MODULE 6

DISTRIBUTED FILE SYSTEM & NAME SERVICE

File System

- describes how files are structured named, accessed, used, protected & implemented.
- used for permanent use of info on secondary storage
- also provide sharing of info

DFS

- supports
  1) Remote Information Sharing
  2) User mobility
  3) Availability
  4) Displaced Workstations
- provides 3 types of services:
  1) Storage Services
  2) True File Services
  3) Named Services

Features of good DFS

1) Transparency
   - → structure transparency
   - → access transparency
   - → naming transparency
   - → replication transparency
2) User Mobility
3) Performance
4) Simplicity & ease of use
5) Scalability
6) High availability
7) High Reliability
8) Data Integrity
9) Security
10) Heterogeneity

File models

1) Unstructured file
   - → file is unstructured sequence of data
2) Structured File
   - → file is presented to file server as ordered sequence of records (smallest unit of data that can be accessed).
3) Mutable File
   - → each update operation on file updates or writes the old content & new content is produced
4) Immutable file
   - → each update operation creates new version of file

File Accessing Model

1) Remote Service Model
2) Data Caching Model
3) Unit of Data Transfer model

Unit of Data Transfer model

- fraction of data that is transferred between client & server due to single read or write operation
- in data caching model of accessing remote file, 4 models are used to transferred the data:
  - → file level transfer model
  - → Block level transfer model
  - → Byte level transfer model
  - → Record level transfer model

Caching

- Remote Access can be served locally so that access can be faster
- Network traffic & server load is reduced which improves scalability
- Network overhead is less when transmitting big amount of data in comparison to remote service
- To maintain consistency,
  - → If less writes then better performance
  - → If more frequent writes then poor performance
- Caching is better for machine with disk or large main memory
- Lower level machine interface is ≠ from upper level UI.

File Replication

- In replication, replica is created at server

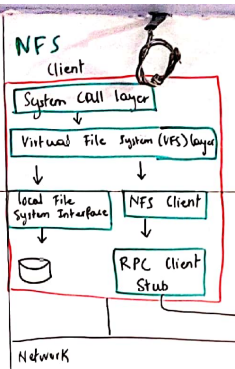
Whereas cached copy is associated with clients

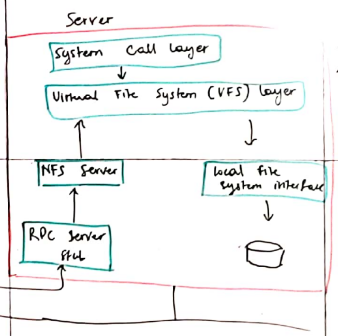
- Replica is more persistent as compared to cached copy
- Existence of replica depends on availability & performance
- Existence of cached copy depends on locality in access patterns
- Cached copy is dependent on replica

Advantages of Replication

- Increased availability
- Increased reliability
- Improved response time
- Improved system throughput
- Reduced Network traffic
- Better Scalability
- Autonomous Operation

NFS





Remote Service

- Remote Access is handled across the network so it is slower in comparison to caching
- Network traffic & server load is increased which degrades performance
- Transmitting series of response to specific request leads to higher network overhead in comparison to caching
- To maintain consistency,
  - → communication between client & server is there to have a master copy consistent with client's cached data
- Remote Service is better when main memory is small
- It is just an extension of local files system interface across network.

GFS

System Architecture of google

- Google Application & Services
- Google Infrastructure (Middleware)
- Google Platforms

Google Infrastructure

Distributed computation	Map Reduce	Saw Zall	
Data & coordination	GFS	Chubby	BigTable
Communication Paradigm	Protocol Buffer	Publish Subscribe	

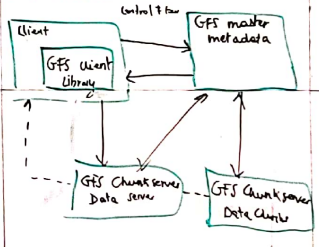
Requirements

1) GFS must run reliably on physical architecture
2) It is optimised for patterns of usage within Google
3) GFS must fulfill all the requirement

GFS operations

1) Create
2) Delete
3) Open
4) Close
5) Read
6) Write

GFS Architecture



Resolution Protocol

- is the retrieval of file underlying numeric values corresponding to computer host names, etc. user names, group names & other named entities