Pregel: A System for Large-Scale graph Processing, *A Comparison of Approaches to Large-Scale Data Analysis*, and Michael Stonebraker's ICDE talk.

Grace McCue

15 March 2015

- Malewicz, Grzegorz, Matthew H. Austern, Aart J.c Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. "Pregel." *Proceedings of the 2010 International Conference on Management of Data - SIGMOD '10* (2010): n. pag. Web.
- Pavlo, Andrew, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. Dewitt, Samuel Madden, and Michael Stonebraker. "A Comparison of Approaches to Large-scale Data Analysis." *Proceedings of the 35th SIGMOD International Conference on Management of Data - SIGMOD '09* (2009): n. pag. Web.
- One Size Fits All- An Idea Whose Time Has Come and Gone. Perf. Michael Stonebraker. N.d.

Pregel: A System for Large-Scale Graph Processing

- This paper explains what Pregel is:
  - A framework for processing large graphs that is expressive and easy to program
  - It is a fault tolerant platform with an API that is sufficiently flexible to express arbitrary graph algorithms
  - It consist of a sequence of iterations called supersets and vertices
  - It is well suited for distributed implementations
  - It is free of deadlocks and data races
  - It allows users to program graph algorithms while managing distribution

Implementation of Pregel

- The pregel program is implemented through a master and many workers on a cluster of machines
- It is similar to MapReduce but has a natural graph API and much more efficient support iterative computations over the graph
- It is a model suitable for large scale graph computation that has a high production quality, it is scalable, and has a fault tolerant implementation

Analysis of Pregel

- I believe the way that the creators of Pregel decided to go about processing large graphs is well thought out and could potentially be of great use to someone who needs to put large amounts of data into large graphs that are easy to program and expressive in the way the content is given.
- I believe it was a great idea to make Pregel similar to MapReduce so that it is user friendly, yet has some changes that will allow the user to better implement large graphs with qualities such as the natural graph API and much more efficient support iterative computations over the graph.
- Having Pregel implemented in a way that has a master and many workers with in a cluster of machines is a smart way to have Pregel give and receive commands while having many workers to carry out those commands and get the job done faster.

A Comparison of Approaches to Large-Scale Data Analysis

- MapReduce
  - Attractive because of its simplicity: consists of only two functions, Map and Reduce
  - Uses cluster computing with many nodes to help do the work that needs to be done
  - Good for environments with few programmers and a limited application domain
  - Structure of data is up to the programer (does not have to be one), but must be agreed on by all programers involved
  - Do not provide built in indexes, again must be agreed upon by all programers
  - If a unit of work fails, the task is automatically restarted on an alternate node
  - Takes time to start up before all nodes are running at full capacity
  - Are not able to change the layout of input data, but it is faster and easier to load input data
- Parallel Database Management Systems
  - Paralleled execution is enabled for the DBMS's because most tables are partitioned over the nodes in a cluster and an optimizer is used that translates SQL's commands into queries whose executions are divided over several different nodes
  - Data must fit in rows and columns
  - Use hash or B-tree indexes
  - If a unit of work fails, the entire query must be restarted
  - Are considered to always be "warm" and waiting for a query so never really have to start up
  - Have the opportunity to reorganize the the input data file at load time

Implementation of MapReduce versus DBMS

- In order to further compare and contrast these two systems, several test were done:
  - Data loading: The DBMSs slower in this task then the MapReduce system.
  - Task Execution: The MapReduce system was noticeable slower in this task then either of the DBMSs.
  - Selection tasks: The MapReduce system is slower then the DBMS's.
  - Aggregation task: The two DBMSs outperform the MapReduce system.
  - Join tasks: The two DBMSs outperform the MapReduce system.
  - System installation, configuration, and tuning: MapReduce systems were easier to install then the two DBMSs.

MapReduce vs DBMS analysis

- MapReduce seems to be the better system if there is just one programer and no changes need to be made to any of the inputed data.
- If there is more then one programmer, or changes could potentially need to be made to the inputed data, then a parallel DBMS is the best choice. Changes can easily be made and all programers do not need to agree on one thing before proceeding in order to work on the task at hand.
- If potential mistakes could occur while implementing the data and failures of particular units of data are bound to happen, MapReduce may also be the quicker option until the kinks are worked out in order to not have to keep restarting entire queries for just one failed piece of data .

Comparison of Pregel, MapReduce, and DBMSs

- Pregel has taken aspects of MapReduce and API natural graphs in order to create a program that will process large graphs that are expressive and easy to program.
- MapReduce by itself does not leave a lot of room to change data as needed, but allows the programer to program the necessary tools to complete the task how he/she wants it.
- Pregel will allow the user to easily create a program that will process and managed large amounts of data into large graphs that can give this data meaning and allow it to be used for information that will be valuable to many individuals and companies.
- With Pregel using cluster computing, it should be speedy and flaws and failures will be handled with ease in a way that will not affect the entire program.
- Pregel will potentially use the best qualities of the MapReduce programs, while still allowing some better qualities of DBMSs such as allowing the programer to change and reorganize data with little difficulty.

Michael Stonebraker's ICDE talk

- People were attempting to make RDBMSs "universal", one size that fits all. Stonebraker wrote a paper in 2005 to contradict this, that one size will never fit all. One size fits none.
- Data warehouses either are or are going to be entirely column stores, these are faster and more efficient then row stores.
- Put all data in main memory
- NoSQL markets - no standards, not traditional
- Data scientist are going to replace business analysts
  - define data on arrays, not on tables in order to be way faster then in SQL
- Streaming market is alive and well. It is not based on traditional row stores. More of a work flow of record processing with SQL like characteristics.
- Graph analytics can be done by column store or array engine.
- Huge diversity of engines, all of which cater to a specific applications.
- Traditional row stores will have no place in any of these markets, meaning one size fits none.

Advantages and Disadvantages of Pregel

- Advantages
  - It is a fault tolerant, flexible platform
  - It is easy to program and processes data into large expressive graphs
  - Takes the good parts of both MapReduce and database management tools for the data itself such as the best ways to store the data and implement the data into the graphs
- Disadvantages
  - May have some of the disadvantages of MapReduce such as hard to change data, and better with just one programer
  - Uses a similar row and column platform to store data, so it is not entirely column stores like Stonebraker would prefer