Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber.

A Comparison of Approaches to large-Scale Data Analysis

Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, Michael Stonebraker

Alliyah Taylor
10|19|2016

Bigtable Main Ideas

i

- A distributed storage system for managing structured data
- Designed to scale to petabytes of data and thousands of machines
- A flexible, high-performance solution for Google products with varied demands.
- Achieved high-applicability, scalability, high performance, and high availability.
- Lets clients dynamically control whether to serve data out of memory or from disk.
- Sparse, distributed, persistent multi-dimensional sorted map.

Bigtable Implementation

- Composed of Three Components
  - One Master Server
    - Responsible for assigning tablets to tablet servers, detecting addition/expiration of tablet servers, balancing tablet-server load and garbage collection of files in GFS.
    - Handles Schema changes
  - Many Tablet Servers
    - Manages a set of tables.
    - Handles reads/writes for tablets it has loaded
  - A Library linked to every client
    - Clients communicate directly with tablet servers, never with the master.
- Tablet Location
  - Tablet location stored in a three-level hierarchy.
    - First level is a fire stored in Chubby that contains the location of the root tablet, which contains the location of all tablets in a Metadata table. Each metadata tablet contains the location of a set of user tablets.
- Tablet Assignment
  - Assigned to one tablet server at a time.
    - If a file no longer exists, the tablet server can no longer serve and kills itself.
    - When a tablet server terminates, it releases it's lock on a file so the master can reassign the tablets.
- Tablet Serving
- Compactions
  - Minor Compaction
    - Shrinks memory usage of tablet server and reduces the amount of data that has to be read if the server dies.
    - When memtable size reaches a threshold, it is frozen and a new one is created. The frozen memtable is converted to an SSTable and written to GFS.
  - Major Compaction
    - Rewrites all SSTables into exactly one SSTable that contains no deletion information and no deleted data.

Bigtable Impressions

- Bigtable is a great system for a company like Google that requires a high level of flexibility.
- Bigtable is not a good system to market outside of a company like Google, however, as it would need extensive support.
- I think that Bigtable is an admirable project, but I think that the lessons learned from creating it are more valuable to the public and other companies than the actual system could ever be.

Comparison Paper Main Ideas

- This paper compares two Parallel Database Management Systems with MapReduce.
  - Map Reduce
    - This model is simple as it only has two functions: Map and Reduce, that are written by a user to process key/value data pairs.
    - The nature of this model is well suited for development by a small number of programmers, but may not be appropriate for longer-term and larger-sized projects.
  - Parallel DBMS
    - Most or all tables are partitioned over nodes in a cluster.
    - The system uses an optimizer that translates SQL commands into a query plan whose execution is divided amongst multiple nodes.
    - Programmers are not burdened by the underlying storage details.

Comparison Paper Implementation

- The systems underwent five tasks to compare the performance of MR with the parallel DBMS.
  - The original MR "Grep task"
    - Each system must scan through a data set of 100-byte records looking for a three-character pattern.
    - Each record consists of a unique key in the first 10 bytes, followed by a 90-byte random value.
    - The search pattern is only found in the last 90 bytes once in every 10,000 records.
  - Selection Task
    - A lightweight filter to find the pageURLs in the Rankings table with a pageRank above a user-defined threshold.
    - For this experiment, the threshold parameter is 10.
  - Aggregation Task
    - Each system must calculate the total adRevenue generated for each sourceIP in the UserVisits table grouped by the sourceIP column.
    - Designed to measure the performance of parallel analytics on a single read-only table.
  - Join Task
    - Consists of two sub-tasks that perform a complex calculation on two data sets.
    - Each system must then calculate the average pageRank of all the pages visited during this interval.
    - Stresses each system using fairly complex operations over a large amount of data.
  - UDF Aggregation Task
    - Systems must compute the inlink count for each document in the dataset.
    - Systems must read each document file and search for all the URLs that appear in the contents.
    - Then, for each unique URL, systems must count the number of unique pages that reference that particular URL across the entire set of files.

Comparison Paper Analysis

- I think that it is good that the experimentation in this paper went beyond the original scope of the MapReduce testing.
- For complete data, I think that the experiment should be scaled up to further test the scalability of the different alternatives.
- Parallel databases seem to have a definite advantage on large scale projects currently, but that has the potential to change very soon.

Both Papers

- I think that the comparison tests had better intensive testing and broad applicability.
- The Bigtable paper did a better job of showing what real life implementation of the system looked like and demonstrated that it was a good system for Google.
- The Bigtable paper made a MapReduce approach seem much more attractive and useful, particularly when there are varied needs that must be met.
- The comparison paper leaned more towards parallel DBMSs due to their speed and structure.

Stonebraker Main Ideas

- The Stonebraker talk focused on the fact that the time of the Relational DBMS is passed.
- DBMS was once looked at as the “one size fits all” solution, which stagnated the field in the 80’s and 90’s.
- Now is a good time to be in the Database Management Field.
- New technology will battle it out for market share to determine supremacy.

Bigtable in context of Stonebraker and Comparison

- I still believe that Bigtable is a good system specifically for Google.
- I think that the ideas developed in the making of this system have potential to assist this style of system in growing more popular.
- Bigtable is a shift away from the Database Systems of the past, and follows in the trend of Database Systems innovation that is growing traction in the field.