The Google File System
(http://labouseur.com/courses/db/papers/ghemawat.sosp03.gfs.pdf)

A Comparison of Approaches to Large-Scale Data Analysis
(http://labouseur.com/courses/db/papers/pavlo.sigmod09.comparison.pdf)

Michael O'Rourke
Due: 5/8/2015

Main Ideas: The Google File System

- The Google File System (GFS) is aimed to simplify data across many platforms and pieces of hardware

  - breaks the total amount of information and spreads it in chunks to various devices

  - uses a single master (not unlike the Zerg Overmind) to dictate how the information communicates and how it is organized

- A big part of the GFS is that multiple users can access and use, modify, update, etc to the database at the same time without sacrificing speed or performance

The Google File System Implementation

- The big issue with using the GFS, as the article stated, was that parts break and require fixing. This downtime is not contributory to the overall scheme of things. This means the system would require constant monitoring.
- The paper explained further issues occurring with parts breaking and how they should be spread to many pieces of hardware so that the entirety of any entity does not go down if a single part breaks, rather a portion.
- Another note is that the way the GFS uses memory to store the data is had a likelihood to cause memory waste. It would allocate more space to a particular piece of data than needed. However, this does allow for updates on the data and expansion if needed.

My Thoughts about The Google File System

- My first reaction was that this is a cost effective method to create a large database to store a lof of data very effectively.
- Overall I think this could be very effective as a method to store a lot of data despite the failures of "commodity" hardware, as the paper addressed it.
- Although this was originally intended just for Google to use it I feel that it has a much broader possibilities

Main Ideas: *A Comparison of Approaches to Large-Scale Data Analysis*

- MapReduce (MR) – very simple two function using the commands Map and Reduce

  - easy to run and use but fails when it comes to speed

  - benefits when a failure occurs due to "saving" after each action which reduces its speed

- Parallel DMBS – uses a dual system which acts much quicker than MapReduce

  - more complex to run and has draw backs on failures meaning that it must redo more operations from the last point than MR

  - benefits from much faster speed when not in a failure mode as it does not "save" any intermediate data unlike MR

A Comparison of Approaches to Large-Scale Data Analysis Implementation

- MR is more hands on and requires constant tweaking to get it right and have a lot of fine tuning out of the box.

  - map takes data and passes it in and produces information that reduce can utilize

  - reduce takes this info and summarizes based on the criteria chosen

  - is not running on OS start

- DMBS is mostly configured before any code is run, upon install most configurations are set up and work out of the box.

  - provides many functions which all operate under their own conditions

  - runs on OS start and is considered “warm,” as the author states, to be used immediately

My thoughts about *A Comparison of Approaches to Large-Scale Data Analysis*

- I feel these two systems, MR and DMBS, are systems that serve specific functions.
- MR seems like a much simpler program to work with and run despite constant tweaks
- DMBS seems like a much more complex, but much more rewarding system with a user who knows how to handle the data

Comparison

- As the GFS is used as a way to store big data the same can be said about the MR and DMBS, however the way in which these operate is vastly different.
- The GFS is a much cruder way to store and access the data while the DMBS and MR are a much more directed and specific method of accessing and using the information.
- While the GFS was created with a specific purpose the DMBS and MR were meant as much broader scope systems.

Main Ideas: The Stonebraker Talk

- No single database will ever work best on everything that would require one
- What was thought as correct and working previously has fallen away in light that no single database system will work helping to prove the previous point
- Each system will eventually work toward having a specific type of database for its own needs

Advantages and disadvantages of The Google File System against the other paper and the Stonebraker talk

- The GFS is both direct in its use and can hold a large amount of information, the downside is that it was intended for a specific use and has to be adapted to others
- The MR and DMBS systems are good for broad scope used and are scalable. However, the DMBS is vastly more scalable than MR.
- The Stonebraker talk addresses the concerns of big data issues and the future of big data. His main being that no single database will cover all of the database spectrum needs.