Michael Guarino
CMPT308_111
March 12, 2016

1)Bigtable: A Distributed Storage System for Structured Data
Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., . . . Gruber, R. E. (2008). Bigtable. *ACM Trans. Comput. Syst. TOCS ACM Transactions on Computer Systems, 26*(2), 1-26. Retrieved March, 2016.

2)A Comparison of Approaches to Large-Scale Data Analysis
Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., Dewitt, D. J., Madden, S., & Stonebraker, M. (2009). A comparison of approaches to large-scale data analysis. *Proceedings of the 35th SIGMOD International Conference on Management of Data - SIGMOD '09*. Retrieved March, 2016.

3) Michael Stonebraker on his 10-year Most Influential Paper Award at ICDE 2015
Michael Stonebraker on his 10-year Most Influential Paper Award at ICDE [Motion picture]. (2015).

Slide 1: Bigtable, the Main Idea

- Bigtable is a proprietary distributed storage system built on Google File System for managing structured data at Google.
  - Used by more than 60 Google products and projects (Google Earth, Google Analytics, etc.)
- Goals of Bigtable: wide applicability, scalability, high performance, and high availability.
  - Bigtable clusters used span wide range of configurations to support unique demands for storage.

Slide 2: Bigtable, Implementation

- **Bigtable Data Model:**
  - Bigtable does not support a full relational data model.
  - Bigtable is a “distributed, persistent multidimensional sorted map.”
  - Column Oriented storage.
- **Bigtable Map:**
  - indexed by a row key, column key, and timestamp (therefore 3-dimensional mapping)
  - Each value represented as an uninterpreted array of bytes.
  - Rows
    - Data under single row key is atomic.
    - Maintained in lexicographic order by row key.
    - Row dynamically partitioned.
    - Each row range is called a tablet.
  - Columns
    - Column keys are grouped into sets called Column Families.
    - Column Families form the basic unit of access control.
  - Timestamps:
    - Each cell can contain multiple versions of the same data indexed by the timestamp.
- **Google SSTable**
  - File format used internally to store Bigtable data
  - Persistent, immutable map from keys to values.
  - SSTable contains a sequence of blocks, block indexes used to locate blocks(binary search time).
- **Chubby:**
  - Persistent distributed lock service.
  - Ensures that there is at most one active master(serving requests) at a time.

Slide 3: Analysis of Bigtable and implementation

- Column oriented storage
  - Increases flexibility and scalability of storage while allowing speed of querying
  - Column Families implemented avoid excessive use of joins
- Lexicographic order by row key allows clients to group related rows together.
- Storage of Bigtable metadata on top of SSTable allows binary search time of Bigtable data.
- Bigtable’s dynamic structure needed to support unique needs of clients and products leaves Bigtable with less referential integrity and could produce dangerous data.

Slide 4: Comparison of Approaches Large-Scale Data Analysis, Main Idea

- DBMSs compared to Map Reduce for processing data in parallel to solve computing problems.
- DBMSs
  - Processed data in parallel much faster under all testing conditions
  - DBMSs separated schema from application which allows for higher level abstractions storing a set of catalogs that can be queried
- Map Reduce
  - Processed data in parallel slower than DBMSs under all testing conditions
  - Data processing in parallel improved when the number of nodes per cluster increased due to Map Reduce high start up cost.
  - Map Reduce leaves great flexibility in the hands of the programmer which increases the likelihood of corruption of data.

Slide 5: Comparison of Approaches Large-Scale Data Analysis, Implementation

- **Schema Support:**
  - DBMSs
    - Separate schema from application protecting data and enforcing data integrity.
  - Map Reduce:
    - MR framework and its underlying distributed storage system has no knowledge of rules of schema which increases likelihood of data corruption.
    - Often programmer must write a custom parser in order to derive appropriate semantics and structure for their inputs.
- **Data Distribution**
  - DBMSs
    - Parallel query optimizer balances computational workload which minimizes data that needs to be transmitted.
  - Map Reduce
    - Programmer must manually decide where to schedule map instances before passing documents to a reduce function which groups files by original site then sends them.
- Map Reduce is highly flexible and implements an object-relational mapping pattern which has high expressive capabilities.
- **Indexing**
  - DBMSs support indexing which accelerates access to data.
  - Map Reduce is such a simple framework that it does not provide indexing.
- **Fault Tolerance**
  - Map Reduce
    - High fault tolerance because Map Reduce scheduler can automatically restart the task on an alternate node. Outputs materialized locally instead of being streaming to the nodes running the reduce tasks while materializing intermediate results to files each step of the way.
  - DBMSs
    - DBMSs transaction based so if error occurs transaction must be completely restarted.

Slide 6: Comparison of Approaches Large-Scale Data Analysis, Analysis

- Map Reduce model is well suited for development environments with small number of programmers and limited applications; however, because there is a lack of constraints, it may not be appropriate for longer-term and larger-sized projects.
- Map Reduce doesn’t not perform as well as DBMSs; however, despite the performance gap closing as scale increases the likelihood of an error also increases.
- Map Reduce currently has flaws in its implementation that make is a less than optimal solution for data processing.

Slide 7: Comparison of ideas and implementations of Paper 1 and Paper 2

- Paper 1: Bigtable
  - Column oriented database that allows for great flexibility to accommodate the wide range of client needs.
  - Supports locks to prevent transaction errors via Chubby.
  - Block indexes in SSTable provide ability to search in binary time.
- Paper 2: Map Reduce vs. DBMSs for data processing
  - Map Reduce is very flexible which adds to its expressive power but can easily lead to corruption of data specifically at scale.
  - Map Reduce does not provide indexes therefore processing can be costly and timely.
  - DBMSs process data faster and safer; however, can be restricting.
- Comparison
  - Variety of unique client needs leading to many equally unique differently implemented data management solutions.
  - Still great need to protect data from itself by enforcing data integrity.

Slide 8: Main ideas of Stonebraker talk

- Field of Database research relatively static from 1980s and 1990s.
- Due to acceleration in growth of technology a variety of different Database solutions have appeared to cater to the unique demands of new technology.
- Stonebraker believes that there will not be one dominating database solution there will be many.

Slide 9: Advantages and disadvantages of Paper 1 in context of Paper 2 and Stonebraker

- Bigtable has to accommodate to a variety of different client needs and must scale to tremendous size therefore it must be very flexible while still robust.
- A drawback to Bigtable is that it is too universal in its design, trying to fit the varied needs of the Google's entire corporation and clients.
- Stonebraker put emphasis on not having one universal solution but rather having many that cater to the unique demands of different software markets.