Hadoop and Google File Systems

Reading Assignment 1

(COL733: Cloud Computing Technology Fundamentals)

- *Aditi Singla (2014CS50277)*

Hadoop Distributed File System (HDFS):

HDFS is a distributed file system designed to run on a commodity hardware, which has no reliability. It assumes write-once-read-many access for files. The main features of HDFS are that it is completely fault-tolerant with quick detection of faults and automatic recovery, supports large files and it has been optimised for batch processing, with high throughput (with some latency), since generally, applications using HDFS, need streaming access to data. Due to this, a few of POSIX requirements have been relaxed, to increase the throughput and hence, the performance.

HDFS Architecture: It follows master/slave architecture. An HDFS cluster consists a NameNode (master server) and a lot of DataNodes (slaves), one for each node in the cluster.

- NameNode stores the metadata, manages the namespace and regulates client access to the files.
- DataNode manages storage attached to the nodes that they run on. A file is actually split into blocks and these blocks are stored in a set of DataNodes.
- While NameNode executes file system namespace and determines the mapping of blocks to DataNodes, DataNodes are responsible for serving the read/write requests, along with performing block creation, deletion and replication.

Fault Tolerance and Data Replication: HDFS is designed to reliably store very large files across the machine. The files are stored in blocks, which are replicated for fault tolerance, where replication factor can be specified by the application. The NameNode, which maintains the file system namespace, receives a HeartBeat and a BlockReport from each of the DataNode in the cluster. This HeartBeat actually ensures that the DataNode is functioning properly.

Replicas are placed across the hardware such that data reliability and high performance are ensured. Large instances of HDFS run on a cluster of computers, spread across many racks. Generally, when the replication factor is three, one replica is placed on one node in the local rack, and the rest two replicas on two different nodes of a remote rack. This allows data reliability and an improved write performance. While reading a file, the nearest replica of the corresponding block is read. NameNode enters SafeMode, until each block has the minimum number of copies, after which restores the NormalMode.

FileSystem MetaData: HDFS uses a transaction log called EditLog, which stores every change in the metadata. The complete file system namespace, mappings from file to block and its properties are stored in a file called FsImage. FsImage plays a crucial role in the process of Checkpoints, where we take the EditLog, apply all the transactions to the in-memory FsImage, then save the new FsImage on the disk & truncate EditLog. This regulates the size of log files.

Communication Protocol: HDFS uses RPC based communication protocols on top of TCP/IP. Remote Procedure Calls are made by Client/DataNode, while NameNode can never initiate a RPC and just responds to the ones made by Client/DataNode.

Google File System (GFS):

GFS is a scalable distributed file system for large distributed data-intensive applications. It assumes multiple-writes-multiple-read access for files. Like HDFS, the main features of GFS are that it ensures fault tolerance, while running on inexpensive commodity hardware, and delivers high aggregate performance to a large number of clients. GFS doesn't implement a standard API like POSIX and also, follows a tradition hierarchical file organisation.

GFS has snapshot and record append operations. Snapshot basically a file/directory copy at low cost. Record append allows multiple clients to append data to the same file system concurrently. This allows implementation of multi-way merge results and producer-consumer queues to which many clients can append simultaneously, without locking.

GFS Architecture: A GFS cluster consists of a single master and multiple chunkservers and can be accessed by multiple clients. Files are divided into fixed-size chunks. Each chunk is identified by a unique 64 bit chunk handle assigned by the master at the time of chunk creation.

- Chunkservers store chunks on local disks as Linux files and read or write chunk data specified by a chunk handle and byte range. For reliability, each chunk is replicated on multiple chunkservers. By default, we store three replicas.
- Master maintains all file system metadata. This includes the namespace, access control information, the mapping from files to chunks, and the current locations of chunks. It also controls system-wide activities such as chunk lease management, garbage collection of orphaned chunks, and chunk migration between chunkservers. The master periodically communicates with each chunkserver in HeartBeat messages to give it instructions and collect its state.

Like above, most of the features are common with HDFS. The major differences between HDFS and GFS are:

- **File Structure:** GFS is divided into 64 MB chunks, which are further divided into 64 KB blocks and each block has a checksum. While in HDFS, there are 128 MB blocks, and NameNode holds the replica as two files, one for data and one for checksum.
- **I/O:** HDFS assumes write-once-read-many access for files, while GFS assumes multiple-writes-multiple-read access for files. Also, HDFS allows only appending data, while random writes are possible in case of GFS.
- **Processes:** NameNode and DataNode in HDFS are similar to Master and ChunkServer in GFS.
- **Snapshots:** Snapshot feature allows us to roll back a corrupted file system instance to a previously known good point. GFS do have the feature of Snapshots, while HDFS doesn't.
- **Platform Dependence:** HDFS is written in Java and is platform independent, while GFS is written in C,C++ and is Linux based. Also, HDFS uses Standard API, POSIX, to maintain compatibility between operating systems, while GFS doesn't.
- **Communication Protocol:** HDFS uses RPC based communication protocol on top of TCP/IP, while GFS uses TCP/IP based protocol.

References:

1. *Reading Material :R01_HDFS*
2. *Reading Material :R02_Google_File_System*