Today

- Dynamo
- End class: what have we learned? (wrap up)
- Admin
  - class presentations
  - midterm #2
  - other

Dynamo

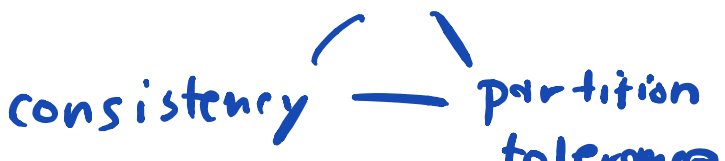
Problem: DBs of time => didn't scale

Goals:

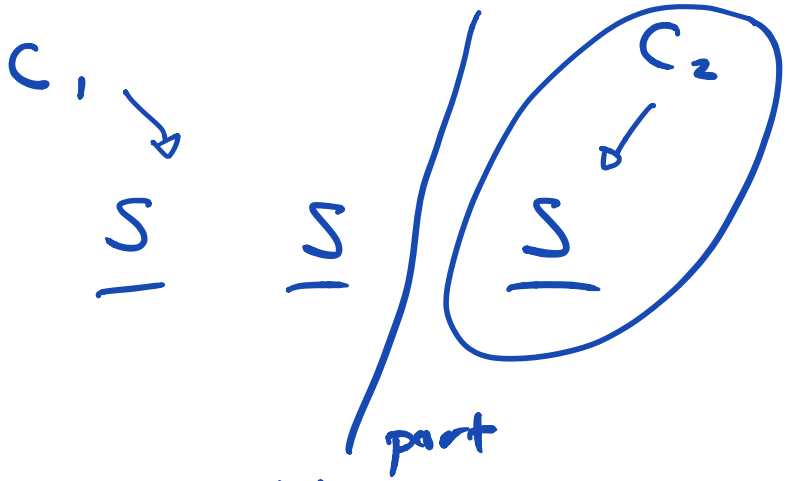
Scale (incremental)

Availability: at all costs

=> Brewer: "CAP" theorem => truism



=> pick any two



Dynamo : availability

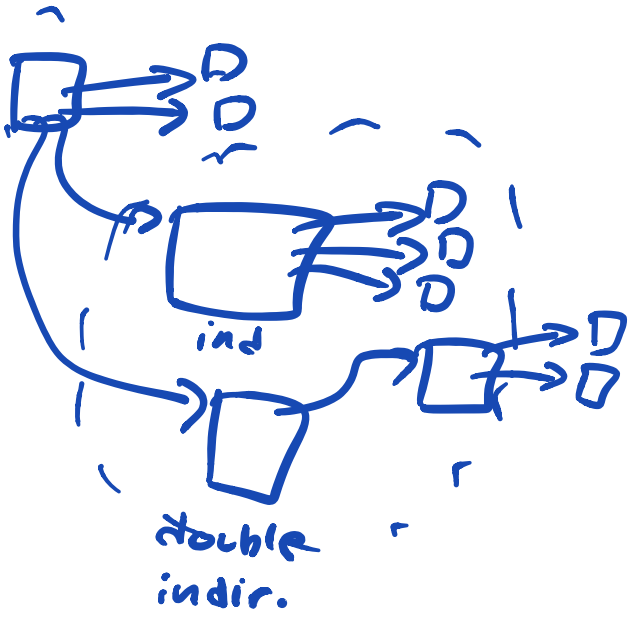
always writeable → conflict res on reads (later)
→ just accept writes

conflicts:
app-specific (Bayou)

Performance:
latency => tail latencies (99.9th percentile)
(predictable)
{Service Level Agreements}

=> known h/w
=> admission control

e.g. inode →



System Arch

API: get / put "key value"

hash (key) => lookup

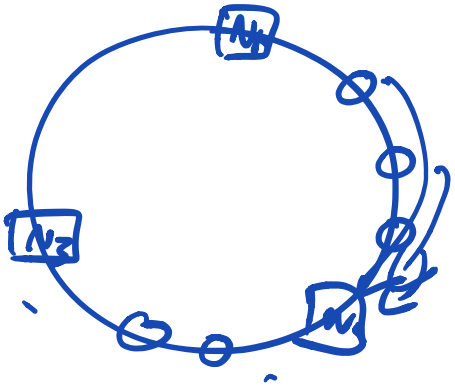
Partitioning:

"consistent hashing"

virtual nodes:

-> load balance under join/leave

-> hetero.



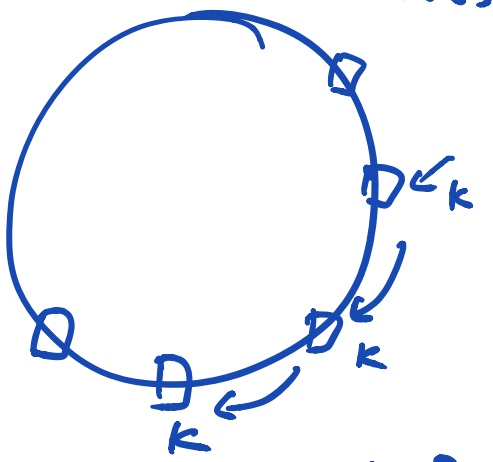
Replication : N-way

writes: => coordinator

↙

replicate key/value => success

careful (N distinct phys nodes)



N = 3

Versions "eventual consistency"

W .....

R
R
R

} may see new or old

e.g. shopping cart

how to map shop cart ops onto get/put?

=) op => put add cart => put
              del cart => put

~ versions

=) vector clocks

list of (node, counter) ...

allows detection of conflicts, non-conflicts

concern: size => lists can get big

limit: 10

Get/Put:

=) routing

2 options:

-> Load Balancing tier

-> client

$-v_1$

$-v_2$

$-v_3$

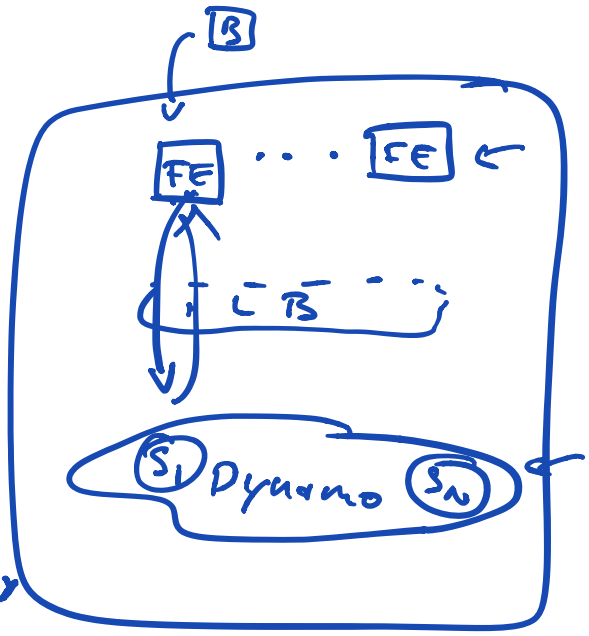
$-v_4$

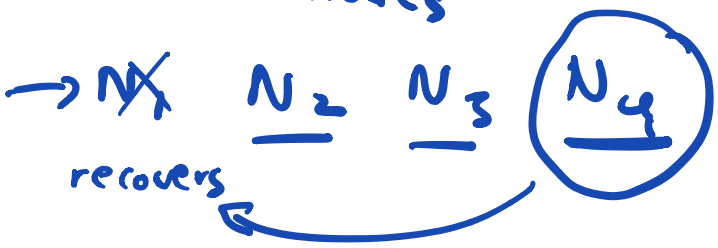
$-v_5$

writes: coordinator =>
replicate to first
N-1 "healthy" nodes

quorum: N, R, W
read write

"sloppy" quorums:
→ avail.
replicate to N healthy nodes



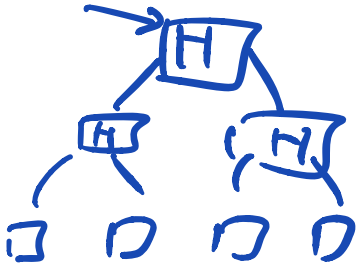


list: eventually
contact
N_1 => replica
"hinted handoff"

Replicas Sync.

periodic nodes sync w/ each other

Merkle trees over key/values
=) efficient sync

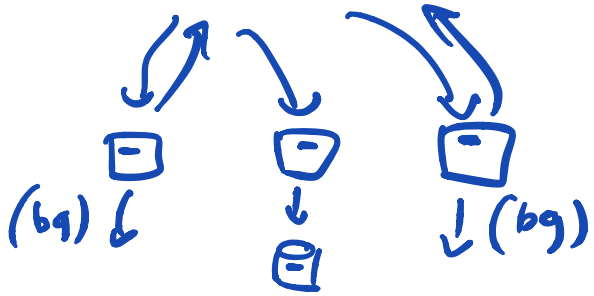


Experience + Lessons

Perf vs. Durability:

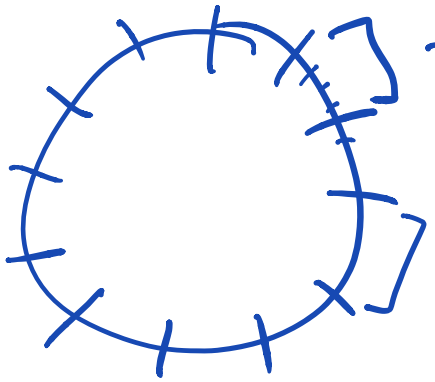
write buffer: 5x tail latency

write to N nodes



Q w=2

Load Distribution



fixed-size partition

M Trees

Background Tasks:

replica sync, etc.

⇒ bg tasks can slow fg tasks

Schedule bg work:

→ driven by measurement
fg tail latency

→ throttle bg work accordingly

~ TCP

Conclude:

Early NoSQL → Cassandra
Riak
⋮

Interesting:

wasn't super popular
inside amazon (@first)

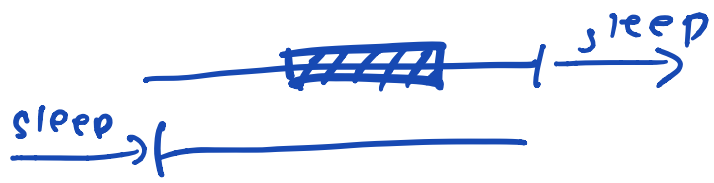
→ system, not service
(includes
op.)

Admin:

- Presentations:
  - 20 minute talk
  - slides, bring laptop
  - Structure:
    1) Intro → Title, Who; Problem; Solution; Summary
    2) Body — Outline →
    3) Conclusion — Outline →
       - Summary + Thoughtful conclusion

2nd Midterm: no thanksgiving questions(!)

- take home
  - typing ok
  - Sun a.m. → Mon p.m (6)



Project write up:

{ short
5-pages (+) }