Lessons Learned: Building a Big Data Research and Education Infrastructure

G. Hsieh, R. Sye, S. Vincent and W. Hendricks

Department of Computer Science, Norfolk State University, Norfolk, Virginia, USA
[ghsieh, wthendricks]@nsu.edu, [r.sye, s.m.vincent]@spartans.nsu.edu

Abstract – *Big data is an emerging technology which has been growing and evolving rapidly in related research, development, and applications. It is used by major corporations and government agencies to store, process, and analyze huge volumes and variety of data. The open source Apache Hadoop platform and related tools are becoming the de facto industry standard for big data solutions. It is a very powerful, flexible, efficient, and feature-rich framework for reliable, scalable, distributed computation using clusters of commodity hardware. On the other hand, Hadoop and its ecosystem are very complex and they change rapidly with new releases, features, packages, and tools. They are also relatively new, and thus lack adequate documentation and broad user experience base that come with mature products. Hence, it is very challenging to learn, install, configure, operate, and manage Hadoop systems, especially for smaller organizations and educational institutions without plenty of resources. In this paper, we describe our experiences and lessons learned in our efforts to build up a big data infrastructure and knowledge base in our university during the past nine months, using a variety of environments and resources, along with an incremental and iterative learning and implementation approach. In addition, we discuss the plan being implemented to enhance the infrastructure to provide enterprise-class capabilities by the end of 2014.*

Keywords: *big data, Hadoop, lab, learning.*

1 Introduction

In a recent report by the National Institute of Standards and Technology Big Data Public Working Group [1], "Big Data refers to the inability of traditional data architectures to efficiently handle new data sets."

"***Big Data*** *consists of extensive datasets, primarily in the characteristics of volume, velocity, and/or variety that require a scalable architecture for efficient storage, manipulation, and analysis.*"

Big data is an emerging technology which has been growing and evolving rapidly in related research, development, and applications. It is used by major corporations (e.g., Google, Facebook, and Amazon) and government agencies (e.g., Department of Defense) to store, process, and analyze huge volumes and variety of data to help make better decisions, improve situational awareness, grow customer base, and gain strategic advantage. In a recent forecast published in Dec. 2013 [2], International Data Corporation (IDC) "expects the Big Data technology and services market to grow at a 27% compound annual growth rate through 2017 to reach $32.4 billion."

The open source Apache Hadoop platform and related tools [3] are becoming the de facto industry standard for big data solutions. It is a very powerful, flexible, efficient, and feature-rich framework for reliable, scalable, distributed computation using clusters of commodity hardware. On the other hand, Hadoop and its ecosystem are very complex and they change rapidly with new releases, features, packages, tools, and even modified API's distributed in a very fast pace. They are also relatively new, with a major portion in beta or production releases within the past year or two. Thus, they lack adequate documentation and broad user experience base that come with mature products. Overall it is very challenging to learn, install, configure, and operate Hadoop systems, especially for smaller organizations and educational institutions without plenty of resources.

Recognizing the importance of big data, a new research effort was launched at Norfolk State University (NSU) in August 2013, with Raymond Sye and Shontae Vincent, both M.S. students in the Department of Computer Science, conducting research in this subject area for their M.S. Thesis/Project under the supervision of Dr. George Hsieh, a professor in the department. The main objectives of this coordinated research effort were:

(a) Learn the fundamentals of Hadoop architecture, processing model, and key technological components.
(b) Install and configure small-scale Hadoop clusters in the lab to gain hands-on knowledge, skills and experiences.
(c) Apply the acquired knowledge and skills, and use the established lab infrastructure to perform graph-based computations.

1.1 Phased Approach

To accomplish these objectives, an incremental and iterative approach was used to tackle the complexity and challenges discussed earlier.

The main activities for this research effort can be grouped into six major steps in a roughly sequential order:

(1) Get started with Hadoop using Hortonworks Sandbox [4] and its interactive tutorials in a single-node virtual machine configuration.
(2) Install and configure a multi-node Hadoop cluster in virtual machines, using Hortonworks Data Platform (HDP) [5] and Ambari cluster management tool [6] .
(3) Install and configure a five-node Hadoop cluster on commodity hardware, using HDP and Ambari.
(4) Get started with Hadoop application development, using Cloudera QuickStart VM [7], in a single-node virtual machine configuration.
(5) Install and configure a seven-node Hadoop cluster on commodity hardware, using Cloudera's CDH [8] and Cloudera Manager [9].
(6) Develop a Hadoop graph-based application, using the Cloudera based, multi-node Hadoop cluster.

Note that we chose Hortonworks and Cloudera, which are among the top commercial vendors that provide customized Hadoop software distributions based on the common Hadoop code managed by Apache Software Foundation. These vendors also develop and supply additional tools and capabilities beyond the common Hadoop code base, such as Cloudera Manager, to simplify and automate the installation, configuration, and administration of Hadoop systems.

We also chose the open-source CentOS Linux [10] as the base operating system for all of our Hadoop systems, virtual and physical, primarily because of its ease of use, enterprise-class features, and security enhancements.

1.2 Infrastructure Enhancement

In February 2014, Norfolk State University was awarded an equipment grant entitled "Building a Cloud Computing and Big Data Infrastructure for Cybersecurity Research and Education" by the U.S. Army Research Office. The funds from this grant will allow NSU to significantly enhance its big data research and education infrastructure by bringing in enterprise-class capabilities in computing, storage, and networking.

The knowledge, skills, and experiences accumulated through the past year are extremely useful for planning and designing this new phase of infrastructure expansion. To date, all necessary hardware, software, and facilities for the planned expansion have been selected, designed, and ordered. We plan to stand up the expanded infrastructure around the fourth quarter of 2014.

1.3 Outline

The remainder of the paper is organized as follows. In Section 2 we provide an overview of the major steps used in our phased approach. In Section 3 we describe our planned expansion of the infrastructure in more detail. In Section 4 we conclude the paper with a summary and some how-to recommendations for building up a big data research and education infrastructure in a timely and cost-effective manner without requiring significant upfront investments in people and resources.

2 Initial Infrastructure

In this section, we discuss the six steps used to build up our initial big data research and education infrastructure from both human expertise and system resources perspectives, during the past nine months from September 2013 to May 2014.

2.1 Getting Started with Hadoop

Given the complexity and rapid changing pace of Hadoop and its ecosystem, it was truly challenging to figure out an effective way of getting started with Hadoop without relying on professional services or staff.

After a relatively short period of investigation and experimentation, we chose Hortonworks Sandbox as the preferred platform for the "getting-started" training on Hadoop for ourselves and five additional members who joined the research team later.

According to Hortonworks, "Sandbox is a personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials. Sandbox includes many of the most exciting developments from the latest HDP distribution, packaged up in a virtual environment that you can get up and running in 15 minutes!" [4]

We started with Version 1.3 of Sandbox and then migrated to Version 2.0 when the newer version became available. Sandbox is provided as a self-contained virtual machine for three different virtualization environments: (a) Oracle VirtualBox; (b) VMware Fusion or Player; and (c) Microsoft Hyber-V. We tried to use Sandbox with both VirtualBox and VMware environments, and found that Sandbox worked better with VirtualBox which is also the recommended virtualization environment for Sandbox.

After downloading the Sandbox VM image, the next step was to import the appliance into VirtualBox. This step was very straightforward for people with basic knowledge and experience in VirtualBox.

Once the Sandbox VM is started, a user can initiate a Sandbox session by opening a web browser and entering a pre-configured IP address (e.g., http://127.0.0.0:8888/). Once connected to the web server running locally, a user can learn Hadoop on Sandbox by following a dozen or more hands-on tutorials.

We found Hortonworks Sandbox to be a very effective learning environment for "getting-started" with Hadoop. Sandbox's integrated, interactive, and easy-to-use tutorial environment enables a user to focus on the key concepts for the tasks on hand, without having to learn the detailed mechanics behind the scene immediately. It also provides a rich set of video, graphical, and text based instructions along with informative feedback during exercises and suggestions for corrective actions when errors occurred.

Running Sandbox for learning Hadoop does not require a great deal of hardware resources. It runs well on commodity 64-bit systems with virtualization technology hardware support and a minimum of 4 GB RAM. Note that some Intel 64-bit processors do not provide virtualization

technology hardware support, and Sandbox will fail to run on these systems.

2.2 Multi-Node HDP Cluster in VMs

After completing the tutorials provided by Sandbox and having gained the basic knowledge about Hadoop, we proceeded to learn and experiment with installing and configuring a multi-node Hadoop cluster using Hortonworks Data Platform running in multiple virtual machines on the same physical host system. This step was designed to leverage our familiarity with Hortonworks Sandbox gained earlier while tackling the more challenging task of setting up a multi-node Hadoop cluster.

We chose to install and configure the HDP 2.0 based Hadoop cluster using the Apache Ambari Install Wizard [11]. Ambari provides an easy-to-use graphical user interface for users to deploy and operate a complete Hadoop cluster, manage configuration changes, monitor services, and create alerts for all the nodes in the cluster from a central point, the Ambari server.

The first step in this implementation was to layout a design for the Hadoop cluster including:

(a) The number of hosts: 6.
(b) The types of hosts - Ambari Server: 1; Masters: 2; Slaves: 2; and Client: 1.
(c) FQDN and IP address for each host (without using DNS).

The second step was to create six VMs each loaded with CentOS 6.4 (or newer). Then configure each VM to set up the appropriate hostname, IP address, host file, password-less SSH, and other prerequisite software (e.g., JDK).

The third step was to install Apache Ambari on the host designated as Ambari Server. Once the Ambari service was started, the next step was to access the Ambari Install Wizard through a web browser to complete the installation, configuration and deployment of this Hadoop cluster.

We found the Apache Ambari to be a very easy-to-use tool for installing, configuring, and deploying a Hadoop cluster, as it automates many of the underlying tasks for the user who only needs to supply high-level information such as the hosts, their roles (manager, master, slave, or client), and the services to be assigned to the hosts.

Ambari allocates these services to the appropriate hosts for load balancing and reliability concerns, and then proceeds to install, configure, start and test the appropriate software on these hosts automatically.

For this exercise, we used commodity Windows based PC's with moderate processing power (e.g., 64-bit CPU with 2 to 4 cores, and 8 to 16 GB RAM). Installation using Ambari did run into capacity related problems from time to time, especially when the Ambari server was downloading and installing software to all targeted hosts simultaneously. Some of the underlying tasks could fail and thus cause the installation to fail. One side effect of this failure was that the rpm database often got corrupted. Rebuilding the rpm database often resolved this kind of problem and allowed the installation to proceed (at least incrementally until the next failure occurred).

2.3 Multi-Node HDP Cluster

The experiences gained in designing, installing, configuring, and operating the six-node HDP based Hadoop cluster in a virtual environment were very useful for our next step of setting up a multi-node HDP cluster using multiple physical hosts.

As shown in Table 1, four commodity systems were used for this Hadoop cluster based on HDP and managed by Apache Ambari.

Table 1. Multi-node HDP cluster

Hosts	Hostname/ local IP	CPU	RAM	Disk
Ambari Server	HDPcs2AMBARI 199.111.112.169	Intel Xeon (4C) 3GHz	8 GB	500 GB
Master Node	HDPcs2MASTER 199.111.112.171	Intel Core 2 Duo 3GHz	4 GB	250 GB
Data Node 1	HDPcs2DN1 199.111.112.180	Intel Core 2 Duo 3GHz	4 GB	250 GB
Data Node 2	HDPcs2DN2 199.111.112.189	Intel Xeon (4C) 3.2GHz	12 GB	900 GB

Monitoring and managing a large scale distributed Hadoop cluster is a non-trivial task. To help the users deal with the complexity, Ambari collects a wide range of information from the nodes and services and presents them in an easy-to-use dashboard, as shown in Figure 1. Ambari also allows users to perform basic management tasks such as starting and stopping services, adding hosts to a cluster, and updating service configuration.

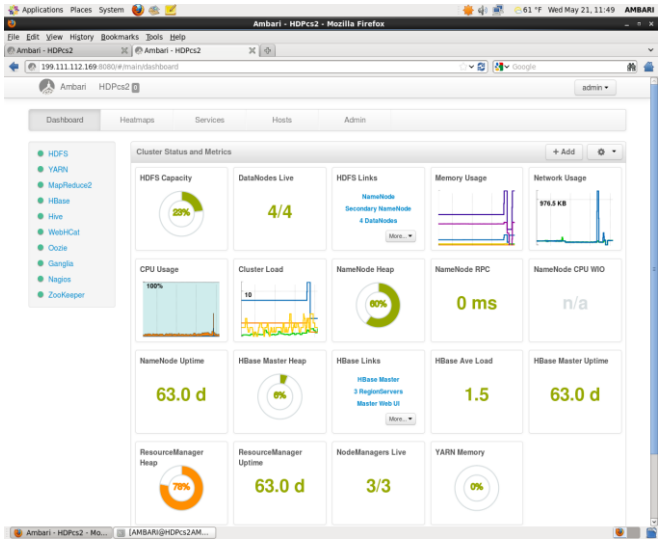

Figure 1. Ambari Dashboard display

2.4 Cloudera QuickStart VM

Cloudera has been considered the market leader among pure play Hadoop vendors that provide Hadoop related software and services. It also builds proprietary products on top of open source Hadoop code with an "open source plus proprietary model".

On such product is Cloudera Manager [9] which is included in Cloudera Express and Cloudera Enterprise. With Cloudera Express, which is a free download, users can easily deploy, manage, monitor and perform diagnostics on Hadoop clusters. Cloudera Enterprise, which requires an annual subscription fee, includes all of these capabilities plus advanced management features and support that are critical for operating Hadoop and other processing engines and governance tools in enterprise environments.

Given Cloudera's market leadership position and the potential benefits of its proprietary products, we simply did not want to ignore it.

The easiest way to get started with Cloudera's products was to use its QuickStart VM [12] which contains a single-node Apache Hadoop cluster including Cloudera Manager, example data, queries, and scripts. The VM is available in VMware, VirtualBox and KVM flavors, and all require a 64 bit host OS. This VM runs CentOS 6.2. We used primarily the CDH 4.4 and CDH 4.6 versions of the QuickStart VM.

Cloudera QuickStart VM did not provide an integrated tutorial environment or a collection of tutorials that were as easy to use as those provided by Hortonworks Sandbox. On the other hand, it provided all the commonly used Hadoop platform and tools. Thus, users did not need to download, install, and configure these packages individually.

In addition, Cloudera QuickStart VM included many of the commonly used software development tools (e.g., Eclipse and JDK) which made it a more suitable platform for developing Hadoop applications than Hortonworks Sandbox.

Getting started with developing Hadoop applications, beyond the simple "Hello World" type of tutorial app, can be quite challenging. Many tasks require executing Linux shell scripts with long lists of command line arguments. In addition, the binaries, shell scripts, and configuration files can be in different locations, depending on how the Hadoop system is installed and configured and which Hadoop distribution is used. Furthermore, the user accounts can be set up differently. All these factors make it challenging to get started with Hadoop application development, as the user needs to first gain a good understanding of the lay of the land so (s)he can navigate around these issues. The user also needs to have a sufficient level of proficiency in working with Linux OS and prior software development experiences in general.

To gain the basic knowledge and skills in Hadoop application development, we used primarily two books as resources. The first book entitled "Hadoop Beginner's Guide" [13], by Gary Turkington and published in February 2013, provided a very useful introduction to Hadoop application development with clear description and good example code. It was also not too difficult to get started with running the example code, as we used the Cloudera QuickStart VM as the platform which already contained the vast majority of the Hadoop software and prerequisite software development tools. Furthermore, the example code, although written with the older versions of Hadoop software and tools at the time of publication, worked well with the newer versions bundled with Cloudera QuickStart VM.

Another book we used for learning Hadoop application development was entitled "Hadoop in Practice" [14] by Alex Holmes and published in October 2012. The Appendix A contained background information on all the related Hadoop technologies in the book. Also included were instructions on how to build, install, and configure related projects.

To set up an environment as specified in the appendix, we started with creating a virtual machine loaded with CentOS 6.4 (Software Development Workstation option). We next installed the Hadoop base using CDH 3.0 distribution and configure our Hadoop system for the pseudo-distributed mode. We then installed and configured the remaining nineteen packages manually and individually. These packages included MySQL, Hive, Pig, Flume, HBase, Sqoop, Oozie, R, etc.

It was very challenging to go through all the steps to install and configure this target Hadoop system using primarily manual procedures and separate packages one at a time. Many of these challenges could be alleviated by using cluster provisioning and management tools such as Apache Ambari and Cloudera Manager.

Nonetheless, going through this process helped us to gain much deeper understanding and appreciation of the interdependencies and intricacies involved in getting all these packages installed and configured correctly so they can function together. This kind of knowledge and skills are important for troubleshooting problems and customizing installations, configurations, and operations, even with the cluster management tools available. Some of the important lessons learned include:

(a) Installing a Linux OS option pre-packaged with software development tools can save a lot of time and effort, as numerous extra packages are generally required to be downloaded, installed, configured, and even built on demand.

(b) The installed directories for the same software could be different, depending on the installation procedures and instructions. For example, installing from tarballs versus installing via rpm/yum could install the same software in different directories. So it is important to recognize this potential difference and make plans or adjustments accordingly.

(c) Make sure all the required environmental variables (e.g., PATH), and profiles are set up correctly. It is useful to have them set up consistently across user accounts and across hosts. Some Hadoop packages require specific global environmental variables to be defined in their specific configuration files.

(d) There could be many hard and symbolic (soft) links in the file system allowing multiple filenames (or directories) to be associated with a single file (or directory). It is important to understand these links to make sure that the correct files (or directories) are updated and links are not broken accidentally.

(e) Similarly, it is important to understand the Linux alternatives system which uses symbolic links extensively to manage the alternatives for a given generic name. For example, several different Java packages and JDK's may be installed on the same system. Activating the specific packages may require rearranging the alternatives (in their preferences).

2.5 Multi-Node CDH Cluster

The next step was to set up a multi-node Hadoop cluster using Cloudera distribution while taking advantage of the capabilities provided by Cloudera Manager.

For this exercise, we installed and configured a seven-node cluster, one as the Manager, two as masters, and four as slaves. The Manager node has two Ethernet connections, one to Internet and the other to an internal network for the Hadoop cluster. All remaining nodes are connected only to the internal network physically. The Manager node also performs IP forwarding for the remaining nodes so they can access the Internet indirectly through the Manager node. Figure 2 shows the connectivity among the nodes.

Again, the nodes were implemented using commodity machines all running CentOS 6.4 (Software Development Workstation option). The Cloudera software deployed was based on Cloudera Express 5.0.0-beta-2 release which contained Hadoop Version 2.2.0. Also installed was Hue Version 3.5.0 which is an open-source Web interface that supports Hadoop and its ecosystem. Hue provides a Web application interface for Apache Hadoop. It supports a file browser, JobTracker interface, Hive, Pig, Oozie, HBase, and more. Table 2 shows the hardware configurations for the Cloudera based Hadoop cluster.

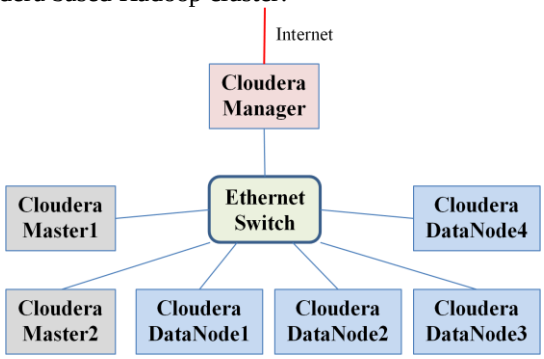

Figure 2. Multi-node Cloudera cluster

Table 2. Hardware configuration for CDH cluster

Hosts	Hostname/ local IP	CPU	RAM	Disk
Manager	CDHcs1mgr 192.168.48.1	Intel Xeon (4C) 3GHz	8 GB	500 GB
Master1	CDHcs1MN1 192.168.48.10	Intel Xeon (4C) 2.5GHz	8 GB	150 GB
Master2	CDHcs1MN1 192.168.48.2	Intel Xeon (4C) 2.5GHz	8 GB	150 GB
DataNode1	CDHcs1DN1 192.168.48.11	Intel Xeon (4C) 2.5GHz	8 GB	150 GB
DataNode2	CDHcs1DN2 192.168.48.12	Intel Xeon (4C) 3.2GHz	8 GB	150 GB
DataNode3	CDHcs1DN3 192.168.48.13	Intel Core 2 Duo 3GHz	4 GB	250 GB
DataNode4	CDHcs1DN4 192.168.48.14	Intel Core 2 Duo 3GHz	4 GB	250 GB

We chose to use the Cloudera Express 5.0.0-beta-2 release, because a decision had been made around that time to deploy Cloudera distribution for the new equipment being acquired for our infrastructure enhancement effort. Thus, we wanted to become familiar with the Cloudera 5.0 release, even when it was still in beta stage, so we would be prepared to work with it when the new equipment is deployed. As a result, we had to work with the beta version of the Cloudera Manager Installation Guide which did not contain as much information as the most recent Version (5.0.1) of the guide [15] published on May 28, 2014.

Although Cloudera Manager provided an automated installation option, "This path is recommended for demonstration and proof of concept deployments, but is not recommended for production deployments because it's not intended to scale and may require database migration as your cluster grows." [15].

Based on this recommendation, we chose to follow the Installation Path B – Manual Installation Using Cloudera Manager Packages. This path required a user to first manually install and configure a production database for the Cloudera Manager Server and Hive Metastore. Next, the user needed to manually install the Oracle JDK and Cloudera Manager Server packages on the Cloudera Manager Server host. To install Oracle JDK, Cloudera Manager Agent, CDH, and managed service software on cluster hosts, we used Cloudera Manager to automate installation.

Table 3 shows the roles assigned to the CDH cluster hosts to implement the selected features while balancing the computing, storage, and networking resources needs. Figure 3 shows the status display of the deployed cluster by Cloudera Manager.

Table 3. Roles assigned to CDH cluster hosts

Hostname	Roles
CDHcs1mgr	Cloudera Activity Monitor; Cloudera Alert Publisher; Cloudera Event Server; Cloudera Host Monitor; Cloudera Reports Manager (enterprise version); Cloudera Service Monitor. Hive Gateway; Hive Metastore. Hue Server.
CDHcs1MN1	HBase Master. HDFS Httpfs; HDFS Namenode-Active. Hive Gateway; Hive HiveServer2. Spark Master. Zookeeper Server – follower.
CDHcs1MN2	HBase Region Server. HDFS Datanode; HDFS NFSGateway; HDFS NameNode – Secondary. Solr Server. Spark Worker.
CDHcs1DN1	HBase Region Server. HDFS Datanode. HIVE Gateway. Oozie Server. YARN Job History; YARN Node Manager; YARN Resource Manager.
CDHcs1DN2	HBase Region Server. HDFS Datanode. Hive Metastore; Hive HiveServer2; Hive WebCat. YARN Node Manager.
CDHcs1DN3	Flume Agent. HBase REST server. HDFS Datanode. YARN Node Manager. Zookeeper

	Server – leader.
CDHcs1DN4	HBase Thrift Server. HDFS Datanode. YARN Node Manager. Zookeeper Server – follower.

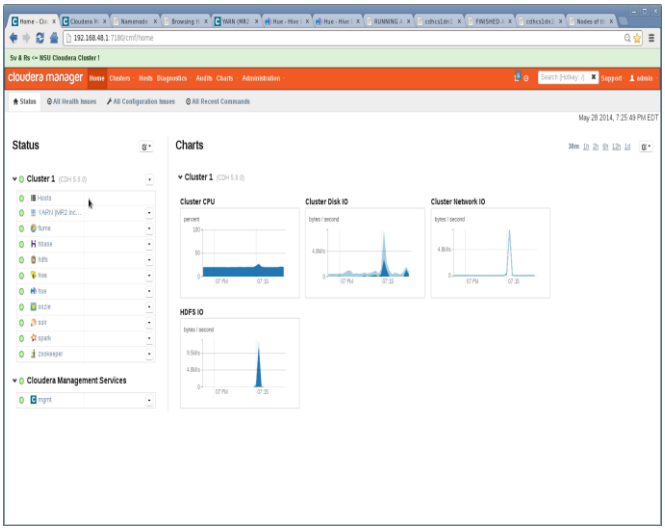

Figure 3. Cloudera Manager status display

Installing and configuring this Hadoop cluster using the semi-automated approach was quite challenging. Some of the important lessons learned include:

(a) Permissions. Many of the Hadoop packages require access to shared resources. The beta version of the installation guide, as far as we knew, did not include detailed instructions on setting up the appropriate permissions for various Hadoop components so they could work together. Hence, we needed to figure out how to grant permissions, primarily through group memberships and group permissions, to various Hadoop software and resources (e.g., mapred must be a member of the hadoop group as well). This issue has been addressed by the latest version of the installation guide.

(b) User accounts and groups. Similarly, the latest guide also provided detailed instructions on the user accounts and groups that Cloudera Manager and CDH used to complete their tasks. This standardized setup should be followed to make sure the user accounts, groups, and permissions are consistent across all hosts. This also makes it easier to ensure that the environmental variables and profiles are set up consistently across hosts.

(c) Interdependencies. Although it might not be stated in the documentation, the order in which the various packages are installed may make a difference in the ease of configuring these packages that have interdependencies. For example, our experience indicated that it was better to install and configure the ZooKeeper before installing Hive. Attempting to install ZooKeeper after Hive was installed could cause issues with the HiveServer service.

(d) Performance. Although Hadoop has a very flexible distributed architecture, sometimes it is better to run closely related services/tasks on the same physical host to reduce the latency and overhead. This was especially important during the installation phase and using hardware with limited resources.

(e) It is critical to keep a close watch on disk storage and memory use. The available disk space could be depleted when a large volume of log files were generated. The available memory could also be depleted after a period of operation. Running low on disk space and memory usually caused systems to reboot or become nonresponsive.

(f) Files in some directories could be deleted by Cloudera Manager after making configuration changes through Manager. Make sure important files are not kept in these directories or they are backed up somewhere else.

2.6 Hadoop App for Graph Processing

Graph-based processing was one of the first categories of Hadoop applications in which we were interested. So we worked with Apache Giraph (v1.0.0) which "is an iterative graph processing system built for high scalability. For example, it is currently used at Facebook to analyze the social graph formed by users and their connections." [16]

Again, we used a phased approach in working with Giraph. First, we followed the Giraph Quick Start guide [17] to install and run Giraph in a single-node, pseudo-distributed Hadoop cluster on a VM loaded with Ubuntu Server 12.04.2 (64-bit) OS. We verified that the installation was operational by running the "SimpleShortestPathsComputation" example job and obtaining the desired output successfully.

Next we proceeded to install Giraph on the multi-node Cloudera based cluster described in the previous section. For this exercise, we used the information contained in another resource [18] to help install Giraph on CentOS which is the base OS for our Cloudera based cluster. Again, we ran the "SimpleShortestPathsComputation" example job to verify that the Giraph installation on this cluster was operational.

Our experience indicated that the node on which Giraph is executed should also have YARN (MR2) Node Manager service, HDFS DataNode service and ZooKeeper service running on the same node for better performance and increased level of robustness.

Without accessibility to Zookeeper, we experienced problems with running example Giraph jobs, as multiple failures could occur without clear error messaging. Also, other execution errors occurred with Giraph when the job was not run on a node with YARN Node Manager or the YARN node is not specified. Giraph and YARN work closely together. With large Giraph calculations, the connectivity to a remote Mapreduce service could become disconnected and cause the Giraph job to fail.

Running Giraph job was a bit of a challenge. As stated before, denoting the nodes that run the ZooKeeper service can help prevent failures. Giraph does come with example code that provides a wide range of functionality. For example, "SimpleShortestPath" works well with a properly formed file with adjacency lists. However, a user needs to make sure that no extraneous white spaces or blank lines are

included in the input text file. Otherwise, this example job could fail. However, the "PageRankBenchmark" example job did not actually produce any output, although it could be completed successfully.

3 Infrastructure Enhancement Planned

As mentioned earlier, an enhancement to the current infrastructure is planned for completion by 4Q2014. A new "production" system with five master nodes and twelve data nodes will be installed in a server room, while another new "integration and testing" system with five master and data nodes will be deployed in a research lab. The current systems will remain in the research lab and used primarily for learning, development, and development testing purposes.

The new equipment will add approximately six hundred Intel Xeon 64-bit CPU cores, 350 terabytes of disk storage, 3 terabytes of RAM, and three high-performance L2/L3 Ethernet switches supporting 40GbE connectivity.

4 Summary

This paper presented our lessons learned in building a big data research and education infrastructure. As big data continues to gain rapid growth in research, development, and deployment, it is important and beneficial for organizations in both public and private sectors to leverage big data to gain insights and improve their operation. It is also important and beneficial for educational institutions to engage in big data related research, education, and workforce development to help advance the state of the art of this critically important technology, and address the talent shortage problem forecast for many years to come.

However, due to the complexity, immaturity, and fast pace in evolving of big data platforms and tools, it is very challenging to build up a big data research and education infrastructure in both human and system resources, especially for small to medium businesses, organizations, and educational institutions without plenty of resources.

We took an incremental and iterative approach to build a small size infrastructure at a university with about 6,000 students in enrollment, without requiring investments in staff and hardware/software resources. The knowledge and skills were acquired through student research projects required for their degrees. This approach provided additional benefits to the students' professional development. The hardware used for this effort was all commodity hardware already available in the institution. The software used was all open source or free.

Even so, it was very challenging to get it done. Good planning, perseverance, and dedicated personnel can prevail.

5 Acknowledgement

This research was supported in part by U.S. Army Research Office, under grant numbers W911NF-12-1-0081 and W911NF-14-1-0045, and U.S. Department of Energy, under grant number DE-FG52-09NA29516/A000.

6 References

[1] NIST Big Data Public Working Group, "DRAFT NIST Big Data Interoperability Framework: Volume 1, Definitions (Draft Version 1)," April 23, 2014.

[2] International Data Corporation, "Worldwide Big Data Technology and Services 2013–2017 Forecast," Dec 2013.

[3] "Apache Hadoop," [Online]. Available: http://hadoop.apache.org/. [Accessed 31 May 2014].

[4] "Hortonworks Sandbox," [Online]. Available: http://hortonworks.com/products/hortonworks-sandbox/. [Accessed 31 May 2014].

[5] "Hortonworks Data Platform," [Online]. Available: http://hortonworks.com/hdp/. [Accessed 31 May 2014].

[6] "Apache Ambari," Hortonworks, [Online]. Available: http://hortonworks.com/hadoop/ambari/. [Accessed 31 May 2014].

[7] "Cloudera QuickStart VM," [Online]. Available: http://www.cloudera.com/content/cloudera-content/cloudera-docs/DemoVMs/Cloudera-QuickStart-VM/cloudera_quickstart_vm.html. [Accessed 31 May 2014].

[8] "Cloudera CDH," [Online]. Available: http://www.cloudera.com/content/cloudera/en/products-and-services/cdh.html. [Accessed 31 May 2014].

[9] "Cloudera Manager," [Online]. Available: http://www.cloudera.com/content/cloudera/en/products-and-services/cloudera-enterprise/cloudera-manager.html. [Accessed 31 May 2014].

[10] "CentOS," [Online]. Available: http://www.centos.org/. [Accessed 31 May 2014].

[11] "Hortonworks Data Platform: Installing Hadoop Using Apache Ambari," Hortonworks, 2013.

[12] "Cloudera QuickStart VM," [Online]. Available: http://www.cloudera.com/content/support/en/downloads/download-components/download-products.html?productID=F6mO278Rvo. [Accessed 31 May 2014].

[13] G. Turkington, Hadoop Beginner's Guide, Birmingham: Packt Publishing, 22 Feb 2013, p. 398.

[14] A. Holmes, Hadoop in Practice, Shelter Island, NJ: Manning Publications Co., October 2012, p. 536.

[15] Cloudera, "Cloudera Manager Installation Guide (Version 5.0.1)," Cloudera, 2014.

[16] "Apache Giraph," Apache Software Foundation, [Online]. Available: https://giraph.apache.org/. [Accessed 4 June 2014].

[17] "Apache Giraph Quick Start," Apache Software Foundation, [Online]. Available: http://giraph.apache.org/quick_start.html. [Accessed 4 June 2014].

[18] "Install giraph in hadoop node," [Online]. Available: http://www.sbarjatiya.com/notes_wiki/index.php/Install_giraph_in_hadoop_node. [Accessed 4 June 2014].