DATA WARS

A NEW HOPE

By Joe "Lucas" Strauss
5/09/15

The Google File System by
Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

A Comparison of Approaches to Large-Scale Data Analysis by
A. Pavlo, E. Paulson, A. Rasin, D.J. Abadi, D.J. DeWitt S. Madden, and M. Sontebraker

IDCE 2015 10-Year Most Influential Paper Award Presentation
Michael Stonebreaker

The Google File System (GFS)

Google has left behind data normalization for a new form of control. Armed with the phase, “data failures are the norm rather than the exception.” A new file management system has been created to elevate a few key elements:

- Must support hundreds of inexpensive hardware devices (i.e. cheap disks and obsolete hardware)
- Must support large files usually spanning over 1GB in size in lieu of segmentation
- most files are mutated by appending new data rather than overwriting existing data
- Bandwidth is more important than low latency

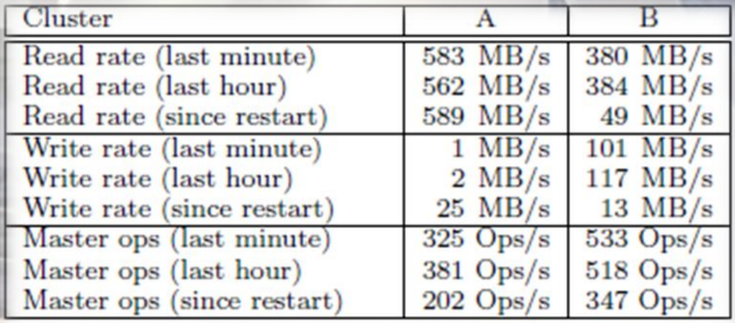
The Google File System (GFS)

Google has developed a Master & Slave system. The architecture is then broken down to a single master sever which contains the indexes for the slaves. The slaves are “chunkservers” which contain 64MB chunks of data. A client requests the data from the Master. The Master provides the index location to the client who then accesses the chunkserver. All communication then happens between the client and chunkserver. At the end of the mutation, the chunkserver reports the changes back to the Master.

- All transactions between clients and chunkservers are atomic
- All files are saved in a Linux FHS Hierarchy with the ability to create, delete, open, close, read, and write files
- Records are made via a Snapshot and Record Append system in which a snapshot is taken of the file and a copy is made with the appended data.

The Google File System (GFS)

Cluster A is used regularly for research and development by over a hundred engineers. Cluster B is primarily used for production data processing.



Cluster	A	B
Read rate (last minute)	583 MB/s	380 MB/s
Read rate (last hour)	562 MB/s	384 MB/s
Read rate (since restart)	589 MB/s	49 MB/s
Write rate (last minute)	1 MB/s	101 MB/s
Write rate (last hour)	2 MB/s	117 MB/s
Write rate (since restart)	25 MB/s	13 MB/s
Master ops (last minute)	325 Ops/s	533 Ops/s
Master ops (last hour)	381 Ops/s	518 Ops/s
Master ops (since restart)	202 Ops/s	347 Ops/s

We see that there is a dramatic difference between the read and write time in the for the clients. Particualrly in where there are numerous clients accessing the data. Most notably is once the chunkserver is restarted how quick the writes begin. This means that there becomes a significant backup in appends on the data. A chunkserver was specifically taken down to failure and was able to be replicated again within 30 minutes. Overall, the file system was a success because of the generation of extensive logs and applications to study those logs when errors occur. It would seem that this approach will replace and crush normalized databases.

A Comparison of Approaches to Large-Scale Data Analysis

- This is an analysis to see if traditional DBMS such as SQL relational tables are obsolete to newer MapReduce large-scale data analysis methods.
- MapReduce only has two functions that are used to process key/value pairs. Its rival, parallel DBMSs uses optimizers to divide execution over multiple nodes.
- It is the author's stance that in fact large-scale analytics tools such as Hadoop (GFS) are designed not to handle real-word scenarios.
  - Rarely are "computer-clusters" of more than a couple hundred need and never a thousand.These are the following ideas that will be tested to compare "cluster-computing" to traditional methods:
- Why not use parallel MSDB rather than cluster-computing?
- Are potentially quicker results worth not having a well-defined schema? Whereas MR permits arbitrary formats.
  - SQL is a language that is quick to learn an can be used even by a novice whereas it is essential for programmers to specifically define the way in which data can be searched and train other on how to use those methods. Also the complication of the newly defined language many times needs to be run by third party tools such as "Pig".

A Comparison of Approaches to Large-Scale Data Analysis

- In order to test the MR computer-cluster against traditional methods of DBMS set up Hadoop system with 256MB data chunks with a max heap of 1024MB which totaled 3.5GB per node. They setup a DBMS-X (Parallel SQL) system and Vertica a parallel SQL specifically designed for large data warehouse purposes.
- The team used five tests:

1. Using the test from the original MapReduce Paper:

   Grep task – each system must scan through a data set of 100-byte records looking for a 3 character pattern. The search pattern is only found in the last 90 bytes one in every 10,000 records.

2. Four addition tasks based on HTML processing. Each node is given 600,000 randomly generated docs and URLs. They also generated two additional data sets meant to emulate server traffic log files. The tests were on four basic functions:

   1. Loading data into temporary tables
   2. Pattern searching for particular fields
   3. Using a light-weight filter to find the page URLs with a pageRank above a user defined threshold.
   4. Each system must calculate the total adRevenue generated for each sourceIP in the User Visits table (20GB/node), grouped by the sourceIP column.

A Comparison of Approaches to Large-Scale Data Analysis

- The Rebel Alliance eventually lost Hoth, as did this paper. There were three key points that crippled DBMSs battle verses the empire that is Google.
  - The initial cost to setup the DBMS systems is substantial compared to the Hadoop system
  - The setup of the DBMSs (specifically DBMS-X) were difficult and needed a lot of ‘tweaking’ to work as desired
  - Hadoop was relatively quick to get running and Pig made tasks run simple to execute as opposed to the “trial ‘n’ error” of SQL commands

Inevitable ably there is a key lesson to be learned here. The experiments found that Parallel DBMSs do in fact compete with MR and in many cases perform better; however for the past 30 years these systems have been have enjoyed the luxury of consumers being forced to live with the product made by DBMS vendors. Software like Hadoop is the antithesis of these closed-source behemoths. They are essentially designed by the consumer for the consumer. Instead of a “one-size-fits-all” they are more like a buffet line. If relational DBs are going to stay main-stream, old ideas need to be scrapped and vendors must listen to what consumers want.

THESE X-WINGS AIN'T GONNA PILOT THEMSELVES
BECOME PART OF THE REBEL ALLIANCE!

Choose Wisely

- The Google file system laid the underbelly of its premise on two highly flawed ideas
  - To create a file system that would be robust enough to compensate for Pentium III processor machines and hard drives that are not designed for data center use. This is not “real world” examples. Hardware (i.e. transistors/dollar and GB/dollar) has dropped in price considerably.
  - That a relational table will always give your users lower bandwidth access. There are circumstances in which the front-end configuration may be difficult but having a well-formed schema will create easier code and more robust applications with excellent data integrity in the future.

- The comparison paper found a “real-world” lesson that trumped all the data
  - DBMS must evolve with Big Data. Big Data is not a fad that will pass. It is the “Yoda” of how businesses invest and Data Analytics is the “Force” that it has chosen. DBMS must incorporate these forms of data query into applications for consumers. They must also begin to refine the software so that it is not just for the “Database Elitist”.

Stonebreaker: “One-Size-Fits-None”

- Michael Stonebreaker is one of the leading pioneers of the DBMS movement. Not just as one of the designers of PostgreSQL but also a scientist who has advanced SQLs theoretical bounders.
- Stonebreaker wrote a paper over a decade ago in which he came to the conclusion that Classical DBMSs will not be efficient enough to hand the needs which consumers are looking for.
- His paper has been a accurate as DBMSs are losing shares in almost all traditional DB markets and have little ground in the newer markets such as Big Data.
- Stonebreaker’s idea is that DBMS has not evolved because traditional designers fear losing current market shares by revamping their software to implement new methods of database such as MapReduce.

STAR WARS
HAN SOLO
STAR WARS WALLPAPER NO. 7 . MADE BY IG-88. STAR WARS: ORD MANTELL HOLONET. WWW.STARWARS.PRIV.PL

Slide 10

A NEW HOPE

Michael Stonebreaker's paper was a foreshadowing of things to come. Google has created a new force in the DBMSs that will not fade. It is now up to creators to choose wisely. Leaving data normalization behind will destroy data integrity as we know it in order for convenience. The data community must be ever vigialant and Codd alive!