H2H - A Data Analysis Tool for Commodity Laptops

Thesis for Master of Science in Computer Science

by Rebecca Randall-Szostak

Research Advisor: Dr. Valerie Sessions

Research Committee: Dr. Lin, Dr. Hayes and Dr. Grieves

Charleston Southern University

August, 2018

Contents

List of Figures

List of Tables

1. INTRODUCTION

Big Data represents large complex data sets, terabytes or even petabytes in size, that require more than just database management tools or data processing applications to be analyzed efficiently. Inconceivable quantities of information are produced on the web daily from individuals, industries, and devices. The need to perform big data analytics on a company's data can disclose valuable information which may influence a decision-making process and potentially impact the success of a business. However, frequently small and medium size enterprises (SMEs) fail to effectively implement technical and structural frameworks to construct the abilities to tie together some of the potential that data can be obtained from the data. SMEs often believe that big data analysis is for big corporations such as Amazon, Ebay or Google but, regardless of the size of a company, data is more important today than it has ever been for any company size. H2H is a framework of existing toolsets that are constructed in such a way that they can run efficiently on less expensive hardware, thus lowering the cost associated with data analysis. We will discuss here the framework itself, costs associated with running the stack, and analyze the results and limitations of this stack on a commodity laptop.

Analyzing big data can allow SMEs to notice things regarding their company and indicate connections, risks, chances that would have otherwise been unnoticed. Due to this, analyzing big data provides the possibility to improve decision support systems. It can also allow SMEs to simulate different scenarios and improve current products or service and even develop new ones based on the information obtained from their data analysis.

The task of setting up a framework to allow for SMEs to analyze their data can be a complex undertaking. Apache Hadoop software library is the main open-source framework for distributed

storage and distributed processing of very large datasets among clusters of computers [1]. Hadoop delivers a basic framework with tools to store data and improves the processing times for data analytics.

Other tools can be used in combination with Hadoop to improve and ease the process of analyzing data. Hive is a data warehouse software that simplifies the process of reading, writing, and managing large datasets located in Hadoop's distributed storage. Hue provides a web based user interface that allows for data stored in Hadoop to be analyzed with Hive. These three tools, Hadoop, Hive and Hue, combined yield the H2H big data analysis tool that SMEs can utilize on a commodity laptop to analyze their data. Through this research, H2H is implemented on a commodity laptop to perform data analysis on Amazon Product Review data sets. The analysis of the data will reveal:

- Products by Category with the most positive reviews
- Top 10 positive reviewers for the Amazon Instant Video category
- Top 25 Products within the CDs and Vinyl category with the largest number of positive reviews
- Top 10 Products with the most number of negative ratings in the Office Products category
- Top reviewer per category
- Top 10 Reviewers for the Home and Kitchen category
- Select all the reviews created by the 'The Movie Guys Movies from A to Z' for the Movies and TV category.

Different data sizes will be tested to test the effectiveness of H2H when analyzing the data on a commodity laptop.

2. RELATED WORK

As [17] states, big companies in all areas are creating substantial improvements in their customer associations, product choices/developments through the analysis of their big data. SMEs on the other hand have been slow to adapt to the new technologies surrounding the analysis of big data. This could be detrimental for SMEs as they play a vital role in the economy and the challenges they encounter need to be addressed or they may be negatively impacted [17]. One of the reasons identified by [17] that poor adoption existed amongst SMEs with performing big data analytics was due to the lack of intuitive software. It's either requires highly complex solutions or implementations that are simple but not so effective when analyzing the data. This research demonstrates how H2H can be deployed on a commodity laptop and used to perform data analysis to reveal information that may have otherwise been missed.

Another reason noted by [17] for the lack of adoption was due to financial barriers. Limited financial resources effect the decision of SMEs to make new investments external to their specific business scope. In [16], cloud servers were suggested as the infrastructure SMEs should adopt for analyzing their big data. While this infrastructure is viable for some SMEs to implement, others may want a smaller footprint, such using a commodity laptop, to understand the benefits of analyzing big data and what it can do for their company. H2H can be installed on any type of machine with adequate resources to run the cluster. A dedicated server machine is expensive and that is multiplied when several servers are used within a H2H cluster. Using a commodity laptop provides a less expensive alternative in comparison to servers. It also provides portability of the system in the event that one would need to work from a remote location. Using

a laptop also provides a less expensive way to allow for SMEs to experiment with H2H and how it can be used to analyze their data.

Specific requirements are needed in a laptop to effectively run H2H to analyze data. The better the processor, the better H2H will run so a fairly performant laptop is required. At a minimum, the processor should be either a dual-core or better i7 or an equivalent AMD processor. Since H2H is a three-node cluster ran in a virtual environment, the laptop will need at a minimum 12 GB of RAM to ensure adequate memory can be allocated to each virtual machine based on the component it runs. The type of storage used is less significant but a solid-state driver (SSD) over a hard disk drive (HDD) drive is preferred as it has a faster overall access speed that will improve the data manipulation process. For this research, a laptop with an Intel Core i7-7700HQ and 2.8-3.8 GHZ processor, 16GB Memory and 128GB SSD plus a 1 TB HDD of storage was use. The cost of the laptop was $1500 but other laptops are available within the identified specifications that cost anywhere between $1000 and $3500.

Related research to this thesis has been conducted to study how big data can be analyzed with Hadoop and Hive such as with [20], [21] and [22]. As [20] states, Hadoop provides a framework for tools to analyze big data. They studied how Hive can be used to manage the data and run queries to analyze the large data sets stored within Hadoop. The case study for [20] fails to mention the type of hardware/resources that were used when analyzing the data and providing their results.

In [20], [21] and [22], Microsoft Azure was used to host a cluster of four data nodes to conduct the data analysis. For [21], 314 MB of NYSE financial data was used and 3.6 GB of airline data for [22]. External applications were used by [21] and [22] to provide a graphical representation of the data analysis results. In [22], they mentioned using Microsoft Windows

Server 2012 R2 Datacenter as their operating system. For SMEs to implement a solution with such an operating system, a great expense will be incurred.

The research conducted in the paper will demonstrate how H2H, which is composed of Hadoop, Hive and Hue, does not have to be run on expensive physical servers nor through a cloud computing service. Instead, SMEs can run H2H on a commodity laptop with appropriate specifications to perform data analysis. This provides SMEs with a cost effective alternative to server clusters which can cost thousands of dollars to implement and involves a more complex implementation. H2H also provides a graphical user interface when performing the data analysis thus making the process of data analysis less intimidating and more user friendly than a command line interface. Through H2H, a graphical representation of the data analysis results is provided thus there is no need to export the raw data to an external application to obtain a visual graphical representation of the results.

3. H2H ANALYSIS TOOL

Other components are used in combination with Hadoop to aid with the data analysis process. Apache Hive is a data warehouse infrastructure that resides on top of Hadoop to provide data summarization and SQL-like querying with HiveQL [18]. With an SQL interface, users are spared to have to code tedious and occasionally hard MapReduce programs to manipulate the data stored in the HDFS. Hive was originally created for Facebook but was later provided to the open source community [19]. Hive allows for extract/transform/load (ETL), reporting, and data analysis jobs [2]. Hive also provides a way to enforce structure on a range of different data formats.

Hue is used to provide an interactive graphical user interface to visualize and share data that is queried from within the Hadoop stack. The combination of Hadoop with Hue and Hive yield the H2H data analysis tool. This research demonstrates how SMEs can use H2H on a commodity laptop as a method to perform data analysis. Implementing H2H on a commodity laptop provides a low cost startup alternative for SMEs to start analyzing their big data.

4. HADOOP ARCHITECTURE

Apache Hadoop provides a scalable, flexible and reliable distributed computing framework for big data intended for a cluster of systems with storage volume and native computing power using commodity hardware [1]. A computer cluster is generally known as a collection of multiple computers that function together as a single system. A Hadoop cluster is known as a computational computer cluster of a collection of independent components that process data in parallel to store and analyze big data through a dedicated network in a distributed environment [3]. Hadoop follows a Master Slave architecture for the transformation and analysis of large datasets through the Hadoop MapReduce paradigm. The Master manages, maintains and monitors the slaves, whereas slaves are the real worker nodes. The three core components of the Hadoop architecture are Hadoop Distributed File System (HDFS), MapReduce and Yet Another Resource Negotiator (YARN) [4]. HDFS is used for data storage, MapReduce provides the distributed data processing, and YARN delivers a workload scheduling in addition to a resource management facility which in turn allows Hadoop to perform more than just MapReduce tasks. The concept behind the MapReduce model is that process is done in two stages. During the Map stage, data is processed locally and during the Reduce stage, the results are consolidated. This concept allows for moving the computation to the data instead of the data to computation. The primary components for Hadoop are depicted in figure 1 with the data storage and data computation layers. The computation layer consists of one ResourceManager per cluster and one NodeManagers per slave node. The responsibility of the ResourceManager is to track the resources within a cluster and schedule jobs such as MapReduce jobs.

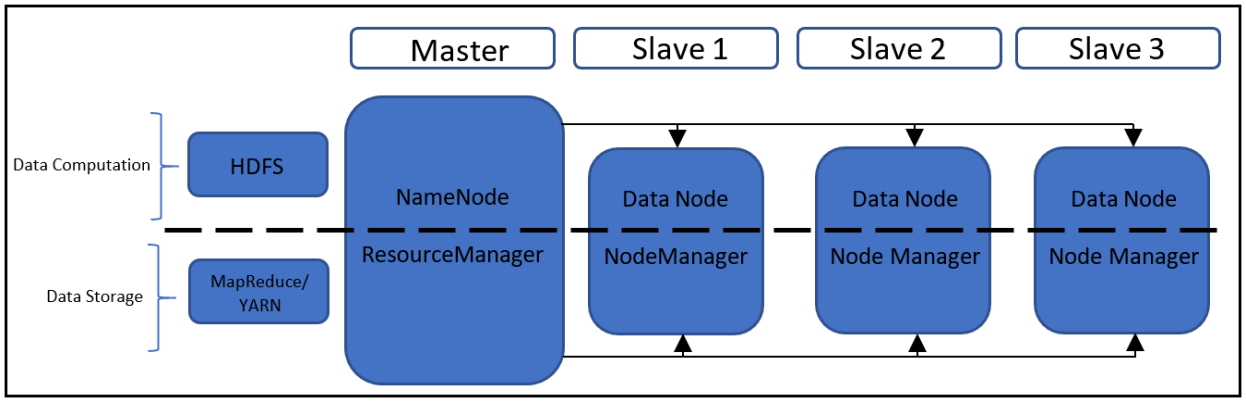

Figure 1 Hadoop Components

4.1 Hadoop Distributed File System (HDFS)

Hadoop clusters are known as a share-nothing architecture because the only thing that connects them together is the network between them. TCP based protocols are used to communicate between the nodes. This share-nothing type of architecture reduces the processing latency that would be required for large amounts of data. Hadoop also follows a master/slave architecture for the conversion and analysis of big datasets. An HDFS cluster contains one NameNode (master) and one or more DataNodes (slaves) which are vital components of the Hadoop HDFS architecture. NameNodes manage file system namespace and control access to files through clients and DataNodes manage storage attached to the nodes that they run on. HDFS provides a file system namespace and allows for the storage of user data. From an internal perspective, a file stored on HDFS is fragmented into blocks that are stored in DataNodes and replicated within the Hadoop cluster. A block on HDFS is the minimum size of data that can be read/written from the disk; default is 128MB [4]. In comparison, Linux only has a 4KB data block size and such a small size for Hadoop's block would create a tremendous overhead and increased traffic. On the other hand, a large block would cause the system to wait a long time for

the last unit of the data to finish processing its task. HDFS replicates the contents of a file to several DataNodes based on the replication factor to guarantee data reliability. Each data block is replicated by a replication factor that is defaulted to three in Hadoop. Based on this replication factor, the data is replicated on other data nodes. Both block size and replication factor can be configured per block within Hadoop [4]. The NameNode is responsible for the file system namespace operations to files and directories as well as defining the mapping between blocks and DataNodes. The NameNode stores the file system metadata in an FsImage file on its hard disk and controls file access. Changes to the file system metadata (e.g. create new file, change replication factor) are stored in the EditLog and it stored on the NameNode's local disk. The FsImage is loaded into RAM when a NameNode starts and the transactions log in the EditLog are applied. A new persistent FsImage file is created and the old EditLog is deleted thus creating a checkpoint. The DataNodes server read and write requests from the file system's clients and creates, deletes and replicates block as requested by the NameNode. In comparison to traditional DFS, Hadoop implements the concept of data localization where only KB size code is transferred over the network, instead of TB of data, to be processed.

4.1.1 File Read and Write in HDFS

HDFS implements a Write-once-Read-many model so existing files cannot be edited but data can be appended by reopening the file. The client sends a request to the NameNode to read/write a data block. The NameNode will determine if the client has the appropriate permissions. If so, the read/write operation will proceed.

4.1.1.1 Data Read Operation

The client makes a request to the NameNode to obtain the list of DataNodes in which the file it is requesting to read is replicated at. The NameNode will provide the list which is ordered by distance to the client. The client then contacts the first DataNode on the list for each block and reads all the blocks in order. A checksum is also provided to the client to ensure the data hasn't been tampered with or is corrupted. If there is an issue with the DataNode, then the client makes the request to the next DataNode on the list for the replica. The failed DataNodes are not further contacted for this read request [4].

4.1.1.2 Data Write Operation

The client requests the NameNode to create a new file. The NameNode will first check if the client has the appropriate permissions and if so, will grant the right to write the file. The client will request a list of data nodes to which it will use to store the replicas of the block. The NameNode will return a unique block id along with a list of DataNode addresses. With this information, the DataNode will establish a connection through TCP and send the data as a sequence of packets. The client pushes the packets to the first DataNode on the list and it will forward the data to the following DataNode in the pipeline. In addition to the data, a checksum for each block is also provided to the DataNodes which is stored in the metadata file. Upon receiving a packet, an acknowledgement is sent back to confirm receipt. This process is repeated until all the blocks of the file have been written [4].

4.2 MapReduce (MPv2)

MapReduce [5] is a software framework and one of the core Hadoop ecosystem components that provides data processing. MapReduce offers a method to develop applications

to process most of structured and unstructured data that is stored in the HDFS. MapReduce programs execute in parallel which provides improved speed and reliability especially when executing large-scale data analysis using multiple machines in a cluster. The MapReduce component is composed of two phases: Map phase and the Reduce phase. Each phase possesses key-value pairs as input and out. The Map function divides the initial input data into smaller independent pieces and process the Map tasks in parallel. The Reduce function processes the output from the Map function to yield the result. MapReduce handles failures due to the replication factor. If data becomes unavailable, then it looks for its replica which will contain the same key pair thus allowing the subtask to be solved. The execution and monitoring of the tasks are managed by the framework itself. The figure below illustrates the MapReduce data flow.

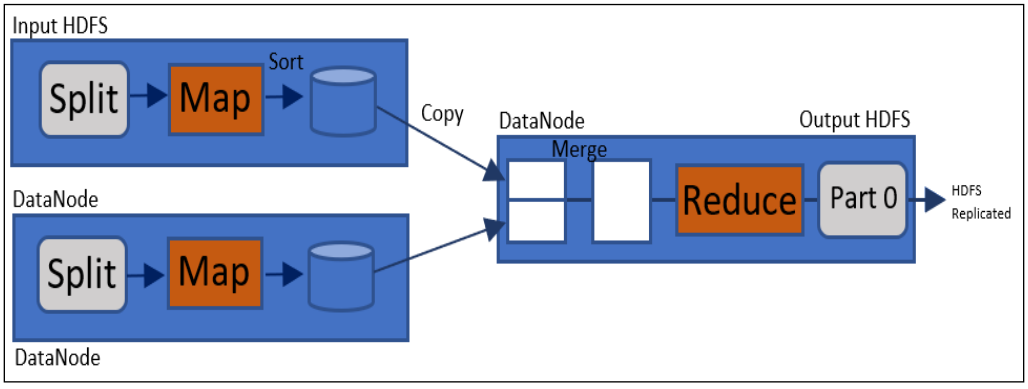

Figure 2 MapReduce Data Flow

Data is stored on HDFS and split based on the configured block size. Processing starts when the Map receives the block and its output is written to the local disk of the machine running the Mapper. When data shuffling from the Mapper completes processing, the out is

provided to the Reducer node where the second stage of processing begins. The output generated by the Reducer is the final out that will be written to the HDFS.

4.3 YARN

YARN [7] is the default resource manager for Apache Hadoop cluster. The idea is to separate the resource management from the job scheduling/monitoring. Resource management becomes more import when using a multi-node cluster as its complexity increases significantly to manage, allocate and release system resources (e.g. memory, CPU and disk) [7]. The ResourceManager is ultimately in charge of managing and allocating cluster resources. The NodeManager is per node agent and manages and enforces node resource allocation. The ApplicationMaster is per-application and manages the application lifecycle and task scheduling. Containers can run different types of tasks and vary in sizes of RAM, CPU, etc. The following figure from Apache Hadoop illustrates the YARN framework.

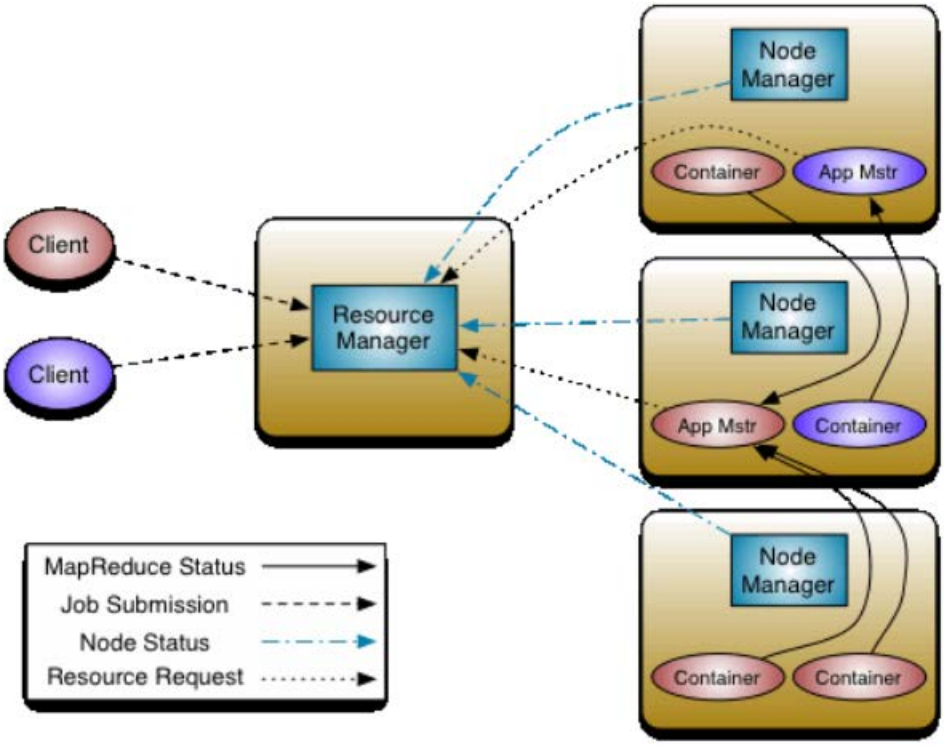

Figure 3 YARN Framework

5. RESEARCH ENVIRONMENT

The environment to conduct the research consisted of a Cloudera Hadoop cluster containing three nodes running on a commodity laptop. A Type 2 hypervisor is where hypervisors run on a host operating system and deliver virtualization services. This type was used to run H2H on a commodity laptop. The resources for the laptop were 16 GB of memory, 1 TB of storage and an Intel Core i7-7700HQ CPU at 2.8GHz. A substantial amount of time was dedicated to the setup of the test environment due to lack of documentation for a Hadoop cluster deployment within a virtualized environment using VirtualBox. This resulted in numerous failed attempts due to the variability of the configuration. The process of installing and configuring the environment are detailed within this section to allow for future related work to be conducted.

Core Hadoop CDH5 services that were installed for the test environment were: HDFS, YARN, ZooKeeper, Oozie, Hive and Hue. HDFS is the distributed file system used by Hadoop. YARN is the framework that will maintain track of the resources, submit job to the cluster, execute jobs, and log progress. Map Reduce is API that requires data processing implementation. Zookeeper [14] is a service that sustains configuration information, naming, providing distributed synchronization, and providing group services. OOZIE facilitates the creation of a workflow to execute complete data integration processes. Hue is the location where all Hadoop ecosystem tools are accessible from one location. Figure 4 depicts the Hadoop cluster role deployment used within this research.

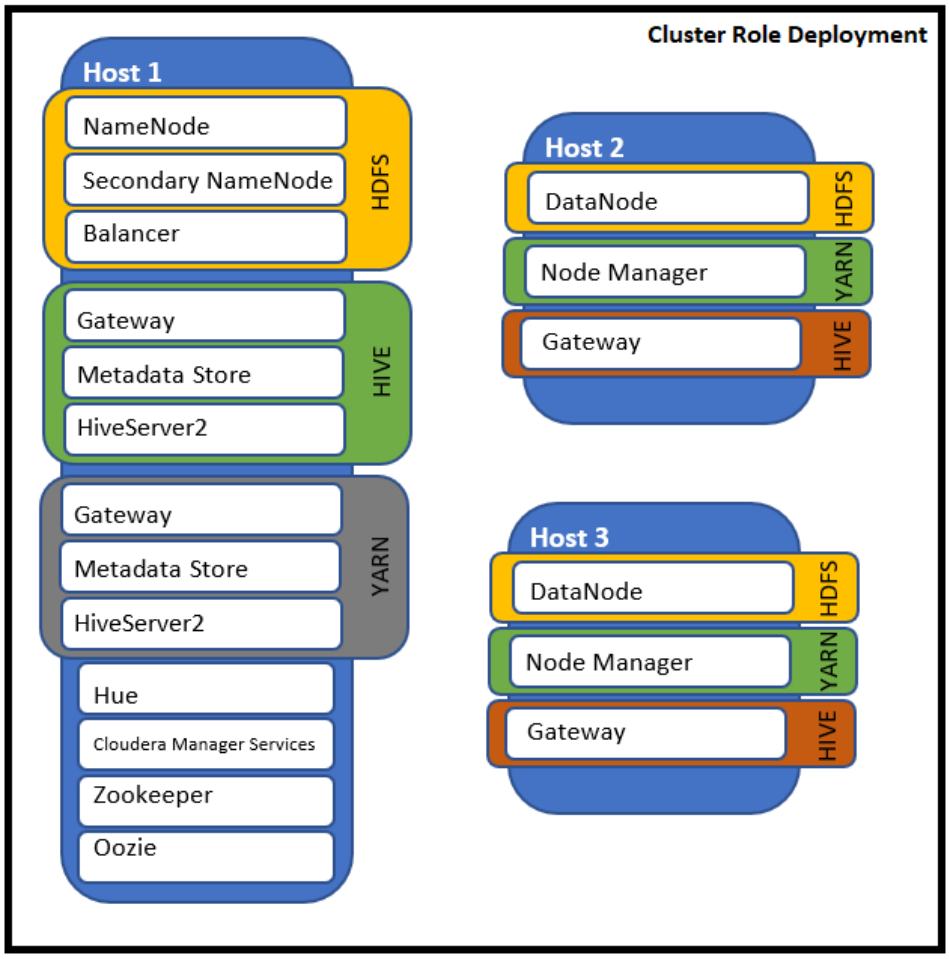

Figure 4 Hadoop Cluster Deployment

Note: Cloudera Manager defaults its replication factor to three but due to the limitation of the available resources, the replication factor was changed to 2.

5.1 Server Preparation

The deployment of the Cloudera Hadoop cluster is composed of three servers. One server was used to host Cloudera Manager and its Hadoop Services (4 CPU, 10 GB RAM, and 100 GB storage) and the other two servers were data nodes (1 CPU, 2 GB RAM, and 100 GB storage).

According to the Cloudera Manager documentation [6], certain requirements are advised to be configured on the servers for a successful install. Network Time Protocol service must be configured on each host within the cluster to ensure time synchronization between hosts. Otherwise, warnings may be generated due to the time differences. Host names must be defined on each server to ensure the members of each host can communicate with each other. Each host file on each server /etc/hosts) must contain the IP addresses and fully qualified domain names of all hosts within the cluster. Security-Enhanced Linux (SELinux) is a system property that allows you to set access control through policies. This setting must be set to permissive to avoid running into issues with any existing policies. The firewall is also recommended to be disabled during install.

The Linux kernel parameter, vm.swappiness [6], must be set to a value between 0-10. This setting controls the swapping of application data from physical memory to virtual memory on disk. When the parameter is set to a high value (max is 100), the more aggressively inactive processes are swapped out from physical memory. With a lower value, fewer processes are swapped out of physical memory thus, making filesystem buffers to be emptied. Most systems are configured to 60 by default which is not adequate for Hadoop clusters because processes are occasionally swapped even when sufficient memory is available. This can affect the stability and performance through extensive garbage collection pauses for important system daemons. Finally, the Transparent Huge Page Compaction setting must be disabled [6]. It’s a Linux memory management system that decreases the overhead of Translation Lookaside Buffer (TLB) lookups on systems with big quantities of memory with larger memory pages.

Cloudera Manager [6] utilizes several databases to store information regarding the Cloudera Manager configuration in addition to information about the health of the system or task progress. The Cloudera Manager database contains all the information about services that are configured along with their role assignments, all configuration history, commands, users, and running processes thus making it the most important component to ensure it is backed up regularly.

- Oozie database: contains Oozie workflow, coordinator, and bundle data.
- Activity Monitor database: holds information about past activities.
- Reports Manager Database: tracks disk utilization and processing activities.
- Hive Metastore Server database: contains Hive metadata.
- Hue Server database: contains user account information, job submissions, and Hive queries.
- Cloudera Navigator Audit Server database: holds auditing information and the Cloudera Navigator Metadata Server contains authorization, policies, and audit report metadata.

For this research, some of these databases were consolidated into one yielding a Hive (metastore) database, Hue database, Oozie database and a Cloudera Manager Server (SCM) database which is utilized by the rest of the services.

5.2 Installing Cloudera Manager and CDH

Several phases are required for the installation of Cloudera Manager to deploy a multi cluster Hadoop system.

- Phase 1: requires the installation of JDK which is used by the Cloudera Manager Server, Management Service, and CDH.
- Phase 2: configures the databases previously created.
- Phase 3: installs the Cloudera Manager Server.
- Phase 4: installs the Cloudera Manager Agents on all nodes within the Hadoop cluster.
- Phase 5: installs, configures, and starts CDH and managed services on all hosts.
- Phase 6: configures and starts CDH and managed services.

Once all the phases are completed, Cloudera Manager can be access through a web browser to manage and monitor the cluster [6].

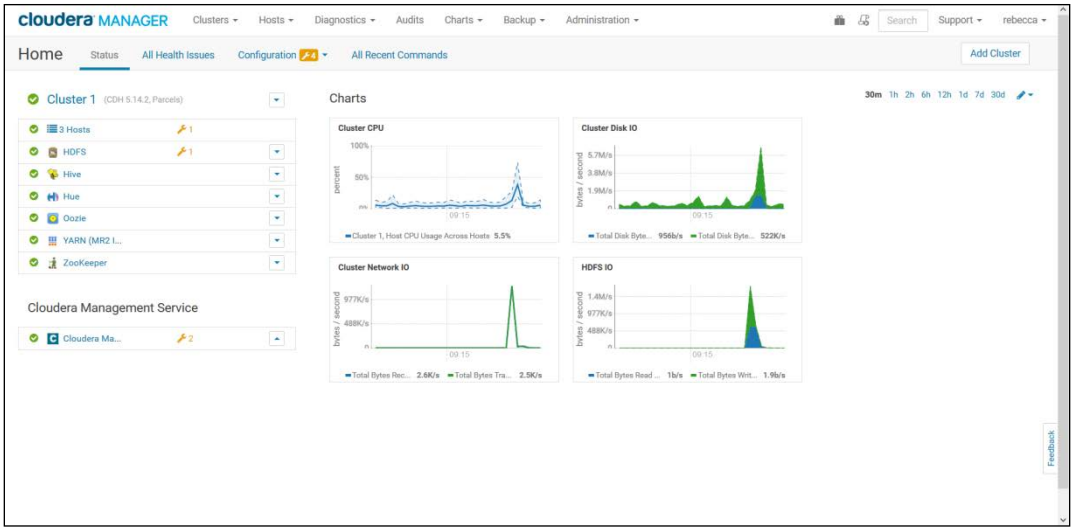

Figure 5 Cloudera Manager

6. TEST SETUP

H2H provides an analytics workbench, a data warehouse and an editor to perform data analysis on Amazon's Product Review data. The workbench is provided by Hue to provide a web interface for analyzing data with Hadoop . The data warehouse is provided by Hive which runs on top of Hadoop and uses a query language similar to SQL. Hive allows for easy data summarization and ad hoc querying and analysis of large volumes of data. HiveQL is the query language used by Hive where the statements are automatically converted into MapReduce jobs to process the requests. This research demonstrates how H2H can be used to analyze Amazon product review data sets running on a commodity laptop. Different data set sizes are utilized to determine the effectiveness of analyzing data with H2H on a commodity laptop are identified.

The Amazon Product Review data sets can be analyzed with H2H by generating HiveQL queries to retrieve valuable information from the data. H2H provides an editor where HiveQL queries are ran against the data. In addition, H2H also generates the results of the query in a graphical representation thus allowing the user to see the data results in a graphical representation.

For any SMEs, the key to selling the right services and products is vital to its success. Amazon sells millions of products through their website and customers have the options to write a product review as to whether they were satisfied with the product. For some customers, those reviews play a role when making the decision to purchase the item or not. Amazon may notice a decrease in sales for a specific product which likely correlates to the increase of negative reviews. It's vital for Amazon to track this kind of information. On January 10, 2018, it's was reported that Amazon had 562,382,292 products for sell [13]. Each one of those products may

potentially have one or more reviews associated with it. Processing and analyzing the information from these reviews could yield valuable information that Amazon may use when making future business decisions. Using H2H to store and analyzing the data is a viable solution and is demonstrated through the data analysis conducted on the Amazon Product Review data sets.

6.1 Amazon Product Review Data

H2H will be used to analyze the Amazon Product Review data sets. The data sets were obtained from the University of California San Diego in a JavaScript Object Notation (JSON) file format. JSON is an open-standard file format that provides a method of to transmit object in a human-readable format. The objects consist of attribute-value pairs (e.g. “course”:“CSCI698”). Each of the Amazon product review data sets contain the following information: reviewer’s ID, product ID, reviewer name, helpfulness rating of the review, text of the review, overall rating of the product, summary of the review, time of the review (raw and Unix time). Below is a sample entry for an entry in the review data set.

Sample review:


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the
piano.  He is having a wonderful time playing these old hymns.
The music  is at times hard to read because we think the book
was published for singing from more than playing from.  Great
purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


where

- `reviewerID` - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- `asin` - ID of the product, e.g. 0000013714
- `reviewerName` - name of the reviewer
- `helpful` - helpfulness rating of the review, e.g. 2/3
- `reviewText` - text of the review
- `overall` - rating of the product
- `summary` - summary of the review
- `unixReviewTime` - time of the review (unix time)
- `reviewTime` - time of the review (raw)

Figure 6 Amazon Product Review sample

The Amazon Product Review data sets that are used within this research are listed in the table below along with the number of reviews per category and their file size (MB). A total of 15.65 GB of Amazon data was analyzed using H2H on a commodity laptop.

Table 1 Amazon Product Review data sets

Amazon Product Review Data Sets		
Category	**# Reviews**	**Data Size (MB)**
Amazon Instant Video	37,126	27.45
Apps for Android	752,937	329.483
Automotive	20,473	13.943
Baby	160,792	118.529
Beauty	198,502	137.493
Books	8,898,041	9,236.30
CDs and Vinyl	1,097,592	1,333.50
Cell Phones and Accessories	194,439	138.37
Clothing Shoes and Jewelry	278,677	149.564
Electronics	1,689,188	1,444.30
Grocery and Gourmet Food	151,254	111.112
Health and Personal Care	346,355	255.587
Home and Kitchen	551,682	411.435
Kindle Store	982,619	808.404
Movies and TV	1,697,533	1,938.90
Musical Instruments	10,261	7.272

Office Products	53,258	54.603
Patio Lawn and Garden	13,272	14.204
Pet Supplies	157,836	108.017
Sports and Outdoors	296,337	203.199
Tools and Home Improvement	134,476	119.742
Toys and Games	167,597	126.67
Video Games	231,780	311.986
Total	**18,122,027**	**15654.109**

H2H was used to obtain the following information when performing data analysis of the Amazon Product Review data:

- Products by Category with the most positive reviews
- Top 10 positive reviewers for the Amazon Instant Video category
- Top 25 Products within the CDs and Vinyl category with the largest number of positive reviews
- Top 10 Products with the most number of lowratings in the Office Products category
- Top reviewer per category
- Top 10 Reviewers for the Home and Kitchen category
- Select all the reviews created by the ‘The Movie Guys Movies from A to Z’ for the Movies and TV category.

Different data set sizes will be used to evaluate the performance on H2H on the commodity laptop.

6.2 Loading and Parsing Amazon Product Review Data

To perform data analysis on the Amazon Review data sets, the data must be stored on HDFS. Hue provides a web interface to browse and manipulate files and directories in HDFS. The following shows the Amazon Product Review data sets initially stored on the HDFS.

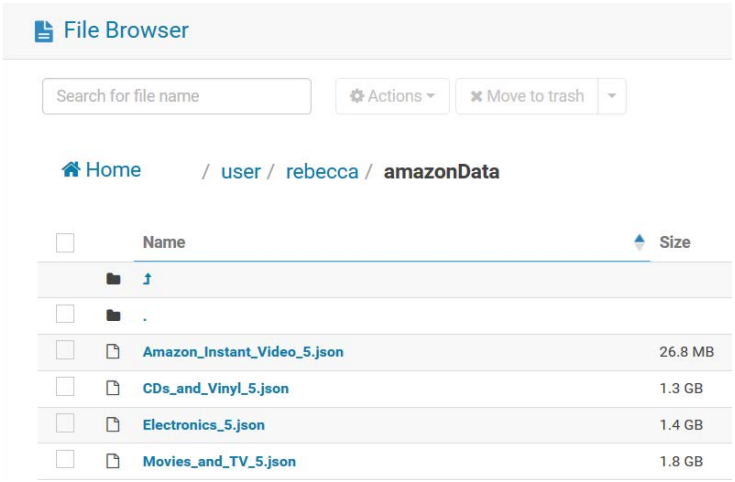

Figure 7 Hue File Browser - Amazon Review Product Data

Each of the Amazon Product Review data sets are parsed and stored in a table with the required fields and types prior to performing any data analysis. Two initial tables are created for each product category. One table will hold the JSON input file in string format and the other table will hold specific data from the parsed JSON file. The figure below depicts the tables created after running the queries to create the tables.

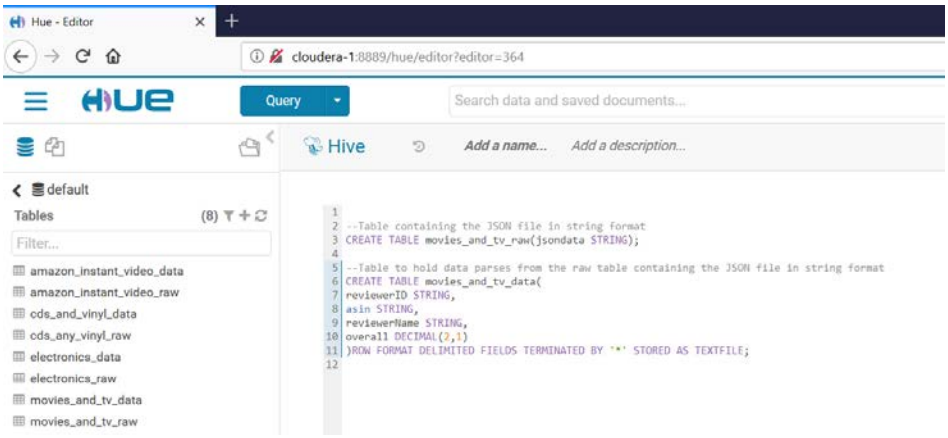

Figure 8 Amazon Product Review Database Tables

Data can be stored in internal or external tables. With internal tables, H2H completely manages the life-cycle of the data within the table. When the data is loaded, the data file is moved from its initial directory to the warehouse directory. With internal tables, if the table is dropped from the database then the data file is deleted as well. On the other hand, with external tables, the data remains even after the table is dropped. For this research, the data was stored in internal tables (tables with ending '_raw' table names). The following query was used to load the JSON file into an internal table as a string.

```
1 LOAD DATA INPATH '/user/rebecca/amazonData/CDs_and_Vinyl_5.json' INTO TABLE cds_and_vinyl_raw;
```

Figure 9 Query to load JSON data

A sample of the imported table can be viewed through the workbench to ensure the JSON objects were imported correctly. Each row from the table equals one JSON object.

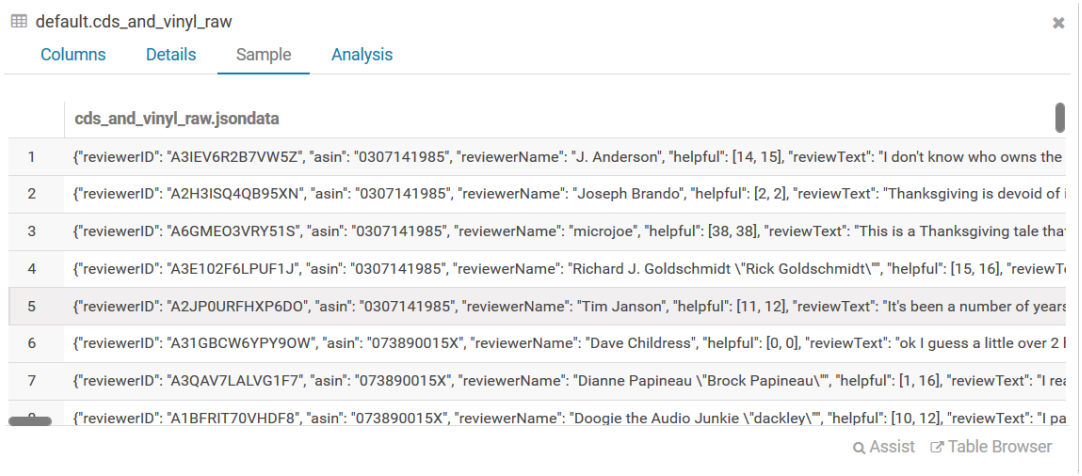

Figure 10 Sample of Electronics raw table

The JSON string in the table is parsed into the necessary fields for the data analysis and loaded into its respective table. The following figure is a sample of the table after parsing and storing the JSON string.

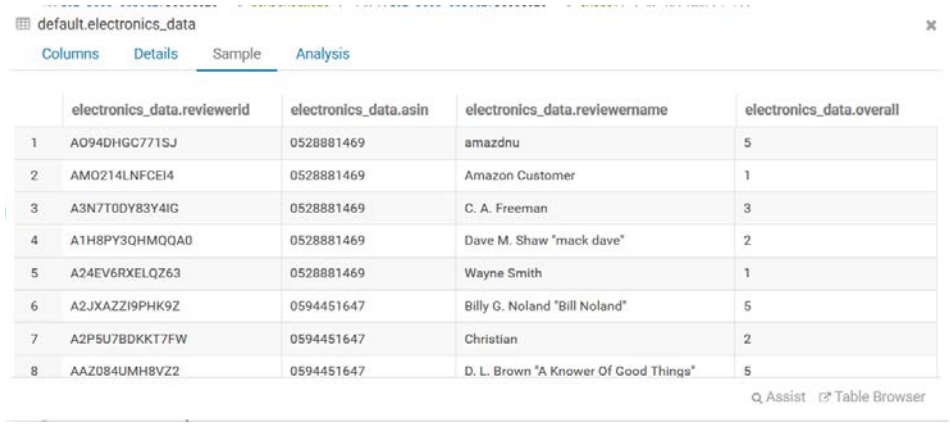

Figure 11 Table with parsed data

7. RESULTS

The following sections provide the metrics with regards to how H2H performed on a commodity laptop to analyze Amazon Product Review data sets.

7.1 Performance metrics

Performance metrices were captured for the data analysis conducted on the Amazon Product Review data sets. Metrics were captured based on the number of Amazon Product Reviews within a data set. The following metrics were captured to measure the performance:

- Number of Review – quantity of reviews within a data set
- File size – file size of the data set in megabytes
- Cluster CPU – host CPU usage across all hosts.
- Cluster Disk IO - total disk bytes written across disks.
- Cluster Network IO - total bytes transmitted across network interfaces.
- HDFS IO - total bytes read across dataNodes.
- Completed Execution Time (sec) – time to execute the query

Different data set sizes were ran to evaluate the performance on H2H on a commodity laptop. Additional performance metrics were captured based on the type of Hive SQL query that was ran. Table 2 provides the performance metrics captured when performing data analyzes on the Amazon Product Review data sets. From table 2, one can observe that as the size of the data set increases, so do the values for the majority of the performance metrics. This is due to the increase in the size of the data set which requires more processing time and resources to analyze the data. The query to retrieve the products by category with the most positive reviews contained 15.65 GB of data and H2H produced a resulted in 29 minutes and 10 seconds.

Table 2 Performance Metrics for difference sizes of Amazon Product Review data sets

H2H Performance Metrics Analyzing Amazon Data											
Amazon Category	**Number of Reviews**	**Data Size (MB)**	**Cluster CPU** Across Hosts (%)			**Cluster Disk IO** Written (MiB/sec)	**Cluster Network IO** Across Network Interfaces (MiB/sec)		**HDFS IO** Across Data Nodes (MiB/sec)		**Completed Execution Time (sec)**
			Max	Mean	Min		Transmitted	Received	Written	Read	
Products by Category with the most positive reviews	18,122,027	15654	87.9	32.1	3.7	7.6	3.1	1.9	1.9	3.7	1750
Top 10 positive reviewers for the Amazon Instant Video category	37,126	27.5	26.1	9.87	1.7	6.4	2.2	1.9	1.7	1.8	66.44
Top 25 Products within the CDs and Vinyl category with the largest number of positive reviews	1,097,592	1,333.5	43.7	22.03	4.2	10.4	1.8	1.5	1.8	2.5	73
Top 10 Products with the most number of lowratings in the Office Products category	53,258	54,603	20.1	10.97	1.3	5.1	1.9	1.4	1.8	1.8	63
Top 10 Reviewers for	551,682	411.4	38.4	24	5.6	6.4	2.4	1.9	1.8	2.2	66

the Home and Kitchen											
Select all the reviews created by the 'The Movie Guys Movies from A to Z' for the Movies and TV category.	1,697,533	1,938.9	29.1	23.67	14.5	6.9	1.8	1.2	1.2	2.4	69

The amount of data that is processed with H2H will have an impact on the time it takes to produce a result. To test H2H's processing time, a HiveQL query was ran for every Amazon Product Review category. Table 3 provides a graph with the relation between size of the data and the processing time when analyzing the Amazon Product Review data with H2H to identify which product has the most reviews. The HiveQL query used is provided in figure 12.

```
1 SELECT 'Amazon Instant Video', asin AS `Product ID`, COUNT(asin) AS `Num Reviews`
2 FROM amazon_instant_video_data
3 GROUP BY asin
4 Order BY `Num Reviews` DESC
5 LIMIT 1;
```

Figure 12 HiveQL for product with most reviews

Within the table, you can see that average processing time regardless of the data size, is around one minute. The longest processing time was for the book category as it would be expected since it has the largest file size. These times will vary depending on the query, available resources for each node and the network load. The reason for the constant value of one minute for running the query in figure 12 on the majority of the data sets is that the data required for the query is all in one table instead of multiple tables. This wasn't the case for the majority of the queries ran to obtain the information in table 2. There is also a certain amount of overhead time that is the same for each data set being processed regardless of its size. The query needs to be compiled and the command executed to determine the number of jobs, splits and estimated reducers that are necessary to run the query on the specified data size.

Table 3 Processing time versus data size

Item Most Reviewed Per Category		
Category	**Data Size (MB)**	**Time (sec)**
Muscial Instruments	7.2	63
Automotive	13.943	63
Patio Lawn and Garden	14.204	63
Amazon Instant Video	27.45	63
Office Products	54.603	63
Pet Supplies	108.017	63
Grocery and Gourmet Food	111.112	63
Baby	118.529	63
Tools and Home Improvement	119.742	63
Toys and Games	126.67	63
Beauty	137.493	63
Cell Phones and Accessories	138.37	63
Clothing Shoes and Jewelry	149.564	64
Sports and Outdoors	203.199	63
Health and Personal Care	255.587	64
Video Games	311.986	64
Apps for Android	329.483	64
Home and Kitchen	411.435	63
Kindle Store	808.404	77
CDs and Vinyl	1,333.50	67
Electronics	1,444.30	76
Movies and TV	1,938.90	75
Books	9,236.30	184

7.2 Data Analysis HiveQLs

The data analysis performed with H2H used HiveQL queries. This section provides some of the queries that were used on the Amazon Product Review data sets to retrieve specific information from the data sets and the value it could bring to retrieve such information.

H2H was used to determine which item per category was reviewed the most whether it was a positive or negative review.

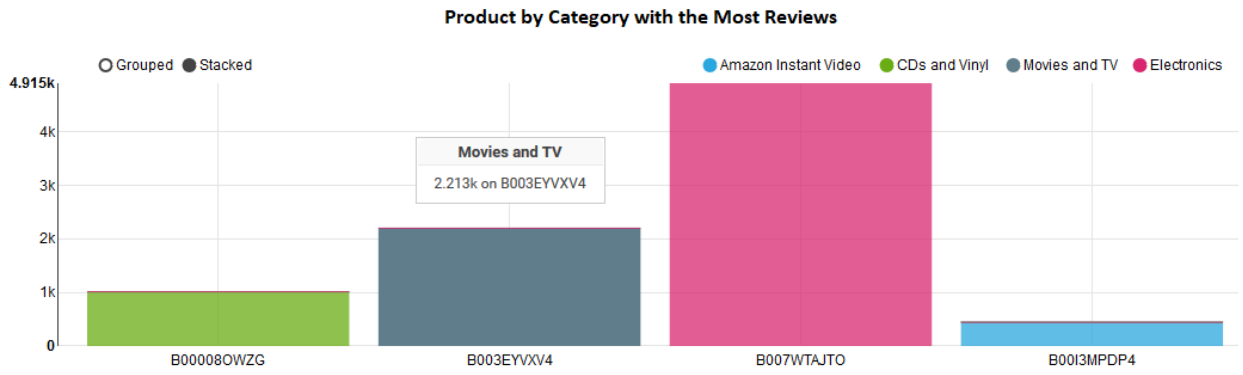

Figure 13 Products per Category with the Most Reviews

This information along with other information associated with the product (e.g. sales, cost, etc.) could provide Amazon with better insight to make business negotiations with its supplier.

It may also be vital for Amazon to find out which items have the most positive reviews. Those products with an overall score of greater than three are classified as a positive review. Figure 13 below, provides the product within each category with the most positive reviews.

Product by Category with the Most Positive Reviews (overall >=3)

	_u1.category	_u1.product id	_u1.num reviews
1	Amazon Instant Video	B00I3MPDP4	417
2	CDs and Vinyl	B000000IRB	803
3	Movies and TV	B001KVZ6HK	2009
4	Electronics	B007WTAJTO	4587

Figure 14 Products per Category with the Most Positive Reviews

Figure 14 provides the HiveQL to display the results of the top 10 products within the Amazon Instant Video category with the most number of positive reviews.

```
SELECT 'Amazon Instant Video', asin AS `Product ID`, COUNT(asin) AS `Num Reviews`
FROM amazon_instant_video_data
WHERE overall >=3
GROUP BY asin
Order BY `Num Reviews` DESC
LIMIT 10;
```

Figure 15 HiveQL for Top 10 Positive Reviews for Amazon Instant Video Products

The information from determining which products have the largest number of positive reviews can be used, in addition to other product information (e.g. sales data), to make better business decisions with the supplier. It may reveal that the supply on those products may need to be increased. Figure 15 provides a graphical representation of the results for the top 10 products sold within the CDs and Vinyl category with the most positive reviews (overall >=3).

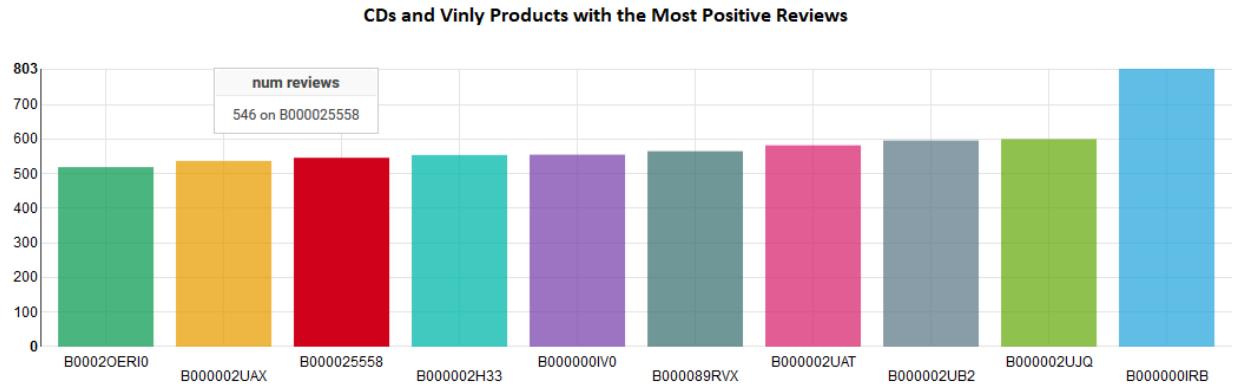

Figure 16 CDs and Vinyl Products with the Most Positive Reviews

Negative reviews can also provide valuable insight as to why a product is negatively rated by its customers or what potential improvements could be made, especially since Amazon sells its own products. The reviews could also be utilized by the supplier to improve on the product. These negative reviews could potentially provide Amazon with information necessary to decide in potentially reducing or cancelling future purchases from the supplier. Items with the most number of negative reviews per category can also be yielded (negative reviews are those with an overall rating of below 3).

Low Rating Products by Category (Overall <3)

	category	product id	review ct
1	Electronics	B00DR0PDNE	558
2	CDs and Vinyl	B00008OWZG	557
3	Movies and TV	B0001VL0K2	425
4	Amazon Instant Video	B00I3MMTS8	121

Figure 17 Low Ratings Products by Category

Not only can products be analyzed based on their reviews, but analysis on the reviewers can be performed as well. Through the data analysis, it may be beneficial to determine who the top reviewers are within each category. Amazon may want to give incentives to these reviewers for taking the time to review their products. Figure 18 provides the HiveQL query to obtain the top reviewers per category.

```
--Reviewer with the Most Reviews

SELECT 'Amazon Instant Video' AS `Category`, reviewerid, reviewername,
       COUNT(asin) AS `Num Reviews`
FROM amazon_instant_video_data
GROUP BY reviewerid, reviewername
ORDER BY `Num Reviews` DESC
LIMIT 1
UNION ALL
SELECT 'CDs and Vinyl' AS `Category`, reviewerid, reviewername,
       COUNT(asin) AS `Num Reviews`
FROM cds_and_vinyl_data
GROUP BY reviewerid, reviewername
ORDER BY `Num Reviews` DESC
LIMIT 1
UNION ALL
SELECT 'Electronics' AS `Category`, reviewerid, reviewername,
       COUNT(asin) AS `Num Reviews`
FROM electronics_data
GROUP BY reviewerid, reviewername
ORDER BY `Num Reviews` DESC
LIMIT 1
UNION ALL
SELECT 'Movies and TV' AS `Category`, reviewerid, reviewername,
       COUNT(asin) AS `Num Reviews`
FROM movies_and_tv_data
GROUP BY reviewerid,reviewername
ORDER BY `Num Reviews` DESC
LIMIT 1;
```

Top Reviewers Per Category

Query History | Saved Queries | Results (4+)

	_u1.category	_u1.reviewerid	_u1.reviewername	_u1.num reviews
1	Amazon Instant Video	AV6QDP8Q0ONK4	The Movie Guy "Movies from A to Z"	123
2	CDs and Vinyl	A9Q28YTLYREO7	mistermaxxx08 "mistermaxxx08"	3571
3	Electronics	ADLVFFE4VBT8	A. Dent "Aragorn"	427
4	Movies and TV	A3LZGLA88K0LA0	Michael Butts	2362

Figure 18 Top Reviewers per Category

The HiveQL query to retrieve the top 10 reviewers within the Amazon Instant Video category is displayed in the below figure along with a table containing the results.

Top Reviewers in Amazon Instant Video

	category	reviewerid	reviewername	num reviews
1	Amazon Instant Video	AV6QDP8Q0ONK4	The Movie Guy "Movies from A to Z"	123
2	Amazon Instant Video	A27H9DOUGY9FOS	K. Harris "Film aficionado"	116
3	Amazon Instant Video	AW3VZ5O895LRK	carol irvin "carol irvin"	104
4	Amazon Instant Video	A1XT8AJB7S9JJG	Tony Heck	88
5	Amazon Instant Video	A16XRPF40679KG	Michael Dobey	67
6	Amazon Instant Video	ABO2ZI2Y5DQ9T	Tsuyoshi	58
7	Amazon Instant Video	A328S9RN3U5M68	Grady Harp	56
8	Amazon Instant Video	A2HVL790PBWYTU	H. Bala "Me Too Can Read"	55
9	Amazon Instant Video	A18758S1PUYIDT	Viva	50
10	Amazon Instant Video	A3QLAOOTFEHCJI	M. Oleson	50

Figure 19 Top 10 Reviewers in Amazon Instant Video Category

Further analyze can be performed to the above data regarding the top 10 reviewers within the Amazon Instant Video category. One may want to know how many of the reviews are positive and how many were negative.

Number of Positive Reviews by The Movie Guy "Movies from A to Z"

	reviewerid	reviewername	num positive reviews
1	AV6QDP8Q0ONK4	The Movie Guy "Movies from A to Z"	91

Number of Negative Review by The Movie Guy "Movies from A to Z"

	reviewerid	reviewername	num negative reviews
1	AV6QDP8Q0ONK4	The Movie Guy "Movies from A to Z"	32

Figure 20 Number of Positive and Negative reviews by The Movie Guy

Through this research, it has been proven that H2H can be an effective method to analyze Amazon Product Review data sets on a commodity laptop based on different data set file size.

7.3 Performance based on Query

Cloudera Manager was used to capture the performance metrics for analyzing with H2H. It provides metrics (e.g. CPU and memory) for services and roles running within the cluster. The following figures (figures 21-24) provide the charts with the metrics captured when running two separate HiveQL queries. The first query (Q1) ran against the Amazon Product Review data sets to obtain the top 10 positive reviews for the Amazon Instant Video category. The second query (Q2) was yields the products by category with the most reviews.

Query 1

```
SELECT 'Electronics', asin AS `Product ID`, COUNT(asin) AS `Num Reviews`
FROM electronics_data
GROUP BY asin
Order BY `Num Reviews` DESC
LIMIT 1
UNiON ALL
SELECT 'CDs and Vinyl', asin AS `Product ID`, COUNT(asin) AS `Num Reviews`
FROM cds_and_vinyl_data
GROUP BY asin
Order BY `Num Reviews` DESC
LIMIT 1
UNION ALL
SELECT 'CDs and Vinyl', asin AS `Product ID`, COUNT(asin) AS `Num Reviews`
FROM cds_and_vinyl_data
GROUP BY asin
Order BY `Num Reviews` DESC
LIMIT 1
UNION ALL
SELECT 'Amazon Instant Video', asin AS `Product ID`, COUNT(asin) AS `Num Reviews`
FROM amazon_instant_video_data
GROUP BY asin
Order BY `Num Reviews` DESC
LIMIT 1
UNION ALL
SELECT 'Movies and TV', asin AS `Product ID`, COUNT(asin) AS `Num Reviews`
FROM movies_and_tv_data
GROUP BY asin
Order BY `Num Reviews` DESC
LIMIT 1;
```

Query 2

```
--Top 10 Reviewer in Amazon Instant Video
SELECT reviewerid,
       reviewername,
       COUNT(asin) AS `Num Reviews`
FROM amazon_instant_video_data
WHERE overall >=3
GROUP BY reviewerid,
         reviewername
ORDER BY `Num Reviews` DESC
LIMIT 10;
```

Figure 21 Queries ran for metrics

The Cluster CPU chart provides the Host CPU usage across all hosts. The usage is obtained by running three samples. The minimum CPU usage was 10.6% from host 1; the maximum usage was 56% from host 2. The mean is 31.73% which is represented with the solid blue line within the chart. The small peak was caused by running Q1 and the higher peak was caused by running Q2.

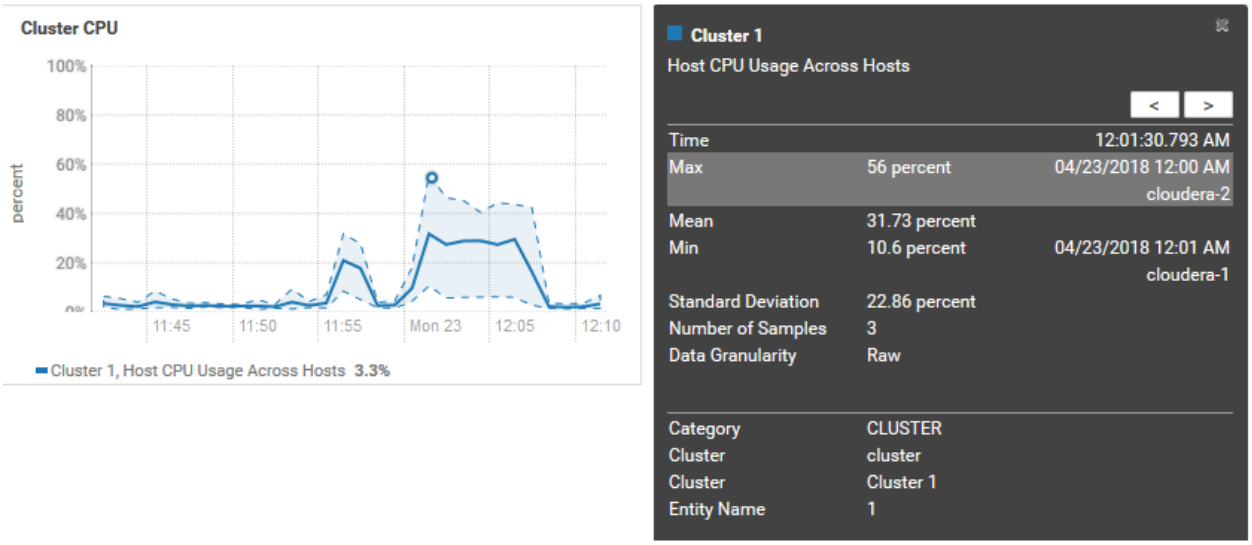

Figure 22 Cluster CPU

The cluster Disk IO captures the total disk bytes written across disks. The highest peak occurred when Q2 was running. During that time, 6.2MB/s were written across disks where the green dot on the chart is located. The lower peak was when Q1 was running which did not require much disk bytes to be written across disks.

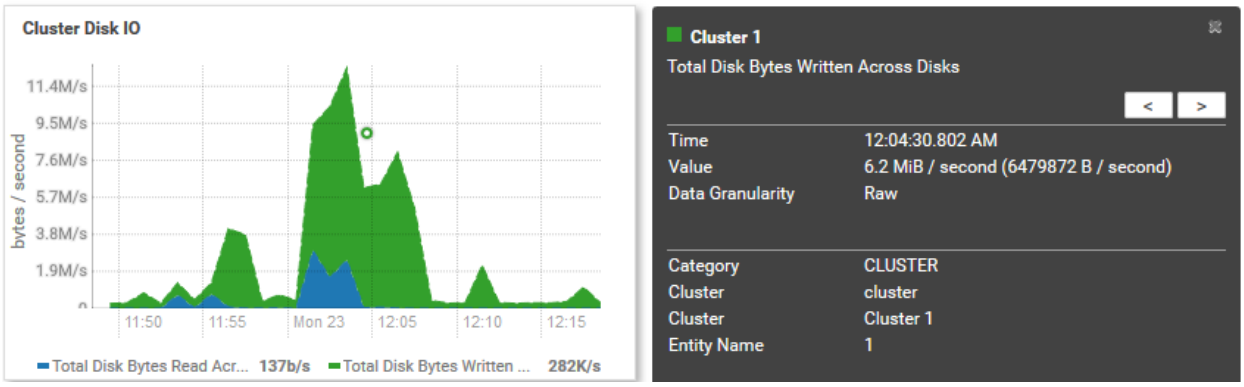

Figure 23 Cluster Disk IO

The cluster Network IO represents the total bytes transmitted across network interfaces. At the first peak which was when Q1 was running, the total bytes transmitted across network interfaces was 1.8MB/sec within the cluster. When Q2 ran, the peaks also averaged out at 1.9MB/sec.

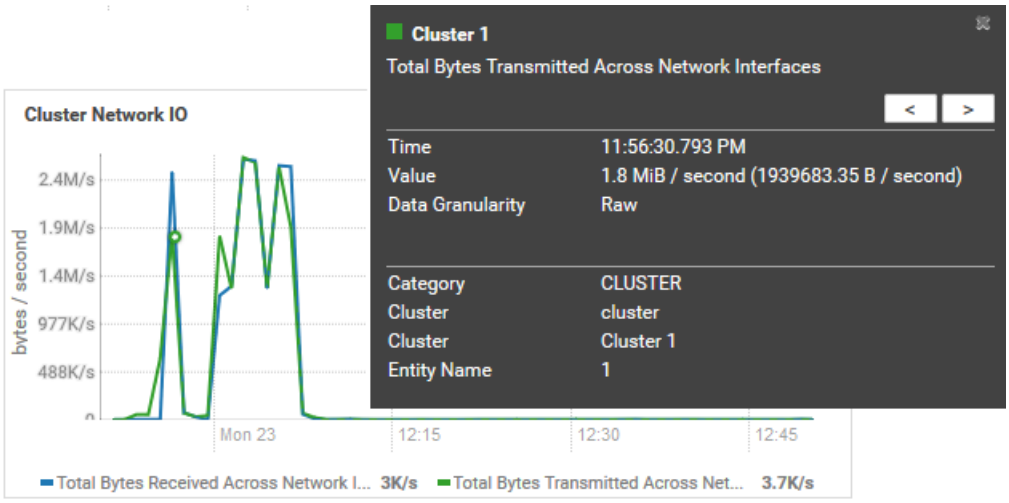

Figure 24 Cluster Network IO

Another metric that was captured during the execution of the queries was the HDFS IO. This metric captures the total bytes read across dataNodes. When the queries start, MapReduce will analyze and process the data. Once completed, the results are stored in the cluster, resulting in an HDFS write. An HDFS read occurs when the results are read from HDFS. The bytes read across dataNodes during the execution of Q2 were 2.5MB/sec.

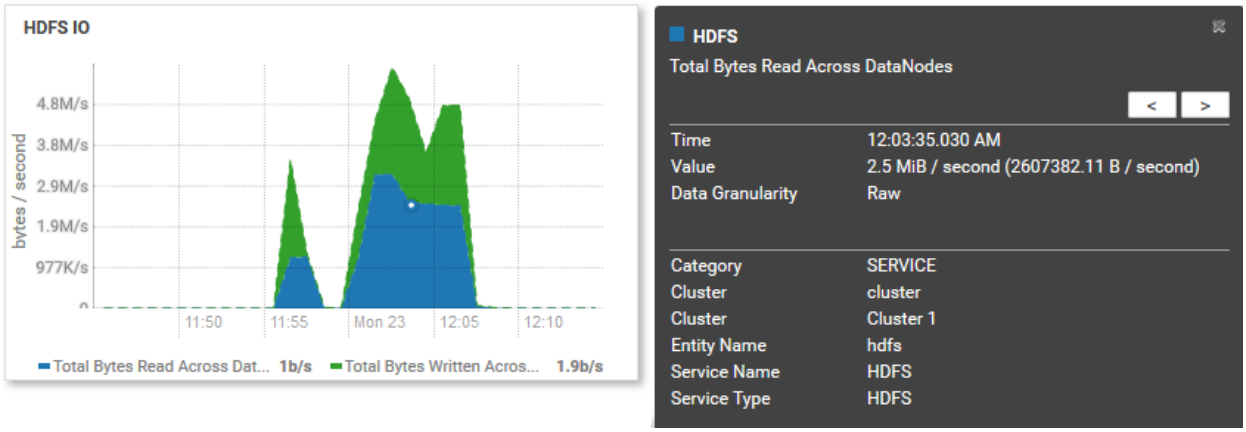

Figure 25 HDFS IO

These metrics are vital when evaluate the performance of H2H to analyze large amounts of Amazon Product Review data.

7.4 Related Work Comparison

Other studies have been conducted on how Hadoop and Hive can be used to perform data analysis since traditional systems are unable to handle large amounts of big data. The study performed in [21] applied Hadoop to financial big data to analyze and retrieve specific information from the data. The use of Hive reduces the programming complexity required to perform the data analysis in comparison to MapReduce which is a compiled language. Hive also requires less development efforts than MapReduce due to its SQL like resemblance. For the study conducted in [21], Hadoop was hosted on a cloud service with 314 MB of stock data where as H2H was hosted on a commodity laptop and was used to analyze over 15.6 GB of Amazon Product Review data. The platform that was used to host H2H did not affect its ability to peform data analysis. If anything, it was proven that effective data analysis can be performed with H2H hosted on a commodity laptop in comparison to using a cloud service. Another benefit of hosting H2H on a commodity laptop in comparison to cloud services, is the portability it provides.

Should a company want to take the laptop to another location, there would be no issue with transporting the laptop. In addition, using H2H on a commodity laptop is a via solution for those companies that do not want their data on a cloud service for extra security measures.

8. SUMMARY AND FUTURE WORK

Merv Adrian first defined Big Data as “Big data exceeds the reach of commonly used hardware environments and software tools to capture, manage, and process it within a tolerable elapsed time for its user population” [8]. With data being produced at an exponentially rapid rate these days, the need to properly analyze the data has gain importance. More companies are realizing the need to analyze their data to obtain insights they would have otherwise not be able to deduce. Through the analysis they could also potentially improve their efficiency, gain competitive advantage and generate innovative corporate dimensions by using H2H to analyze their data. H2H provides a less complex solution for SMEs than what big buisnesses use so they can start analyzing their data and reap the benefits.

This research has shown how SMEs can perform big data analysis by running H2H on a commodity laptop. The main advantage H2H provides is its capability to abstract data processing from the underlying MapReduce through its ease of programming. Those developers familiar with SQL will only have a small learning curve to analyze data with H2H since its query language is very similar to SQL.

Optimization of the queries improves the performance of the data analysis; further research will be made in this area. Other projects are available that can deliver added support such as Apache Spark, Apache HBase, Apache Sqoop, and Apache Flume. Spark is the open standard for flexible in-memory data processing that allows batch, real-time, and cutting-edge analytics on Apache Hadoop system [9]. In comparison to MapReduce, Spark uses its own runtime engine and can perform one hundred times faster than MapReduce since it loads its data in memory [9]. HBase is a distributed database constructed on HDFS to deliver random, real-time read/write

admittance to big data [10]. Sqoop [11] provides a means to transfer data between relational databases and Hadoop and Flume [12] provides a way to collect, aggregate and move large amounts of log data in a distributed, reliable way to a central location.

In future research efforts, these tools will be researched to evaluate their performance on analyzing big data with H2H running on a commodity laptop and how SMEs can adopt them to analyze their data. The end goal of big data analysis is for the data to be properly analyzed to retrieve valuable information and to make an educated and informed decision. Through this research, it has been demonstrated that a non-complex system can be implement to perform data analysis. H2H can run on a commodity laptop which in turn allows SMEs to start analyzing their data like the bigger businesses are currently undertaking. Thus, the importance of big data is not related to its size but how it can properly be analyzed to extract and reveal information that otherwise would not have been possible. As the importance for generating significance out of big data increases, so does the need to develop tools to analyze the big data. The methods for generating data are not slowing down nor are the needs to analyze data at a faster speed.

APPENDIX A: REFRENCES

[1] Apache Hadoop, Welcome to Apache Hadoop. Apache, http://hadoop.apache.org/, accessed November 2017, last updated November 2017.

[2] Apache Hive, Apache HIVE™. Apache, http://hive.apache.org/, accessed November 2017.

[3] SAS, Hadoop What is it and why does it matter?, https://www.sas.com/en_us/insights/big-data/hadoop.html#hadoopimportance, accessed November 2017.

[4] Apache Hadoop, HDFS Architecture Guide, Apache, https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html, accessed November 2017, last updated August 8, 2013.

[5] Apache Hadoop, MapReduce tutorial. Apache, http://hadoop. apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/ MapReduceTutorial.html, accessed October 2017, n.d.

[6] Cloudera, Cloudera Hadoop Documentation, https://www.cloudera.com/documentation.html, access November 2017.

[7] Apache, Apache Hadoop Yarn, https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html, accessed January 2018.

[8] Trnka, Andrej, University of Ss. Cyril and Methodius , Big Data Analysis, July 2014

[9] Apache, Apache Spark Lightning-fast unified analytics engine, https://spark.apache.org/, accessed March 2018.

[10] Apache, Apache HBase, Welcome to Apache HBase™, https://hbase.apache.org/, accessed March 2018.

[11] Apache, Apache Scoop, http://sqoop.apache.org/, accessed April 2018.

[12] Apache, Apache Flume, https://flume.apache.org/, accessed April 2018.

[13] Scrape Hero, Scrape Hero, https://www.scrapehero.com/many-products-amazon-sell-january-2018/, access April 15, 2018, last updated January 28, 2018

[14] Apache, Zooker, https://zookeeper.apache.org/, accessed January 28, 2018

[16] Rising, Carl. Kristensen, Michael. Is Big Data too Big for SMEs?. Stanford University. 2014

[17] Coleman, Shirley, Gob, Rainer. How Can SMEs Benefit from Big Data? Challenges and a Path Forward, Quality and Reliability Engineering International. 2016

[18] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. 2009. Hive: A Warehousing Solution Over a Map-Reduce Framework. PVLDB 2, 2 (2009), 1626–1629.

[19] Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Antony, S., Liu, H. and Murthy, R. 2010. Hive-a petabyte scale data warehouse using hadoop. Data Engineering (ICDE), 2010 IEEE 26th International Conference on (2010), 996–1005.

[20] Potharaju, Sai. Srinivas, Shanumuk. CASE STUDY OF HIVE USING HADOOP, SRES COE. 12 March 2017.

[21] Mehta, Jay. Woo, Jongwook. Big Data Analysis of Historical Stock Data Using HIVE. ARPN Journal of Systems and Software. August 2015.

[22] Swathi, P. Jumari, J. BIG DATA ANALYSIS OF AIRLINE DATA SET USING HIVE. International Journal of Computer Science and Mobile Computing. 6 June 2017.