Advanced Topics in Distributed Systems

Dr. Ayman Abdel-Hamid
Computer Science Department
Virginia Tech

Distributed File Systems

Based on Ch11, Distributed Systems: Principles and Paradigms 2/E

And Google File System paper in Reading List

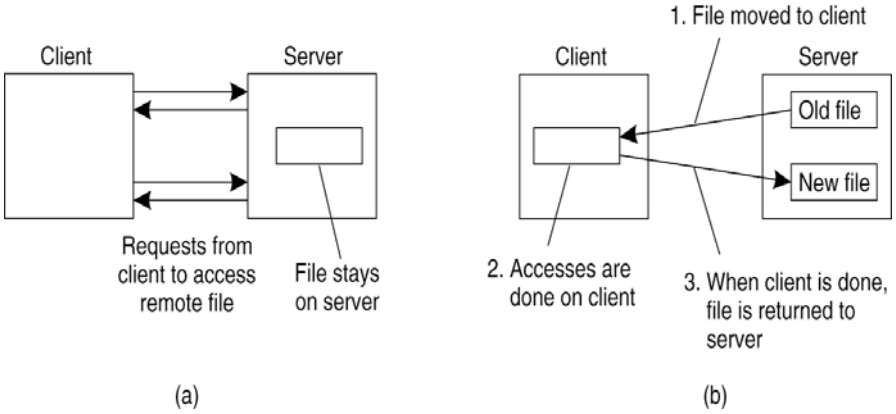
Outline

- Architectures of Distributed File Systems
  - Client-server architectures
    - NFS
  - Cluster-based DFS
    - Google File System
  - Symmetric Architectures

Client-Server Architectures 1/3

- Network File System
- Each file server provides a standardized view of its local file system
- Communication protocol to access files stored on a server
- Heterogeneous collection of processes on different OS and machines to share a common file system
- NFS independent of local file system

Client-Server Architectures 2/3



- Figure 11-1. (a) The remote access model. (b) The upload/download model.

Client-Server Architectures 3/3

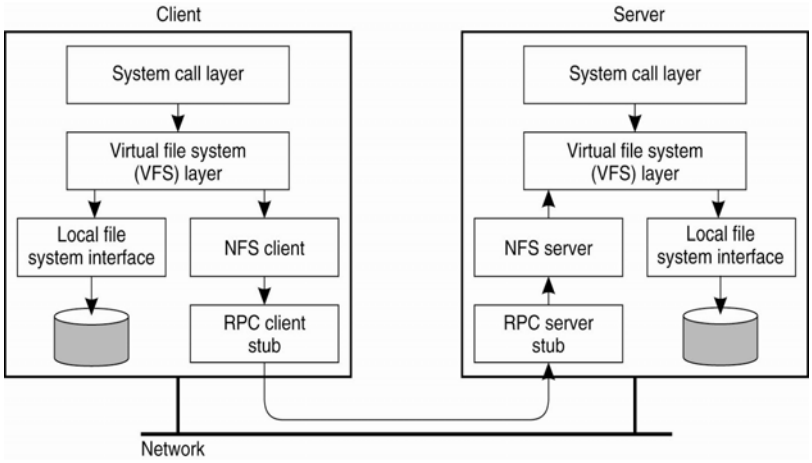


Figure 11-2. The basic NFS architecture for UNIX systems.

File System Model 1/2

Operation	v3	v4	Description
Create	Yes	No	Create a regular file
Create	No	Yes	Create a nonregular file
Link	Yes	Yes	Create a hard link to a file
Symlink	Yes	No	Create a symbolic link to a file
Mkdir	Yes	No	Create a subdirectory in a given directory
Mknod	Yes	No	Create a special file
Rename	Yes	Yes	Change the name of a file
Remove	Yes	Yes	Remove a file from a file system
Rmdir	Yes	No	Remove an empty subdirectory from a directory

File System Model 2/2

Operation	v3	v4	Description
Open	No	Yes	Open a file
Close	No	Yes	Close a file
Lookup	Yes	Yes	Look up a file by means of a file name
Readdir	Yes	Yes	Read the entries in a directory
Readlink	Yes	Yes	Read the path name stored in a symbolic link
Getattr	Yes	Yes	Get the attribute values for a file
Setattr	Yes	Yes	Set one or more attribute values for a file
Read	Yes	Yes	Read the data contained in a file
Write	Yes	Yes	Write data to a file

Cluster-Based Distributed File Systems 1/7

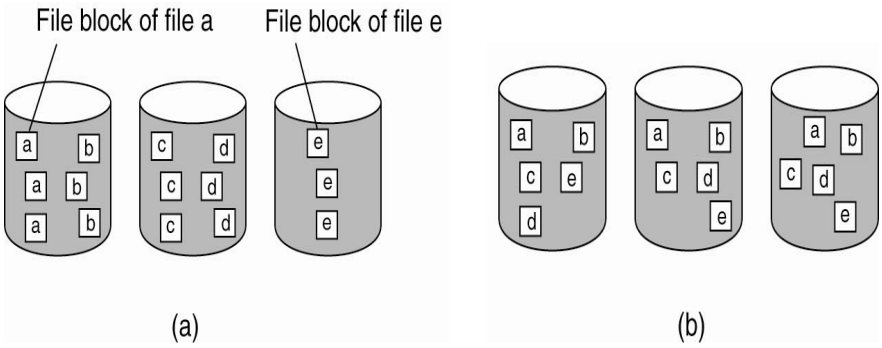


Figure 11-4. The difference between (a) distributing whole files across several servers and (b) striping files for parallel access.

Cluster-Based Distributed File Systems 2/7

- Organizing a DFS for very large data centers (Amazon, Google, …)
- GFS (Google File System)
  - Files very large (multi gigabytes)
  - Contains lots of smaller objects
  - I/O assumptions and block sizes?
  - Updates take place by appending data rather than overwriting parts of file
  - Server failures are the norm

Cluster-Based Distributed File Systems 3/7

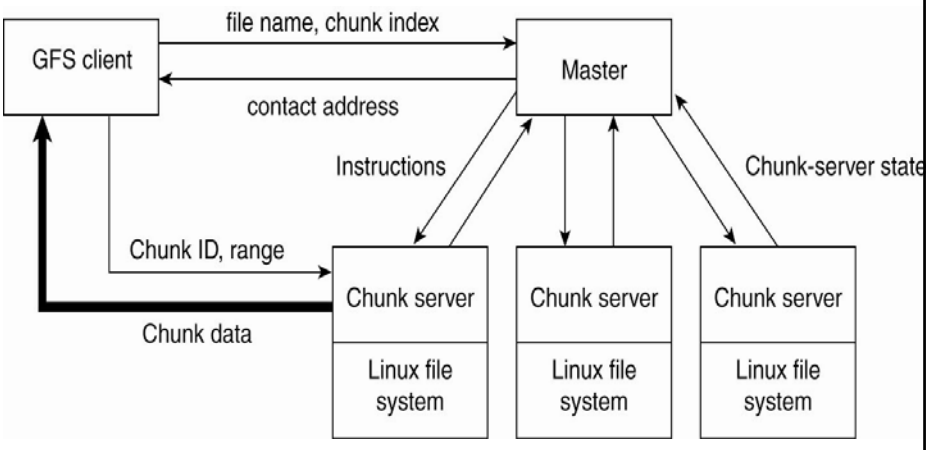


Figure 11-5. The organization of a Google cluster of servers.

Cluster-Based Distributed File Systems 4/7

- A GFS cluster consists of a single master along with multiple chunk servers
- GFS file divided into chunks of 64 MB each and chunks distributed across chunk servers (with replication, default 3 replicas)
- GFS master contacted for metadata (file name and chunk index), returning a contact address for the chunk
- Chunk index obtained by mapping file name and byte offset (fixed chunk size)

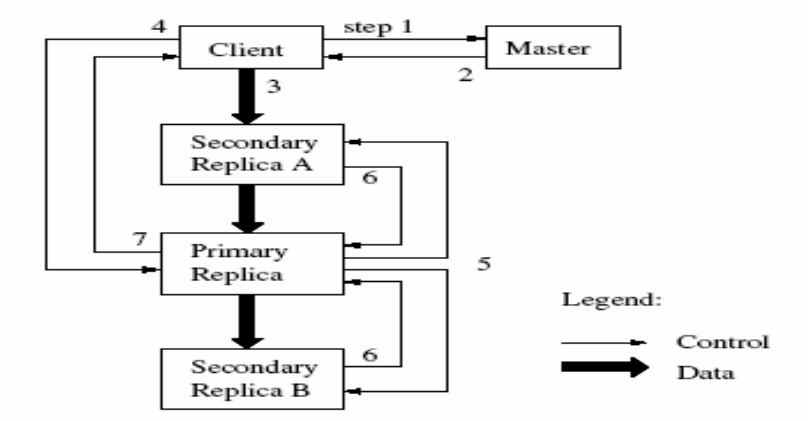
Cluster-Based Distributed File Systems 5/7

- GFS master occasionally contacts chunk servers to record stored chunks
- GFS master refreshes chunk allocation information by polling chunk servers (chunk server failure or crash effect?)
- Chunks replicated according to a primary-backup scheme (GFS master not involved)

Cluster-Based Distributed File Systems 6/7

- Hierarchical name space implemented using a single-level table
- Path names mapped to metadata
  - Less than 64 bytes of metadata for each 64 MB chunk
- Table kept in main memory, and
  - mapping of files to chunks
  - Locations of each chunk's replicas (not persistent)
- Updates logged to persistent storage
- Multiple clusters can work together

Cluster-Based Distributed File Systems 7/7



Write Control and Data Flow

Symmetric Architectures

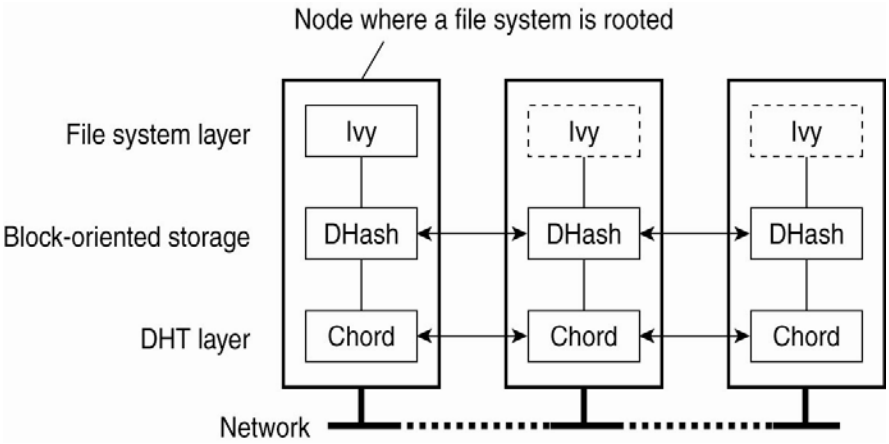


Figure 11-6. The organization of the Ivy distributed file system.