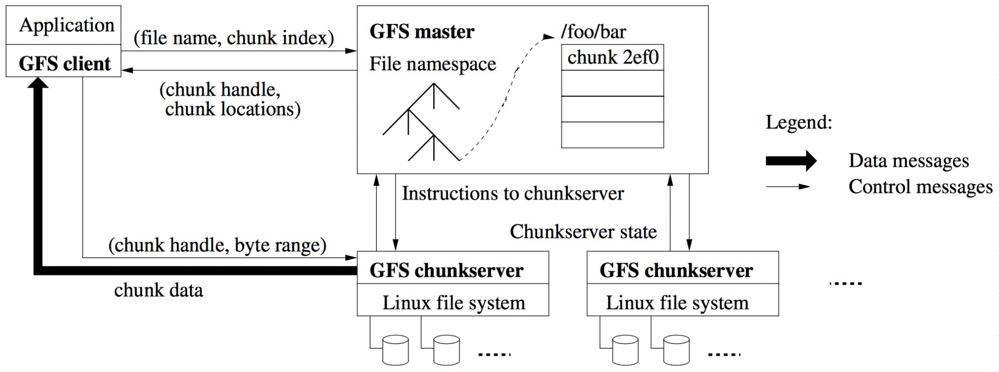
Distributed Filesystems

file system shared by being simultaneously mounted on multiple servers

data - partitioned and replicated across network

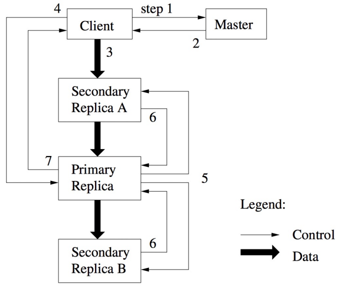
Google File System (GFS)



Storage

- single file = many objects
- files into 64MB chunks with unique identifier
- files are replicated (>=3)
- master maintains all file system metadata
- chunkservers store chunks on local disk as Linux files

Reads



- client sends read request to GFS master
- master replies with chunk handle and locations
- client caches metadata
- client sends data request to replicas
- chunk server responds with requested data

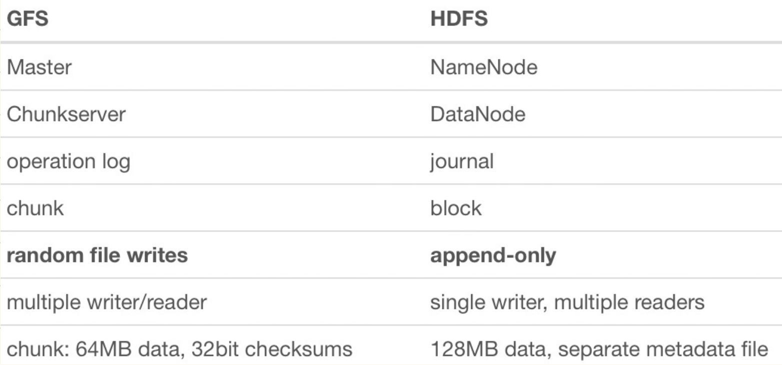
Writes

- client sends request to allocate primary replica chunkserver
- master responds with locations of chunkserver replicas and primary replica
- client sends write to all replicas chunk server’s buffer
- client tells primary replica to begin write function
- secondary replicas complete write function

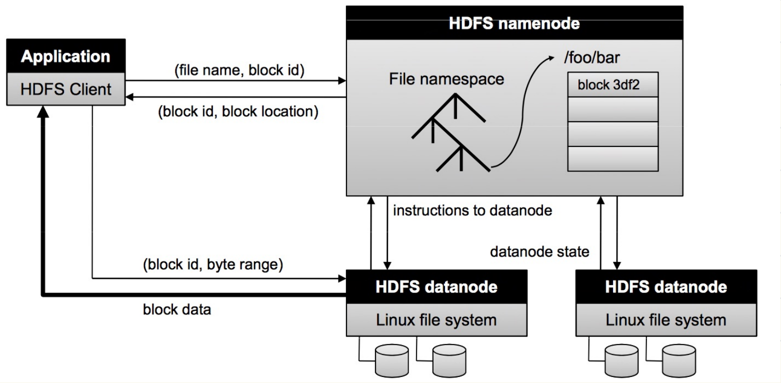
Operation

- master doesn’t keep persistent record of chunk locations
- queries chunk servers at startup, updated by periodic polling
- journaled - all operations added to log first, then applied
- node failure:
    - master - external instrumentation to start it elsewhere
    - chunkserver - restart
- chunkservers use checksum to detect data corruption
- master keeps chunk version number (up-to-date vs stale replica)

Hadoop File System (HDFS)



GFS	HDFS
Master	NameNode
Chunkserver	DataNode
operation log	journal
chunk	block
random file writes	**append-only**
multiple writer/reader	single writer, multiple readers
chunk: 64MB data, 32bit checksums	128MB data, separate metadata file


HDFS namenode
Application
HDFS Client
(file name, block id)
(block id, block location)
File namespace
/foo/bar
block 3df2
instructions to datanode
datanode state
(block id, byte range)
block data
HDFS datanode
Linux file system
HDFS datanode
Linux file system