

Introducing Parallel and Distributed Computing concepts through the use of Flashcards and a Card Game

Mary L. Smith, Ed.D.
Computer Science
Hawaii Pacific University
Honolulu, HI, USA
mlsmith@hpu.edu

Srishti Srivastava, Ph.D.
Computer Science
University of Southern Indiana
Evansville, IN, USA
fsrishti@usi.edu

Abstract—This paper describes two active learning strategies to teach and review fundamental PDC concepts in early computer science courses. Questions were created based on eight PDC concept categories. In the first phase, flashcards were created for students to review the concepts. In the second phase, a card game called PDC Quest was created to allow groups of students to engage collaboratively in learning and reviewing the concepts.

Index Terms—Parallel and Distributed Computing, Student Engagement, Active Learning, Early Computer Science Undergraduate Curriculum, Flash Cards Game

I. INTRODUCTION

Initiatives by the National Science Foundation (NSF) in conjunction with the Institute for Electronic and Electrical Engineers' Technical Committee of Parallel Processing (IEEE-TCPP) encourage and promote the introduction of parallel and distributed computing (PDC) concepts in the early CS undergraduate courses [26]. We believe that PDC is a broad and complex subject and should be introduced first through terminology, essential concepts, and fundamentals while progressing into how to program and implement parallel and distributed systems. Certainly, textbooks and lectures are excellent tools for learning new subjects; however, many include so much detail that students can get lost and miss the basics. In our prior work, we have established a proof of concept for employing unplugged activities as a supplement to traditional teaching methods to increase student engagement toward learning complex PDC concepts [30]. To advance learning the PDC concepts in early CS undergraduate courses, we created active learning tools to supplement the teaching and learning of PDC concepts by utilizing one of the oldest study methods, flashcards. We also developed a card game that can be played in groups of students to make learning more engaging and collaborative.

Active Learning: Active learning is any instructional technique involving students in learning by actively participating

in activities, discussions, and problem-solving rather than passively listening to a lecture or watching a video. Active learning can be done individually by reflecting on learning through activities such as reflective writing assignments or flashcards. Active learning is often used with groups of students in discussions, hands-on activities, or games. Research has shown that active learning increases student engagement, motivation [27], and performance [28].

II. BACKGROUND AND RELATED WORK

A. PDC Inclusion in Early CS Education

Parallel and distributed computing has become an important topic of research and educational interest in CS and other STEM disciplines, mainly in areas of applications of computational science and engineering for several years. Cohen [5] has pointed out “the power of parallel thinking” and has stressed promoting a parallel structure for processing information. However, not every CS professional is familiar with PDC and has not been exposed to a parallel thinking approach. PDC generally envelopes areas such as distributed systems, parallel computational hardware, high-performance computing, and others. However, recent CS graduates, the leading workforce for current research and industries, are not versed in the fundamental knowledge of PDC. An online course available as a MOOC by Mullen et al., [6] focused on teaching high-performance computing to professionals. However, pursuing learning in a complicated field such as PDC from online sources can be challenging. In addition, there has been an effort to teach PDC courses in some universities as an elective CS course. For example, the work presented by Lin in [7] provides a fair evaluation of teaching effectiveness and discusses the survey as their data collection and the related test results. Much work has also been reported on integrating PDC topics throughout the CS undergraduate curriculum through a modular approach [8]- [13]. In other work, authors are proposing possible teaching materials to infuse “parallel thinking” for undergraduate students and sharing teaching materials for PDC education [14]- [16]. Bogaerts [17]- [20] has worked on introducing parallelism in CS0 and CS1 level courses by

The authors would like to thank the NSF-supported Center for Parallel and Distributed Computing Curriculum Development and Educational Resources (CDER) Early Adopter Program

including some hands-on in-class programming activities as a few additional lectures in existing CS courses. Overall, authors in some of the literature [19]- [25] explain that students struggle to understand the theoretical concepts and parallel programming due to the complexity imposed by the inherent nature of PDC. Thus, there could be better approaches to teaching PDC topics to CS undergraduate students rather than simply introducing the topic via traditional lecture methods and programming-based teaching.

B. Flashcards for Active Learning

Flashcards promote active recall, engage metacognition, and confidence-based repetition. When students look at a flashcard's *question* side and think of the answer, they actively recall. The student attempts to recall the concept rather than merely staring at the information in a textbook. Research has found that the kind of active recall retrieval practice found in flashcards can lead to better retention than passive studying [1] [2]. When students look at the *answer* side of the flashcard, they evaluate how their answer compares to the correct answer and how well they know the answer. The act of reflecting on learning is considered metacognition. Research has consistently found that applying metacognitive strategies embeds memories deeper into a student's knowledge, leading to better learning outcomes [3]. Because flashcards are separate items and not tied to a document or book, students can separate them into piles based on whether they need to study the concept again. Therefore, they can study the concepts they are not confident in more often, a type of confidence-based repetition. Confidence-based repetition, a derivative of spaced repetition, has consistently been proven in research as a way to improve memory performance [4].

III. METHODOLOGY

In phase one of this project, we created a set of flashcards introducing fundamental concepts of PDC in eight categories shown in Table I.

TABLE I: PDC concept categories

PDC Concept Category	Code
Concurrency	CC
Mutual exclusion	ME
Consistency in state/memory manipulation	CM
Message passing	MP
Shared-memory models	SM
Scalability	SC
Scheduling and load balancing	SL
Speedup/Amdahl's law	SA

The questions on the flashcards are based on our teaching experience and knowledge of PDC. Depending on the class level will determine the categories covered. For example, in a CS 1 course, the Concurrency and Speedup/Amdahl's law categories may be the only concepts covered, whereas, in a Systems course, all categories may be covered. A list of sample questions and answers based on those PDC topics follows.

Sample Questions and Answers:

1) Concurrency

- What is concurrency as it relates to parallel and distributed computing (PDC)?

Answer: It is the process of executing multiple tasks at the same time; however, it may not be done simultaneously.

- How do you achieve concurrency on a single-core processor?

Answer: With the use of multiprogramming. In multiprogramming, the OS rapidly switches back and forth between multiple programs that have been loaded into memory. We can say that blocks of the two programs' instructions are interleaved, meaning that the processor alternates which program is running. Because this happens so quickly, the concurrency provided by multiprogramming creates the illusion of parallel execution.

- What are the three main types of concurrent computing?

Answer: threading, asynchrony, and preemptive multitasking

2) Mutual Exclusion

- What are the two types of mutual exclusion algorithms in a distributed computing system?

Answer: token-based and non-token based

- What would be an example of implementing fairness in the context of mutual exclusion contention problem?

Answer: timestamps - requests for access to the shared resource (or critical section) are satisfied in the order of their timestamps

3) Message Passing

- What are the two basic inter-process communication operations in a message passing model in parallel computing?

Answer: send() and receive()

- What is MPI?

Answer: Message passing interface (MPI) is a standard specification in the form of a library of functions that programmers can call from C, C++, or Fortran code to write parallel programs for parallel computation in distributed-memory systems.

- What are the three different communication methods that MPI processes can use to communicate with each other?

Answer: Point-to-point, collective, and one-sided

4) Shared Memory Models

- What are the classifications of shared memory machines?

Answer: Uniform Memory Access (UMA) of Symmetric Multi Processor (SMP) and Non Uniform Memory Access (NUMA) machines

- Most commonly used shared memory programming model?
Answer: Thread model. Example POSIX threads, OpenMP

5) Scalability

- What are two types of scaling in parallel computing?
Answer: Strong scaling and weak scaling
- What type of scaling occurs when the overall problem size remains fixed but the number of processors executing the problem increases?
Answer: Weak scaling
- What type of scaling occurs when the problem size per processor remains fixed when more processors are added to the parallel system?
Answer: Strong Scaling

6) Parallel Speedup/Amdahl's Law

- What is Amdahl's law?
Answer: Amdahl's Law states that potential program speedup is defined by the fraction of code (P) that can be parallelized.
- What will be the speedup for a system that has 10 processors executing an application code that is 50% parallel?
Answer: 1.82

7) Scheduling and Load Balancing

- What is the parallel execution time of an application running on a system of 5 parallel processors? Following are the listed time (in seconds) to compute the application tasks scheduled on each processor: P1-20s, P2-5s, P3-37s, P4-17s, P5-50s.
Answer: 50 seconds
- What kind of parallel computing concept can be observed in a scheduler-task pool approach, where each task (mapped to a processor) finishes its work, it receives a new piece from the central work queue?
Answer: Dynamic load balancing

A. Phase 1- Flashcards

To create the flashcards, we searched for free online flashcard apps and evaluated the ones listed in Table II. The apps had similarities while also having unique features. For example, Canva's free flashcard design maker allows the creator to create flashcards, print them or save them as a .jpg or .png file to share with others. Cram allows the creator to create flashcards, use the flashcards online, test oneself, and play a matching game called Jewels of Wisdom. Brainscape is a flashcard creation application and a web and mobile education platform that allows students and educators to use existing flashcards, create new flashcards, gamify them, and share them.

We used Brainscape to develop the flashcards in Phase 1. A sample flashcard created using the Brainscape app is shown in Figure 1. Similarly, the other flashcards were produced. As seen in Figure 2, the multi-colored number values on the bottom of the card represent the confidence rating of the

TABLE II: Free Flashcards Apps

Apps	URL
Canva	https://www.canva.com/create/flashcards/
Cram	https://www.cram.com/
Brainscape	https://www.brainscape.com/

learner. The learner selects a number based on self-evaluation of how well the learner knows the answer. If the learner selects 5 (rated as perfect to the "How well did you know this" question), the respective card is removed from the collection of flashcards that appear before the learner during their learning process (also known as study mode). After selecting the confidence rating, the learner is presented with the answer to the question. Flashcards can be used as an individual learning strategy or with a partner to make them more of a collaborative learning strategy.

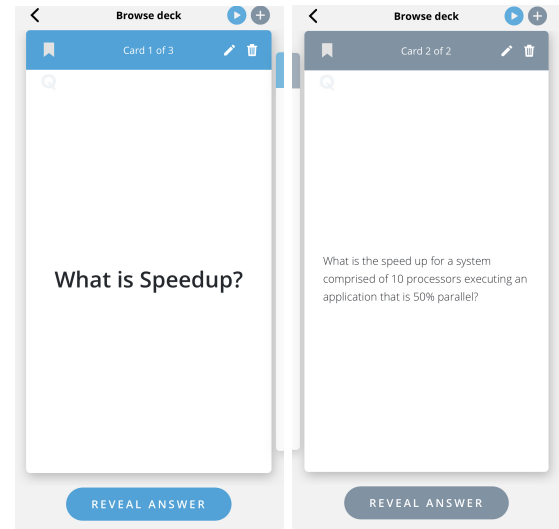


Fig. 1: Illustration of two questions seen on the flashcards before the answer is revealed.

B. Phase 2 - PDC Quest Card Game

In this phase, we present a card game called PDC Quest which includes 80 cards (questions) based on the eight PDC categories listed in Table I. The cards are numbered and coded, with each category having a distinctive number (1-8) and a two-letter abbreviation, as displayed in Table I. Each category will have ten cards (questions). PDC Quest could be played after PDC concepts are covered from one or more categories or before a quiz or exam on the PDC concepts. The game is meant to be played in groups of three. Certainly, the game can be modified for more than three players. However, the following description is based on three people playing. Based on the flashcards from the first phase, a single deck of 80 cards is created as the Deal stack. An example of a card is shown in Figure 3. Note that the front of the card has the name of the game, the PDC category number, and the code. The back of the card has the question and answer. Below the

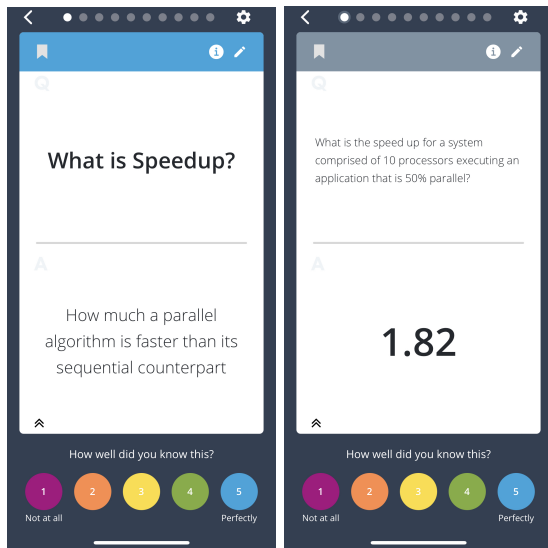


Fig. 2: Illustration of flashcards with revealed answers to the respective questions

answer is a QR code providing a more in-depth explanation of the topic. The explanation can help the evaluators rate the answer. Before the game begins, the category or categories of cards to be reviewed should be pulled from the main deck and placed in the Deal stack. Ensure that the Deal stack is thoroughly shuffled. To begin, Player 1 selects a card from the Deal stack and passes it to Player 2. Player 2 reads the question aloud, waiting for Player 1 to answer. Player 1 has 60 seconds to provide an answer. Upon Player 1 answering the question, Players 2 and 3 evaluate and rate the answer given by Player 1 by assigning a score of 0 for no answer or incorrect answer, 1 for a partially correct answer, and a score of 2 for a complete and correct answer. The evaluation process could involve examining a more comprehensive description of the subject matter via the QR code, prompting a discussion among the players and contributing to their learning experience. The score for Player 1 should be recorded on the scoresheet, and the card is placed in another stack called the Played stack.

The game continues as Player 2 draws from the Deal stack and gives the card to Player 3 to read aloud, and Player 2 answers the question. Players 1 and 3 evaluate the answer, assign a score by documenting it on the scoresheet, and place the card played in the Played stack. The game continues in this manner until no more cards are left in the Deal stack. The scoresheet is kept, which indicates the players' names, categories, and scores received. An example of the scoresheet is shown in the table in Figure 4. At the end of the game, players average their scores from each category by taking the total score from each category and dividing it by the number of cards played in that category. The scoresheet can act as a gauge of revealing what categories the player needs to study more. The maximum number of points a learner can get is equal to 2 multiplied by the total number of cards played. The maximum average score for each category is 2. As noted

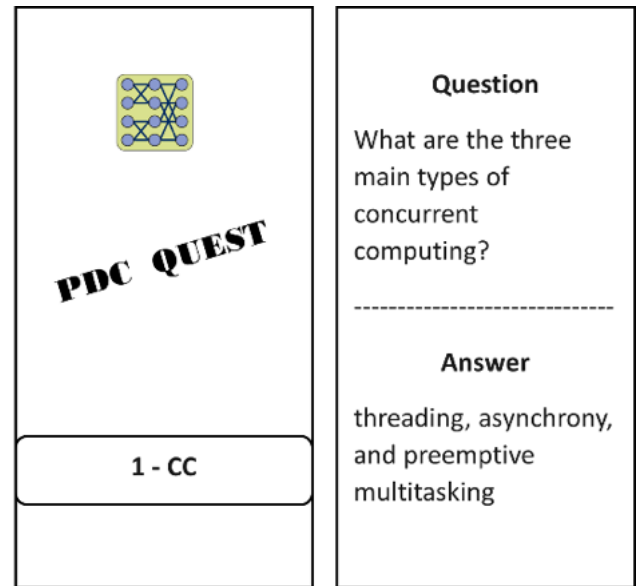


Fig. 3: PDC Quest Sample Card

in Figure 4, a categorization of the learner is as follows:

- Mastered Understanding: If a learner's average score is between 1.8-2.0
- Good Understanding: If a learner's average score is between 1.6-1.78
- Average Understanding: If a learner's average score is between 1.4-1.58

SAMPLE SCORESHEET			
PDC Quest Scoresheet			
Score (0- no or incorrect answer, 1- partially correct answer, 2- complete & correct answer)			
Average score for each category (total score for each category/number of cards played in that category)			
Average Score:			
Mastered Understanding: 1.8 and above			
Good understanding: 1.6-1.78,			
Average understanding: 1.4-1.58			
Category	Player's Name 1	Player's Name 2	Player's Name 3
Avg 1-CC			
Avg 2-ME			
Avg 3-CM			
Avg 4-MP			
Avg 5-SM			
Avg 6-SC			
Avg 7-SL			
Avg 8-SA			

Fig. 4: Sample Scoresheet

The percentage of students categorized as Mastered or Good can be used to measure the success rate of students learning the targeted PDC topics.

IV. BENEFITS AND LIMITATIONS

As reviewed in this work, the benefits of using active learning tools such as flashcards, and gamification of learning using tools such as the PDC Quest card game, can promote student engagement and the process of active recall, engage

metacognition, and confidence-based repetition. In addition, the utilization of such active learning tools as a supplement to other traditional teaching methods can improve student learning. In our earlier work, we have established a proof of concept for employing unplugged activities as a supplement to traditional teaching methods to increase student engagement toward learning complex PDC concepts [30]. Development of the PDC Quest card game is a step forward towards building a comprehensive set of unplugged activities to promote active learning when incorporated within a traditional teaching environment.

One of the key challenges in our research has been to identify the undergraduate CS courses for implementing the PDC Quest card game to enhance the learning of PDC concepts. It is our understanding that the developed game activity can only apply to a subset of CS courses in a traditional CS undergraduate curriculum. The PDC categories integrated into the card game may differ across courses depending on the course level. For example, in a freshman-level CS course, such as a CS0 or CS1 course, the instructor could use only the cards from the Concurrency and Speedup categories. However, for a junior or senior-level CS course, such as a systems course, cards from all the PDC categories can be used in the game. Another identified key challenge for the implementation of the course is the scalability of the game with the class size as the number of cards remains limited. A potential solution to this challenge is to actively engage most or all students in a small/medium sized class. For larger class sizes the game could be demonstrated with a smaller group of volunteering students in class and then assigned as a group study activity that students can work on their own. Presently, the primary limitation of this research is that the flashcards and the proposed PDC Quest card game are at a conceptual stage and have yet to be administered in a CS course to evaluate if these tools can help in better learning of complex PDC concepts. Our proposed flashcards and the PDC Quest card game is intended to be employed in two different levels of CS courses in the Fall 2023 semester. The plan is to introduce the card game towards the second half of the semester after the respective theoretical PDC concepts have been introduced to the students. Further, we will use the ASPECT survey tool [29] as it was utilized in our prior work in [30] to evaluate the effectiveness of the PDC Quest card game in improving student engagement and learning. The outcomes of the implementation of the PDC Quest card game will be reported in a future paper.

V. CONCLUSIONS AND FUTURE WORK

As researched in this work, PDC concepts need to be introduced early in undergraduate CS programs. PDC is a complex subject to understand, and there are fundamental concepts to learn before programming in parallel. In addition to theoretical and programming-based teaching, the flashcards and the PDC Quest card game developed in this work are meant to be used as active learning tools for reviewing and learning fundamental concepts. Although these tools are not meant to replace textbooks or traditional PDC lectures,

they are intended to be valuable supplemental material for reviewing complex concepts. In addition to the benefits of using the proposed flashcards and the PDC Quest card game, we have reported current challenges and limitations of our work in Section IV. We plan to address the challenges and limitations as we complete the design of the flashcards and the PDC Quest card game before the beginning of the Fall 2023 semester. We plan on using the proposed tools as active learning tools in CS 1, CS 2, CS 3, or introduction to PDC course(s) in the Fall 2023 semester. We will use Wiggin's et. al. ASPECT survey [29] to measure student engagement after each learning tool is used. The ASPECT survey includes 16 questions measuring student engagement with questions related to three constructs (value of the activity, instructor contribution, and personal effort) [29]. The measured results and outcomes of the implementation of the proposed active learning tools will be reported in a future paper.

ACKNOWLEDGMENT

The authors would like to thank the members of the National Science Foundation Center for Parallel and Distributed Computing Curriculum Development and Educational Resources PDC Curriculum Early Adopter Grant and Summer Training Program for their guidance through the initial phase of our research implementation. The authors would also like to thank the iPDC Summer Institute at Tennessee Technological Institute for their training and support.

REFERENCES

- [1] J. D. Karpicke and H. L. Roediger, "The critical importance of retrieval for learning," in *Science*, 2008, 319(5865), pp. 966-968. doi: 10.1126/science.1152408
- [2] J. D. Karpicke and J. R. Blunt, "Retrieval practice produces more learning than elaborative studying with concept mapping," in *Science*, 2011, 331(6018), pp. 772-775. <https://doi.org/10.1126/science.1199327>
- [3] E. R. Lai, "Motivation: A Literature Review Research Report," 2011, http://images.pearsonassessments.com/images/tmrs/Motivation_Review_final.pdf
- [4] Tabibian, Behzad et al., "Enhancing human learning via spaced repetition optimization," in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116,10 (2019), pp. 3988-3993. doi:10.1073/pnas.1815156116
- [5] M. D. Cohen, "The power of parallel thinking," in *Journal of Economic Behavior and Organization*, vol. 2, no. 4, pp. 285- 306, 1981.
- [6] J. Mullen, C. Byun, V. Gadepally, S. Samsi, A. Reuther, and J. Kepner, "Learning by doing, high-performance computing education in the mooc era," in *Journal of Parallel and Distributed Computing*, vol. 105, pp. 105-115, 2017.
- [7] H. Lin, "Teaching parallel and distributed computing using a cluster computing portal," in *Parallel and Distributed Processing Symposium Workshops and Ph.D. Forum (IPDPSW)*, 2013 IEEE 27th International. IEEE, 2013, pp. 1312-1317.
- [8] R. Brown and E. Shoop, "Csinparallel and synergy for rapid incremental addition of pdc into cs curricula," in *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops PhD Forum*, 2012.
- [9] D. W. Juedes and F. Drews, "Engineering a new curriculum: Experiences at ohio university in incorporating the ieee-tcpp curriculum initiative during a transition to semesters," in *Parallel and Distributed Processing Symposium Workshops and PhD Forum (IPDPSW)*, 2012 IEEE 26th International. IEEE, 2012, pp. 1279-1282.
- [10] S. K. Prasad, A. Gupta, K. Kant, A. Lumsdaine, D. Padua, Y. Robert, A. Rosenberg, A. Sussman, C. Weems et al., "Literacy for all in parallel and distributed computing: guide-lines for an undergraduate core curriculum," in *CSI Journal of Computing*, vol. 1, no. 2, pp. 82-95, 2012.
- [11] R. Brown and E. Shoop, "Modules in community: injecting more parallelism into computer science curricula," in *Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM, 2011, pp. 447-452.
- [12] D. J. John and S. J. Thomas, "Parallel and distributed computing across the computer science curriculum," in *Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2014 IEEE International. IEEE, 2014, pp. 1085-1090
- [13] W. B. Gardner, "Should we be teaching parallel programming?" in *Proceedings of the 22Nd Western Canadian Conference on Computing Education*, ser. WCCCE '17, 2017.
- [14] S. J. Matthews, "Teaching with parallella: a first look in an undergraduate parallel computing course," *Journal of Computing Sciences in Colleges*, vol. 31, no. 3, pp. 18-27, 2016.
- [15] S. K. Prasad, A. Gupta, A. Rosenberg, A. Sussman, and C. Weems, *Topics in Parallel and Distributed Computing: Introducing Concurrency in Undergraduate Courses*, 1st ed. Morgan Kaufmann, August, 2015.
- [16] S. K. Prasad, A. Gupta, A. Rosenberg, A. Sussman, and C. Weems, *Topics in Parallel and Distributed Computing: Enhancing the Undergraduate Curriculum: Performance, Concurrency, and Programming on Modern Platforms*, 2nd ed. Springer International Publishing, 2018.
- [17] S. Bogaerts, K. Burke, B. Shelburne, and E. Stahlberg, "Concurrency and parallelism as a medium for computer science concepts," in *Curricula for Concurrency and Parallelism workshop at Systems, Programming, Languages, and Applications: Software for Humanity*, 2010.
- [18] S. Bogaerts, "Hands-on exploration of parallelism for absolute beginners with scratch," in *2013 IEEE International Symposium on Parallel and Distributed Processing, Workshops and Ph.D. Forum*. IEEE, 2013, pp. 1263-1268.
- [19] S. A. Bogaerts, "Limited time and experience: Parallelism in cs1," in *Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2014 IEEE International. IEEE, 2014, pp. 1071-1078.
- [20] S. A. Bogaerts, "One step at a time: Parallelism in an introductory programming course," in *Journal of Parallel and Distributed Computing*, vol. 105, pp. 4-17, 2017.
- [21] E. Saule, "Experiences on teaching parallel and distributed computing for undergraduates," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2018.
- [22] A. Branco, A. L. De Moura, N. Rodriguez, and S. Rossetto, "Teaching concurrent and distributed computing—initiatives in Rio de Janeiro," in *Parallel and Distributed Processing Symposium Workshops and Ph.D. Forum (IPDPSW)*, 2013 IEEE 27th International. IEEE, 2013, pp. 1318-1323.
- [23] D. Grossman and R. E. Anderson, "Introducing parallelism and concurrency in the data structures course," in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. ACM, 2012, pp. 505-510.
- [24] M. Johnson, R. H. Liao, A. Rasmussen, R. Sridharan, D. D. Garcia, and B. Harvey, "Infusing parallelism into introductory computer science curriculum using Mapreduce," EECS Department, University of California, Berkeley, Tech. Rep, 2008.
- [25] S. V. Moore and S. R. Dunlop, "A flipped classroom approach to teaching concurrency and parallelism," in *Parallel and Distributed Processing Symposium Workshops*, 2016 IEEE International. IEEE, 2016, pp. 987-995.
- [26] S. K. Prasad, A. Y. Chtchelkanova, S. K. Das, F. Dehne, M. G. Gouda, A. Gupta, J. Jaja, K. Kant, A. La Salle, R. LeBlanc, et al., "NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing: core topics for undergraduates," in *SIGCSE*, vol. 11, 2011, pp. 617-618.
- [27] Prince, M., "Does Active Learning Work? A Review of the Research," in *Journal of Engineering Education*, 2004, 93: 223-231. <https://doi.org/10.1002/j.2168-9830.2004.tb00809.x>
- [28] Freeman, S., et al., "Active learning increases student performance in science, engineering, and mathematics," in *Proceedings of the National Academy of Sciences*, 2014, 111(23), 8410-8415. <https://www.pnas.org/doi/full/10.1073/pnas.1319030111>
- [29] B. L. Wiggins, S. L. Eddy, L. Wener-Fligner, K. Freisem, D. Z. Grunspan, E. J. Theobald, J. Timbrook, and A. J. Crowe, "Aspect: A survey to assess student perspective of engagement in an active-learning classroom," *CBELife Sciences Education*, vol. 16, no. 2, p. ar32, 2
- [30] S. Srivastava, M. Smith, A. Ghimire and S. Gao, "Assessing the Integration of Parallel and Distributed Computing in Early Undergraduate Computer Science Curriculum using Unplugged Activities," *2019 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC)*, Denver, CO, USA, 2019, pp. 17-24, doi: 10.1109/EduHPC49559.2019.00008.