Yul Puma

COMP 440

Mushkatblat

18 May 2017

SQL Relation Database versus Hadoop

To compare/contrast between SQL relational databases and Hadoop, it is important to understand how they are used individually. A SQL relational database is a database structured to recognize relations among stored items of information. There are multiple tables associated with a relational database each with primary and foreign keys that connect tables to each other to obtain more information. Hadoop, on the other hand, is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. Judging by their uses a relational database schema can be implemented onto Hadoop since it is for organizational purposes and Hadoop stores large sets of data. In terms of differences, looking at ACID and CAP theorems, ACID consistency is all about database rules since a relational database focuses on its schema while CAP consistency is all about promises that every copy of the value is spread the same across the distributed system where in Hadoop organization is not its prime use. Knowing when to use each as well as the use cases, functionalities, and industry uses is important when deciding relational database over Hadoop, vice versa.

Through the papers I have been reading, it seems that Relational Database Management Systems (RDBMS) AND Hadoop share similar functionalities. For instance, RDBMS and Hadoop can collect, store, process, retrieve, extract, and manipulate data (Le). When talking about ACID and CAP theorem in correlation with RDBMS and Hadoop, when looking at Atomic Consistent Isolated Durable (ACID) states how a RDBMS works. Since all components of a transaction are treated as a single action, transactions follow the rules of the database and durability falters if the connection with operations is not accommodated. Consistent Available Partition Tolerant (CAP) , on the other hand, make sure that the same data across the distributed system are the same regardless if it is, for example, an XML, JSON, or CSV file. And, more importantly, the system is designed to operate in the face of network connectivity loss (Hugg). There are clear distinctions between the two that correlate to RDBMS and Hadoop, respectively. ACID theorem works based off the design and structure of the database which is how a relational database is supposed to be, while Hadoop does not worry as much and is a framework similarly CAP is a tool that responds to queries despite their format and can handle unplanned network issues. Both have their use-cases as well as their advantages, knowing when to use one over the other is helpful in constructing a database.

Initially when thinking of using a SQL relational database versus Hadoop the user must think to themselves the complexity of the problem, whether huge amounts of distinct data is being processed or if it is a simple relationship. A relational database would be most advantageous when used in a fast-paced structured environment scenario. For instance, in Instagram has “x” number of users and “y” number of photos with various like and comments from different users then a clear pre-defined structure is built when building the application. On the other hand, in machine learning when getting data from a search engine, large sets of data are being taken for analysis in which case the search could vary but they are all categorized as “clusters” (Horwitz). The clear distinction between choosing the two is that relational databases are when multiple query calls are made for specific information while Hadoop is more advantageous in applications like machine learning where even more data is being worked with and processed. Ironically, more query calls are being made in a relational database environment which would warrant better durability, CAP. In my opinion, a relational database seems to be more beneficial since there are more basic use-cases for it. Similar to the Instagram example mentioned before, successful applications, Twitter, Facebook, Snapchat, etc. would benefit greatly from this style. Especially in an agile environment, if the application is getting constant updates then having a relational database where only a few attributes would need to be changed would be most beneficial to the developers. Having a relational database allows for better visualization of the physical, conceptual, and external levels of a database architecture. In Hadoop, this structure is not as defined which would cause more time and money spent in developing changes to the system. Hadoop, however, does have superiority in data handling which is as important as organization and is more cost efficient than having redundant data.

Even though Hadoop is increasing in popularity, RDBMS is a fundamental that can still serve a purpose. A small article talking about earnings show the different databases from industry leaders like Oracle, Microsoft, IBM, etc. Oracle lead the market in 2011 having 48.8% of the market compared to IBM’s DB2 database and Microsoft’s SQL Server databases (Fontecchio). This goes to show how successful having a RDBMS has been and how it fairs against other leading companies. In “Funding a Revolution: Government Support for Computing Research,” standardization is expressed as what keeps the US economy running (Funding a Revolution…). The most important fact of a relational database is that it sets a standard for others to follow so when using other applications or making transaction not much effort needs to be put into it since they are all similar. Computer applications are built on these types of databases and process the transactions and exchanges the U.S. economy uses, one of the leading powers in the world. Hadoop is not a “world” standard so making transactions and standardization is not as advanced a RDBMS model.

In conclusion, both Hadoop and a SQL Relational Database share their advantageous and disadvantageous. Hadoop is seemingly getting more and more popular especially with the emergence concepts like machine learning and higher usage of data. Hadoop is definitely a good alternative to a relational database, however, at this point not a complete replacement for it. A relational database is the standard that all database strive to be however, not everyone has the money for it and Hadoop’s quick framework assist in meeting some of those requirements. Until a more cost-efficient or “better” way to implement a relational database is found, RDBMS will be relevant and dominant in the field.

Works Cited

Fontecchio, Mark Fontecchio Profile: Mark. "Oracle the Clear Leader in $24 Billion RDBMS

Market." *Eye on Oracle*. N.p., 12 Apr. 2012. Web. 16 May 2017.

< http://itknowledgeexchange.techtarget.com/eye-on-oracle/oracle-the-clear-leader-in-24-

billion-rdbms-market/>

Horwitz, Joel. "KDnuggets." *KDnuggets Analytics Big Data Data Mining and Data Science*.

N.p., n.d. Web. 16 May 2017. < http://www.kdnuggets.com/2014/11/r-hadoop-make-

machine-learning-possible-everyone.html>

Hugg, John. "Disambiguating ACID and CAP." *VoltDB*. N.p., 27 Mar. 2017. Web. 16 May

2017. < https://www.voltdb.com/blog/2015/10/22/disambiguating-acid-cap/>

Le, ThienSi. "Hadoop vs. RDBMS." *LinkedIn*. N.p., 6 Dec. 2015. Web. 16 May 2017.

<https://www.linkedin.com/pulse/hadoop-vs-rdbms-thiensi-le>

"Read "Funding a Revolution: Government Support for Computing Research" at

NAP.edu." *National Academies Press: OpenBook*. N.p., n.d. Web. 16 May 2017.

<https://www.nap.edu/read/6323/chapter/8#159>.