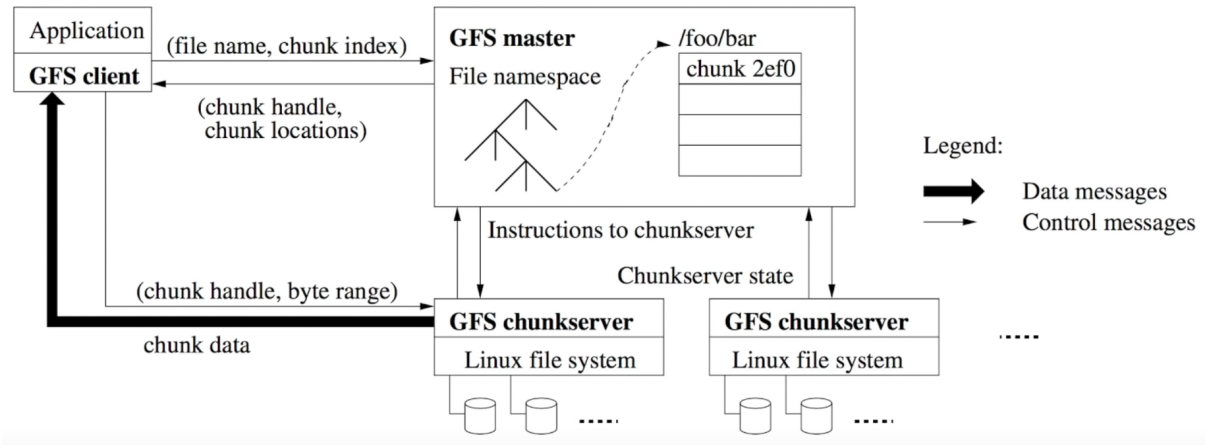
Vivek Puar-2019201044
Sanjay Papaiahgari-2019201077
Swapnil Satpathy-2019201078
Ashish Kempwad-2019201091

Google File System from its' Research Paper

INTRODUCTION:

Google File System is a scalable distributed file system for large distributed data-intensive applications. It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients.
Google has designed and implemented a scalable distributed file system for their large distributed data intensive applications. They named it Google File System, GFS. GoogleFile System is designed by Sanjay Ghemawat, Howard Gobioff and Shun-Tak Leung of Google in 2002-03.GFS provides fault tolerance, while running on inexpensive commodity hardware and also serving large number of clients with high aggregate performance. Even though the GFS shares many similar goals with previous distributed file systems, the design has been driven by Google's unique workload and environment.Google had to rethink the file system to serve their "very large scale" applications, using in expensive commodity hardware.

OUR IMPLEMENTATION:
Having been inspired by the paper we have tried implementing the google file system.
The figure below depicts the overall layout of GFS.



Key Components:

1)Master Server:
It stores the metadata of entire file chunk wise. It is the backbone behind the entire system. Client contacts Master Server while uploading and downloading any file.
It redirects the client to chunk servers appropriately.

HeartBeat:
Running on a different thread at a time interval of some seconds(6 in our case) the master server will check whether all the chunk-servers are active or not. If a chunk server fails, then when the master server catches it, it will trigger a functionality which will send all the file chunks that the failed chunk server had from their replicas stored in other chunk servers to other appropriate chunk servers which will maintain the condition that there will always be two replicas in every case of every chunk of a file.

Command to run the Master_Server:
Python3 Master_Server.py

***2)*Backup Master Server:**
It comes into play whenever the Master server goes down.
Its basic working is similar to Master server.

Command to run Backup Master_Server:
Python3 Backup_Master_Server.py

3)Client:
It is basically the end user trying to upload and download the document without being worried about the complex operations. A particular client can put a lease on certain file so as to restrict the file from being accessed by other clients.No other client can put a lease on that file for that particular time. Client can later un-lease the file as well. If a client does not un-lease the file then it is un-leased by default after 60 seconds.

Command to run client:
Python3 client.py
Command to Upload a file:
upload filename
Command to Download a file:
download filename
Command to put a lease:
lease filename
Command to remove a lease:
unlease filename

4)Chunkserver:

Default Implementation(In code)
Chunk-size : 2KB
No of chunk servers: 4 [Port numbers: 6467,6468,6469,6470]
No of replica : 2(1 primary+ 1 secondary)

Command to run chunkservers:
Python3 chunkserver.py 6467
Python3 chunkserver.py 6468
Python3 chunkserver.py 6469
Python3 chunkserver.py 6470

Four chunk servers would be running on four different terminals.
The file is stored in format is chunks in round robin manner in chunk server whenever it is uploaded for first time. Then every time a client uploads a file, it is stored in forms of chunks based on load balancing. The replicas of the chunks is maintained in load balanced manner.
A list is maintained for every chunk servers whenever any chunk server goes down master tells the appropriate chunk servers to redistribute the chunks from that particular chunk server to the another chunk server to maintain the constraint that there will be every time one replica of a particular chunk of the file. It means every chunk of a file will be stored on a primary chunk-server and a secondary chunk server at all circumstances.