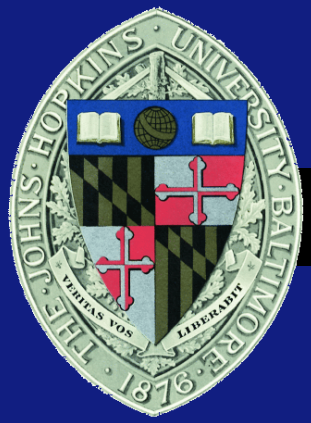
Lecture 14.3
Map/Reduce Systems

EN 600.320/420

Instructor: Randal Burns

27 March 2017



Department of Computer Science, *Johns Hopkins University*

System Issues

- What kind of problems arise with which the runtime system needs to deal?

System Issues

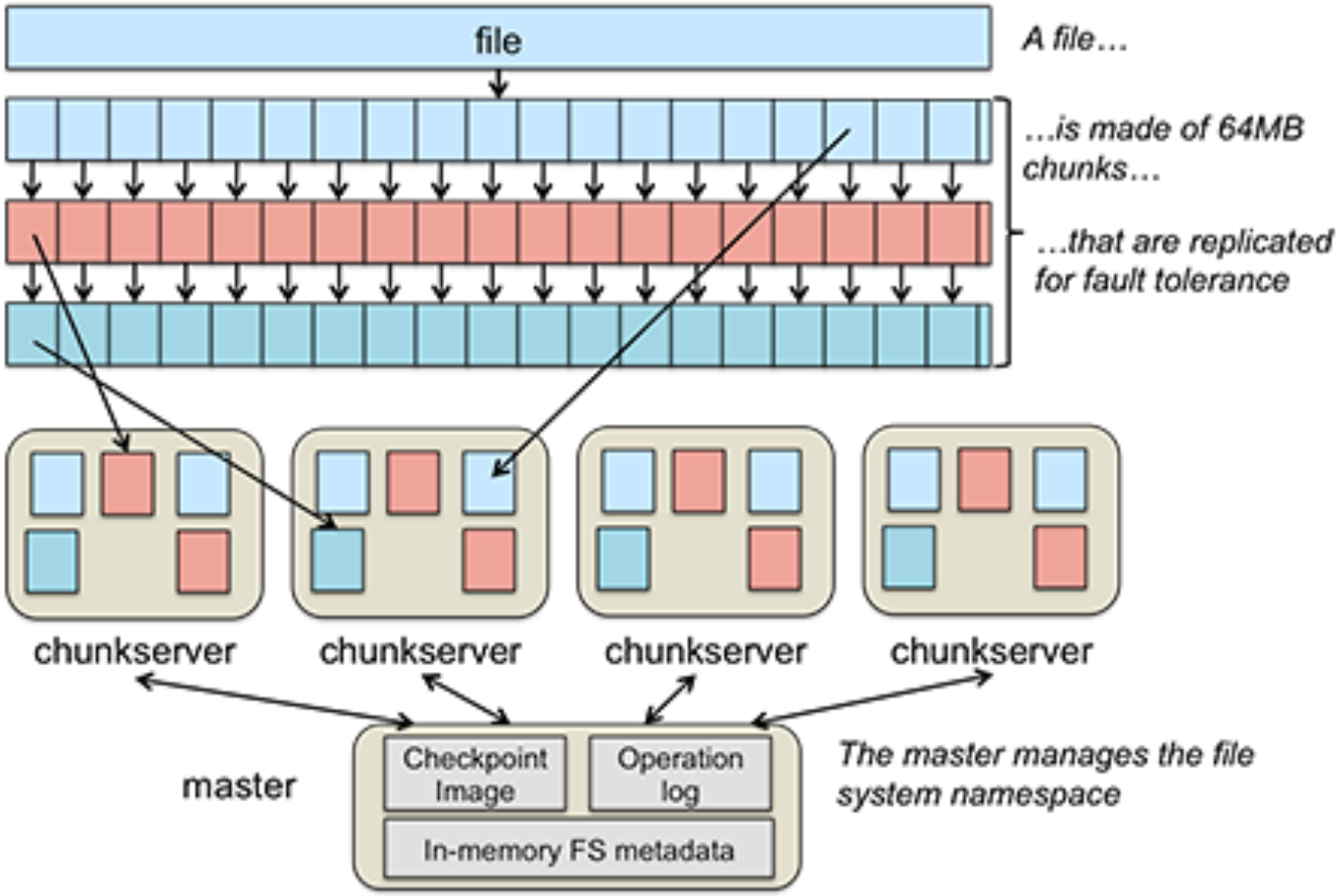
- Master failure
  - Checkpoint/restart, classic distributed systems/replication problem
- Failed worker
  - Heartbeat liveness detection, restart
- Slow worker
  - Backup tasks
- Locality of processing to data
  - Big deal, they don't really solve
- Task granularity
  - Metadata size and protocol scaling (not inherent parallelism) limit the size of *M* and *R*

Google File System

- GFS
- Same goals
  - Wide-distribution
  - Commodity hardware
  - High (aggregate) performance
- Different environment?
  - Component failures are normal behavior
  - Files are huge (new to Google environment ca. 2004)
  - Most files have append-only writing,
    - Mandate append-only writing to realize good I/O properties
    - *Why append only?*

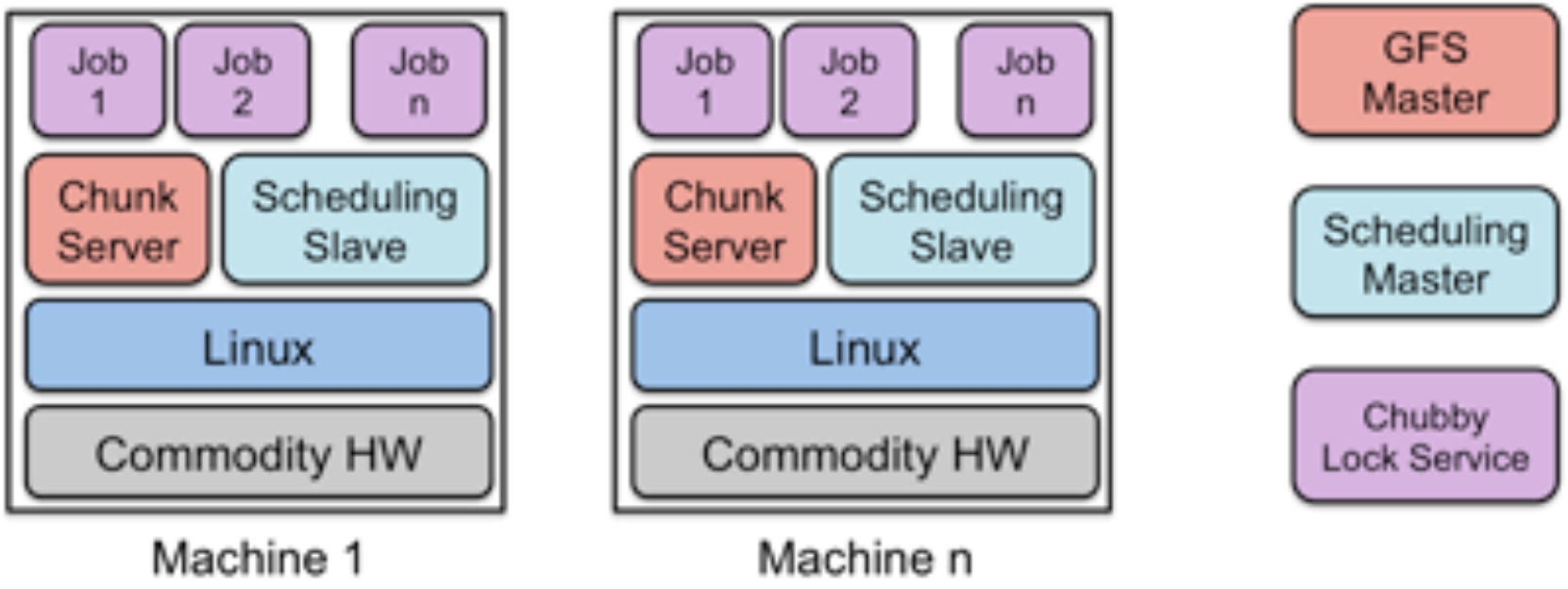
Google File System

- Data design (from https://www.cs.rutgers.edu/~pxk/417/notes/16-dfs.html)



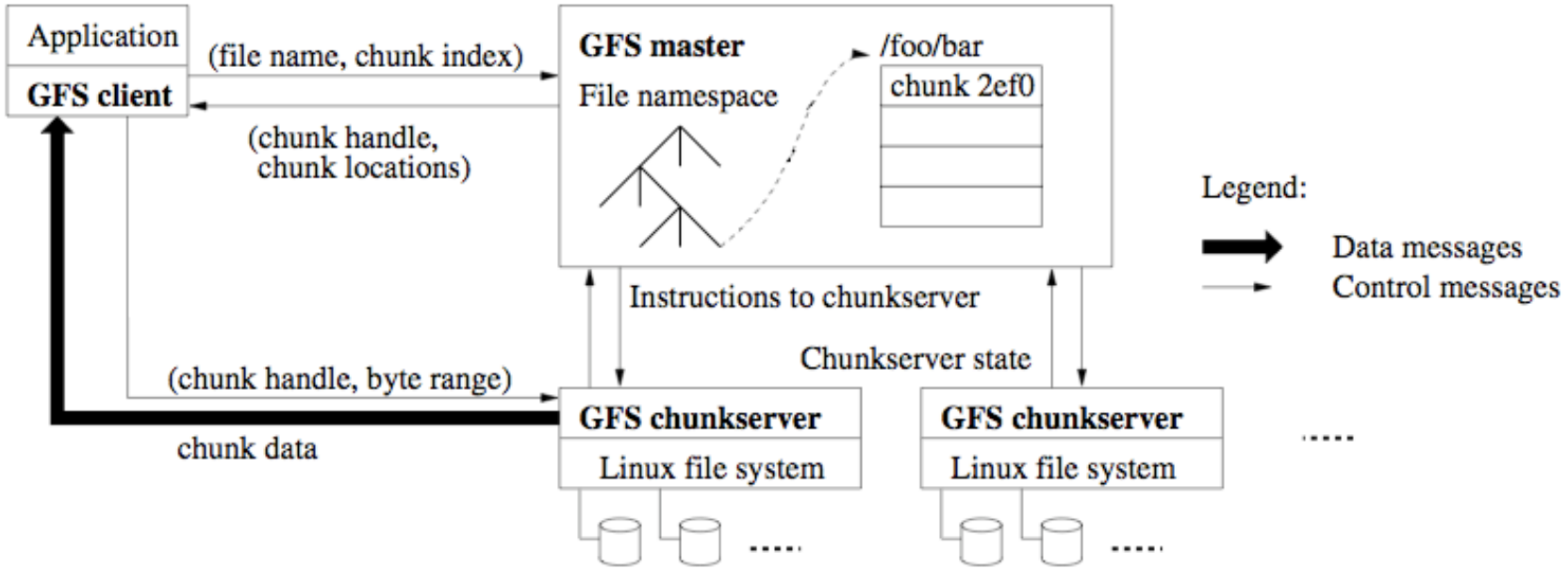
Google File System

- Cluster design (from https://www.cs.rutgers.edu/~pxk/417/notes/16-dfs.html)
  - All services are distributed and reliable



Architecture

- Looks like many *out-of-band* FS



How GFS changed the world

- Atomic checkpoint and append
  - Major mode for writing
  - Great semantics for limited usage
  - Abandon POSIX file system semantics
- In-memory metadata at Master
  - Gotta keep it small
  - Even for big data (scale metadata memory in proportion to aggregate storage)
- Re-replication
  - Keep three, detect missing on read/write
  - Forget reliable storage, forget RAID
- Design for failures (not recovery)