Google File System

Krupa Mavani (351960088) and Neha Keshan (876119212)

Features implemented:

- Implemented Distributed File System Network and Setup communication between client and master, client and chunk servers, master and chunk servers.
- Implemented Read Request from a Client Server.

Architecture:

GFS Master – 1

GFS Client – 1

GFS Chunk Servers – 5

Chunk Size- 11KB

Minimum replication factor- 3

Communication Protocol:

TCP Socket Connection

Metadata prepared:

Master-

Static- Initially it stores the information regarding all files in the system, the number of chunks each file has and their chunk_handle. (Data.txt)

- Data.txt: It contains list of filenames, its chunks, each chunk's unique chunk_handle, each chunk's chunk server locations.

Dynamic- Once the chunk servers are up and working, they connect to master and send information (the chunk servers IP address and Port number along with the chunks it has) to the master to store as the metadata. (Data2.txt)

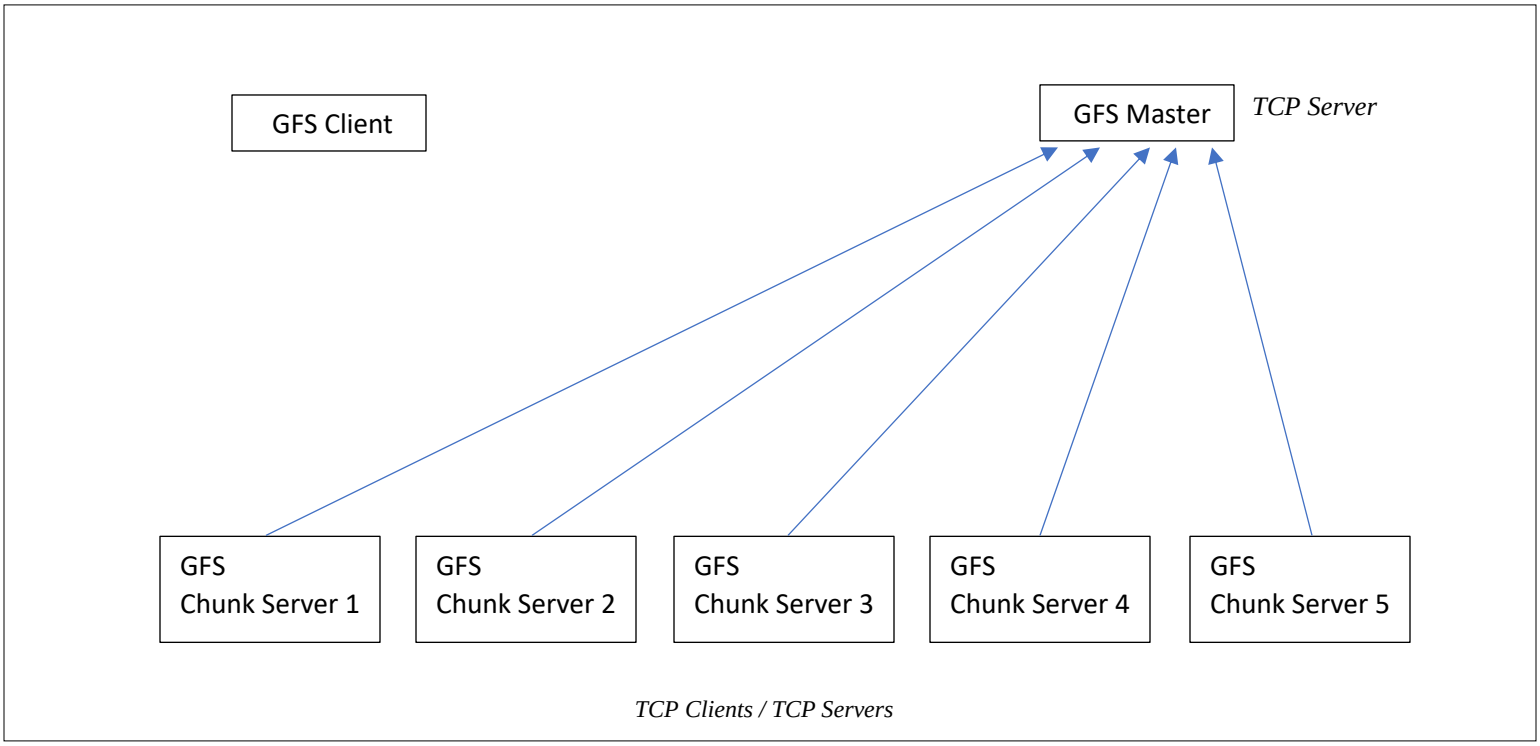
- Data2.txt: It contains chunk server locations and the chunk_handles of the chunks that each chunk server has.

Chunk Servers- Stores chunks of files.

Flow:

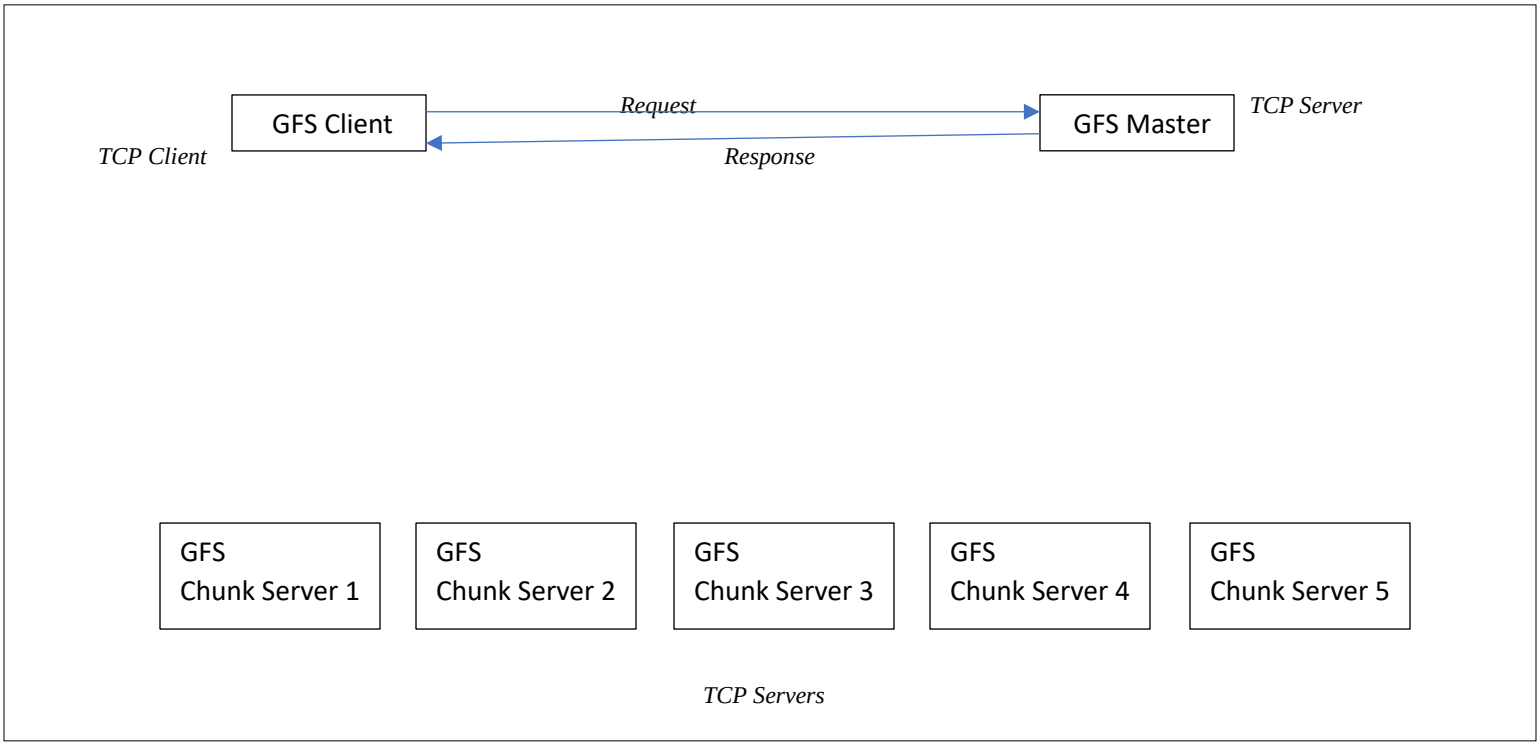
Phase 1-

1. GFS Master acts as TCP Server.
2. GFS Chunk Servers comes up, connects to GFS master as TCP Clients and send its meta data (the chunk servers IP address and Port number along with the chunks stored in this chunk server) to GFS Master.
3. GFS Chunk Servers also acts as TCP Servers for accepting GFS Client Server's read request.



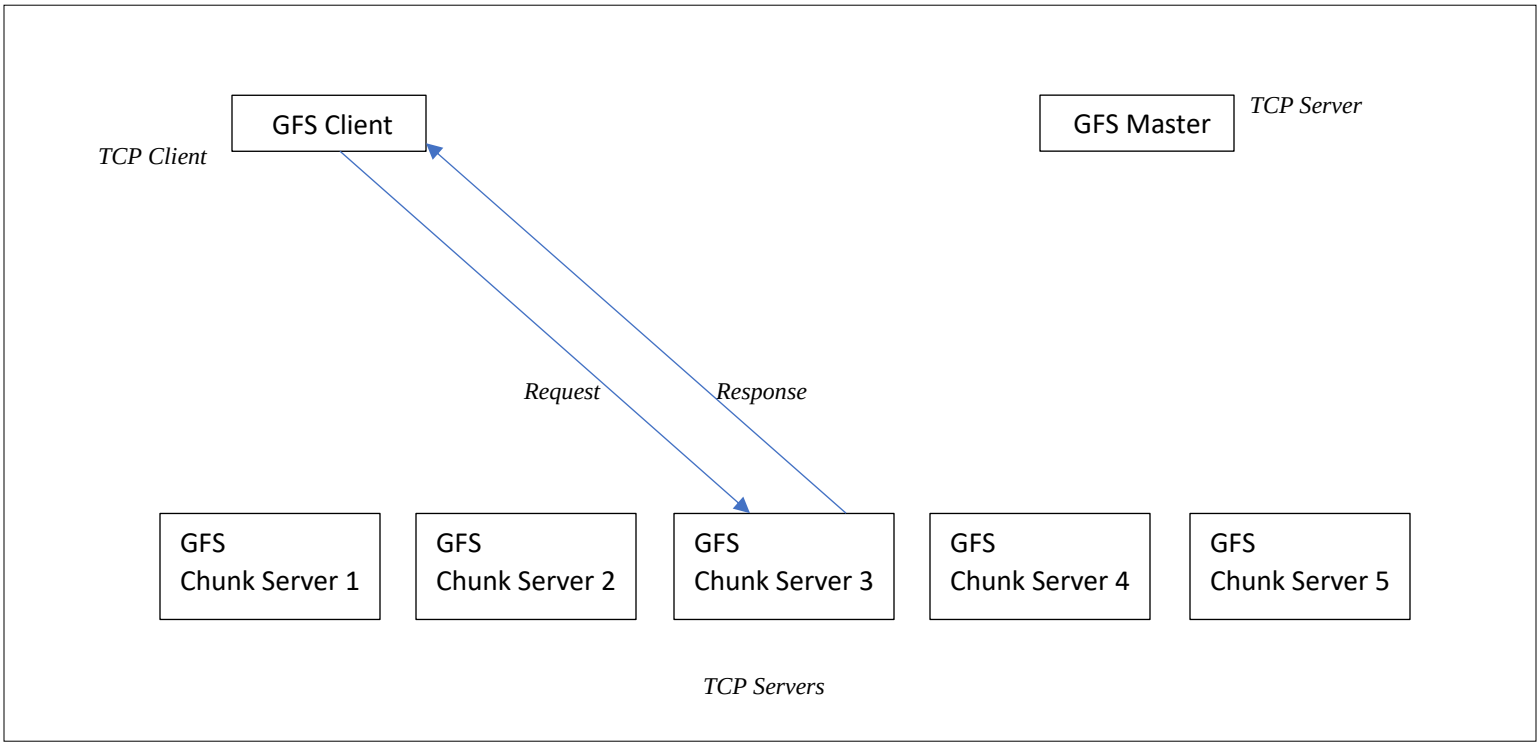
Phase 2-

1. GFS Client Server connects to GFS Master as a TCP Client.
2. GFS Client Server sends a read request (file name, chunk number) to GFS Master.
3. GFS Master returns results (chunk id of requested chunk, chunk server locations; IP and port; having that chunk) to GFS Client Server.



Phase 3-

1. GFS Client Server connects as a TCP Client to respective GFS Chunk Server.
2. GFS Client sends a request (chunk id) to the first GFS Chunk Server from the list received from GFS Master.
3. GFS Chunk Server returns relevant data (requested chunk as a file).



How to deploy?

-Either deploy all projects on same system or on multiple systems to simulate the distributed file system feature of GFS.

-Currently we have mentioned steps with 2 systems.

-There are in total 7 Projects to be deployed. (Names indicates its function)

-Each Project refers a separate JNachos instance.

1. GFS_Master
    a. Create a Project GFS_Master on System1.
    b. Open file jnachos/kern/Master1.java.
    c. The port for the master is set as 1000 within the try catch of call function. Change if needed.

2. GFS_ChunkServer1
    a. Create a Project GFS_ChunkServer1 on System1.
    b. Open file jnachos/kern/CS1Client.java.
    c. Within the try catch of the call function, do the following:
        i. Set the IP of respective system in:
        sock=new Socket(<IP of GFS_Master>,<Port of GFS_Master>);
        (to be used to connect to master as a TCP client)
        ii. String ip=<IP of this system>;
        (to be used when Chunk Server acts as TCP server)
        iii. String port=<Some new port to be used for this ChunkServer>;
        (to be used when Chunk Server acts as TCP server)
        iv. Change the **dirPath** to the path of the project (upto GFS_ChunkServer1) where the chunks are stored.

    d. Open file jnachos/kern/ChunkServer1.java
    e. Within the try catch of the call function, do the following:
        i. The port for the this chunkserver is to be set as port same as '**port'** variable in '**2.c.iii**' within the try catch of call function.

3. GFS_ChunkServer2
    a. Create a Project GFS_ChunkServer2 on System1.
    b. Open file jnachos/kern/CS2Client.java.
    c. Within the try catch of the call function, do the following:
        i. Set the IP of respective system in:
        sock=new Socket(<IP of GFS_Master>,<Port of GFS_Master>);
        (to be used to connect to master as a TCP client)
        ii. String ip=<IP of this system>;
        (to be used when Chunk Server acts as TCP server)
        iii. String port=<Some new port to be used for this ChunkServer>;
        (to be used when Chunk Server acts as TCP server)

iv. Change the **dirPath** to the path of the project (upto GFS_ChunkServer2) where the chunks are stored.

d. Open file jnachos/kern/ChunkServer2.java
e. Within the try catch of the call function, do the following:
    i. The port for the this chunkserver is to be set as port same as '**port'** variable in '**2.c.iii**' within the try catch of call function.

4. GFS_ChunkServer3
    a. Create a Project GFS_ChunkServer3 on System1.
    b. Open file jnachos/kern/CS3Client.java.
    c. Within the try catch of the call function, do the following:
        i. Set the IP of respective system in:
        sock=new Socket(<IP of GFS_Master>,<Port of GFS_Master>);
        (to be used to connect to master as a TCP client)
        ii. String ip=<IP of this system>;
        (to be used when Chunk Server acts as TCP server)
        iii. String port=<Some new port to be used for this ChunkServer>;
        (to be used when Chunk Server acts as TCP server)
        iv. Change the **dirPath** to the path of the project (upto GFS_ChunkServer3) where the chunks are stored.

    d. Open file jnachos/kern/ChunkServer3.java
    e. Within the try catch of the call function, do the following:
        i. The port for the this chunkserver is to be set as port same as '**port'** variable in '**2.c.iii**' within the try catch of call function.

5. GFS_ChunkServer4
    a. Create a Project GFS_ChunkServer4 on System2.
    b. Open file jnachos/kern/CS4Client.java.
    c. Within the try catch of the call function, do the following:
        i. Set the IP of respective system in:
        sock=new Socket(<IP of GFS_Master>,<Port of GFS_Master>);
        (to be used to connect to master as a TCP client)
        ii. String ip=<IP of this system>;
        (to be used when Chunk Server acts as TCP server)
        iii. String port=<Some new port to be used for this ChunkServer>;
        (to be used when Chunk Server acts as TCP server)
        iv. Change the **dirPath** to the path of the project (upto GFS_ChunkServer4) where the chunks are stored.

    d. Open file jnachos/kern/ChunkServer4.java
    e. Within the try catch of the call function, do the following:

i. The port for the this chunkserver is to be set as port same as '**port'** variable in '**2.c.iii**' within the try catch of call function.

6. GFS_ChunkServer5
    a. Create a Project GFS_ChunkServer5 on System2.
    b. Open file jnachos/kern/CS5Client.java.
    c. Within the try catch of the call function, do the following:
        i. Set the IP of respective system in:
        sock=new Socket(<IP of GFS_Master>,<Port of GFS_Master>);
        (to be used to connect to master as a TCP client)
        ii. String ip=<IP of this system>;
        (to be used when Chunk Server acts as TCP server)
        iii. String port=<Some new port to be used for this ChunkServer>;
        (to be used when Chunk Server acts as TCP server)
        iv. Change the **dirPath** to the path of the project (upto GFS_ChunkServer5) where the chunks are stored.

    d. Open file jnachos/kern/ChunkServer5.java
    e. Within the try catch of the call function, do the following:
        i. The port for the this chunkserver is to be set as port same as '**port'** variable in '**2.c.iii**' within the try catch of call function.

7. GFS_ClientServer
    a. Create a Project GFS_ClientServer on System2.
    b. Open file jnachos/kern/GFSClient.java.
    c. Within the try catch of the call function, do the following:
        i. Set the IP of respective system in:
        sock=new Socket(<IP of GFS_Master>,<Port of GFS_Master>);
        (to be used to connect to master as a TCP client)

How to run all projects?

1. Run GFS_Master first.
2. Run all GFS_ChunkServer projects next.
3. Run GFS_ClientServer last. It will ask for input.

Sample Input to be typed in console of GFS_ClientServer:

gfs 3

-Here ‘gfs’ is filename and ‘3’ is chunk number.

Output:

-Output is the chunk that was requested in the form of a file.

-It is stored under the location of the GFS_ClientServer project.

- Clean GFS_Master/Data2.txt file before running the program second time or else there will be multiple copies of same metadata.
- After a request is fulfilled by ChunkServer close all connections manually from project console.

Note: If TCP Client is not getting connected to TCP Server, please disable firewall on server machine.