A Comprehensive Course on Big Data for Undergraduate Students

Jawwad A. Shamsi, Syed Zain ul Hassan, Narmeen Bawany, Nausheen Shoaib
Systems Research Laboratory, Computer Science
National University of Computer and Emerging Sciences
Karachi, Pakistan
jawwad.shamsi@nu.edu.pk, zain.hassan@nu.edu.pk, narmeen.bawany@nu.edu.pk, nausheen.shoaib@nu.edu.pk

***Abstract*—**
Continuous growth and wide-scale popularity of big data systems have highlighted the need of effective incorporation of big data in Computer Science at the undergraduate level. This need has also been inspired by the increasing demand from the industry. This paper is aimed to address this need by proposing a comprehensive elective course on big data systems. Our proposed course has strong emphasis on developing both theoretical as well as practical skills. The course has wide-scale coverage of significant topics related to big data systems. These include platform-related topics such as Hadoop and Spark, batch and stream processing, machine learning and iterative systems. Supported through weekly labs and course projects, the course also encouraged students to apply concepts in solving real-world problems. This paper describes design and structure of the course and explain course contents and students' evaluations. The paper is useful for the community members who would like to strengthen big data curriculum at the undergraduate level.

***Index Terms*—Big Data, Hadoop, MapReduce, Spark, Hive, NoSQL, Streaming, Data Analytics, Cloud, Containers.**

I. Introduction

Teaching Big Data [1] to undergraduate students requires important considerations. The topic has emerged as one of the leading and popular topics of the Parallel and Distributed Curriculum (PDC). Big data curriculum should include details of cloud computing platforms which can meet the growing demands of scalability, analytics, batch processing, and streaming. Further, the curriculum should also incorporate knowledge of parallel and distributed systems which can host big data systems [2]. In addition, hands-on skills on programming and analytics are needed to be imparted, comprehensively. Another important consideration is to link the curriculum with applied topics in order to encourage students to solve real-world problems.

Existing efforts on big data curriculum spans around a wide variety of topics including cloud computing, MapReduce, Mahout, Physical [3] and Virtual Clusters [4], and Machine Learning. Many universities around the world offers these topics under the banner of big data curriculum. Students usually have options to take one or more courses for these topics. However, big data curriculum do not concentrate much on spark - an emerging platform for big data analytics and machine learning.

While existing courses on big data are well established, there is a need for a comprehensive course on big data which could include essential topics in a single course. Specifically, we believe that a course which could cover both MapReduce (batch processing) and Spark (Machine Learning and Streaming) in a single course is essential. This is because the topic of big data has now evolved from the core application of batch processing to the applications of streaming, iterative processing, and machine learning.

Another important consideration is that students should be given exposure to both Virtual Machines (VMs) and containers - the two widely used platforms for hosting big data systems. Further, hands-on skills are needed to be developed in order to elucidate the task of big data processing and analytics. The students should also be able to apply concepts and skills to solve real-world applications.

This paper is motivated to address above mentioned needs. Considering these requirements, we present structure, design, and implementation of CS479- Introduction to Cloud Computing course, which has recently been offered at National University of Computer and Emerging Sciences, Karachi. Specifically, our proposed course has following major features:

- It has strong emphasis on strengthening theoretical as well as hands-on skills.
- It has extensive coverage on topics for batch processing (Hadoop) as well as streaming (Spark).
- It covers Virtual Machines (VMs) as well as containers, two widely used cloud computing solutions for hosting big data systems
- It encourages and train students to apply the learned techniques in solving real world problems.

In this paper, we present design and details of the course and share results of students' evaluations. We anticipate that the proposed course would be useful for educators who aim to strengthen the curriculum of big data.

The rest of the paper is organized as follows. Section II describes significant work by other educators and researchers

This work has been supported by IEEE TCPP Early Adapter Awards Fall 2012, Fall 2013, and Fall 2015.

in this field. Section III explains the course contents, whereas section IV explains results of students' assessments. Section V concludes the paper with some suggestions.

II. Related Work

Big Data Curriculum has been widely adopted at various levels of education. In this section, we present an overview of these efforts. Our aim is to highlight core points which have been adopted to incorporate Big Data Curriculum. For clarity, the section has been divided into three sub-sections.

A. Big Data Curriculum

Phillip et al. [5] has incorporated Big Data contents in middle school curriculum. The authors have highlighted significance of having proper alignment with the existing curriculum. Grillenberger [6] has focused on teaching big data at the secondary school level. The focus has been on the topics of data management. Martnez-Arocho et al. [7] are motivated to introduce computer science in the high school curriculum. The authors propose a game-based approach for teaching Big data in the middle school curriculum. Similarly, Prayaga and Devulapalli [8] has made efforts to teach data analytics using Hadoop to high school students. The authors adopt a simple approach to teach simple MapReduce problems to the students. Contrary to these efforts, our focus has been on undergraduate studies.

Zhao et al. [9] have mentioned that undergraduate curriculum lacks hands-on exercises in big data. The authors proposed four labs for big data teaching. These include Hadoop Configuration, Hbase querying, wordcount on Hadoop, and analyzing network logs of a website. In our paper, we are inspired by their work; however, our focus has been on teaching big data hands-on skills in greater depth. These include Spark, Streaming, Hadoop joins, and accessing cloud etc.

Faraz et al. [10] has incorporated big data problems as motivational tools for explaining computer science curriculum. The authors mentioned that encouraging students to formulate real-life parallel problems and big data examples have been useful and effective. Our work is similar in a manner that we are also motivated to enhance real-life problem solving skills of students.

Efforts have been made to integrate different components of Big Data curriculum [11]. Silva et al. have incorporated contents of MapReduce, NoSQL databases (such as Hbase), and NewSQL in the curriculum. Our work is supplemental to these efforts as we also concentrate on extensive coverage of these topics. However, in addition to these topics, we have extensive coverage of topics related to big data streaming, big data analytics using machine learning, and cloud platforms such as VMs and containers.

Nicholas et al. [12] have highlighted the significance incorporating statistical tools in the curriculum of data science. The authors mentioned that such tools are significant in reproducible analysis, data visualization, and management. In our work, while we do not focus on enhancing skills related to statistical analysis; however, we believe that our proposed course can be useful in strengthening foundation for data science programs.

Anderson et al. [13] have elaborated on a seventy seven credit hour four-year degree program in data science. The program is concentrated on many components of data science including predictive analytics, machine learning, and data mining. Tang and Sae-Lim [14] analyzed Data Science (DS) programs from eight disciplines. The analysis was made with respect to program description, curriculum requirements, and program contents. Their analysis showed that there is a significant difference between the degree programs offered across the US. Further, the authors observed that some DS programs have limited coverage of upper-level analytical skills. Our work can be considered as supplemental in strengthening analytical skills.

Jacobi et al. [15] highlighted the need for interdisciplinary education. The authors argued that students from interdisciplinary programs have the rights skills to meet the needs of the employers.

Johnson [16] mentioned that there is enough demand for data scientists and graduates from the related fields. The author argues that despite the growing number of graduates with these skills, there are many vacancies in the market. The author describe hands-on exercises using the CoNVO framework for an undergraduate course in the MIS program. Our work is supplemental to these efforts as we are motivated to meet the growing demands of the industry by developing students' hands-on skills.

Rachel et al. [17]. highlighted the significance of big data analytics in health professional education. The authors exemplify that health education can benefit if big data analytics is incorporated in the curriculum. In our work, we are also motivated by the significance of teaching big data to undergraduate students. In this context, we propose, a comprehensive course on big data.

Eckroth [18] has described a comprehensive course on Big Data Analytics. In comparison, our proposed course is more comprehensive, as it includes several other topics such as streaming, MongoDB and Hive as NoSQL databases, and containers and virtual machines as platforms for hosting big data.

B. Cloud computing

Many educators have used cloud computing to teach contents of big data. These efforts are mainly focused on imparting knowledge related to platforms such as Hadoop/MapReduce and Cloud for big data storage and processing.

The University of Harvard [19] has offered this course with a wide range of topics including Virtual Machines, AWS, Azure, Restful API, load balancing, MapReduce, and Hadoop. At NUCES, we offered several related topics in cloud computing; however, our focus has also been to cover some emerging topics such as data streaming and data analytics. For this purpose, we have sufficient focus on hands-on skills using Spark.

Martinez et al. [20] surveyed use of cloud computing for different educational institutes. The authors identified and analyzed benefits of cloud computing for education industry. Nabil Sultan [21] has highlighted the need of using cloud computing for education. Calyam et al. [22] have developed three laboratory exercises for an undergraduate course in cloud computing. The exercises are developed on GENI - a popular architecture for distributed computing. Lab contents include account setup on GENI, measurement of slice characteristics on GENI, and QoS control and load balancing using open flow. Our work is similar in nature as we also provide details of lab exercises; however, our focus has been on different components of big data curriculum such as cloud storage, access, and processing.

Changchit [23] has mentioned that cloud computing can be used as a cost effective and popular mechanism to teach technological advancements to students. Our work is similar in that we utilize cloud computing to teach big data systems.

C. Parallel and Distributed Computing Curriculum

Efforts have been made in strengthening PDC topics through various stages of the undergraduate curriculum. These include core as well as elective courses [24].

Kumar mentioned the importance of PDC topics and teaching them in parallel with other courses [25]. Similarly, Bogaerts have made efforts to introduce PDC topics through introductory programming courses [26]. In order to introduce PDC topics early in the CS undergraduate curriculum, Newhall et al. [27] have introduced a new core course, which is mandatory for all the students who intend to graduate with either a major or minor in CS. While we realize the significance of teaching PDC topics in the curriculum, through our proposed course students can benefit if parallelism is introduced early in the curriculum. At NUCES, there are several PDC topics such as threads, multiple processes, replication synchronization, and distributed memory, which are introduced early in the curriculum. Our proposed course utilize these concepts to enhance students' learning.

Suzanne [28] have utilized Phoenix++ MapReduce to teach PDC concepts. The work has been motivated to address the requirement of working on a hadoop cluster. Our work is similar in context that we eliminate the requirement of working on a hadoop cluster and utilize cloudera containers to teach Spark and Hadoop.

Sakellariou [29] introduced a second-year course to teach PDC topics. The course has been offered at the University of Manchester and is focused on four key pillars of distributed systems. These include trade-offs, failures, concurrency and synchronization, and performance.

Adams introduced parallel design patterns in the CS curriculum [30]. The work has been motivated by the need that students should be taught to think in parallel patterns which would enhance learning.

Our efforts are supplemental to these efforts as we are also concentrated on strengthening PDC topics. However, the main focus of this paper is to propose a comprehensive course on teaching big data. These contents are covered through an elective course.

III. Course Design and Course Contents

The course in consideration is Introduction to Cloud Computing (CS-479). It was offered in the Fall 2017 semester to the students of seventh semester at the National University of Computer and Emerging Sciences (NUCES), Karachi. There were 45 students registered in the course. The course is focused on big data curriculum. Semester at NUCES consists of 16 weeks of teaching, with mid exams during 6th and twelfth week. Final exam is held after the conclusion of classes, during the 17th week. It is a 3-credit hour course with two hours of theory and one-hour of lab. The course has pre-requisite of Operating Systems and Computer Networks. The students have also covered fundamental topics of PDC.

A. Pedagogical Goals

The course has four major goals. These are motivated by the need and challenges mentioned earlier.

Goal 1: The first major goal of this course is to ensure that fundamental concepts related to big data are taught. Big data systems have evolved from transactional and batch processing systems to analytical and stream processing. Further, topics such as CAP theorem, data locality, and SIMD are also needed to be covered.

Goal 2: The second goal of this course is to have an extensive coverage of big data systems. The topic of big data has expanded from Hadoop/MapReduce to Spark. Further, use of container and VM-based cloud has provided necessary features such as scalability, fault tolerance, and replication.

Goal 3: The third major goal is to develop hands-on skills. Strong hands-on skills are important to develop students' abilities for big data processing including batch and stream processing.

Goal 4: The final goal of the course is to encourage students to apply the skills in solving real-problems of big data.

Table I summarizes the four goals described here. The table also describes how contents related to each goal are covered. Goal 4 was achieved through course projects and class lectures, whereas all the other goals were accomplished through lectures and labs.

B. Course Contents

Table II shows weekly plan of the course along with their alignment with goals. The contents of the course were designed to cover all the necessary topics including big data programming, platforms, and iterative and batch processing. We also have fundamental coverage on topics of big data security and privacy and fog computing (delay-sensitive big data applications). Both these topics highlight the emerging areas for big data systems.

The course also had sufficient coverage of recommended PDC topics. These are listed in table III. The table also lists bloom's taxonomy. This mapping is shown according to the revised version of the blooms taxonomy [31].

TABLE I
PEDAGOGICAL GOALS

Goals	Description
G1: Impart fundamental concepts about Big Data	Fundamentals of Big Data Systems, CAP Theorem, SIMD, Data Locality,
G2: Teach Big Data Platforms	Big data platforms, Thorough understanding of programming platforms MapReduce and Spark, Hadoop Architecture, Hive, HDFS systems, cloud-based platforms, Virtual Machines, and Containers
G3: Develop hands-on skills experience	Hands-on experience using Spark, MapReduce, Hive, and Pig, Accessing AWS through libraries, etc.
G4: Apply concepts to solve real big data problems	Applying concepts to solve real problems such as cluster management, Cloud Management, Opinion Mining, Network Security etc.

TABLE II
WEEKLY PLAN

Week	Topics Covered (W)	Alignment with Goals
1	Introduction to Big Data, Fundamental Concepts about Parallel Computing, Introduction to HPC and Cloud Computing, Flynn's taxonomy	G1
2	Virtual Machines and Containers, Centralized and Distributed File Systems, Data Center Architecture	G1, G2
3	Big Data Fundamentals, CAP Theorem, Transactional vs Analytical Systems, ACID vs BASE, Introduction to NOSQL Systems	G1, G2, G3
4	Introduction to Hadoop, HDFS, MapReduce Platform, Word Count using MapReduce	G2, G3
5	MapReduce Programming, Combiner Functions	G2, G3
6	**Mid 1 - Examination**	
7	Join Operations using MapReduce, Pig, Hive, and Hbase	G2, G3
8	MongoDB, CAP Theorem and MongoDB, Network Issues in Big Data, TCP Incast Problem	G2, G3
9	Scalability Problems in Hadoop, YARN, Fog Computing and Time-sensitive applications	G2,G3
10	Introduction to Spark, Spark Programming, Project Proposals	G2, G3, G4
11	Spark Programming: Data Partitioning, Hash Partitioning, Join Operations	G2, G3
12	**Mid2- Examination**	
13	Spark Mlib- Machine Learning using Spark	G2, G3
14	Kafka and Spark Streaming, Limitations of Batch Processing	G1, G2, G3
15	Scalable Systems for Big Data: Open Stack. Privacy and Security in Big Data Systems, Side Channel Attacks	G2, G3
16	**Projects Review and Presentations**	G4
17	**Final Examination**	

TABLE III
COVERAGE OF PDC TOPICS

Topic	Bloom Level	How Covered
Architecture Topics		
Flynn's Taxonomy	U/C[2]	Conceptual Explanation
SIMD	U/F	Conceptual Explanation
Distributed Memory	U/P	Conceptual Explanation
Programming		
SIMD and SPMD	A/P	MapReduce and Spark Programming
Data Locality	A/F	MapReduce and Spark Programming
Speedup	A/P	MapReduce and Spark Programming
Algorithms		
Time	E/F	Data and Task Division
Speedup	E/F	Data and Task Division
Reduction	A/P	How Reduction works in MapReduce
Cutting Edge Topics		
Cluster Computing	A/F	Creating Clusters on AWS, Distributed File system and Network File System
Cloud Computing	A/F	AWS and VM-based Cloud, Open Stack
Web search	E/P	Partition Aggregate Model adopted by Big Data Systems
Consistency	A/F	CAP Theorem and How Big Data Systems benefit from it.
Locality	A/F	Data Locality in Hadoop
Fault Tolerance	A/C	Fault Tolerance concepts in Spark and Hadoop

[1] The cognitive process dimension
[2] The knowledge dimension
U=Understand, A= Analyze, E= Evaluate, C=Conceptual, F=Factual, P=Procedural

These topics are divided according to four mappings recommended in the PDC curriculum, i.e. Architecture, Programming, Algorithms, and Crosscutting Topics. The table reflects the mapping in three dimensions. For the taxonomy, the

TABLE IV
LAB ACTIVITIES

Lab No	Details	Learning Goals/Outcomes
Containers and AWS		
1	Amazon AWS account setup and cluster creation	To get the fundamental understanding of cloud
2	Creation of Docker application	To understand the functionality of container based cloud
3	Using Boto3 SDK to manage AWS EC2 instances and S3 objects	To learn accessing cloud using API
Hadoop MapReduce		
4	HDFS file copy example	To understand functionality of HDFS
5	Wordcount: Count no of words in multiple files using MapReduce	To comprehend MapReduce programming
6	Computing inlinks of websites using MapReduce	To learn use of MapReduce in Search Engines
7	Join Operation Using MapReduce	To evaluate use of MapReduce as a database system
NoSQL		
8	Data query using Apache Hive and Hbase	To learn hands-on skills on key,value NoSQL systems
9	Performing MongoDB shell operations	To learn hands-on skills on document-oriented NoSQL systems
Apache Spark		
10	Introduction to Spark programming. Word Count Problem and Compute Pi with MonteCarlo Method	To learn spark programming
11	Frequent item set mining using Spark	To comprehend data analytics using Spark
12	Mlib examples with Spark	To understand machine learning and data analytics using Spark
13	Spark streaming and Kafka	To learn stream processing on big data

‘

TABLE V
PROJECTS

S. No	Project Title
Big Data Analytics	
1	Performance Analysis on EspnCricinfo data
2	YouTube Data Analysis Using Hadoop
3	Analysis of emails in Amazon Common Crawl Dataset using MapReduce
4	Analysis of Twitter data for police-involved fatalities
Network Security	
5	Log analyzer for Data Center Equipment (Firewall, Servers, Application)
6	DDoS Attack Detection with Hadoop Initial Draft
Container Management	
7	Load Balancing on Swarm Cluster
8	Implementation of Kubernetes
Information Retrieval and Text Mining	
9	Sentiment Analysis on Tweets with Apache Pig
10	Analyzing Wikipedia content via Streaming
Bio Informatics	
11	Parallel Suffix Tree Construction For Genome Sequence Using Hadoop
IaaS/SaaS	
12	CCTV Footage Analysis
13	Building infrastructure of e-commerce store for handling unusual traffic
14	Recommendation System for Customers
15	Dockerized elastic price comparison application

numerator refers to the cognitive process dimension, whereas the denominator denotes the dimension of knowledge. For instance, U/C maps to the topics which were discussed in the class and concepts were explained. Similarly, U/F refers to the topics which were discussed in more detail along with some practical examples. A/F indicates deeper coverage with analysis and factual coverage. These are the topics which were used for laboratory assignments. A/P is used for topics with analysis and procedural coverage. These are covered with lab exercises along with more emphasis on real problems.

Machine Learning and Data Streaming We have strong emphasis on several important components of big data. These include relational operations, machine learning, and streaming.

1) Relational Operations: For relational operations such as joins, MapReduce and Spark programming codes were discussed. For MapReduce, both mapper and reducer side join techniques were covered. We dedicate a specific lab for MapReduce based join Operations. For spark, join operations were easier because of built-in functions.
2) Machine Learning: Various machine learning tasks require iterative computation. Therefore, running machine learning tasks on MapReduce leads to higher cost. We explained the reasons of this inefficiency to students. We also discussed how Spark can provide improved performance for iterative computation. In addition, we explained the usage of Spark to students using python code. Machine learning algorithms such as clustering, classification, regression, were explained to students. We

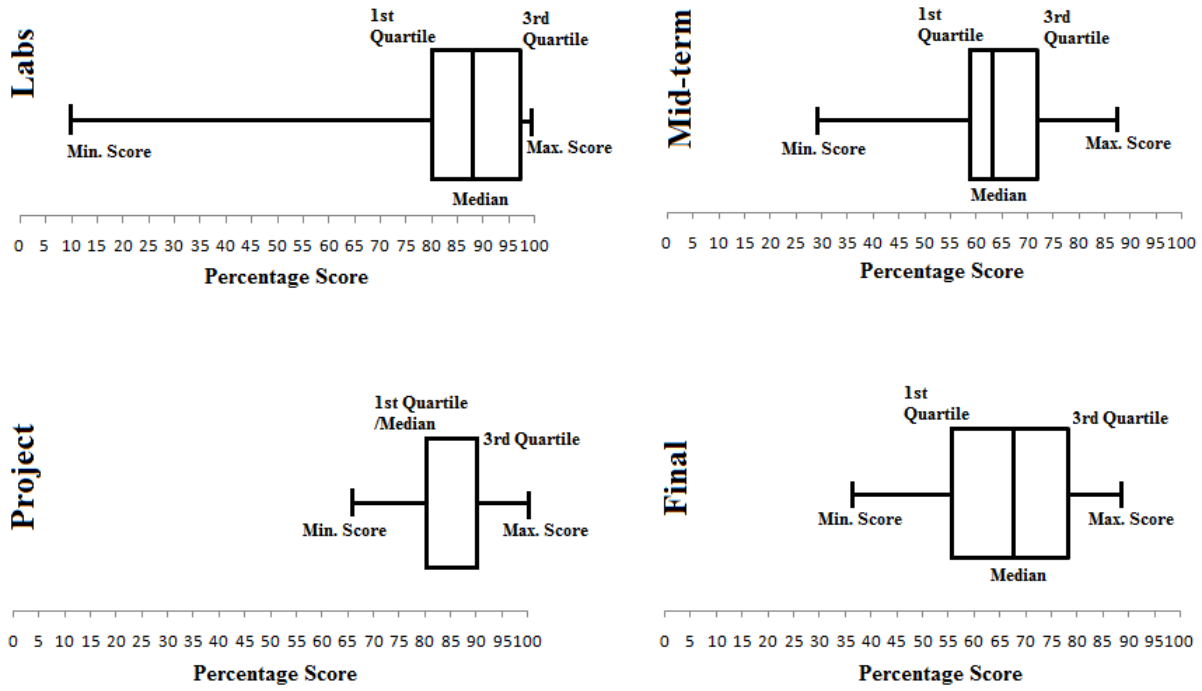

Fig. 1. Students Overall Assessments - Whisker Plots

also explained topics related t feature extraction and TF-IDF model. For all these algorithms, we discussed python code. Students also performed two lab tasks related to frequent item set mining and spam classification. In addition, we explained YARN, which promotes scalability for spark.

3) Streaming: Processing streaming data is important for Big data systems. We explained limitations of Mapreduce in processing streaming data and discussed how Spark solves this problem. We also discussed various architectural modifications and platforms such as Kafka and fog computing. Students performed a lab activity in which they setup Kafka topics and execute spark code to perform stream processing.

C. Lab Activities

Table IV shows weekly lab activities. These are aligned with the theoretical topics covered in table II. Table IV also lists learning outcomes of these lab exercises. Lab activities are categorized into four broad categories, i.e., Cloud Platform, Hadoop/MapReduce, NoSQL, and Apache Spark. There were 13 labs in total, as there were no lab activities during the two weeks of mid exams. Further, the last week was dedicated to project presentations. For labs, we utilized container-based cloudera distribution of Hadoop and Spark. Each node served as a single node cluster. Each lab exercise was designed for 1-hour of lab activity. However, students had the option to continue working beyond lab hours and submit the lab activity for evaluation with in a week.

D. Student Projects

Projects were integral part of the course. Students completed projects in a group of two to three students. Project proposals were submitted during the tenth week and final demonstrations and presentations were conducted during the sixteenth week. Students selected projects based on their interests. Projects were aimed to enhance students' abilities in solving real-world problems of big data (Goal 4). Table V shows the list of projects along with their application areas. A large number of students selected projects related to data analytics. In addition, projects related to network security and cloud management were also selected.

A large number of projects were focused on data analytics. Students applied various data analytical techniques, they learned during the course, on different datasets to solve real world problems. For cloud-related projects, students either did configurations or developed a cloud applications.

IV. Student Evaluations

Evaluations were important in assessing performance of students. Further, they are helpful in identifying weak students and determining areas which are needed to be strengthened.

Students were evaluated throughout the semester. Evaluations were based on labs, mid and final exams, and projects. For each lab activity, students were given a few tasks. Students were required to complete the tasks and submit them for grading. Similarly, for projects students were required to either solve a problem or implement a solution for a known problem. They were required to show the results with increasing dataset.

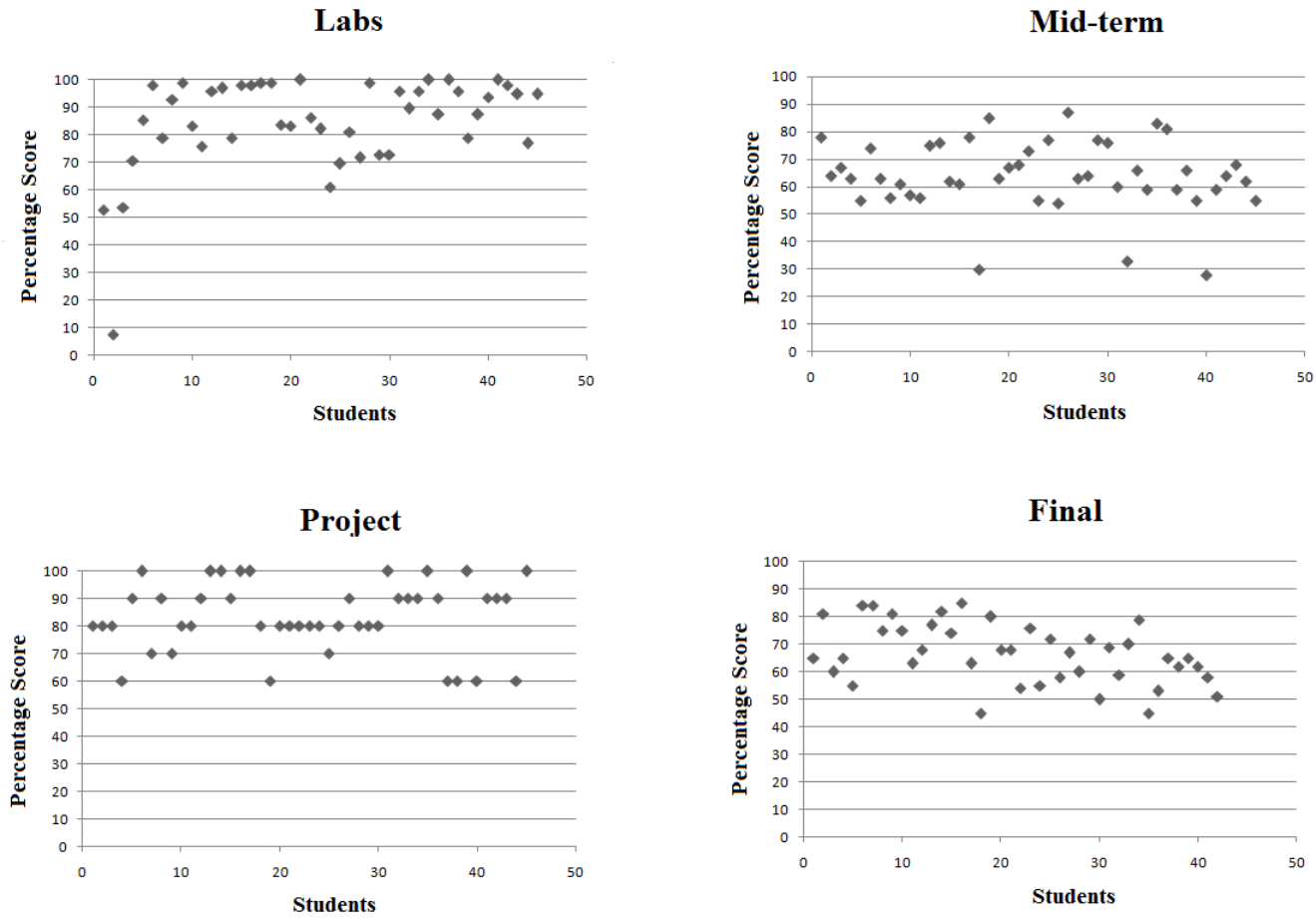

Fig. 2. Students Overall Assessments - Scatter Plots

For exams, students were given a few questions along three categories of programming, analytics, and theory.

For mid and final exams, we noted students performance in the three categories of questions. We noted that students' performance in theoretical and programming questions was almost similar. Further, for programming questions, minimum score was lower, because few students could not perform well. Overall, for analytical questions, students performance was lower then their performance in theoretical and programming questions.

We also noted that for analytical questions, students' performance was more scattered. However, for the remaining two categories, the performance was similar.

We also computed whisker plots for students performance in labs, mid-term exams, projects, and finals. Figure 1 shows the whisker plots for students' performance in these components. The smallest variation was observed for projects. This is because of the fact that group-based learning helped students in achieving their targets. The largest variation was observed for labs. This is most likely due to non-submission of labs by a few students. However, for labs and projects, median and maximum scores were quite high. In comparison, students' performance was lower in mid and final exams. This is understandable because exams were comprehensive and time-constrained. Figure 2 shows the performance through scatter plots. The figure confirms the earlier observation that labs and projects were high scoring as compared to exams. In general, these results align with the overall results at the university.

V. Suggestions and Lessons Learned

This paper presents details of a comprehensive course on big data. The course includes batch as well as stream processing on big data. Further, it has sufficient coverage on MapReduce and Spark - the two widely used platforms for big data computing. Through weekly labs, the course developed students' hands-on skills and prepared them to solve big data problems. The course also has coverage on topics related to data analytics and machine learning. Further, cloud and cluster computing concepts are also covered using containers and VMs. The methodology and the design presented in this course has been very effective. Following are a few important suggestions from the course:

- Big data curriculum contains extensive topics. A wide-scale adaptation of big data topics at the undergraduate

level is essential in producing quality graduates which can meet the growing needs of the industry.

- Although the IEEE TCPP curriculum has been very effective and broad, it needs to be further updated to incorporate topics related to big data. For instance, topics related to batch and iterative processing in a distributed environment, distributed file system, and stream processing are needed to be incorporated.
- Projects are useful in developing students' skills to solve real-world problems.

ACKNOWLEDGMENT

This work has been supported by NSF TCPP Early Adapter Awards Fall 2012, Fall 2013, and Fall 2015. The work has also been supported through Higher Education Commission (HEC), Pakistan, NRPU grant 5946.

REFERENCES

[1] J. Shamsi, M. A. Khojaye, and M. A. Qasmi, "Data-intensive cloud computing: requirements, expectations, challenges, and solutions," *Journal of grid computing*, vol. 11, no. 2, pp. 281–310, 2013.

[2] J. A. Shamsi, N. M. Durrani, and N. Kafi, "Novelties in teaching high performance computing," in *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*, pp. 772–778, IEEE, 2015.

[3] J. Eickholt and S. Shrestha, "Teaching big data and cloud computing with a physical cluster," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pp. 177–181, ACM, 2017.

[4] J. Eckroth, "Teaching big data with a virtual cluster," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pp. 175–180, ACM, 2016.

[5] P. S. Buffum, A. G. Martinez-Arocho, M. H. Frankosky, F. J. Rodriguez, E. N. Wiebe, and K. E. Boyer, "Cs principles goes to middle school: learning how to teach big data," in *Proceedings of the 45th ACM technical symposium on Computer science education*, pp. 151–156, ACM, 2014.

[6] A. Grillenberger, "Big data and data management: a topic for secondary computing education," in *Proceedings of the tenth annual conference on International computing education research*, pp. 147–148, ACM, 2014.

[7] A. G. Martínez-Arocho, P. S. Buffum, and K. E. Boyer, "Developing a game-based learning curriculum for big data in middle school," in *Proceedings of the 45th ACM technical symposium on Computer science education*, pp. 712–712, ACM, 2014.

[8] L. Prayaga and K. Devulapalli, "Data analytics with hadoop for juniors,"

[9] T. Zhao, K. Qian, D. Lo, M. Guo, P. Bhattacharya, W. Chen, and Y. Qian, "Problem solving hands-on labware for teaching big data cybersecurity analysis," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2014.

[10] F. Hussain, N. Deo, and S. K. Jha, "Early adoption: High-performance computing for big data introducing parallel programming and big data in the core algorithms curriculum," in *Procceedings of the 4th NSF/TCPP Workshop on Parallel and Distributed Computing Education (EduPar 2014)*, 2014.

[11] Y. N. Silva, S. W. Dietrich, J. M. Reed, and L. M. Tsosie, "Integrating big data into the computing curricula," in *Proceedings of the 45th ACM technical symposium on Computer science education*, pp. 139–144, ACM, 2014.

[12] N. J. Horton, B. S. Baumer, and H. Wickham, "Teaching precursors to data science in introductory and second courses in statistics," *arXiv preprint arXiv:1401.3269*, 2014.

[13] P. Anderson, J. Bowring, R. McCauley, G. Pothering, and C. Starr, "An undergraduate degree in data science: curriculum and a decade of implementation experience," in *Proceedings of the 45th ACM technical symposium on Computer science education*, pp. 145–150, ACM, 2014.

[14] R. Tang and W. Sae-Lim, "Data science programs in us higher education: An exploratory content analysis of program description, curriculum structure, and course focus," *Education for Information*, vol. 32, no. 3, pp. 269–290, 2016.

[15] F. Jacobi, S. Jahn, R. Krawatzeck, B. Dinter, and A. Lorenz, "Towards a design model for interdisciplinary information systems curriculum development, as exemplified by big data analytics education," 2014.

[16] S. Jensen, "Integrating big data services into an undergraduate mis curriculum," *International Journal of Systems and Service-Oriented Engineering (IJSSOE)*, vol. 7, no. 2, pp. 58–73, 2017.

[17] R. H. Ellaway, M. V. Pusic, R. M. Galbraith, and T. Cameron, "Developing the role of big data and analytics in health professional education," *Medical teacher*, vol. 36, no. 3, pp. 216–222, 2014.

[18] J. Eckroth, "Teaching future big data analysts: Curriculum and experience report," in *Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International*, pp. 346–351, IEEE, 2017.

[19] Z. B. Djordjevi, "Cloud Computing." https://canvas.harvard.edu/courses/4077/assignments/syllabus, 2015.

[20] J. A. González-Martínez, M. L. Bote-Lorenzo, E. Gómez-Sánchez, and R. Cano-Parra, "Cloud computing and education: A state-of-the-art survey," *Computers & Education*, vol. 80, pp. 132–151, 2015.

[21] N. Sultan, "Cloud computing for education: A new dawn?," *International Journal of Information Management*, vol. 30, no. 2, pp. 109–116, 2010.

[22] P. Calyam, S. Seetharam, and R. B. Antequera, "Geni laboratory exercises development for a cloud computing course," in *Research and Educational Experiment Workshop (GREE), 2014 Third GENI*, pp. 19–24, IEEE, 2014.

[23] C. Changchit, "Cloud computing: Should it be integrated into the curriculum?," *International Journal of Information and Communication Technology Education (IJICTE)*, vol. 11, no. 2, pp. 105–117, 2015.

[24] J. A. Shamsi, "A laboratory based course on gpu programming: Methods, practices, and lessons," in *Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International*, pp. 367–374, IEEE, 2017.

[25] S. Kumar, "oriented teaching of pdc topics in integration with other undergraduate courses at multiple levels: A multi-year report," *Journal of Parallel and Distributed Computing*, vol. 105, pp. 92–104, 2017.

[26] S. A. Bogaerts, "One step at a time: Parallelism in an introductory programming course," *Journal of Parallel and Distributed Computing*, vol. 105, pp. 4–17, 2017.

[27] T. Newhall, A. Danner, and K. C. Webb, "Pervasive parallel and distributed computing in a liberal arts college curriculum," *Journal of Parallel and Distributed Computing*, vol. 105, pp. 53–62, 2017.

[28] S. J. Matthews, "Using phoenix++ mapreduce to introduce undergraduate students to parallel computing," *Journal of Computing Sciences in Colleges*, vol. 32, no. 6, pp. 165–174, 2017.

[29] R. Sakellariou, "Experiences with teaching a second year distributed computing course," in *European Conference on Parallel Processing*, pp. 28–37, Springer, 2016.

[30] J. C. Adams, "Patternletsa teaching tool for introducing students to parallel design patterns," *Journal of Parallel and Distributed Computing*, vol. 105, pp. 31–41, 2017.

[31] D. R. Karthwohl and W. Anderson, "A revision of blooms taxonomy: An overview theory into practice," *The Ohio State University*, 2002.