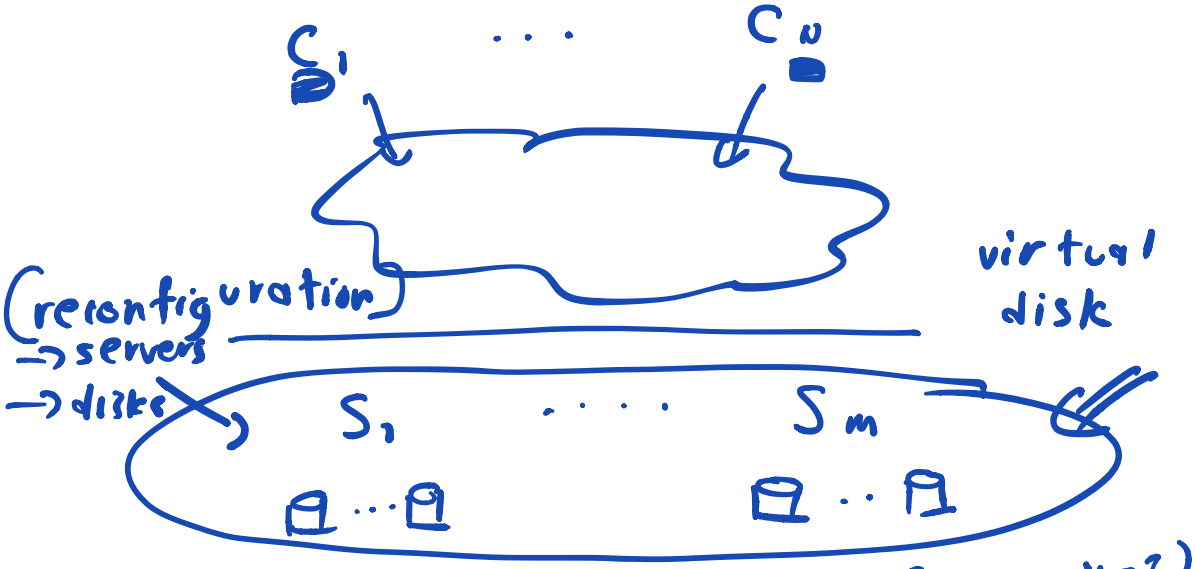
Petal / Frangipani [Case Studies]

interface / abstraction:



for each VirtDisk: blocks (0 ... N-1)

read/write (offset)

(+delete)

[All servers cooperating to provide abstraction]

Liveness: heartbeats + majority consensus

Global state:

[membership, global mapping into] } widely replicated

=) Paxos

=) common pattern { inner ring: paxos; outside: not }

[V -> P translation]

=) <v Disk, offset>

!! [key]

<server, disk, local offset>

Data Access

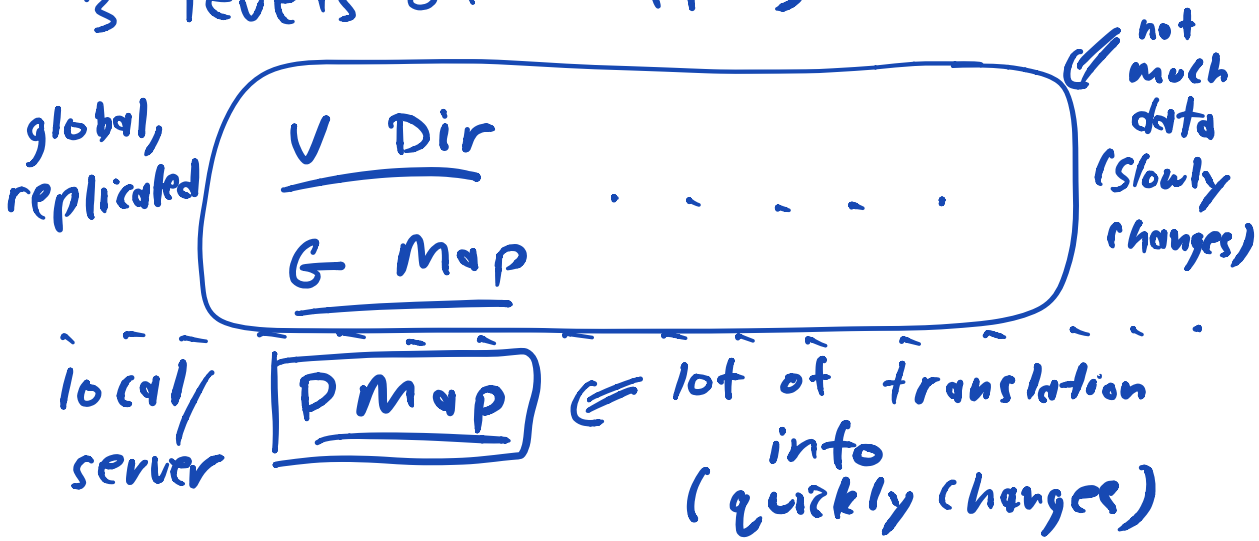
=) mirroring, striping

Recovery

=) . . . .

V → P (virtual → physical)

3 levels of mapping



Sec 2.1:

V Dir: Vdisk ID → (Global Map ID, epoch #)

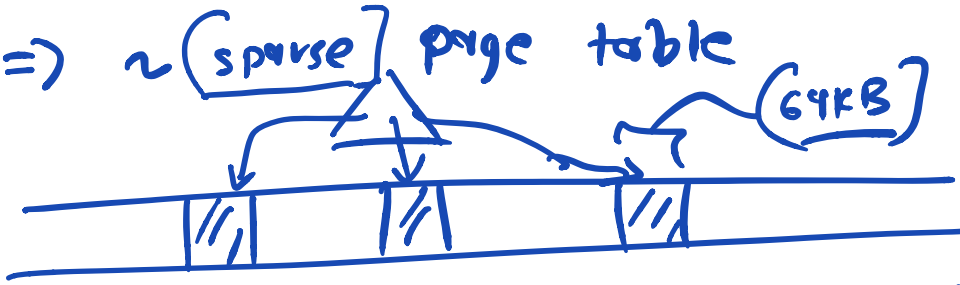
G Map: GMap ID → server(s)

[servers, redundancy scheme e.g. chained declustering]

PMap (per server):

offset (v disk) → local disk, local offset

⇒ ~ (sparse page table)



Backup: snapshots

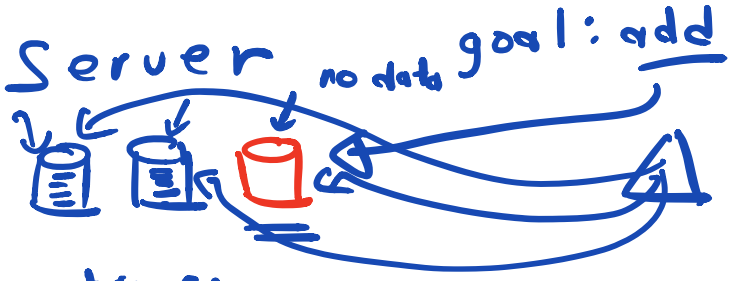
(GMap ID, epoch)

GMap: immutable

=) are not "consistent"
(from app perspective)

Incremental Reconfiguration:

1) Local



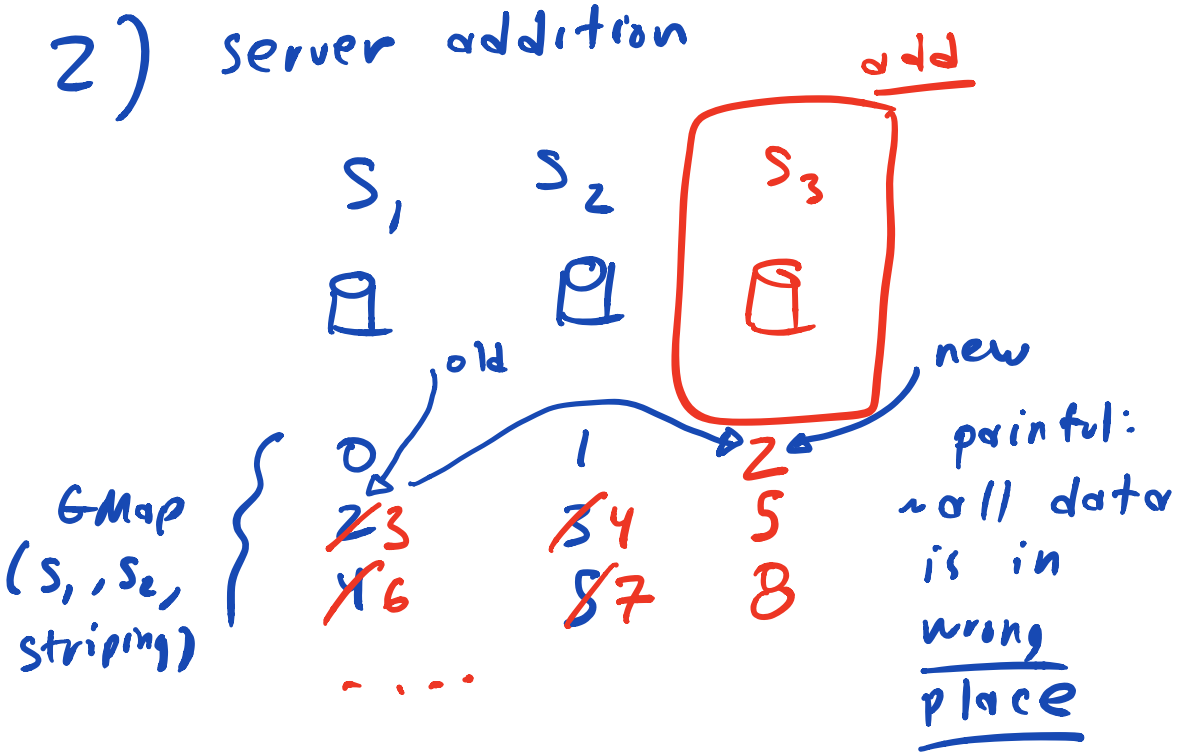
actions:

1) writes → easy
2) reads → hard

offline / background rebalance

All local: "easy" (no Paxos, consensus)

2) server addition



Process: reconfig

{
create new gmap'
change vDir → gmap'
move all data
continue normal operation
}

"it depends"

"Basic" algorithm:

gmap_new ⟵ gmap_old

writes: new

reads: use new map
but, it fails,
consult old map

(bg: moving data old map → new)
@ some rate

Problem? too slow

Solution: relocate parts of map @ time

v Disk:

regions: old, new, fenced ⟵

old: use old

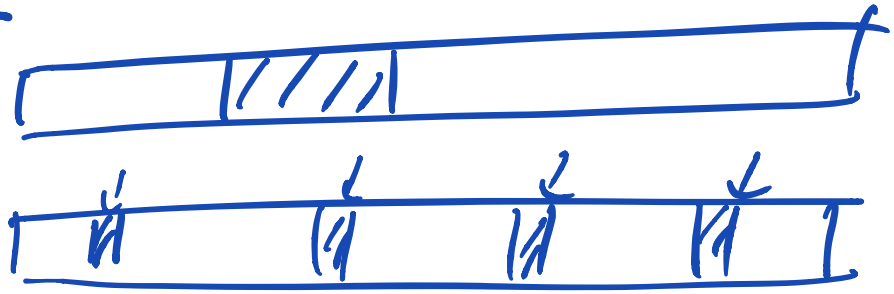
new: use new

fenced: in flight
(new, old?)
use "basic"

Perf problem? Hot Spots

Locality in W/L (workload)

Solve? => guessing that

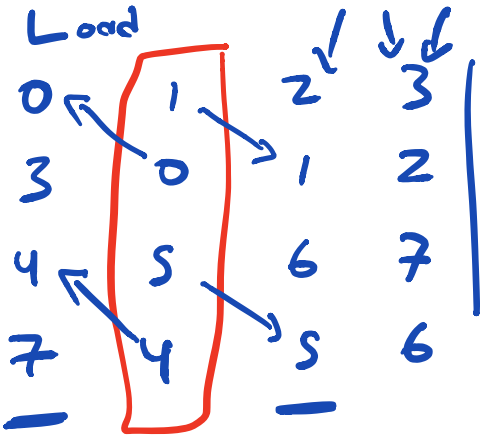


Data Access / Recovery

1) Redundancy scheme
   => striping
   => chained declustering '89

mirror:

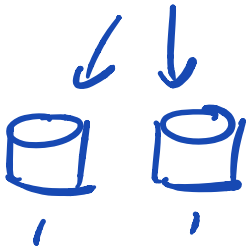
load problem
balance



better: ✓

dynamic load balancing:
clients track outstanding # of reqs / disk
=) balance

Petal : replicas



approach: logging: busy

Primary / Backup

=) writes: slower logging

writes

next optimization:
=) (lazy) in marking "not busy"

Petal :

virtual Disk

target for FS's, DBMS's

Limits : scale

reconfig => expensive