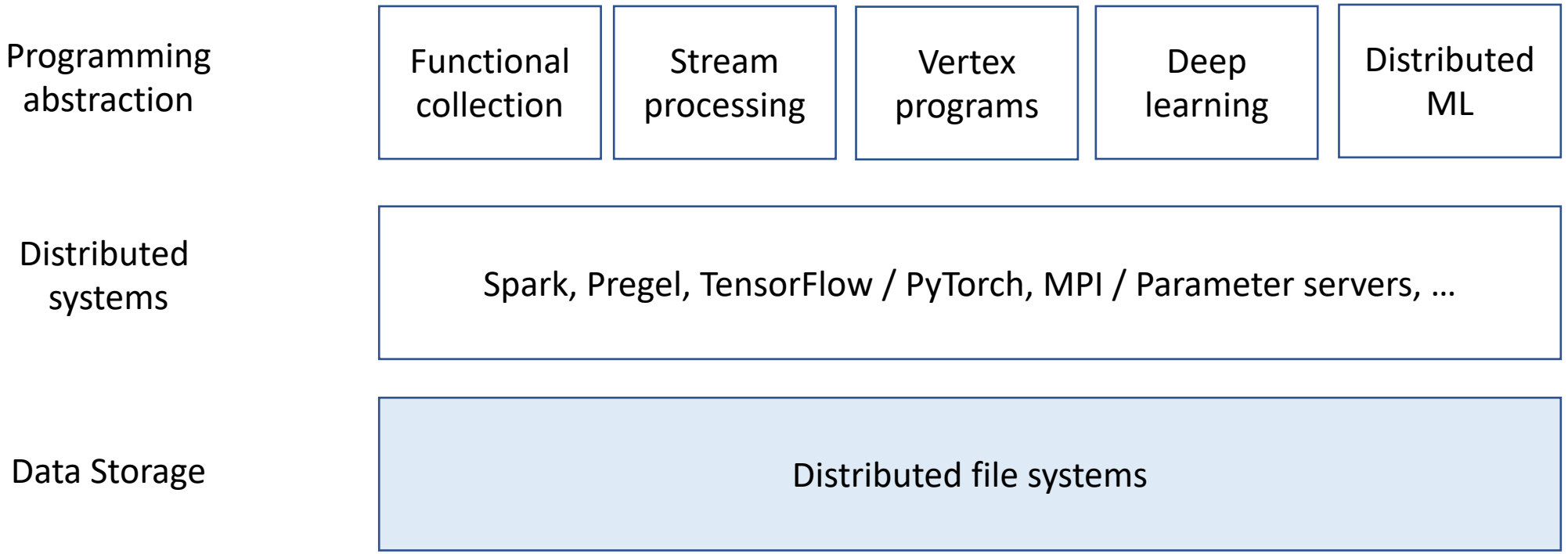
THE UNIVERSITY of EDINBURGH
informatics

Distributed File Systems

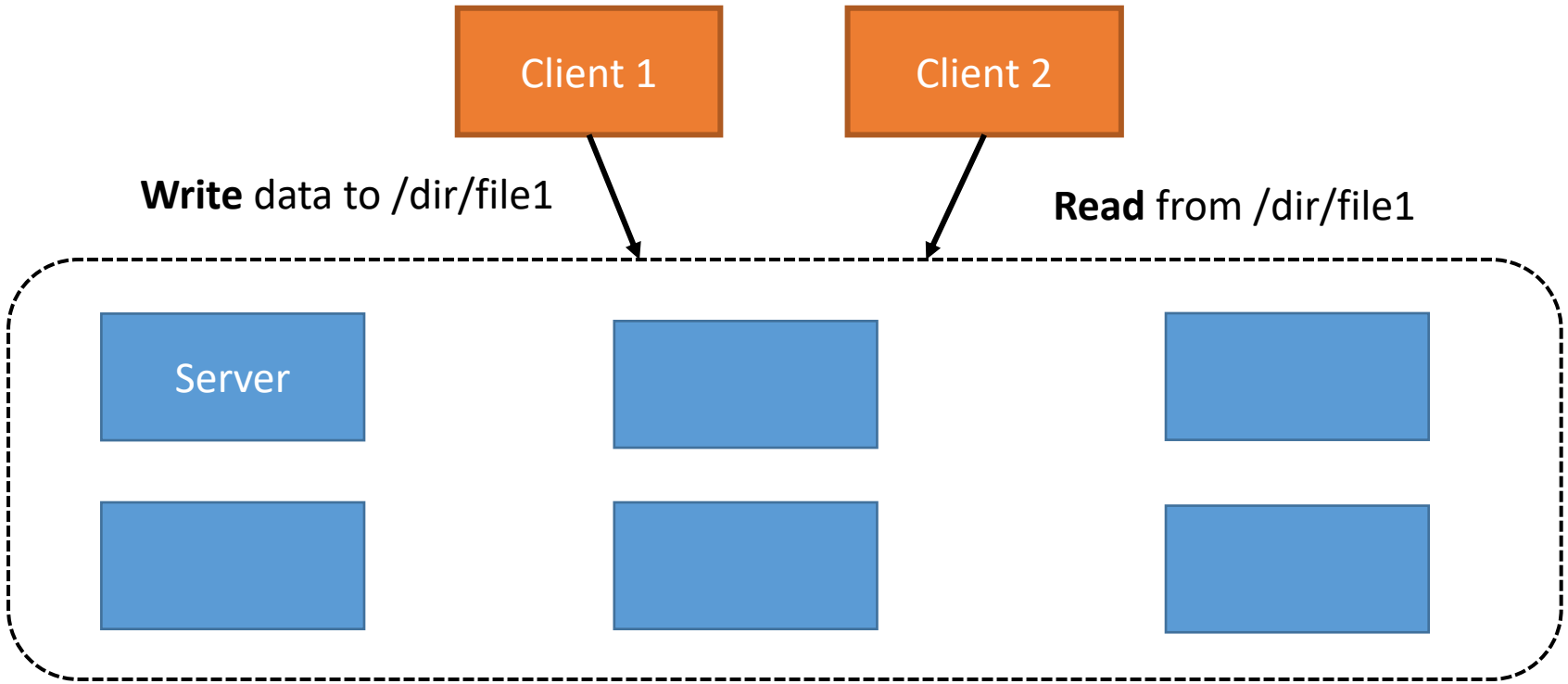
Luo Mai

University of Edinburgh

Big picture


Programming abstraction
Functional collection
Stream processing
Vertex programs
Deep learning
Distributed ML
Distributed systems
Spark, Pregel, TensorFlow / PyTorch, MPI / Parameter servers, …
Data Storage
Distributed file systems

What is a distributed file system?



1 cluster = 1,000s of **commodity servers**

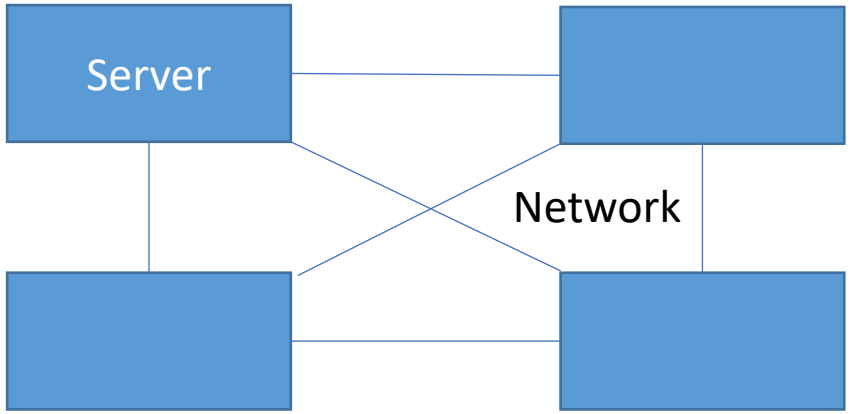
Commodity hardware

Commodity servers

- Off-the-shelve hardware (e.g., memory)
- Easy to scale horizontally

Failures are common

- Server (e.g., Disk corruption)
- Network (e.g., Switch fault)
- Software bugs (e.g., OS bug)
- Human errors

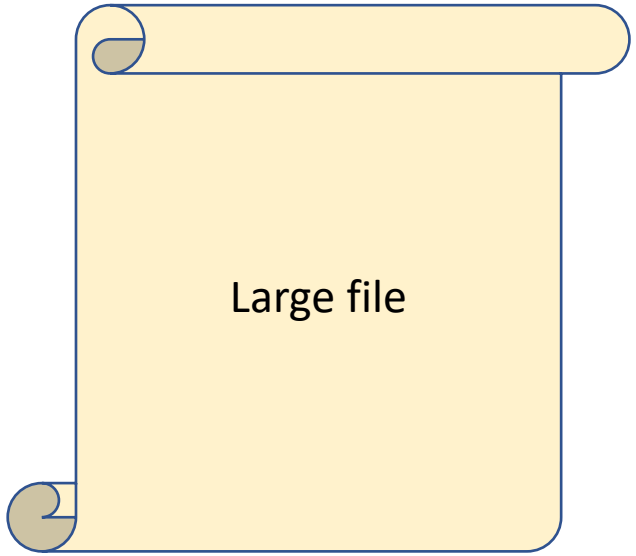


Commodity cluster

Large files

File sizes are up to multi-TB

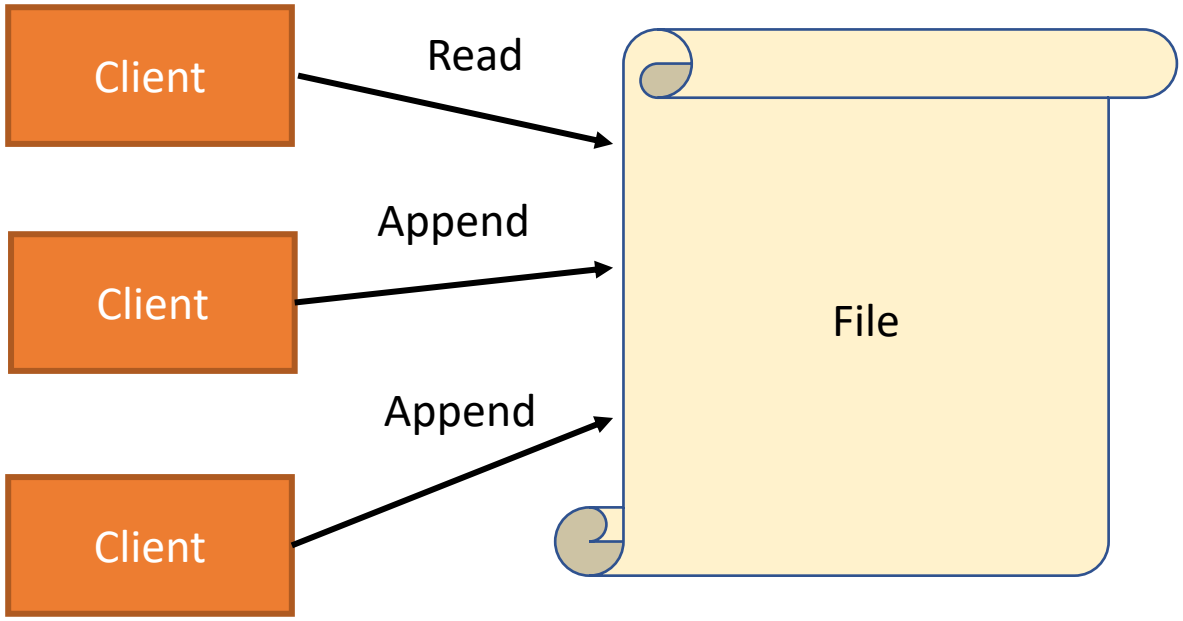
- Web documents
- Server logs
- ML model checkpoints
- …



File operations

Read + Append-only write

- Examples:
  - Appending Internet data
  - Reading data to build search index
- Sequential read is dominating
- Very rare random writes



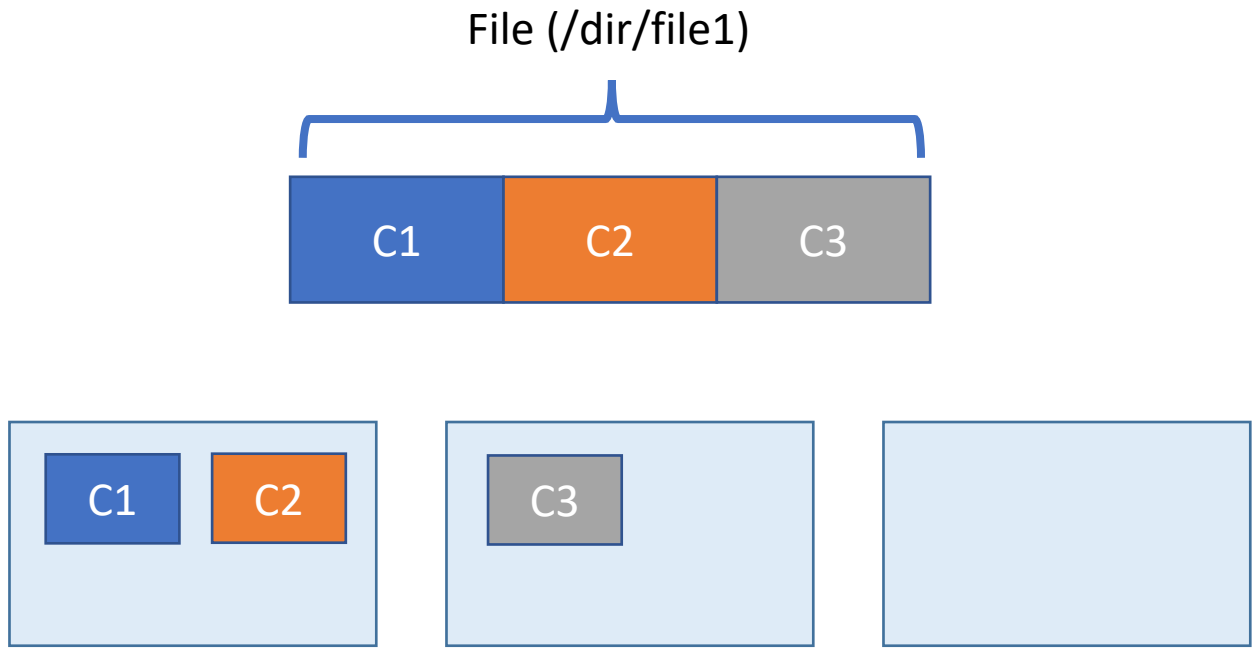
Chunks

Files split into chunks

- Each chunk 64MB
- Identified by 64-bit ID
- Stored in Chunkserver

Bring computation to storage

- Data locality
- Chunk servers also serve as compute servers



Chunks of a single file are stored on multiple servers

Replicas

Robustness

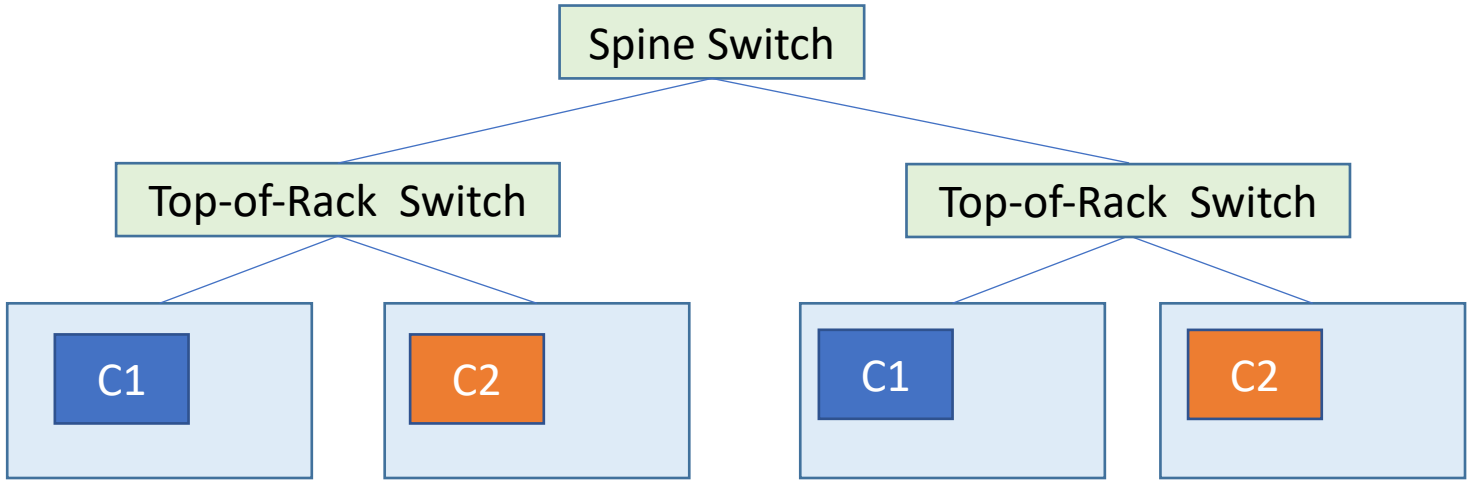
- Tolerate failures

Replicas

- Throughput vs. Space cost

Rack-level replica

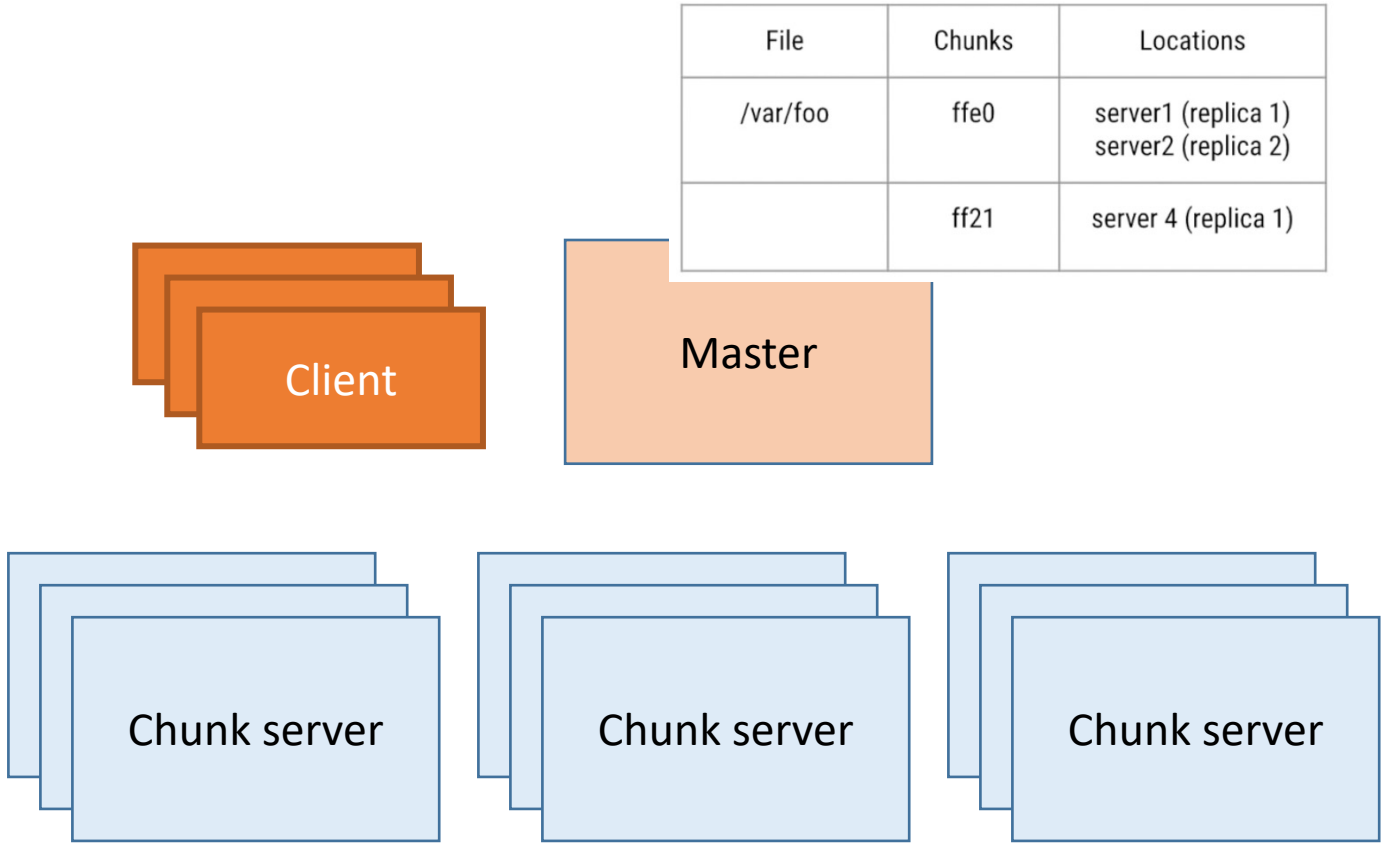
- Top-of-Rack switch fault



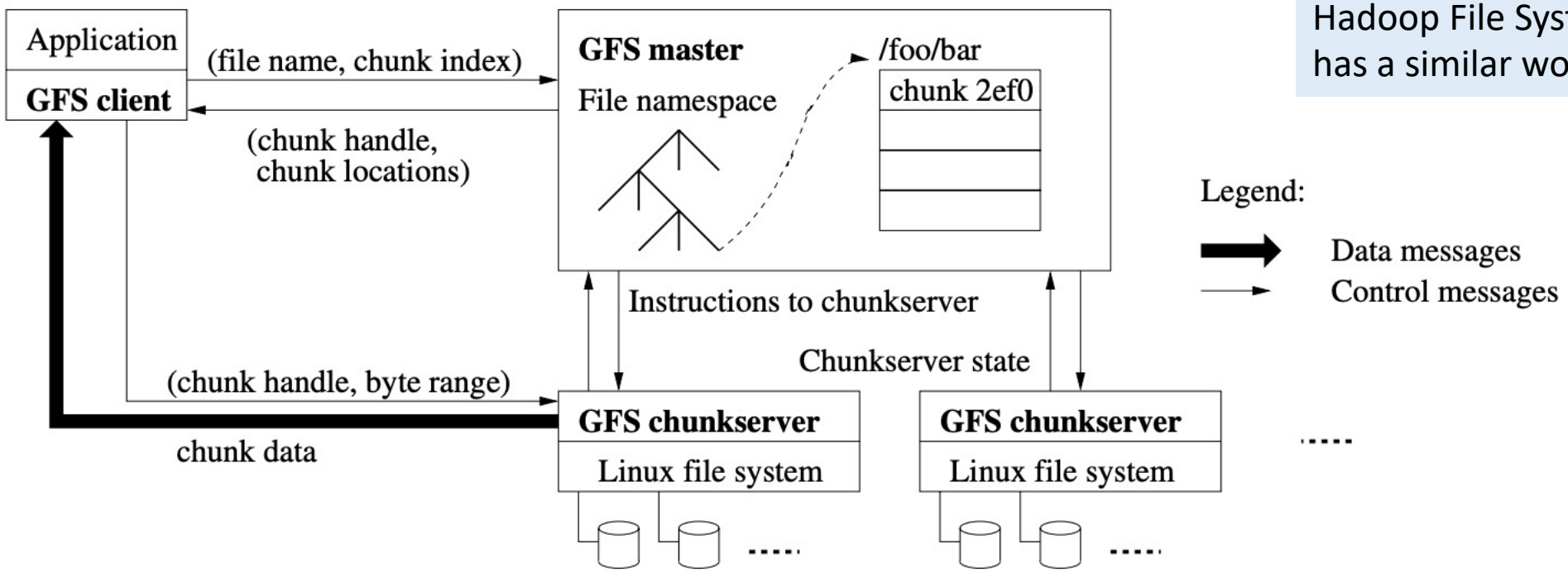
Questions?

Master

- Store server information
  - Liveness of servers
- Store chunk information
  - File namespace
  - File-chunk mapping table
- Access control



Reads

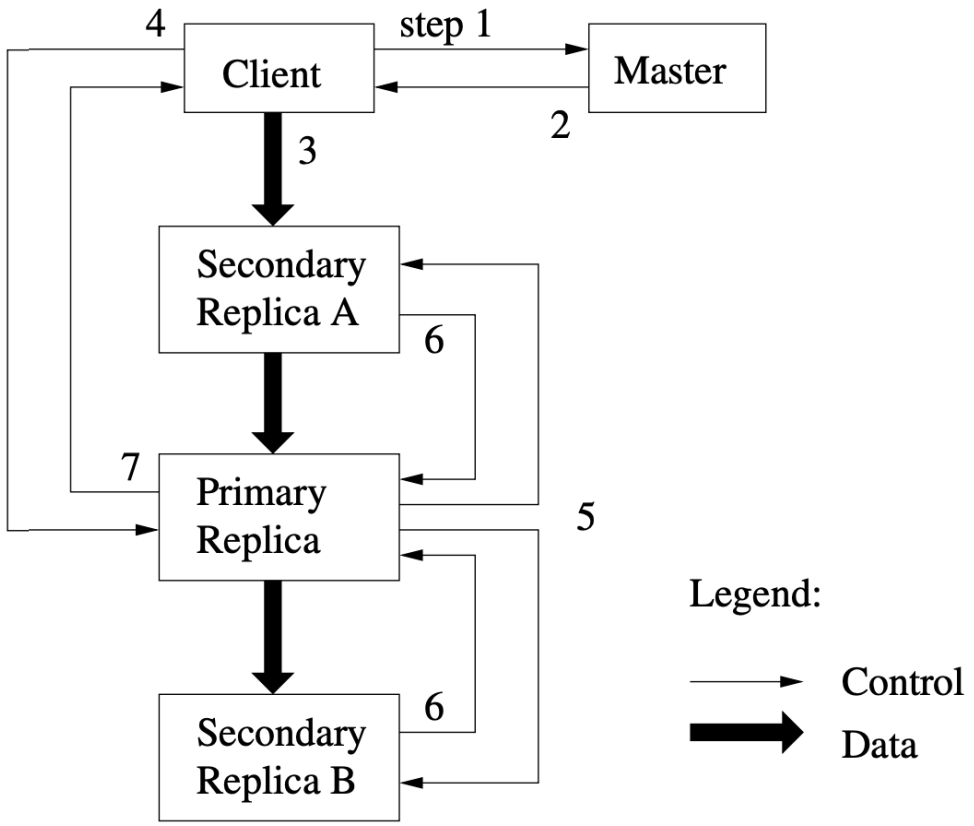


Hadoop File System (HDFS) has a similar workflow

[1] The Google File System, 2003

Writes

1. Ask for locations to write
2. Get replica location
3. Write data to the cache buffers of all replicas
4. Request commit to primary
5. Primary coordinates all replicas to apply write
6. All replicas acknowledge
7. Primary ack to client

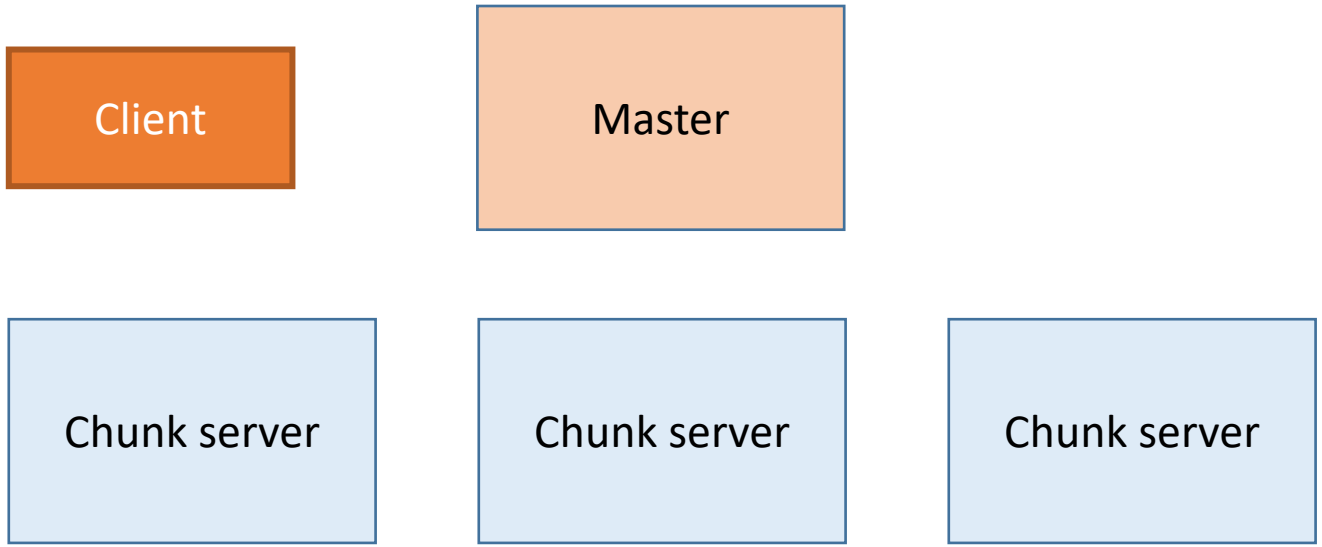


[1] The Google File System, 2003

More master performance

Avoid single-node bottleneck

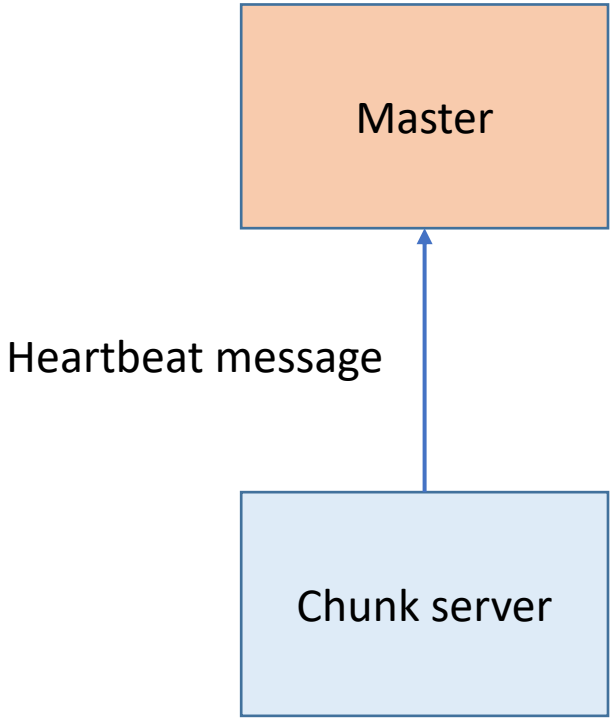
- Large chunk size
- Reduced meta data
- Reduced client interaction
  - Client caches location data



Questions?

Heartbeats

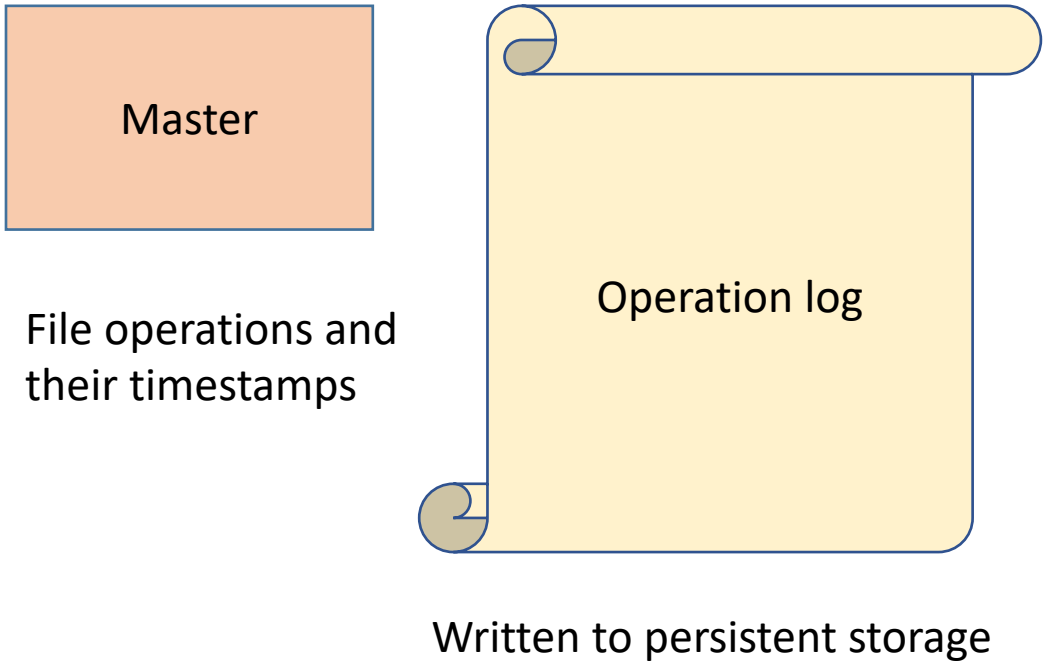
- Regular heartbeats to ensure chunk servers are alive
- Ensure replica count when recovering chunk servers



Operations log

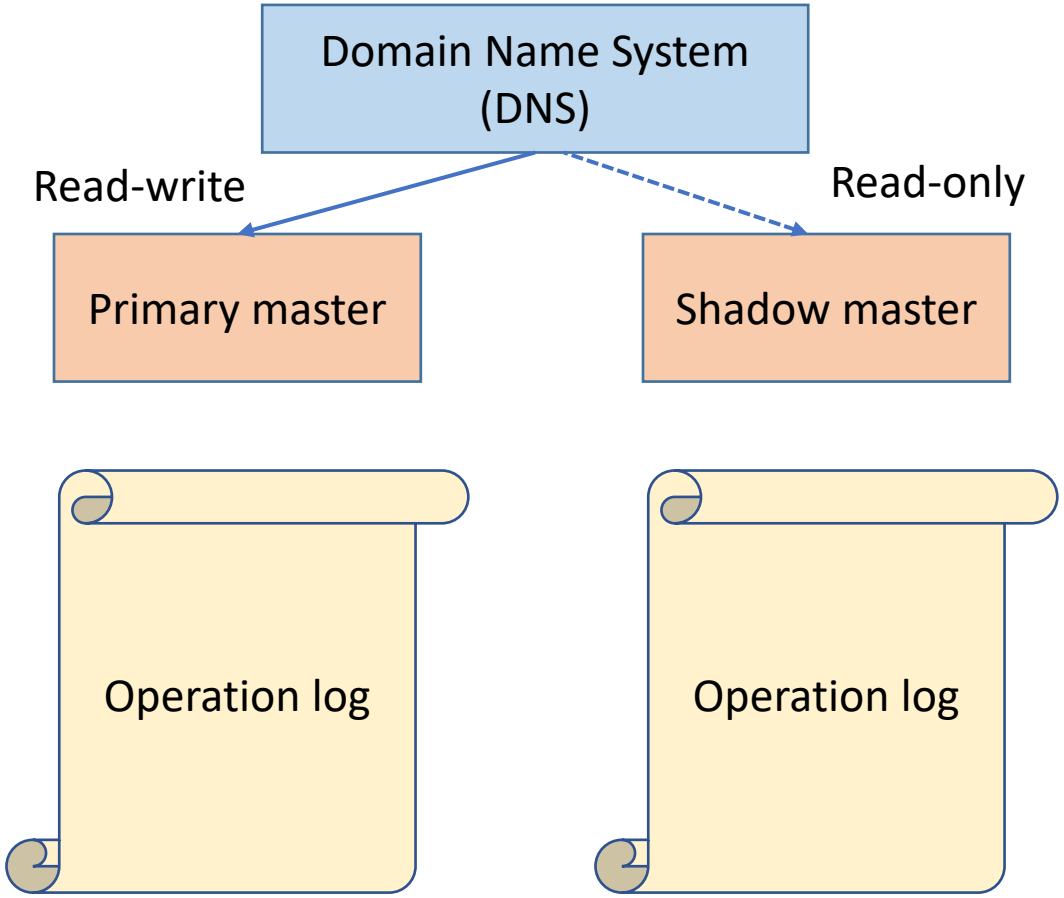
Tracking master operations

- Checkpoint regularly
- Happens in background thread
- Used if master crashes
- Rebooted master replay log



Master recovery

- Recover primary locally (instantly) or elsewhere (small delay)
- **Shadow master** serve read-only requests when the master is being recovered
  - Shadow master may lag slightly



Summary

- Commodity hardware
- Large files
- Chunks & Replicas
- Read + append write
- Performance
  - Reducing client request overhead and interaction
  - Decouple control flow and data flow
- Robustness
  - Chunk server: Heartbeat
  - Master: Operations log, shadow masters

Reading

- Compulsory reading
  - The Google File System
- Optional reading
  - Hadoop design

Guest lecture announcement

- Laurence Moroney on AI: Breakthroughs, opportunities and the future
  - Speaker: Laurence Moroney, AI Advocacy Lead of Google
  - Time: 21/11/2022, 1pm – 2pm
  - Book your place for in-person attendance: https://www.eventbrite.co.uk/e/laurence-moroney-on-ai-breakthroughs-opportunities-and-the-future-tickets-439017060847
  - Virtual attendance available via Zoom

- We may have another guest lecture in the week of 2nd December
  - Speaker to be confirmed

Questions?