THE GOOGLE FILE SYSTEM AND A COMPARISON OF APPROACHES TO LARGE-SCALE DATA ANALYSIS

BEN DELGIORNO

MAY 9, 2014

GHEMAWAT, SANJAY, HOWARD GOBIOFF, AND SHUN-TAK LEUNG. "THE GOOGLE FILE SYSTEM." ACM SIGOPS OPERATING SYSTEMS REVIEW 37.5 (2003): 29. PRINT.

PAVLOV, ANDREW, ERIK PAULSON, ALEXANDER RASIN, DANIEL ABADI, DAVID DEWITT, SAMUEL MADDEN, AND MICHAEL STONEBREAKER. A COMPARISON OF APPROACHES TO LARGE-SCALE DATA ANALYSIS. 2009. PRINT.

THE MAIN IDEA

- The Google File System (GFS) is a scalable distributed file system for large distributed data-intensive applications
  - Runs on inexpensive hardware
  - Separate computers are connected together in a network
  - Delivers high aggregate performance to a large number of clients
- The GFS was created to address all of Google's data storage and usage needs
  - It is extremely fault tolerant
  - It can handle and process multi-TB data
  - Favors higher sustained bandwidth over lower latency

IMPLEMENTATION

- The architecture of the GFS can be divided into three essential components:
  - A Single Master
  - Multiple Chunkservers
  - Multiple Clients
- The Master handles all of the metadata and keeps track of the operations log
- Chunkservers store “chunks” of data (64MB) on local disks as Linux files and read or write chunk data specified by a chunk handle and byte range
- Clients interact with the master for metadata operations, but all data-bearing communication goes directly to the chunkservers

MY ANALYSIS

- I feel that the GFS is an impressive, efficient, and robust system that appropriately handles all of Google's data needs
  - It's incredibly cost effective due to the components it utilizes
  - Having only one master greatly simplifies the design of the system and makes it easier to implement
  - The large 64MB chunks reduces client's need to interact with the master and also reduces network overhead
  - The overall system is highly available due to fast recovery and chunk replication strategies
  - The system provides a high level of redundancy and fault tolerance since chunkservers store three replicas of every chunk

COMPARISON TO OTHER PAPER

- The Large-Scale Data Analysis Paper focuses on the MapReduce framework in contrast to parallel Database Management Systems
  - The MapReduce model is similar to the GFS in that they both handle large-scale data processing
  - MapReduce uses only two functions called map and reduce, which are written by a user to process key/value data pairs
    - The Map procedure performs filtering and sorting of data
    - The Reduce procedure performs a summarization operation of data
  - In contrast, basic commands in the GFS include open, create, read, write, and close
  - MapReduce does not need to have any structure or adhere to a particular schema like parallel DBMS
  - Both MapReduce and the GFS utilize clusters
  - The most popular open source implementation of the MapReduce framework is the Hadoop system

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Comprised of inexpensive components and hardware
- Great Fault Tolerance
- High level of redundancy
- The Master and Chunkservers are able to recover very quickly
- Large Chunk size at 64MB
- When a file is deleted, it is not permanently deleted, just hidden

DISADVANTAGES

- Not optimized for small-scale data processing workloads
- Redundancy can waste disk space
- Components can often fail
- At least one server is guaranteed to be down all the time
- Many replicas of chunks can result in information not being updated in all places