Review, BigTable

Robert Hoff

November 9, 2011

1 Summary

Bigtable [1] is a distributed storage system for persistent data, it was developed by Google to meet demands of their applications in terms of scale and availability. Simplicity was emphasised in the design to meet requirements of size, so the system mainly supports key-value lookups and some basic operations, such as traversing the data. In this respect the system looks more like a table than a database; the data is denormalised. Google uses Bigtable today to provide data for many of their key products such as the search engine, Google Earth and Google Analytics.

2 The Problem

Foremost, Google's search system relies on indexing most of the web, which requires persistent storage of large amounts of data (order of petabytes), which has to be available for continuous reassessment of it's search results. Similar requirements for referencing large datasets also emerges in other applications, such as Google Earth and Google Finance.

The data needs to be structured, using methodologies similar to familiar database systems, principally using key-value pairs and some simple queries. The main difference of Google's system compared to existing projects are size and reach. The design goals of the storage system, called Bigtable, are to reliably scale to petabytes of data using thousands of storage nodes, it should offer wide applicability, high performance and high availability.

3 Approach

Bigtable is implemented on top of the Google File System (GFS). A Bigtable is split between files using a file format called *SSTable*. The files are split into data blocks and block indexes used to locate data by binary search.

In general there will be numerous clients reading and writing to the data, in the case of the search engine the multiple clients are both analysing the data, and the crawlers are continuously modifying and adding to it. The

lock service that protects the data for consistency is called Chubby [3], also developed at Google.

Data is addressed by walking a three-level tree hierarchy, divided into logical segments called tablets. The top level is the root tablet, and the second level are other metadata tablets that store further references to the data itself (stored in the SSTable Files). The second and third level of these tables can grow to any arbitrary sizes by an incremental process of splitting existing tablets in two. Each tablet is assigned to one tablet server, implemented by the GFS, in this way a Bigtable can be supported by more than one GFS.

4 Evaluation

In tests, random reads are slow mainly because the underlying GFS needs to transfer files in blocks of 64 KB even though only a partial amount of data is required. Random reads also require walking the whole tree structure starting from the root. Sequential reads are faster, both because of a single underlying transfer may address multiple requests, and the client can keep requesting the data from tablets further down the hierarchy. For applications with random reads the block size can be reduced. Writes are faster than reads because of a special support for appending data by the GFS.

Throughput increases significantly with the number of tablet servers, as expected, but not quite linearly because of load imbalance. Given enough machines the system was able to saturate the bandwidth of the network.

Considering real applications, Google Analytics is well suited for Bigtable because the data it collects is usually appended sequentially to the table. For Google Earth, the main data is satellite imagery of the world's surface. The demands are more unpredictable, with tens of thousands of queries per seconds. The preprocessing relies on MapReduce [2], and the table is replicated across hundreds of tablet servers.

5 My Opinion

Bigtable is a key piece of Internet infrastructure.

6 Questions

1. Bigtable looks more like a table than a database, it doesn't support object-relational features of typical query languages, such as the JOIN command. So wondering if there are any applications that rely on more complex querying results that are growing to the type of 'Internet-scale' sizes seen by products like Google Earth?

References

[1] Chang F. and et al. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems*, 2008.

[2] Dean J. and Ghemawat S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 2008.

[3] Burrows M. The chubby lock service for loosely-coupled distributed systems. *OSDI symposium on Operating systems design and implementation*, 2006.