GFS

Evan McElheny

Google File System

Distributed file system.

Master server organizes data into chunks for a chunk server to handle.

Many chunk servers store data so there is minimal bottleneck at the master.

Optimizes around append and read functions.

These were seen as the most necessary functions

Other functions are available but not optimized for.

Fault tolerant.

As the data is stored on commercial hardware failures are expected.

Google File System

Master server controls all access to data.

- Keeps a heartbeat connection with chunk servers to ensure that they are still online
- When data is requested the master server sends the chunk id's to the client who then queries the chunk server.

Optimizations for Append

- When data is to be appended it is sent to a chunk server that master gives a temporary lease.
- The data is then appended at least once to all other copies of the chunk.

Google FIle System-Analysis

The GFS is very scaleable.

- The master server only directs clients to which chunks they want access to.
- The only bottleneck for scale is the master running out of memory. “The master maintains less than 64 bytes of metadata for each 64MB chunk.”
- More data in the GFS does not make operations take significantly longer as more chunk servers can distribute the data better.

Ensures redundancy without slowing down the system.

- Copy operations are capped to prevent them from delaying other requests.

Comparison Paper

They wanted to compare three different different distributed database systems, and tested a wide range of different operations on them.

Hadoop:

A open source file system of the MapReduce framework.

DBMS-X:

A parallel SQL DBMS. Compression was enabled as it sped up result times.

Vertica:

A DBMS that stores all data as columns, rather than rows. It operates on compressed tables by default.

Comparison Paper

Organized each test by the amount of data, the number of nodes and by different types of operations.

The researchers had to sometimes ‘translate’ some operations from one system onto another.

Comparison Paper-Analysis

The researchers took a series of regular queries and executed them on each of the different systems.

Found that Hadoop was significantly slower than both DBMS-X and Vertica.

They attributed that it was mostly due to Relational databases having more development then map databases.

GFS and Comparison paper

The comparison paper did not use any file systems.

The GFS does not support many different operations on data, it is optimized on read and append.

Most of the advantages of the GFS were barely touched on in the comparison paper, they were mostly concerned with manipulating the data.

Stonebraker talk

He had wanted to limit differences across sql systems.

Realized that relational databases were not the answer for everything.

Found that column stores were significantly faster than row stores.

Wanted there to be more options of storing data for specific purposes, did not want ‘jack of all trades’.

GFS, Comparison paper and Stonebraker

GFS was very specialized around the applications that Google needed to use it for.

It is fairly redundant and handles failures well.

GFS was not built for complex queries.

As such it is only optimized around the areas that it expects to work in (append, write, and read operations)