Bigtable: A Distributed Storage System for Structured Data
And
A Comparison of Approaches to Large-ScaleData Analysis

Big Data Paper Summary

Hannah Riedman 3-2-17

Bigtable: A Distributed Storage System for Structured Data

MAIN IDEAS

- A simple data model with interactive client control over data layout and format
  - Not a full relational data model!
  - Clients can control whether to serve data out of memory or from a disk
- Scattered, dispersed, and perpetual multidimensional sorted map
  - Indexed by row key, column key, and timestamp
- Used in more than sixty Google products and projects
  - Known to handle demanding workloads

IMPLEMENTATION

3 major parts:

- a library that is linked into every client
    - Client doesn't communicate with master and only directly communicate to tablet servers.
- one master server
    - Assigns tables to tablet servers
    - Detects addition and expiration of tablet servers
    - Balances tablet server load
    - Garbage collects files in GFS
    - Handles schema changes
- tablet servers
    - Manages a set of tables
    - Handles read and write requests
    - Splits tables that have grown too large

Bigtable: A Distributed Storage System for Structured Data

ANALYSIS

I think Bigtable is a great storage system to use for structured large amounts of data. It is a little different than relational databases so it might be hard to adapt to using it but it is proven successful in the many Google products that use it. Having newer systems that aren’t relational databases is important in this golden age of database research. We need to explore other methods of sorting data and this is a step in the right direction.

A Comparison of Approaches to Large-Scale Data Analysis

MAIN IDEAS

- Comparing the new MapReduce computing model and parallel SQL database management systems (DBMS)
    - Judged on:
        - Performance
        - Development complexity
- Rise in Cluster Computing has lead to ideas of different approaches to large data analysis
    - Cluster Computing is when you take large numbers of processors working in parallel to solve computing problems.

A Comparison of Approaches to Large-Scale Data Analysis

IMPLEMENTATION

- MapReduce
  - 2 functions
    - Map
      - Reads, records, then filters/transforms records and outputs set of intermediate records in form of key/value pairs
    - Reduce
      - Input for each instance is from files that are transferred over the network from the map nodes local disks.
      - Combines records in a certain way and writes the records to an output file

A Comparison of Approaches to Large-Scale Data Analysis

IMPLEMENTATION

- Parallel DBMSs
  - Support all standard relational tables and SQL
  - Data is supported on multiple machines
  - Two key aspects
    - Most tables are partitioned over nodes in a cluster.
    - System uses optimizer that translates SQL into a query plan that is divided among multiple nodes.

A Comparison of Approaches to Large-Scale Data Analysis

ANALYSIS

DBMS took longer to load data and set up but it worked alot better than MR, DBMS-X was 3.2 times faster than MR. It was surprising that the newer system was easier to set up but gave more disappointing results. Also, MR is far better when dealing with hardware failure. This paper goes in depth on advantages and disadvantages of one system over another and I think we should focus on changing, and creating systems with all the positive features we strive to have in a data system.

A Comparison of Both Papers

The second paper talks about the MapReduce called Hadoop and BigTable is an application of using it in a distributed file system. We have to note that since BigTable is from Google, google had its own MapReduce algorithm and did not release their code to be tested for the research of the comparison paper. So when arguing for disadvantages and advantages between MR and a parallel DBMS, we cannot know how google's MR will hold up. After reading the paper about BigTable, it sounds like an improvement in the right direction with a step away from the old models.

"10 Year Test of Time" Talk

- RDBS are not universal and using row store will not work for everything (one size fits none)
- There are many different types of data models/engines
  - Column stores are 2 orders of magnitude faster than row stores
  - Move toward main memory deployments
  - Data models with noSQL
  - Complex Analytics
    - Use arrays not tables. This will create a move toward less SQL, tables etc.
  - Stream processing engines
  - Graph Analytics
    - In column store, array engine, special purpose graph engines but no row stores
- Great time to be a DBMS researcher!

Final Comparison

I think it is time to look toward newer and more innovative systems. Things like BigTable might not be easy to use yet, or be the best in the world but they are important steps in innovation and moving toward a better system to handle all this big data that exists in the world. Some old systems may be faster or reliable but they may not be completely feasible anymore. Newer systems aren’t going to be perfect--at least yet--but they are important steps in solving this crisis of almost infinite amounts of data in our fast paced world.