

The Google File System (GFS) is a highly fault tolerant, running on cheap hardware, and delivers high performance to a large number of clients system. The system self-monitors for issues and maintenance, handles a modest number of large files, and also mostly deals with actions like appending data to existing data. Atomicity is also a very important aspect of GFS due to how we have a large # of clients, concurrently doing tasks on a file. GFS is divided into 3 parts: Master, which handles metadata of all chunks in the system, Chunkservers, which store chunks on local disks and operate on chunks as needed, and Chunks, which are our 'Data blocks' of fixed size, usually replicated for reliability. A Client, a requested task by someone, interacts with the Master for metadata operations, but any data-bearing operations are done via chunkservers, it's also important to note Clients first interact with ^{the} Master, then using the metadata obtained, connect with addressed chunkservers, usually this interaction is done where the Client has lots of tasks in one request, hence reducing latency and improves efficiency.

The Master holds 3 key properties, the file/chunk namespaces, mapping from files to chunks, and location of each chunk's replica. It's important to note that replicas of chunks are not tracked by the Master, instead, on startup, we pull information from chunkservers for that information, this is done when the 'heartbeat' signal is sent from chunkserver to Master. It's also important to note that there is an operation log that contains past 'Critical' metadata changes. This log is replicated on multiple machines and respond to a client operation only after flushing the corresponding log record to disk both locally/remotely.