Teaching Big Data and Cloud Computing: A Modular Approach

Debzani Deb
Department of Computer Science
Winston-Salem State University
Winston-Salem, USA
debd@wssu.edu

Sebastian Cousins
Department of Computer Science
Winston-Salem State University
Winston-Salem, USA
scousins116@rams.wssu.edu

Muztaba Fuad
Department of Computer Science
Winston-Salem State University
Winston-Salem, USA
fuadmo@wssu.edu

***Abstract*—Big data and cloud computing collectively offer a paradigm shift in the way businesses are now acquiring, using and managing information technology. This creates the need for every CS and IT student to be equipped with foundation knowledge in this collective paradigm and to possess some hands-on-experience in deploying and managing big data applications in the cloud. We argue that for substantial coverage of big data and cloud computing concepts and skills, the relevant topics need to be integrated into multiple core courses of CS/IT curriculum rather than creating additional standalone dedicated core or elective courses. Our approach to including these topics is to develop learning modules and to suggest specific core courses in which their coverage might find an appropriate context. In this paper, two such modules are discussed and our classroom experiences during these interventions are documented. Specifically, we discuss the learning outcomes, module contents, their designs and implementations, student assessment results and the lessons learned. Our objective is to share our experience with the instructors who aim at incorporating similar pedagogy that enhance student knowledge on this collective paradigm.**

Keywords-big data; cloud computing; curriculum; mapreduce

I. INTRODUCTION

Big data and cloud computing collectively offer a paradigm shift in the way businesses are now acquiring, using and managing information technology. With the fast growth of this paradigm, businesses are struggling to find experienced people who not only have the deep analytical skills, but also have the data hosting, storage and management skills to effectively leverage this collective model. This observation is supported by the recent prediction made by the International Data Corporation (IDC), where the forecast is that by 2020 big data staffing shortages will expand from analysts and data scientists to include architects and experts in data management. With an estimated number of 50 billion devices that will be networked by 2020, Internet of Things (IoT) will only intensify this demand as it will be one of the main sources of big data, and the cloud will be an enabler for storing it for a long time and for performing complex analysis on it. Increasing adoption of this collective paradigm in solving problems from a variety of domains is also making studying and performing research on this paradigm crucial.

We believe that every CS and IT student should be equipped with foundation knowledge in this collective paradigm and should possess some hands-on experience in deploying and managing big data applications in the cloud to acquire skills that are necessary to meet current and future industry demands as well as to enable them to carry out applied research on this paradigm. However, the challenge is that many of the tools and techniques of the big data and cloud computing paradigm have emerged only in the last few years and have not yet transitioned into the most recent ACM/IEEE Joint Curriculum recommendations [1] or the ABET curriculum requirements. Many 2-year and 4-year institutions develop their CS curriculum around these guidelines and requirements and as a result cannot afford to include these contemporary topics as required (core) courses in their densely packed curricula. A number of institutions are now offering non-core specialized courses [2, 3, 4] to cover a variety of aspects of data science and big data analytics, where students are primarily taught data acquisition, cleaning, analytical and visualization skills. While these courses help students in developing skills related to transforming data into knowledge, it does not provide them with concepts and experiences related to hosting, storing, deploying and scaling up applications within performance and budgetary constraints. A few research-intensive universities offer specialized standalone courses [5, 6, 7, 8] such as “Cloud computing”, “Big data management” etc., where the abovementioned collective paradigm is addressed in a greater extent. However, being a special topic course and offered at a handful of universities, only a small number of students receive the benefit. There is a big gap between the advances in big data and cloud computing and their inclusion in college-level instructions and this paper aims to address this gap.

We argue that for substantial coverage of big data and cloud computing concepts and skills, the interventions need to be gradual and should be integrated into multiple core (required) courses of CS/IT curriculum. Our approach to including these topics is to develop learning modules with specific learning goals, lessons plans, and assessment resources and to suggest specific core courses in which their coverage might find an appropriate context. The assessment resources include proficiency questions, sample programs, in-class hands-on exercises, cloud-based resources, projects

978-1-5386-5555-9/18/$31.00 ©2018 IEEE
DOI 10.1109/IPDPSW.2018.00070

etc. Our goal is not only to teach students the techniques and tools for big data analytics, but also to enrich them with the understanding of the tradeoffs between performance and cost that is a significant aspect of the abovementioned collective paradigm, and the knowledge of which is very necessary while scaling up computations.

In this paper, two such modules are discussed and our classroom experiences during these interventions are documented. The first (Introductory) module is designed for a CS0 course and the focus is on understanding the data and computation at scale, learning the basics of cloud computing and big data analytics, and gaining a complete picture of the application of this collective paradigm by experiencing an in-class demonstration of running an analytics task on the Amazon EC2 cloud. The second (Intermediate) module is designed to be included within an Algorithm class and the emphasis is on understanding the MapReduce programming framework, making use of Apache Hadoop, HDFS, MapReduce and Apache Spark in designing and deploying applications, and understanding the cost vs. performance tradeoff by using various input data sizes and by utilizing different execution setups in chameleon [9] cloud. For both courses, the student comprehension and satisfaction were evaluated by using carefully designed assessment instruments and pre- and post- survey questions.

In this paper, we report on our experiences in offering the proposed two modules in respective courses. More specifically, we discuss the learning outcomes, module content, its design and implementation, student assessment results and the lessons learned. Our objective is to share our experience with the instructors who aim at incorporating similar pedagogy that enhance student knowledge on this collective paradigm.

II. INTRODUCTORY MODULE

The introductory module is designed for the students of the CS0 course with the goal of introducing the basic big data and cloud computing concepts to the freshman/sophomore students earlier in their curricula. The next few sections detail the learning outcomes, lesson plan, student assessment instruments, and the results of the classroom deployment.

A. Learning Outcomes

1. Understand the need for scalability and the role of parallel and distributed computing to achieve this (LO1).
2. Recognize the key properties, techniques, strengths and challenges of big data and cloud computing (LO2).
3. Become well-versed and grow further interest in these topics (LO3).

B. Lesson Plan

The module was assigned at the last quarter of the course and two (2) 75-minute classes (one week) were devoted for discussing the related topics. The first class focuses on two concepts as detailed as follows

- Computing at Scale: The applications mostly utilized by college freshmen such as Netflix, Facebook, Google, Youtube are explained, emphasizing how many objects and users they contain and how much data they are dealing with on an everyday basis. The lecture then focuses on the need for parallel processing in order to manage so much data and users and perform computations on them. The scaling up of the infrastructure from PC to Server to Cluster to Data Center and finally to the Globally Distributed Networks of Data Centers are discussed at this point to give students some idea about the magnitude of computation.
- Cloud Computing: The lecture on this topic starts with the well-known "power plant" analogy and explains the important cloud computing characteristics such as scalability, on-demand access, measured service, and elasticity in the context of that example. The lecture then continues with the concept of "services" and the kinds of services (such as SaaS, PaaS and IaaS) that the cloud provides. The lecture also clarifies the distinction among public, private and community clouds. The lecture then spends a good amount of time explaining "virtualization" with an easy to comprehend visual example while pointing out the important aspects of it such as Migration, Time-sharing, Isolation etc. Finally, the lecture explores the benefits of utilizing cloud services within a business and the challenges associated with adoption such as data confidentiality, performance unpredictability, etc.

The second class discusses the following

- Big Data Computing: The lecture starts with the definition of big data computing and discusses its application in the domains that the students are mostly familiar with such as Netflix and Amazon recommender systems, use of big data insights in sports, weather prediction, medical diagnosis, etc. The lecture also briefly discusses the 3 V's of big data such as volume, variety, and velocity while emphasizing the recent proliferation of varieties of data such as real-time, streaming, etc. Big data tools and techniques are then explained with easy to comprehend real-life examples, and big data systems and platforms such as MapReduce and Hadoop are discussed very briefly.
- Demonstration on AWS and Data Visualization: The instructor demonstrates querying page view statistics data for Wikipedia projects [10] using Apache Hive in AWS EC2 using S3 as storage. Interesting visualizations based on 4TB of page view data are pre-created and shared in the classroom to answer some interesting questions such as "How many times in each month Trump or Clinton were searched for" (the module was deployed during the month of the 2016 presidential election). There is no expectation that CS0 students would be able to create and run their data analytics job in the cloud after completing the module. However, this demonstration provides an end

to end perception of the collective paradigm and allows students to experience real-life applications of the concepts and techniques that are discussed earlier in this module.

C. Assesment Instruments

Pre- and post- tests that include ten factual multiple-choice questions and three opinion questions are used as assessment instrumentations to ascertain, if, in the short-term, the learning outcomes of this module are being achieved. The ten knowledge questions (multiple choice) focus on gauging the progress students have made retaining the concepts such as scalability, parallelism, and the key properties, techniques, strengths and challenges of big data and cloud computing. The same questions appear in both pre-quiz (at the beginning of the first class) and post-quiz (at the end of the second class), the differences in the scores of the two tests provide a rather straightforward measurement of the impact of the module. The ten knowledge questions are used to assess LO1 and LO2. On the other hand, the three opinion questions are utilized to evaluate LO3. The students provide their opinions about the three statements using a Likert scale of five values such as Strongly Agree, Agree, Neutral, Disagree and Strongly Disagree. The opinion questions are as follows

- O1 - I found the topics big data and cloud computing interesting.
- O2 - If a friend asks me what big data and cloud computing are, I will be able to explain for 2-3 minutes.
- O3 – I would like to learn more about big data and cloud computing and would like to explore more in my future courses.

D. Results & Lessons Learned

The introductory module was deployed in two semesters (Fall 2016 and Fall 2017) in the CS0 course (CIT 1307: Introduction to Information Technology) at Winston-Salem State University (WSSU). All students were IT majors and were mostly freshman. Table 1 shows descriptive statistics for the grades (out of 10) that the students attained while answering the pre- and post- knowledge questions. The results indicate that there were significant differences between students' pre- and post- tests performances. The results also indicate that the students' pre-intervention knowledge on the concerned topics were mostly uniform (reflected in the lower standard deviation value for both semesters), whereas, after intervention, it varied more widely. These results show enhancements in students' short-term knowledge acquisition on the concerned topics.

Based on the F16 classroom experience, the instructor made a few updates and they were deployed during F17 classroom implementation. For example, F16 intervention presented pre-created visualizations based on 4TB datasets as described in section II.B, however did not encounter the demonstration on AWS based on a smaller data set. During F17 deployment, the demonstration was included in the

TABLE 1: Descriptive statistics of pre- and post- test scores

	F16-Pre (N: 11)	F16-Post (N:13)	F17-Pre (N: 10)	F17-Post (N:10)
Average	2.91	6.85	1.7	7.4
Median	3	7	1	8
Std. Deviation	1.26	2.31	1.25	2.59

module to furnish students with an end-to-end example. The other topics of the modules were also upgraded to include more examples, scenarios, environments that freshman students are likely to use in their daily lives. The assessment section was revised in F17 to include the choice "I do not know" in the knowledge questions of the pre-quiz (only) as previously (in F16) students were forced to choose an answer for the knowledge questions even though they do not have any idea about the specific question or the answers.

Both pre- and post- tests include the same three opinion questions (described in section II.C), where Opinion question 1 (O1) asks about the topic being interesting, O2 inquiries about being well-versed on the topic and O3 asks about a desire to learn more. Figure 1 and 2 shows the comparison of the pre- and post-results for the three opinion questions during F16 and F17 semesters. The reader should note that during F16 semester, a different number of students completed the pre- (N:11) and post-(N:13) survey and therefore the 100% stacked column chart shown in Fig. 1 could be bit misleading at first. Therefore, the numbers of responses are added as data labels in each case to clearly show the survey results. In general, more students strongly agreed or agreed to all three questions after the interventions

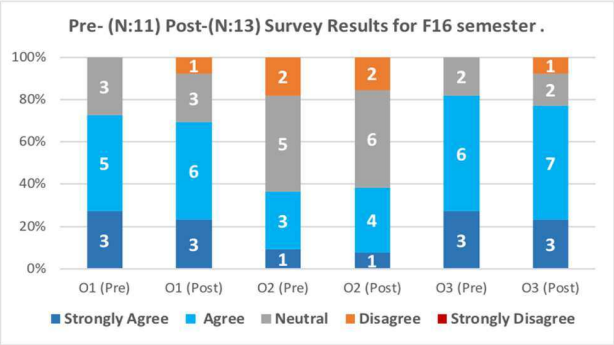

Figure 1. Students' pre- and post- survey results for F16.

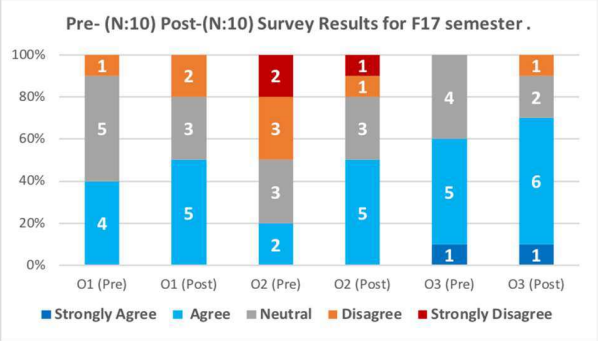

Figure 2. Students' pre- and post- survey results for F17.

than before the interventions for both semesters. However, F16 results show marginal enhancements on agreement, whereas the F17 students exhibit slightly better agreement during post- survey. Most notably, during F17 intervention (Fig. 2), only 20% of the students agreed about being well-versed on the concerned topics during the pre- survey, whereas about 50% of the students confirmed their agreements during the post-survey. Similarly, about 10% more students agreed that they found the topics interesting and would like to explore more of them after the intervention. There are also notable disagreements in post-surveys, which means that more students developed reservations after taking the module than before taking it, possibly due to many of them being exposed to what learning big data and cloud computing involves for the first time and recognizing the extra effort required.

Overall, the results of the pre- and post- tests show that the module is effective in attaining the learning outcomes. Results in Table 1 reveal enhancements in students' short-term recall of the covered concepts (LO1 and LO2) and more students agreed on being well-versed and growing interests (LO3) after the intervention. It is clear, that the maturity of the students (most of them being freshman) was an important factor and the module would benefit more from including contents or assessments that are suitable for their level and from systematic evaluations of the classroom deployments and from ongoing updates based on the evaluation results. The marginally enhanced F17 post- test grades and post- survey results suggest the value of the updates that were made based on F16 classroom experiences and evaluation and verify the importance of performing continuous evaluation and updates. Instructors planning to offer this module must also be well-prepared (i.e the demonstration on AWS) and organized in the classrooms to utilize their time in the best possible way.

III. INTERMEDIATE MODULE

This module is designed for the students of the Algorithm course with the goal of introducing the popular cloud analytics engine (such as Hadoop and Spark) to the junior CS majors. This module is targeted for the students who have completed CS1, CS2, and a data structure course and are comfortable with and reasonably proficient using programming language such as Java or Python. The lecture slides from the first (introductory) module are provided to the students as reading material in order for them to gain basic ideas about big data and cloud computing. The next few sections detail the learning outcomes, lesson plan, student assessment instruments, and the results and the lessons learned from the deployment.

A. Learning Outcomes

1. Recognize the key properties, techniques, strengths and challenges of MapReduce and Spark Framework (LO1).
2. Build scalable applications based on MapReduce programming model using Hadoop and HDFS. (LO2).
3. Develop basic experience with big-data analysis on cloud platforms with performance and cost constraints using Spark/Hadoop platforms (LO3).

B. Lesson Plan

The module was assigned at the last quarter of the course and two (2) 75-minute classes (one week) were devoted to discussing MapReduce and Spark and providing hands-on-experience to the students. The first class is lecture-based and covers the following contents.

- MapReduce Programming: The first class explores parallel computing at the beginning and introduces MapReduce as a framework that can quickly process large data sets by splitting them into individual chunks that are processed in parallel. The concept of key-value pair is introduced and the *map, shuffle* and *reduce* phases are then explained. The classic WordCount application is illustrated followed by explaining a complete Java implementation. Since learning how to divide an entire computation into multiple *map* and *reduce* tasks is the essence of designing MapReduce programs, at this stage, the lecture spends a good amount of time showing students how this breakdown occurs in the context of other examples such as "find the frequency of each URL in a weblog", "find what documents contain a specific word", etc.
- Hadoop and HDFS: Apache Hadoop as an implementation of MapReduce model is introduced next along with its distributed file system HDFS. The concept of a Hadoop cluster along with the master-worker framework is briefly explored.
- Apache Spark Framework: This last section of the lecture very briefly discusses the problems with the classic MapReduce programming and emphasizes the need for an in-memory analytic engine such as Apache Spark. The lecture then very briefly discusses Spark's runtime distributed architecture including driver, executors, and directed acyclic graph (DAG).

The second class, on the other hand is a lab-based class where the instructor provides some demonstrations and a tutorial for the students to follow. The goal of this lab session is to provide hands-on experience in using Hadoop, HDFS, Spark and cloud environment as this knowledge is crucial to complete the hands-on project part of the module. The demonstration (and the tutorial) covers the following steps

1. Preparing Hadoop Work Environment with VirtualBox and Cloudera's VM: One of the goal of this module is to provide students with some skills in MapReduce implementations and to gain that skill they need to have continuous access to a development environment. We recommend the use of Cloudera's VM [11] in their local machines, which has all the necessary packages already installed and configured properly. That way, students with minimal Linux background can start focusing on

coding from the beginning rather than spending time on configuring and troubleshooting.

2. Compiling and executing MapReduce Applications: This part of the tutorial provides students with step by step instruction of setting up HDFS, compiling and executing a MapReduce application, and retrieving output from HDFS. All the Java files (driver, mapper and reducer) and input datasets are provided, and the students are able to understand the various stages of a MapReduce job and its execution while following the steps with an application that finds the year-wise maximum temperature (MaxTemperature.java). Students then reinforce their recently learned skills while executing a second WordCount application (WordCount.java) that works with a rather large dataset.
3. Being Familiar with Chameleon cloud environment and executing Spark-SQL application in it: This particular step introduces students to Chameleon cloud and shows them how to ssh to a particular Chameleon instance (all students are registered and added to our Chameleon project before the tutorial starts). We implemented a Spark on YARN cluster (with HDFS) on the Chameleon Cloud and the compiled Spark-SQL application that the students use for experimentation along with the dataset are pre-loaded to that instance. The application finds the trending topics given the Wikipedia page views information [10]. During the lab session, students are taught how to execute a Spark application in Yarn cluster by using the *spark-submit* command, and how to configure cluster resources for its execution by varying parameters such as *--num-executors*, *--executor-memory*, and *--executor-cores*. Students are also taught to verify current resource allocation and to check the execution time and other performance metrics of a spark application by using Spark's web user interface (UI).

C. Assesment Instruments

This module is assessed by utilizing both hand-on project and questions that appear in the final exam for this course. There are four multiple-choice questions and two design questions developed to assess students' comprehension of Hadoop/Spark framework and the MapReduce programming model. In the design questions, students are asked to write pseudocode of map and reduce functions (or a series of them) for certain cases. The hands-on project, on the other hand, is built on top of the lab session (second class) and includes the following tasks:

Task 1: Modify the MaxTemperature.java (Section III.B), so that it produces Average Temperature of each year instead of Maximum Temperature.

Task 2: Modify the WordCount.java (Section III.B), so that it outputs the number of words that start with the letters 'a', 'b' and 'c'.

Task 3: Find the attached OrderDB.txt file where each line records an order in the form {Order-ID, Customer_id, Order_date, total}. Write a MapReduce program that outputs the total amount spent by each customer considering all her orders.

Task 4: This step of the project requires students to explore spark application performance in the cloud environment through running them with various runtime configuration settings, and to gain some insight about the resource provisioning and the performance vs. cost tradeoff. Students are presented with two Wikipedia data sets (100GB and 200GB) and are provided with two Spark clusters (one-node cluster and two-node cluster, each compute node with 24 cores, 128 GB memory) in the Chameleon testbed. Students are asked to run the trending Wikipedia spark application with the given two input datasets while trying various configuration setups for both clusters. Although use of Chameleon is free of charge, we introduce a basic cost model to the students (i.e. 1 service unit = 1 core with 1 GB memory) and ask them to compare different cluster configurations and gain some insight about performance vs. cost tradeoff. Students are further asked to write a report detailing their experimental results and their findings along with their supporting arguments.

A short survey in pre- and post- form was administered as part of this module. The pre-survey opinion questions are very similar to the introductory module:

- O1 - I found the topic Parallel Computing and working in MapReduce and Spark framework interesting.
- O2 - If a friend asks me to explain MapReduce/Spark framework, I will be able to explain for 2-3 minutes.
- O3 – I would like to learn more about MapReduce and Spark framework and would like to explore more in my future courses.

In addition to the above questions, students are also asked the following three questions in the post-survey.

- O4 - I am able to recognize main properties, strengths and limitations of MapReduce and Spark Framework.
- O5 - I consider myself familiar with Hadoop and Spark Environment.
- O6 - I have enjoyed the hands-on experience related to MapReduce and Spark in this course.

D. Results & Lessons Learned

The intermediate module was deployed in the Spring 2017 offering of the Algorithm course (CSC 3331: Analysis of Algorithms) at WSSU. All students were CS majors and were mostly juniors. On average, students attained 68% on the four multiple choice questions in the final exam. For the two design questions, students' average scores were 62% and 38%. The second design question (with average score 38%) involves writing a series of *map* and *reduce* tasks; and while many students provided a partially correct answer, only a few figured it out completely. These results indicate that the students were able to retain concepts taught as part of the module and were able to answer both multiple choice and analytical questions successfully to a certain extent. After the rigorous lab session and many other individual

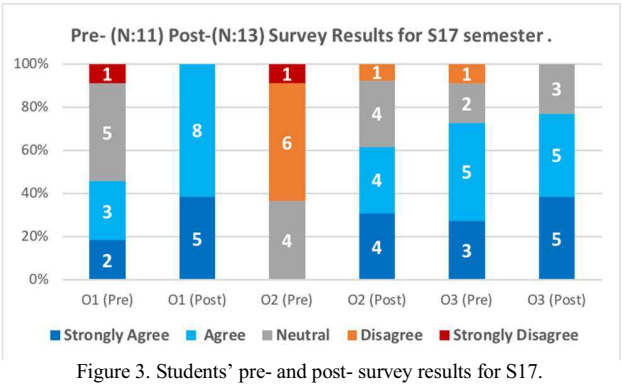

Figure 3. Students' pre- and post- survey results for S17.

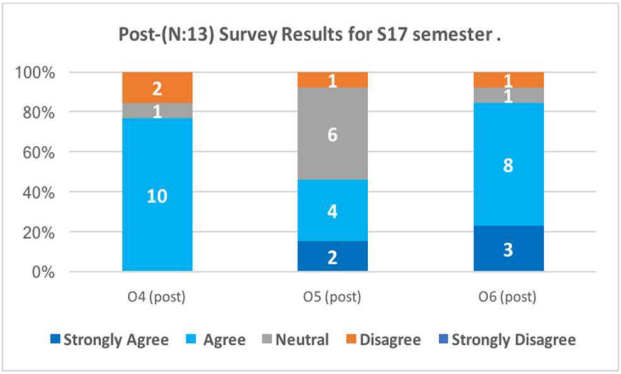

Figure 4. Post- survey results for S17

troubleshooting sessions with the instructor, all students were able to perform Task 1 and 2 (Section III.C) of the Project. Task 3 was successfully completed by 85% of the students and Task 4 was completed by 30% of the students.

Pre- and post- survey results for O1, O2 and O3 (Section III.C) are depicted in Fig. 3. These results show that 100% students found (strongly agreed or agreed) the conveyed topics interesting after the intervention whereas only 45% found them interesting beforehand. Similarly, 61% of the students strongly agreed or agreed that they are well-versed on the topics compared to none before the treatment. Students' desire to learn also increased from 72% to 77% after the module was taught in the class. Fig. 4 shows the survey results for the opinion questions O4, O5 and O6 which only appeared in the post-survey. According to these results, a majority (77%) of the students agreed that they were able to recognize the main properties, strengths and challenges of MapReduce and Spark framework (O4). On the other hand, only about 46% students were confident about the familiarity with Hadoop/Spark environment (O5). Lastly, about 85% of the students expressed that they enjoyed the hand-on experience provided via the lab session and project component of the module.

Overall, the student assessment and survey results justify the module and show its effectiveness in achieving the proposed learning outcomes. Student assessment results of the multiple choice and analytical questions in the final exam evidently supports students' in-depth understanding of the covered concepts (LO1). The results of the first three project tasks show student competency in developing simple MapReduce applications (LO2). The final task in the project exposed students to the cloud-enabled Spark environment and allowed them to understand runtime tradeoffs (LO3). Unlike the CS0 intervention, most students who utilized this intermediate module revealed significantly greater appreciation and growing interest after being treated, which is evident from the pre- post- survey results (Fig. 3). The higher number of agreements with the self-reflection questions (O4, O5, O6) in the post survey also demonstrated that the students valued this experience and felt somewhat comfortable with the Hadoop/Spark environment even though the duration of the intervention was very short. Several students volunteered to add anonymous comments in the post-survey that showed the usefulness of the module and their eagerness to spend more time on the topics. Two of such comments are as follows:

> "While I very much enjoyed the MapReduce/Parallel Computing topic, I felt rushed to complete the assignment and a little stressed. I wish we could have invested a little more class time along with a little more time to complete the project. Otherwise, I very much enjoyed doing this project and am very glad that we went over this topic."

> "I think that the programming is very interesting. Although I had challenges doing the project but I guess it is as a result of me not being used to Linux. But I would like to learn more about these concepts."

Development of this module content, lab and project related artifacts such as tutorial, compiled programs, cloud instructions, input datasets, and assessment instrumentations required a large time investment and substantial class preparation. Although the results are promising, during the current offering (Spring 2018) of this module we plan to make few updates to make it more relevant and more effective, such as 1) focus on Spark programming (some students were confused to experience two different frameworks within such a short time), and 2) utilize the AWS cloud as our testbed. Our class size was rather small and the instructor and her research assistant were able to support each student's needs in a timely manner. However, instructors who are planning to offer a similar module to a larger class must acquire enough TA resources for the duration of the module.

IV. CONCLUSIONS

The goal of this study is to explore the integration of big data and cloud computing modules into core undergraduate CS/IT courses and to evaluate its effectiveness. A substantial advantage of the modular approach is that a large number of CS/IT majors can be exposed to these

contemporary topics and technologies via systematic and increasing integration throughout the computing curricula, and without the need of developing an additional core or elective course. This paper presents two such modules and our classroom experiences while deploying them. The student-generated evidence based on student performance and survey data supports our pedagogy, inspires us to continuously assess and update our intervention, and allows us to extend our interventions across multiple semesters. Our experiences with the introductory module suggest that it is possible to introduce students to these important concepts earlier in their curriculum and that the students mostly were able to recognize the benefit of this early introduction and developed further interest in the topics. The intermediate module results clearly show that the students were able to relate to the topics very well, found them to be interesting enough to explore and to retain, and developed significant interest and confidence after the interventions. The maturity (Junior vs. Freshman) and the background (CS major vs. IT major) clearly impact the way students approach these topics initially, the enthusiasm that they handle them with, and the appreciation that they develop after being intervened. Interested readers are advised to explore resources posted at the project website [12].

Both modules were deployed at WSSU, an HBCU that serves unique group of students as 71% of its student population is female and 72% are African American. While the composition of the intervened classes approximates the similar minority student demographics, only about 20% to 30% of our class populations were female students. The modules are therefore carefully designed to incorporate pedagogies such as active learning, peer instruction, instructional scaffolding, etc. which are recognized by many research studies in addressing some of the challenges that underrepresented, and minority students typically face during their college years. One limitation of the study is the lower number of students impacted by the modules and further repetitions of the interventions are paramount to make stronger conclusions about their effectiveness.

In the future, we would like to develop similar modules for other core courses. For example, a module that discusses distributed and cloud computing architecture can fit in very well with the topics that are typically covered in a computer architecture course. We would also like to perform research on more gradual and systematic integration of the developed modules across the curriculum, and on assessing their collective effectiveness, rather than measuring the efficacy of a single module.

ACKNOWLEDGMENT

This research was supported by National Science Foundation Award # 1600864.

REFERENCES

[1] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society. 2013. Computer Science Curricula, 2013.

[2] B. Ramamurthy, "A Practical and Sustainable Model for Learning and Teaching Data Science", Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE), March, 2016.

[3] S. J. Matthews, "Using Phoenix++ MapReduce to introduce undergraduate students to parallel computing", Journal of Computing Sciences in Colleges, vol. 32 Issue 6, June 2017.

[4] J. Eckroth, "Teaching Future Big Data Analysts: Curriculum and Experience Report", IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), May, 2017.

[5] 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 (SIGCSE), 2012.

[6] M. S. Rehman, J. Boles, M. Hammoud, M. F. Sakr, "A Cloud Computing Course: From Systems to Services, Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE), 2015.

[7] CS5412: Cloud Computing, http://www.cs.cornell.edu/courses/cs5412/2018sp/

[8] CS309: Cloud Computing http://cs309a.stanford.edu/

[9] Chameleon cloud, https://www.chameleoncloud.org

[10] Wikipedia Page Views http://dumps.wikimedia.org/other/pagecounts-raw/

[11] Cloudera, Cloudera QuickStart VM https://www.cloudera.com/downloads/quickstart_vms/5-12.html

[12] http://ibigcloud.altl.org, 2018.