IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Lecture 40: Advanced Topics: The Google File System

Agenda

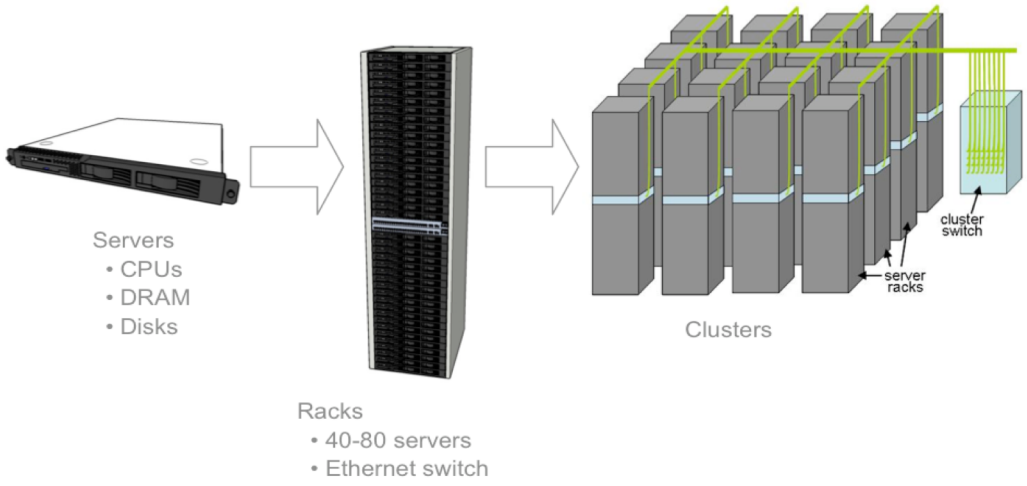
- Recap
- Advanced Topics
  - The Google File System

Recap

- What's cloud computing
  - Informal:
    - computing with large datacenters
    - users generally do not even know the exact location of "their" resources or even which country they are located in
  - NIST's five essential characteristics
- Types of cloud
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)
  - Public vs. private clouds

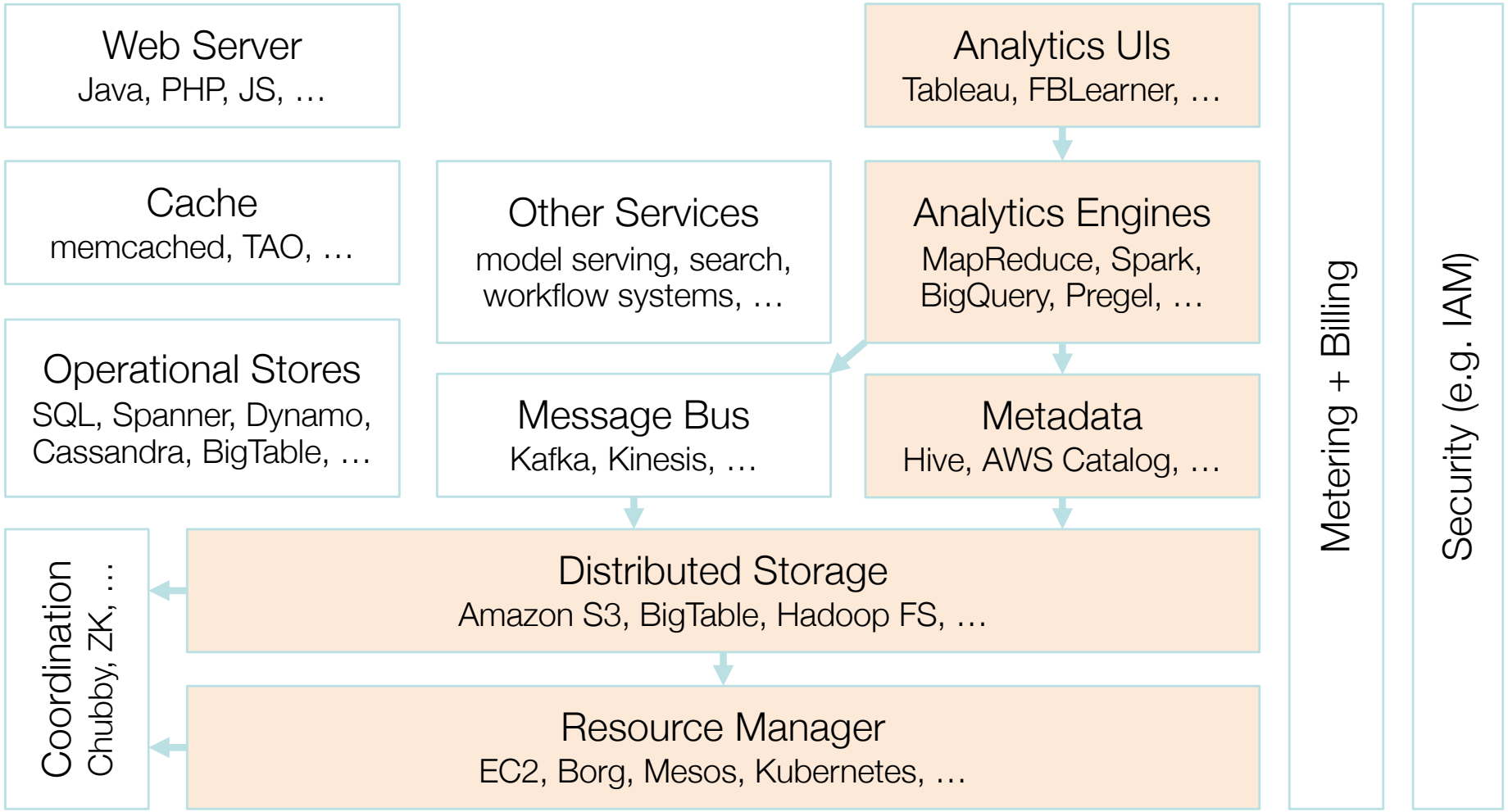
Recap

- Cloud Hardware
  - "Warehouse-Scale Computer"
    - Datacenter as a computer
      - Rows of rack-mounted servers
      - Often organized as a few mostly independent clusters



Recap

- Cloud Software



Agenda

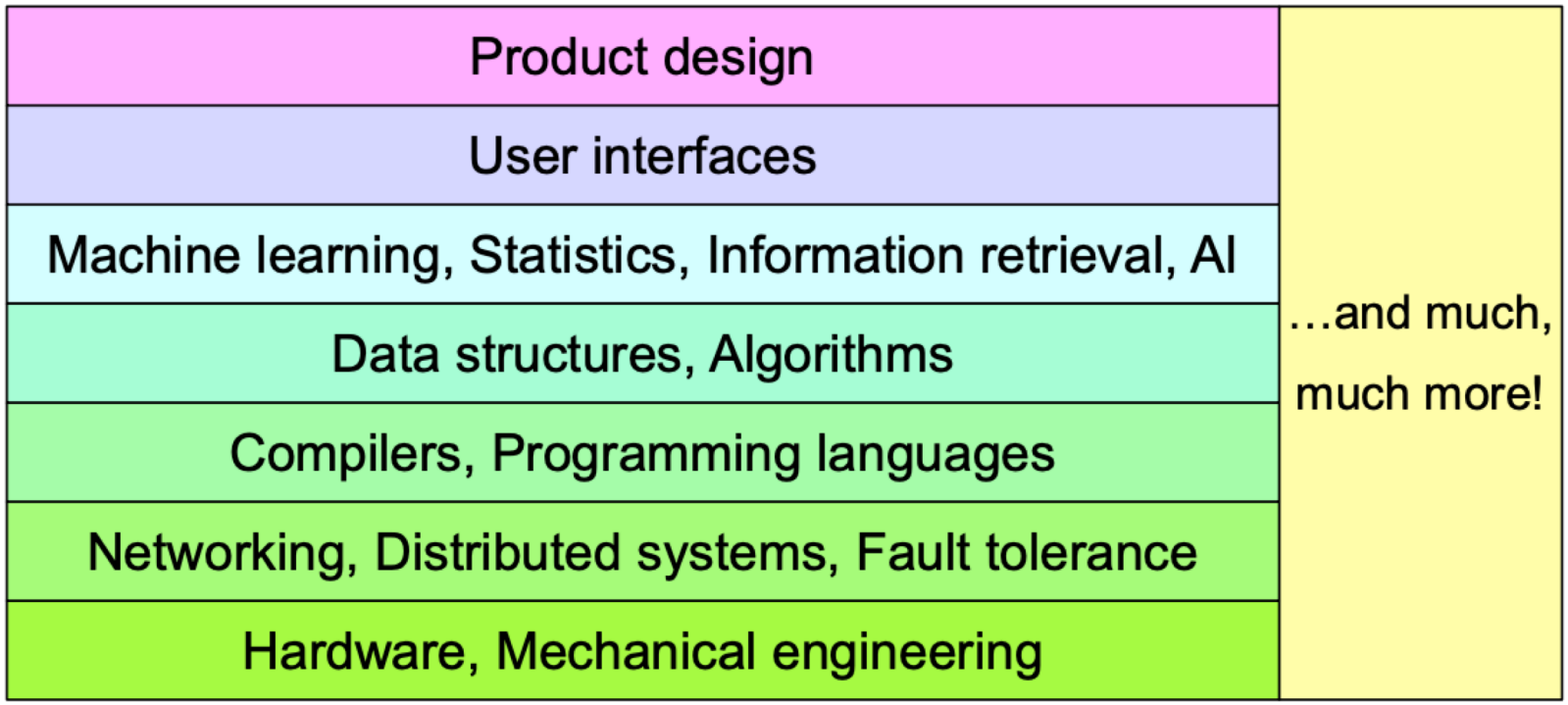
- Recap
- The Google File System

User's View of Google

- Search engine, Gmail, Google Maps, Ads, ...

ECpE Students' View of Google (ideally)

- A wide range of knowledge/techniques/problems/opportunities



Hardware Design Philosophy

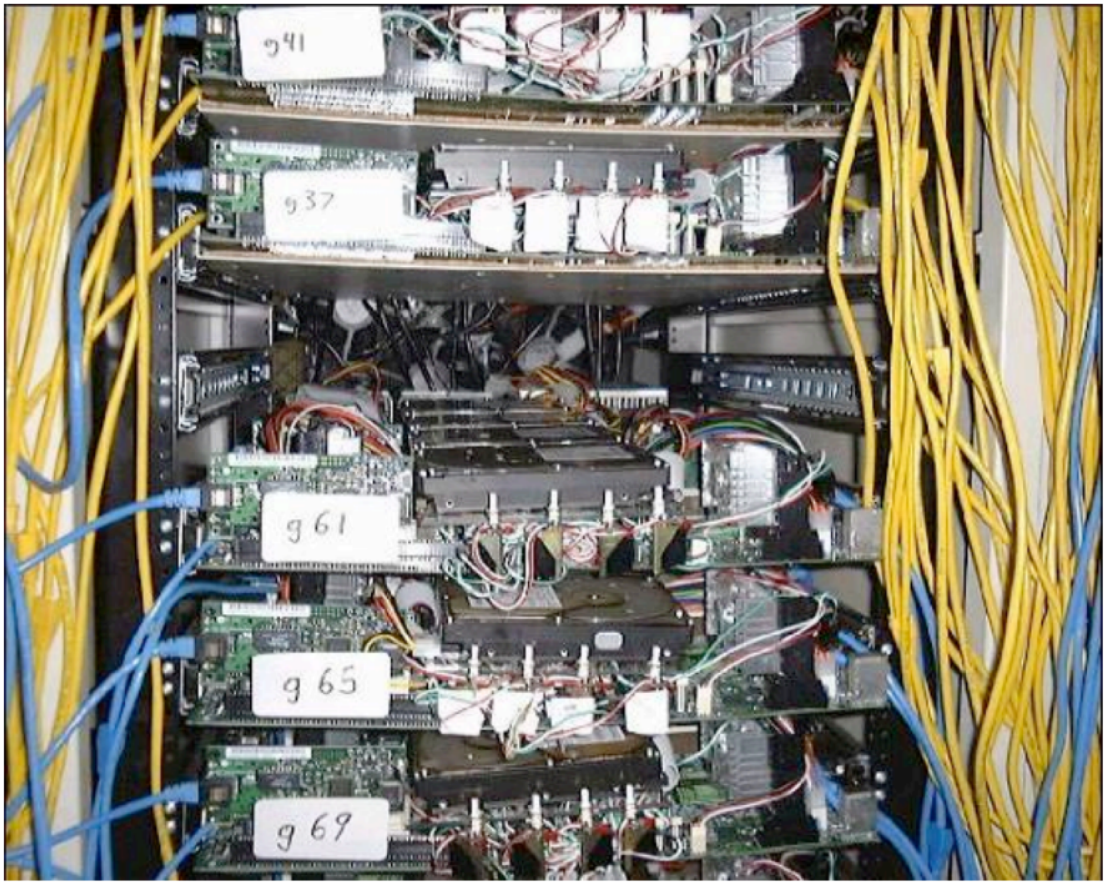
- Google prefers low-end server/PC-class designs
  - Build lots of them!
- Why?
  - Single machine performance is not interesting
    - Even smaller problems at Google are too large for any single system
    - Large problems have lots of available parallelism
- In contrast to the supercomputer designs in national labs
  - “Scale out” vs. “Scale up”

“Google” Circa 1997

- Started by 2 graduate students at Stanford University

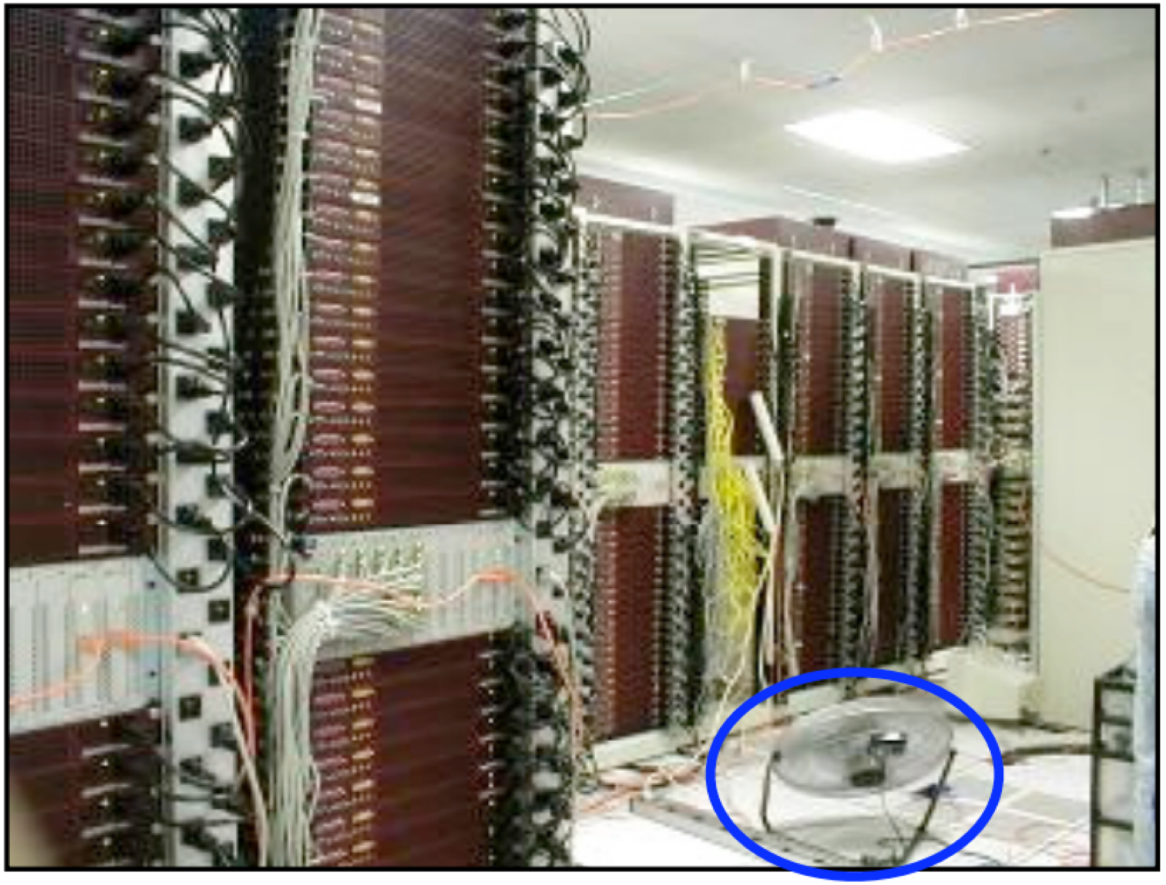
Google Circa 1999

- Multiple blade servers



Google Circa 2000

- “Data center”



Google Circa 2001

- New Data Center
  - in three days ...

Google Today

- Multiple data centers around the world

The Joy of Real Hardware at Scale

- Typical first year for a new cluster:
  - ~0.5 overheating (power down most machines in< 5 min, ~1-2 days to recover)
  - ~20 rack failures (40-80 machines instantly disappear)
  - ~1 PDU failure (~500-1000 machines suddenly disappear)
  - ~0.5 overheating (power down most machines in <5 mins)
  - ~5 racks go wonky (40-80 machines see 50% packet loss)
  - ~8 network maintenances (4 might cause connectivity losses)
  - ~1000 individual machine failures
  - ~thousands of hard drive failures
  - ~slow disks, bad memory, etc.

- Long-haul networking breaks for unusual reasons, too
  - Wild dogs, dead horses, thieves, blasphemy, drunken hunters and sharks

Implications

- Stuff Breaks
  - If you have one server, it may stay up 1,000 days
  - If you have 10,000 servers, expect to lose ten a day
- "Ultra-reliable" hardware doesn't really help
  - At large scales, super-fancy reliable hardware still fails, albeit less often
    - software still needs to be fault-tolerant
    - commodity machines without fancy hardware give better perf/$

Implications

- Stuff Breaks
  - If you have one server, it may stay up 1,000 days
  - If you have 10,000 servers, expect to lose ten a day
- "Ultra-reliable" hardware doesn't really help
  - At large scales, super-fancy reliable hardware still fails, albeit less often
    - software still needs to be fault-tolerant
    - commodity machines without fancy hardware give better perf/$
- **"Failures are the norm rather than exception"**
  - Reliability has to come from the software

The Google File System

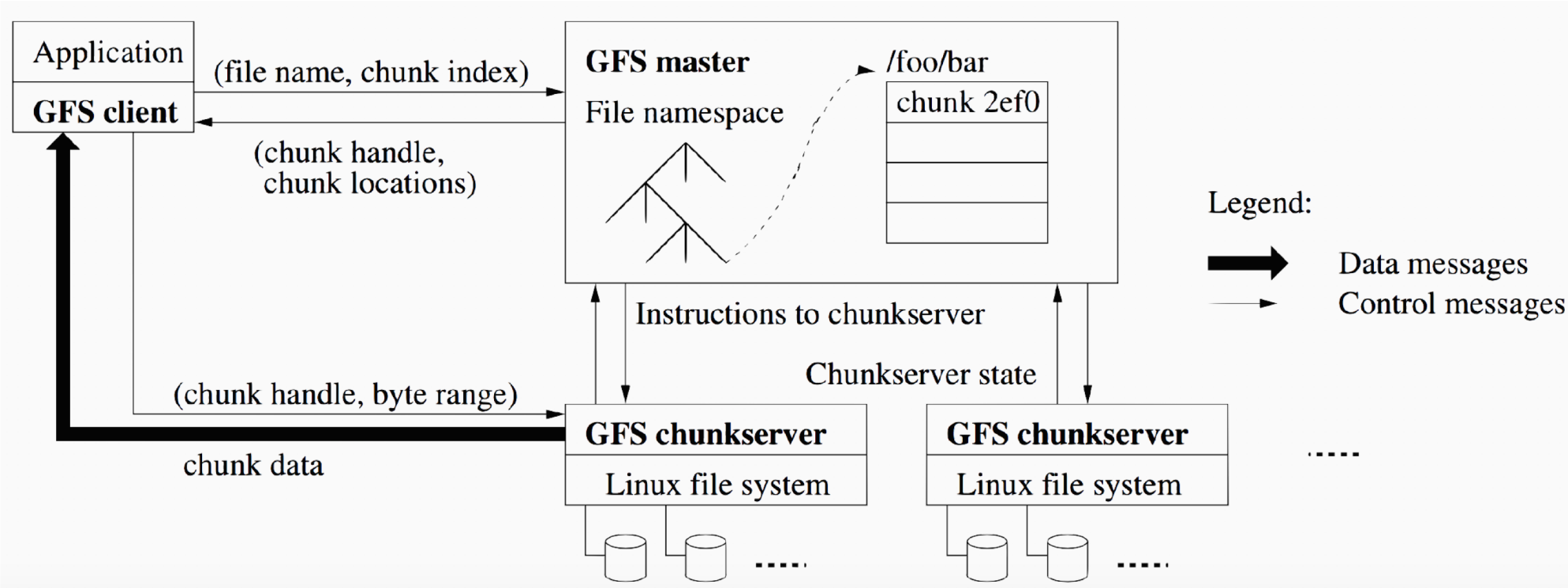
- One key infrastructure empowering Google
- Major Observations
  - Component failures are normal
    - due to the scale
  - Files are huge
    - by tractional standards (many GB/TB)
  - Most files are modified by appending at the end
    - Random writes (and overwrites) are practically non-existent
  - Co-design applications and the file system benefits the overall system a lot
    - increasing flexibility

The Google File System

- Interface
  - Does not implement POSIX
  - Familiar
    - Create, delete, open, close, read, write
  - Novel
    - snapshot
    - record append
      - guarantee atomicity with multiple concurrent writes

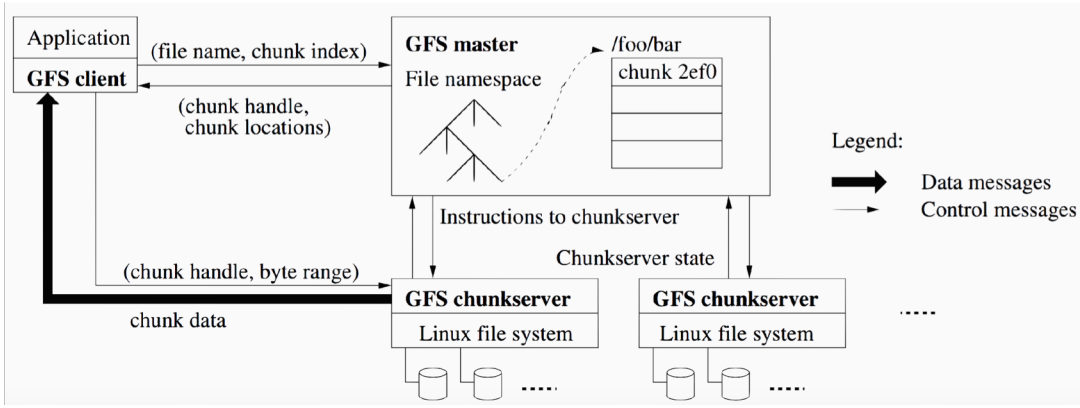
The Google File System

- Architecture
  - three types of nodes
    - master, chunkservers, clients



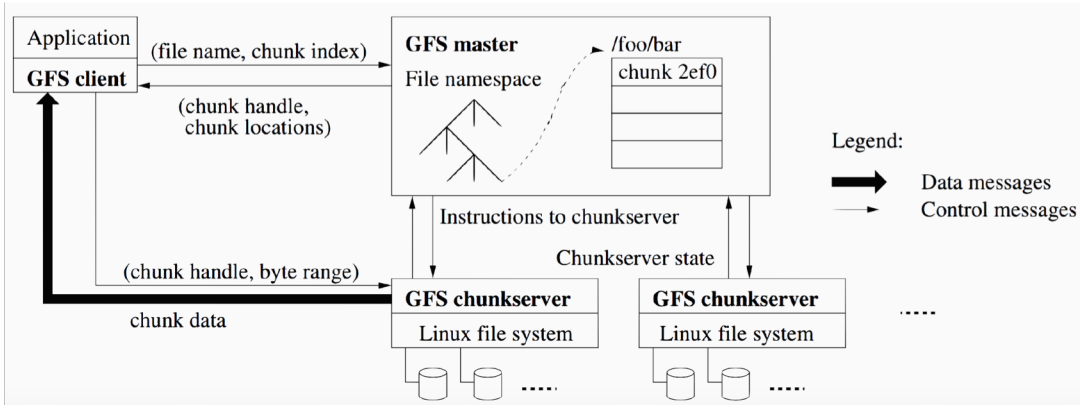
The Google File System

- Chunkservers
  - store all user files
    - in fixed-size chunks
      - 64 MB by default
      - each is identified by a 64 bit unique handle
  - 3-way replication
    - each chunk is replicated on three different chunkservers



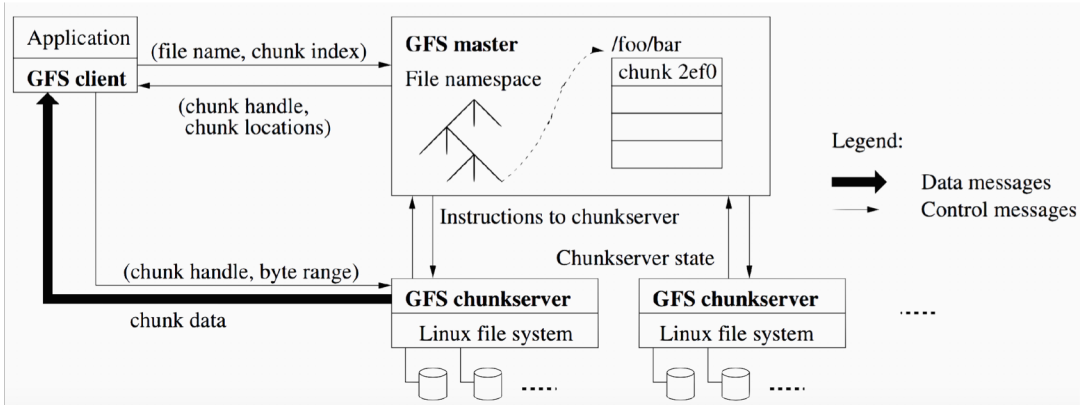
The Google File System

- Master
  - store all metadata
    - namespace
    - access-control information
    - chunk locations
    - “lease” management
      - define global mutation order
  - heartbeats
  - single master
    - simplifies design



The Google File System

- Clients
  - GFS code implements APIs
  - Cache only metadata

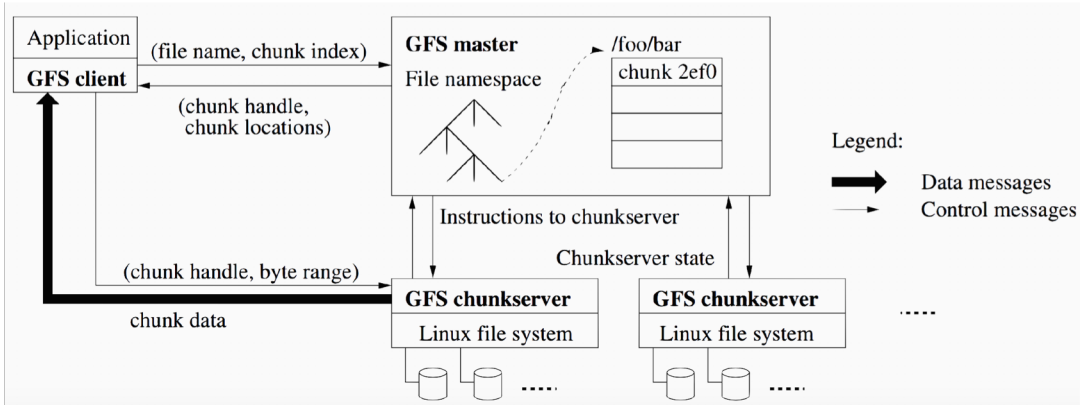


The Google File System

- Workflow
  - client: using fixed chunk size, translate filename & byte offset to chunk index, send request to master
  - master: replies with chunk handle & location of chunkserver replicas (including which is “primary”)
  - client: cache info using filename & chunk index as key
  - client: request data from nearest chunkserver
    - no need to talk more about this 64MB chunk until cached info expires or file reopened
    - often initial request asks about sequence of chunks

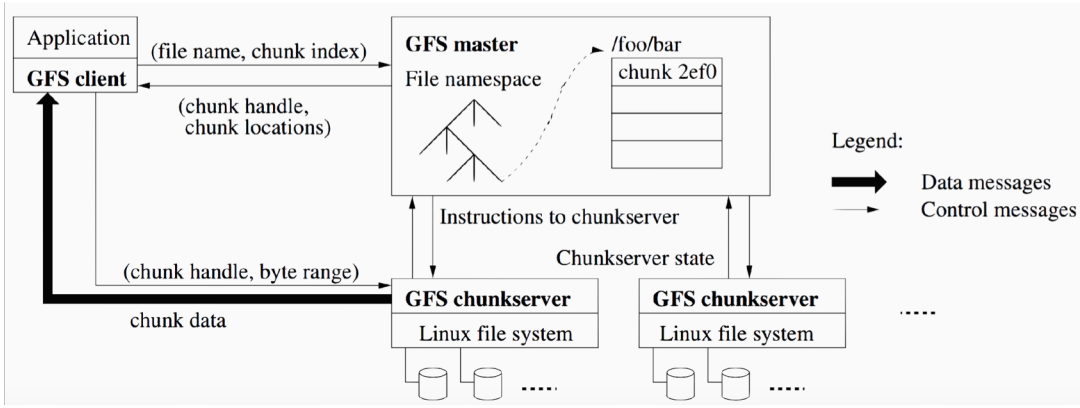
The Google File System

- Metadata
  - three types stored on master
    - file & chunk namespaces
    - mapping from files to chunks
    - location of chunk replicas
  - stored in memory
  - kept persistent through logging



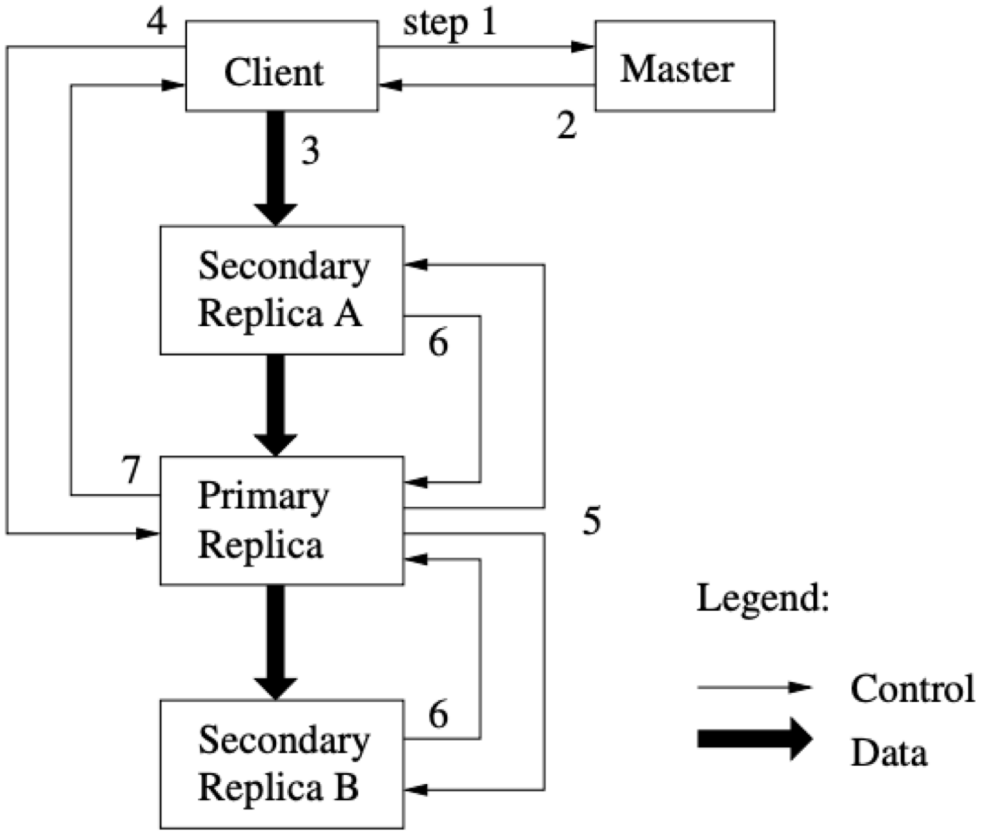
The Google File System

- Lease and Mutation Order
  - Master grants “lease” to one replica
    - called “primary” chunkserver
  - Primary serializes all mutation requests
    - communicates order to replicas



The Google File System

- Write Control and Data Flow

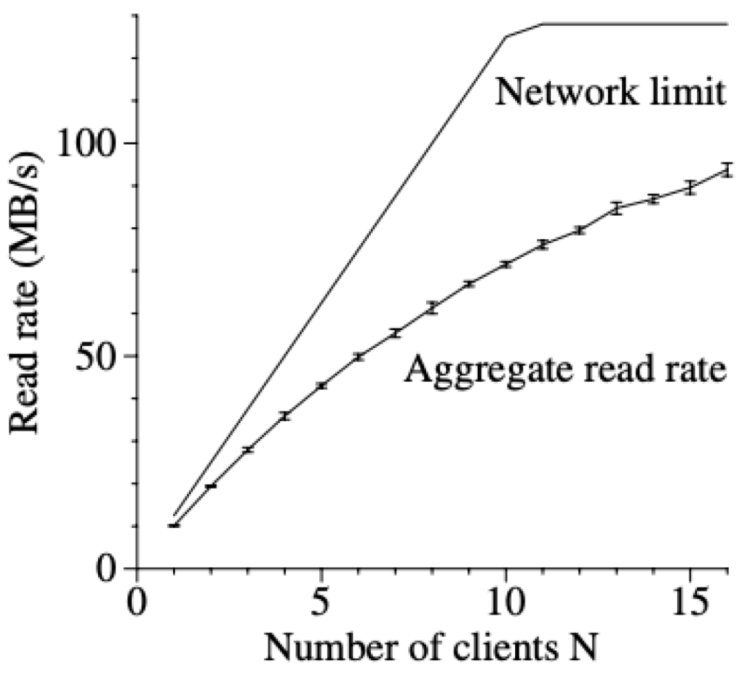


The Google File System

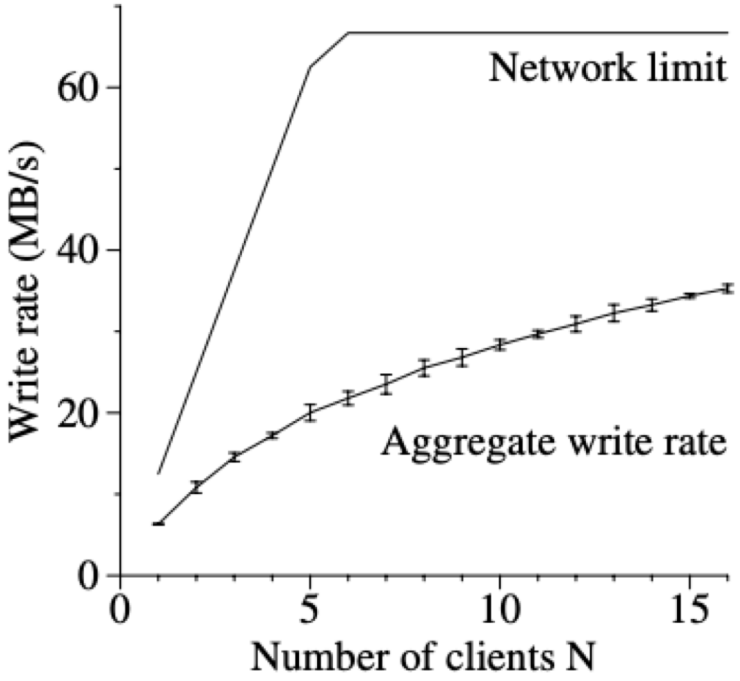
- Other issues
  - Atomic Appends
  - Fast snapshot
  - Master operation
    - namespace management & locking
    - replica placement & rebalancing
    - garbage collection (deleted/stale files)
    - detecting stale replicas
  - Master replication
    - master log & checkpoints replicated
    - shadow masters
      - provide read-access when primary is down

The Google File System

- Read & Write Performance
  - scalable



(a) Reads



(b) Writes

Agenda

- ~~Recap~~
- ~~Advanced Topics~~
  - ~~The Google File System~~

Questions?

*acknowledgement: slides include content adapted from “Modern Operating Systems” by A. Tanenbaum, “Operating Systems Concepts” by A. Silberschatz etc., “Operating Systems: Three Easy Pieces” by R. Arpaci-Dusseau etc., lectures from Google, Stanford, UC Berkeley, and anonymous pictures from internet