# A Cloud Computing Course: From Systems to Services

M. Suhail Rehman[1], Jason Boles[2], Mohammad Hammoud[2], Majd F. Sakr[1]
suhailr@cs.cmu.edu, jboles@cmu.edu, mhhamoud@cmu.edu, msakr@cs.cmu.edu

[1]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA

[2]Carnegie Mellon University in Qatar,
Education City, P.O Box 24866,
Doha, Qatar

## ABSTRACT

We have designed, developed and administered a course on cloud computing that was taught to over 700 students at our institution over two years. The goal of this project-based course is to provide students with foundational systems concepts as well as experience in developing the required skills to design and deploy viable, robust and elastic web-services within performance and budgetary constraints. We present our objectives, learning outcomes, projects, learning model, outcomes and lessons learned. So far, for this demanding course, our student retention rate is above 80% and enrollment is doubling every year.

## Categories and Subject Descriptors:

K.3.2 [**Computers and Education**]: Computer and Information Science Education – *computer science education, curriculum*

## General Terms: Design

**Keywords:** Distance Education; Web-Based Techniques; Instructional Technologies; Cloud Computing.

## 1. INTRODUCTION

Cloud computing offers a paradigm shift in the way organizations procure, use and manage information technology. The cloud offers computing facilities on-demand, served over a network from a shared pool of resources, typically residing in a large data center. Cloud computing offers a new approach to the way applications are written and deployed.

One of the primary benefits of cloud computing is *elasticity*, whereby resources can be automatically expanded during demand spikes, or contracted during demand lulls; thereby improving user experience and system utilization. As the cloud model is rapidly gaining acceptance, industry is scrambling to find experienced people who can effectively leverage, employ and operate the model. In contrast, academia has not responded quickly enough (so far) to offer a satisfactory number of well-trained cloud computing engineers, architects, developers and researchers.

To help fill this critical void, we designed a novel course on cloud computing that was taught to over 700 students across over the last two years. The course methodically covers the building blocks and enabling technologies of cloud computing, including data centers, virtualization, cloud storage, cloud programming models, and analytics engines, among others. We were successful at accomplishing our specified pedagogical goals, and learned a great deal of lessons throughout the process. As a result, the course became popular at our university, and currently enrollment is doubling every year.

In this paper, we report on our experience in offering the course. In particular, we discuss the course content, its design and implementation, outcomes, and the lessons learned. Our objective is to share our experience with instructors who aim at offering similar courses in cloud computing.

## 2. MOTIVATION & GOALS

Cloud computing is an emerging technology that affects users, developers and organizations alike. With the fast growth of the cloud, it is very likely that computer science graduates, at various levels, will find themselves evaluating, developing and/or deploying systems, services and applications for the cloud.

A recent report from the Bureau of Labor Statistics (BLS) indicated that hiring in data processing, hosting and related services has been growing steadily in 2013. Over 3600 new jobs were added in July 2013, marking the strongest growth since June 1998[36]. The report also projects that over 671,300 new jobs will be added in this sector by 2020. Alongside, the International Data Corporation (IDC) reported that two-thirds of the industry are either planning, implementing or using cloud computing, with 50 percent of businesses acknowledging that cloud computing is currently a top priority[21].The report also estimates that 1.7 million cloud-related positions remained unfulfilled worldwide due to lack of necessary skills and/or adequate training.

Moreover, students who are planning to pursue research in computer science or other fields will find that cloud computing is increasingly being adopted in a variety of domains, from Big Data management and storage, to analytics, through computational science, to online education; hence, making studying cloud computing crucial.

Our goal is to develop students' foundational knowledge in systems and hands-on skills in cloud services in order to equip them with the skills necessary to meet current and future industry demands as well as to enable them to carry out applied research on the cloud. To achieve this task, we intentionally placed the course content and projects at the intersection between systems concepts and cloud services.

*SIGCSE '15,* March 4–7, 2015, Kansas City, MO, USA
Copyright © 2015 ACM 978-1-4503-2966-8/15/03…$15.00
http://dx.doi.org/10.1145/2676723.2677298

# 3. DESIGN

## 3.1 Intended Audience

This course was tailored for junior and senior undergraduate Computer Science and Engineering students as well as graduate students from various technical majors. The course was designed as a 9-unit (i.e., 9 work-hours per week) course for undergraduate students and a 12-unit course for graduate students. The extra 3 units, for graduate students, require them to work on a large team-based project. Students who wish to take the course must have completed an introductory computer systems course as a prerequisite.

## 3.2 Course Objectives

The course was designed to be an introductory course on cloud computing. It provides an overview of the field, while offering hands-on experience with various cloud computing services and infrastructures. Students gain a broad understanding of the economic and technological factors that led to the emergence of cloud computing. In addition, they attain a considerable comprehension of the main building blocks (e.g., resource sharing, storage and programming models) that compose a cloud system and the ensuing cloud services.

We also struck a balance between theory and practice in the course to provide our students with satisfactory preparation to target either industry or academia. The projects were designed to give practical experience with several cloud enabling services and systems such as Amazon Web Services (AWS), Hadoop[40], and NoSQL systems[37]. In short, we attempted to craft a course that covers the cloud computing field with sufficient depth and breadth. We intended to allow students to easily proceed with, and hopefully excel at, a cloud-related position in industry or academia.

## 3.3 Learning Outcomes

Given our motivation, goals, intended audience and objectives, we specified the following high-level Learning Outcomes (LOs) for the course. Specifically, after finishing the course, students are expected to: (1) explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, and the characteristics, advantages and challenges introduced by the various models and services in cloud computing; (2) apply the fundamental concepts in data centers to understand the tradeoffs in power, efficiency and cost; (3) discuss the virtualization technology and outline its role in enabling cloud computing; (4) illustrate the key concepts of cloud storage such as Amazon S3[3] and HDFS[14], DynamoDB[5], HBASE[27] and; (5) analyze different cloud programming models such as Elastic MapReduce[2]; (6) develop working experience with big-data analysis on public cloud platforms; (7) demonstrate the high-level use of resource management front-ends to develop elastic applications to meet performance constraints; (8) demonstrate proficiency in deploying, and comparing and contrasting new generation cloud-based storage systems; (9) employ MapReduce to solve a real-world data analytics problem; (10) design, assemble, deploy and evaluate an end-to-end web-service solution for a very large data set given performance and cost constraints.

## 3.4 Bottom-Up Approach

The course organization was motivated by presenting concepts from the bottom-up to facilitate learning, performance and retention. In particular, the sequence of topics started with an introduction, then presented data centers and their design considerations, virtualization technologies, cloud storage solutions and then programming models. The approach was to gradually peel away the layers that comprise any cloud. For each concept, we started by presenting a bird's eye view of the concept prior to delving systematically into the details. By that, we sought for students to easily grasp the material and effectively piece together the various cloud technologies and principles. We further attempted to teach students, not only how a certain system (e.g., MapReduce) works, but also how it can be built so as to enable their creativity in suggesting novel ideas and systems. As an example, in Unit 5, we initially discussed the foundation and the pros and cons of every programming, computational, parallelism and architectural model by which a cloud analytics engine can be constructed, and, subsequently, demonstrated how MapReduce and other frameworks apply such foundations and models.

# 4. TOPICS COVERED

We now elaborate on the topics that we covered throughout the course. Specifically, we present the main concepts that we studied in each of the five Units, the introduction, data centers, virtualization, cloud storage, and cloud programming models and analytics engines Units, respectively.

## 4.1 Introduction

As an introduction to the cloud computing field, we discussed in Unit 1 the motivating factors, benefits, challenges, economic and business models, service models (i.e., software-as-a-service, platform-as-a-service and infrastructure-as-a-service), deployment models (i.e., private, public and hybrid clouds), and the main building blocks (i.e., data centers, virtualization, storage, and analytics engines) of cloud computing.

## 4.2 Data centers

Data centers constitute the infrastructure of every cloud deployment model, being private, public or hybrid. They enable many of the economic and technological benefits of the cloud paradigm. In Unit 2, we described several concepts behind data center design and management. In particular, we provided historical overview of data centers, outlined the major components and architectures of modern data centers, and indicated the design considerations for building cloud infrastructures for various cloud deployment models.

## 4.3 Resource Sharing and Virtualization

Virtualization is an important layer in cloud computing. Virtual resources are typically constructed from the underlying physical resources and act as proxies to them. In Unit 3, we studied various ingredients of the virtualization technology and the crucial role it plays in enabling cloud computing. Specifically, we first identified major reasons for why virtualization is becoming essential, especially for the cloud. Second, we discussed how multiple hardware and software images can share a single resource while provided with security, resource and failure isolations; all being critical for cloud computing. Third, we examined two virtual machine types, process and system virtual machines. Lastly, we studied how CPU, memory and I/O resources are virtualized, with examples from Xen[12] and VMWare[9]. We concluded with a representative case study, using the Amazon Elastic Compute Cloud[1].

## 4.4 Cloud Storage

The cloud storage is used as a repository for Big Data. In Unit 4, we described storage concepts for clouds, including problems of Big Data scale, management, distribution, durability, consistency and redundancy. We detailed the three distinct layers of cloud storage; that is, data (i.e., structured vs. unstructured and static vs.

dynamic), abstractions (i.e., file systems, databases and object storage), and physical and virtual block devices. We concluded with multiple case studies, including HDFS[14], PVFS[34], Apache HBase[27], Apache Cassandra[30], and Amazon's S3[3].

## 4.5 Programming Models and Analytics Engines

Researchers and practitioners have evolved various models that mitigate the challenges for designing and implementing software systems that successfully exploit the capabilities of the cloud. The current generation of these models builds upon *classical* predecessors. In Unit 5, we studied some classical programming (e.g., shared-memory and message-passing), computational (e.g., synchronous and asynchronous), architectural (e.g., master-slave and peer-to-peer), and parallelism (e.g., data-parallel and graph-parallel) models by which cloud programs and analytics engines can be constructed. Afterwards, we present popular cloud analytics engines (such as MapReduce) and designate the programming, computational, architectural and parallelism models that each of them adopts and offers.

# 5. LEARNING MODEL

Cloud computing has been taught twice at our university as a traditional course with typical in-class lectures and recitations. From spring 2013 semester onwards, the instructors decided to transform this course to an online version so as to enable:

- Students from across the globe to access elite college materials and instructions
- Students to learn at their own pace and time
- Students to choose among a variety of learning styles and explore course material in an interactive fashion.

With these objectives in mind, we designed and offered an introductory undergraduate/graduate online course in cloud computing, which suits the needs of the students across our university's global campuses.

The learning model that we chose to deploy this course on engages students in four different settings (Figure 1). Particularly, students access the course content on the Online Learning Initiative (OLI) Platform[7], ask questions on Piazza[8], complete projects on Amazon Web Services (AWS)[1] and attend, or watch videotaped, weekly recitations.

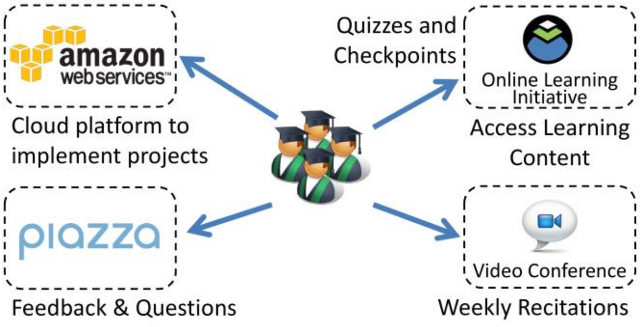

**Figure 1. The learning model for this course**

## 5.1 Online Learning Initiative (OLI)

The Open Learning Initiative (OLI) is a grant-funded effort at Carnegie Mellon University. OLI's approach to instructional design is outcome-driven, evidence-based, and technology-assisted. Its infrastructure offers high-quality, scientifically-based, and classroom-tested tools for online course design, development, delivery, and assessment[39].

OLI provides a web-based platform which allows developers to encode course content, including text, images and videos. The content is structured hierarchically as *Units* that contain a sequence of *Modules*. In return, each Module constitutes a sequence of *Pages*. OLI allows developers to embed various types of activities and assessment methods. A crucial aspect of OLI's web platform is the ability to provide hints and specific feedback based on students' responses to different activities. In addition, OLI offers a learning dashboard which provides extended information to educators. For example, instructors can quickly see which topics are mostly confusing students or how individual students are overall performing.

## 5.2 Amazon Web Services (AWS)

In a course on cloud computing, it is necessary to give students hands-on experience with a public cloud platform. For that sake, we chose Amazon Web Services (AWS), an industry-leader in online cloud computing services. AWS has fairly wide portfolio of cloud services including Compute (i.e., Amazon Elastic Compute Cloud), and Storage (i.e., Amazon Elastic Block Store and Simple Storage Service) services, to mention a few. According to[31], AWS is the most widely-used cloud platform, making many of its services de-facto industry standards. Of course, the cloud computing market is still evolving. As such, we tried to ensure that the basic skills that our students get on AWS are transferrable to other platforms or providers.

Although our university has multiple private clouds, we avoided using any of them in this course. This is because on our private clouds (and typically on similar private clouds), there is always the risk of not having enough resources to satisfy the class demand, especially during peak periods (e.g., at submission deadlines). Such a risk is completely eliminated on AWS.

## 5.3 Piazza

In addition to OLI and AWS, students in this online course need to clarify doubts and conduct discussions with their peers. We chose Piazza, an online Q&A platform that is designed for university courses. Piazza is easy to use and popular among students. Students and instructors can post notes and queries on Piazza. In return, queries can be answered by instructors, TAs, or other enrolled students.

## 5.4 Recitations

In order to allow students to obtain some face-time with the instructors, weekly recitation sessions were arranged and videotaped. During these sessions, announcements were made and pressing issues were discussed. When necessary, certain aspects of the programming projects were demoed for the students' benefit.

# 6. DEVELOPMENT

Course development commences after defining the course objectives, LOs and course outlines. This phase included the development of the individual Units and their content, activities for students to practice what they learn, quizzes to test and assess student learning, and programming projects to develop expertise in cloud systems and services. Although we have offered the course in the traditional format twice before, the course had to be re-designed to better suit the web-based style of OLI, and the material had to be updated and evolved to reflect the latest developments in cloud technologies.

## 6.1 Student Assessment

As with any course, a good LO assessment plan is important to gauge the progress students have made in retaining and applying the knowledge offered by the course. In this course, student assessment is done in two forms: The end-of-unit quizzes and programming projects. Since the hands-on experience is a main objective of this course, we chose to emphasize the programming projects by making 75% of the final grade assigned to programming projects and 25% to end-of-unit quizzes.

### *6.1.1 Quizzes*

Each of the 5 Units of the course comes with a checkpoint quiz, which the students have to complete within a stipulated deadline. The quizzes consists of 20-50 questions (depending on the Unit), and typically take 2-3 hours to complete. Each question is selected from a pool of questions randomly to limit cheating. The quiz questions are encoded as a checkpoint activity in OLI, which will track the student progress and will automatically grade the students on their work. Students will have one attempt at the quiz before the due date. After the due date, they will receive their scores and feedback for each individual question they have attempted.

### *6.1.2 Projects*

The programming projects in this course were designed to give the students hands-on experience with systems concepts and cloud services that Amazon provides. The projects were designed to be easily transformed to other public cloud platforms. The projects address the hands-on LOs of the course. We designed 5 projects, the details of which are summarized below:

Project 1 was designed to be the introductory project, whereby students can get familiar with AWS and big data. After creating accounts and linking them with the course for payment purposes, they used EC2 instances to explore a data set, and created simple UNIX tools/scripts as well as sequential code to extract information and answer questions based on a small sample of the data set. Afterwards, a much larger version of the data set was provided for students to analyze. It would be impractical to analyze the larger data set on a single machine using their sequential code. Students, hence, get their first taste of a big data analytics problem and then proceed to employ a MapReduce job to carry out their analysis. The scenario employed for Project 1 was to find trending topics given the Wikipedia page views information[10]. The small data set asked them to analyze a single hour's worth of page views, while the larger data set required finding overall trending topics based on the page views of an entire month.

Project 2 dealt with a real world scenario to evaluate scaling methods to meet a web service's performance constraints. Students learned to use the AWS SDK to manage and monitor EC2 instances, and programmatically scale them up and down using Amazon's AutoScale API and Elastic Load Balancer (ELB) to meet the demands of a varying web load generator.

In Project 3, students worked on and evaluated the capabilities and limitations of various cloud storage technologies. Students compared flat files and databases for storing structured information. They also evaluated different storage back-ends on Amazon to compare their performance for an OLTP benchmark[4]. Students then implemented a horizontally scaling database using the Elastic Load Balancer and Amazon AutoScaling. Students worked with Amazon DynamoDB (a NoSQL database) to create an image archive[23] that was reachable from a web interface. Finally, students employed the Yahoo! Cloud Serving Benchmark (YCSB)[18] to compare and contrast two NoSQL systems, DynamoDB[5] and HBase[27].

In the fourth project, students worked on developing an application using the MapReduce programming model. As a final task, students developed an input text predictor, similar to Google Autocomplete, by using a subset of the Gutenberg[28] or Wikipedia [25] text corpora. Students performed a number of NLP tasks, using MapReduce, including generating a list of n-grams from the text corpus and creating a statistical language model. Finally students created a database to access the language model in real-time from a web interface.

In the team-based project for graduate students, the theme was to build an end-to-end cloud service to meet service specifications, performance and cost constraints. The input data set is a large dump of tweets collected using the Twitter streaming API over a period of several months. Student need to build a web service that can respond to specific queries of varying complexity. Students must then choose an appropriate scalable database backend, perform Extract Transform Load (ETL) to ingest the data and design and connect a scalable front end to service the web queries. An automated load generation system was developed and employed to test the students' web services. Students' systems were evaluated based on correctness, performance and the cost, per-hour, to run their web service on the public cloud. In order to encourage competition between teams, a live web-based scoreboard ranks teams on performance and cost so teams can improve their cloud service.

### *6.1.3 Cheat Checking*

All projects were performed on AWS's infrastructure, with each student having an individual account on AWS. All the student accounts were linked to a master paying account owned by the university. This enables us to collect a log of student provisioning, activity and expenditure. The project descriptions and instructions were all encoded on OLI, and students had to complete a project checkpoint activity after they have completed each task within a project. The checkpoint activity asks for a random item or value that can only be obtained after the project is completed successfully. This enables different students to answer questions for different parts of the data set and the system can automatically grade students on their project work. As mentioned above, some projects required an entire testing system to be developed. Since students utilized system images designed by the course staff, all work on these instances was logged to a server for verification. Furthermore, student code submissions were compared to current and past submissions using MOSS[6] which generated a similarity report.

To identify potential cheating cases, we developed an analytics system that ingests data from the multiple end points described above. Our system applies certain pre-determined rules to red-flag students who may have cheated. Examples include, correctly answering the checkpoint quiz without provisioning the necessary resources on AWS. This system is extremely powerful at letting students know that we are collecting information and can, with a little bit of effort, build a lot of evidence to prove a cheating case. Interviews with red-flagged students shed a lot of light on how much collaboration took place and dissuades all students from committing violations of academic integrity.

# 7. OUTCOMES

## 7.1 Matriculation, Attrition and Performance

The course has had a very favorable response with students from diverse programs and majors. So far 700 students have enrolled in the course over the last 2 years. About 10% are undergraduate, 1% are PhDs, and the rest are Masters students. Our retention rate hovers between 82% and 83%, which is similar compared to traditional courses at our department (80-85%). Successful completion rates are in the high ninety percent. The student grades follow a similar distribution to other courses offered at our institution.

## 7.2 Costs

Since we have adopted a public cloud infrastructure for our projects, evaluating cost is an important long-term factor for sustainability and scale. The course was supported by an AWS educational grant [4]. In the first few semesters, our university granted additional funds towards AWS costs for the course. In our first few offerings, we struggled with managing runaway costs by negligent students. During the first three offerings, the course cost approximately $300, on average, per student.

However, with the deployment of near-real-time logging, monitoring and alarm systems, we have significantly reduced abuse. Furthermore, the cost of AWS resources has significantly decreased in the last 12-months. Students who spend more than the pre-set project budget receive an automatic warning email indicating their infraction. Repeat offenders are liable to receive grade penalties. As a result, we have been able to reduce costs down to approximately $150 per student. Another aspect of cost is Teaching Assistant (TA) support. The TAs provide logistical and technical support via Piazza, office hours and the weekly recitation. They are also involved in the developing and testing our projects, load generator, auto-grading and cheat-checking systems. At our institution, every twenty five students allow the hiring of a single TA who can spend up to 20 hours a week supporting the course.

## 7.3 Student Feedback

At the end of each course offering, we administer an anonymous online survey, which 90% of the students respond to. Overall student feedback on the course has been steadily positive with most respondents saying that they would recommend this course to their friends. This has been reflected in doubling enrollment with each year. Overall, students were very enthusiastic about learning new programming models such as MapReduce and have asked for future versions of the course to spend more time on such topics. Students found the Unit dealing with programming models to be the most useful in the course, followed by cloud storage. Initially, students did not find the Unit on data centers particularly useful. Since then we have revamped the material and its relevance to the course which has improved their views of this unit. The graduate survey responders found Project 5 to be very useful and believed that they learned the most from it. However, these students complained that the project demanded a lot of their time given the open-ended scenario and rigorous evaluation. Students indicated that they enjoyed combining several technologies and services together to create a full-fledged application.

## 7.4 Lessons Learned

The course development required a large time investment to design and develop content, videos, activities, auto-graded quizzes and partially auto-graded projects. With platforms such as OLI, this has to be completed well in advance before the semester starts. Significant testing is required to remove bugs in activities, quizzes and projects before they are presented to students. Good and timely feedback is critical in enabling student learning. Instructors should maximize the capabilities of their chosen platforms to provide a variety of activities with useful feedback. Our content benefits from properly structured and well-thought activities, quizzes and projects that are aligned with the course objectives and learning outcomes. This enables the systematic evaluation and assessment of the course to enable learning. Such efforts are always a work in progress and require ongoing updates.

From observing the OLI dashboard and AWS activity, it is clear that students are deadline driven and typically do not consume content or work on projects until close to the deadlines. It is best for students to have weekly deliverables on content and projects. We are experimenting with the idea of expiring tokens to encourage students to test their solutions against our load generators prior to the deadline.

Instructors planning to offer similar courses with projects on AWS must pay careful attention to cost. Projects that require expensive resources such as larger instance types or faster storage can drive up costs very quickly. Daily monitoring of student activity and resource usage is important to keep the course within budget. Carelessness on the part of students who forget to terminate resources after they are done can be especially precarious to budgets if left unchecked.

# 8. SIMILAR COURSES

Border[13] presented his experience in teaching cloud computing at Rochester. He stressed on the necessity to teach the cloud's enabling technologies in order to allow students to effectively apply the cloud computing concepts. Indeed, many universities have already started featuring undergraduate and/or graduate courses on cloud computing. For instance, University of Maryland's course on cloud computing[11] is generally regarded as the first in the field, wherein it focuses mainly on MapReduce [24]. University of California at Berkeley has also a graduate-level seminar course on cloud computing[20] as well as an undergraduate-level one, which focuses on programming the cloud[15]. Cornell teaches two classes on the cloud, one that stresses on resource sharing[17] and another that mainly concentrates on cloud storage[22].

To the contrary of developing a new standalone course dedicated to cloud computing, Radenski[33] at Chapman University suggested integrating cloud computing in a High-Performance Computing course. The course focuses on data-intensive processing with MapReduce as a main, available and easy-to-use parallel programming model for cloud computing. Holden[29] proposed using cloud computing to teach a databases course at Rochester. In particular, the course uses AWS to provide a secure, flexible and adaptable lab environment that is accessible off-campus and meets the course's predefined curricular needs. Likewise, Salah[35] at Khalifa University of Science, Technology and Research, UAE promoted utilizing cloud-based labs in a cybersecurity course. The course aims at leveraging the cloud to provide hands-on experience and training to students as well as remedy the cost, operation, scheduling, and management problems typically associated with classical cybersecurity labs. Lastly, Wang[31] at Michigan Technological University incorporated cloud computing into a computer systems and network administration course through a single high-level class project. Specifically, the project entails building and testing two clouds using Eucalyptus[32] and Xen[12].

To this end, some cloud courses are also being taught online, and seem to be catering primarily to industry professionals. Notable among these are Berkeley's Software-as-a-Service course (CS169.2X) on edX[19], and Stanford's cloud computing course (CS309A)[16]. We find that most of these online courses focus either too much on a few technologies or offer a very broad overview from an information system's perspective.

As compared to all these cloud-enriched or pure-cloud classical and online courses, our course is uniquely positioned to provide foundational knowledge in systems with practical hands-on experience with designing, building and evaluating real-world solutions using real-world scenarios and data sets. Students develop projects while optimizing for performance and running cost under a strict development budget. This course bridges the gap between academia and industry with respect to preparing the next generation of cloud computing architects, engineers, developers and researchers.

## 9. FUTURE OFFERINGS

The course is offered every semester for students at our institution. We plan to continue to incorporate many enhancements based on feedback collected and lessons learned. We plan to explore a more diverse set of topics such as streaming programming models, cloud-enabled mobility, security and software defined networks and storage[26,38]. Furthermore, we are exploring the possibility of incorporating our projects on multiple public cloud infrastructures such as Microsoft and Google public clouds. It is our intent that students will get to choose the cloud platform for each project as part of their solution design.

## 10. REFERENCES

[1] Amazon Elastic Compute Cloud http://aws.amazon.com/ec2/

[2] Amazon Elastic MapReduce http://aws.amazon.com/elasticmapreduce/

[3] Amazon S3 http://aws.amazon.com/s3/

[4] AWS in Education http://aws.amazon.com/education/

[5] DynamoDB http://aws.amazon.com/dynamodb/

[6] MOSS: A System for Detecting Software Plagiarism http://theory.stanford.edu/~aiken/moss/

[7] Online Learning Initiative http://oli.cmu.edu

[8] Piazza Technologies http://www.piazza.com

[9] VMWare http://www.vmware.com

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

[11] Web-Scale Information Processing Applications http://www.umiacs.umd.edu/~jimmylin/cloud-2008-Spring/index.html

[12] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A., 2003. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review 37*, 5

[13] Border, C.B., 2013. Cloud computing in the curriculum: fundamental and enabling technologies. In *Proceeding of the 44th ACM technical symposium on Computer science education* ACM, 147-152.

[14] Borthakur, D., 2008. HDFS architecture guide. *Hadoop Apache Project. http://hadoop.apache.org/common /docs/current/hdfs_design.pdf*

[15] CS 194-17: Programming the cloud http://db.cs.berkeley.edu/w/cs-194-17-programming-the-cloud/

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

[17] CS5412: Cloud Computing http://www.cs.cornell.edu/courses/cs5412/2012sp/

[18] Cooper, B.F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R., 2010. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing* ACM, 143-154.

[19] CS169.2x, Software as a Service https://www.edx.org /course/uc-berkeley/cs-169-2x/software-service/1005

[20] CS 294: Cloud computing: systems, networking, and frameworks http://www.cs.berkeley.edu/~istoica/classes/cs294/11/

[21] Cushing, A. and Gantz, J.F., *Climate Change: Cloud's Impact on IT Organizations and Staffing, IDC Whitepaper.* IDC.

[22] CS5320: Big Data http://www.cs.cornell.edu/Courses/cs4320/2012fa/

[23] Caltech-256 Dataset http://www.vision.caltech.edu/Image_Datasets/Caltech256/

[24] Dean, J. and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. *Communications of the ACM 51*, 1

[25] Wikimedia Downloads http://dumps.wikimedia.org/

[26] Foundation, O.N., 2012. Software-defined networking: The new norm for networks. *ONF White Paper*

[27] George, L., 2011. *HBase: the definitive guide*. O'Reilly Media, Inc.

[28] Project Gutenberg http://www.gutenberg.org/

[29] Holden, E.P., Kang, J.W., Bills, D.P., and Ilyassov, M., Databases in the cloud: a work in progress. In *Proceedings of the Proceedings of the 10th ACM conference on SIG-information technology education* (Fairfax, Virginia, USA2009)

[30] Lakshman, A. and Malik, P., 2010. Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review 44*, 2

[31] Leong, L., Toombs, D., Gill, B., Petri, G., and Haynes, T., 2014. *Magic Quadrant for Cloud Infrastructure as a Service.* Gartner. ID: G00261698

[32] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D., 2009. The Eucalyptus Open-Source Cloud-Computing System. In *Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on*, 124-131.

[33] Radenski, A., 2012. Integrating data-intensive cloud computing with multicores and clusters in an HPC course. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* ACM, 69-74.

[34] Ross, R.B. and Thakur, R., 2000. PVFS: A parallel file system for Linux clusters. In *in Proceedings of the 4th Annual Linux Showcase and Conference*, 391-430.

[35] Salah, K., Harnessing the cloud for teaching cybersecurity. In *Proceedings of the Proceedings of the 45th ACM technical symposium on Computer science education* (Atlanta, Georgia, USA2014)

[36] Bureau of Labor Statistics, Employment Situation Summary http://www.bls.gov/news.release/empsit.nr0.htm

[37] Stonebraker, M., 2010. SQL databases v. NoSQL databases. *Communications of the ACM 53*, 4

[38] Thereska, E., Ballani, H., O'shea, G., Karagiannis, T., Rowstron, A., Talpey, T., Black, R., and Zhu, T., IOFlow: a software-defined storage architecture. In *Proceedings of the Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles* (Farminton, Pennsylvania2013)

[39] Thille, C., 2008. Building open learning as a community-based research activity. *Iioshi and Kumar (Eds.). Opening up Education: The Collective Advancement of Education through Open Technology, Open Content, and Open Knowledge*

[40] White, T., 2012. *Hadoop: the definitive guide*. O'Reilly.