Why industry is shifting form traditional RBDMS to Hadoop platform

Yuesen Dong

Third Year

May 2017 – August 2018

Work term Number 1

Table of Content

Executive summary

Big data is a very hot topic in recent time. Lots and lots of business starting to implement their datacenter or data storage facilities using the Hadoop framework other than a traditional RDBMS (Relational Database Management System). Throughout this report, you will learn what is Hadoop framework and all its advantage compared to traditional RDBMS including bigger data Volume, more Data Variety, easier scalability and cheaper cost. With the above advantages, businesses can utilize this framework to provide better data management practice at a lower cost.

Introduction

Everyone is talking about big data and Hadoop nowadays. But what it is really? In this report, I will state what Hadoop is and why it is the perfect tool for business to handle “Big data”. I will not discuss the technical detail of the implementation of Hadoop. Instead, I will focus on the architecture and the reasoning behind it.

Work term Overview

I was employed by Enterprise Architecture team from Canadian Imperial Bank of Commerce. Enterprise Architecture is responsible for the providing Architecture to all system and application that is operating within the bank. There are two types of architects within the bank. The domain architect will provide technical standards or guideline for the delivery group to follow in order to meet banks strategic goal. For example, the data architecture team will provide a guideline on how all the application handles data where to store and how to access them. The solution architect will provide a specific solution to a certain business need. For an example, they might want to open a new kind of bank product.

I was working with Data architecture team for the most of my work term. My role is Application Developer that helps architect research certain technology then build PoC projects and also figure out the best way to advertise the new technology to both bank stuffs and customers. I spend most of my time with the Hadoop framework during this term and that is the main topic of my report.

What is big data?

The term “big data” is fairly new. However, the act of gathering and storing huge amount of data for analysis purposes is ages old. This process starts to rise in the early 200s when industry analyst Doug Laney articulated the now-mainstream definition of the big data are volume, velocity, and variety is also known as the three Vs.

Volume

Volume is probably the first thing come to people's mind when we talk about big data. It is a really straightforward concept, the volume is big. But how big exactly?

Let's take Facebook for an example. We all know Facebook stores photographs. That shouldn't really concern any developer until you realize that Facebook has more user than the population of China. We can safely assume a large portion of that user has more than one post including a picture. As a matter of fact that Facebook is storing about 250 billion images. "So in terms of big data, when we are talking about volume, we're talking about insanely large amounts of data."(zdnet.com, 2018) Not in gigabytes but most of the time in terabytes or petabytes.

And the same logic applies to banks as well, as the number of clients starting to grow and more data we start to gather from our clients. Similar challenges rise extremely quickly.

Velocity

Another challenge rise with the volume is velocity. Let's go back to our Facebook for a bit. On average Facebook user upload more than 900 million photos on a daily basis. And I can guarantee you that no one has the patience to wait even ten minutes in the upload queue for a picture to be uploaded. And it is similar for banks as well, at peak time there are millions of transactions happening every minute. Nobody has the patience to wait in a Tim Horton for five minutes waiting for the payment to go through.

"Velocity is the measure of how fast the data is coming in." (zdnet.com, 2018) Lots of business need to handle a big amount of data in an extremely limited time.

Variety

Another really important characteristics of the "big data" is its variety. Let's go back to Facebook again, there are photographs, videos, geographic information and all another kind of stuff going into the Facebook server. This data isn't the old rows and columns and database joins in our forefathers. It is different from business to business, application to application, and much of it is unstructured.

Let's take a look in the banking world. Similar things are happening as well. A lot of data that are driving really important business decisions are sometimes videos or voice recordings. In the future, we might include Face ID or Touch ID for bank's authentication. That is another variety we need to incorporate within our data facility.

How we used to handle data

Since the first release of IBM DB2 in 1983. RBDMS is the most common way for a business to handle their data. A relational database management system is a database

management system based on the relational model invented by Edgar F. Codd at IBM. You can think of it as a place where we store a lot of Excel sheets. It has its own language to access those data, Structured Query Language also known as SQL. I am not going to explain what an RBDMS is as it can easily become its own report. I do recommend to take CSCC43/343 and CSCD43/443 if you want to learn more about RBDMS.

What is Hadoop

According to Wikipedia Hadoop is a "collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model. … All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework." (En.wikipedia.org, 2018)

Even though all my teacher warned me not to trust the material you found on Wikipedia but this is pretty accurate. As matter of fact, that is too accurate that is even challenging for some early stage Computer Science students to understand. Please allow me to put this into layman terms. Hadoop is basically a free software that unites multiple computers together to solve a very challenging problem that usually involving a lot of data and computation. It will support distributed storage and handle most of the hardware failures to a certain point. And those are achieved by three things, HDFS, Map Reduce, and YARN.

Hadoop Cluster

In order to better understand the Hadoop framework, let's take a look at how a Hadoop cluster is designed. Like I mentioned earlier that Hadoop united a lot of computers to complete a single but challenged task. Each computer or machine is considered as a node. Every node has its own disk or storage space and processing capabilities. The node can be a physical machine like a laptop, a virtual machine or a container. There are three types of nodes within a Hadoop cluster.

The Master node is the head of the Hadoop cluster, it saves data into HDFS and running parallel computations on that data using MapReduce. "The Name Node oversees and coordinates the data storage function (HDFS), while the Job Tracker oversees and coordinates the parallel processing of data using Map Reduce".(bradhedlund.com, 2018) Slave Nodes usually is the majority of the cluster who basically does all the dirty work. Each slave has a data Node and Task Tracker daemon who communicates with the master nodes that they report to. The Task Tracker daemon works for the Job Tracker and the data node works for the Name Node. The client node is to load data into the cluster or submit a Map Reduce job describing how data supposed to be processed and then show the result. It is more like a UI for the outside world.

Since a node could be a virtual machine or a container. It is common that for the smaller

cluster, a single physical server will act for all three roles. As a matter of fact, you could run a Hadoop cluster on your laptop if you have more than eight gigabytes of ram. I personally have a seven-node cluster running on my laptop when I was doing the PoC project. But when we are looking at the larger cluster, then usually each node it is own Linux server.

HDFS

HDFS stands for Hadoop Distributed File System. Data in a Hadoop cluster is broken down into smaller pieces called blocks that are distributed among Data Nodes throughout the cluster. "The goal of Hadoop is to use commonly available servers in a very large cluster, where each server has a set of inexpensive internal disk drives." (Ibm.com, 2018) MapReduce assign workloads to those servers where the data is stored and to be processed. And that is what we are referring as data locality. It will be saving a huge amount of bandwidth compared to transferring the data that needed to be processed over the network.

Another very important feature is the ability for HDFS to handle hardware failure. Like what we mentioned earlier that Hadoop is to use commonly available servers in a very large cluster. Those commonly available servers will not be reliable all time especially when it is not in the same geographic location. HDFS will break a large file into blocks, and copies of these blocks are stored on other servers in the Hadoop cluster for at least twice. In an unfortunate event that HDFS will recognize that one of its nodes is offline. It will find whatever information had been stored on that node and find its back up or original copy and back it up in a different node again.

MapReduce

"MapReduce™ is the heart of Apache™ Hadoop®. It is this programming paradigm that allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster. The MapReduce concept is fairly simple to understand for those who are familiar with clustered scale-out data processing solutions." (Ibm.com, 2018) So basically MapReduce does two things, Map and Reduce. Mapping is simply dividingthe task to each server or node to perform on their local data. And reduce is simply using appropriate algorithms to combine all the result we have from each node to combine it into one final result.

Let's see an example, think of a .txt file that contains everyone's personal information in this country and email address is one of them. That file will be broken into blocks and store on multiple servers. In order "to achieve availability as components fail, HDFS replicates these smaller pieces onto two additional servers by default. (This redundancy can be increased or decreased on a per-file basis or for a whole environment; for example, a development Hadoop cluster typically doesn't need any data redundancy.) This redundancy offers multiple benefits, the most obvious being higher availability." (Ibm.com, 2018) In this case let's say this file is been broken into five parts for the sake of simplicity.

Someone wants to know how many people are using Gmail to determine the size of the

email server that Google should have in Canada. So this task will be broken down into 5 smaller task. Each server will report how many emails that is @gmail.com in their local data. Then the reduce will simply add them up to give the final result for the entire data set.

YARN

As we mentioned earlier that the Hadoop cluster is made up of multiple nodes or servers to form one giant cluster. It is extremely important to properly manage all the resources within the network. Here we come the YARN which stands for “Yet Another Resource Negotiate”. It is a layer that sits between the resource management layer and the processing components layer. It solves the Scalability bottleneck caused by having a single JobTracker. I will only briefly mention it during this report as it can easily be it’s own report as well. If you are interested you can read the article on the IBM website

Hadoop vs RDBMS

In this section, I will be comparing the differences and advantages of Hadoop and RDBMS. There are three major differences comparing Hadoop and RDBMS, centralization, data variety, and scalability.

Centralization

As we mentioned earlier that RBDMS is a centralized system. Which means that all hardware must be in the same physical location. It does provide some benefits such as easier maintenance and such. However, since all data and the hardware that stores the data are in one physical location, it also increases the risk for lost and damages. If there are natural disaster occurs or anything unfortunate happen that all data will be lost. Meanwhile, the distributive nature of Hadoop can resolve the problem easily as their hardware could be potentially in different geographic locations as we mentioned above.

Data Variety

As the name relational database management system suggested, it only handles structured data. Any data that is not really a primary data type like numbers or strings will not be stored into the database properly. We often rely on external urls for such data. Moreover, that SQL doesn’t have any power process unstructured data. When we need to make some data processing we often relies on another programing language with database connectors for example JDBC.

Scalability

Both data storage solution can scale up to meet performance requirement with just a few differences. RBDMS provides vertical scalability which is basically upgrading the server. You can put more ram and a better CPU on the machine to achieve better performance. Hadoop not only offer vertical scalability, you can always to put more hardware on each individual notes to make it more powerful. It also offers horizontal scalabilities which means add more server to the cluster will also increase its overall performance. By comparing those two we can easily see that Hadoop have a huge advantage in this category as there is only so much you can do vertically.

Let's take our computer for example. Of course, you can put more ram into your machine as long you have additional ram slots. However, home grade motherboard usually only supports up to 4 channels of ram. If you want to put more ram on it, you need a bigger motherboard which not might fit into your existing computer case which in that case you need a new case. And that case might not fit in your house then you have to buy a new house and etc. However, you can just simply buy a new machine and add to the cluster for Hadoop.

What does it mean for businesses (conclusion and recommendations)?

Since I was working within a bank. I will use the bank as an example.

Before the adoption of Hadoop. Bank used to have more than one hundred fifty different data environment for a different line of businesses. Due to the limited performance of the RBDMS, it is just not likely to put all data into one system as it is just too big. That creates significant problem is that when the management trying to make huge business decisions they usually don't know where to find the data they need. Knowledge is preserved in each individual teams but not a centralized environment.

Also, we lost a lot of business intelligence because of that. For example, if we saw there is a large amount of money transferred to one's chequing account witch is an account with lower interest. There is a high chance that this person is about to punches something valuable like a car or house that might result in a mortgage or car loan. Without a centralized environment, the mortared team won't have a clue of this potential client and cloud have missed this opportunity.

Another thing is that the bank really values data security. Hadoop distributive characteristics will allow the bank to easily link multiple data center together and reduce the possibility of total data lost. Also, the horizontal scalability is really important when a bank trying to expand their data center.

Overall that is why a lot of businesses are moving towards Hadoop or another big data platform instead of traditional RBDMS because of its numerous advantages.

Reference

Hedlund, B. (2011, September 10). Understanding Hadoop Clusters and the Network. Retrieved August 16, 2018, from http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters-and-the-network/

W. (2017, November 24). Difference between Big Data Hadoop and Traditional RDBMS. Retrieved from https://www.w3trainingschool.com/difference-big-data-hadoop-traditional-rdbms

Gewirtz, D. (2018, March 21). Volume, velocity, and variety: Understanding the three V's of big data. Retrieved August 16, 2018, from https://www.zdnet.com/article/volume-velocity-and-variety-understanding-the-three-vs-of-big-data/

IBM. (n.d.). What is MapReduce? Retrieved from https://www.ibm.com/analytics/hadoop/mapreduce

Kawa, A. (2014, August 12). Introduction to YARN. Retrieved August 16, 2018, from https://www.ibm.com/developerworks/library/bd-yarn-intro/

Lam, C. (2015, February 10). Comparing SQL databases and Hadoop. Retrieved August 16, 2018, from http://bigdata-madesimple.com/comparing-sql-databases-and-hadoop/

McKissick, K. (2015, November 25). Just how does Facebook store billions of photos? Retrieved August 16, 2018, from https://news.usc.edu/88075/how-does-facebook-store-billions-of-photos/

Miller, R., MillerI, R., A., G., D., Dutch, S., . . . S. (2017, March 21). Inside Facebook's Blu-Ray Cold Storage Data Center. Retrieved August 16, 2018, from https://datacenterfrontier.com/inside-facebooks-blu-ray-cold-storage-data-center/

Shields, A. (2014, July 25). Overview: What is "big data"? Retrieved August 16, 2018, from https://marketrealist.com/2014/07/overview-big-data

Shukla, A. (2014, July 28). Why traditional database systems fail to support "big data". Retrieved August 16, 2018, from https://finance.yahoo.com/news/why-traditional-database-systems-fail-210014958.html

Sinha, S. (2017, October 25). What is the difference between Hadoop and HDFS? Retrieved August 16, 2018, from https://www.quora.com/What-is-the-difference-between-Hadoop-and-HDFS

Verma, A. (2018, February 15). Comparing SQL Databases and Hadoop. Retrieved August

16, 2018, from https://www.whizlabs.com/blog/hadoop-vs-sql-database/
What is HDFS? (n.d.). Retrieved from https://www.ibm.com/analytics/hadoop/hdfs