Journal of Parallel and Distributed Computing 157 (2021) 303–315

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

www.elsevier.com/locate/jpdc

Integrating big data and cloud computing topics into the computing curricula: A modular approach

Debzani Deb *, Muztaba Fuad

Department of Computer Science, Winston-Salem State University, Winston-Salem, NC, USA

ARTICLE INFO

Article history:
Received 1 November 2020
Received in revised form 23 February 2021
Accepted 18 July 2021
Available online 27 July 2021

Keywords:
Bigdata
Cloud computing
Module
CS curriculum
Competency-based learning

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 student to be equipped with foundational knowledge in this collective paradigm and possess some hands-on experience in deploying and managing big data applications in the cloud. This study argues that, for substantial coverage of big data and cloud computing concepts and skills, the relevant topics need to be integrated into multiple core courses across the CS curriculum rather than creating additional courses and performing a major overhaul of the curriculum. Our approach to including these topics is to develop autonomous competency-based learning modules for specific core courses in which their coverage might find an appropriate context. In this paper, four such modules are discussed, and our classroom experiences during these interventions are documented. Student performance data and survey results show reasonable success in attaining student learning outcomes, enhanced engagement, and interests.

1. Introduction

In today's world, the analysis of 'bigdata' becomes a high-priority task for many fields of study, and data-driven discovery and decision processes now guide many sectors of our business and economy. At the development level, analyzing bigdata requires proficiency in specialized algorithms and methodologies due to the fundamentally distributed and parallel nature of the workloads. In contrast, cloud computing skills are paramount at the infrastructure level to acquire a pool of virtual resources in a 'pay-as-you-go' fashion to deploy and manage these workloads. Being an applied field of Parallel and Distributed Computing (PDC), Bigdata 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. A recent report (September, 2020) from the Bureau of Labor Statistics (BLS) [31] indicated that hiring in data processing, hosting and related services will steadily grow in future years as reflected in the below quote:

> "Employment in computer and information technology occupations is projected to grow 11 percent from 2019 to 2029, much faster than the average for all occupations. These occupations are projected to add about 531,200 new jobs. Demand for these workers will stem from greater emphasis on cloud computing, the collection and storage of bigdata, and information security."

Alongside, the International Data Corporation (IDC) also emphasized the expansion of cloud-centric infrastructure and applications after the Covid-19 era, and the #1 item in their 2021 worldwide IT industry predictions [32] becomes "By the End of 2021, Based on Lessons Learned, 80% of Enterprises Will Put a Mechanism in Place to Shift to Cloud-Centric Infrastructure and Applications Twice as Fast as Before the Pandemic."

These statistics clearly exemplify that this collective paradigm will dominate the industry in the coming years, and there will be a severe shortage of skilled professionals to maintain the need and growth of such industries. Increasing adoption of this collective paradigm in solving problems from various domains is also making study and research on this paradigm crucial.

It is imperative that every CS and IT undergraduate student be equipped with foundational knowledge and competency in this collective paradigm. The students should be provided with hands-on experience in deploying and managing bigdata applications in the cloud to acquire skills necessary to meet current and future industry demands and enable them to carry out applied research

* Corresponding author.
E-mail addresses: debd@wssu.edu (D. Deb), fuadmo@wssu.edu (M. Fuad).

https://doi.org/10.1016/j.jpdc.2021.07.012

in this paradigm. However, the challenge is that many of the tools and techniques of the bigdata 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 (CS2013) [16] or the ABET curriculum requirements [1]. 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 topics within a core course in their densely packed curricula. Although, a newer version of ACM/IEEE Joint Curriculum recommendations (CC2020) [9] is undertaking a major expansion that includes these emerging topics, and so are other curriculum efforts such as Data Science Task Force [13] and NSF/IEEE-TCPP Curriculum Initiative on PDC [24], only the draft versions of these curriculums are currently available. Additionally, CC2020 will be a significant overhaul compared to CS2013 as the focus is shifting from knowledge-based learning to competency-based learning [33]. Therefore it may take years to experience broader adoption of the new curriculum providing meaningful guidelines about accommodating the critical bigdata and cloud computing topics into the CS curricula.

This study argues that for substantial coverage of bigdata and cloud computing concepts and skills, students need to be intervened more often, gradually, and the topics should be integrated into multiple core courses of the curriculum. Our approach to including these applied PDC topics is to develop a series of short, self-contained learning modules with specific learning goals, lesson plans, and assessment resources and suggest specific core courses in which their coverage might find an appropriate context. Each module covers topics of the collective paradigm in the context of a conventional core CS/IT course, thus enabling us to better expose students to the bigdata and cloud computing concepts without a significant overhaul of the curriculum. Many recent curriculum efforts such as IT2017 [29], MSIS2016 [30] embraced the concept of competency as the primary characteristics of curriculum definition as it describes the practical benefits of computing education to its stakeholders and the society more effectively. Competency provides a comprehensive perspective on education that enhances knowledge (knowledge: know-what) with its hands-on application (skill: know-how) persuaded by purpose (motivation; know-why) to realize a task. Inspired by this recent shift and considering the collective paradigm as an applied field where proficiency in learning must be demonstrated by both knowledge and skills, the presented work also adopts a competency model while describing the module outcomes. The presented modules are designed with a rich set of tutorials, sample programs, descriptions of cloud-based resources, and substantial hands-on projects to support the "skill" dimension of the competency frameworks. The "motivation" dimension is realized by augmenting the modules with relevant content and skills and placing them in core courses in which their coverage might find an appropriate context. Reusability and adaptability are other major concerns, and as a result, the modules are designed to be autonomous, made available for downloading in GitHub and public websites [20,35], and are detailed enough for easier adoption by other instructors with minimal experience. This paper details the modules' deployment in specific courses at our institution; however, they could be adapted for deployment in some other courses with minimal effort.

This paper reports on our experiences in offering four such modules in respective CS/IT core courses. We also discuss the competency outcomes, module content, assessment instruments, and student assessment and survey results for each of the four presented modules. Our objective is to share our experience with the instructors who aim to incorporate similar pedagogy that enhances student knowledge on this collective paradigm. Earlier versions of this research were presented at Edupar 2018 [14] and SIGCSE 2019 [15] conferences.

The rest of the paper is organized as follows: Section 2 details our motivation and the related initiatives for including bigdata and cloud computing topics into the curriculum. Section 3 discusses the basic design principles followed while developing the modules and provides an overview of all four modules. Sections 4–7 discuss the four modules and their classroom deployment experiences. Section 8 elaborates on student interests and learning experiences, Section 9 summarizes the results, and section 10 concludes the paper.

2. Motivation and related works

Limited by the fact that ACM or ABET provides no standard guidelines to integrate the bigdata and cloud computing topics into the CS curriculum, the computer science education community has taken multiple approaches to address the need to produce a satisfactory number of well-trained professionals in these fields. Our extensive literature survey identifies three such approaches from Academia. Firstly, several institutions offer non-core specialized courses [17,21,26] to cover various aspects of data science and bigdata analytics, where students are primarily taught data acquisition, cleaning, analytical, and visualization skills. While these courses help students develop skills related to transforming data into knowledge, they do not provide the students with concepts and experiences related to hosting, storing, and deploying applications within the cloud environment and scaling up applications within performance and budgetary constraints. The second category of approaches integrates bigdata and/or cloud computing topics in existing CS/IT courses (mostly non-core) such as Parallel and Distributed Computing, High-Performance Computing, Networking, and Cybersecurity. These interventions are often sporadic, mostly ignited by the instructor's interests and experiences, and therefore cannot garner substantial student interest and knowledge on this collective paradigm. A few research-intensive universities assume a third approach by offering specialized standalone courses [11,12,25,27] such as "Cloud computing," "Bigdata management," etc., where the abovementioned collective paradigm is addressed to a greater extent. However, being an elective (non-core) course offered at a handful of universities, only a handful of students receive the benefit. The abovementioned approaches are limited in intervening a substantial portion of the CS/IT undergraduate student population with the necessary expertise of this collective paradigm as they are mostly undertaken via non-core and special topic courses.

A few years ago, a survey was administered by CDER [7] through SIGCSE listserv to better comprehend instructors' perspectives on the amount and quality of PDC coverage in the undergraduate CS/CE curriculum. Thirty-five educators across the world completed the survey. Several of the survey questions and their results were motivating for the presented study as they reflected the inadequacy of coverages of the applied PDC topics across the CS curriculum and revealed some of the challenges hindering the broader coverage. One such question (Q6) asked the participants about the emerging course topics relevant to PDC that their departments are offering and whether these courses are core (required) or not. Fig. 1 shows the summarized survey results pertaining to that question and suggested that although many institutions are conducting various specialized courses related to bigdata and cloud computing at the junior and the senior level, only a few of them are core courses. When asked about the challenges for broader adoption of PDC topics in the CS curriculum (Q10), the majority of the educators either strongly agreed or agreed that the "Limited room in the existing densely-packed curriculum" is the main challenge, as reflected in Fig. 2. These survey results confirmed our previous observations and findings from the literature that the interventions are mostly happening on the non-core course levels,

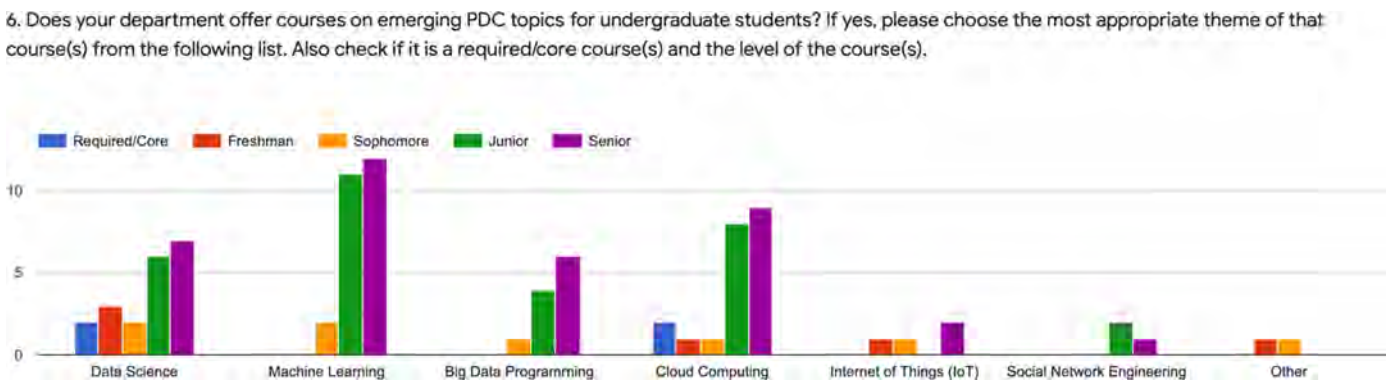

Fig. 1. SIGCSE listserv Survey results showing emerging PDC topics courses. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

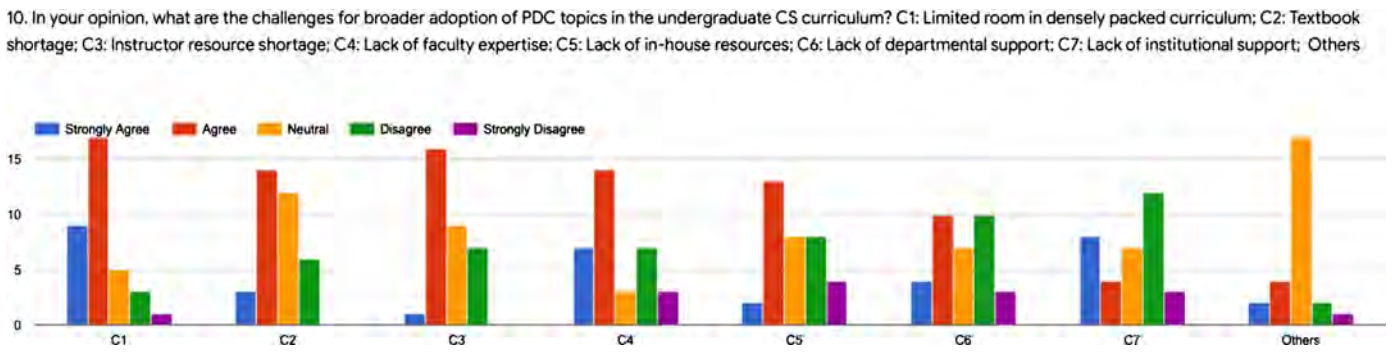

Fig. 2. SIGCSE listserv Survey results showing challenges for broader adoption of PDC topics courses.

and accommodating these current topics in the densely-packed existing curriculum is a significant challenge that hinders massive adoption. There is certainly a big gap between the advances in bigdata and cloud computing and their inclusion in college-level instructions, and this paper aims to address this gap by proposing an alternative module-based approach that covers the related concepts through a set of autonomous modules dispersed over several core courses across the existing curriculum.

3. Modules

3.1. Module design principles

The presented study argues that any systematic approach to module design must be considered within the context of a theoretical framework that underpins effective learning. As indicated in Section 1, this study adopted a competency-based model while designing module outcomes, activities, and assessments. This section details the basic design principles derived from the established learning theories that are utilized in designing the proposed set of short, self-contained modules on bigdata and cloud computing. Table 1 shows a set of envisioned design principles along with their theoretical foundations and the related design activities that are implemented to incorporate these principles into the module design. This study's most notable design choice is the inclusion of the Constructive Alignment model [4,5], as stated below.

> "The fundamental principle of constructive alignment is that a good teaching system aligns teaching method and assessment to the learning activities stated in the objectives so that all aspects of this system are in accord in supporting appropriate student learning."
>
> [Biggs, 1999: 64]

Modules become congruent in an explicit way when there is good alignment and flow between a module's intended learning outcomes, teaching and learning activities, and student learning assessments. This study aims to support students in developing as much meaning and learning as possible from a well-designed, coherent, and aligned module that instigates quality learning and deep engagements.

The study also adopted Project-based learning (PBL) [6,22] by including a substantial hands-on project into each module that spans over extended periods and acts as a vehicle for teaching the important knowledge and skills students need to learn. The goal is to offer competency and deep learning in bigdata and cloud computing, as PBL focuses on real-world problems and challenges and relies on problem-solving, decision-making, and investigative skills. Some other critical strategies of PBL, such as teamwork, producing report/presentation as the results, are also implemented to provide students with core competencies such as collaboration, communication, and reflection. Additionally, temporary support structures such as instructional scaffolding [3] were provided to help students accomplish their project work. The goal is to increase the likelihood for students to meet the learning competencies by supporting their progression through few initial project tasks and providing a welcoming and caring environment as they learn these challenging topics and skills. Lastly, this study adopted Bloom's hierarchy of cognitive processes [2] to specify the degree of skill expected in successful task accomplishment. The goal is to realize and assess the skill dimension of the competency-based philosophy by paring an element of knowledge along with a level of skill with which it is applied.

3.2. Overview of modules

The proposed modules are designed to be short and autonomous for easier adoption, and as a result, each of them spans

Table 1
A set of module design principles and implemented activities.

Design principles	Learning theory basis	Design activities
Modules should be outcome-based. What students should know and demonstrate at the end of the module must be the main guideline for developing the modules.	Constructive Alignment Approach [4,5]	The module outcomes are developed first. Then the assessment criteria are developed. Finally, the module activities are organized that teach the students how to meet the assessment criteria (and hence the module outcomes).
Students must learn by actively engaging in real-world and meaningful projects.	Project-Based Learning [6,22]	Each module is equipped with a hands-on project where students determine how to approach a problem and what activities to pursue. Their learning is connected to a "real", not an academic problem, and involves core competencies such as problem-solving, decision making, collaboration (working in pairs), communication, and reflection (writing analytical reports).
The modular intervention should offer a supportive learning environment.	Instructional Scaffolding [3]	Scaffolding is embedded into the project activities, where initial project activities were first demonstrated (via a tutorial or in-class demonstration) by the instructor while involving students in an interactive way. The demonstrations were followed by a set of challenging project activities while gradually progressing to more extensive and higher-order problems and, at the same time, gradually removing the scaffolding support.
Student learning should be assessed at a variety of cognitive levels.	Bloom's Revised Taxonomy [2]	Module learning outcomes are specified as a set of competencies where each element of knowledge and the requisite level of competency, as a level of Bloom's cognitive process, are paired and specified explicitly.

two to three 75-minute class periods and includes specific competency outcomes, lecture notes, and assessment resources. The assessment resources contain quiz questions, tutorials, sample programs, cloud-based resources, hands-on projects etc. Detailed tutorials are developed for multiple platforms, enabling straightforward adoption at other institutions. The modules are expected to be taught in a lecture-lab setting, where the first class is typically used for discussing the new concepts and techniques, and the second (and third) class is used to introduce students to the hands-on project along with the necessary tools and resources they will require to complete the project. The take-home project contains multiple tasks and usually has two weeks of deadline.

The first of the four modules is designed to expose students to the cloud computing fundamentals and provide them with hands-on experience in using the public cloud environment. The module is adopted to the core "Computer Architecture" (CA) course at our institution. It is important to provide context for the module within the typical course materials where it is deployed so that the students do not perceive the module as an isolated and disruptive topic. This context is established in the CA class by asking students to execute CPU/IO benchmarking applications (typical CA topics) in the cloud setting. The second module is focused on MapReduce programming and popular cloud analytics engines (such as Hadoop and Spark) and is deployed in the "Analysis of Algorithm" (AA) class. The context is established by allowing students to explore the cost vs. performance tradeoff for running analytics applications within the cloud environment. The third module is designed to illustrate the importance of SQL within various big database management systems (BDBMS) and is deployed within a "Database Management" (DB) course. The students could easily see the connection when they utilize Spark-SQL programs to load and query both structured and unstructured data sets. The fourth module is implemented in an "Advanced Operating System" (OS) class to provide a hands-on understanding of important operating systems concepts in a distributed setting. The context is established through project tasks where students investigate common OS issues such as performance, scalability, fault tolerance etc. All four modules are available in their GitHub [35] repository, along with specific deployment instructions for the educators. The following few sections discuss each module along with the student assessment results.

4. "Computer Architecture" (CA) module

This module is integrated into the "Computer Architecture" class and is designed to expose students to virtualization and cloud computing fundamentals and provide students with hands-on experiences in using AWS cloud. This is a required course for both CS and IT majors at our institution, and mostly junior and senior students attend the course. Specific learning competencies are

1. Demonstrate understanding of the key properties, techniques, strengths, and challenges of cloud computing (CA-LO1). (*Skill Level: Understanding*)
2. Develop hands-on experience with Amazon Web Services (AWS) for virtual machine (VM) provisioning and management (CA-LO2). (*Skill Level: Applying*)

4.1. CA: lesson plan

The first 75-minute class is lecture-based and provides an overview of the field, including a discussion on the economic and technological factors that led to the emergence of cloud computing (i.e., advancements in PDC). The lecture further discusses important cloud computing characteristics such as scalability, on-demand access, measured services, and elasticity. 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 distinction between public, private, and community clouds is also clarified. The lecture then explains resource sharing and virtualization while pointing out its important aspects such as migration, timesharing, 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 mainly comprises a tutorial and a lab session that instructs students on the basic provisioning and management of AWS EC2 instances. The tutorial contains platform-

specific instructions and screenshots to help both the Mac and Windows users in provisioning VMs in AWS.

4.2. CA: assessment instruments

Student learning was assessed by utilizing a quiz composed of factual questions and a take-home project involving AWS EC2 services. The quiz includes three true/false, six multiple choices, and four analytical questions and focuses on gauging the progress students have made comprehending and retaining concepts related to virtualization and cloud computing fundamentals. As part of the take-home project, students are paired to form a group of two persons and asked to perform the following tasks

Task 1: Create two EC2 instances with different hardware and software configurations.
Task 2: Analyze instance performances by benchmarking them with CPU- and IO-bound applications and determine how performance scales with the different VM types.
Task 3: Represent the results of the analysis graphically and report on them critically.

Students are asked to utilize Systester [28] as a CPU-bound benchmark, which calculates the nth digit of π using the Gauss-Legendre algorithm while experimenting with various values of n such as 128 K, 256 K, 512 K, 1 M, 2 M, 8 M, 16 M. Additionally, the project requires the students to use the iozone benchmarking tool from the Phoronix [23] benchmarking suite to assess IO performances. Students set up experiments to benchmark both read and write performances by running iozone for different record sizes (4 KB and 64 KB) and file sizes (512 MB, 2 GB, and 8 GB). The students then repeated the experiments for both VM instances with different configurations that they set up in Task 1, expressed their benchmarking performance results graphically, and then evaluated their performance analysis results critically. The goal for the take-home project is to provide the students with the necessary skillset in acquiring and managing virtual resources in AWS and in utilizing them to execute CPU/IO benchmarking applications with very large problem sizes with the realization that it would not be possible to accomplish this using their local environments.

4.3. CA: assessment results

The CA module was deployed in the Spring 2018 offering of the CA class with 30 students in it. Overall, students performed very well (mean: 87%, median: 89%) in answering the quiz questions. The detailed performance results are expressed in Fig. 3 and Fig. 4. Fig. 3 shows the quartile performance for the quiz (red quartile) and reflects that the majority of the students scored around 83% to 93% in the CA-Quiz, with very few of them residing in the outlier group. Fig. 4 shows the overall quiz grade distributions. These statistics indicated students' knowledge acquisition and comprehension on the concerned topics and therefore showed the module's effectiveness in attaining the outcome CA-LO1 with the associated skill level of "Understanding".

Students were more challenged with the take-home project, and 9 out of 30 students did not submit the project. The remaining 21 students who spent the time to accomplish the project did reasonably well (Mean: 74%, Median: 80%), while most of them scored in the 60%–100% range as reflected in Fig. 3 (blue quartile), certainly exhibiting a larger variability than the quiz grades. Fig. 4 shows students' project grade distributions and reveals that although 38% of the students did very well in the project, about 29% received D/F grades and struggled to accomplish all three tasks. A closer analysis of their project grades reveals that although all students were comfortable with the AWS web interface

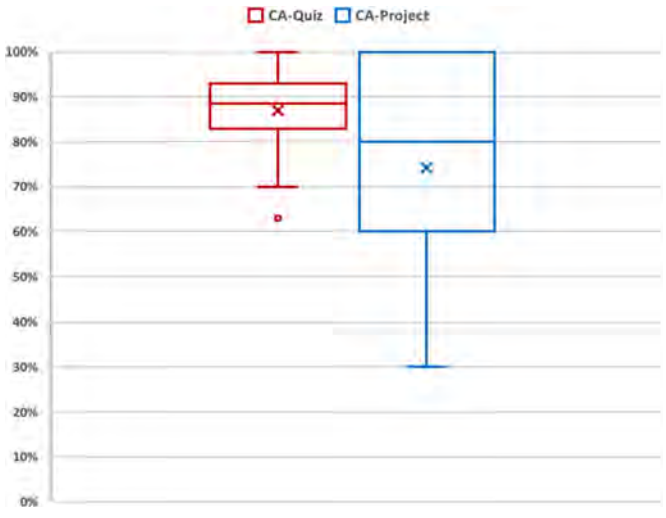

Fig. 3. Student performances in CA module.

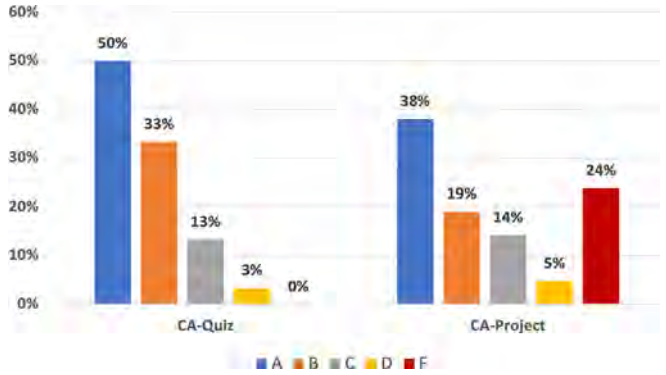

Fig. 4. Student grade distributions in CA module.

for VM management (especially after the detailed tutorial session) and accomplished Task 1, some had trouble correctly executing the benchmarking applications on the VM instances (Task 2), and a group of students faced challenges while comparing and presenting their results and reporting on them critically (Task 3). These results suggest that majority of the students (71% attained passing grades) were able to apply their skills in provisioning and managing VMs in the AWS cloud, indicating that the outcome CA-LO2 was met marginally. It is clear that the student population being part of two different majors and backgrounds (CS and IT) was an important factor that contributed to the variations in assessment results and that the module would benefit more if it were offered to a particular major and tailored accordingly. However, we did not have the opportunity at our institution to accommodate that.

5. "Analysis of Algorithm" (AA) module

This module is designed for the junior-level Algorithm course students to introduce the MapReduce programming framework and the popular cloud analytics engines such as Hadoop and Spark. This module is targeted only for the CS majors, who have completed CS1, CS2, and a data structure course and are comfortable with and reasonably proficient using programming languages such as Java or Python. The learning outcomes of the module are

1. Recognize the key properties, techniques, strengths, and challenges of MapReduce and Spark Framework (AA-LO1). (*Skill Level: Understanding*)
2. Build scalable applications based on MapReduce programming model using Hadoop and HDFS (AA-LO2). (*Skill Level: Applying*)

3. Analyze performance and cost constraints using cloud platforms (AA-LO3). (Skill Level: Analyzing)

5.1. AA: lesson plan

This module spans for three 75 minutes classes. The first 75-minute class is lecture-based and 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 key-value pair concept 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 maps and reduce tasks is the essence of designing MapReduce programs, at this stage, the lecture spends a fair 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. As an implementation of the MapReduce model, Apache Hadoop is then introduced along with its distributed file system HDFS. The concept of a Hadoop cluster, along with the master-worker framework, is briefly explored. Lastly, the classic MapReduce programming problems are discussed briefly, and the need for an in-memory analytic engine such as Apache Spark is emphasized with a brief description of Spark's runtime distributed architecture, including driver, executors, and directed acyclic graph (DAG).

On the other hand, the second class is a lab-based class where the instructor provides some demonstrations and a tutorial for the students to follow. One of this module's goals is to provide students with some skills in MapReduce implementations and for that they need continuous access to a development environment. We recommend using Cloudera's VM [10] 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. The tutorial, therefore, provides instructions on running Hadoop through Cloudera's VM on the local machine. It also provides students with step-by-step instructions to set up HDFS, compile and execute a MapReduce application, and retrieve output from HDFS. All the Java files (driver, mapper, and reducer) and input datasets were provided, and the students were able to understand the various stages of a MapReduce job and its execution through executing an application that finds the year-wise maximum temperature (MaxTemperature.java). Students then applied their recently learned skills while executing a second WordCount application (WordCount.java) that works with a larger dataset.

The third class is also a lab-based demonstration class that introduces students to Chameleon cloud [8] and shows them how to ssh to a particular Chameleon instance. All students were 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 application that the students use for experimentation and the datasets are preloaded to that instance. The concerned Spark application finds the trending topics given the Wikipedia page views information [34]. Students were taught how to execute a Spark application in Yarn cluster during the lab session 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 were also taught to verify the current resource allocation and check the execution time and other performance metrics of a spark application using Spark's web user interface.

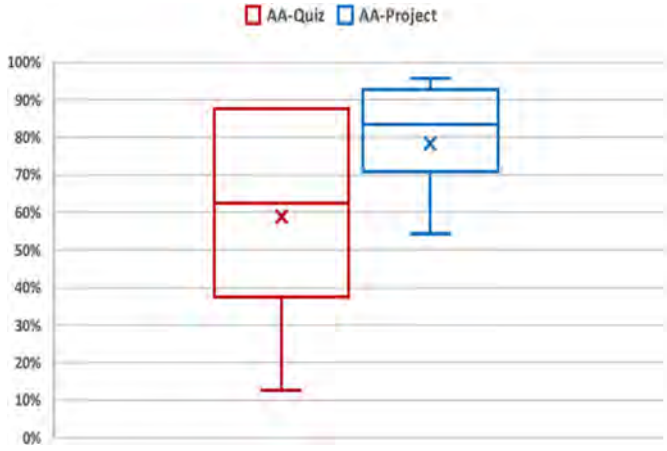

Fig. 5. Student performances in AA module.

5.2. AA: assessment instruments

This module is assessed by utilizing both hands-on project and quiz questions. As part of the quiz, there are four multiple-choice questions and two design questions developed to assess students' comprehension of the 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 two lab sessions and includes the following tasks:

Task 1: Modify the MaxTemperature.java (Section 5.1), so that it produces the Average Temperature of each year instead of the Maximum Temperature.

Task 2: Modify the WordCount.java (Section 5.1), 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 particular step of the project requires students to explore Spark application performance in the cloud environment by running them with various runtime configuration settings and gaining insight into the resource provisioning and the performance vs. cost tradeoff. Students are presented with two Wikipedia data sets (100 GB and 200 GB) 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. All code and data files were pre-loaded in the Spark clusters. Although Chameleon's use 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 performances while executing the applications in different cluster configurations and to 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.

5.3. AA: assessment results

The module was deployed in the Spring 2017 offering of the AA course (N:14). Fig. 5 shows the quiz (red) performances and

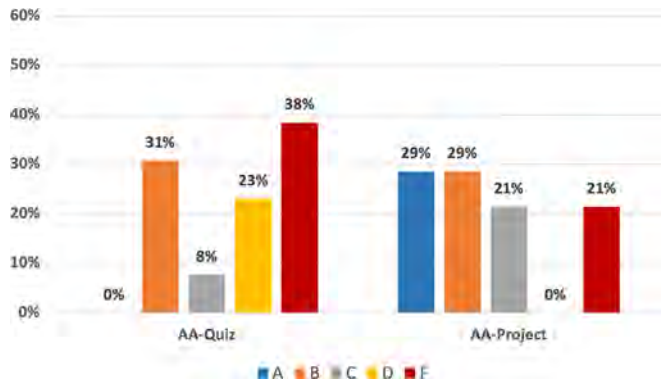

Fig. 6. Student grade distributions in AA module.

reveals that students performed poorly in this assessment with a mean score being 59%, the median being 63%, and the quartiles being 38% and 88%. The quiz questions were part of an exam that occurred two weeks after the module intervention, and students had difficulties retaining difficult concepts associated with MapReduce design. 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 an average score of 38%) involves writing a series of *map-reduce* tasks, and while many students provided a partially correct answer, only a few explored it completely. Fig. 6 shows the quiz grade distributions with 0% of students achieving A grade, only 39% of the students achieving passing grades (B/C), and about 60% of the students failing (D/F) in the quiz. These results indicate that the students were able to comprehend the concepts taught as part of the module to a certain extent, and therefore the outcome AA-LO1 was met only partially.

After the rigorous lab session and many other individual troubleshooting sessions supported by the instructor and the TAs, all students were able to perform Task 1 and Task 2 of the project, Task 3 was successfully completed by about 80% of the students, and Task 4 was completed successfully by only 30% of the students. The quartile statistics (Fig. 5) of the students' project grades (mean: 78%, median:83%), and the project grade distributions, as reflected in Fig. 6, with 29% receiving A grades and about 80% receiving passing grades (A/B/C) show student competency in developing simple MapReduce applications using Hadoop (AA-LO2). The project's final task exposed students to the cloud-enabled Spark environment and allowed them to understand and analyze the performance-cost tradeoffs (AA-LO3) of executing Spark application in Chameleon cloud environment; however, only a few (30%) students were able to complete this task successfully.

6. "Database Management" (DB) module

Currently, numerous application scenarios require processing very large datasets in a highly scalable and distributed fashion. Various types of bigdata systems have been designed to address this challenge, and many of them have recognized the strengths of SQL as a query language. The proposed module for the introductory Database Management course is designed to expose students to the various types of bigdata systems and to integrate the study of SQL within these systems. This course is required for both CS and IT majors at our institution and is typically attended by sophomore and junior students. The learning outcomes are

1. Summarize the key properties, strengths, and limitations of important big database management system (BDBMS) such as MapReduce, No-SQL, and New-SQL (DB-LO1) (*Skill Level: Understanding*)
2. Develop hands-on experience in using SQL within Spark framework to load and query big datasets (DB-LO2). (*Skill Level: Applying*)

6.1. DB: lesson plan

The first class (lecture) discusses the limitations of relational database systems and key properties, strengths, and limitations of various bigdata management systems such as MapReduce, No-SQL, and New-SQL. The second class introduces the Spark distributed data processing framework, with Spark-SQL being a component within it for handling structured data processing. The lecture then introduces the students to the concept of DataFrames, emphasizing it can be created from a wide array of sources. The instructor then provides students with a tutorial and source code that instructs students on utilizing Jupyter notebook for developing and executing Spark-SQL applications. The tutorial provides instructions on creating DataFrames from JSON and CSV files, manipulating them, and running SQL queries programmatically within them.

6.2. DB: assessment instruments

A hands-on project and quiz questions were utilized as assessment instruments. The quiz contains four multiple-choice questions that assess students' understanding of No-SQL and New-SQL systems and a detailed analytical question that assesses their comprehension of Spark-SQL programming. The hands-on project is a progression of the in-class tutorial and involves developing a Spark application that loads historical Facebook stock prices [19] and uses Spark SQL to query the data. The dataset is downloaded as a CSV file, where each record contains the stock values of a single date and includes the following attributes: date, open price, high price, low price, close price, volume, and adjClose (close price adjusted for dividends and splits). Students are asked to develop a Spark-SQL application that loads the file, creates a DataFrame based on the content, registers the DataFrame as a SQL temporary view, and then includes queries that show the answers to the following questions

1. Show all records that have gained value during daily transaction in the year 2018 (close >= open).
2. Which day did Facebook stock gain maximum value?
3. Show the first 10 highest stock values.
4. Show the average sale volume for last 5 years.
5. Add two more questions and write queries to answer them.

Students were taught to infer schema based on JSON and CSV input during the tutorials, and the project requires them to perform similar inference while querying the Facebook data set. However, the project also contains a challenging part (extra credit) where the students were asked to load and query a text file where schema should be enforced programmatically. Students were not explicitly taught the schema enforcement but are provided with some resources that they might find useful in implementing this challenging part.

6.3. DB: assessment results

Overall, students were able to recognize the key properties, strengths, and limitations of an important big database management system, which is reflected by their quiz performances. Fig. 7 shows the quiz quartile performances (red), with the mean and median score being 78%. Fig. 8 shows the grade distributions with 31% of the students receiving a grade of A and 81% of the students receiving passing grades. Similarly, students also performed

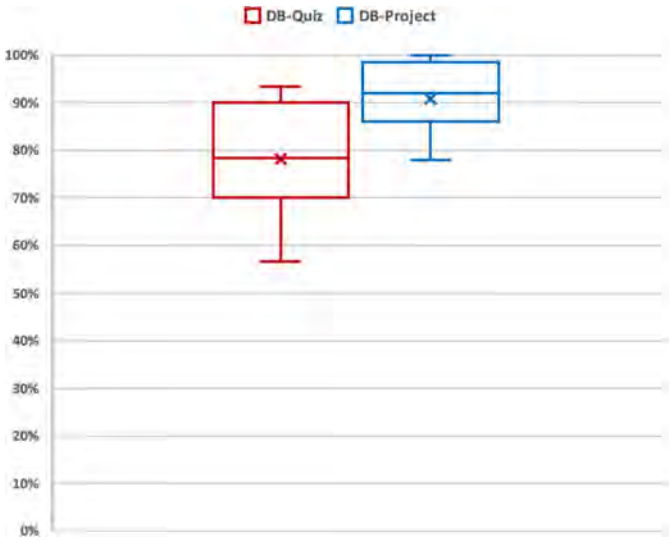

Fig. 7. Student performances in DB module.

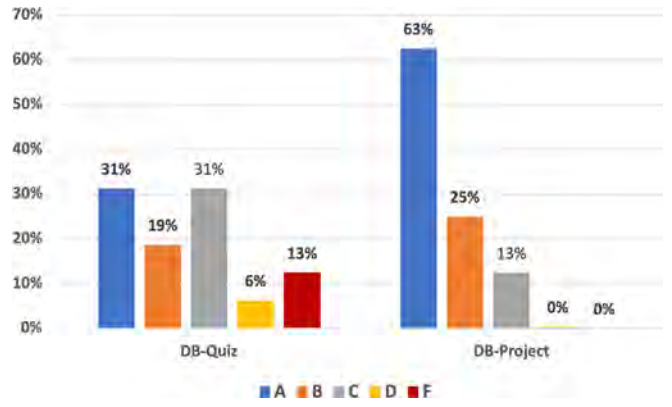

Fig. 8. Student grade distributions in DB module.

well in the essential part of the project (infer schema automatically) with the mean score being 91% and median being 92% with relatively lower variability (blue quartile in Fig. 7). Fig. 8 reveals that 63% of the students attained A grade, while all of them received passing grades in the project. However, the extra credit part (define schema programmatically) was successfully completed by only 12% of the students. Overall the students' quiz performance validates their thorough understanding of the key properties of the various BDBMS systems (DB-LO1), and their project grades reflect their competency in developing a Spark-SQL application in analyzing and querying big datasets (DB-LO2).

7. "Operating Systems" (OS) module

This module is designed to be deployed in an operating system class to provide a hands-on understanding of a parallel distributed processing framework such as Apache Spark and distributed storage engines such as HDFS through their deployments in Google Cloud Dataproc [18]. Dataproc supports automatic cluster management in the Google Cloud Platform (GCP), and as a result, it becomes easy to build fully managed Spark or Hadoop clusters in GCP. At our institution, we utilized this module in an Advanced OS class targeted for graduate students to provide experiences in data parallelism and fault tolerance. However, by utilizing the additional supports provided by the video lecture and detailed steps, one could use the module (at least partially) in an undergraduate OS class. The learning outcomes are

1. Deploy and configure Apache Spark and HDFS in GCP Dataproc (OS-LO1). (*Skill level: Applying*)
2. Execute Spark application in the cluster and analyze its performance given different scenarios related to scalability, replication, and fault tolerance (OS-LO2). (*Skill level: Analyzing*)

7.1. OS: lesson plan

Students had assigned reading on Apache Spark [36] and was supplied with Google Cloud Platform (GCP) tutorials and an instructor-made video lecture explaining necessary steps such as redeeming coupons in GCP, creating Dataproc cluster in GCP and configuring them with various resources, creating buckets to store the application and the data, submitting a Spark job in the cluster, and monitoring and assessing its execution. The only 75 minutes lab-based class offered as part of this module was devoted to answering students' questions and troubleshooting their problems while running experiments to answer questions that are part of their take-home project.

7.2. OS: assessment instruments

A take-home project was utilized as an assessment instrument for this OS module. Students were supplied with a PySpark implementation of a WordCount application and a substantially large data file. Students were then asked to set up experiments to answer the below questions and submit a detailed report analyzing their results and supporting them with screenshots, tables, graph etc.

Question 1. What is the default block size on HDFS? What is the default replication factor of HDFS in Dataproc?

Question 2. Using pg100.txt as input, run the word-count.py program on a Single Node cluster using 4 cores. What is the completion time of the task? Take a snapshot of your VM instances monitoring page while running.

Question 3. Using pg100.txt as input, run the word-count.py program under HDFS inside a 2 node cluster (1 master, 1 worker nodes). Is the performance getting better or worse in terms of completion time? Explain.

Question 4. Using pg100.txt as input, run the word-count.py program under HDFS inside a 3 node cluster (1 master, 2 worker nodes). Is the performance getting better or worse in terms of completion time? Explain.

Question 5. For this question, change the default block size in HDFS to be 64 MB and repeat Question 4. Record run time, is the performance getting better or worse in terms of completion time? Briefly explain.

Extra Credit:

Question 6. Run the settings in Question 4, kill one of the worker nodes immediately. You could kill one of the worker nodes by going to the VM Instances tab on the Cluster details page and click on one of the workers' name. Then click on the STOP button. Record the completion time. Does the job still finish? Do you observe any difference in the completion time? Briefly explain your observations.

7.3. OS: assessment results

The module was deployed in the Fall 2020 offering of an advanced operating system class targeted for the graduate CS students (N:5). Students' project grades reflect comprehension and competency, with the mean score being 93% and the median being 90%. Students grade distribution shows that 100% of the students were able to complete Questions 1–4 along with their performance analysis, 60% of the students were able to complete all questions except the extra credit question regarding fault tolerance, and only 20% of the students were able to critically report on the extra credit task after successfully executing that. All students critically reported on their experiments and augmented their arguments with screenshots, graphs, and tables as necessary. All students were able to recognize the issues related to scalability. For example, they correctly recognized an improvement in application completion time when switching from a single node cluster to a 2-node cluster. They further recognized a slight increase in completion time when experimenting with one master and two worker nodes as the overhead associated with distribution and communication outperformed the benefits associated with parallel execution of the not so large load contained in the input file supplied with this project. Additionally, 60% of the students were able to analyze the impact of choosing a smaller block size by noticing a decrease in performances due to having more swaps between smaller blocks. Only one student was able to correctly execute the extra credit question and observed and analyzed that even after shutting down a worker in the middle of the execution, the job still was able to complete successfully due to the replication and fault tolerance support provided by Spark. However interrupting the data stream and Java connection refusal takes a little longer to adapt to the loss of the worker and therefore additional time was needed to complete the job. Overall, all students were able to deploy and configure Apache Spark and HDFS in the Google Cloud Dataproc cluster, and therefore outcome OS-LO1 was met. All of the students were able to analyze the performance issues related to parallel execution, while some analyzed the impact of replication and fault tolerance (OS-LO2).

8. Student interest and learning experience

An IRB-approved three-question student survey (Appendix A) was designed to assess students' perceptions of their learning experiences and their level of confidence and interests and is administered after each course intervention. The students provided their opinions about the three statements using a Likert scale of five values such as Strongly Agree, Agree, Neutral, Disagree, and Strongly Disagree. The questions are as follows, where X refers to the module specific topics.

- Q1 – I found the topics X interesting.
- Q2 – If a friend asks me what X are, I will be able to explain for 2–3 minutes.
- Q3 – I would like to learn more about X and would like to explore more in my future courses.

Fig. 9 shows the Q1 survey results for all four courses. The results clearly show that most of the students for all four courses found the conveyed topics interesting. More specifically, 79% of the CA course students either strongly agreed or agreed that they enjoyed the topics, whereas the rates of agreements for AA, DB, and OS courses are 100%, 74%, and 100%, respectively. It is important to note that although students did not perform well in AA module compared to the other course modules (Section 9), all of them unequivocally agreed that they found the topics interesting and enjoyed the extensive programming and hands-on cloud experiences.

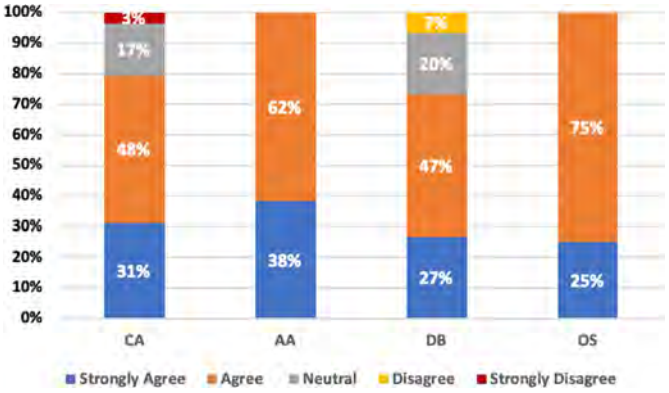

Fig. 9. Survey Results for Q1: I found the topics X interesting.

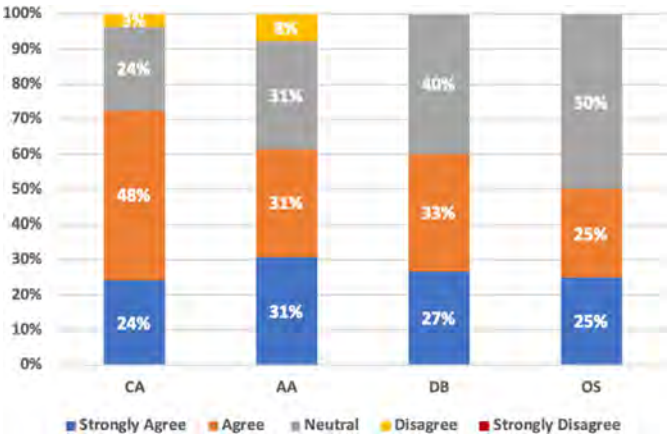

Fig. 10. Survey Results for Q2: If a friend asks me what X are, I will be able to explain for 2–3 minutes.

Perhaps, spending three 75 minutes classes, including two lab sessions, made the students to feel more involved and enthusiastic about the challenging works that they performed as part of this module.

Fig. 10 shows the survey results reflecting students' self-reported level of confidence in the learned topics. It is not surprising to notice a slightly lower number of agreements (strongly agree or agree) across all four courses compared to Fig. 9 and many neutral responses. The amount of time spent on each module is not enough to provide the students with the necessary self-reliance. However, 72% of the CA course students either strongly agreed or agreed that they are familiar with the topics. Similarly, 62%, 60%, and 50% of the AA, DB, and OS students reported their familiarity with the covered topics. A larger group of students was unsure about their fluency and therefore remain neutral, however, there were very few disagreements noticed in Fig. 10.

Fig. 11 shows the survey results reflecting students' self-reported desire to learn the topics more after the intervention. Overall, students eagerly expressed their desire to learn more about the topics in the future. More specifically, 86%, 76%, 87%, and 75% of the CA, AA, DB, and OS students enthusiastically revealed their desire to learn the topics more.

9. Discussions

The presented modular intervention addressed 65 CS/IT majors across four core CS courses at our institutions from 2017 to 2020. Table 2 shows the mapping of module outcomes to corresponding assessment instruments and the percentage of students achieving passing grades in those instruments. It is evident from Table 2 that except for the Algorithm module, other interventions were quite

Table 2
Mapping of LOs to assessments along with student performance results.

Learning Outcome (LO)	Assessment instruments	% of students received A/B/C grades
CA-LO1: Demonstrate understanding of the key properties, techniques, strengths and challenges of cloud computing. (*Skill Level: Understanding*)	CA-Quiz	96%
CA-LO2: Develop hands-on experience with Amazon Web Services (AWS) for virtual machine (VM) provisioning and management. (*Skill Level: Applying*)	CA-Project	71%
AA-LO1: Recognize the key properties, techniques, strengths and challenges of MapReduce and Spark Framework. (*Skill Level: Understanding*)	AA-Quiz	39%
AA-LO2: Build scalable applications based on MapReduce programming model using Hadoop and HDFS. (*Skill Level: Applying*)	AA-Project (Task 1–3)	79%
AA-LO3: Analyze performance and cost constraints using cloud platforms. (Skill Level: Analyzing)	AA-Project (Task 4)	30%
DB-LO1: Summarize the key properties, strengths, and limitations of important big database management system (BDBMS) such as MapReduce, No-SQL, and New-SQL. (*Skill Level: Understanding*)	DB-Quiz	81%
DB-LO2: Develop hands-on experience in using SQL within Spark framework to load and query big datasets. (*Skill Level: Applying*)	DB-Project	100%
OS-LO1: Deploy and configure Apache Spark and HDFS in GCP Dataproc. (*Skill level: Applying*)	OS-Project (Q1–Q4)	100%
OS-LO2: Execute Spark application in the cluster and analyze its performance given different scenarios related to scalability, replication, and fault tolerance. (*Skill level: Analyzing*)	OS-Project (Q1–Q6)	60%

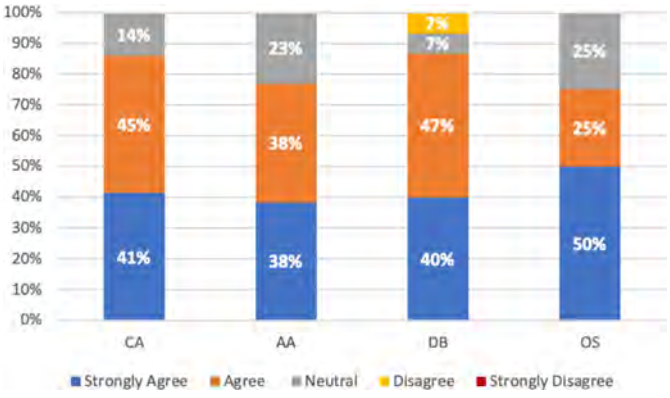

Fig. 11. Survey Results for Q3: I would like to learn more about X in future.

successful in attaining competency outcomes as specified as part of the modules. The Algorithm instructor spotted a few reasons for the poor student performance such as 1) delayed (2 weeks after the intervention) offering of the quiz 2) introduction of two different frameworks such as Hadoop and Spark in such a short period that incurs additional load and confusion for some students, and 3) use of Chameleon cloud for analyzing performance-cost trade-offs instead of using a well-interfaced popular cloud platform such as GCP or AWS. It is anticipated that the students would perform better with the module focusing on Spark framework only and utilizing easy-to-use popular cloud platforms. However, during the intervention (2017), there were several technical and organizational limitations that governed our decisions at that time. Hopefully, our future interventions and other interested instructors would be more careful about their bigdata processing frameworks and cloud platform choices.

It is also worth mentioning that there were concessions and adjustments made to the final grading due to the modules' experimental nature and after evaluating their impacts on students' overall grades. For example, the AA-Quiz grades were scaled, and task # 4 of AA-Project was declared as "extra credit" to avoid penalizing the students. At the same time, stronger students were provided with ample challenging opportunities such as extra credit parts of DB and OS modules. We hope that the future adopters will learn from our experiences and the decisions we made to mitigate potential poor impacts and to keep the diverse group of students engaged.

The relatively higher number of agreements with the self-reflection survey questions demonstrate that the students valued the experience, felt comfortable with the module content, and grew further interest even though the intervention duration was very short. Several students volunteered to add anonymous comments in the post-survey that showed the module's usefulness and their eagerness to spend more time on the topics. Three 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."
>
> "It would be cool to see this expanded on. For instance, a project based around it where a student could start small and build on it throughout the semester to get even more familiar with the cloud platform."

The abovementioned comments also articulated some important suggestions such as "the need for assigning more time to complete the project", "the need for the students to have prior Linux skill", and "the need for a semester long comprehensive course". While most future adopters could easily address the first two suggestions by assigning more time for the project and by having the students covering some content and practice on Linux outside of the classroom, the third suggestion could be challenging to adopt for many institutions without proper guidelines enforced by CS Curriculum standards as outlined in Section 1.

The development of the constructively aligned modules required a significant amount of time and thoughts on the instructor's part. Considering the short period available for class intervention, the deployment also required large time investment and substantial class preparation. Our class size was relatively small, and the instructor and her teaching assistants 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 module's duration. Both modules were deployed at WSSU, an HBCU that serves a unique group of students as 71% of its student population is female, and 72% are African American. Therefore, the modules are carefully designed to incorporate pedagogies such as project-based learning, instructional scaffolding, etc., which are recognized by many research studies in addressing some of the challenges that underrepresented minority students typically face during their college years.

One limitation of the study is that the number of students (N:65) impacted by this study was not that significant, and further repetitions of the interventions are necessary to make more robust conclusions about their effectiveness. Additionally, as bigdata and cloud computing are rapidly changing topics, an instructor's goal should be to offer students the most up-to-date experiences. Therefore, the modules need to be evaluated regularly from a technological perspective and perhaps need regular updating to be aligned with current technologies and frameworks. It is also recognized that the presented competency model is not complete as it is missing the specification of the "Disposition" dimension [9] of such model. Disposition in a competency model includes socio-emotional skills, behaviors, and attitudes that control whether and how an individual is motivated to use her skills. While working on creating the proposed competency model with "Knowledge" and "Skill" dimensions, it was recognized that more research is needed for the authors to understand the "Disposition" dimension effectively in order to include this in the proposed learning outcomes. In the future, we will focus on completing the Disposition attributes for each listed competency.

10. Conclusions

This study aims to explore the integration of bigdata and cloud computing modules into core undergraduate CS/IT courses and evaluate its effectiveness. A substantial advantage of the modular approach is that many CS/IT majors can be exposed to these contemporary topics and technologies via systematic and increasing integration throughout the computing curricula without developing an additional core or elective course. This paper presents four such modules and our classroom experiences while deploying them. The specific contributions are as follows

1. Literature review and the results of an instructor survey suggested that the emerging and important topics such as bigdata and cloud computing have not yet transitioned into the densely-packed undergraduate CS curricula at many institutions.
2. For broader and systematic adoption of such topics, a framework is proposed in this study where a series of short, self-contained learning modules with specific learning goals, lessons plans, and assessment instruments are developed and dispersed over several core courses across the existing CS/IT curriculum without performing a major overhaul of the curriculum or creating additional courses.
3. The modules are designed following established learning theories and pedagogies that adequately characterized the embraced competency-based model where learning must be demonstrated by both knowledge and skills.
4. Each of the developed modules encompasses a hands-on project on using cloud analytics engines such as Hadoop and Spark on popular cloud platforms such as Amazon web services (AWS), Google Cloud Platform (GCP), and Chameleon.
5. Student performance and survey results (N:65) demonstrate reasonable success in attaining student learning outcomes, enhanced engagement, and interests.
6. The modules are designed to be reusable and adoptable and are available for downloading at our GitHub repository.

The student-generated evidence based on student performance and survey data supports our pedagogy, inspires us to assess and update our interventions continuously, and allows us to extend our interventions across multiple courses and semesters. The assessment results clearly show that the students could relate to the topics very well, found them to be interesting enough to explore and retain, and developed significant interest and confidence after the interventions.

As part of our future works, we would like to focus on converting the presented modules into flipped classroom modules to be able to 1) deploy them in hybrid/virtual platforms which is increasingly gaining importance during and after the COVID-19 era, and 2) support the adoption in challenging situations where an instructor cannot afford to use a week of her classtime exploring these modules but would like her students to experience them. Flipped modules will be supported by pre-recorded lecture/demo, so that the instructor could only use one of her class answering students' concerns and troubleshooting their problems instead of using two or three classes.

In the future, we would also like to perform research on more gradual and systematic integration of the developed modules across the curriculum and research on assessing their collective effectiveness rather than measuring the efficacy of a single module. We are also currently developing a follow-on capstone course that includes all of the presented topics and assessments in more detail to reinforce the concepts and provide the students with a comprehensive set of skills in applied parallel and distributed computing. It would be interesting to analyze the student performance and perception data for that course in the future in a pre- post- way to understand to what extent students retain these modules and whether there are any enhancements after covering all topics comprehensively within a single course.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Sushil Prasad, University of Texas, San Antonio (co-author); Erik Saule, University of North Carolina, Charlotte (co-chair).

Acknowledgments

This research is supported by UNC Research Opportunities Initiative (ROI) award "Winston-Salem State University Center for Applied Data Science", FY2021–2023 and NSF Award #1600864.

Appendix A

CSC 3322: Computer Architecture
Cloud Computing: Survey

1. I found the topic Cloud Computing Interesting
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree
2. If a friend asks me what "Cloud Computing" is, I will be able to explain for 2–3 minutes
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree
3. I would like to learn more about Cloud Computing and AWS framework and would like to explore more in my future courses
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree

CSC 3331: Analysis of Algorithms

Parallel Programming and MapReduce/Spark Framework: Survey

1. I found the topic Parallel Programming with MapReduce/Spark Framework Interesting
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree
2. If a friend asks me what "MapReduce/Spark programming Framework" is, I will be able to explain for 2–3 minutes
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree
3. I would like to learn more about Parallel Computing and MapReduce/Spark programming and would like to explore more in my future courses
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree

CSC 3355: Database Management Systems

Big data Management Systems: Survey

1. I find the topic big database management systems (BDBMS) and Spark SQL framework Interesting
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree
2. If a friend asks me what big database management systems or Spark SQL are, I will be able to explain for 2–3 minutes
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree
3. I would like to learn more about big database management systems and Spark SQL programming and would like to explore more in my future courses.
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree

CST 5321: Advanced Operating System

Spark Framework: Survey

1. I found executing Spark applications in the Google Cloud Framework (GCP) Interesting
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree
2. If a friend asks me what Spark and GCP are and how these tools and frameworks could be utilized to execute big data applications in the cloud platform, I will be able to explain for 2–3 minutes
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree
3. I would like to learn more about Spark and GCP and would like to explore them more in my future courses.
   a. Strongly Agree
   b. Agree
   c. Neutral
   d. Disagree
   e. Strongly Disagree

References

[1] ABET, Accreditation Criteria and the Accreditation Policy and Procedure Manual, https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2021-2022/, 2021. (Accessed 8 February 2021).

[2] L. Anderson, D. Krathwohl, P. Airasian, K. Cruikshank, R. Mayer, P. Pintrich, J. Raths, M. Wittrock, Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, 1st edition, Pearson, 2001.

[3] B.R. Belland, Instructional Scaffolding in STEM Education: Strategies and Efficacy Evidence, 2016.

[4] J. Biggs, What the student does: teaching for enhanced learning, High. Educ. Res. Develop. 18 (1) (1999) 57–75.

[5] J. Biggs, Aligning teaching for constructing learning (online), High. Educ. Acad. (2013).

[6] P.C. Blumenfeld, E. Soloway, R.W. Marx, J.S. Krajcik, M. Guzdial, A. Palincsar, Motivating project-based learning: sustaining the doing, supporting the learning, Educ. Psychol. 26 (3–4) (1991) 369–398, https://doi.org/10.1080/00461520.1991.9653139.

[7] Center for Parallel and Distributed Computing Curriculum Development and Education resources (CDER), (n.d.), https://grid.cs.gsu.edu/~tcpp/curriculum/. (Accessed 10 February 2021).

[8] Chameleon Cloud, (n.d.), https://www.chameleoncloud.org. (Accessed 10 February 2021).

[9] A. Clear, A.S. Parrish, J. Impagliazzo, M. Zhang, Computing curricula 2020: introduction and community engagement, in: Proc. 50th ACM Tech. Symp. Comput. Sci. Educ., Association for Computing Machinery, New York, NY, USA, 2019, pp. 653–654.

[10] Cloudera QuickStart VM, (n.d.), https://www.cloudera.com/downloads/cdp-private-cloud-trial.html. (Accessed 10 February 2021).

[11] CS349D, Cloud Computing Technology, (n.d.), http://web.stanford.edu/class/cs349d/. (Accessed 10 February 2021).

[12] CS5412, Topics in Cloud Computing, (n.d.), http://www.cs.cornell.edu/courses/cs5412/2018sp/. (Accessed 10 February 2021).

[13] A. Danyluk, L. Cassel, P. Leidig, C. Servin, ACM task force on data science education draft report and opportunity for feedback, in: SIGCSE 2019 – Proc. 50th ACM Tech. Symp. Comput. Sci. Educ., 2019.

[14] D. Deb, S. Cousins, M. Fuad, Teaching big data and cloud computing: a modular approach, in: Proc. – 2018 IEEE 32nd Int. Parallel Distrib. Process. Symp. Work. IPDPSW 2018, 2018.

[15] D. Deb, M. Fuad, K. Irwin, A module-based approach to teaching big data and cloud computing topics at CS undergraduate level, in: SIGCSE 2019 – Proc. 50th ACM Tech. Symp. Comput. Sci. Educ., 2019.

[16] S. Draft, Computer Science Curricula 2013, ACM IEEE Comput. Soc. Inc., New York, NY, USA, 2013.

[17] J. Eckroth, Teaching future big data analysts: curriculum and experience report, in: Proc. – 2017 IEEE 31st Int. Parallel Distrib. Process. Symp. Work. IPDPSW 2017, 2017.

[18] Google Cloud Dataproc, (n.d.), https://cloud.google.com/dataproc. (Accessed 10 February 2021).

[19] Historical Facebook Stock Prices, (n.d.), https://finance.yahoo.com/quote/FB/history/?guccounter=1. (Accessed 10 February 2021).

[20] Integrating Big Data & Cloud Computing into the Computer Science Curricula and Research, (n.d.), http://ibigcloud.altl.org/resources.html. (Accessed 10 February 2021).

[21] S.J. Matthews, Using Phoenix++ MapReduce to introduce undergraduate students to parallel computing, J. Comput. Sci. Coll. 32 (2017).

[22] H. Mills, D. Treagust, Engineering education. Is problem-based or project-based learning the answer?, Aust. J. Eng. Educ. 3 (2003).

[23] Phoronix Test Suite, (n.d.), https://www.phoronix-test-suite.com/. (Accessed 10 February 2021).

[24] S.K. Prasad, C.C. Weems, J.P. Dougherty, D. Deb, NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing: status report, in: Proc. 49th ACM Tech. Symp. Comput. Sci. Educ., Association for Computing Machinery, New York, NY, USA, 2018, pp. 134–135.
[25] A.S. Rabkin, C. Reiss, R. Katz, D. Patterson, Experiences teaching MapReduce in the cloud, in: SIGCSE'12 – Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ., 2012.
[26] B. Ramamurthy, A practical and sustainable model for learning and teaching data science, in: SIGCSE 2016 – Proc. 47th ACM Tech. Symp. Comput. Sci. Educ., 2016.
[27] M.S. Rehman, J. Boles, M. Hammoud, M.F. Sakr, A cloud computing course: from systems to services, in: SIGCSE 2015 – Proc. 46th ACM Tech. Symp. Comput. Sci. Educ., 2015.
[28] System Stability Tester, (n.d.), http://systester.sourceforge.net/. (Accessed 10 February 2021).
[29] T.G. on Information Technology Curricula, Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology, Association for Computing Machinery, New York, NY, USA, 2017.
[30] H. Topi, H. Karsten, S.A. Brown, J.A. Carvalho, B. Donnellan, J. Shen, B.C.Y. Tan, M.F. Thouin, MSIS 2016 global competency model for graduate degree programs in information systems, Commun. Assoc. Inf. Syst. 40 (2017), https://doi.org/10.17705/1cais.04018.
[31] U.D. of Labor, Occupational Outlook Handbook, https://www.bls.gov/ooh/computer-and-information-technology/home.htm, 2020. (Accessed 8 February 2021).
[32] A.N. Rick Villars, Holly Muscolino, Wayne Kurtzman, Serge Findling, Ritu Jyoti, Dan Vesset, Mario Morales, Jennifer Cooke, Deepak Mohan, Jonathan Lang, Al Gillen, Carrie MacGillivray, No Title, IDC Futur. Worldw. IT Ind. 2021 Predict., https://www.idc.com/research/viewtoc.jsp?containerId=US46942020, 2020. (Accessed 8 February 2021).
[33] R.A. Voorhees, Competency-based learning models: a necessary future, New Dir. Inst. Res. 2001 (2001), https://doi.org/10.1002/ir.7.
[34] Wikipedia Page Views, (n.d.), http://dumps.wikimedia.org/other/pagecounts-raw/. (Accessed 10 February 2021).
[35] WSSU CADS, (n.d.), https://github.com/CADS-WSSU/CADS. (Accessed 10 February 2021).
[36] M. Zaharia, R.S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M.J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, I. Stoica, Apache spark: a unified engine for big data processing, Commun. ACM 59 (2016), https://doi.org/10.1145/2934664.

Debzani Deb (BS'96 SUST, Bangladesh; MS'01 University of Adelaide, Australia; PhD'08 Montana State University, Bozeman) is an Associate Professor of Computer Science at the Winston-Salem State University. She is also the founding director of Center for Applied Data Science (CADS) at WSSU. Previously she also worked as an assistant professor at Indiana University of Pennsylvania, PA and as a visiting faculty at University of North Carolina at Greensboro, NC. She authored and coauthored over 40 referred journal and proceeding articles and has been awarded over 2.6M of federal funding as PI and Co-PI. She has over 15 years of experience in Higher Education as faculty member in three different countries such as USA, Australia and Bangladesh. Dr. Deb is a regular reviewer for journals including Journal of Parallel and Distributed Computing (JPDC), conferences including ACM Technical Symposium on Computer Science Education (SIGCSE), ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE), and IEEE Frontiers in Education (FIE), and workshops including EduPar and EduHPC. She served as the program chair of EduHPC workshops at SC18 and SC19.

Dr. Muztaba Fuad is a Professor of Computer Science at The College of Arts, Sciences, Business, and Education at Winston-Salem State University. His research interests include mobile computing, Self-adaptive distributed systems, and computer science education. His research activities have resulted in several National Science Foundation-funded works and numerous publications in prestigious journals and proceedings. Graduate and undergraduate students' research under his supervision produced peer-reviewed publications in leading conferences. Dr. Fuad is a professional member of The Association for Computing Machinery (ACM) and ACM SIGCSE with active involvement in the professional arena. He is a computer science program evaluator for Accreditation Board for Engineering and Technology Inc. (ABET) and promotes evidence-based approaches in assessing, evaluating, and continuously improving computer science programs. Dr. Fuad is the recipient of the Wachovia Excellence in Teaching Award and Wilveria B. Atkinson Distinguished Research Award at Winston-Salem State University.