Online-Journals.org

iJEP International Journal of Engineering Pedagogy

iJEP | eISSN: 2192-4880 | Vol. 13 No. 5 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijep.v13i5.38867

PAPER

MapReduce Solutions Classification by Their Implementation

Kamila Orynbekova[1](✉), Andrey Bogdanchikov[1], Selcuk Cankurt[2], Abzatdin Adamov[3], Shirali Kadyrov[1]

[1]Suleyman Demirel University, Almaty, Kazakhstan

[2]Vistula University, Warsaw, Poland

[3]ADA University, Baku, Azerbaijan

kamila.orynbekova@sdu.edu.kz

ABSTRACT

Distributed Systems are widely used in industrial projects and scientific research. The Apache Hadoop environment, which works on the MapReduce paradigm, lost popularity because new, modern tools were developed. For example, Apache Spark is preferred in some cases since it uses RAM resources to hold intermediate calculations; therefore, it works faster and is easier to use. In order to take full advantage of it, users must think about the MapReduce concept. In this paper, a usual solution and MapReduce solution of ten problems were compared by their pseudocodes and categorized into five groups. According to these groups' descriptions and pseudocodes, readers can get a concept of MapReduce without taking specific courses. This paper proposes a five-category classification methodology to help distributed-system users learn the MapReduce paradigm fast. The proposed methodology is illustrated with ten tasks. Furthermore, statistical analysis is carried out to test if the proposed classification methodology affects learner performance. The results of this study indicate that the proposed model outperforms the traditional approach with statistical significance, as evidenced by a p-value of less than 0.05. The policy implication is that educational institutions and organizations could adopt the proposed classification methodology to help learners and employees acquire the necessary knowledge and skills to use distributed systems effectively.

KEYWORDS

MapReduce, big data, Apache Hadoop, Apache Spark, problems classification, solutions categorization, course design

1 INTRODUCTION

Big data, defined as data sets that are too large and complex for traditional data-processing methods, has become an essential focus for organizations and individuals. The sheer volume of data generated by sources such as the Internet of Things, cyber security, social media, and bioinformatics makes it impossible to analyze using traditional methods or on a single machine. However, when analyzed correctly, big data can provide valuable insights and improve business operations,

Orynbekova, K., Bogdanchikov, A., Cankurt, S., Adamov, A., Kadyrov, S. (2023). MapReduce Solutions Classification by Their Implementation. *International Journal of Engineering Pedagogy (iJEP)*, 13(5), pp. 58–71. https://doi.org/10.3991/ijep.v13i5.38867

Article submitted 2023-02-14. Resubmitted 2023-05-15. Final acceptance 2023-05-16. Final version published as submitted by the authors.

decision-making, customer experiences, healthcare outcomes, risk management, and resource allocation in various sectors. Effectively analyzing big data is becoming a critical skill in many industries. With the rise of big data, organizations that can harness its power will have a significant advantage over those that do not.

To this end, a distributed system like Apache Hadoop can be used. Apache Hadoop is an ecosystem for storing and computing big data, which works on a cluster of extendable hardware. The storing layer is a Hadoop Distributed File System, and the processing layer is a MapReduce consisting of two steps: map and reduce. MapReduce is a computational model specifically developed to handle large-scale data processing through parallelization on a cluster of machines. Data processing is performed by manipulating data tokens represented as key-value pairs.

The steps of the MapReduce process are as follows (see Figure 1):

1. Big data is split into chunks and sent to mappers
2. Mappers perform their actions on those data chunks and emit intermediate key-value pairs
3. Apache Hadoop Ecosystem identifies each key's reducer by some hash function. Then it sends that key-value pair in sorted order to a definite reducer, called the shuffle stage.
4. Reducers accept sorted key-value pairs and finish data processing by sending the final result to the file system.

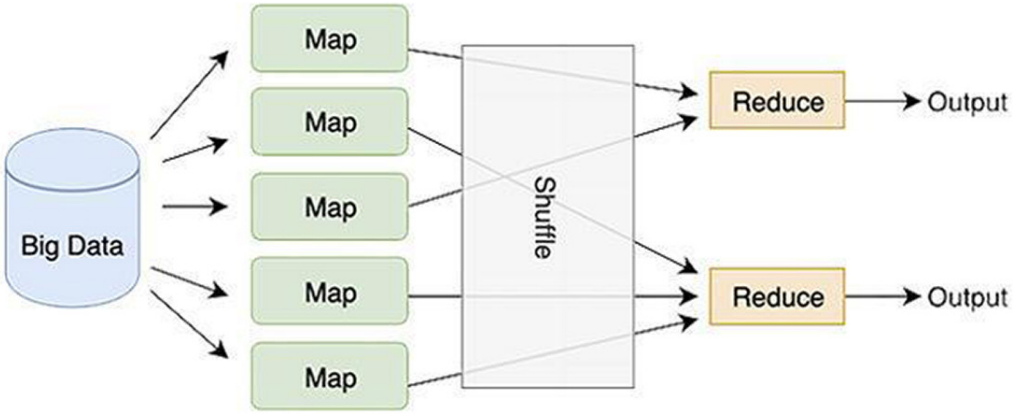

Fig. 1. MapReduce process

Many tools exist that work on top of Apache Hadoop and use the MapReduce model, e.g., Apache Spark. Apache Spark has gained significant popularity recently due to its many advantages over Apache Hadoop [1]. Apache Spark provides a flexible programming model by allowing users to choose from various programming languages, such as Scala, Python, R, SQL, and Java, to implement their solutions [2]. This flexibility simplifies the process of big data processing, as users are not required to perform pure MapReduce coding in Java. It is essential for individuals utilizing Apache Spark to have a thorough understanding of the MapReduce paradigm, even if they do not write the map and reduce methods in Java classes. This understanding is necessary to develop efficient solutions that fully leverage the capabilities of the MapReduce model when executed on a distributed system.

There is a lack of research on the most effective curriculum design for teaching the core concepts of MapReduce and Apache Spark that balance theoretical explanations with hands-on exercises and real-world examples to enhance students' understanding of the technologies.

The research contribution of the current work can be summarized as follows:

- Investigating and developing a novel approach that effectively teaches the core concepts of MapReduce and Apache Spark by including a balance between theoretical explanations, on the one hand, and hands-on exercises and real-world examples, on the other, which can enhance the student's understanding of the technologies.
- Evaluating the classification method's effectiveness in student performance in learning the core concepts of MapReduce and Apache Spark in distributed systems and providing feedback to improve the curriculum design.

This research contributes to the field of big data education by addressing the current gap in the literature on effective curriculum design for teaching MapReduce and Apache Spark and by providing an evaluation of the classification method in terms of student performance, which can inform future curriculum design and instruction. Additionally, this research could provide valuable insights into the most effective ways to classify single machine tasks in distributed systems, which is crucial for understanding big data processing, and optimizing the performance of distributed systems.

2 LITERATURE REVIEW

This section reviews the literature on curriculum development, educational theories, the MapReduce programming model and its applications, and compares MapReduce with other big data technologies. Constructivism, a learning theory emphasizing learners' active involvement in constructing their knowledge, has been incorporated into the curriculum development process, along with cognitive load theory, which suggests that learners have a limited capacity for processing information. The MapReduce programming model is a widely used paradigm for processing large datasets in parallel across a cluster of machines. It has been applied in various domains to solve big data challenges. However, Apache Spark, another popular big data processing framework, has emerged as an alternative to MapReduce. Several studies have been conducted on the curriculum design for teaching MapReduce, and the effectiveness of MapReduce education has also been evaluated. Based on the literature reviewed, there is a need for a curriculum that effectively teaches the core concepts of MapReduce on Apache Spark.

2.1 Curriculum development and educational theories

Constructivism, a learning theory emphasizing learners' active involvement in constructing their knowledge, has been incorporated into our curriculum development process [3], [4]. The aim is to facilitate meaningful learning experiences by encouraging learners to actively engage with the material, particularly in teaching the core concepts of MapReduce on Apache Spark.

Furthermore, cognitive load theory has been considered, suggesting that learners have a limited capacity for processing information [5], [6]. The curriculum has been carefully structured to present concepts in manageable chunks. Using a balance of theoretical explanations, hands-on exercises, and real-world examples is aimed at optimizing the learning experience for our students and maximizing their retention of the material [7].

By incorporating these relevant theories into the curriculum-development process, a learning experience that is both engaging and effective has been created. The curriculum's goal is to equip students with the knowledge and skills necessary to succeed in distributed systems, and we recommend using these theories in the development of future curricula.

2.2 MapReduce programming model and its applications

The MapReduce programming model is a paradigm for processing large datasets in parallel across a cluster of machines. It was introduced by Google in a 2004 paper [8] and has since been widely adopted in industry and academia. The basic idea behind MapReduce is to divide the processing into two phases: the Map phase, which transforms the input data, and the Reduce phase, which aggregates the intermediate data produced by the Map phase. The Hadoop framework [9] is one of the most popular ways to implement MapReduce, and it provides a robust and scalable infrastructure for running MapReduce jobs.

MapReduce has been applied in various domains to solve big data challenges. For example, MapReduce has been used in finance to identify fraudulent activity and improve risk management [10]; in healthcare, it has been used to analyze electronic health records to improve patient outcomes [11]; in social media, it has been used to analyze user behavior and sentiment [12]. The MapReduce model has also been applied in bioinformatics for gene expression analysis [13] and genome assembly [14], [15] in teaching quality assessment [16].

While the MapReduce model has proven effective for many big data use cases, it is not always the best choice. Apache Spark [2] is another popular big data processing framework, which is an in-memory processing engine and provides a more general-purpose data processing engine than MapReduce, with the ability to perform batch processing, interactive querying, graph processing, and machine-learning tasks. So, comparison studies between the two frameworks were made.

2.3 Comparison of MapReduce with other big data technologies

MapReduce is a widely used programming model for processing large datasets in parallel across a cluster of machines. With the growing importance of big data in various industries, there is a need for individuals with knowledge and skills in MapReduce programming. Therefore, designing a curriculum for teaching MapReduce is crucial for preparing students for careers in big data. Curriculum design for teaching MapReduce can include modules taught in distributed computing courses to upper undergraduate students [1], as well as cloud-based platforms for knowledge sharing and big data [17], [18].

Several studies have been conducted on the curriculum design for teaching MapReduce in higher education [19]. These studies focus on improving teaching efficiency, providing students with more practical capabilities and knowledge, and incorporating hands-on activities into the syllabus [17]. Other research has examined the construction of curriculum ideological and political collaborative education mechanisms [20], [21].

Several studies have been conducted to evaluate the effectiveness of MapReduce education [22]. These studies focus on building efficient algorithms, task and node faults, performance evaluation, and data-mining analysis. For example, in a study [23]

conducted at the University of California, Berkeley, researchers utilized Amazon Elastic Compute Cloud (EC2) to instruct undergraduate students on the concepts of MapReduce. The study demonstrated a significant improvement in processing speed. However, the researchers raised concerns regarding the applicability of cloud-based billing services in an educational setting.

Furthermore, studies have focused on developing and evaluating teaching materials, such as lab assignments and tutorials, to help students learn MapReduce efficiently. For example, Ngo et al. [24] describes their teaching experiences and student feedback in a Hadoop MapReduce course. It provides best practice for lecture materials, a computing platform, and teaching methods, and it groups the materials into four categories. The survey showed positive student feedback, and the authors suggest a centralized shared computing resource for students to set up individual Hadoop clusters. Another study [25] describes the reimplementation of an entry-level graduate course in high-performance parallel computing aimed at physical scientists. It utilizes development to teach students about high-performance computing. The curriculum is designed to be hands-on and practically teach senior graduate students.

In conclusion, curriculum design for teaching MapReduce is essential to preparing students for careers in big data. Many studies have been conducted on curriculum design, materials development, and the evaluation of the effectiveness of MapReduce education. Based on the literature reviewed, one research problem that needs to be addressed is developing a curriculum that effectively teaches the core concepts of MapReduce on Apache Spark. There is a need for a curriculum that balances theoretical explanations with hands-on exercises and real-world examples, which can enhance the student's understanding of the technologies. To address this research gap, we asked the following research questions:

1. Is it possible to develop and apply a curriculum that effectively teaches the core concepts of MapReduce on Apache Spark, using a balance of theoretical explanations, hands-on exercises, and real-world examples, to classify single-machine tasks in distributed systems?
2. Is the classification method effective in terms of student performance in learning the core concepts of MapReduce on Apache Spark in distributed systems?

3 METHODOLOGY

3.1 Instructional design methodology

We used an instructional design approach, called the ADDIE model, to address the first research question. The instructional design approach is a systematic process for creating effective and efficient instructional materials and activities. When designing a course with expert opinion, the instructional design approach includes five steps: Analyze, Design, Develop, Implement, and Evaluate.

The analysis step involves gathering information about the target audience, their needs, and the instructional goals and objectives of the course [26]. This information is gathered through expert opinion and other research methods such as needs analysis and formative evaluations. In the next step, based on the information gathered in the analysis step, instructional materials and activities are designed to meet the target audience's needs and achieve the instructional goals and objectives [27]. This step also involves identifying the specific content and skills that need

to be taught and developing a scope and sequence for the course. Then comes the development, which involves creating instructional materials and activities, such as lesson plans and assessments. During this step, expert opinion ensures that the materials align with best practices and evidence-based practices in the field [28]. Once the instructional materials and activities are developed, they need to be implemented by teachers, and expert opinion can be used to ensure the instruction is being implemented as intended. The final step is evaluating the instruction materials, activities, and the course to measure whether the course met the instructional goals and objectives and how the students received it.

Since the last two steps are closely related to our second research question, we provide more details of the implementation and evaluation methods below.

3.2 Summative evaluation and statistical analysis

To address our second research question, we utilized the summative evaluation method used to assess a course's effectiveness after completion [29]. It involves measuring the extent to which the course achieved its objectives and making necessary changes for future iterations. One of the most common ways to conduct summative evaluation is by using an experimental design consisting of a control group and an experimental group [30]. The experimental group consisted of 21 students exposed to the proposed novel approach, while the traditional teaching approach was used for the control group, which also consisted of 21 students. The average age of the students was 19–20 years old; the year of study was 3–4. The male-to-female ratios in the experimental and control groups were 16:5 and 15:6, respectively. To determine whether there was a significant difference between the mean performance of the two groups, we utilized a statistically independent sample t-test. Before conducting the t-test, an outlier analysis was performed using box plots. This helps identify values that deviate significantly from the rest of the data set, allowing for a more accurate and reliable data analysis. To ensure that the sample selected was representative of the population and that the sampling process was done randomly, initial hypothesis testing was conducted on the students' GPAs.

4 RESULT

4.1 Implementation-based classification of single machine tasks

In this section, we seek to answer whether a curriculum that effectively teaches MapReduce core concepts on Apache Spark to classify single-machine tasks in distributed systems can be developed.

The ADDIE methodology was employed over approximately eight weeks to address the shortcomings of the traditional Distributed Big Data Systems course. In the first phase, three experts with relevant expertise analyzed the course's learning outcomes and found that they were not being met. In response, the experts proposed a new curriculum based on their expert opinions. The experts developed new course objectives incorporating active learning methods to establish new course objectives. Subsequently, the study's first author designed the course content, materials, and possible activities.

During the development phase, the experts held regular meetings for approximately five weeks to ensure that the quality of the course content and activities met their standards. The experts finalized the following course materials through analysis, design, and development.

Five classes were determined by comparing a single-machine solution and a distributed MapReduce solution. For example, problems such as pirate speech and log analysis (SQL injection) are simple to implement on MapReduce because the lines of data do not interact. Without interaction, only map steps can be realized without a reducer. The map will be implemented very similarly to a single machine solution, but only for one line, and will be repeated till the end of the data (see Figure 2). Sentiment analysis can also be performed without a reducer, but another solution was mentioned in this paper.

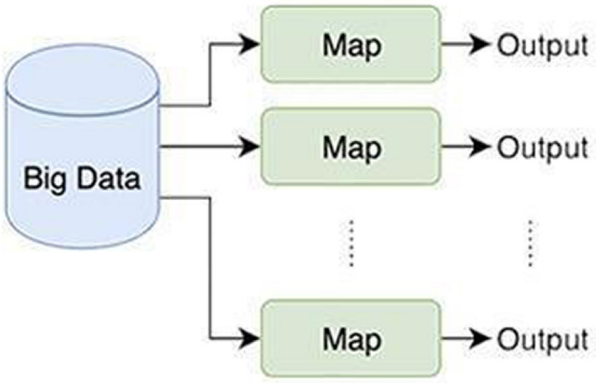

Fig. 2. Job process without reducer

Another type of problem involves counting (wordcount, GooglePlay frequency). Solutions are easy: the map splits data and sends it to reducer keys that need to be counted with value 1. These values will come to the reducer in combined form as a set of 1's. The reducer works with one key and one set of values. In this case, it summarizes values and emits the key and its count (see Figure 3).

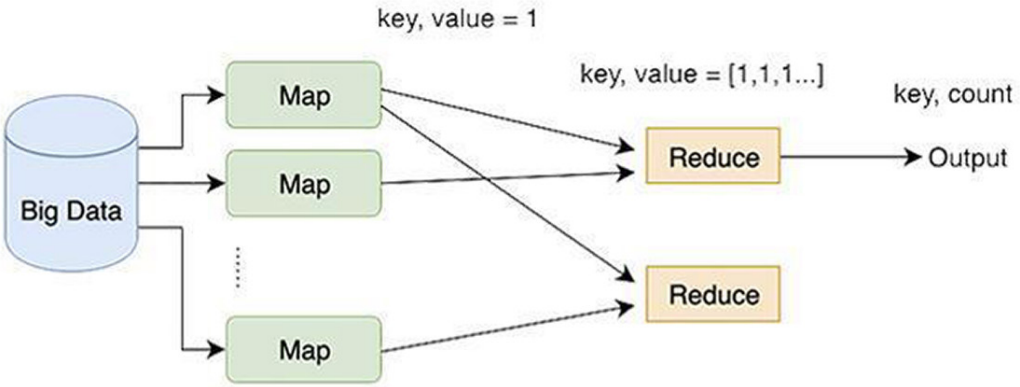

Fig. 3. MapReduce process: wordcount example

The trending wordcount problem shows the joining of data when the output of one or more MapReduce job can be an input of another MapReduce job (see Figure 4). Here also, users will understand how the environment performs the shuffle (sorting, combining).

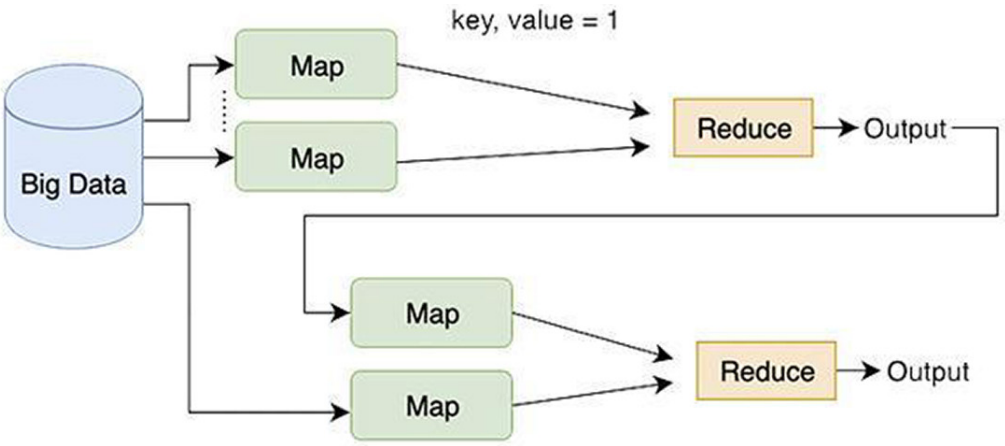

Fig. 4. MapReduce process: joining data

Problems might contain very few keys and need to find many min, max, and k-mean values. In these problems, keys are initially known. First, aps need to identify corresponding values. Then, the reducer receives a key with a set of values, performs actions with the set, and emits the key with the result (see Figure 5).

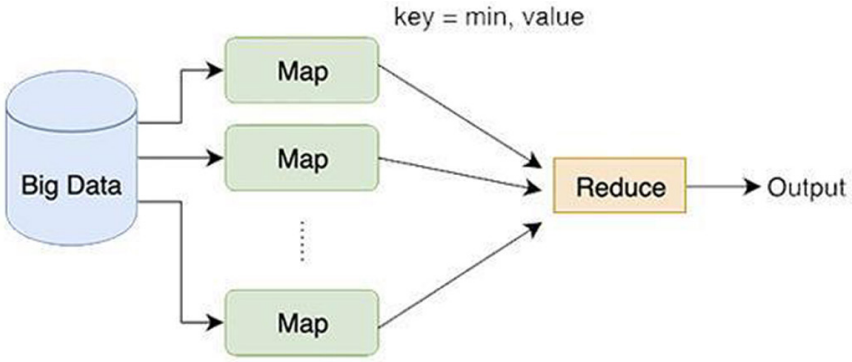

Fig. 5. MapReduce process: finding min/max example

Decision-tree and a priori algorithm problems are more challenging to implement, just as in single-machine solutions. Here the data lines interact, and keys and values change during the process. The MapReduce job repeats until the condition yields a "yes" output. (see Figure 6).

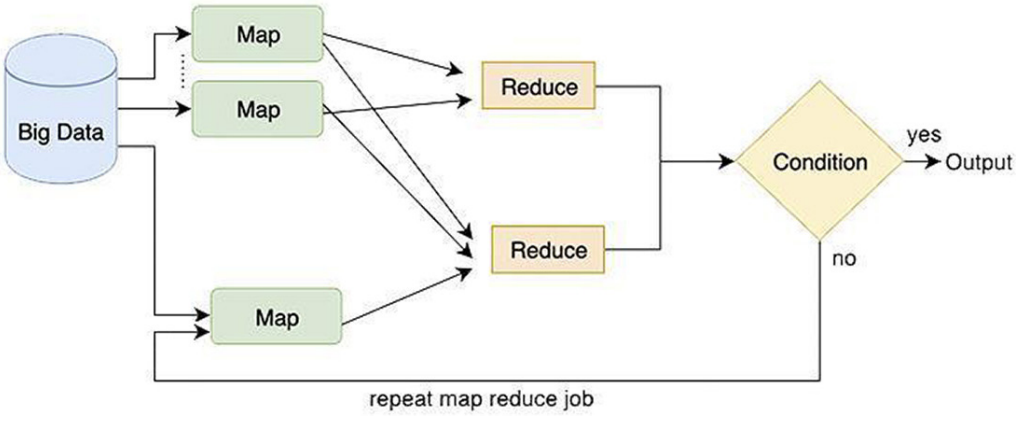

Fig. 6. MapReduce process, with condition

The five problem classes discussed above and additional problem types are summarized in Table 1.

Table 1. Ten MapReduce problem types and their classifications*

#	Title	Type
1	WordCount [31]: the aim is to calculate the frequency of appearance of words in text	#2 (related to counting)
2	Pirate speech [32] aims to change the text's style to a pirate's speaking style. Ex.: change "ing" to "in'," "the" to "da."	#1 (lines of data do not interact with each other)
3	GooglePlay frequency: the aim is to find how many applications were created on GooglePlay each month	#2 (related to counting)
4	Log analysis [33]: the aim is to determine the SQL injection and DDOS attacks	#1 (lines of data do not interact with each other), #2 (related to counting)
5	Finding min and max [31]: the aim is to find min and max	#3 (contains few known keys and many unknown values)
6	Trending WordCount [31]: the aim is to identify the frequency of word appearance by date and total in Twitter, expected output: word, date, sum_date, total_sum	#4 (needs to join data and use the output of one MapReduce job in another MapReduce job)
7	Sentiment analysis [34]: the aim is to identify if the sentence is positive, negative, or neutral.	#1 (lines of data do not interact with each other), #2 (related to counting)
8	k-means clustering [35]: the aim is to find centroids of clusters	#3 (contains few known keys and many unknown values), #5 (problem is related to the condition)
9	Decision tree [36]: the aim is to build a decision tree	#5 (the problem is related to the condition)
10	A priori algorithm [37]: the aim is to calculate the frequency of a set with symptoms of diseases.	#5 (problem is related to the condition)

Note: *Pseudocodes can be provided upon request.

The new teaching methodology was implemented for approximately two weeks—significantly shorter than the traditional six-week approach. In the first class, the experimental group was introduced to the new approach and the materials. The extended version of the material discussed earlier was provided to students for self-study to help them better understand the MapReduce concept. A role-playing exercise was conducted in the subsequent class, with one student acting as the Hadoop environment. In contrast, others assumed roles such as mapper, reducer, and batches from a dataset. This exercise was intended to facilitate a deeper understanding of the course concepts among the students.

A problem-based active-learning strategy was implemented in the subsequent classes by providing various take-home tasks for students to complete. The students' progress was closely monitored through various assessments, and regular feedback was provided to help them improve their performance. Throughout the course, experts regularly visited the classes and organized meetings to evaluate the effectiveness of the newly developed course.

The abbreviated course duration limited students' ability to grasp the material thoroughly. However, the problem-based approach and implemented active-learning strategies were intended to mitigate this potential limitation. Overall, the new methodology effectively facilitates a deeper understanding of the MapReduce programming model among students. Further research could investigate the effectiveness of the new approach over longer durations and in different contexts.

4.2 Evaluation of the novel approach

In this section, we report the evaluation results of the proposed method. We carried out a hypothesis test to compare the mean GPA of the control group (mean = 2.73, SD = 0.56) with that of the experimental group (mean = 2.80, SD = 0.64). From the test results, we obtained a t-value of –0.37 and a p-value of 0.71, indicating insufficient evidence to reject the null hypothesis that the mean GPA of the two groups was equal. This shows that the two groups can safely be assumed to be randomly distributed.

We used an independent sample t-test to investigate if the students' performance increases with the proposed novel approach compared with the traditional teaching method. Before applying the t-test, we checked for outliers using box plots, as shown in Figure 7.

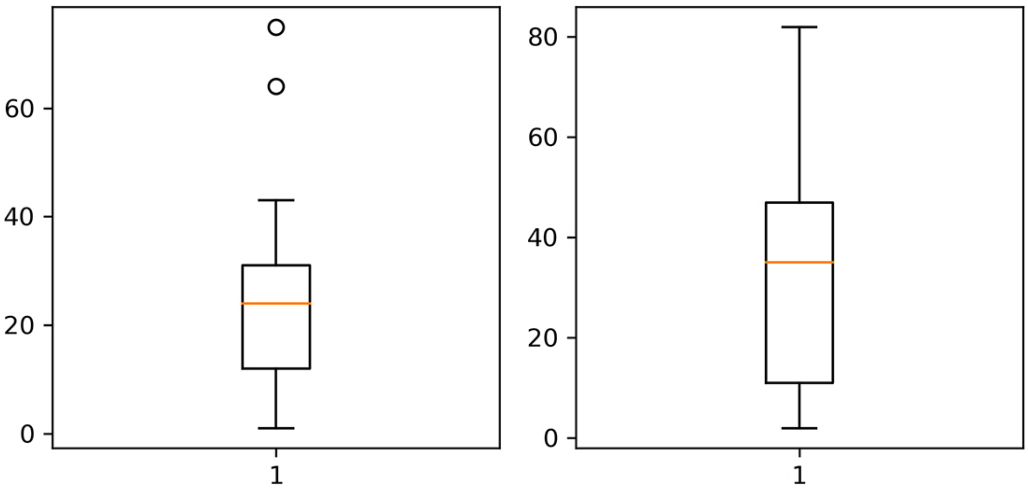

Fig. 7. Box plot of MapReduce exam grades for control and experimental groups, respectively

From the box plots, we see that there are two outliers in the control group. Removing those two outliers, we are left with 19 students in the control group with a mean of 20.47 (SD = 12.25) and 21 students in the experimental group with a mean of 32.67 (SD = 25.12). An independent sample, a one-tailed t-test, was utilized to determine if there was a statistically significant difference in the experimental group's mean compared with that of the control group. The analysis results revealed a t-value of –1.92 and a p-value of 0.0313. Since $p < 0.05$, we conclude that the experimental group significantly outperforms the control group.

Overall, the findings of this study suggest that the proposed novel teaching approach can improve student performance in MapReduce exams compared with traditional teaching methods. However, it should be noted that further studies with larger sample sizes are needed to confirm these findings.

5 DISCUSSION AND CONCLUSION

The primary objective of this study was to investigate novel methodologies for teaching the MapReduce programming model through task-based classification and to evaluate the effectiveness of the proposed model on student performance. To address this objective, the study aimed to answer two research questions.

The first research question aimed to identify and classify various problems that can be addressed using the MapReduce programming model. To accomplish this, an instructional design methodology was used in conjunction with expert opinion to categorize the tasks into five distinct problem categories: problems requiring no need for reducers, counting problems, problems with initially known keys, problems requiring the joining of data from separate outputs, and problems involving shuffling keys and values. These problem categories were presented in Table 1 to help readers understand the characteristics of each category. As a result, users can classify their problems and find similar solutions from the appropriate category.

The second research question aimed to evaluate the effectiveness of the proposed model. A summative evaluation methodology was used to accomplish this, and the results were analyzed using a statistically independent sample t-test. The results of this study reveal that the proposed novel approach outperformed the traditional approach, with a statistically significant difference ($p < 0.05$) in student performance on the summative MapReduce examination.

In conclusion, the study aimed to investigate novel methodologies for teaching the MapReduce programming model and evaluate the proposed model's effectiveness on student performance. Through the identification and classification of various types of problems, the study has demonstrated the effectiveness of task-based classification in MapReduce learning and that the proposed novel approach was superior to the traditional approach regarding student performance. Additionally, readers will understand the MapReduce concept and how to solve big data problems without the need for extensive experience in writing MapReduce jobs. The study also highlighted that the set of problems that can be addressed using MapReduce is constantly expanding and not limited to the five categories presented in this paper. Therefore, in future work, the authors plan to continue to investigate new problem categories and expand the set of problems that can be addressed using the MapReduce programming model.

It is important to note that our study evaluated student performance only on a summative examination. Future research could explore the long-term retention of knowledge and skills learned through this approach. Additionally, while we aimed to balance theoretical explanations with hands-on exercises and real-world examples, some students may have struggled with the technical aspects of the course material. Further research could investigate ways to support students needing additional assistance in this area.

As a policy implementation, the proposed classification methodology can be implemented in educational settings and workplaces to aid in acquiring the knowledge and skills required to utilize distributed systems effectively.

6 REFERENCES

[1] M. Zaharia *et al.*, "Apache spark: a unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016. https://doi.org/10.1145/2934664

[2] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," presented at the 2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10), 2010.

[3] E. Von Glasersfeld, "Constructivism in education," *Int. Encycl. Educ.-Res. Stud. Suppl.*, vol. 1, pp. 162–163, 1989.

[4] E. Fatourou, N. C. Zygouris, T. Loukopoulos, and G. I. Stamoulis, "Teaching concurrent programming concepts using Scratch in primary school: methodology and evaluation," *Int. J. Eng. Ped. (iJEP)*, vol. 8, no. 4, pp. 89–105, 2018. https://doi.org/10.3991/ijep.v8i4.8216

[5] J. Sweller, "Cognitive load during problem-solving: effects on learning," *Cogn. Sci.*, vol. 12, no. 2, pp. 257–285, 1988. https://doi.org/10.1207/s15516709cog1202_4

[6] P. Silapachote and A. Srisuphab, "Engineering courses on computational thinking through solving problems in artificial intelligence," *Int. J. Eng. Ped. (iJEP)*, vol. 7, no. 3, pp. 34–49, 2017. https://doi.org/10.3991/ijep.v7i3.6951

[7] R. Wood, A. McGlashan, C. Moon, and W. Kim, "Engineering education in an integrated setting," *Int. J. Eng. Ped. (iJEP)*, vol. 8, no. 3, pp. 17–27, 2018. https://doi.org/10.3991/ijep.v8i3.7857

[8] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008. https://doi.org/10.1145/1327452.1327492

[9] T. White, *Hadoop: The definitive guide*. O'Reilly Media, Inc., 2012.

[10] J. Chen, Y. Tao, H. Wang, and T. Chen, "Big data based fraud risk management at Alibaba," *J. Finance Data Sci.*, vol. 1, no. 1, pp. 1–10, 2015. https://doi.org/10.1016/j.jfds.2015.03.001

[11] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health Inf. Sci. Syst.*, vol. 2, pp. 1–10, 2014. https://doi.org/10.1186/2047-2501-2-3

[12] I. Ha, B. Back, and B. Ahn, "MapReduce functions to analyze sentiment information from social big data," *Int. J. Distrib. Sens. Netw.*, vol. 11, no. 6, p. 417502, 2015. https://doi.org/10.1155/2015/417502

[13] Y. Abduallah, T. Turki, K. Byron, Z. Du, M. Cervantes-Cervantes, and J. T. Wang, "MapReduce algorithms for inferring gene regulatory networks from time-series microarray data using an information-theoretic approach," *BioMed Res. Int.*, vol. 2017, 2017. https://doi.org/10.1155/2017/6261802

[14] A. McKenna *et al.*, "The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data," *Genome Res.*, vol. 20, no. 9, pp. 1297–1303, 2010. https://doi.org/10.1101/gr.107524.110

[15] S. Ghoneimy and S. Abou El-Seoud, "A MapReduce framework for DNA sequencing data processing," *Int. J. Recent Contrib. Eng. Sci. IT IJES*, vol. 4, no. 4, pp. 11–20, 2016. https://doi.org/10.3991/ijes.v4i4.6537

[16] H. Hu and J. Zheng, "Application of teaching quality assessment based on parallel genetic support vector algorithm in the Cloud Computing Teaching System.," *Int. J. Emerg. Technol. Learn.*, vol. 11, no. 8, 2016. https://doi.org/10.3991/ijet.v11i08.6040

[17] Y. Zhao and H. Liu, "Cloud curriculum resource management platform based on Hadoop," *Meas. Control*, vol. 53, no. 9–10, pp. 1782–1790, 2020. https://doi.org/10.1177/0020294020948088

[18] K. Xiangsheng, "Big data X-learning resources integration and processing in Cloud environments.," *Int. J. Emerg. Technol. Learn.*, vol. 9, no. 5, 2014. https://doi.org/10.3991/ijet.v9i5.3783

[19] X. Li *et al.*, "Curriculum reform in big data education at applied technical colleges and universities in China," *IEEE Access*, vol. 7, pp. 125511–125521, 2019. https://doi.org/10.1109/ACCESS.2019.2939196

[20] F. Yang, Y. Rao, K. Wu, G. Wang, Y. Bao, and C. Liu, "Construction of curriculum ideological and political collaborative education mechanism based on Edge Computing and Neural Network algorithm," *Comput. Intell. Neurosci.*, vol. 2022, 2022. https://doi.org/10.1155/2022/3596665

[21] C. Q. Li, "Teaching mechatronics to non-traditional mechanical engineering students-An adaptive approach.," *Int. J. Eng. Ped. (iJEP)*, vol. 11, no. 3, 2021. https://doi.org/10.3991/ijep.v11i3.15833

[22] Y. Han, "Hadoop Data Mining Analysis of Network Education Platform based on PDM New Media Data Perspectives," presented at the 2022 International Conference on Inventive Computation Technologies (ICICT), IEEE, 2022, pp. 1144–1147. https://doi.org/10.1109/ICICT54344.2022.9850575

[23] A. S. Rabkin, C. Reiss, R. Katz, and D. Patterson, "Experiences teaching MapReduce in the cloud," presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education, 2012, pp. 601–606. https://doi.org/10.1145/2157136.2157310

[24] L. B. Ngo, E. B. Duffy, and A. W. Apon, "Teaching HDFS/MapReduce systems concepts to undergraduates," presented at the 2014 IEEE International Parallel & Distributed Processing Symposium Workshops, IEEE, 2014, pp. 1114–1121. https://doi.org/10.1109/IPDPSW.2014.124

[25] J. A. Shamsi, N. M. Durrani, and N. Kafi, "Novelties in teaching high performance computing," presented at the 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, IEEE, 2015, pp. 772–778. https://doi.org/10.1109/IPDPSW.2015.88

[26] R. A. Reiser and J. V. Dempsey, *Trends and issues in instructional design and technology*. Pearson Boston, MA, 2012.

[27] P. L. Smith and T. J. Ragan, *Instructional design*. John Wiley & Sons, 2004.

[28] G. R. Morrison, S. J. Ross, J. R. Morrison, and H. K. Kalman, *Designing effective instruction*. John Wiley & Sons, 2019.

[29] M. Janus and S. Brinkman, "Evaluating early childhood education and care programs," in *International encyclopedia of education*, Pergamon, 2010, pp. 25–31. https://doi.org/10.1016/B978-0-08-044894-7.01197-0

[30] C. S. Reichardt, *Quasi-experimentation: A guide to design and analysis*. Guilford Publications, 2019.

[31] D. Miner and A. Shook, *MapReduce design patterns: building effective algorithms and analytics for Hadoop and other systems*. O'Reilly Media, Inc., 2012.

[32] C. Lam, *Hadoop in action*. Simon and Schuster, 2010.

[33] Y. Azizi, M. Azizi, and M. Elboukhari, "Log files analysis using MapReduce to improve security," *Procedia Comput. Sci.*, vol. 148, pp. 37–44, 2019. https://doi.org/10.1016/j.procs.2019.01.006

[34] P. Gupta, P. Kumar, and G. Gopal, "Sentiment analysis on Hadoop with Hadoop streaming," *Int. J. Comput. Appl.*, vol. 121, no. 11, 2015. https://doi.org/10.5120/21582-4651

[35] T. H. Sardar and Z. Ansari, "Partition-based clustering of large datasets using MapReduce framework: An analysis of recent themes and directions," *Future Comput. Inform. J.*, vol. 3, no. 2, pp. 247–261, 2018. https://doi.org/10.1016/j.fcij.2018.06.002

[36] W. Dai and W. Ji, "A MapReduce implementation of C4. 5 decision tree algorithm," *Int. J. Database Theory Appl.*, vol. 7, no. 1, pp. 49–60, 2014. https://doi.org/10.14257/ijdta.2014.7.1.05

[37] S.-Y. Choi and K. Chung, "Knowledge process of health big data using MapReduce-based associative mining," *Pers. Ubiquitous Comput.*, vol. 24, no. 5, pp. 571–581, 2020. https://doi.org/10.1007/s00779-019-01230-3

7 AUTHORS

Kamila Orynbekova is a Senior Lecturer in the Computer Sciences Department, Faculty of Engineering and Natural Sciences, and a Head of Distributed Systems and Computing Laboratory in Suleyman Demirel University, Kaskelen, Almaty, Kazakhstan. Also she is a PhD student in the Computer Sciences educational program. (email: kamila.orynbekova@sdu.edu.kz)

Andrey Bogdanchikov holds the title of Associate Professor at Suleyman Demirel University's Faculty of Engineering and Natural Sciences, within the Department of Information Systems, and Vice-Rector of Academic Affairs situated in Abylai khan street 1/1, Kaskelen, Kazakhstan. He obtained his Doctor of Philosophy degree in 2014 from Suleyman Demirel University, Kazakhstan. His areas of expertise include the fields of Distributed Systems, Parallel Computing and Programming Languages. (email: andrey.bogdanchikov@sdu.edu.kz).

Selcuk Cankurt holds the title of Assistant Professor at Vistula University in the Department of Computer Engineering in Warsaw, Poland. He graduated from the University of Marmara, Istanbul, Turkey in 1997. He received the M.S. and Ph.D. degrees in Information Technologies from International Burch University, Sarajevo, Bosnia and Herzegovina, in 2011 and 2015, respectively. He has studied in the areas of database systems, data warehouse, data cubes, data mining, business intelligence, artificial intelligence, machine learning, and fuzzy systems. His present research interests are data science, data lake, big data, big data analytics, deep learning and natural language processing. (email: s.cankurt@vistula.edu.pl).

Abzatdin Adamov is a director of the Center for Data Analytics Research and faculty member at the School of Information Technology and Engineering, ADA University. He is an adjunct professor at the Computer Science Department, George Washington University. He is SIEEE and the Founding General Chair of the IEEE International Conference on Application of Information and Communication Technologies. (email: aadamov@ada.edu.az).

Shirali Kadyrov holds the title of Professor at Suleyman Demirel University"s Faculty of Engineering and Natural Sciences, within the Department of Mathematics and Natural Sciences situated in Kaskelen, Kazakhstan. He obtained his Doctor of Philosophy degree in 2010 from The Ohio State University in Columbus, Ohio, USA. His areas of expertise include the fields of Dynamical Systems, Mathematics Education, and Data Science. (email: shirali.kadyrov@sdu.edu.kz).