Deploying Big Data Analytics to the Mobile Cloud Environment

Ngu Wah Win
University of Computer Studies, Yangon
nguwahwin@ucsy.edu.mm

Abstract

Big data analytics are the significant solution of business decisions of many organizations by examining large data sets. On the other hand, mobile devices are playing an increasingly important role in the future of business. Because of this stream of change of technology requirements, big data analytics have shifted from personal desktop computer to mobile devices. For this reason, mobile cloud computing was emerged and it is an infrastructure to solve the limitations of mobile devices, where both the data storage and the data processing happen outside of the mobile devices. The aim of this research is to develop a big data analytic platform for mobile devices with seamless connectivity, efficient query processing time, and data visualization method.

Keywords: big data analytics, mobile cloud computing, Hive, data visualization

1. Introduction

As data volumes continue to increase exponentially, the data tsunami can easily overwhelm traditional analytics tools or platforms designed to ingest, analyze and report. Explosion in big data and infrastructure demands of big data analytics have led to a rise in big data centers. Big Data focuses on achieving deep business value from deployment of advanced analytics and trustworthy data at Internet scales. Big Data is at the heart of many cloud services deployments. As private and public cloud deployments become more prevalent, it will be critical for end-user organizations to have a clear understanding of Big Data application requirements, tool capabilities, and best practices for implementation.

Cloud computing is a method of providing a set of shared computing resources that include applications, computing, storage, networking, development, and deployment platforms, as well as business processes. Cloud computing turns traditional computing assets into shared pools of resources based on an underlying Internet foundation. In cloud computing, everything, from compute power to computing infrastructure and from applications and business processes to data and analytics, can be delivered as a service.

Nowadays, complex smart phone applications are developed that support gaming, navigation, video editing, augmented reality, and speech recognition which require considerable computational power and battery lifetime. The cloud computing provides a brand new opportunity for the development of mobile applications. Mobile Hosts (MHs) are provided with data storage and processing services on a cloud computing platform rather than on the MHs. Mobile cloud computing is a new paradigm that aims at using cloud computing techniques for storage and processing of data on mobile devices, thereby reducing their limitations.

The main goal of this platform is to reduce the workload of mobile devices because of their resource limitations. Therefore, all of the work and data are moved to the cloud server backend except data representation. The request from the mobile user is sent to the cloud by using web service's URL and it invokes the Mapper and Reducers of HDFS. After processing the MapReduce steps, the results is received by web service layer and transformed the result into JSON format. After that, it return back this result to mobile device and which represent the results into bar chart, pie chart and line chart.

2. Theory Background

The rise of cloud computing and cloud data stores has been a precursor and facilitator to the emergence of big data. Cloud computing is the commodification of computing time and data storage by means of standardized technologies. Three main cloud architecture models have developed over time; private, public and hybrid cloud. They all share the idea of resource commodification and to that end usually virtualize computing and abstract storage layers.

The categorization of three cloud service models defined in the guideline is also widely accepted nowadays. The three service models are namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization. Companies utilizing advanced analytics platforms to gain real value from big data will grow faster than their competitors and seize new opportunities. Traditional IT infrastructure is simply not able to meet the demands of this new "Big Analytics" landscape. For these reasons, many enterprises are turning to the Hadoop (open source project) as a potential solution to this unmet commercial need [8]. There are five characteristics for big data, also known as 5Vs: volume, velocity, variety, veracity, and value [6, 7].

The rapid development of mobile computing and cloud computing trigger novel computing paradigm called Mobile Cloud Computing [4]. Nowadays, the market of mobile phone has expanded rapidly. However, mobile devices still lack in resources compared to a conventional information processing device such as PCs and laptops.

Also, the limitation of battery restricts working time. How to augment capability of mobile phone has become the important technical issue for mobile computing. The paradigm of cloud computing brings opportunities for this demand. CloneCloud, Cloudlet, AlfredO, and Hyrax are the most significant research by using mobile cloud paradigm.

With the advent of the MapReduce paradigm from Google, Hadoop is the most popular open source framework for big data analytics platforms. At its core, it has two components: The Hadoop distributed file system (HDFS), which is a low-cost, high-bandwidth data storage cluster. The MapReduce engine, which is a high performance, distributed/parallel processing implementation. It helps to break data into manageable chunks and then makes it available for either consumption or additional processing [4]. There are many big data analytics platform with Hadoop in today's business world, such as Microsoft Azure HDInsight, HP Vertica, IBM BigInsights, Intel Data Platform, and Actian.

Analyzing big data set at terabyte or even petabyte scale is called big data processing. Big data processing can be categorized as batch-based, stream-based, DAG-based, interactive-based, and visual-based according to the processing techniques.

3. Objectives and Problem Statement

The main objectives of this thesis is to develop a big data analytics platform for mobile devices to access big data anytime and anywhere. The second objective is to provide seamless connectivity between mobile client and cloud server. To measure and compare the performance analysis between MapReduce based high level query languages, Apache Pig, Apache Hive and, JAQL, is the third objective of this thesis. The basic requirement of big data analytics platform is to improve the processing time of the system. In this proposed platform, the performance of the processing time is rely on client side (mobile device) and server side (cloud storage). So, the next objective of this thesis is to improve the performance of processing time on both sides. The last objective of the thesis is to present the analytical results that produced by cloud storage with user friendly view on mobile clients.

There are many challenges when developing a big data analytics platform for mobile devices. The first one is to tackle the resource limitations, storage and processing capacity, of mobile devices. And the second one is to provide the seamless connectivity between mobile devices and cloud server. Finally, the platform needs to improve the performance of data processing and to response the user request as simple as possible.

4. Performance Analysis of HLQLs in Traditional Big Data Platform

The MapReduce model proposed by Google has become a key data processing model, with a number of realizations including the open source Hadoop implementation. A number of HLQLs (High Level Query Languages) have been constructed on top of Hadoop to provide more abstract query facilities than using the low-level Hadoop Java based API directly. Pig, Hive, and JAQL are all important HLQLs [9].

4.1 Apache Pig

Programs written in Pig Latin are firstly parsed for syntactic and instance checking. The output from this parser is a logical plan, arranged in a directed acyclic graph, allowing logical optimizations, such as projection pushdown to be carried out. The plan is compiled by a MR compiler, which is then optimized once more by a MR optimizer performing tasks such as early partial aggregation, using the MR combiner function. The MR program is then submitted to the Hadoop job manager for execution.

4.2 Apache Hive

Hive [10, 11] QL provides a familiar entry point for data analysts, minimizing the pain to migrate to the Hadoop infrastructure for distributed data storage and parallel query processing. Hive supports queries expressed in a SQL-like declarative language - HiveQL, which are compiled into MR jobs, much like the other Hadoop HLQLs.

4.3 JAQL

JAQL is a functional data query language, which is built upon JavaScript Object Notation Language (JSON) [5]. JAQL is a general purpose data-flow language that manipulates semi-structured information in the form of abstract JSON values.

4.4 Architecture of Traditional Big Data Analytics Platform

Figure 1 shows the architecture of traditional big data analytics platform. It composed of three layers, and all of these layers are worked on cloud backend.

Storage layer consists of NameNode, Secondary NameNode, and DataNode which are worked as a cloud storage cluster. The data from the storage layer is processed by JobTracker and TaskTracker which perform as a processing layer of the platform. In application layer, there are three types of high level MapReduce based query languages are applied and these query languages communicate with the storage layer by passing through the processing layer. Finally, the analytical results are return back to the client.

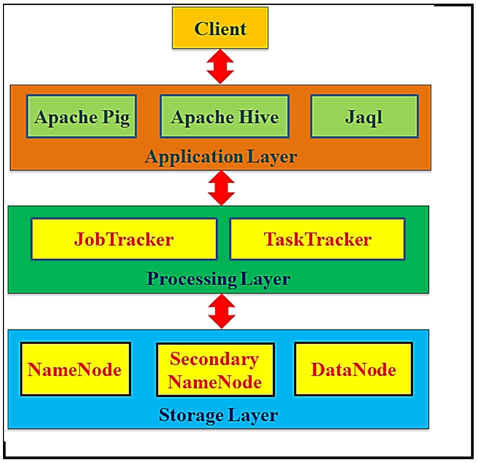

Figure 1. Architecture of traditional big data analytics platform

In this platform, the client is not a mobile client and if mobile user wants to access the data on HDFS there is needed a layer that connects the client and storage.

4.5 Experiment Results for Performance Analysis of HLQLs

The evaluation of traditional big data analytics platform is done on performance analysis of different query languages. US census dataset [1] is used to evaluate the performance of two big data analytics platforms. The dataset consists of 331 tables. Population table is used to evaluate the query performance of two big data platforms. It consists of 12905514 records for 52 states (50 US states, the District of Columbia, and Puerto Rico).

The performance of two big data platforms, Ubuntu OS and Red Hat OS, has been evaluated on commodity Linux cluster. The VMs are interconnected via a 1-Gigabit Ethernet. The host machine runs Windows 8 and has Intel ® Core™ i7-4700MQ CPU @ 2.40GHz processor, 8GB physical memory, 1000GB hard disk. As software components, Hadoop 1.1.2, Pig 0.10.0, Hive 0.9.0 and JAQL 0.5.1 are used. Eight virtual machines which are connected on Ethernet is used as a Hadoop cluster.

Four different queries are used as sample analytical workloads for performance evaluation of HLQLs on two big data analytics platform with different record size.

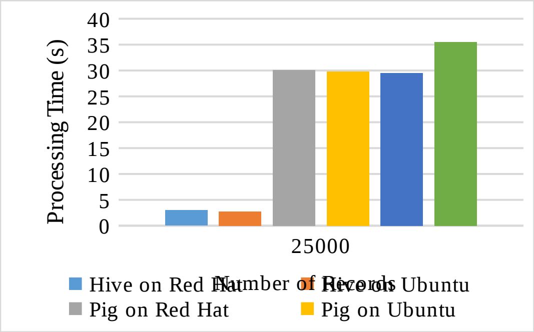

Figure 2. Query 1's processing time on Ubuntu and Red Hat

The first query is selects only those records whose population is greater than 30000, and displays the result. The second query is to find the number of records in population table. The third query is to find the total population for each state. The fourth query is to find the number of record for each state.

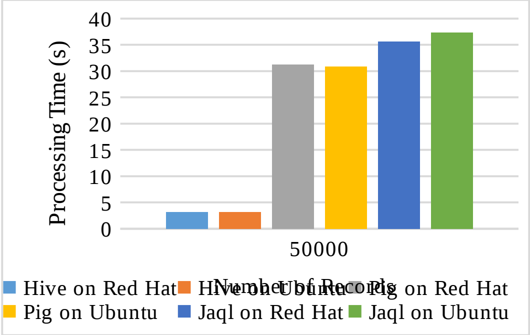

Figure 3. Query 2's processing time on Ubuntu and Red Hat

Figure 2 and 3 show the processing time of HLQLs on Ubuntu and Red Hat OS for first and second query. The testing workload for Figure 2 is 250000 records and 500000 records for Figure 3.

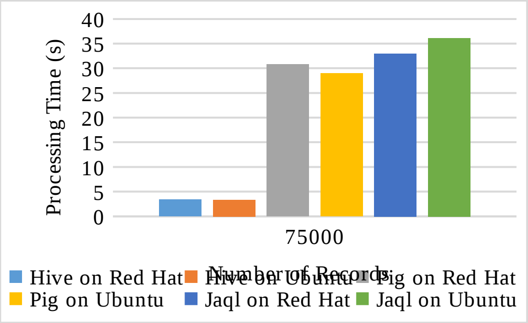

Figure 4. Query 3's processing time on Ubuntu and Red Hat

Figure 4 and 5 show the processing time of HLQLs on Ubuntu and Red Hat OS for third and fourth query. The testing workload for Figure 4 is 750000 records and 1000000 records for Figure 5.

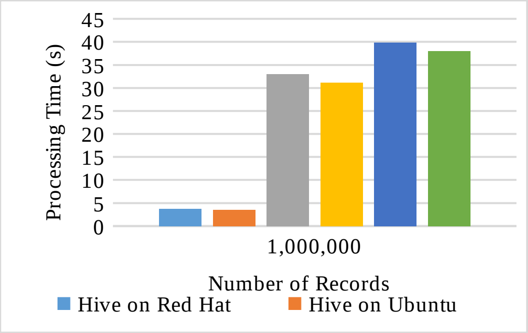

Figure 5. Query 4's processing time on Ubuntu and Red Hat

According to the all experiment results, it can conclude that the Hive can give better performance on both platform, i.e. Ubuntu OS and Red Hat OS. For developing a big data analytics platform for mobile device, Hive and Ubuntu OS is chosen to improve the processing performance of the platform.

5. Hadoop-based Proposed Big Data Analytics Platform

In this section, big data analytics platform on mobile cloud computing with Hadoop MapReduce framework and HDFS over distributed storage system is proposed. To improve the performance of this platform, BDA_PTraM (Big Data Analytics with Predefined Transformation Model) model is proposed and which implemented the predefined operations based on different operators. BDA_PTraM reduces the overall processing time of the simple query and the proposed platform used the HiveQL for complex queries type. The experiments results of the proposed platform are tested with four sample datasets.

Figure 6. Proposed big data analytics platform

The proposed big data analytics platform, shown in Figure 6, consists of four layers. The lowest layer is storage layer, the second lowest layer is processing layer. After that, application layer and web service layer accordingly. All of these layers are work on Hadoop Distributed File System with MapReduce programming model.

The main component of the application layer is MapReduce. The traditional big data analytics platform use high level query languages to analyze the data. According to the experiment results from the previous section, HiveQL is most suitable for proposed platform. In this layer, this platform proposes a BDA_PTraM Model to improve the overall processing time of the platform for simple queries.

In this proposed Model, Driver class acts as an invoker program for Map and Reduce functions. All of this functions are implemented with Java programming model and communicate with HDFS cluster. The work of Driver function is to assign the job on. JobTracker and set the key value type for Map and Reduce outputs. And then, it invokes Mappers by specifying input directories from HDFS location. To receive the better processing performance on proposed platform, the Mapper and Reducer have to reduce the workload by ignoring unwanted value from input data. So, Mapper filters the column and only send the required column to the Reducer. After receiving the output from Mappers, the Reducer starts the operations according to user request.

Algorithm 1: Mapper(LongWritable, Text, Text, IntWritable)

```
Input: i_file = inputFile
Output: output = {<$k_1$,$v_1$>, <$k_2$,$v_2$>, …, <$k_n$,$v_n$>}
int missing  ↼ 9999;
String param ↼ null;
String[] operator_number;
Configure(JobConf job)
param ↼ job.get("inputvalue")
operator_number ↼ param.split(":")
Map(LongWritable key, Text value,
        OutputCollector<Text,Text>output,
        Reporter reporter)
String[] operator_number;
String sline ↼ value
String cvsSplitBy ↼ ","
String [] column ↼ sline.split(cvsSplitby)
output ↼ <column[operator_number[0]-1,
          column[operator_number[1]-1>2
```

Figure 7. Mapper algorithm for BDA_PTraM model

In Figure 7, Mapper filters the population value column and pass this column to the reducer as input data. The reducer then work upon this column and produce the final output.

Figure 8 shows the workflow of Reducer class and which produces the final key-value output to the temporary folder. Web service layer takes this text output to convert the JSON file format and send it to the mobile user.

Algorithm 2: Reducer(LongWritable, Text, Text, IntWritable)

```
Input: MapoutputCollection<key,value>
Output: <key,value>
int comp_value;
String param ↼ null;
String[] operator_number;
String temp_output ↼ path_of_temp_output_folder
Configure(JobConf job)
param ↼ job.get("inputvalue")
operator_number ↼ param.split(":")
Reduce (Text key, Iterator<Text> values,
Outputcollector<Text,Text> output, Reporter
        reporter)
foreach value ∈ values do
        String split_value ↼ values.next()
        String[] parse_value ↼ split_value.split(",")
        int status ↼ parse_value[0]
        int value ↼ parse_value[1]
        if (operator_number[2] = operator)
            if (value satisfy with comp_value)
                comp_value ↼ value
            end
        end
end
PrintWriter out ↼ new_PrintWriter(new_
        BufferedWriter (new_FileWriter
                                   (temp_output)
output ← <key, new_Text(comp_value)>
```

Figure 8. Reducer algorithm for BDA_PTraM model

5.1 JSON

JSON is a simple text-based message format that is often used with RESTful Web services. Like XML, it is designed to be readable, and this can help when debugging and testing. JSON is derived from JavaScript, and therefore is very popular as a data format in Web applications [2, 12]. However, JSON can be read and written by many programming languages.

5.2 Evaluation of Proposed Platform

The experiment comparison is done over four operations such as finding Maximum, Minimum, Number of records, and Sum with eight operators (No operator, equal, not equal, greater than, less than, greater than or equal, less than or equal, and between).

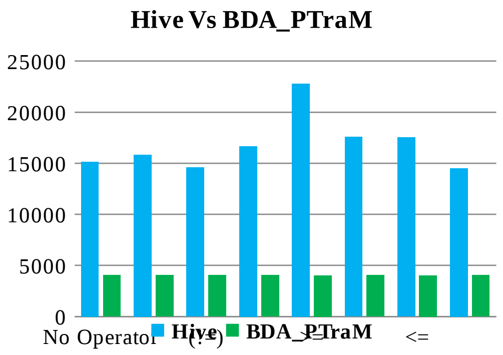

Figure 9. Comparison of the processing time of maximum operation for Hive and BDA_PTraM model

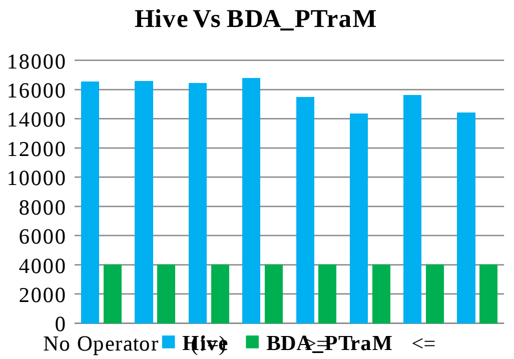

Figure 10. Comparison of the processing time of Minimum operation for Hive and BDA_PTraM model

Figure 9 and 10 illustrate the comparison of processing time for Maximum and Minimum operations between BDA_PTraM and Hive.

Figure 11 and 12 present the comparison of processing time for Maximum and Minimum operations between BDA_PTraM and Hive.

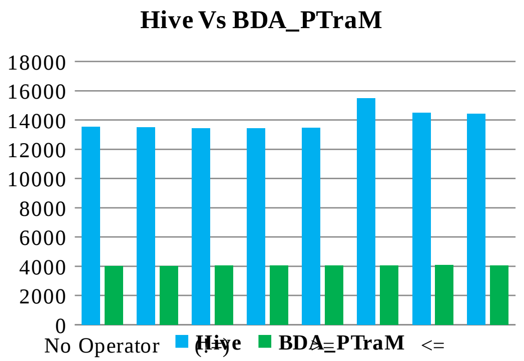

Figure 11. Comparison of the processing time of Count operation for Hive and BDA_PTraM model

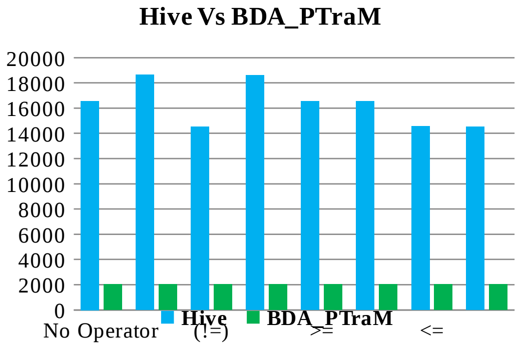

Figure 12. Comparison of the processing time of Sum operation for Hive and BDA_PTraM model

This section presents the detail architectures and workflows of four layers of proposed platform: storage layer, processing layer, application layer, and web service layer. After that, describes about the Datasets and experiment environments of Hadoop clusters. Finally, compares the two methods, BDA_PTraM and Hive, with four operations with eight operators. According to this experiment results, we can conclude that the big data analytics platform with BDA_PTraM model is three time faster than Hive in simple query. For complex query, the platform used Hive to achieve the better performance.

6. Data Representation on Mobile

The seven steps of data visualization are [3] – acquire, parse, filter, mine, represent, refine, and interact. In these steps, the first four steps are operated on the cloud backend and the mobile device needs to perform the representation step only. In this representation steps, data are passed through the web service layer of cloud storage with JSON format and mobile extract the data to present the users.

A sample test case for Count operations of 10 states are used to evaluate the data representation on mobile device with bar chart, pie chart and line chart. In test cases user click the predefined request on Android' UI and request is operated by cloud server.

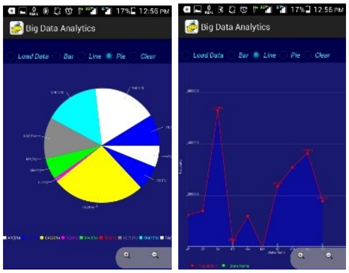

Figure 14. Data representation on mobile device

Figure 14 shows the data representation on mobile device with bar chart, pie chart, and line chart respectively. According to this figures, users can easily make the decision on analytical results because of the data representation.

This section presents the seven steps of data visualization on mobile device with Android OS. This platform covers within five steps to represent the data. The data extraction program on mobile device used the Android Chart Engine Libraries to plot the chart and used the JSON input file for input data.

7. Conclusion

This paper has proposed a big data analytic platform for mobile device on distributed scale-out storage system. The software framework that used for big data analytics is Hadoop 1.1.2 version and it works as master/slave architecture.

For complex query, this platform used the HiveQL and user can send their request via mobile device. The analytic process is done on cloud backend server and the results are returned back to the mobile client with web view interface. The main purpose of this proposed platform is to provide the big data analytics facilities on mobile device for any type of query. In this platform, simple query is worked by BDA_PTraM model and the ad-hoc query used Hive model.

References

[1] Census Dataset, “U.S Census Dataset”, [Online] Available http://www2.census.gov/census_2010/, [Accessed 31/8/2014].

[2] D. Crockford, “The Application/JSON Media Type for Javascript Object Notation (JSON)”, [Online] Available: https://tools.ietf.org/html/rfc4627 , July 2006, [Accessed 12/10/2014].

[3] B. Fry, "The Seven Steps of Visualizing Data", [Online] Available: www.safaribooksonline.com/ [Accessed: 25/6/2015].

[4] F. Halper, “Eight Considerations for Utilizing Big Data Analytics with Hadoop”, TDWI Checklist Report, March 2014.

[5] Jaql, “Jaql, Query Language for JavaScript Object Notation (JSON)”, [Online] Available https://code.google.com/, [Accessed 13/12/2015].

[6] B. Marr, “Big Data: The 5 Vs Everyone Must Know”, 6 March, 2014, [Online] Available https://www.linkedin.com/ [Accessed 24/6/2014].

[7] OReilly Media, “Volume and Velocity, Variety: What You Need to Know About Big Data”, 19 January, 2012, [Online] Available www.forbes.com/, [Accessed 16/6/2015].

[8] Revolution, “Advanced ‘Big Data’ Analytics with R and Hadoop”, [Online] Available www.revolutionanalytics.com/, [Accessed 16/7/2015].

[9] R. J. Stewart, P. W. Trinder, and H. Loidl, “Comparing High Level MapReduce Query Languages”, In Proceedings of the “9th International Conference on Advanced Parallel Processing Technologies, APPT 2011”, 26-27 September, 2011, Shanghai, China, pp. 58-72, ISBN 978-3-642-24150-5.

[10] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Anthony, H. Liu, and R. Murthy, “Hive-A Petabyte Scale Data Warehouse using Hadoop”, In Proceedings of the “26th IEEE International Conference on Data Engineering”, 1-6 March, 2010, Long Beach, California, USA, pages 996-1005, ISBN 978-1-4244-5444-0.

[11] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, “Hive: A Warehousing Solution Over a MapReduce Framework”, In Proceedings of the “VLDB Endowment”, Volume 2, Issue 2, August 2009, pp. 1626-1629, doi: 10.14778/1687553.1687609.

[12] J. Wyse, “Why JSON is better than XML”, [Online] Available http://blog.cloud-elements.com/, [Accessed 5/7/2015]