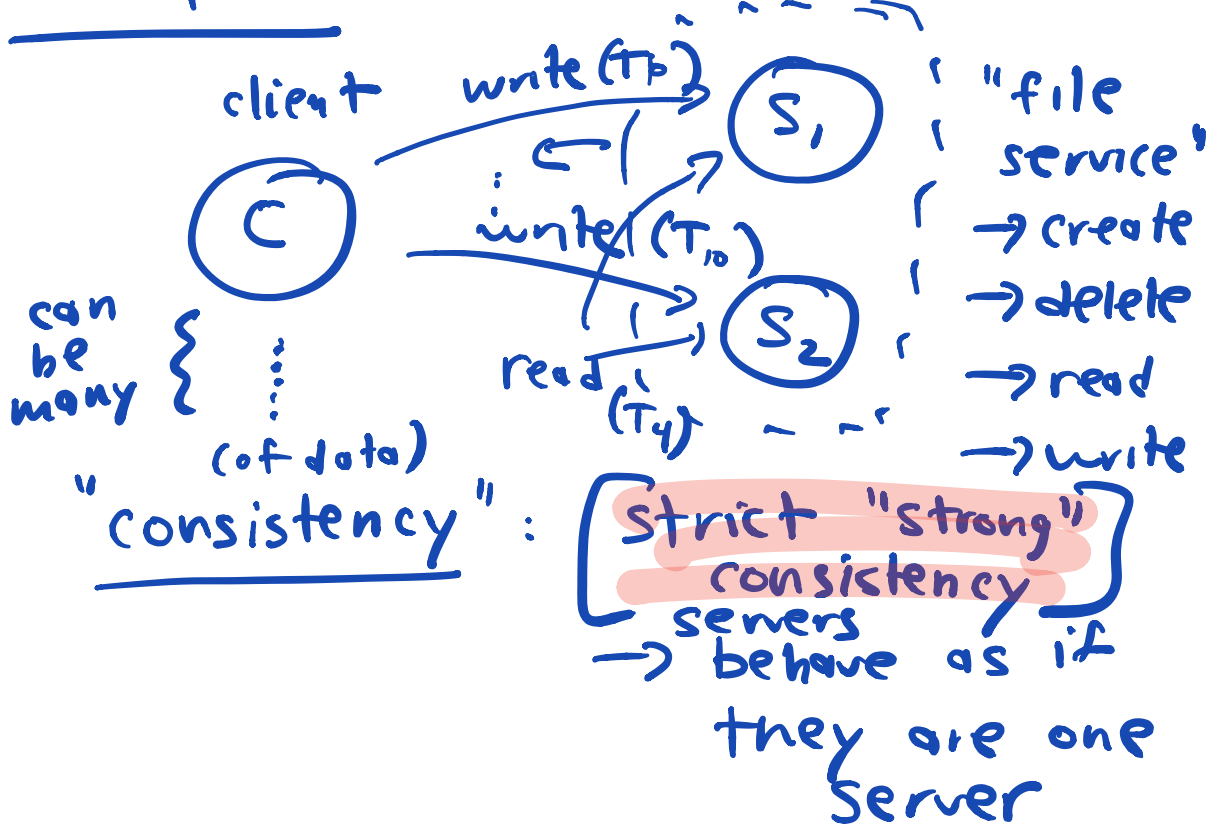


e.g.)

2-generals problem
other team

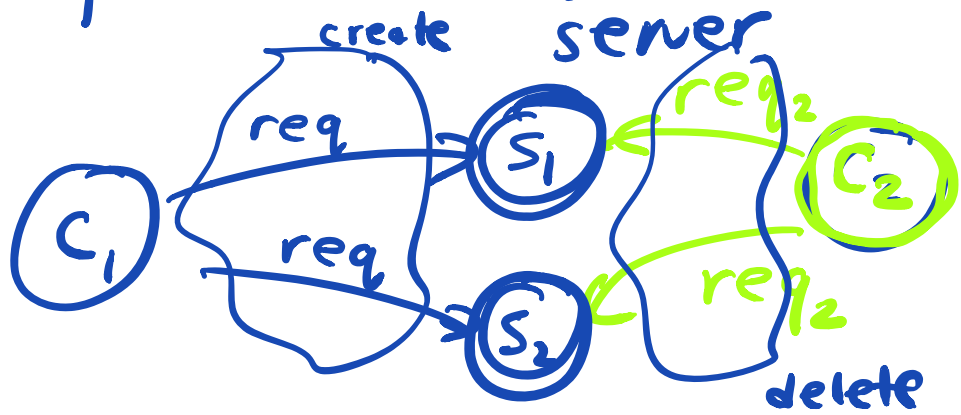


Example: Replicated Service



Approach #1: (broken)

client: request just sends req to each server



Problems:

=> ORDER

replica
=> divergence

=> ordered network
("atomic broadcast")

=> still needs agreement

=> Determinism => challenging

=> divergence

=> Faults / Failure

=>



=> miss updates

down
for
a
while

=> divergence

depends on failure model:

-> fail stop: down,
stay down,
and easily

more realistic: detected

→ fail recover:

crash, ..., reboot

miss ↑
updates

how to recover?

Break:

→ (Mid term)

→ Mon eve (7:15pm)

→ 2 hours long ;)

→ "open mind"

+ pen/pencil

old
exams
on
web
page

if
desired

[1 page cheat
sheet]

(2 sides)

[8.5" x 11"]

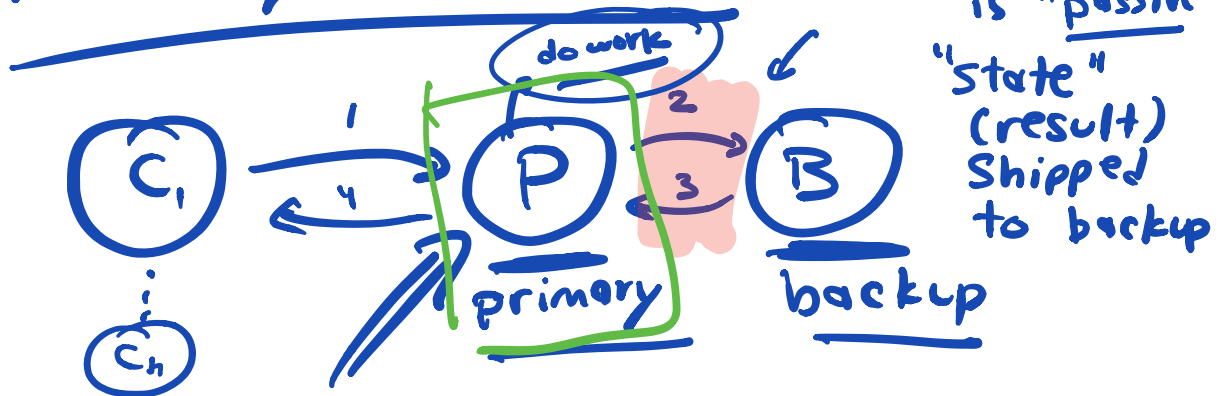
no posters

"Daily" writeups

→ ["Correct"
ans]

(share reasonable
from class w/
all)

Primary / Backup



solves!

ordering ✓

determinism ✓

hard: how to deal w/ failures

→ failure detection

"heartbeats" + timeout

problem:



backup takes over,
but primary still up

"split brain"

conservative:

=> lock of availability

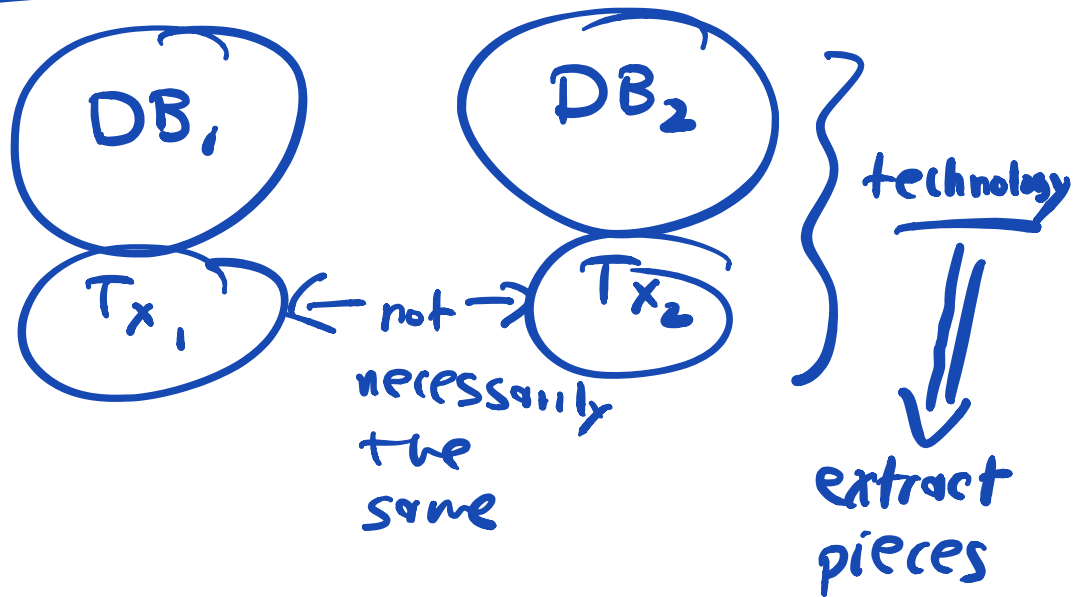


new problems?

consistency

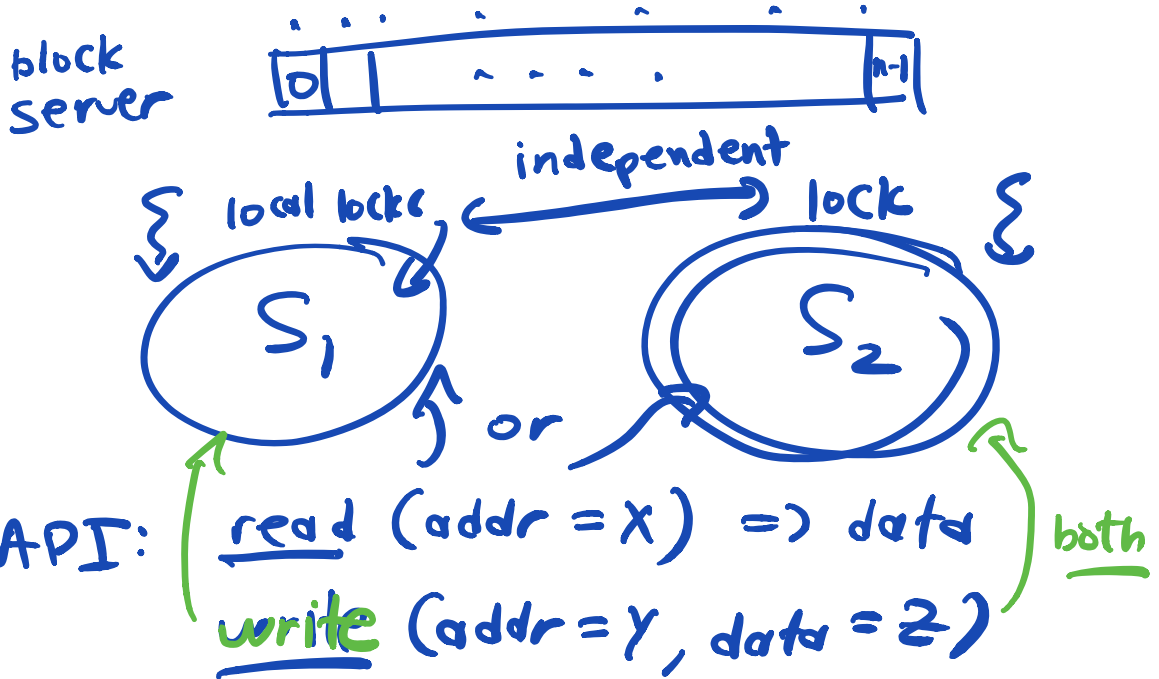
agreement => which backup should take over?

Two-Phase Commit:



ZPC:

example: dist. storage server

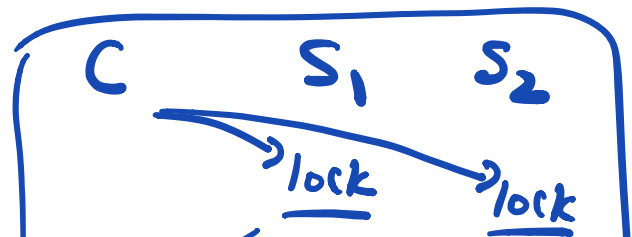


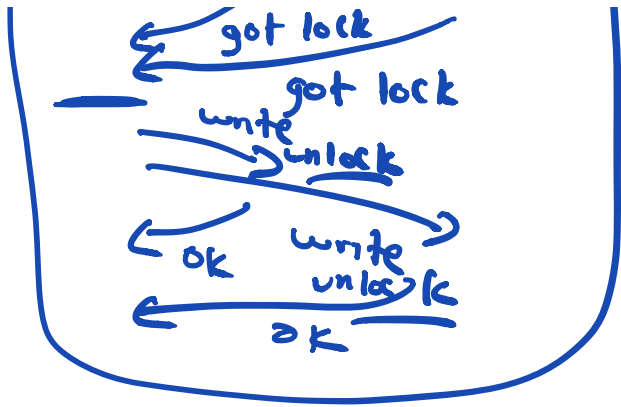
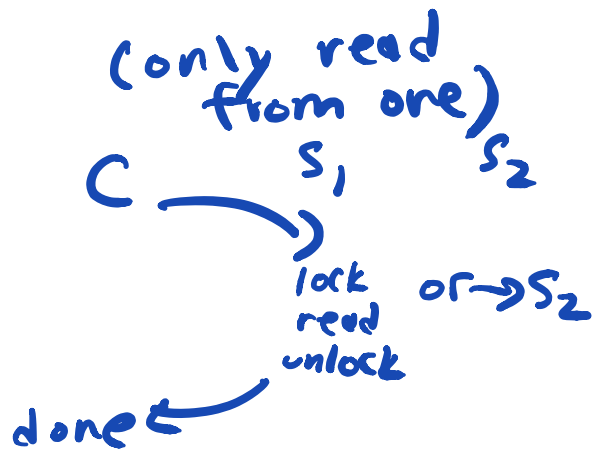
"single copy" semantics:
"strong" consistency

on each machine:
need locks (mutex)

Read

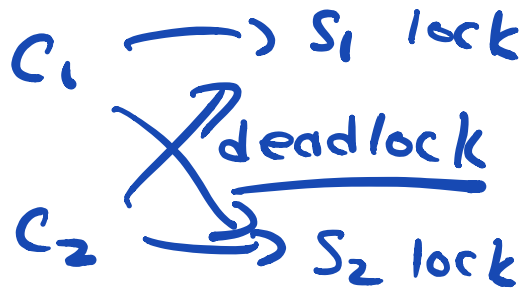
write





Problems:

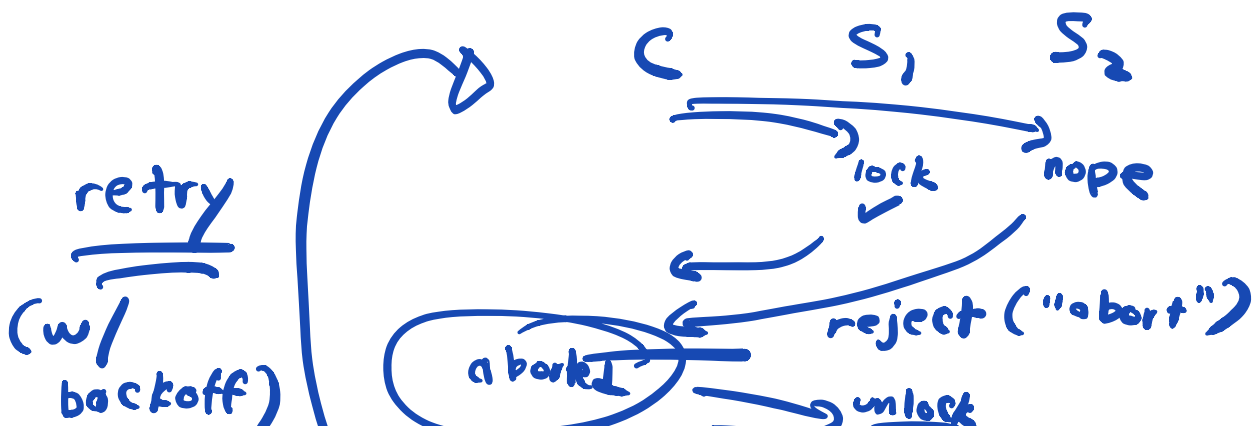
> 1 client:

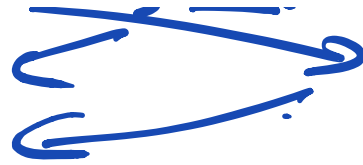
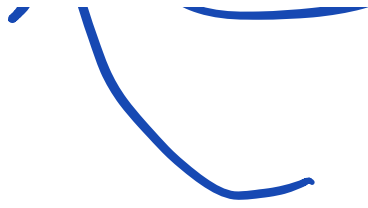


solve deadlock: how?

[ordering: S₁ then S₂] => high latency

[say no:]





nodes fail:

log
at

some state
particular
points