REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
	Technical Report	-

4. TITLE AND SUBTITLE
Big Data and Big Data Analytics

5a. CONTRACT NUMBER
W911NF-12-1-0081

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
206022

6. AUTHORS
Kaila Perry

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Norfolk State University
700 Park Avenue

Norfolk, VA 23504 -8060

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
60475-CS-REP.26

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
Big data and big data analytics are among the fastest growing and most strategically important technologies for businesses and organizations alike in both public and private sector. They are also among the most active and rapidly evolving areas of research, development, and application deployment.
The primary objectives for this research project are: 1) Understand the fundamentals of big data and big data analytics; 2) Gain hands-on knowledge and skills in working with these technologies; and 3) Apply these techniques for performing representative security analysis tasks.

15. SUBJECT TERMS
Big data

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Chung-Chu (George) Hsieh
UU	UU	UU	UU		19b. TELEPHONE NUMBER 757-823-8313

Report Title

Big Data and Big Data Analytics

ABSTRACT

Big data and big data analytics are among the fastest growing and most strategically important technologies for businesses and organizations alike in both public and private sector. They are also among the most active and rapidly evolving areas of research, development, and application deployment.

The primary objectives for this research project are: 1) Understand the fundamentals of big data and big data analytics; 2) Gain hands-on knowledge and skills in working with these technologies; and 3) Apply these techniques for performing representative security analysis tasks.

This project uses the open source Apache Hadoop platform and related tools, which are the most widely used building blocks for big data and big data analytics. Two different Hadoop distributions, Cloudera QuickStart and Hortonworks Sandbox, have been used. They both run as VirtualBox virtual machines in a Mac OS X notebook.

The security analysis tasks practiced for this research cover the major areas of the workflows: 1) Identify the data sources; 2) Extract, transform, and load the data into Hive which provides a SQL-like language called HiveQL for querying and managing large datasets residing in Hadoop's distributed file system; and 3) Use HiveQL to query the data and attempt to find answers to the research questions.

Two types of data sources have been used for this project. The first type represents static text files that are downloaded from websites. The second type represents dynamic streaming data, such as log files, that are collected using Apache Flume.

BIG DATA AND BIG DATA ANALYTICS

By

Kaila Perry

Technical Report

CSC 499

COMPUTER SCIENCE SEMINAR II

April 16, 2014

Approved by:

George Hsieh

_____

Dr. George Hsieh, Research Advisor

ABSTRACT

BIG DATA AND BIG DATA ANALYTICS

Kaila Perry

Norfolk State University, 2014

Advisor: Dr. George Hsieh

Big data and big data analytics are among the fastest growing and most strategically important technologies for businesses and organizations alike in both public and private sector. They are also among the most active and rapidly evolving areas of research, development, and application deployment.

The primary objectives for this research project are: 1) Understand the fundamentals of big data and big data analytics; 2) Gain hands-on knowledge and skills in working with these technologies; and 3) Apply these techniques for performing representative security analysis tasks.

This project uses the open source Apache Hadoop platform and related tools, which are the most widely used building blocks for big data and big data analytics. Two different Hadoop distributions, Cloudera QuickStart and Hortonworks Sandbox, have been used. They both run as VirtualBox virtual machines in a Mac OS X notebook.

The security analysis tasks practiced for this research cover the major areas of the workflows: 1) Identify the data sources; 2) Extract, transform, and load the data into Hive which provides a SQL-like language called HiveQL for querying and managing large datasets residing in Hadoop's distributed file system; and 3) Use HiveQL to query the data and attempt to find answers to the research questions.

Two types of data sources have been used for this project. The first type represents static text files that are downloaded from websites. The second type represents dynamic streaming data, such as log files, that are collected using Apache Flume.

Table of Contents

INTRODUCTION

Data can be collected from everywhere and everything, resulting in large amounts of data to be stored. All of this data collection is leading to a "mountain of data" known as big data [1]. Big data is large volumes of data in its most raw form that is difficult to store, and even more difficult to analyze to retrieve valuable information. Big data comes from a variety of sources including uploading and downloading pictures, interactions on social media sites, online transactions, sensors on bridges, and a plethora of other things. From a security perspective, big data can reveal valuable information that can be used to detect cyber crimes such as espionage, computer intrusions and computer fraud [2]. From many other perspectives, big data is used as a means of revealing patterns to better understand correlations, visualize the information, and make better decisions or verify/disprove theories based off of that information [2]. The process of analyzing all of this data is referred to as big data analytics, which researches the big data in order to find hidden patterns and secret correlations to gain useful knowledge and insights [3].

To help alleviate the hassle of filtering through the data and trying to find ways to retrieve valuable information, tools have been created to assist the process. Throughout this research, in order to get a better understanding and to have a hands-on experience, various platforms and tools will be used for learning, experimenting, and developing prototype applications. One of the tools used to analyze big data consists of Apache™ Hadoop®. Hadoop is a software framework that can be used to develop programs to process large sets of data [3]. The goal of this project is to install Hadoop and other platforms, understand how to operate them by completing the tutorials, and then to use the platforms and tools to develop an application that will sort through the data to show

the frequency of word occurrences and the security implications depicted from it. The expected deliverables include the installation and operation of a small-scale big data development environment, and a demonstration of the big data security application.

BACKGROUND

Analyzing big data is not something that is typically taught in class, but it is a technology that is becoming increasingly important for businesses and government. Big data analyticts is something that is learned through experimenting and finding others who have done similar work. Through research, it was found that search engines and social networks use data analysis techniques to find search phrases and to recommend friends for users [4]. Some of the techniques and algorithms used to search through data include shingling, minhashing, locality-sensitive hashing, collaborative filtering, and the Girvan-Newman algorithm [4]. Shingling, minhashing, collaborative filtering, and locality-sensitive hashing are used for finding similarities between documents, and the Girvan-Newman algorithm is used for finding similarities between social network users [4]. Shingling finds similarities by comparing short strings (comparing documents character by character) [4]. Minhashing compresses large datasets to find similarities through using matrices and permutations [4]. Collaborative filtering is a process that recommends items to a user based on that item being liked by other users with similar taste [4]. Locality-sensitive hashing focuses on finding the pairs of results that are most likely to be similar [4]. The Girvan-Newman algorithm uses social networks, which are "modeled as graphs," and finds the shortest path from one node to another [4].

Furthermore, the tools that were found to potentially be used to implement this project include Coreservlets, Cloudera, and Hortonworks Sandbox which are all Hadoop based. As mentioned before, Hadoop is a linux-based, open-source, software framework that allows for the storage and processing of large sets of data across clusters of computers using simple programming models [5]. The tutorials on "developing Big Data Applications with Hadoop" that are distributed by Coreservlets, was a tool that was briefly used [6]. These tutorials seem to be more useful for people who are more experienced with using Unix commands and have an idea of what the results should look like. For beginners, when using the Coreservlets tutorials it is also difficult to complete the tutorials on the virtual machine that show the user how to run the code. In general, the tutorials are not as descriptive as they could be.

Another tool that was used for experimentation was the Cloudera Quickstart VM. The Cloudera environment is a real time platform that allows querying of data [7]. It has a website with access to Apache Hadoop tutorials, which was briefly investigated as a tool for learning to develop big data applications, but was not used at first because of hardware and software compatibility issues [8].

Of the big data environments investigated, Hortonworks Sandbox seemed to be the most attractive for the purpose of learning to analyze big data. It is a portable Hadoop environment that is easy to use and understand [9]. The Sandbox environment runs on a virtual machine, but the interface is displayed in a browser that allows the user to experiment with the data and view the tutorials simultaneously.

METHODOLOGY

In order to complete this project, research on big data and big data analytics were completed to get a basic understanding of the application domain. Research was also completed to determine the tools that are available and most useful for the implementation of this project. Experimenting with the big data environments that were identified was vital to getting a grasp of how they work, as well as taking into consideration their pros and cons.

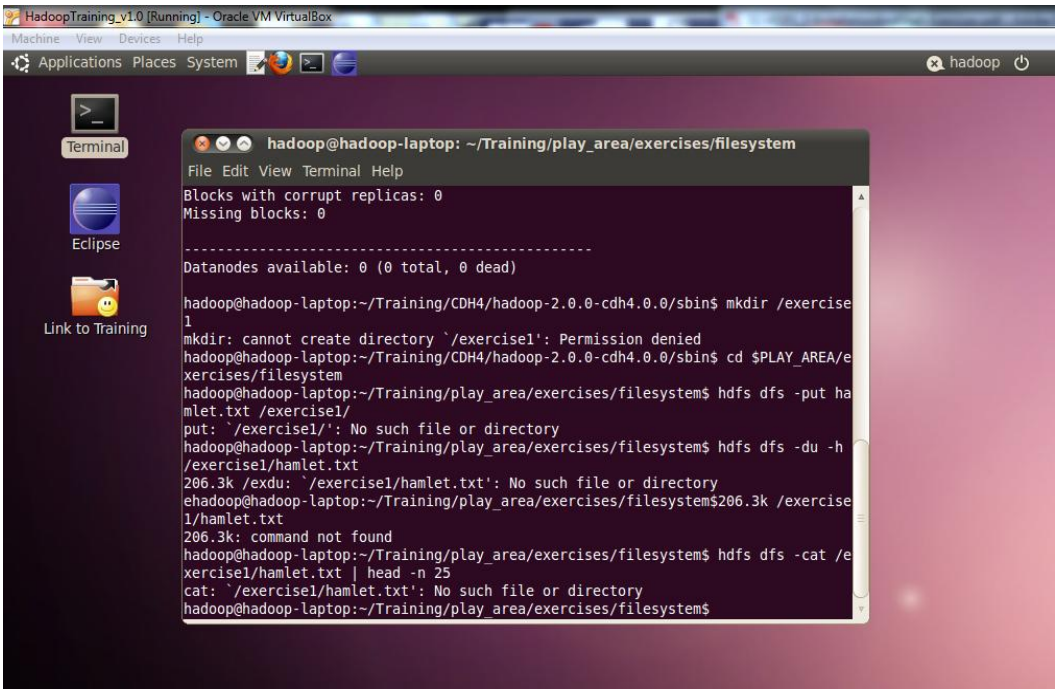
In regards to getting an understanding of big data, the research found revealed that big data is usually categorized into the "four V's of big data" which are volume, velocity, variety and veracity [10]. Volume refers to the amount of data, velocity refers to the speed at which the data can be processed, variety is the number of types of data, and veracity is the uncertainty of data [10]. Being able to have an understanding of the different versions of data will allow companies or even individuals to make better decisions from the data, or to get meaningful information from it. Some companies that currently use big data to advertise or make decisions for their companies include Google, Facebook, EBay, Yahoo!, Twitter, and LinkedIn, just to name a few [11]. Just to note, all of these companies are using Hadoop to analyze their data.

Additionally, because all of the big data applications used were Hadoop based, having a basic understanding of Hadoop was fundamental. As mentioned before, Hadoop is an open-source framework that allows processing of large data sets across clusters of low-cost commodity hardware in a highly efficient manner [12]. It began with Google publishing two papers in 2003 and 2004, called the Google File System and MapReduce.

Doug Cutting began working on the implementations of those two papers, and successfully, Hadoop was born [12]. The two main components of Hadoop are now the Hadoop Distributed File System (HDFS) and MapReduce, hence the two papers that were published before the creation of Hadoop. The purpose of HDFS is to store large sets of data [12]. The "distributed file system" aspect simply means that the data storage is spread across multiple nodes. Some features of HDFS include its ability to store files in blocks much larger than the storage size of most filesystems, it is "optimized for throughput over latency," and each storage node runs a process called a "DataNode" that is ran by a 'master' NameNode [12]. MapReduce is a data paradigm that takes specifications of how the data will be input and output, and then applies the method across large datasets [12]. It integrates with HDFS by running tasks directly on the nodes that hold the data being analyzed [12]. Another feature of MapReduce is that it uses a "divide and conquer" concept to complete tasks [12].

Moreover, to begin experimenting with the big data application development environments, proper installation needed to be done. All of the environments found can be run in virtual machines (VM). The hypervisors used to run these virtual machines were Oracle VM VirtualBox, VMWare Workstation (for Windows), and VMWare Fusion (for Macintosh).

The first environment that was installed was the "Hadoop Training VM" from Coreservlet.com. This virtual machine required VirtualBox for installation. Once installed, there were a series of tutorials to help with becoming more comfortable with using the environment. A few of the tutorials were completed, but were not as helpful as expected, so further investigation of the tool did not occur.

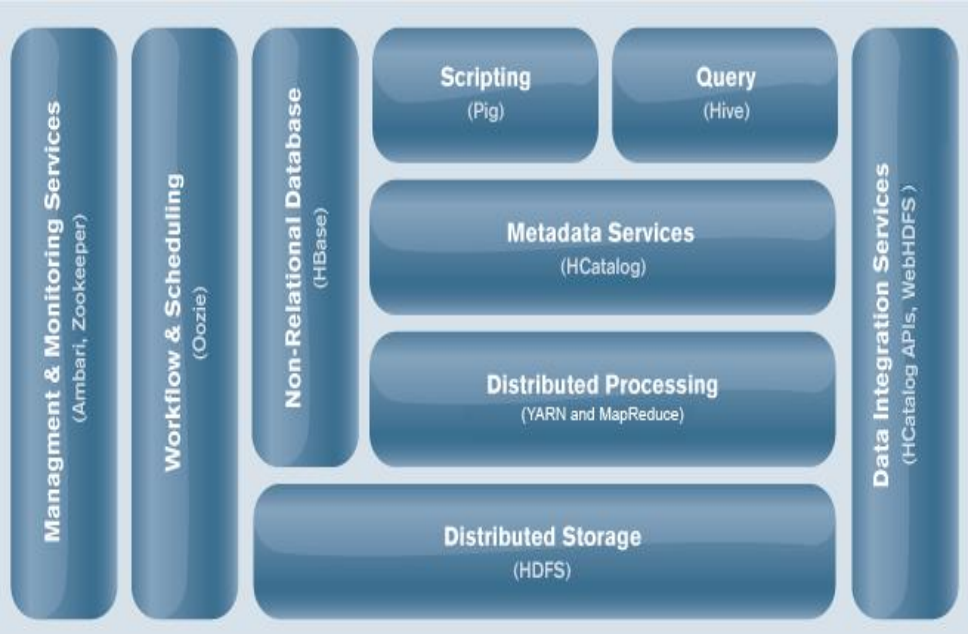


This is a sample of what the CoreServlets VM looks like

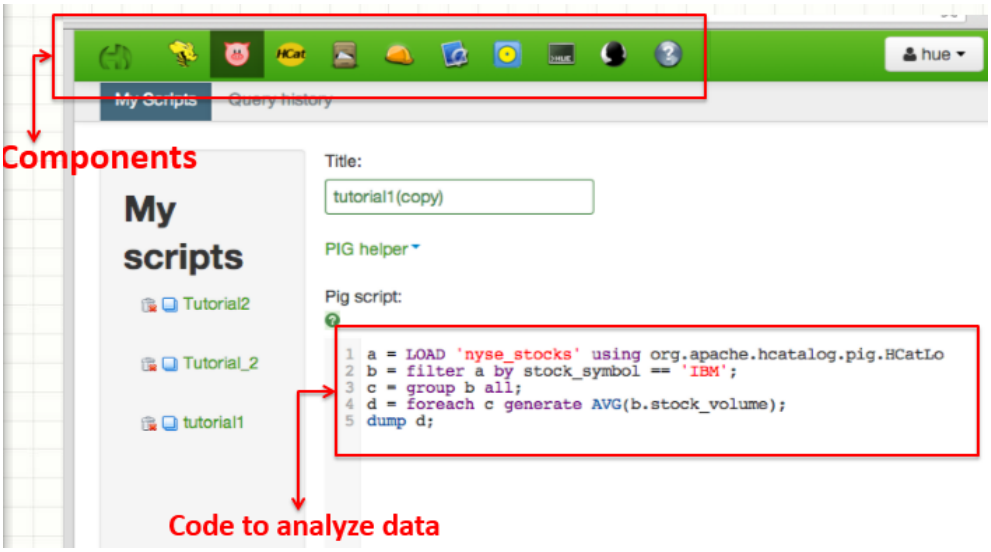
In an attempt to install Cloudera on VirtualBox, there were some issues with the Windows operating system virtualization technology. The laptop being used was a Dell Inspiron, and it did not support Intel VT-X, which was needed in order to install Cloudera. Cloudera was never installed on that particular machine, but it was installed on a Windows desktop machine in the lab in Robinson Technology Center in Room 300, and later on an Apple, MacBook Pro. Because of the hardware issues with the Windows laptop, and finding a different tool that was more user-friendly, experimentation with Cloudera lie dormant for some time. After realizing that this tool may be the most beneficial in terms of independent application development, the Cloudera Quickstart VM was installed on a Macintosh laptop on a VirtualBox hypervisor. The Cloudera QuickStart VM was found to be better for application development because it comes pre-installed with software development tools such as Eclipse, and it contains a working

browser within the environment that allows the user to install add-ons or other packages within the virtual machine. Overall, it was more flexible in terms of experimentation than the more "user-friendly" tool found, called Hortonworks Sandbox.

With that said, the installation of Hortonworks Sandbox seemed to be more advantageous and user-friendly at first. It required VirtualBox for installation, and it was found that it does not work well with other VM hypervisors such as VMWare Workstation. Once installed, a few of the tutorials were completed to analyze the sample big data that came with the framework. In order to fully understand how the environment worked, research was done on the different components. Some of the components of the Hortonworks Sandbox include File Browser, Apache Hive, HCatalog, Pig, and Beeswax. The File Browser component is comparable to a file manager used in a Windows operating system. It stores uploaded files. Apache Hive is like a data warehouse that "facilitates querying and managing of large datasets" [13]. Apache Hive is comparable to the Structured Query Language (SQL). Equally important, HCatalog uses the uploaded files that are stored in the File Browser to create a table. Pig is the scripting language that is used with Apache Hadoop [14]. Finally, Beeswax is the interface for Hive that allows for the data to be seen as a table. The images below show the Hortonworks Data Platform, and the location of the components in the virtual machine's window during the experimentation of the tool. The diagram of the Hortonworks Data Platform is just a subset of Hadoop, and it displays that Hadoop is powerful, but also very complex.



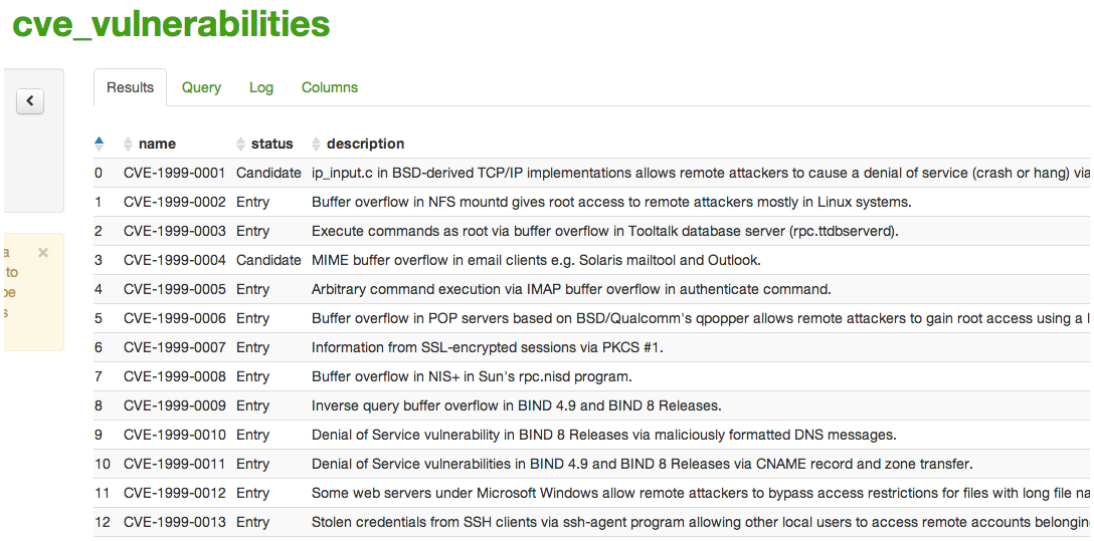
Hadoop with additional tools: diagram of the Hortonworks Data Platform 2.0 [15]



View of Hortonworks Sandbox during experimentation showing the location of its components and where potential code can be input.

Other exploration with Hortonworks entailed uploading a static text file, in the form of a comma-separated value (CSV) file, into the platform. The CSV file came from the Common Vulnerabilities Exposure (CVE) website. The file was composed of information of the known vulnerabilities including the name, a description, a phase, the

status, votes, comments, and references. By uploading the file into Hortonworks through the File Browser, the HCatalog component allowed for that file to be formatted into a table. The table created looks like the image below, except all of the fields are not visible in the image.



cve_vulnerabilities

Results | Query | Log | Columns

	name	status	description
0	CVE-1999-0001	Candidate	ip_input.c in BSD-derived TCP/IP implementations allows remote attackers to cause a denial of service (crash or hang) via
1	CVE-1999-0002	Entry	Buffer overflow in NFS mountd gives root access to remote attackers mostly in Linux systems.
2	CVE-1999-0003	Entry	Execute commands as root via buffer overflow in Tooltalk database server (rpc.ttdbserverd).
3	CVE-1999-0004	Candidate	MIME buffer overflow in email clients e.g. Solaris mailtool and Outlook.
4	CVE-1999-0005	Entry	Arbitrary command execution via IMAP buffer overflow in authenticate command.
5	CVE-1999-0006	Entry	Buffer overflow in POP servers based on BSD/Qualcomm's qpopper allows remote attackers to gain root access using a l
6	CVE-1999-0007	Entry	Information from SSL-encrypted sessions via PKCS #1.
7	CVE-1999-0008	Entry	Buffer overflow in NIS+ in Sun's rpc.nisd program.
8	CVE-1999-0009	Entry	Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases.
9	CVE-1999-0010	Entry	Denial of Service vulnerability in BIND 8 Releases via maliciously formatted DNS messages.
10	CVE-1999-0011	Entry	Denial of Service vulnerabilities in BIND 4.9 and BIND 8 Releases via CNAME record and zone transfer.
11	CVE-1999-0012	Entry	Some web servers under Microsoft Windows allow remote attackers to bypass access restrictions for files with long file na
12	CVE-1999-0013	Entry	Stolen credentials from SSH clients via ssh-agent program allowing other local users to access remote accounts belongin

View of the CVE file as a table in Hortonworks Sandbox

Once the file was formatted as a table in Hortonworks, the Hive component was used to be able to search through all of the data to find information based off of a string value. In this case, the Hive query searched through the table in the field labeled “description” and found all of the instances of vulnerabilities that contained a buffer overflow. The query is shown below.

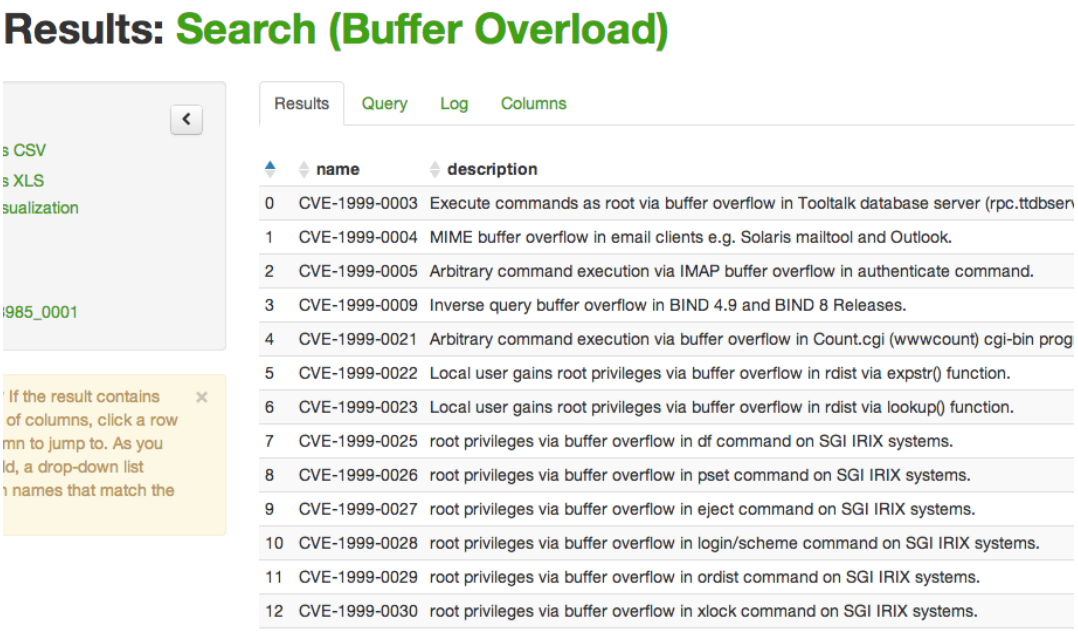
Query Editor : Search (Buf

Lists vulnerabilities w/ buffer overload in description

```
1 SELECT name, description FROM cve_vulnerabilities
2 WHERE description LIKE '%buffer overflow%';
```

Hive query used to search for all instance of a "buffer overflow" in the description of the vulnerabilities.

The query implements this task by selecting the name and description of the vulnerability from the "cve_vulnerabilities" table for every instance of the description containing the string "buffer overflow." The results of this query show the names and descriptions of all of the vulnerabilities that have a "buffer overflow" in their description.



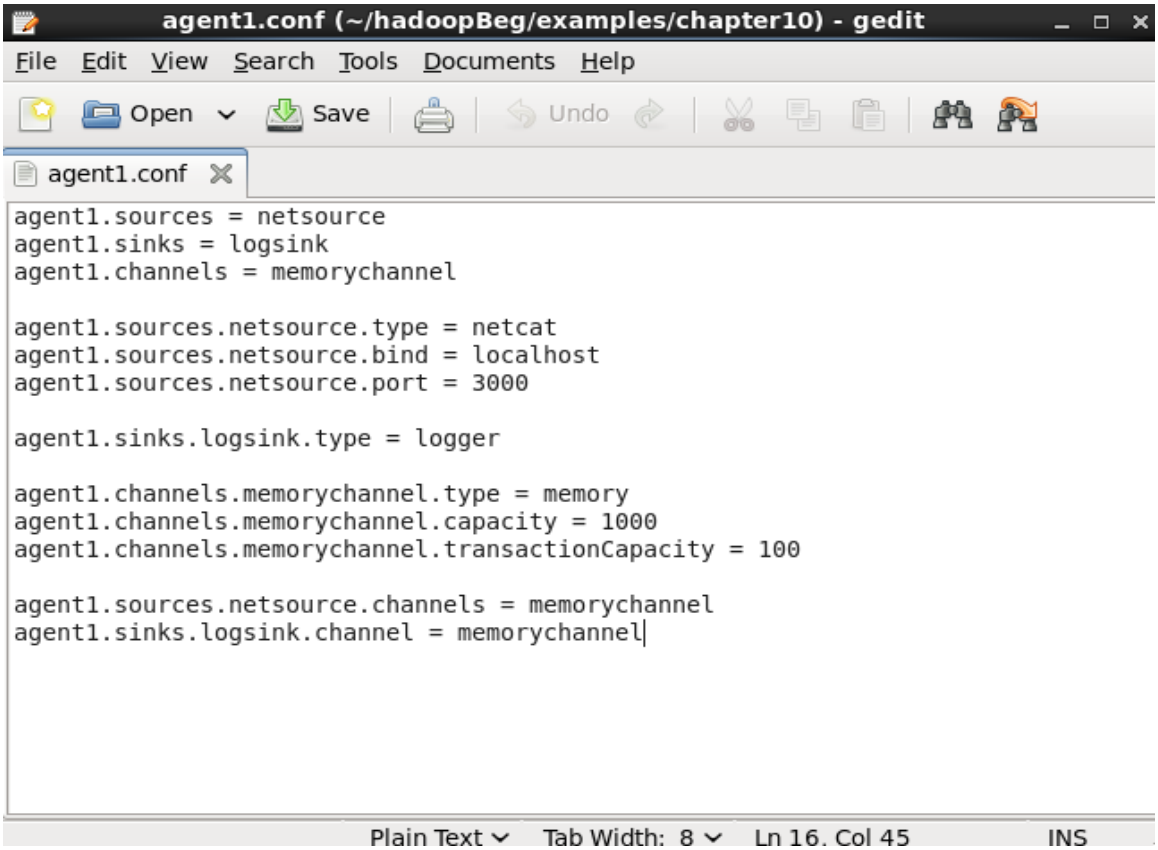
Results displayed after the search query is executed

As a result of this, it was found that analyzing data in this fashion would be useful for getting specific data from a file to make the data set into a smaller, more manageable file. A benefit of Hortonworks Sandbox is that once the query has been executed successfully, the results can be downloaded onto a personal machine into another CSV file or an excel (XLS) file. By being able to reduce the size of the data sets, an even more specific search can be done. For instance, now that all of the vulnerabilities from this file that contain a buffer overflow have been gathered into one group, a more complex query

can be done to find more exact results. While this seems ideal for some, a disadvantage of analyzing data in this way could be that if the data is very unstructured, it may not work; however, this is an assumption that has not yet been tested.

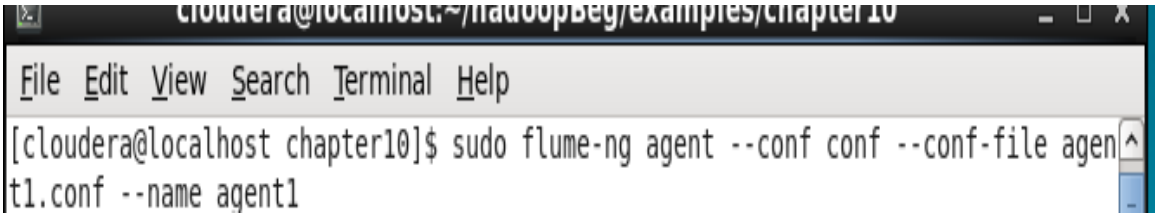
Furthermore, when first working with Cloudera, many issues occurred; however, the issues encountered did not prevent experimentation with the tool. The Cloudera Quickstart VM was revisited with the intent of using a tool called Flume to retrieve data from a webserver. Flume is simply a tool that can retrieve data from multiple sources and deliver it to a variety of destinations including a log file or the console of the machine [12]. The goal was to capture some data using Flume, upload that data into the Hadoop Distributed File System, and then run queries on the data with HiveQL.

In order to do this, flume had to be installed on the Cloudera VM. Because it was already installed and configured, only a configuration file for the Flume agent needed to be created. The Flume agent was created inside of the working directory that contained Flume. The purpose of the Flume agent, named "agent1.conf," was to receive data through a Netcat source, that is binded to the local host and listens on a port for network connections. When the source receives a connection, it uses a sink, which in this case is a logger, to write the output to a log file (a dynamic text file). Below is the configuration file that has a Netcat source that binds to the local host and listens on port 3000.



Screen shot of the Flume agent configuration file

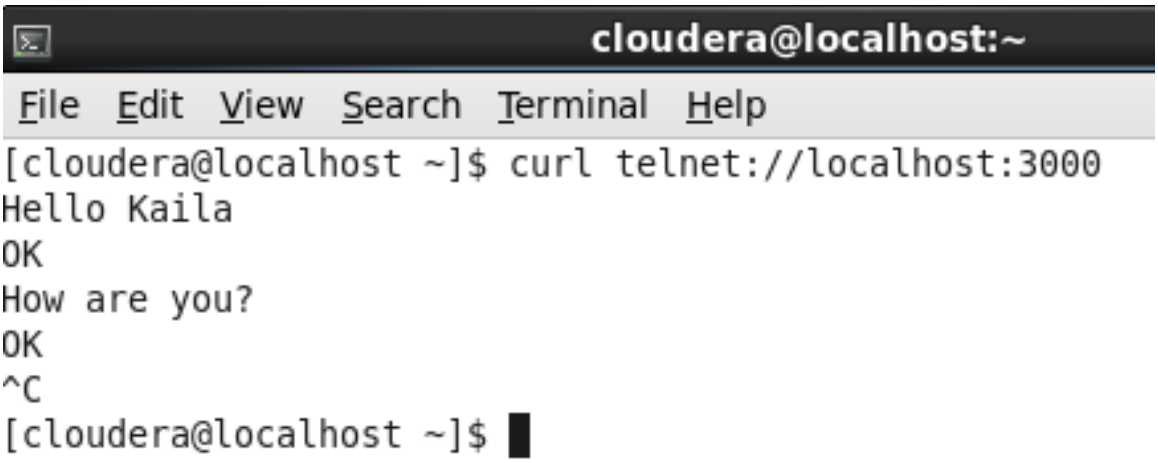
Once the configuration file has been saved, the Flume agent can be ran by navigating to the working directory, and using the 'flume-ng' command.



Command to run the Flume agent

After the Flume agent has successfully been executed, it will listen for a connection. In another terminal, a telnet connection will need to be made followed by entering some text. Below is the telnet connection, which will listen on port 3000 of the local machine.

Note that the telnet connection listens on the localhost and port 3000 because that is the information in the configuration file.



Command to open the telnet connection using curl, followed by some input

When the text is entered in the telnet connection, it is logged in a log file called 'flume.log.' To close the connects the 'ctrl + c' command is used. Now that the dynamic log file has some data in it, it can be used in Hive to run queries on.

The next step is to capture the necessary data from the original, 'flume.log,' file and input it into another log file. The data from the 'flume.log' file can be extracted into a new log file by using the 'grep' command.

```
[cloudera@localhost ~]$ cd /var/log/flume-ng
[cloudera@localhost flume-ng]$ sudo su
[root@localhost flume-ng]# grep headers < flume.log > flume.kaila.log
[root@localhost flume-ng]#
```

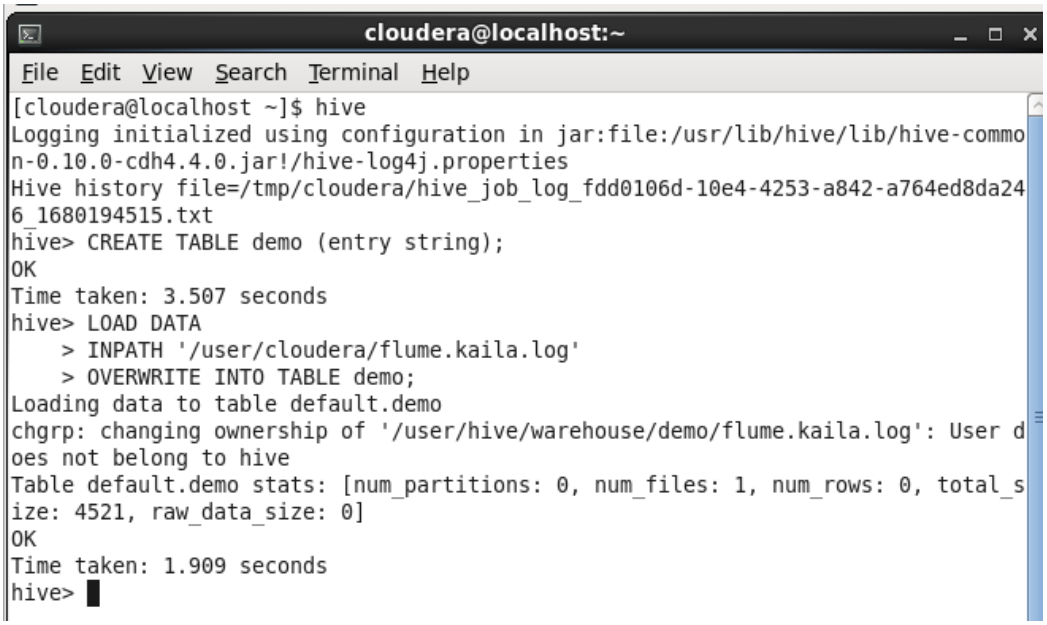
Command to extract headers from flume.log into flume.kaila.log

In this case, the headers are extracted from the 'flume.log' file into a new file called 'flume.kaila.log.' The new log file can now be copied from Cloudera into the Hadoop Distributed File System using the HDFS commands 'hdfs dfs.'

```
exit
[cloudera@localhost flume-ng]$ hdfs dfs -copyFromLocal /var/log/flume-ng/flume.k
aila.log /user/cloudera
```

HDFS command to copy kaila.flume.log into the Hadoop Distributed File System

After copying the new log file into HDFS, it can be copied into Hive in the form of a table. First, a table has to be created in Hive. To start Hive, simply type the 'hive' command in the terminal. When have has loaded, a table can be created using the HiveQL command, "CREATE TABLE," followed by the name of the required fields and their data types. The field names should correlate to the information that will be loaded into the table. In this case, the table is called 'demo' and it only has one field of type string called 'entry' that will hold the information of the headers from the log file. Below is an example of how a table can be created in Hive. The screen shot also reveals how the data can be input into Hive from HDFS using the HiveQL commands "INPUT DATA, INPATH, OVERWRITE INTO TABLE."



Screen shot of starting Hive, creating a table, and inputting the log file from HDFS into Hive

Finally, once the data is in Hive, it can be queried to search for information to help get a better understand of what the data consists of. For instance, HiveQL queries can be used to find the number of entries in the table, or certain keywords can be used to find certain entries.

```
hive> select *
    > from demo
    > where entry LIKE '%Kaila%'
```

HiveQL query to select everything from the table named 'demo' where the field called 'entry' has the word "Kaila" in it.

```
OK
07 Apr 2014 13:08:10,406 INFO  [SinkRunner-PollingRunner-DefaultSinkProcessor] (
org.apache.flume.sink.LoggerSink.process:70)  - Event: { headers:{} body: 48 65
6C 6C 6F 20 4B 61 69 6C 61                Hello Kaila }
23 Apr 2014 15:26:06,190 INFO  [SinkRunner-PollingRunner-DefaultSinkProcessor] (
org.apache.flume.sink.LoggerSink.process:70)  - Event: { headers:{} body: 4D 79
20 6E 61 6D 65 20 69 73 20 4B 61 69 6C 61 My name is Kaila }
28 Apr 2014 16:10:30,435 INFO  [SinkRunner-PollingRunner-DefaultSinkProcessor] (
org.apache.flume.sink.LoggerSink.process:70)  - Event: { headers:{} body: 48 65
6C 6C 6F 20 4B 61 69 6C 61                Hello Kaila }
Time taken: 18.272 seconds
hive>
```

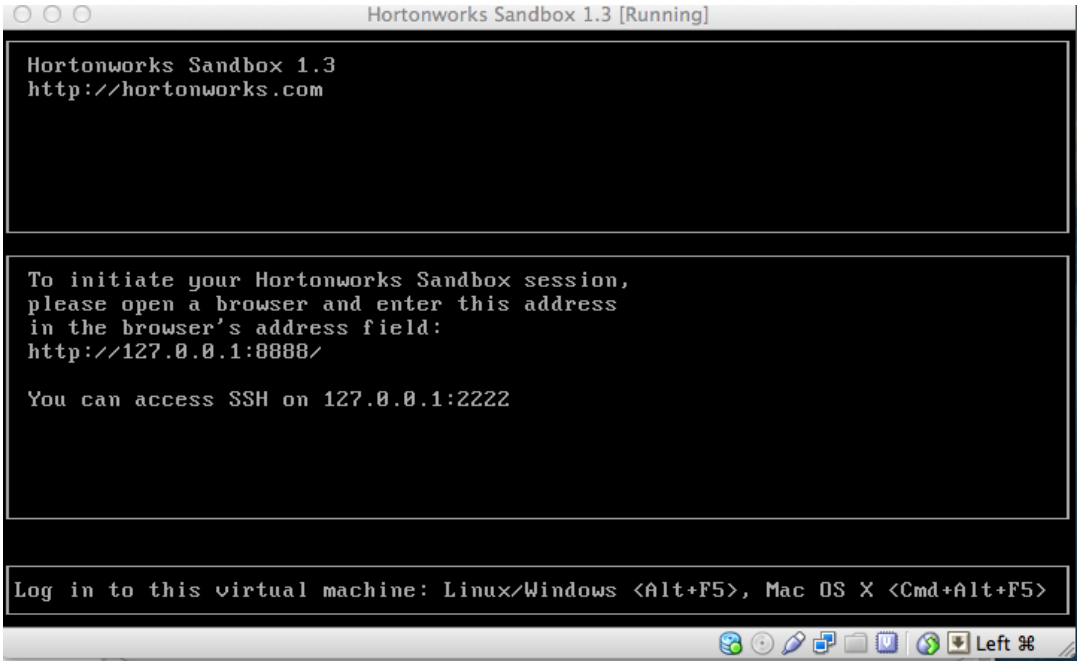
Results of the query that searched for "Kaila" in the "entry" field

Having the ability to query data sets using Hive is advantageous because it is a faster, more simplistic way to find information without strenuous work. Of course, the queries could be more complex, but for beginners, it is a beneficial way to understand big data and how to use big data analysis tools.

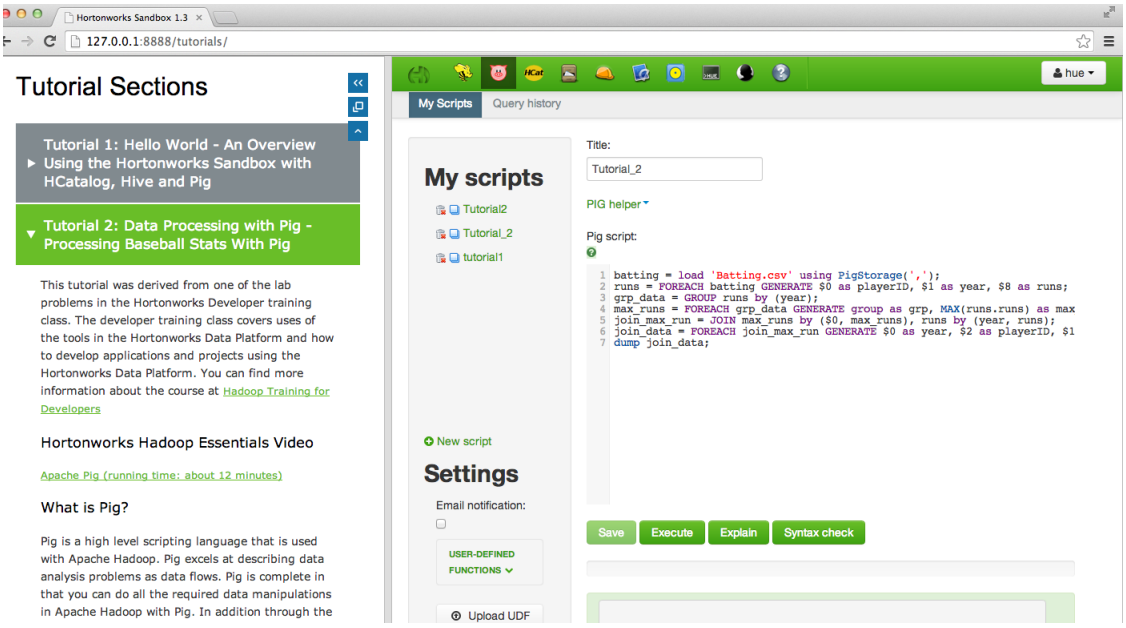
DISCUSSION

By researching and experimenting with tools associated with big data and big data analytics, it was found that there were unexpected problems with some of the environments. While completing the tutorials from Coreservlets, it was found that they were not as helpful as expected because they were not user-friendly for beginners. It seemed as though the tutorials would be more helpful for a person who already knows what he is doing and has expectations of what the outcome will be. The Cloudera environment also seemed like it was created for users with experience. One needed to be fairly experienced with linux commands to be successful.

On the contrary, Hortonworks Sandbox was very beneficial for learning the technicalities of big data application development environments. The interface is easy to use and the tutorials are straightforward and comprehensive. The Hortonworks Sandbox environment itself is ran in VirtualBox, but the interface for the environment is displayed in a browser that is ran on the host machine. When Hortonworks Sandbox is powered up, an interface in VirtualBox will be displayed with the URL, in the form of an IP address and port, to access the environment's tools and features. The URL is entered into a browser (preferably Google Chrome) on the user's host machine. When the Hortonworks Sandbox environment is accessed and the tutorial feature is selected, the tutorial documents are displayed on the left side of the screen and the tools are on the right side of the screen. This allows the user to read the tutorial and perform the actions simultaneously, eliminating the process of switching between screens in order to complete the tutorials.



Hortonworks interface that displays the URL to access the environment (http://127.0.0.1:8888).



Displays the Hortonworks Sandbox environment interface with the tutorial section on the left of the screen, and the actual environment tools on the right.

Furthermore, Cloudera is more beneficial as a software development environment. It is more flexible than Hortonworks Sandbox because the user is able to

download and install add-ons through the browser. Despite the set backs during the first use of Cloudera, using Cloudera was more suitable for completing more complex tasks, and once having a basic understanding of Hadoop tools, Cloudera gives one more experience working with the tools because everything is done hands-on through the terminal.

All of the tasks scheduled be done in Cloudera were not accomplished, but in the future more work can be done to write scripts to automate the process for dynamic data collection from multiple machines, extracting, transforming, and loading (ETL) the data into Hadoop, as well as analyzing the data in Hadoop. More complex tasks could also be implemented in Cloudera such as processing non-structured data in the form of photos or unstructured text. Additionally, for future work, tasks can be done to enhance the security of data in Hadoop by including additional authentication and access control mechanisms.

CONCLUSION

In this paper, the idea of big data and big data analytics were discussed, as well as some of the various tools used to develop applications for big data. Of the tools found, they all consisted of the Hadoop open-source framework. Through experimentation, it was found that some of the platforms were not as helpful as expected for learning purposes. However, of the tools used, Hortonworks Sandbox was highly helpful with understanding the concepts of how big data can be analyzed and the requirements for the environment, and Cloudera was more beneficial for the experimentation/software

development aspect of the project. Both tools were generally used to query a set of data using HiveQL to get some useful information from it.

Overall, it is important to keep in mind that big data technologies are becoming increasingly important for businesses and governments, and the platforms and tools are evolving and expanding rapidly. The tools used to analyze and store big data are complex, yet very powerful in order to get the job done. It is important to learn about big data and big data analytics because we are "drowning in data, but starving for knowledge." What's the use of accumulating all of this data if nothing can be done with it?

REFERENCES

[1] T. White, Hadoop: The Definition Guide Third Edition, O'Reily Media, Inc. , 2012.

[2] "Security/Intelligence Extension," [Online]. Available: http://www-01.ibm.com/software/data/bigdata/use-cases/security-intelligence.html. [Accessed 28 October 2013].

[3] S. Sagiroglu and D. Sinanc, “Big Data: A Review,” 2013.

[4] A. Rajaraman and J. D. Ullman, Mining of Massive Datasets, Cambridge: Cambridge University Press, 2012.

[5] "Hadoop," Apache Software Foundation, 16 October 2013. [Online]. Available: http://hadoop.apache.org/. [Accessed 28 October 2013].

[6] “Hadoop Tutorial:: Developing Big Data Applications with Apache Hadoop,” coreservlets.com, 2013. [Online]. Available: http://www.coreservlets.com/hadoop-tutorial/. [Accessed 1 October 2013].

[7] A. Holmes, Hadoop in Practice, Shelter Island: Manning Publications Co., 2012.

[8] “Hadoop Tutorial,” Cloudera, Inc., 2013. [Online]. Available: http://www.cloudera.com/content/cloudera-content/cloudera-docs/HadoopTutorial/CDH4/Hadoop-Tutorial.html. [Accessed 1 October 2013].

[9] "Hortonworks Sandbox Version 2.0," Hortonworks, 2013. [Online]. Available: http://hortonworks.com/products/hortonworks-sandbox/. [Accessed 8 October 2013].

[10] IBM, [Online]. Available: http://www.ibmbigdatahub.com/sites/default/files/infographic_file/4-Vs-of-big-data.jpg. [Accessed 4 February 2014].

[11] "Hadoop Wiki," 26 February 2014. [Online]. Available: https://wiki.apache.org/hadoop/PoweredBy#Y. [Accessed 9 April 2014].

[12] G. Turkington, Hadoop Beginner's Guide, Birmingham - Mumbai: Packt Publishing Ltd., 2013.

[13] "Welcome to Apache Hive," Hadoop, 15 October 2013. [Online]. Available: http://hive.apache.org/. [Accessed 18 October 2013].

[14] "Tutorial 2: How to Process Data with Apache Hadoop," Hortonworks Sandbox, 2013. [Online]. Available: http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-pig/. [Accessed 2 December 2013].

[15] "Hortonworks," [Online]. Available: http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.0.0.2/bk_getting-started-guide/content/ch_about-hortonworks-data-platform.html. [Accessed 2014].