Photo by Ferdinand Stöhr on Unsplash

Selected Works:

Hive – A Petabyte Scale Data Warehouse Using Hadoop
A Comparison of Approaches to Large-Scale Data Analysis
Michael Stonebraker on his 10-Year Most Influential Paper Award at ICDE 2015.

Big Data Paper Summary

Andrew Bauman
CMPT 308N
Prof. A. Labouseur
October 30, 2017

Slide 1- the main idea of the paper you chose

- The amount of data the business intelligence industry manages is growing at an incredible rate.
- Employing old methods to work with data are inefficient and time consuming.
- New methods of managing big data tend to be expensive.
- Hadoop, a popular open-source MapReduce implementation, while a low-cost solution, is a low-level language, requiring developers to create custom programs for even basic tasks.
- In seeking a solution, Hive was created by Facebook to operate above Hadoop and simplify the process of working with MapReduce.

Slide 2 - how that idea is implemented

- “Hive supports queries expressed in a SQL-like declarative language - HiveQL” (Thusoo, et. al.).
- Additionally, the language allows for users to include their own map-reduce scripts to build on and enhance their HiveQL queries.

Slide 3 - your analysis of that idea and its implementation

- Having the ability to work with a language which appears familiar to developers affords them the ability of completing tasks faster while also taking advantage of the more efficient MapReduce.

Slide 4 - the main ideas of the comparison paper

- There has been an ongoing debate as to whether or not MapReduce (MR) is superior to traditional parallel SQL DBMS.
- Working with a system of 100 nodes, the authors of this paper seek to compare the two with the hope of providing an answer.

Slide 5 - how those ideas are implemented

- The paper discusses differences existing between MR and Parallel DBMS:
  - *Schema Support*: MR does not require schema to be specified upon initialization like DBMS, however it must still be managed as new data is being added. Therefore "when no sharing is anticipated, the MR paradigm is quite flexible" (p. 167)
  - *Indexing:* MR does not provide built-in indexes and if programmers desire to employ them to speed up searches, they must do so themselves.
  - *Programming Model:* MR is low-level and can be difficult to understand, although alternatives like Pig and Hive have become available, allowing users to use high-level languages to work with MR.
  - *Data Distribution*: Using the Map and Reduce functions, an MR query must assemble all data before refining it. DBMS queries can specify to search using a 'where' clause rather than having the system search sequentially.
  - *Fault Tolerance:* If a node failure of some sort occurs in an MR system, the scheduler can automatically re-assign the failed tasks to another node. Conversely, if a node failure occurs in a DBMS, the entire query must be restarted.

Slide 6 - your analysis of those ideas and their implementations

- While both MapReduce and parallel DBMSs can be implemented to work in most situations and, in the end, solve the same tasks, there are trade offs to using each one over the other.
- MapReduce appears to fit applications like twitter and Facebook, providing data from queries from real-time streams of information.
    - It is okay if a few tweets do not appear in a list of results right away due to a node failure while performing a query on twitter. Performing the same search on a DBMS would take significantly longer if a failure occurred. Therefore, MapReduce is a good use for social media with big data.
- Parallel DBMSs are better, however, for situations where the user administering the queries requires all information to be outputted together in order.
    - One example could be any situation working with transactions of some sort.

Slide 7 - comparison of the ideas and implementations of the two papers

- As mentioned in the comparison piece, MapReduce is a low-level language and can be difficult for some to use.
- Luckily, programming languages like Facebook's Hive have been created to remedy this issue, allowing users to use familiar SQL-like commands and functions to query data organized in a MapReduce structure.

Slide 8 - the main ideas of the Stonebraker talk

- Stonebreaker reminds his audience that in the thirty years leading up to the 2000s there was an attempt to form a “one size fits all” RDBMS. Disproved this in 2005 paper.
- He claims legacy systems including DB2, Oracle, and SQL server are “One Size Fits None”
- Goes on to explain the diverse existence of engines and how row storage systems do not solve problems in most markets.
- Touches upon opportunities for this technology in the upcoming future.

Slide 9 - advantages and disadvantages of the main idea of the chosen paper in the context of the comparison paper and the Stonebraker talk

- Stonebreaker touches upon the fact that markets are diverse and come with their own set of unique problems to solve.
  - The comparison paper recognizes this in that it goes on to explain the trade offs of using MapReduce over RDBMS.
- Towards the end of his talk, Stonebreaker mentions the future opportunities for the industry. As technology continues to advance and inevitably becomes faster, the systems including MapReduce and RDBMS will also become more efficient.
  - Additionally, new, alternative methods are also likely to come into existence and might serve the markets currently employing the other two systems in a more effective manner.