Implementation of Hadoop and Sqoop for Big Data Organization

Rohan Sidhu Deanna Chea Rachita Dhakal
Annie Hur Mark Zhang

Nomenclature

HDFS	**H**adoop **D**istributed **F**ile **S**ystem
RDBMS	**R**elational **D**ata**b**ase **M**anagement **S**ystem
SQL	**S**tructured **Q**uery **L**anguage
MSCH	**M**y**S**QL **C**onvert to **H**Base
CLI	**C**ommand **L**ine **I**nterface
RDD	**R**esilient **D**istributed **D**atasets
SSH	**S**ecure **Sh**ell

Abstract

With the popularity of consumer web services, such as Amazon and Yahoo!, data is stored in millions of terabytes and gigabytes. In order for computer scientists to efficiently analyze and manipulate this memory, Hadoop, a distributing processor, emerged less than a decade ago and is frequently used within the consumer industry. There are two fundamental features to Hadoop, HDFS (Hadoop Distributed File System) acting as a storage unit and MapReduce playing the role of the processor. HDFS is not limited to serving MapReduce, but other programs including Sqoop, Mahout, Hive, and Pig. Despite the various softwares available, we chose to examine Sqoop and its impact.

In order to gain a deeper understanding of the internal workings of Hadoop and Sqoop, academic literature research was conducted, as well as experiments using the Hadoop MapReduce framework. Various Hadoop technologies were also investigated, including Apache Pig, Hive, and Spark. Our study was limited to a smaller dataset to copy to the HDFS in order to viably test the program in the allotted timeframe. This report ultimately encompasses the technicalities of Hadoop and its myriad tools, focusing on Sqoop.

Keywords -

Hadoop, Sqoop, Big Data, HDFS

Introduction

In a society that is continuously advancing technologically, an increased demand for mass storage, analysis, and manipulation of massive datasets has emerged. As consumers of services are commonly found online, data is stored in large quantities to be utilized by industries and global businesses. User data is used to improve customer satisfaction and ensure future business transactions while these industries are simultaneously attempting to capitalize on profits to meet demands. In order to do so, they have shown to be amenable to new programs designed to meet these goals. In parallel with the increasing globalization of online services and users, software and frameworks have been developed in order to work more efficiently with large datasets, such as the modern Apache Hadoop and its array of tools.

This article concentrates on the specific Hadoop tool, Sqoop, and its optimum uses and applications. Following the initial 2012 release of Sqoop 1.4.0, its development has been catalyzed by the rising demand from businesses concerned with the online interactions with users [1]. Acting as a bridge between rivers of information, Sqoop migrates data to and from databases with commands that can be executed from a CLI (Command Line Interface). It can be used to import and export data from RDBMS (Relational Database Management System) to Hadoop while having few limitations. With the significant improvement of the tool, large businesses such as Yahoo! and Amazon have begun utilizing the technology in order to manage Big Data from customers [2].

While there are a vast number of technologies used with Hadoop's MapReduce framework, this investigation of Sqoop will analyze its many components, from its nascent creation to the wide implementation of the software to store massive data and transfer it to and from the RDBMS and Hadoop.

Literature Review

I. HANDLING BIG DATA WITH HADOOP

Devacunchari's research [3] suggests the widespread usage of the Hadoop MapReduce framework by businesses and others needing mediums to precisely organize sets of data. Hadoop MapReduce is a software framework which allows users to easily script applications which process massive datasets in parallel [4]. It utilizes two main functions: map and reduce. The map functions applies functions to datasets, consequently producing further lists containing its output results. Reduce accepts big data and outputs a single value. Functions used conjoined with the reduce function must have the ability to combine two pieces of data into one, then insert another piece of data, combining, and repeating until there is only one value (See Figure 1).

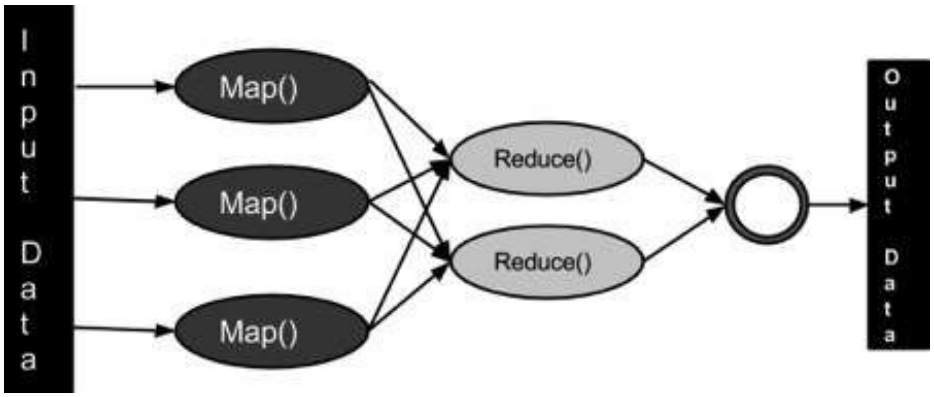


Figure 1. MapReduce flow chart [3].

This program executes in three stages-- map, reduce, and shuffle--in order to organize data. During the map stage, the inputted data is processed, typically big data, then enters the program one line at a time. The mapper then creates smaller chunks of data. The shuffle and reduce stages process these pieces of data and render a list of output, stored in the HDFS. These tasks are lastly sent to corresponding servers.

MapReduce can be utilized to process in parallel on a variety of servers in order to smoothly function, thus enabling it to work efficiently with very large datasets. Data can be mapped and reduced over multiple servers, allowing MapReduce to process scalable code [4]. It is a simple, mostly user-friendly way of downsizing large datasets into smaller pieces, enabling users to then analyze, interpret, and work with the smaller chunks or combine them in new ways.

This programming model operates on <key,value> pairs to process inputs and renders outputs, both stored in a file system. Map functions can process the pairs in order to create intermediate pairs and the reduce function combines corresponding values and keys [4]. Programs are able to process the data, simultaneously stratifying data into smaller, workable chunks. Map tasks process these smaller chunks in parallel, and the framework is able to sort the outputs of the maps, then inputting them to the reduce tasks. Hadoop MapReduce has the ability to organize scheduling tasks, analyze their success, and re-execute any tasks that were not successful.

The optimal times to use MapReduce include tasks such as scheduling or those which require scaling data over multiple servers. It efficiently manages details of data-passing while reducing the data to form results that are sent to Hadoop. MapReduce is not a suitable choice for quick-responding, real time processing, or intermediate processes which need to communicate with each other. It cannot join two large, complex datasets, which are better handled by other programs, it's not savvy at processing graphs or some more complex algorithms, and it is limited due to its close association with Java. It is best used to batch process big data readily-available to the user. To smooth the use of MapReduce, Hadoop has several tools that may be implemented.

II. BIG DATA TECHNOLOGIES

To optimize the Hadoop experience, technologies such as Apache Mahout, Pig, Hive, and Sqoop have been developed in recent decades [5]. These tools are targeted for specific purposes and may work conjoined with one another to ease the workings of Hadoop. These technologies have the ability to aid distributed data processing, commonly needed in the tech-savvy world.

Krishnan [6] contends to the primary purposes of establishing distributed data processing. This purpose is to copy the database management system in a master slave configuration and process the data across the multiple instances. As Hadoop is a platform for storing big data, it does not allow the transfer of other smaller and unstructured data that could play vital role in the process of analyzing the information for necessary purposes [7]. Consequently, in order to transfer the data between the Hadoop system and RDBMS, Sqoop has been developed as a junction which, stated by Krishnan, was one of the primary design goals of Sqoop when it was first introduced [6]. While Sqoop is a significant tool regarding the management of Big Data, many other tools have been developed which are necessary for the smooth functionality of Hadoop.

a. Mahout

Mahout is a machine learning software which carries-out three techniques in order to analyze and sort data. By using recommendations, classification, and clustering, input can be organized in a manner that will benefit the user. Recommendation utilizes the user's already-stored information from the past combined with public information in order to determine the likelihood of the user liking a different piece of data. One example of this technique is Netflix, which uses the viewer's ratings of shows and movies to suggest new items the viewer might also like. Music stations, such as Spotify, also take advantage of this same system when suggesting new songs based on the listener's preferences.

Mahout's second technique, classification, uses known data to determine how new data will be categorized. It uses already-existing groups in which the data is sorted into. Email services use classification as the user labels emails as spam and the software consequently labels other email of the same type as spam. Lastly, Mahout implements clustering as a way to sort data into groups not previously created by the user [8]**.** News websites use this to suggest similar articles to the one the user just read by analyzing similarities between the myriad stories. It is especially helpful for users who do not know how to sort their data in the first place. This Hadoop software allows applications to analyze and sort large sets of data. Now it is commonly used to solve business problems and create an efficient environment suitable for varied browsing. With its implementation, it can separate large amounts of data into smaller, parallel tasks which are then categorized.

b. Pig

When initially developed, Pig was used to analyze large sets of data in an efficient and timely manner. It was designed to work well with virtually any type of data, paralleling the infamous farm animal which eats almost any type of food. Pig is composed of two components: Pig Latin, the language used, and the runtime environment where Pig Latin programs are executed [9]. Pig Latin is a programming language which loads the data from HDFS, runs the data through transformations, and lastly dumps or stores data. Hadoop objects are stored in HDFS, where a program then directs Pig to access it through the LOAD 'data-file' command. When the data is undergoing a transformation, it can filter unnecessary rows, join data files, group data, or order results, in addition to a wide variety of functions. Finally, dump or store commands generate the Pig program's output by either sending the output to the screen to debug or storing results in a file for further use. By taking advantage of Pig, programmers can create custom functions to serve a vast number of tasks. With the aid of this platform, researchers are now able to analyze the big datum that utilizes high-level languages more efficiently than writing MapReduce programs.

c. Hive

Hive is a software with the ability to manage big data in a distributed storage (See Figure 2) [10]. It allows the user to query and read these large datasets by using a language known as HiveQL. Hive may also utilize MapReduce programs to use custom mappers and reducers in order to easily direct the software without HiveQL. It provides different storage types, reduced time for semantic checks, and already-included functions. Other components of the software include indexing to provide acceleration, metadata storage, and the ability to operate on compressed data stored in Hadoop [10]. Hive is similar to traditional code with SQL (Structured Query Language), however it is best-suited to be used for long, sequential scans, and it is not recommended for applications requiring quick responses. A key convenience of Hive is its link with SQL, allowing those fluent in the language to adapt to Hive with ease.

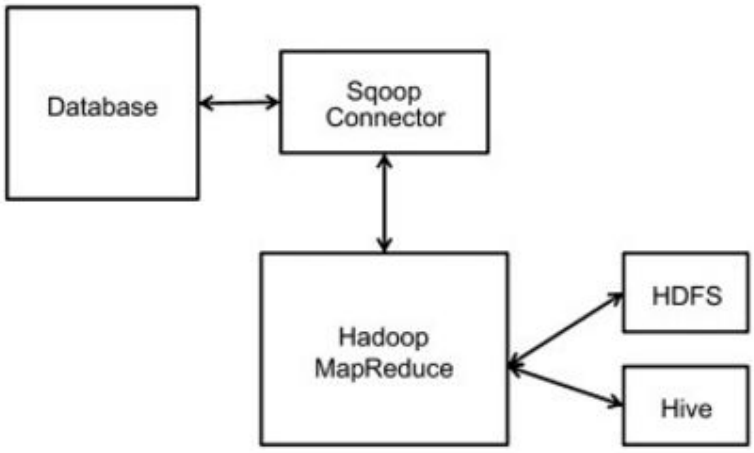


Figure 2. Use of Hive in MapReduce framework [6].

d. Spark

Apache Spark is a processing framework which allows users to run data analysis of large datasets on clustered computers. Spark has the ability to process data from a multitude of sources, including but not limited to: HDFS, SQL, and Hive. Spark has personal memory storage that is used to enhance performance so long as the dataset being analyzed is small enough to fit into that memory; if not, it uses the computer's storage instead. When datasets are being processed on memory, Spark can run up to one hundred times faster than MapReduce, and when processed using a computer's storage, it can run up to ten times faster than MapReduce. Spark can process more batches at a time than MapReduce's maximum limit, and its fault-tolerance system, RDD (Resilient Distributed Datasets), is much more efficient than MapReduce method of copying every single bit of information because it only restores the part that was lost, freeing up a significant amount of memory. This type of system only reads/writes onto disk when needed, contrasting with Hadoop MapReduce, therefore proving Spark as a much faster and more efficient framework than MapReduce [11].

III. USE OF SQOOP

Bhardwaj [2] et al. capitalizes on companies' need for Big Data to make educated business decisions. The complexity of Big Data lends itself to the development of technologies like Hadoop, Hive, Pig, and Sqoop in order to flexibly analyze datasets and improve the performance of the basic Hadoop framework.

Bhardwaj et al. accents the optimal uses for this wide variety of Hadoop technologies that can further be utilized to smooth the usage of MapReduce [2]. Sqoop, in particular, takes advantage of simple data structures to import/export data from RDBMS to Hadoop and vice versa. By acting as a go-between, Sqoop can easily move data from the RDBMS to the Hadoop file system (see Figure 3).

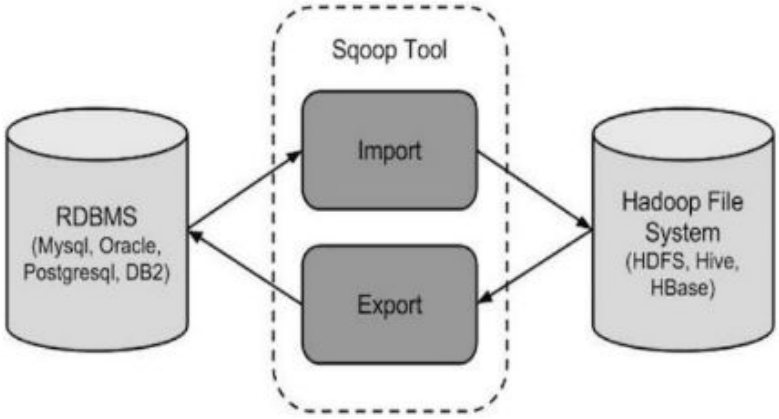
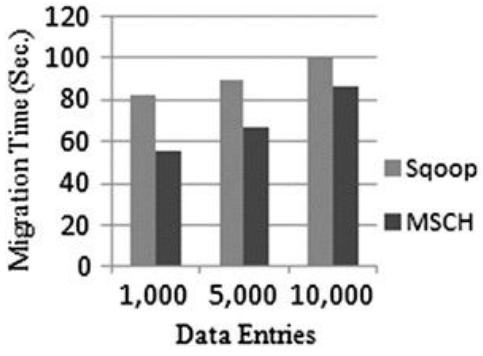


Figure 3. Flow chart of Sqoop profile [2].

Sqoop was first developed by Cloudera and supports external files, differing from numerous other Hadoop tools [12]. Other goals behind the establishment of Sqoop included allowing the transfer of data using simple commands that could be executed from a CLI, allowing for client-side installation, use of Hive and HDFS for data processing and Oozie (Apache job coordinator) for job management; and connector based architecture with plugins from the sellers. Used by large companies such as Yahoo! and Amazon, the open-source Sqoop tool is a practical technology used for Big Data management [13].

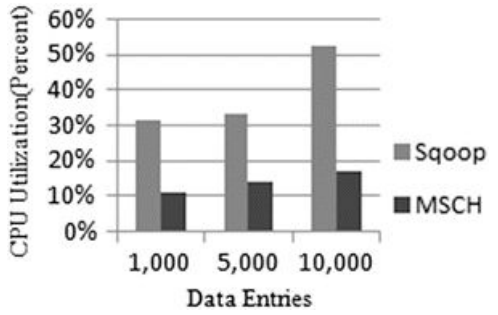
IV. DESIGN ISSUES AND PERFORMANCE

In a 2015 experiment, Yang [7] et al. compares the efficiency of Sqoop and MSCH (MySQL Convert to HBase). Sqoop has the ability to migrate data; however, it has three weakness. Sqoop becomes inefficient as the user must manually enter the migration command in the terminal [14]. It is also notable that not all Sqoop versions are similar; Hadoop and HBase will not recognize all Sqoop programs as the same [15]. Mismatched versions will result in a program fault. Lastly, an increase in data may lead to an increase in CPU utilization and memory usage, causing the computer to stop responding and crash. In an experiment conducted by Yang between Sqoop and MSCH, the amount data storage disposed compared to the amount of entries given was researched. Analyzing the graphical data provided, he discovered that Sqoop consumed a larger amount of memory than MSCH. With a sample group of one thousand, five thousand, and ten thousand entries, Sqoop significantly used an exceeding amount of memory (See Figure 4) [7].



a.

b.



c.

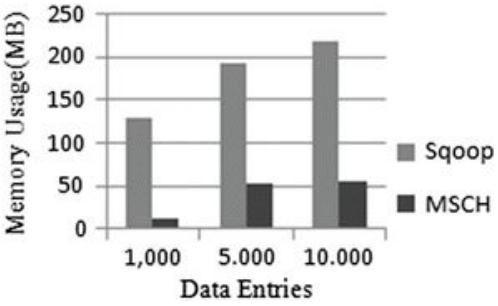


Figure 4. Results of Yang's Sqoop experiment [7].

Project Description

In order to grasp a deeper understanding of the Hadoop framework and Sqoop technologies, we installed Hadoop using a virtual machine and Ubuntu, allowing us to conduct experiments using the tool. We utilized Amazon Web Services to support cloud computing, and, through the Hadoop framework, we implemented sorting algorithms such as word count, allowing us to discover first-hand the workings of Hadoop. Despite Hadoop using several tools to manipulate data, we chose to focus on one in order to assiduously analyze its impact. Prior to investigating Sqoop, we researched Hadoop's purpose and specifications using sites such as IEEExplore. Following comprehensive research, we discovered the applicative components of Hadoop and Sqoop in controlling Big Data.

Processes

We were able to gather information regarding our focus area, Hadoop and Sqoop, through published academic works and our research mentors. Along with the constant analysis of various articles related to Big Data, we also conducted experiments in which we employed Hadoop. We carried out a total of three experiments which helped us gain a deeper understanding of the practicality of the Hadoop cluster. The first experiment was to code a program to be used for reading the text files. The second experiment's task was also to write a code to count the number of particular words that were provided as an input, and the purpose of the third experiment was to properly install Hadoop and implement it.

The first and second experiments that we conducted familiarized us with the elementary processes used by Hadoop to systemize its data. In the second experiment, we were required to code an algorithm to sort a list of fruit names. This task included various tactics such as passing each line to individual mapper instances, map key value splitting, sort and shuffle, and reduce key value pairs. Although we also executed the word count task on the third experiment, the processes were not similar to those of the second experiment.

The Hadoop installation process, the third experiment, required us to follow several commands that we were provided through our mentors. We then used an open-source platform, Ubuntu, to install Hadoop into our system and execute the word count task on it to evaluate its efficiency.

While conducting the third experiment, we began by cloning the Virtualbox, a separate computer within our computers, so that the program we were installing would not derange the files that were already in the system. Then we installed Java because Hadoop is an open source platform built on Java. After the installation of Java, we created and set up SSH (Secured Shells) certificates. SSH allows the user to login without a password and it is also required so that master node can access and run the slave and secondary nodes. We then downloaded and installed Hadoop in Pseudo Distributed Mode (where different Hadoop daemons run in different Java processes) and prepared Environment variables for Hadoop on master and slave nodes by appending several commands to the "bashrc" file. The "bashrc" file provides a place where you can put any command used in your particular environment or modify commands to your preferences. After we updated and restarted the "bashrc" file again, we reset the Java environmental variable in "hadoop-env.sh" file by replacing the "JAVA_HOME" value with the location of Java in our system. We also edited numerous "xml" files such as "mapred-site.xml," "hdfs-site.xml," and "core-site.xml" in order to configure Hadoop. After editing the "xml" files, we created two folders; "namenode" and "datanode" and formatted the Hadoop file system to enable us to use it. Finally, we copied the input data to a text file and tested the efficiency of Hadoop by running the word count task using the MapReduce framework.

Data Results and Discussion

After implementing word count on our selected dataset, Hadoop proved to be quite user-friendly and incisive, requiring clear commands and correct formatting to complete tasks. After conducting an experiment on a text file containing a list of random words, the runtime was approximately one second using one node. We were then able to compare our results with a study from MIT, however, not under the same circumstances. MIT tested a 535 megabyte file of random words and incremented the nodes by ten and twenty-five (See Figure 5) [16]. After scaling down their results, we shared similar runtimes. Despite utilizing a single node, we successfully ran Hadoop. Our result also concluded that we had successfully appointed MapReduce in Hadoop. As MapReduce is Hadoop's most mature framework, the successful appointment of MapReduce also proved the competence of Hadoop cluster. While analyzing the information regarding our focus area, we found that despite the efficiency of Hadoop, it does not allow the processing of unstructured data that could be convenient in certain circumstances; thereby, Sqoop was created to allow such processing in Hadoop.

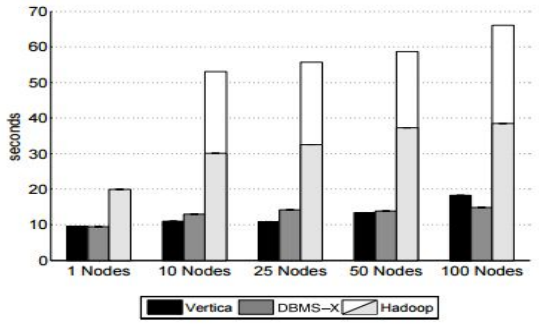


Figure 5. MIT's experiments on a 535 megabyte file [16].

Conclusion

After thorough research and engaged workings with Apache Hadoop, our team was able to gain a hands-on experience with the program. By first installing Hadoop, we witnessed how other program users, such as our peers, may have experienced the same process, while any difficulties concerning the installation were also handled. We were able to complete tasks with MapReduce and managing text files. When faced with the task of implementing word count, we were able to test the user-friendliness of the program and extrapolate its further uses when applied to large businesses such as Yahoo! and Amazon. A text file containing random words was also tested using Hadoop and MapReduce. This experiment resulted in a significantly low runtime using a single node. We concluded that the file size utilized in the Hadoop word count task significantly impacted the runtime of the program.

Limitations of our study included the availability of resources, as Sqoop is a fairly nascent technology, thereby restricting the number of published experiments and articles. We did not use an extensive amount of data during our word count experiment; however, supplemental information was used to confirm the test results. A more extensive time frame to conduct experiments and research may have also resulted in a significantly more detailed investigation, as well as the opportunity to research interconnected areas of knowledge. The limited knowledge of the Virtualbox and Ubuntu prior to the installation of Hadoop also proved more difficult to set-up; however the task was successfully completed.

Extended research on the Hadoop tool, Sqoop, may include further analyzing the time complexity and efficiency when dealing with differing amounts of data or the experimentation with different Hadoop technologies. Ultimately, we found that the simple interface and direct commands make Hadoop a viable tool for data management.

Acknowledgements

Our team would like to offer a sincere 'thank you' to the United States Department of Defense for funding this program, thereby allowing our team the opportunity to research and gain experience with Apache Hadoop and its tools. Research and experimentation would have been difficult without the didactic aid of AEOP Unite instructors, mentors, and facilitators for providing the materials throughout the duration of this study. Lastly, 'thank you' to our research

mentors, Mr. Haysam Selim and Mr. Sai Phani Krishna, for providing a multitude of resources, guidance, and support during our endeavors.

References

[1] “Sqoop - Project License,” Sqoop - Project License. [Online]. Available: http://sqoop.apache.org/license.html.

[2] A. Bhardwaj, Vanraj, A. Kumar, Y. Narayan and P. Kumar, "Big data emerging technologies..//: A Case Study with analyzing twitter data using apache hive", Presented at 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), 2015.

[3] R. Devakunchari, "Handling big data with Hadoop toolkit", Presented at International Conference on Information Communication and Embedded Systems (ICICES2014), 2014.

[4] “MapReduce Tutorial,” MapReduce Tutorial. [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.

[5] C. Data, "20 essential Hadoop tools for crunching big data," in Hadoop, Big Data Made Simple - One source. Many perspectives., 2014. [Online]. Available:

[6] K. Krishnan, "Introducing Big Data Technologies - Data Warehousing in the Age of Big Data - Chapter 4", Sciencedirect.com, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780124058910000040.

[7] S. Yang, C. Tu and J. Lin, "Design Issue and Performance Analysis of Data Migration Tool in a Cloud-Based Environment", Presented at Proceedings of the 4th International Conference on Computer Engineering and Networks, pp. 749-759, 2015.

[8] BTI360, "Introduction to Apache Mahout," in YouTube, YouTube, 2012. [Online]. Available: https://www.youtube.com/watch?v=WB9zr0IZCPQ.

[9] “What is Pig?,” IBM – What is Pig and Pig Latin – United States. [Online]. Available: https://www-01.ibm.com/software/data/infosphere/hadoop/pig/.

[10] “What is Hive?,” IBM. [Online]. Available: https://www.01.ibm.com/software/data /infosphere/hadoop/hive.

[11] "Big Data Processing with Apache Spark – Part 1: Introduction." InfoQ. N.p., n.d. Web. 21 June 2016. <https://www.infoq.com/articles/apache-spark-introduction>. http://bigdata-madesimple.com/20-essential-hadoop-tools-for-crunching-big-data/.

[12] I. Cloudera and Cloudera, Inc., "Sachin2508," 2011. [Online]. Available: http://www.slideshare.net/cloudera/apache-sqoop-a-data-transfer-tool-for-hadoop?nex t_slideshow=1.

[13] T. Hall, "Apache Sqoop," in Sqoop, Hortonworks, 2015. [Online]. Available: http://hortonworks.com/apache/sqoop/.

[14] S. S. Aravinth, S. Shanmugapriyaa, A. Haseenah Begam, and S. Sowmya, "An Efficient HADOOP Frameworks SQOOP and Ambari for Big Data Processing," 2015. [Online]. Available: http://www.ijirst.org/articles/IJIRSTV1I10027.pdf.

[15] "Apache HBase – Apache HBase™ home," 2007. [Online]. Available: https://hbase.apache.org/.

[16] A. Pavlo *et al.*, "A Comparison of Approaches to Large-Scale Data Analysis," 2009. [Online]. Available: http://db.csail.mit.edu/pubs/benchmarks-sigmod09.pdf.