

# Cloud Computing & Big Data: Paper GFS

## 1. What are the key assumptions that GFS makes regarding the workload and usage nature?

The key assumptions include:

- The file system is composed of inexpensive commodity components. These components may fail frequently. Therefore, constant monitoring and prompt recovery are required.
- The system stores modest number of large files. The file system is designed to optimize the storage performance for large files such as multi-GB files. Small files are supported but aren't the major concern.
- The workloads are primarily made up of two types of reads. The first is large streaming reads within which reads are commonly for 1MB or more data and falls into a contiguous region of file. The other is random reads which typically asks for several KB at arbitrary offset.
- The workloads often contain large, sequential writes that append data to files. Small writes are supported but not optimized.
- The system efficiently implements well-defined semantics for concurrent writes from multiple users. Buying atomicity with minimal overhead is necessary.
- High sustainable bandwidth is more important than low latency.

## 2. Why a new file system was designed instead of fixing an existing one?

Because existing file systems hold different system assumptions which leads the researchers to reexamine the traditional technical choices and explore radically different design points. The new system is elaborately designed to meet the rapidly growing demands of Google's data processing needs.

## 3. What is a GFS master and slave architecture?

A GFS cluster contains a single master and multiple Chunkservers and allows multiple clients to access. Each of the components in the architecture is typically a commodity Linux machine. A diagrammatic sketch of the GFS architecture is:

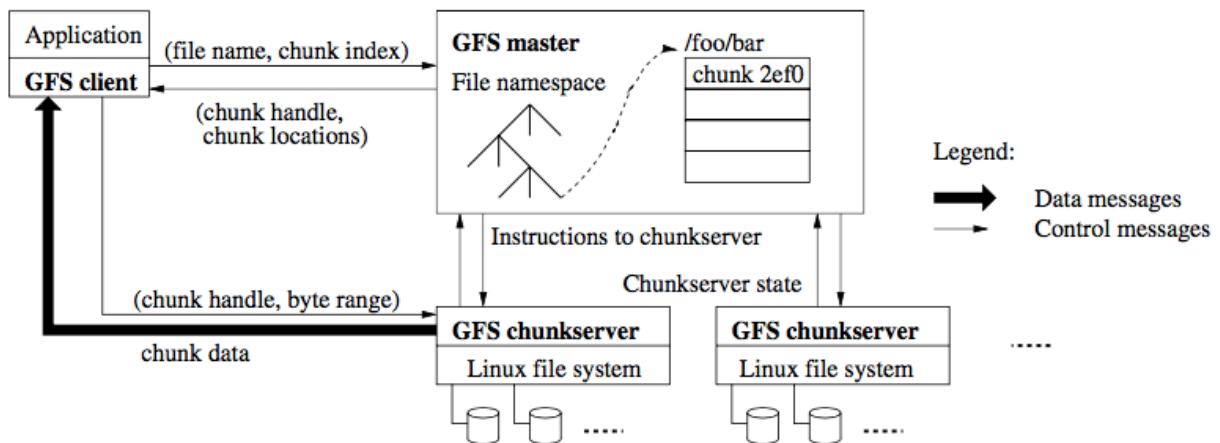


Figure 1: GFS Architecture

#### **4. How is file stored in GFS using the chunk server?**

Firstly, the files are divided into fixed-size chunks. Then GFS master will assign an immutable and globally unique 64bit chunk handle to that file. After the chunk handle is created, the Chunkservers will store the chunk in their local disks as Linux files. At last, for reliability, each chunk is replicated among multiple Chunkservers. The number of Chunkservers that hold the replicated data is typically three. However, this number can be modified corresponding to user's reliability level. Subsequent reads and writes to the file can be completed by specifying chunk handle and byte range.

#### **5. What are key responsibilities of GFS master?**

From my perspective, GFS master functions like a DNS server. It receives request from the client and directs the user to the Chunkserver for actual data transmission. More specifically, all system metadata and system-wide activities are controlled by the GFS master which include: namespace, access control information, mapping from files to chunks, current locations of chunks, chunk lease management, garbage collection of orphaned chunks and chunk migration among Chunkservers. Additionally, the GFS master keeps track of the status of the entire system by sending HearBeat messages to every Chunkserver. From this periodical communication, the master can send instructions to Chunkservers as well as collect data from them.