

Google File System

What is missing in NFS, AFS?

- Capacity ✓
- Scalability
- Reliability



How to use networked systems to
improve reliability?

How to use networked systems to improve reliability?

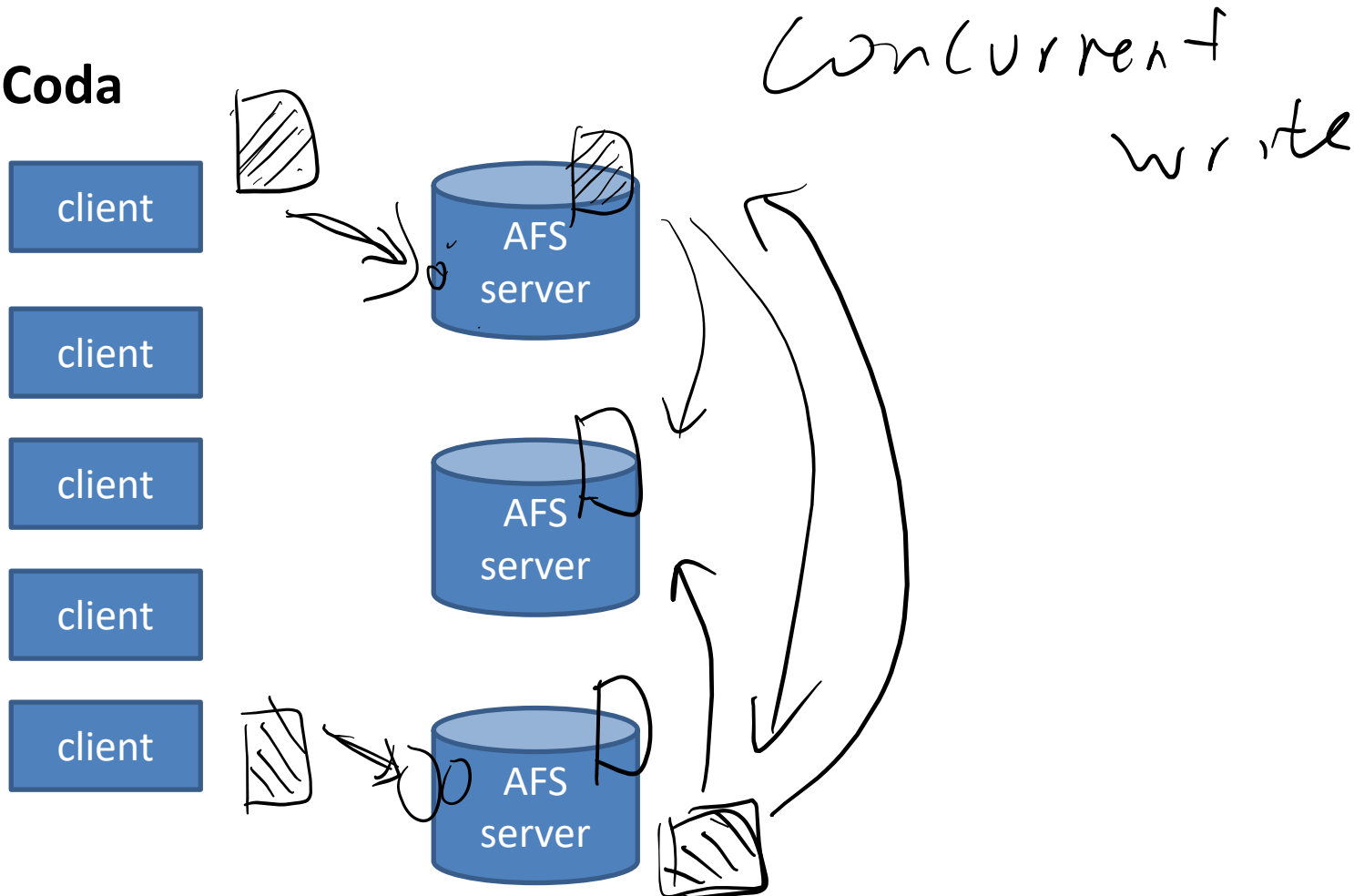
- Replicate!

Two key techniques in networked systems

**Replication
Partition**

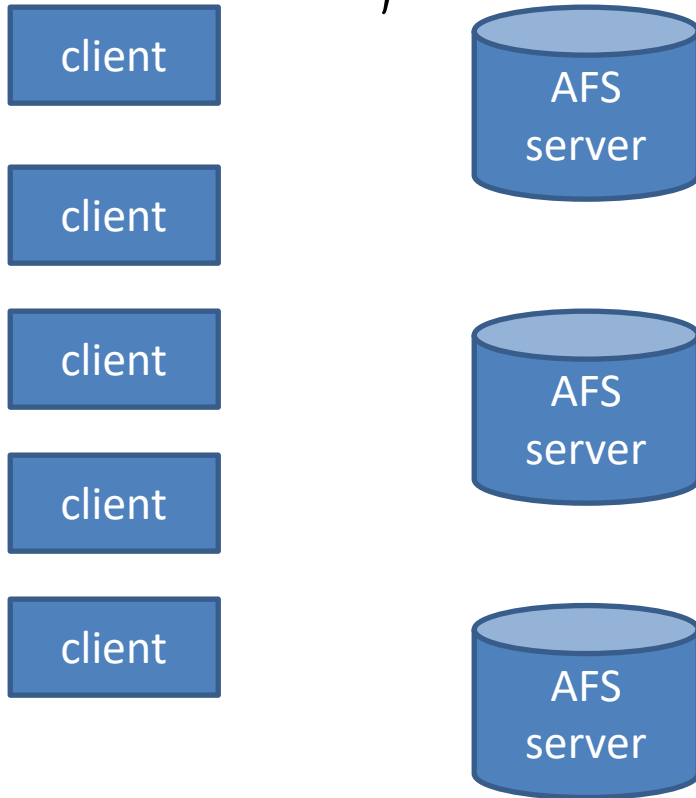
Replication attempt @ Coda

Coda



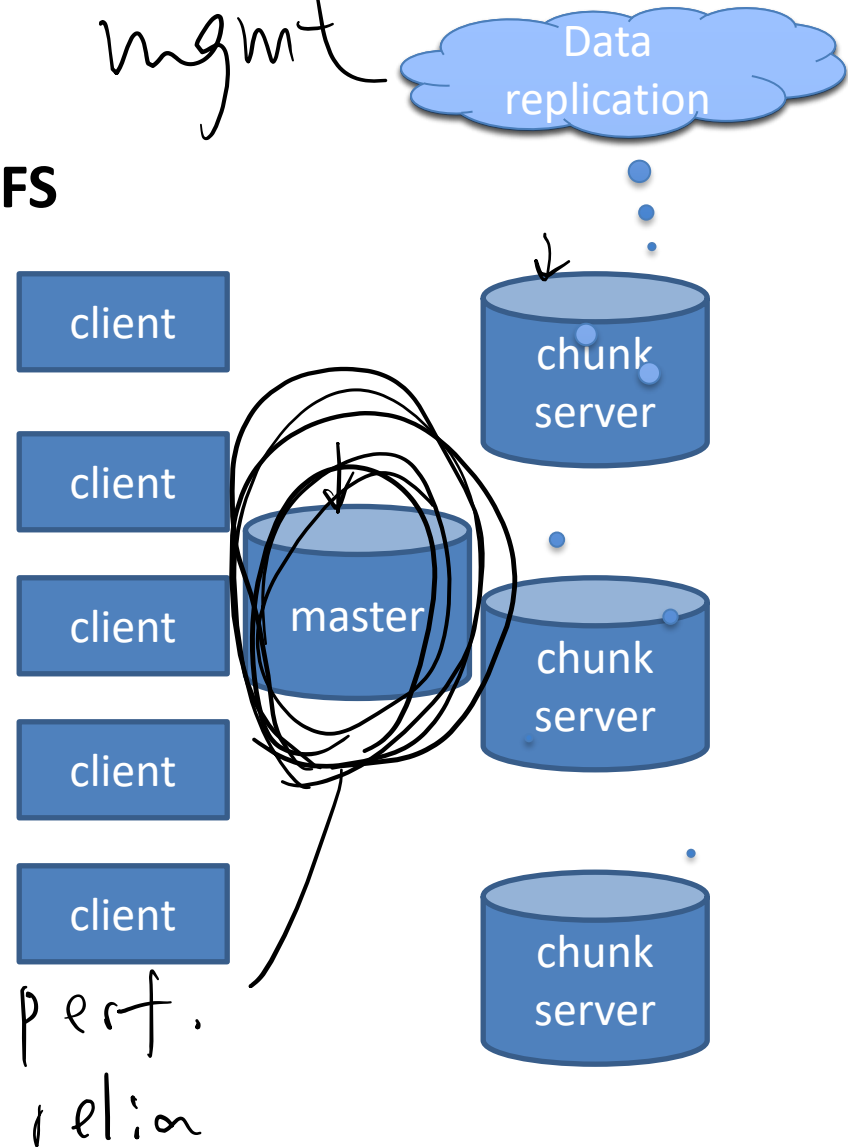
distributed
mgmt

Coda



GFS! centralized
mgmt

GFS



Assumptions/Goals

- Any component could fail
- Some large files instead of many small files
 - Impact
- Append-heavy write; sequential accesses
 - Impact
- ...

➔ Different designs from traditional file systems

Why does GFS have a master?

- Strengths
 - Easy to manage
- Weakness
 - **May become performance bottleneck**
 - May become single point of failure
 - ...



How did
GFS
overcome
this?

Normal file system access (single machine)

- What if I want to read/write “/a/b/c”, 5Kth byte
 - Read the i-node of root “/” (from disk)
 - Search i-node of “/”: find the data block
 - Read the data block of “/”: find #i-node of a
 - Read the i-node of a: find the data block
 - Read the data block of “a”: find #i-node of b ...
 - ...
 - Read i-node of c

Normal file system metadata

- What are meta-datas?
 - i-node
- Where are meta-datas?
 - disk
- What is the data block size? Why?
 - 4 K

Google file system meta-data

- What are the meta-data?
 - Does it still use i-node?
- Where is the meta-data?
 - ??
- What is the block size?
 - Still 4K?

read ((offset) -) 

Google file system meta-data

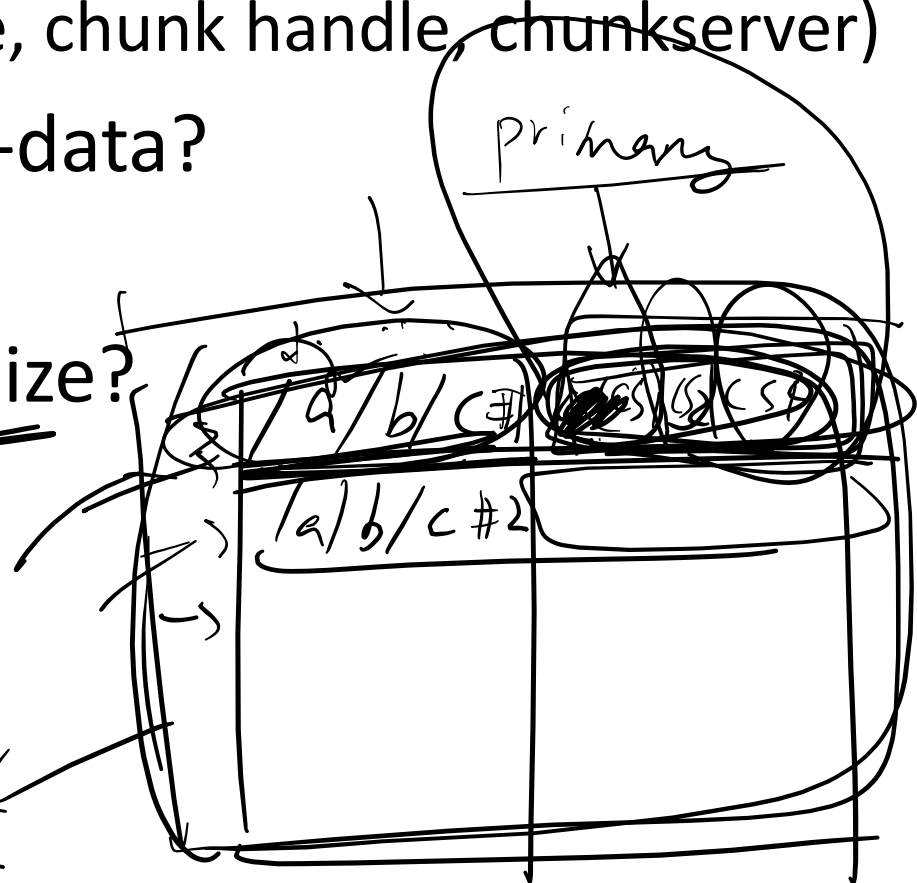
- What are the meta-data?
 - Mapping (filename, chunk handle, chunkserver)
- Where is the meta-data?
 - In memory
- What is the block size?

– 64 M

4K * 1000

16 x 1024

Mem



Google file system read

- What if I want to read “/a/b/c”, 5Kth byte
 - Whom to ask to know where is the block? chunk?

Google file system read

- What if I want to read “/a/b/c”, 5Kth byte
 - Ask master
 - File-name + # chunk → chunk handle → list of chunkserver
 - Contact (closest) chunkserver
 - Compare version number
 - Get the data

Write in GFS

data control

CMD: write!

Data replication

client (

chunk server

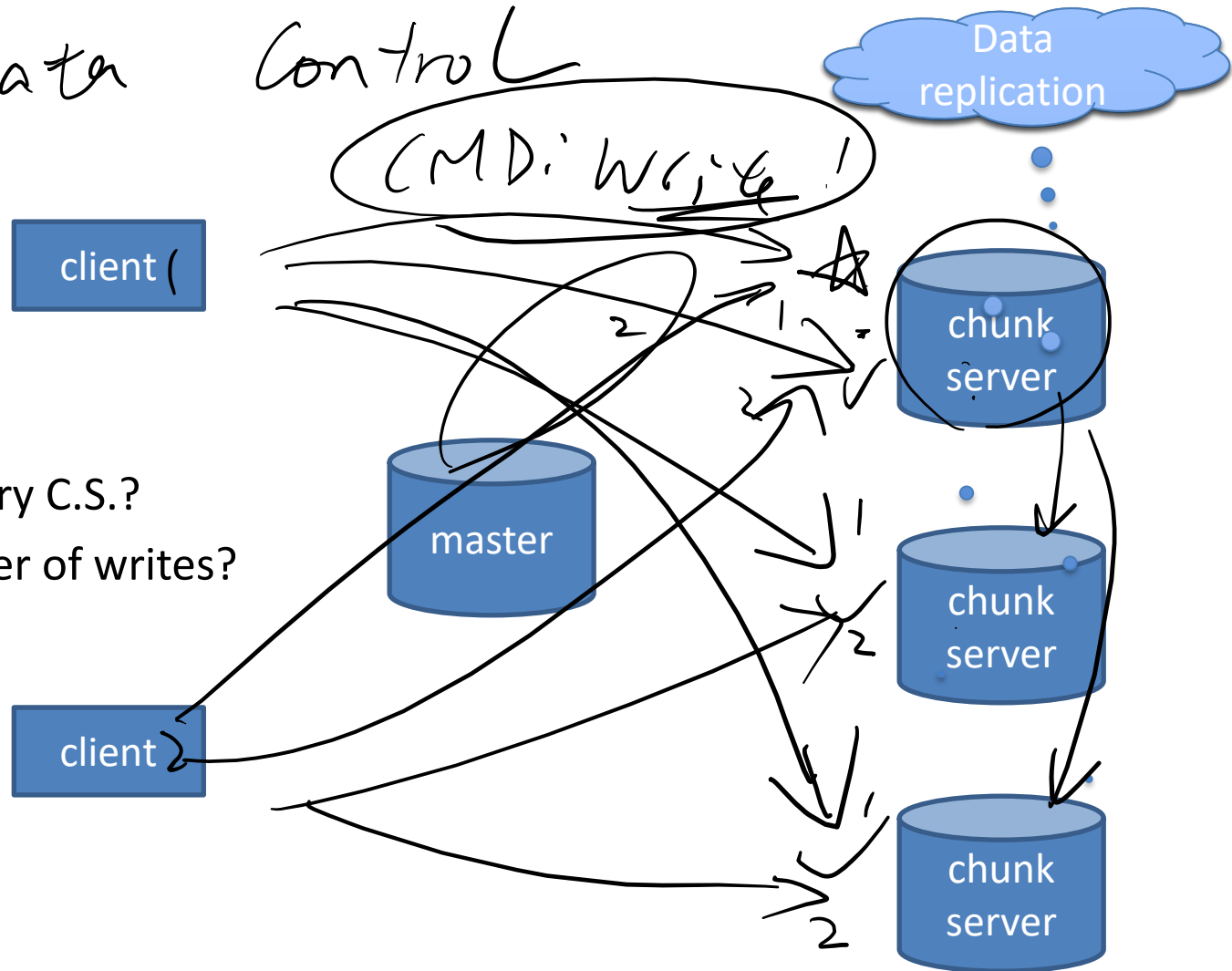
master

chunk server

client)

chunk server

- Which C.S.s?
- Is there a primary C.S.?
- What is the order of writes?



Write in GFS

- Step 1: contact the master; find the chunk handle; find the chunkservers, primary server
- Step 2: propagate the data to all replicas
- Step 3: send the write request to primary
- Step 4: primary decides the order; sends command to all replicas
 - Write to 1 or write to all replicas?
 - all
 - Who decides the order among concurrent writes?
 - Primary chunkserver (i.e., the one has the lease)

Failures in GFS writes

- What if a chunkserver is down?
 - The master will know, and will create another replica on a healthy chunk-server using data on other live chunkservers

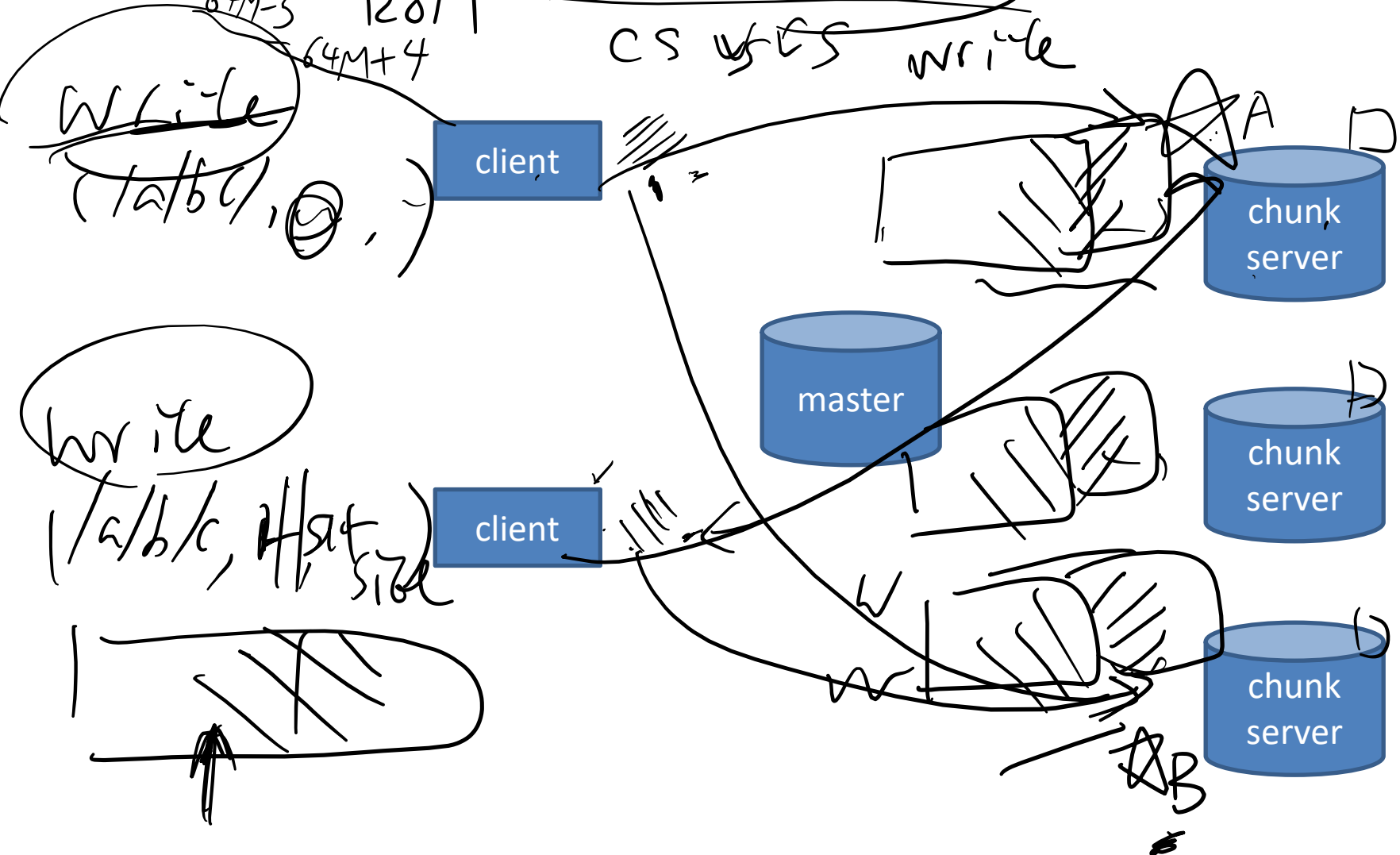
Concurrent updates in GFS

- *Consistent: every replica has the same content*
- *Defined: the content is consistent with what the client intends to write*
- Concurrent write
- Atomic append

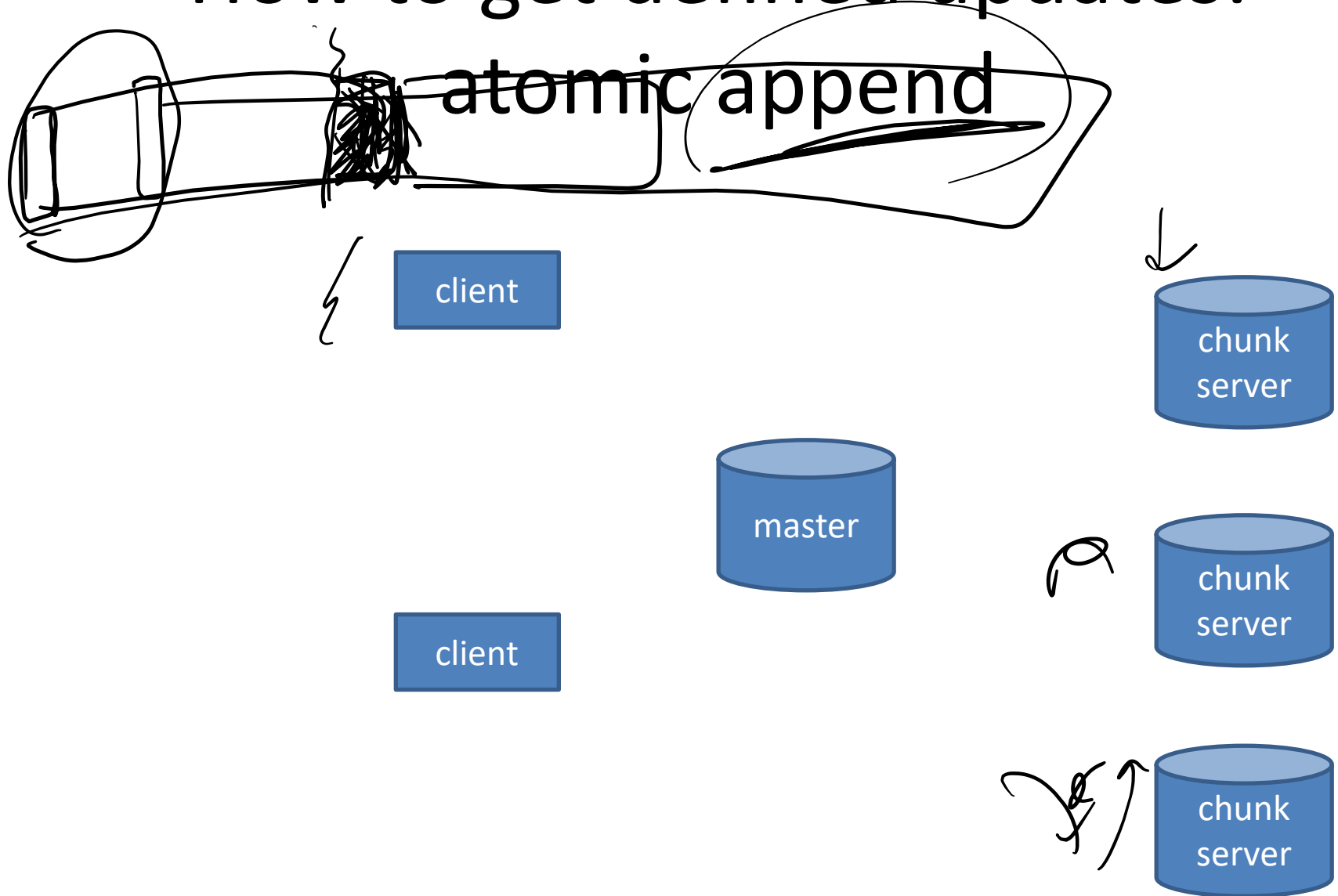
Are concurrent writes defined and consistent?

Client vs GFS

CS vs CS write



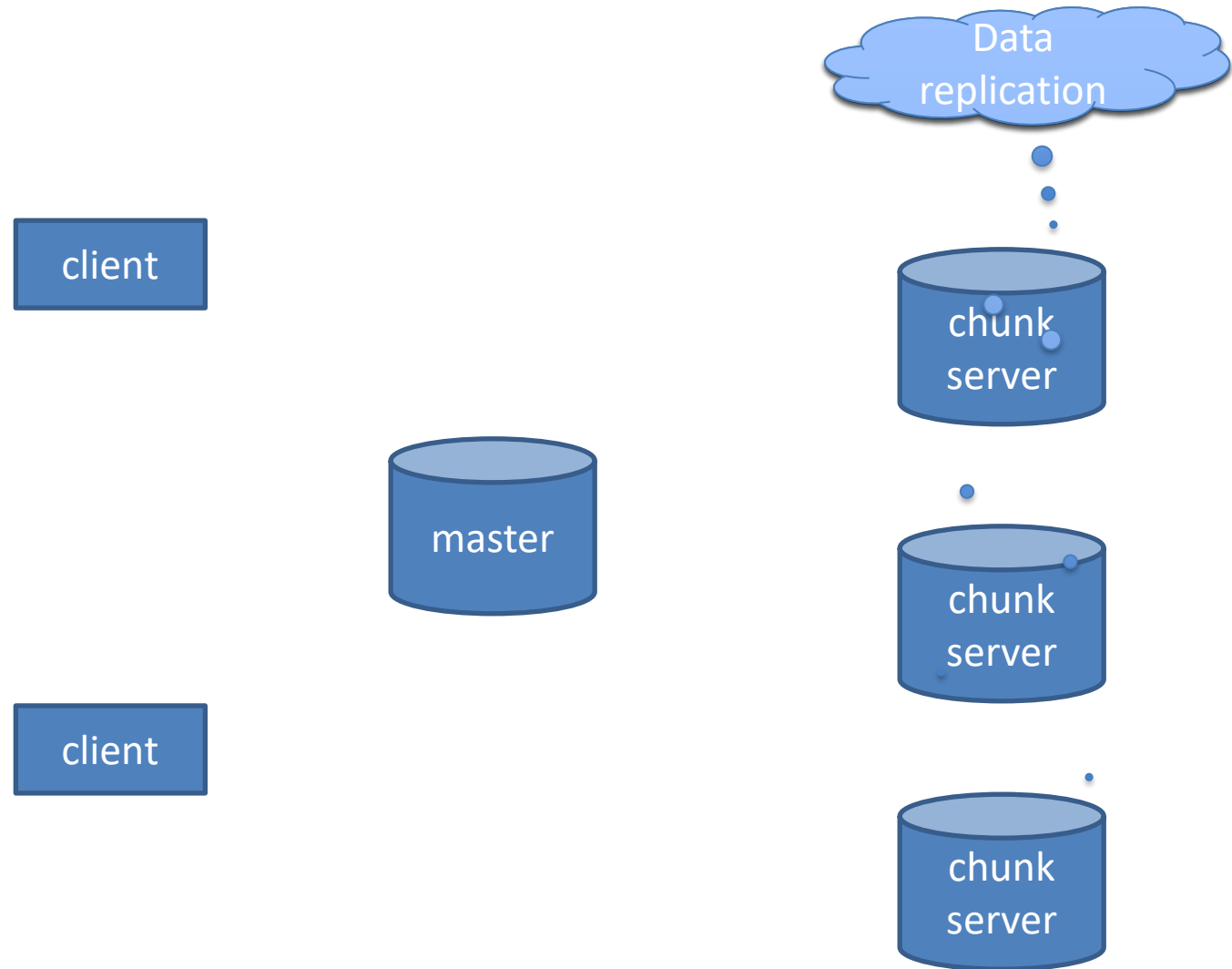
How to get defined updates:



Concurrent updates in GFS

- *Consistent: every replica has the same content*
- *Defined: the content is consistent with what the client intends to write*
- Concurrent write
 - ➔ consistent & undefined
- Atomic append
 - Step 1: (optional) padding
 - Step 2: write at primary specified location
 - Step 3: success, return to
 - ➔ inconsistent & defined

Write in GFS



Failure tolerance

- Is the master the bottleneck?
 - There is a secondary master ready to take over

Summary

- Workload affects design
- Master – chunkserver architecture