Teaching, Learning and Collaborating through Cloud Computing Online Classes

Judy Qiu*, Supun Kamburugamuve*, Hyungro Lee*, Jerome Mitchell*,
Rebecca Caldwell†, Gina Bullock‡, Linda Hayden§
*School of Informatics, Computing, and Engineering, Indiana University Bloomignton
{xqiu, skamburu, lee212, jeromitc}@indiana.edu
†Winston-Salem State University
caldwellr@wssu.edu
‡North Carolina Agricultural and Technical State University
glbulloc@ncat.edu
§Elizabeth City State University
haydenl@mindspring.com

Abstract—**Knowledge of parallel and distributed computing is important for students needing to address big data problems for jobs in either industry or academia; however, many college campuses do not offer courses in these areas due to curriculum limitations, insufficient faculty expertise, and instructional computing resources. Massively Open Online Courses (MOOCs) provide an opportunity to scale learning environments and help institutions advanced curriculum. In this paper, we discuss a Cloud Computing course offered at Indiana University and use it as a model for improving curriculum at institutions, which otherwise wouldn't be exposed to parallel and distributed computing.**

Keywords—Online Education, Cloud Computing, Parallel and Distributed Computing.

I. Introduction

Parallel and distributed computing is becoming ever more important with the exponential growth of data production in areas, such as the web and Internet of Things. Furthermore, modern computers are equipped with multiple processors enabling the need for them to be utilized efficiently. On the other hand, clouds are becoming the standard computing platform for executing both applications and data analytics. With these trends, it becomes increasingly important for the next generation of software engineers and researchers to be familiar with distributed and cloud computing paradigms and how they can be applied in practice, specifically in a parallel fashion. Unlike academia, where one focuses on fundamental computer science problems, cloud computing involves many technologies and software tools widely used by industry and academia for real-world applications, which are now part of everyday life for billions of people. These include Internet-scale web search, e-mail, online commerce, social networks, geo-location and map services, photo sharing, automated natural language translation, document preparation and collaboration, media distribution, teleconferencing and online gaming. However, the underlying fundamentals of these techniques are from different computer science disciplines. including distributed and parallel computing, databases and computer systems architecture. A well-rounded course in cloud computing should cover each of these areas and explain them in the context of cloud computing. To gain practical experience on cloud computing, a student has to master many technologies based on these principles.

In order to facilitate such a learning environment, Indiana University (IU) developed an online cloud computing course [1]; this course has been taught by different faculty for several years for residential students and online students. The course is offered by the graduate program in computer science and data science. Students in the Intelligent System Engineering and Library Science program are also given the opportunity to take the course. The online population of students is geographically located worldwide from London, France, Germany, India to Indianapolis. Most of the students are professionals who take online classes to either update their knowledge and skills or earn a degree.

A primary goal of the course is to maintain the same level of standard as the residential course for the online course. Since this is a programming intensive systems course, it is especially challenging due to limitations on the face to face interactions with online students, such as diverse technical background of students required by the course to be successful. The students are expected to have general programming experience with Linux and proficiency in Java (2-3 years) programming language and scripting. A background in parallel and cluster computing is considered a plus but not required. The statistics present in this paper are related to the latest version of the online course, which had the largest attendance with approximately 160 students, where 100 were residential with the remaining being online students. The popularity of the cloud computing is based from major Internet companies, such as Amazon, Microsoft, Google, IBM, Facebook and Twitter. These companies provide infrastructure, tools or applications in clouds. business, government, academia, and individuals use public or private cloud-based solutions for storage and applications. The course has been used as a model by other institutions to introduce cloud computing to their respective students. This is facilitated by the availability of online course materials. It provides a unique opportunity for collaboration between Elizabeth City State University (ECSU) and Indiana University in remote sensing using cloud computing to involve faculty

[1] http://cloudmooc.appspot.com/preview

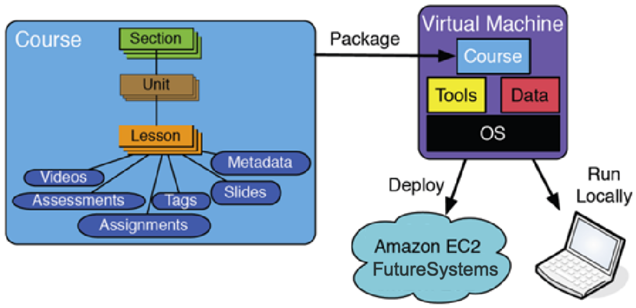

Fig. 1. Model for the MOOC Course Content and Delivery using Cloud

and students from minority serving institutions (MSI) by exploiting enhancements using cloud computing technologies. Computational and data sciences are important areas, which have the capability to host both parallel computations (using MPI and Hadoop) and learning resources (online MOOC) allowing for an attractive focus for universities without a major research history to participate on an equal footing with research intense universities.

The rest of paper is organized with section II curriculum development and course organization, followed in section III course scaling and techniques, in section IV evaluations of the course outcome and knowledge growth for students, and section V ADMI Cloud for scaling the model to MSI institutions. Finally, in section VI we summarize the challenges, impact and future work in modernizing curriculum and workforce development.

II. Curriculum Development and Course Organization

The course is aimed at teaching the basic principles of parallel and distributed computing by exploring applications related to cloud environments. This is a graduate level course with large emphasis on programming and expects prior knowledge of programming in order to be successful. The course follows the cloud computing text book [1]. By the end of this course, students are expected to learn key concepts in cloud computing and have hands-on programming to be able to solve data analysis problems on their own. The organization of the course is shown in Fig. 1.

A. Course Content

The course uses the Google Course builder as the content hosting platform. Google Course builder provides a way to host course content. Its source code is distributed under the Apache License version 2 and is free to modify and redistribute. An individual instructor can develop a course with the features, and since course builder is open source, an instructor can modify the source code to create a more personalized version. The final completed course should be deployed in Google infrastructure using the Google App Engine.

The course content is composed of lecture videos hosted in YouTube. A text version of the content is also possible. The course has been structured as a set of units. Each unit contains a set of lessons with lessons as videos followed by an activity. The instructor creates an activity as a javascript file. The activity contains either multiple-choice questions or text based answer questions with specific answers. Between units there can be course assessments. These assessments can be quizzes, midterm exam and final exam. They also have the same format as activities followed by lessons and features multiple-choice questions and simple text based answer questions. The activities and assessments can be graded and the scores are displayed in the student profile.

The course consists of six units starting with cloud computing fundamentals [1]:

- Chapter 1: Enabling Technologies and Distributed System Models
- Chapter 3: Virtual Machines and Virtualization of Clusters and Datacenters
- Chapter 4: Cloud Platform Architecture over Virtualized Datacenters
- Chapter 6: Cloud Programming and Software Environments
- Chapter 9: Ubiquitous Clouds and The Internet of Things

The course also incorporates five units of state-of-the-practice and hands-on projects. They are organized as infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (Saas), cloud data storage, data analysis and machine learning (ML) applications.

- How to Start VMs (IaaS)
- How to Run MapReduce (PaaS)
- How to Run Iterative MapReduce (PaaS)
- How to Store Data (NoSQL)
- How to Build a Search Engine (SaaS)

Each unit consists of multiple lectures with videos. There are a total of 76 lecture videos were recorded by the instructor with the help of a professional staff for video recording and editing. It took a lot of effort and time to get the videos properly recorded in the first time of offering the course. After the initial videos were finalized it was relatively easy to add more content or update the videos for later offerings of the course.

B. Projects

The course was offered with a comprehensive set of eight cloud application projects, which are interlinked. The overall goal is to build a web search engine. Students can use various tools to build the system one component at a time using cloud based data analytic platforms. The first six projects use Hadoop [2], HDFS [3] and HBase [4] as data processing technologies. The dataset used by the projects were ClueWeb09 [5] available for educational purposes. We only used a moderate datasize from the original because of the resource constraints.

The projects are packaged into a virtual machine and a student can download it to execute projects on his or her machine or on a cloud provider, if they chose. The course expects students to execute projects on their own local machines at the start and migrate to production distributed environments. Each project is accompanied by a video, which explains the project in detail with steps on how to build and execute the project.

The projects start with a small activity, which involves configuring and running a simple Hadoop program. The first building block of the search engine expects students to write a pagerank [6] algorithm in Hadoop to measure the importance of web pages. Next, HBase, a distributed storage, is introduced in order for the students to create an inverted index from word to page to facilitate the search. The next step is to combine the results from pagerank and use the inverted index to do actual searches.

Apart from the search engine projects, students are expected to implement two more applications - a graph algorithm as well as a standard machine learning algorithm using Harp [7], machine learning platform developed at Indiana University. These two projects are related to advanced topics and aimed at teaching students about complex data analytics and how to use parallel processing to speed up a sequential algorithm.

It is a steep learning curve for students to program in a distributed environment. To make it easier to understand, we introduce Single Program Multiple Data (SPMD) as the basic parallel programming paradigm and show detailed steps including data partitioning, execution and communication. For the latter, we further introduce 4 parallel computation models (Locking, Rotation, Allreduce, Asynchronous) for ML based on their synchronization mechanisms and communication patterns. Since each application may have multiple solutions, we recommend students to follow the process and identify a proper parallel pattern for the implementation, and then select a framework such as Harp[8], Spark[9] and Flink[10] to program. Clearly, instructions that separate mechanism from implementation enable in-depth discussions and clarifications over a spectrum of problems and solutions. Students are also encouraged to compare and explain the differences between the choices, either use performance benchmark or discuss their usability. A standard scaling test is based on measuring the execution time and speedup of an application. Initially the algorithms are tested on a single VM and student can use the cloud environment to scale them to multiple nodes. Students are required to draw performance charts, analyze the results and explain possible reasons that lead to non optimal outcomes in their project reports.

C. Assignments & Exams

Assignments are mainly focused on testing basic knowledge about subject matter. Most questions are selected from the text book. Reading assignments were given weekly or bi-weekly. Five quizzes, a midterm and a final were given in class.

D. Student Evaluations

Students are evaluated based on their performance to meet the learning objectives for the class. This include evaluations of eight programming projects, written assignments, and two exams. The exams are focused on core concepts of cloud computing and related underlying principles. For online students, the exams are conducted using Canvas platform and the Adobe connect video conferencing. The projects were graded based on completeness of programming, correctness of results, clarity of analysis in the report, and effectiveness of optimization. Feedback is given to individual student in the grade book and common issues are discussed with students in the lab sessions.

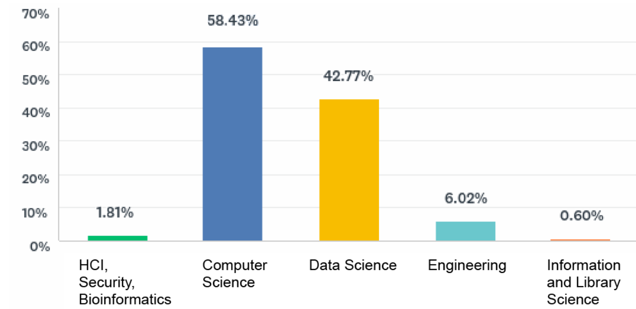

Fig. 2. Departments where the course is cross-listed among five different programs: Informatics, Computer Science, Data Science, Intelligent Systems Engineering, Information and Library Science

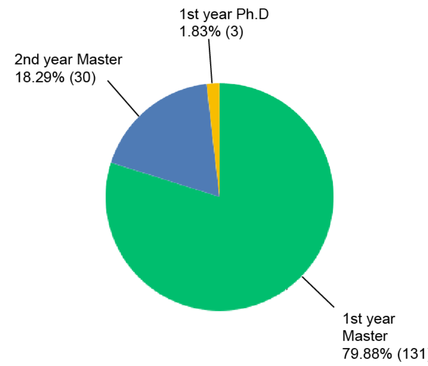

Fig. 3. Student's Level: 81% students in their first year, 19% students in their second year

III. COURSE SCALING AND TECHNIQUES

A. Audience and Diverse Background

The course was targeted towards a wide audience from different backgrounds. As shown in Fig. 2, we found that the student distribution ranged from Informatics, Computer Science, Data Science, Engineering, Information and Library Science, to Industry with diverse knowledge and background about the subject matter and in general of the field.

In the beginning of the class, we provided to understand students' background and expectations. The course is offered to five different programs and therefore collecting survey data is necessary to estimate students' level and preparation for the class. Figures 2, 3, and 4 show course needs to explore several Hadoop-oriented technologies in dealing with big data on cloud computing. Although prior knowledge of the field

Fig. 4. Students' Interests about the course in Cloud Word View

is desirable, most students expressed lack of experiences on these new technologies since they are in the first year of their graduate study. We also observed students eagerness to learn on a wide range of topics about parallel computing, with particular software, such as Apache Hive, Spark, Pig and Lucene being of interest.

B. *Forums*

Since the course is offered to a large number of online and residential students from different time zones and different professions, providing interactive support of course materials, especially about hands-on projects with code implementations is one of the challenging tasks for instructors.

We experimented with several options for class forums, which is a vital part of the course. Because the large class size, an instructor is not always possible to solve problems encountered by an individual student in person. In previous years, the course used Google Forums[2], Indiana University internal forums, and Piazza[3] forums, and we concluded Piazza to be the best option.

The web-based tool, Piazza, is mainly used for communication between instructors and students. Our statistics show 84% of questions received responses within 61 minutes, in average. Fig. 5 shows an overall activities on Piazza in enabling online collaboration of the class.

C. *Hands-on Labs*

The course is organized with biweekly projects to encourage active developments in source code writing and connect between the textbook and latest technologies. The fundamental pedagogy underlying these hands-on projects is to embrace new experiences in learning both theory and practice with minimal barriers, for example, learning a new programming language or preparing computing environments with recent software tools, which takes effort and time to achieve. Fig. 7 gives an indication of students' programming ability associated with project developments of the class. Many students have at least 1 or 2 years language experiences among Java, C#, C++, C and Python, which are abundant to start basic code developments in most assignments.

One of the challenging activities in teaching from previous classes is building a controlled experimental environment over different computing platforms. We built a virtual machine (VM) image to avoid hassles and the choice of computing environments given to students based on their confidence level. The VM image is able to run on a student's desktop via VirtualBox. The transition from using a desktop to run jobs on the cloud environment is a steep learning curve. The labs provide students with step-by-step instructions on how to install and configure a Hadoop cluster on OpenStack Kilo on FutureSystems, cloud computing resources at Indiana University. Their applications can execute on production clusters for projects, such as Hadoop PageRank, BLAST, WordCount, and Harp Mini Batch K-means. The labs were designed and organized to guide students from basics of cloud computing to configuring and running parallel applications in such environments. Each lab is accompanied by detailed text explanations and a video describing the relevant material.

[2]https://groups.google.com

[3]https://piazza.com/

D. *Online Meetings*

To facilitate the questions from students regarding both course content as well as projects, online meetups were conducted every week. These were one hour sessions mainly steered by associate instructors but participated by the instructor. In early course offerings, only a small number of students chose Google Hangouts platform for online meetings. But with larger classes, we switched to Adobe Connect platform for online courses. Every online meeting is recorded and available by YouTube for later viewing by the students. We find the videos are helpful in subsequent runs of the course.

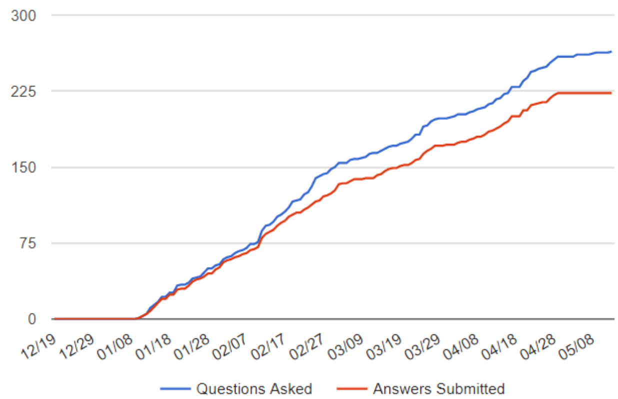

Fig. 5. Questions and responses received on Piazza over the semester

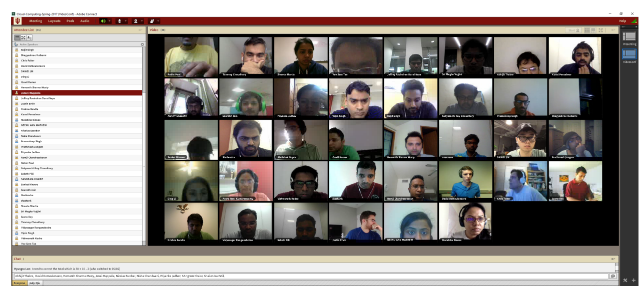

Fig. 6. Supporting remote students through video conferencing

Adobe Connect (now replaced by Zoom) is also provided for weekly class lab session and office hours to instruct how to complete course assignments with step-by-step tutorials and provide individual feedback. Our experience indicates these tools ensure effective learning for students and productive course management by instructors. Fig. 6 is a sample screenshot capturing during a normal video sessions. The chat window at the bottom allows us to have a public and one-to-one conversation among participants and the main window alternates between either a presentation or screen sharing modes for lectures and tutorials. Also, recordings for these sessions have been used for self-study, in case students need to re-visit materials covered in those sessions.

E. *Content Repository*

An innovation from this project is to build on our extensive experience with online education and its technologies, to use MOOC technologies, and build an open source community, X-MOOC repository to explore a modular and customizable process for storing, managing, and sharing course content and learning materials. The developer of the course found the need to share content among different courses run by different instructors. In order to do so, a MOOC platform should be able to share course content among different courses. As part of this course, we developed technology on top of the edX MOOC platform to share content among different courses and moved the content of this course, which is currently in Google course builder to the edX platform. This development will allow instructors from different universities to easily share course content and quickly create new courses, which modify the old.

Link function adds a link to related course videos in assessments. This function can navigate students through specific course content for review. It is particularly helpful when students make mistakes in a quiz or an exam but are unclear about the missing knowledge. The instructors can provide a list of keywords and their associated video tags for each question. If a student chooses an incorrect answer, these video links will automatically appear as hints (buttons) under the question.

We have created and implemented a *playlist* function on both edX and Google course builder. It allows customizable selection and arrangement of lessons according to preference. The user interface provides a drag and drop function with easy interactions. Instructors can assemble a new course from modules in a shared repository on the fly. Students can use this function to navigate through the most important lessons in the course modules more efficiently.

Interoperability could include an automated process in sharing courses between edX and other course hosting platforms including Hubzero. This will allow the authors to freely move from one platform to another. The edX courses are already exported as XML content, so we need to find mechanisms to convert them to an accepted format for other online sites. By tagging those course modules with metadata, we can classify and organize shared course materials and make them easily searchable for others. We can further set up learning objectives and review, rate, and provide feedback on them to ensure a high quality online learning experience.

IV. Course Outcomes and Evaluations

In addition to standard institutional class evaluations, we conducted a post course survey to gather feedback on the details of course content and measure the growth of students in obtaining knowledge and skills. For the cloud computing course, we wanted to know the preference of students in using a VM instead of a distributed environment for the projects. As Fig. 9 show, the majority preferred a VM. Such a preference can be stemming from VM being easier to setup and run programs rather than a distributed cluster.

The initial survey asked the students to rate their preparation in different cloud technologies related to the projects. It is observed in Fig.7 that many students have 0, 1-2 years or no programming experience, although Java is a familiar language to students. As evident in Fig. 8, half of the class had knowledge using VMs or Cloud but most lack experience in using tools in distributed and parallel computing environments. Data analytics skills are desirable in particular as shown in

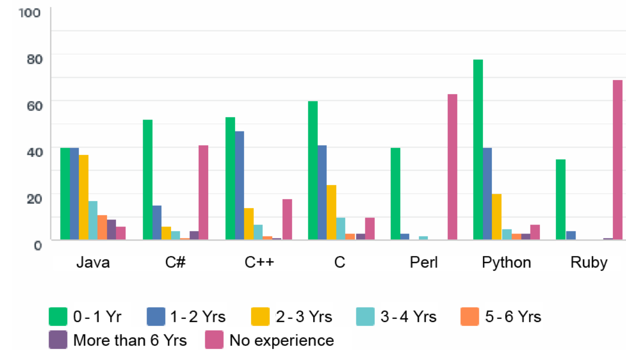

Fig. 7. Student Programming Language Experience. The y axis shows the number of student who are proficient in a given language and number of years.

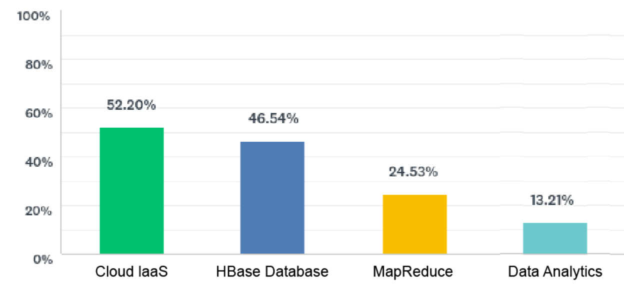

Fig. 8. What Do Students Already know? - Prior Knowledge at the First Day of Class; IaaS (i.e. AWS, Azure, GCE, OpenStack) is one of the existing skills whereas MapReduce, Iterative MR Model (i.e. Twister, Harp), and Data Analytics are desired knowledge to learn during the course.

Fig. 8. Fig. 10 illustrates the knowledge growth of students as seen by themselves in different areas in a scale of 1 to 5 according to the post class survey. Fig. 11 shows the average score of projects related to these areas. The projects are scored according to the correctness of the solution and how efficient the solution is. Also students are expected to write technical reports about details of the implementation and relevant technologies. The correctness of the answer was given more weight. It is hard to master these cloud technologies in a semester as they require vast knowledge and experience. To bridge the gap, having an average knowledge after taking the course seems to be a reasonable expectation.

We observed that the students find difficulties in solving theoretical problems with coding in a distributed environment. For example, one of the projects was about implementing mini-batch K-means algorithm in parallel and most students addressed a lack of backgrounds and technical difficulties according to the number of conversations on Piazza and the grade from Fig. 11. About one third of the questions (29/100) related to the projects were about the K-Means programming task and the followup discussions had made over four weeks between instructors and students. We therefore spent extra time during lab sessions to share common issues and discuss challenges for supporting students who may have different levels of knowledge in distributed computing environments and varying approaches in solving the given tasks. Code snippets, actual log files and configuration settings were useful in helping students learn to implement a program.

We are evaluating the course each year and try to add new

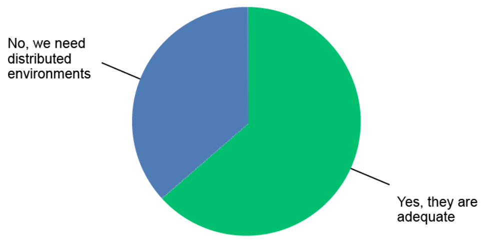

Fig. 9. Survey on single VM environment and preference

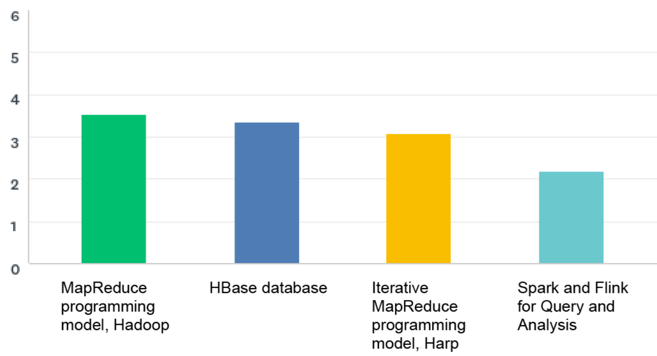

Fig. 10. Student knowledge growth in different cloud technologies on a scale of 1 to 5 from low to high. The averages are shown.

content as well as changing the projects and assignments to facilitate the integration of new knowledge and fast evolving technologies. The course also offers students with extra motivation to take on research projects with the instructor to further enhance their knowledge in the field.

V. ADMI Cloud

The course offered at Indiana University introduced students to parallel and distributed computing and its scalability provided an opportunity to model it for under-served universities. The course was adapted to support institutions involved with the Association of Computer and Information Science/Engineering Departments at Minority Institutions (ADMI). The ADMI

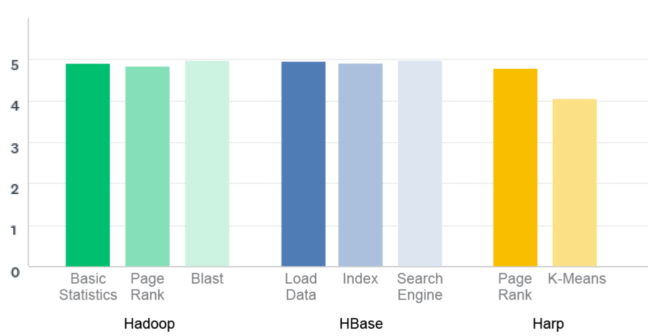

Fig. 11. Student grades in projects related to Hadoop, HBase and Harp. The marks are scaled 1 to 5 from low to high. Eight projects are listed in a row showing the marks for each project given on a specific topic.

cloud focused on collaborative curriculum development and research in remote sensing; this cloud attempts to share modules with participating minority institutions, and if a faculty member is not well-versed in parallel and distributed computing, training seminars were provided throughout the year to support his or her concerns. As an initial module for the ADMI cloud, Elizabeth City State Univeristy's Department of Mathematics and Computer Science offered RS 506 The Principles of Microwave Remote Sensing; this course introduces spaceborne remote sensing of Earth's atmosphere, land, and oceans. The primary methods and applications of microwave remote sensing are considered with both active and passive techniques. The ADMI cloud enables universities to participate using RS 506 course and involving the computational aspects related to it. In this module, students learn about the theory associated with radar remote sensing and apply their knowledge of computational constraints using resources provided by Indiana University, which in addition, provide topics, such as performance evaluations and cloud environments and programming models (Hadoop and MPI) with clouds being the venue to facilitate scalability.

Discussions started with ADMI participating in a summer 2016 training program, which they were trained to use MOOCs, as a first phase of the project. These training exercises were conducted during a three day session at Elizabeth City State University. During the session, participants were encouraged to develop initial content of classes taught at their home institutions. IU faculty and graduate students provided instructions using a mix of virtual and residential modes. The workshop activities include development and delivery using MOOCs for targeted ADMI computer science courses. Two follow up professional development activities were further provided at the ADMI meetings. One is an ADMI Curriculum Enhancement using Cloud Computing and MOOC Workshop, another is the ADMI conference in 2017. In the meetings, the concepts of cloud computing were presented in order to provide information for hosting and sharing new courses, and the ADMI faculty continued discussions from the summer to help exchange ideas about how to implement MOOCs for their classrooms; this technique is well-established and often termed teach the teachers or train the trainers and is studied in the context of professional development for teachers [11] [12] [13] [14] [15].

Although RS 506 was an initial pilot for the ADMI cloud, ADMI faculty were encouraged to add classes offered at their home institutions so others participating members could benefit. As an extension, Applied Java and Robotics offered at North Carolina A&T (NCAT) and Winston-Salem State University (WSSU) sharaed skills and techniques supporting the economic development of students for jobs in the computer and data sciences.

VI. Discussion and Future Work

The Cloud Computing course has been offered for many years to the residential computer science graduate students at the Indiana University and has observed high enrollments each time. The course offers a mix of core concepts of distributed and parallel computing along with their practical applicability. This combination has been proven to be successful in teaching a diverse group of students who are primarily looking towards industry which increasingly demands engineers with experience in distributed and parallel computing domains. Faculty and IU support have helped develop a curriculum for remote sensing materials and this will allow other institutions within the AMDI community to reuse existing materials in order to foster a community of learning.

Clouds and online MOOCs offer cutting-edge technologies to enhance traditional computational science curriculum and research with next-generation learning metaphors. There are many challenges in scaling the course and providing a robust learning environment. We have developed specific methods for effective teaching of large classes (hands-on labs), accommodation for individual student needs (forums, online meetings), and customization for interdisciplinary collaboration (content repository), as well as extensive engagement, outreach and training for a broader community.

Cloud Computing is a fast evolving area with new technologies, processing models and frameworks added daily to the mix. The students of the course would be beneficial if they can work with the latest technologies especially through the projects. It is challenging to keep up to date in such a dynamic environment while maintaining and updating the course content at each offering.

This project builds from existing Indiana University activities, involving REUs for ADMI and other undergraduates, two cloud-related courses offered in Computer Science and Data Science programs. The project activities will include course development and delivery using MOOCs for a cloud-enhanced classes taught by ECSU and other institutions and IU faculty with a mix of virtual and residential modes. The course outcomes will be evaluated to understand the best practices of such shared curriculum across multiple disciplines and institutions. The prototyped cloud-based course modules are made available as examples of open source community X-MOOC repository [16]. For future work, we will continue modernizing curricula that is suitable for our next generation workforce development and connect to the community by systematically introducing multiple courses, teacher training, research support and electronic resources sharing across the ADMI MSI and other teaching university networks.

Acknowledgment

The authors are grateful for the generous support from NSF EAGER Grants 1550784 and 1550720 on Remote Sensing Curriculum Enhancement using Cloud Computing. Google grant on Customizable MOOC for Cloud Computing supported the initial development and offering of the online course. The Harp open source software has been developed and used by students for their course projects, and we gratefully acknowledge generous support from the Intel Parallel Computing Center (IPCC) grant, NSF OCI-114932 (Career: Programming Environments and Runtime for Data Enabled Science), NSF DIBBS 143054: Middleware and High Performance Analytics Libraries for Scalable Data Science. We would like to thank the students who participated in the surveys to provide feedback on the course. We would like to extend our gratitude to associate instructors who worked on this course over the years.

References

[1] K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann, 2013.

[2] T. White, *Hadoop: The Definitive Guide*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2009.

[3] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, May 2010, pp. 1–10.

[4] L. George, *HBase: the definitive guide: random access to your planet-size data*. " O'Reilly Media, Inc.", 2011.

[5] J. Callan, M. Hoy, C. Yoo, and L. Zhao, "Clueweb09 data set," 2009.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[7] B. Zhang, Y. Ruan, and J. Qiu, "harp: Collective communication on hadoop," in *2015 IEEE International Conference on Cloud Engineering*.

[8] Harp: A collective communication library for big data machine learning. [Online]. Available: https://dsc-spidal.github.io/harp/

[9] Spark: A general engine for large-scale data processing. [Online]. Available: https://spark.apache.org/

[10] Flink: An open-source stream processing framework. [Online]. Available: https://flink.apache.org/

[11] J. Van Orshoven, R. Wawer, and K. Duytschaever, "Effectiveness of a train-the-trainer initiative dealing with free and open source software for geomatics," in *Proceedings (J.-H. Haunert, B. Kieler and J. Milde, Eds.) of the 12th AGILE International Conference on Geographic information Science*, 2009.

[12] B. Fishman, S. Best, J. Foster, and R. Marx, "Fostering teacher learning in systemic reform: a design proposal for developing professional development." 2000.

[13] J. H. van Driel, D. Beijaard, and N. Verloop, "Professional development and reform in science education: The role of teachers' practical knowledge," *Journal of Research in Science Teaching*, vol. 38, no. 2, pp. 137–158, 2001. [Online]. Available: http://dx.doi.org/10.1002/1098-2736(200102)38:2¡137::AID-TEA1001¿3.0.CO;2-U

[14] D. Hestenes, "Toward a modeling theory of physics instruction," *American Journal of Physics*, vol. 55, no. 5, pp. 440–454, 1987. [Online]. Available: http://dx.doi.org/10.1119/1.15129

[15] H. Borko, "Professional development and teacher learning: Mapping the terrain," *Educational Researcher*, vol. 33, no. 8, pp. 3–15, 2004. [Online]. Available: http://dx.doi.org/10.3102/0013189X033008003

[16] X-mooc repository: Curriculum enhancements with cloud and mooc for online learning. [Online]. Available: http://cloudmooc2.soic.indiana.edu/