Bigtable: A Distributed Storage System for Structured Data

By: Chang, Dean, Ghemawat, Hsieh, Wallach, Burrows, Chandra, Fikes, Gruber

Andrew Baran

April 28, 2014

Marist College, CMPT 308

A Comparison of Approaches to Large-Scale Data Analysis

By: Pavlo, Paulson, Rasin, Abadi, DeWitt, Madden, and Stonebraker

Bigtable: The Main Idea

- In 2005, Google began created a distributed storage system that could scale up to petabytes of data. This system was known as Bigtable
- Bigtable is in use in over 60 Google produces and projects, including:
  - Google Earth, Google Finance, Google Analytics, and Orkut
- Although it resembles a modern relational database management system (RDBMS), it is actually a simply data model that supports dynamic control over the data layout and format
  - Clients receive a lot of flexibility when using Bigtable
- Works in tangent with Google File System (GFS) for more reliable access and management of large clusters of data (up to petabytes of data)

Bigtable: Implementation

- At highest level of abstraction, Bigtable is a spare, distributed, persistent multidimensional sorted map
  - Index is by row key, column key, and a timestamp (64-bit integer, precision down to microseconds)
- Data is maintained in lexicographic order by the row key, which are arbitrary strings
  - Row ranges for a table are dynamically partitioned into what are called **tablets**
  - One master server controls many different tablet servers that hold the row ranges of data
- The Bigtable cluster is a cluster of a number of tables, who each consist of a set of tablets
  - Tables are automatically split into multiple tablets as the table grows
  - Tablets are stored in a three-level hierarchy analogous to that of a B+ tree

Bigtable: Implementation

- Each table in the Bigtable cluster is a SSTable file format, created by Google, to store Bigtable data
  - Provides a persistent, ordered immutable map from keys to values
  - Used for look with a single disk lookup
- Chubby, a highly-available and persistent distributed lock service, is used to check for reliability in the Bigtable cluster
  - Ensures 1 max master server, discovers tablet servers and handles their lifetime, stores Bigtable schema information, stores control lists

Bigtable: Analysis

- Very interesting and complex implementation of a distributed storage system
  - Speeds and benchmarks were very impressive, as it only takes one single disk lookup
    - Use of a similar structure to a B+ tree is also nice, as having O(log n) lookup is impressive
- Allows the use of MapReduce and other Google systems to further increase performance, which is nice
- The automated system of managing the load of each tablet server and table size is impressive
  - Coordinating this all under a single master server is an incredible accomplishment, as well as the implementation of Chubby to help assist with this

Bigtable: Comparison

- Bigtable is a offshoot of the parallel DBMS model with a mix of MapReduce features
  - Bigtable uses a distributed, multidimensional sorted map to store data
  - Can have MapReduce used on top of it for computations
  - Tablet recovery and management by master server similar to MapReduce reliability
- Has support for high-level queries
  - These queries are very fast, due to indexes and query planning / optimization, as well as being reliable
- Bigtable supports a schema defining data model, as opposed to that of MapReduces lack of schema
  - Provides for reliability, referential and data integrity, and easier access to data

Bigtable: Advantages and Disadvantages

- Advantages:
  - Performance advantages result of number of prominent technologies:
    - Indexes as B-trees to speed up access to data
    - Important storage mechanism, using a schema structure
    - Aggregation and query planning and optimizations
  - Amount of tools available and long history adds to its reliability as a system
  - Amount of disk I/O and hardware overhead of loading and accessing data is much lower for Parallel DBMS
  - Takes much less code to perform the equivalent MapReduce tasks

Bigtable: Advantages and Disadvantages

- Disadvantages:
  - Harder to configure and install the DBMS
  - Response to failure is much slower than that of MapReduce
    - Will restart long query after loss of just a single node in a cluster
  - Extensibility of the DBMS with user-defined types is limited
  - Lack of no proper SQL standard, as each DBMS is different with their own proprietary extensions