Teaching Big Data with a Virtual Cluster

Joshua Eckroth
Math/CS Department
Stetson University
jeckroth@stetson.edu

ABSTRACT

Both industry and academia are confronting the challenge of *big data*, i.e., data processing that involves data so voluminous or arriving at such high velocity that no single commodity machine is capable of storing or processing them all. A common approach to handling big data is to divide and distribute the processing job to a cluster of machines. Ideally, a course that teaches students how to work with big data would provide students access to a cluster for hands-on practice. However, a cluster of physical, on-premise machines may be prohibitively expensive, particularly at smaller institutions with smaller budgets.

In this report, we summarize our experiences developing and using a *virtual cluster* in a big data mining and analytics course at a small private liberal arts college. A single moderately-sized server hosts a cluster of virtual machines, which run the popular Apache Hadoop system. The virtual cluster gives students hands-on experience and costs less than an equal number of physical machines. It is also easily constructed and reconfigured. We describe our implementation, analyze its performance characteristics, and compare costs with physical clusters and the Amazon Elastic MapReduce cloud service. We summarize our use of the virtual cluster in the classroom and show student feedback.

Keywords

big data; virtual machines; cloud computing; curriculum

1. INTRODUCTION

Today's data scientists work with larger datasets than ever before. Known as "big data," these datasets are so massive that new tools, techniques, and expertise are required. Many of these tools and techniques have emerged only in the last few years and have not yet transitioned into the undergraduate curriculum. We believe computer science and information technology students should learn the theory and practice of big data. Doing so may greatly improve their standing in the job market and their ability to keep up with a rapidly changing technology landscape.

Teaching big data presents a challenge. The challenge is not due to a lack of data. A variety of large datasets are publicly available without cost from myriad sources. The challenge is obtaining sufficient computational and storage resources to work with big data. Because big data is normally too large for processing or storage on a single commodity machine, i.e., anything short of a supercomputer, multiple machines must be used in tandem. Machines are often networked into a cluster and the data are stored and processed in a distributed, parallel fashion. However, the costs of building a cluster may present a serious obstacle for smaller computer science departments hoping to give students hands-on experience with big data.

In order to reduce costs while still offering a hands-on big data infrastructure, we developed a *virtual cluster* using virtual machines rather than physical machines. This system was used in a *Big Data Mining and Analytics* course at Stetson University in Spring 2015. The virtual cluster proved successful for the course and was procured with relatively low cost.

In the following sections, we describe the technical details of the virtual cluster (Section 2). In Section 3, we examine the costs of this system and alternatives. This is followed by an evaluation the virtual cluster's performance under a variety of configurations in Section 4. Our system's performance and costs are compared to the Amazon Elastic MapReduce cloud service in Section 5. Discussion follows in Section 6. We then report on our experiences with the virtual cluster in the classroom (Section 7), summarize related work (Section 8), and offer conclusive remarks and plans for future work in Section 9.

2. IMPLEMENTATION

Rather than purchase a number of physical machines to form a cluster, we upgraded an existing server to host virtual machines. Details about the hardware and associated costs are presented in Section 3. In this section, we discuss the software that configures and runs the virtual cluster. We chose only open source software due to cost and the ubiquity of such software in industry and academia.

We opted to make use of the Apache Hadoop platform. Hadoop is a common open source software system for managing distributed storage and processing. Many software packages enhance the Hadoop system, and are commonly used by data scientists. The variety and popularity of Hadoop-based tools greatly enhances the value of Hadoop in an ed-

SIGCSE '16, March 02-05, 2016, Memphis, TN, USA
© 2016 ACM. ISBN 978-1-4503-3685-7/16/03...$15.00
DOI: http://dx.doi.org/10.1145/2839509.2844651

ucational setting. Some examples of Hadoop-based tools include the following.

- Apache Hive provides SQL-like language for querying against large datasets.
- Apache Pig provides the PigLatin language to more easily describe complex dataflow processing tasks.
- Apache Mahout provides algorithms for machine learning applications.

Hadoop stores data in the Hadoop Distributed File System (HDFS) [7]. HDFS is designed to manage the storage of very large datasets with high reliability. It does so by splitting the data into chunks and storing each chunk on a different machine in the cluster. Furthermore, each chunk is replicated on other machines to ensure the original data can be pieced back together if a small number of machines go offline or lose their data.

In a typical Hadoop data processing scenario, the job is split into stages, distributed to multiple machines, and executed in parallel. Ideally, each machine holds the subset of data it is required to process, thus limiting communication over the virtual or physical network. Results from each machine are collected and merged in a final processing stage.

A typical Hadoop system consists of three or more machines (virtual or physical): a ResourceManager, a NameNode, and $N \geq 1$ additional nodes that both store and process data. The ResourceManager is responsible for managing the processing jobs. The NameNode is responsible for managing the HDFS index. The ResourceManager and NameNode are not required to be installed on separate machines, but it is common to do so in order to reduce resource contention. The number of nodes N should be decided based on expected workload. Below, we evaluate a range of values for N in our virtual cluster environment.

Hadoop can be set up in one of two ways:

- *Local-Hadoop*, which runs normal Hadoop jobs but does not actually distribute the workload. All processing is performed by a single machine hosting the local-Hadoop system. Data are stored on the local filesystem, not distributed in HDFS. Local-Hadoop is commonly used for testing purposes.
- *Cluster-Hadoop*, which runs jobs in a distributed fashion across N nodes, and stores data in HDFS. Unlike local-Hadoop, this setup takes advantage of distributed data storage and processing.

Our goal was to give students hands-on experience with Hadoop. This could be accomplished in several ways:

1. Ask students to install local-Hadoop on their own machines.
2. Install local-Hadoop on each student-accessible lab computer.
3. Give students access to a physical cluster of machines running cluster-Hadoop.
4. Give students access to a virtual cluster of machines (hosted on one physical machine) running cluster-Hadoop.
5. Give students access to a cloud service capable of running cluster-Hadoop, such as Amazon Elastic MapReduce, Microsoft Azure HDInsight, or Google Cloud.

Option (1) is untenable because not all students have access to sufficiently powerful personal machines. Both options (1) and (2) do not give students experience with distributed parallel computing since local-Hadoop processes jobs only on a single machine. Options (3) and (4) differ in two dimensions: as shown below, a physical cluster (option 3) may be more expensive than a virtual cluster (option 4), but a physical cluster may also be more performant. We investigate these dimensions below in Sections 3 and 4, respectively. Option (5) presents a different cost model in which a student or school pays for each data processing job rather than one up-front cost. We examine this option in Section 5.

We used LibVirt and the KVM/QEMU hypervisor on a CentOS Linux host machine to run the virtual machines. Each virtual machine runs Ubuntu Linux 14.04. Hadoop version 2.6 was installed in each virtual machine. Vagrant and Ansible were used to automatically generate and configure the virtual machines just by executing a single command. Our cluster configuration software[1] consists of Vagrant and Ansible scripts that build or restart the virtual machines, install and configure Hadoop, and configure student access to the web-based Hadoop management system. The configuration software supports several virtualization platforms, including VirtualBox and VMWare, and a configurable number of nodes N. When used without the Vagrant component, our configuration scripts are able to install and configure Hadoop and related tools on each machine in a physical cluster as well. These scripts enable instructors and/or IT personnel to easily build and manage a virtual or physical cluster. Students likewise can begin using the system without any special configuration on their own machines.

The configuration software consists of a couple primary commands. We assume the number of nodes N is predefined in the Vagrant script.

- `vagrant up` — creates N virtual machines and, if necessary, installs Hadoop and related utilities on each.
- `setup-hadoop.sh` — installs Hadoop configuration to ensure each node is aware of the others and to enable the Hadoop web-based management system.

These two commands build and configure the entire virtual cluster. After they complete, Hadoop jobs may be executed from the host machine (the server hosting the virtual machines) with commands like:

- `hadoop jar wc.jar WordCount /input.txt /output`

Students may view the status of their jobs by accessing the web interface. In our case, we showed students how to communicate with the host server via Secure Shell tunnels. After connecting to the host server, they were able to access the web interface in their normal browser.

Students used Eclipse and a local-Hadoop installation to test jobs on small datasets before submitting them to the virtual cluster. We also provided a local-Hadoop installation on a separate virtual machine for those who did not wish to install Hadoop on their own machines.

[1] `https://github.com/StetsonMathCS/hadoopvirtualcluster`

3. COSTS

One of the two differences between physical clusters and virtual clusters is their costs. The other difference is performance. We evaluate performance of the virtual cluster in Section 4, below. Here, we compare the prices of new hardware for a single server capable of hosting virtual machines and forming a virtual cluster, and the prices of new hardware to form a physical cluster.

While we did not purchase all new hardware for our Spring 2015 course, we have identified a comparable new machine from Dell. We choose Dell in these cost comparisons simply for convenience, as Dell sells a wide variety of hardware. The top section of Table 1 estimates the cost of a server with the same performance characteristics and storage capacity of the server used in our course. This proposed server is capable of hosting a large number of virtual machines, perhaps up to fifty. We did not test with so many virtual machines because we experienced diminishing returns after about twenty.

We estimate that a new server or cluster of servers will require significant upgrades after three years of use. We estimate the power costs assuming three years of continuous usage multiplied by the US national average electricity cost. Other incidental costs, such as personnel required for maintenance, space for housing the computers, air conditioning, etc. are not included in our calculations, as they vary widely and depend on the particulars of the institution.

The cost for an *equivalent* physical cluster is difficult to estimate. Virtualization adds some overhead. Johnson et al. [4] found that KVM-based virtualization adds little overhead for CPU and memory access, but may significantly diminish network performance (communication between virtual machines). It is not clear how a typical Hadoop job is impacted by this observation. Even so, assume that one virtual machine on the proposed server has nearly equal performance characteristics to the low-end servers shown in the table. Suppose the virtual cluster consists of N nodes in addition to the ResourceManager and NameNode. Then the cost of an equivalent physical cluster would be $N \times \$982 + (N+2) \times \$810 + \$2,272$, as shown in the table. Thus, whenever $N \geq 3$, a virtual cluster is cheaper than an equivalent physical cluster.

Assuming three years of use before significant upgrades, the cluster may be used across six academic semesters. Further assume that each class involves 20 students, as is common for smaller colleges and universities, and the curriculum includes five big data projects. Finally, assume each student submits an average of five processing jobs per project. This amounts to 500 jobs submitted per semester, or 3000 jobs over the lifetime of the cluster. Thus, the cost per job for our proposed server is \$2.82. Assuming a 10-node physical cluster, the cost per job is \$7.27. In Section 5, we will compare these numbers with the average cost per job for Amazon Elastic MapReduce under various configurations.

The potential cost benefits of a virtual cluster over a physical cluster should be considered against their relative performance characteristics. We were not able to evaluate the performance of a physical cluster. However, the following section addresses performance evaluation of virtual clusters.

Proposed server:	
Dell R430	\$7,400
Power for three years:	\$1050
Cluster of low-end servers:	
Dell R220 for ResourceManager	\$982
Dell R220 for NameNode	\$1,086
Dell R220 for N nodes	$N \times \$982$
Power for three years for all servers	$(N+2) \times \$810$
Network switch	\$204

Table 1: Costs for proposed virtual cluster server and alternative physical cluster hardware configurations. Configuration specifics follow. Dell PowerEdge R430: Intel Xeon E5-2640, 128 GB memory, hard drives totaling 28 TB; Dell PowerEdge R220: Intel i3-4150, 8 GB memory, 2 TB hard drive; or 16 GB memory for NameNode; Power usage of proposed server estimated at 400 watts, and each low-end server estimated at 300 watts [1]; Power costs estimated assuming US national average electricity rate of \$0.1009/kWh; Network switch: 1 Gbit Cisco 200 series. Prices accurate as of June, 2015.

4. PERFORMANCE EVALUATION

We measured the performance of virtual clusters under various loads and with various node counts N. Such an evaluation can tell us the number of nodes that maximizes job throughput under various loads. With this information, the virtual cluster can be configured with the ideal number of nodes and the Hadoop scheduler configured to support a certain maximum number of simultaneous jobs. We expect that these optimal values depend on the specific hardware and virtualization platform.

Our evaluation consisted of executing a number of equivalent jobs on virtual clusters with various node counts. The job was a typical word count on a 2 GB text file. The text file was uploaded to HDFS as multiple copies to ensure the chunks of each copy were evenly distributed across the nodes. This ensures that the jobs are executed on the widest distribution of nodes, i.e., those that hold chunks of the input text, rather than all executing on the same node.

Figure 1 shows the results of our evaluation. We see that the general trend is low throughput for few simultaneous jobs, then a peak, followed by a modest decline in throughput. This trend indicates that a cluster's resources are not fully utilized until a modest number of simultaneous jobs are submitted. This peak appears to occur around 4-6 jobs in our tests. When more jobs are submitted, and resources are overstressed, one of two contingencies may occur. Jobs may execute in parallel and consume each other's resources, slowing the execution of all jobs. Or, jobs may be postponed by the scheduler until sufficient resources become available. We observed both of these cases occurring, sometimes simultaneously.

A second trend to notice is that $N = 15$ or $N = 20$ is better than $N = 10$ in terms of overall throughput. With fewer nodes, jobs are forced to share a smaller pool of resources. Our host server has 32 CPU cores, so $N = 30$ nodes or so should be feasible before CPU utilization reaches its maximum capacity. As N increases, the number of simultaneous disk input/output operations increases, which reduces overall performance, since there is only one disk (or one array of disks). There is evidence of this phenomenon at 12 simultaneous jobs. A configuration with 10 nodes handles 12

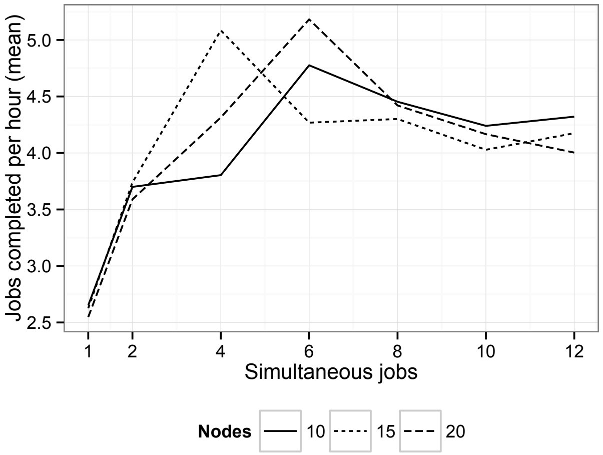

Figure 1: Jobs completed per hour (throughput) for a range of counts of simultaneous jobs and nodes (labeled N in the text). The tests were performed on the virtual cluster described in the text. Throughput was measured as the mean of three tests in each case.

simultaneous jobs better than 15 or 20 nodes because some of the jobs are placed in a queue, waiting for free resources on one of the 10 nodes. Even with this waiting period, jobs complete faster because there is less contention for the disk. Each node was given 8 GB of memory (except the NameNode, which was given 16 GB of memory), but our word count job did not consume enough memory to make memory capacity a limiting factor.

5. COMPARISON WITH AMAZON EMR

Cloud services provide an alternative to housing one or more servers on-premise. In this section, we evaluate Amazon's Elastic MapReduce (EMR) service, though alternatives such as Microsoft's Azure HDInsight and Google's Cloud are available. Elastic MapReduce is capable of executing normal Hadoop jobs on a configurable number of cluster nodes. Furthermore, the nodes may be one of several types with different performance characteristics and costs. We evaluate the m1.medium, m1.large, and m1.xlarge types.

The cost of a processing job is calculated based on the duration of the job multiplied by the number of nodes and the cost of the node type. For example, suppose a job requires 35 minutes of processing time, and 10 nodes of type m1.large are utilized. One extra node (of the same type) is needed to manage the Hadoop job. The cost of an Elastic MapReduce m1.large node is $0.197 per hour (at the time of writing), which is calculated by adding the cost of an m1.large node ($0.175/hour) and the cost of Elastic MapReduce ($0.022/hour). Thus, the cost of the job is $35/60 * 11 *$ \$0.197 = \$1.26. In fact, processing times are rounded up to the nearest hour, so the actual cost is $60/60 * 11 *$ \$0.197 = \$2.17. Our costs reported below are not calculated by rounding-up to the nearest hour; rather, the exact number of minutes required to complete the job is used in our calculations.

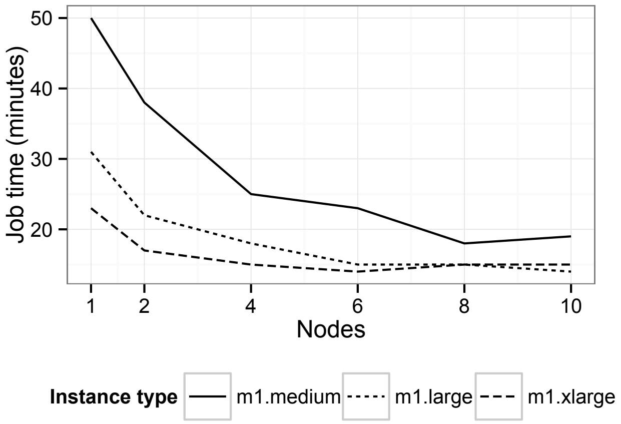

Figure 2: Processing time for a single job under various configurations of Amazon Elastic MapReduce jobs. The job submitted was a word count on a 2 GB text file.

Platform	Job time	Job cost
Virtual cluster	334 min	$2.82
Physical cluster	–	$7.27
Amazon EMR, m1.medium	101 min	$2.02
Amazon EMR, m1.large	56 min	$2.02
Amazon EMR, m1.xlarge	34 min	$2.32

Table 2: Processing times and costs (estimated for virtual and physical clusters) for a word count job on a 37 GB text file. In each case, the cluster is configured with 10 nodes. We did not build and test a physical cluster.

We executed the same 2 GB word count job on the three different node types with varying number of nodes. As expected, the more expensive node types (e.g., m1.xlarge) performed better than the cheaper types (e.g., m1.medium). Figure 2 shows the general trend as the number of nodes increases.

Since the job cost is partly calculated according to the processing time, we expect that node configurations that complete a job faster are more expensive. Figure 3 shows exactly that trend. Thus, it is clear that for cloud services, faster and cheaper are competing desiderata.

We can directly compare performance and costs of the virtual cluster solution and Elastic MapReduce by submitting a larger job. In this case, the job is a word count of a 37 GB text file. The processing times and costs are shown in Table 2. Note that we did not build and test a physical cluster, so we do not know the exact processing time for such a configuration. Conclusions that may be drawn from this table are discussed in the following section.

6. DISCUSSION

Our evaluation of a single server hosting several virtual machines, an equivalent physical cluster, and Amazon's cloud service yields several interesting insights. First, a physical

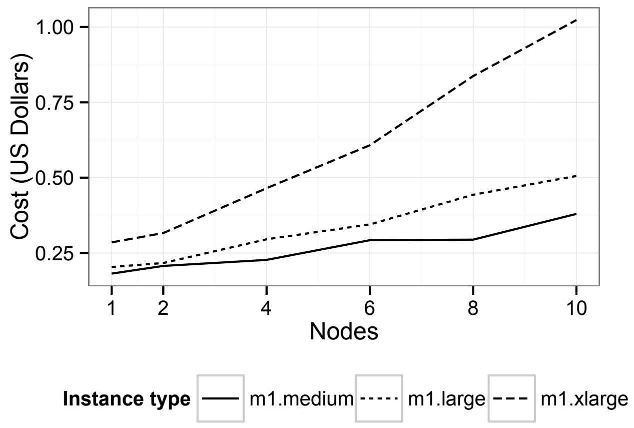

Figure 3: Costs for a single job under various configurations of Amazon Elastic MapReduce jobs. The job submitted was a word count on a 2 GB text file.

cluster is significantly more expensive, per job, than either a single equivalent server or Amazon's offering. Second, Amazon offers a very competitive Hadoop platform both in terms of price and performance.

Before we consider Amazon's service, it is important to point out the Achilles' heel of the virtual cluster solution. The virtual cluster suffers from a performance limitation that is not present in a physical cluster or a cloud platform. Hadoop is designed to make use of the resources of distinct physical machines. In particular, it assumes that when machine A reads or writes data, these operations do not impact machine B's ability to read/write its data. This is because HDFS stores data on each machine and only asks each machine to process the data that it stores. However, in a virtual cluster, there is only one physical hard drive, or array of drives that share a single input/output channel. When virtual machine A accesses data, virtual machine B may need to wait for the input/output channel to become available. Disk input/output becomes a bottleneck in virtual clusters. This bottleneck is especially apparent when processing the 37 GB text file, shown in Table 2, and less apparent when processing the 2 GB text file. When processing the larger text file with more than 10 virtual nodes, I/O contention was so prevalent that some nodes failed to receive any data in a reasonable time and were abandoned by the Hadoop ResourceManager.

Amazon's cloud service is both cheaper and more performant than our virtual cluster solution. We expect similar results from Microsoft's and Google's cloud offerings. This makes cloud services very compelling for computer science education. However, the cost model is fundamentally different. Cloud services are paid for on a per-job basis. If students are required to pay for their own jobs, we expect that they will be reluctant to submit any jobs due to fear of job failure or wrong answers, which require correcting and submitting new jobs. Costs may also accumulate due to accidental failure to terminate the nodes. If a student forgets to turn off 11 m1.large nodes (10 processing nodes plus the Hadoop manager node) over the course of a weekend (48 hours), the student will be presented with a $104.02 bill. Further work is needed to determine how students approach an assignment when out-of-pocket costs are involved.

7. EXPERIENCES IN THE CLASSROOM

Twenty students made use of the virtual cluster. They authored, submitted, and monitored Hadoop jobs, uploaded and downloaded data from HDFS, executed Hive queries, and trained machine learning models with Mahout.

Five projects involved the virtual cluster.

1. A typical word count introductory exercise.
2. "Analyze BackBlaze's hard drive monitoring data[2] to determine if some group of hard drives failed more often than others." The dataset was 4.2 GB in size.
3. "Analyze StackExchange's entire database of questions and answers[3] to determine if persons with high reputation typically answer questions faster." The dataset was about 116 GB in size.
4. "Use image processing and clustering to determine if a dataset of 10,000 cat images is comprised of cats of distinct breeds, based on the cat's colors." The dataset was 4.2 GB in size.
5. "Build a spam classifier model for 75,419 email messages in the TREC 2007 dataset."[4] The dataset was about 0.7 GB in size.

Depending on the project, students crafted MapReduce jobs, wrote Hive queries, and/or executed Mahout commands. In many cases, they were not required to use these tools, but chose to out of necessity or convenience. We wanted students to gain an understanding of why various tools exist, even Hadoop itself. They came to these realizations by discovering that they could not easily solve a problem, such as processing huge datasets without splitting the task with Hadoop, or merging user data and question/answer data in the StackExchange project without SQL-like table joins with Hive.

The system proved very reliable and was never unavailable due to load or system failure. However, we did observe some drawbacks during the course. The fair scheduler described above was not installed at the time, so jobs often waited in a first-in, first-out (FIFO) queue for quite some time. Students found it hard to predict the amount of time a job would take, partly due to the FIFO queue but also due to their inexperience with large processing tasks. Extensive benchmarks, like those presented above, were not performed until after the course ended. Some *ad hoc* adjustments to the number of nodes N were performed during the course to better balance the computational load. These changes disrupted some jobs and could have been avoided with better planning.

Student comments that relate to the virtual cluster and big data tools are shown below. Responding to the question, "Which aspects of this course helped you learn the most?" some students wrote,

[2] `https://www.backblaze.com/hard-drive-test-data.html`

[3] `https://archive.org/details/stackexchange`

[4] `http://plg.uwaterloo.ca/~gvcormac/treccorpus07/`

- "The practice with the various tools and software used in the class."
- "Hands on experience with current trends in the industry."

Those two comments give us reason to believe that the virtual cluster was a success. However, responding to the question, "Do you have suggestions for improving any aspects of the course?" one student wrote,

- "Add additional servers for student use."

More computational power would certainly reduce processing times and increase student satisfaction with the course and tools. However, more investigation is required to find the right tradeoff between budgetary concerns and additional student resources. We also note that these three comments are not necessarily representative of the entire class. In the future, we plan to explicitly ask students about their experiences with the virtual cluster.

8. RELATED WORK

Virtual clusters designed for student use have been developed by computer science departments in several small institutions. For example, St. Olaf College built both virtual and physical clusters using the KVM and the Beowulf platforms [3, 4], respectively. They describe the clusters' architectures and use cases. They also look at energy usage and the costs of cooling a room full of servers. In a more recent report by Brown and Shoop [2], they report that virtual clusters are less performant than physical clusters. A precise characterization of the performance penalty of virtualization depends on the workload. Brown and Shoop also discuss issues of cluster management. They do not compare physical and virtual cluster costs.

Ngo et al. [5] report on their increasingly more sophisticated clusters for teaching about big data with Hadoop. They do not address the costs of building a cluster nor analyze its performance characteristics. They report significant learning on several dimensions and high student satisfaction with their hands-on big data experiences.

Finally, Rabkin et al. [6] review their experiences teaching with cloud clusters provided by Amazon Web Services. A cloud cluster is an on-going expense, unlike a physical or virtual cluster which is a fixed expense. Their work was supported by a grant, so the costs were not passed on to the students. They find that students utilized an average of $45 worth of cloud resources during the semester. However, the efficiency of a student's solution could impact their costs. Even subtle features of code like object creation, string parsing, etc. could produce a substantial effect when processing big data. Some students also had a habit of leaving cloud services running unnecessarily, incurring costs without actually making use of the resources. Overall, the authors conclude that cloud services are a good platform for very large class sizes but costs are difficult to estimate and control.

9. CONCLUSION AND FUTURE WORK

This report summarized our experiences developing and using a virtual cluster for a big data course at a small private liberal arts college. Because we were able to upgrade existing equipment, we found that the virtual cluster design was cost-effective and sufficiently performant for a class of twenty students. Students expressed mostly positive feedback about the virtual cluster, though at least one student felt that the virtual cluster did not provide enough computational resources. The most significant drawback of the virtual cluster design is I/O contention, which dramatically lowers performance on large processing jobs. Experimentation with Amazon's Elastic MapReduce cloud service showed that cloud services can be cheaper and more performant than a virtual cluster solution. However, costs for cloud computing are charged on a per-job basis, which may fundamentally change how students decide to submit a job if they are responsible for the costs.

In future work, we plan to evaluate a wider variety of workloads, such as Hive queries and Mahout machine learning tasks. We are also considering ways in which to bring cloud services into the classroom in a manner that is least impactful to students in terms of costs.

10. ACKNOWLEDGEMENTS

This project was supported by the Brown Center for Faculty Innovation & Excellence at Stetson University.

11. REFERENCES

[1] S. Barielle. Calculating TCO for energy. *IBM Systems Magazine: Power*, pages 38–40, November 2011.

[2] R. Brown and E. Shoop. Teaching undergraduates using local virtual clusters. In *IEEE International Conference on Cluster Computing (CLUSTER)*, pages 1–8. IEEE, 2013.

[3] R. A. Brown. Hadoop at home: Large-scale computing at a small college. In *ACM SIGCSE Bulletin*, volume 41, pages 106–110. ACM, 2009.

[4] E. Johnson, P. Garrity, T. Yates, R. Brown, et al. Performance of a virtual cluster in a general-purpose teaching laboratory. In *IEEE International Conference on Cluster Computing (CLUSTER)*, pages 600–604. IEEE, 2011.

[5] L. B. Ngo, E. B. Duffy, and A. W. Apon. Teaching HDFS/MapReduce systems concepts to undergraduates. In *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*, pages 1114–1121. IEEE, 2014.

[6] A. S. Rabkin, C. Reiss, R. Katz, and D. Patterson. Experiences teaching mapreduce in the cloud. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, pages 601–606. ACM, 2012.

[7] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The Hadoop distributed file system. In *IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–10. IEEE, 2010.