# COMPUTER PROGRAMMING PROJECTS

**Monday, April 14, 2014**

## General architecture of Google File System

GFS is clusters of computers. A cluster is simply a network of computers. Each cluster might contain hundreds or even thousands of machines. In each GFS clusters there are three main entities:

1. Clients
2. Master servers
3. Chunk servers.

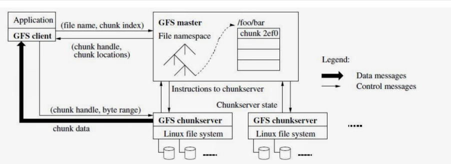



Figure 1 GFS Architecture

**Client** can be other computers or computer applications and make a file request. Requests can range from retrieving and manipulating existing files to creating new files on the system. Clients can be thought as customers of the GFS.

**Master Server** is the coordinator for the cluster. Its task include:-

1. Maintaining an operation log, that keeps track of the activities of the cluster. The operation log helps keep service interruptions to a minimum if the master server crashes, a replacement server that has monitored the operation log can take its place.

2. The master server also keeps track of **metadata**, which is the information that describes chunks. The metadata tells the master server to which files the chunks belong and where they fit within the overall file.

**Chunk Servers** are the workhorses of the GFS. They store 64-MB file chunks. The chunk servers don't send chunks to the master server. Instead, they send requested chunks directly to the client. The GFS copies every chunk multiple times and stores it on different chunk servers. Each copy is called a **replica**. By default, the GFS makes three replicas per chunk, but users can change the setting and make more or fewer replicas if desired.

**Management done to overloading single master in Google File System**

Having a single master enables the master to make sophisticated chunk placement and replication decisions using global knowledge. However, the involvement of master in reads and writes must be minimized so that it does not become a bottleneck. Clients never read and write file data through the master. Instead, a client asks the master which chunk servers it should contact. It caches this

### About Me

B. Roshan

I am a tech enthusiast,computer engineering student at Institute of Engineering, Pulchowk campus.I love to read novels,write blogs and develop software applications.Traveling from one place to another exploring and learning different culture is one of my hobbies.

View my complete profile

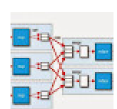
### Popular Posts

General architecture of Google File System

GFS is clusters of computers. A cluster is simply a network of computers. Each cluster might contain hundreds or even thousands of machin...

Timer Programming in 8051 Micro Controller with Keil

Learn More about Interrupt, timer and Serial Programming in 8051 Micro Controller : Click here • The 8051 has two timers/counters,...

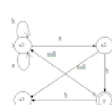


Map Reduce Framework and its components

Map Reduce is a programming model and an associated implementation for processing and generating large data sets. Programs written in this...

Interrupt Programming in 8051 MicroController with Keil

• Concept behind Interrupt – Interrupt vs Polling – What is the advantage of having interrupt based system over po...



Backtracking , Structures,Non Deterministic Finite State Automata [Artificial Intelligence Programming in Visual Prologue] --Part II

Backtracking As has already been seen prolog has built in backtracking mechanism. It tries to prove a goal with all possible instan...

### Sample Text

### Followers

Followers (1)

Follow

information for a limited time and interacts with the chunk servers directly for many subsequent operations.

**General scenario of client request handling by GFS**

File requests follow a standard work flow. A read request is simple; the client sends a request to the master server to find out where the client can find a particular file on the system. The server responds with the location for the primary replica of the respective chunk. The primary replica holds a lease from the master server for the chunk in question.

If no replica currently holds a lease, the master server designates a chunk as the primary. It does this by comparing the IP address of the client to the addresses of the chunk servers containing the replicas. The master server chooses the chunk server closest to the client. That chunk server's chunk becomes the primary. The client then contacts the appropriate chunk server directly, which sends the replica to the client.

Write requests are a little more complicated. The client still sends a request to the master server, which replies with the location of the primary and secondary replicas. The client stores this information in a memory cache. That way, if the client needs to refer to the same replica later on, it can bypass the master server. If the primary replica becomes unavailable or the replica changes then the client will have to consult the master server again before contacting a chunk server.

The client then sends the write data to all the replicas, starting with the closest replica and ending with the furthest one. It doesn't matter if the closest replica is a primary or secondary. Google compares this data delivery method to a pipeline.

Once the replicas receive the data, the primary replica begins to assign consecutive serial numbers to each change to the file. Changes are called mutations. The serial numbers instruct the replicas on how to order each mutation. The primary then applies the mutations in sequential order to its own data. Then it sends a write request to the secondary replicas, which follow the same application process. If everything works as it should, all the replicas across the cluster incorporate the new data. The secondary replicas report back to the primary once the application process is over.

At that time, the primary replica reports back to the client. If the process was successful, it ends here. If not, the primary replica tells the client what happened. For example, if one secondary replica failed to update with a particular mutation, the primary replica notifies the client and retries the mutation application several more times. If the secondary replica doesn't update correctly, the primary replica tells the secondary replica to start over from the beginning of the write process. If that doesn't work, the master server will identify the affected replica as garbage.

**Advantages and disadvantages of large sized chunks in Google File System**

Chunks size is one of the key design parameters. In GFS it is 64 MB, which is much larger than typical file system blocks sizes. Each chunk replica is stored as a plain Linux file on a chunk server and is extended only as needed.

**Advantages**

1. It reduces clients’ need to interact with the master because reads and writes on the same chunk require only one initial request to the master for chunk location information.

2. Since on a large chunk, a client is more likely to perform many operations on a given chunk, it can reduce network overhead by keeping a persistent TCP connection to the chunk server over an extended period of time.

3. It reduces the size of the metadata stored on the master. This allows us to keep the metadata in memory, which in turn brings other advantages.

**Disadvantages**

1. Lazy space allocation avoids wasting space due to internal fragmentation.

2. Even with lazy space allocation, a small file consists of a small number of chunks, perhaps just one. The chunk servers storing those chunks may become hot spots if many clients are accessing the same file. In practice, hot spots have not been a major issue because the applications mostly read large multi-chunk files sequentially. To mitigate it, replication and allowance to read from other clients can be done.

at 11:46 PM

Labels: Big Data Technology, Google File System

## 6 comments:

**srinivas3224** July 14, 2016 at 9:30 PM

good

Reply

**Unknown** August 6, 2016 at 12:57 AM

very good artical

Reply

**unknown** May 17, 2017 at 11:37 PM

thanks for such clear and systematic explanation..

Reply

**Unknown** November 1, 2018 at 7:48 AM

Thanks for the detailed explanation.

Reply

**Moin khan chouhan** January 11, 2020 at 9:33 PM

computer shop near me

We provide the best computer repair service in Albany. Our services are available to residential, business, educational, and governmental clients. Contact us TODAY for a FREE consultation at (518)892-4419.

to get more - https://willpowerpcs.com/computer-repair/

Reply

**Yoga Classes in Enoggera** January 18, 2020 at 3:49 AM

computer store near me

We provide the best computer repair service in Albany. Our services are available to residential, business, educational, and governmental clients. Contact us TODAY for a FREE consultation at (518)892-4419.

to get more - https://willpowerpcs.com/computer-repair/

Reply

To leave a comment, click the button below to sign in with Google.

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.

SIGN IN WITH GOOGLE

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)