A Comparison Between Bigtable: A Distributed Storage System for Structured Data and A Comparison of Approaches to Large-Scale Data Analysis

Justin Esposito
5/8/14

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. 2006. Bigtable: A Distributed Storage System for Structured Data.

Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, and Michael Stonebraker. 2009. A Comparison of Approaches to Large-Scale Data Analysis

Main Idea of Bigtable

- Bigtable is a proprietary distributed data storage system built by Google on the Google File System, SSTable, Chubby Lock Service, and some other Google innovations
- Designed to reliably scale to petabytes of data and thousands of machines
- Allows for data to be stored at many locations as well as the ability to retrieve the data from any location
- Used by dozens of Google's products, such as Google Earth, Google Analytics, and Google Finance, Bigtable is considered to be widely applicable, scalable, fast, and reliable
- Considered to be a sorted map opposed to a relational database
    - Utilizes MapReduce over SQL

Implementation

Implementation is described to have three major components:

- A library that is linked into every client
- One master server
- Many tablet servers of which manage many tablets

Other features of implementation include:

- Every tablet server, which can be removed or added from clusters dynamically, is responsible for a different set of tables
  - These tablets are assigned to tablet servers by the master server in a manner that balances tablet-server load
- Most clients do not communicate with the master; rather the tablet servers
  - Results in a low usage load for the master server

Analysis of Idea & Implementation

- Provides a reliable way to manage obscenely large amounts of data (that’s Google for you)
- Extreme scalability provides a future-proof solution
    - Clusters can be scaled as simply as adding more machines to the system
- Tables are automatically split into multiple tablets if they begin to take up too much memory, which is a good feature to maintain high efficiency
- While the way data is distributed is complex, implementation is rather easy, consisting of only 3 major parts

Key Comparison Points of SQL DBMSs and MapReduce

- Data in MapReduce can theoretically have no structure (no rows or columns necessary)
  - However, customs structures must be manually programmed into the Map and Reduce programs, causing issues when other people try to utilize the data
- Programming “queries” and other things in MapReduce is, according to the comparison paper, significantly more difficult
- More flexible due to lack of structure
- Significantly more adept and handling node failures
- Scales better than SQL

Advantages and Disadvantages

Pros:

- Highly scalable
- More flexible than a traditional RDBMS
- Significantly faster in benchmark testings of load times

Cons:

- Lack of set data structure requires other programmers to look at individual code to analyze data structure
- Query equivalents and other SQL functions are considered to be more difficult
- Most other tasks outside of load times, such as join, aggregation, and selection tasking were significantly slower
- The level of scalability of MapReduce over a traditional RDBMS is currently negligible; for almost every business in the world the scale of a RDBMS is sufficient