Assessing the Integration of Parallel and Distributed Computing in Early Undergraduate Computer Science Curriculum using Unplugged Activities

Srishti Srivastava*, Mary L. Smith†, Amrita Ghimire‡, Shaun Gao§

*Computer Science
University of Southern Indiana
Evansville, IN, USA
fsrishti@usi.edu

†Computer Science and Eng.
Hawaii Pacific University
Honolulu, HI, USA
†*mlsmith@hpu.edu*

‡Computer Science and Eng.
Mississippi State University
Mississippi State, MS, USA
ghimire.ammi@gmail.com

§Computer Science
University of Richmond
Richmond, VA, USA
sgao@richmond.edu

***Abstract*—Due to the pervasive presence of parallel and distributed computing (PDC) in most of the application domains, the need for PDC skills in computing professionals is necessary. This has inspired much interest in including PDC topics in existing computer science (CS) undergraduate curriculum, as evidenced in the ACM/IEEE joint curriculum recommendations. We address this gap by developing new active learning style PDC activities. The application and evaluation of the unplugged activities are described in this paper. The activities highlight key benefits of parallel computing, and the appropriate application of important PDC concepts. An assessment of student engagement has been utilized to measure the effectiveness of our active learning PDC modules. We determine whether the engagement differs by gender, age, and class standing (freshman, sophomore, junior, senior). Results indicated there were slight differences with respect to age and class standing. This research contributes to helping other practitioners in the CS community by providing PDC lesson modules to be integrated into their existing courses. The developed activities and the related lesson plans are intended to provide easier learning of the fundamental PDC concepts in an early undergraduate CS curriculum.**

***Keywords*-parallel computing; active learning; student engagement; unplugged activity**

I. Introduction

Parallel and distributed computing (PDC) has conformed into a pervasive way of computing in research and industries of many science and engineering domains. This imposes the need for PDC skills in computing professionals and necessitates that every programmer understands how parallel processing affects problem-solving. As evidenced in the ACM/IEEE joint curriculum recommendations (2013) [1], it is suggested to incorporate PDC fundamentals at various levels within the existing undergraduate level computer science (CS), computer engineering, and other related computational science and engineering curricula. In this study, two unplugged activities have been used for introducing PDC topics in CS0, CS1, CS2, and CS3 level courses across four different universities. This work is a first research effort towards implementing the concepts learned at the national science foundation (NSF) center for parallel and distributed computing curriculum development and educational resources (CDER) PDC curriculum early adopter grant and summer training program, for integrating PDC concepts in traditional undergraduate level CS curricula. Student engagement has been identified as a popular metric for measuring the accountability of applied teaching methodologies [2]. To the best of our knowledge, this study is the first research effort that involves measurement of student engagement to evaluate the incorporation of the developed active learning based PDC modules into existing undergraduate level CS courses. In this paper, we present a study to evaluate student engagement with respect to demographic factors, such as gender, age, and class standing (freshman, sophomore, junior, senior), when the same active learning PDC teaching modules are used at four different universities.

The authors in this study have diverse levels of knowledge with respect to PDC concepts, which varies from being an expert in the field to having a very basic understanding of the PDC field. Being the first effort towards a collaborative pedagogical study of incorporating active learning PDC teaching modules across four universities, the analysis in this work is limited to a single group post test analysis. Moreover, to address the challenges of employing a validated data collection methodology, a pre-validated survey tool called ASPECT [2] has been utilized for measuring student engagement. Although, our work is focused on measuring student engagement, the implied learning outcomes for this study, which will be used in our future study on measuring student learning, are based on Bloom's taxonomy *Know the term* (K) level [3]. Our activities require the students to know the following concepts: (i) knowledge of why and what is parallel/distributed computing, (ii) knowledge of Amdahl's law, (iii) knowledge of parallel sorting, (iv) knowledge of scheduling and load balancing fundamentals of PDC.

Key Novel Contributions:
1. Employing an active learning based PDC teaching module that consists of a set of two unplugged PDC activities for

introducing PDC concepts in existing undergraduate level CS courses at four universities.
2. Using ASPECT [2], a pre-validated survey tool, for measuring student engagement towards the incorporated PDC modules in existing CS courses at four different universities.
3. Using statistical analysis to study the impact of age, gender, and class standing (freshman, sophomore, junior, or senior) on student engagement towards the active learning PDC modules.

A more detailed overview of the background and related work is given in Section II. The research plan and design are described in Section III. Data collection methods, results, and the analysis are explained in Section IV. Conclusions and directions towards future work are listed in Section V.

II. Background and Related Work

Parallel and distributed computing has become a significant topic of research and educational interest in CS and other STEM disciplines, mostly in areas of applications of computational science and engineering for several years. Cohen [4] has pointed out "the power of parallel thinking" and has stressed on promoting a parallel structure for processing information. However, not every CS professional is familiar with PDC, and have not been exposed to a parallel thinking approach. In general, PDC envelopes areas such as distributed systems, parallel computational hardware, high-performance computing, and others. However, recent CS graduates, who are the leading workforce for current research and industries, are not versed with the fundamental knowledge of PDC. An online course available as a MOOC by Mullen et.al., [5] 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 in teaching PDC courses in some universities as an elective CS course. For example, work presented by Lin in [6] provides a fair evaluation of teaching effectiveness and discusses the survey as their data collection and the related test results. Several other literature, such as the work presented in [7][8][9][10][11][12][13][14], report introducing PDC via CS courses taught at the junior, senior, and graduate level.

Furthermore, considering the importance of including PDC in undergraduate CS education, the ACM/IEEE-CS 2013 curriculum report [1] recommends computer science educators to incorporate PDC topics in CS undergraduate curriculum as one of the core topics. In addition, the NSF/IEEE-TCPP curriculum initiative on PDC has been contributing in various ways, such as outlining recommendations for covering important PDC topics [15], conducting training workshops for educators to teach PDC at introductory level CS courses. Much work has also been reported on integration of PDC topics throughout the CS undergraduate curriculum through the use of a modular approach [16][17][18][19][20][21]. In other work, authors are proposing possible teaching materials to infuse parallel thinking for undergraduate students and sharing teaching materials for PDC education [22][23][24][25]. Bogaerts [26][27][28][29] has worked on introducing parallelism in CS0 and CS1 level courses by including some hands-on in-class programming activities as a few additional lectures in existing CS courses. In addition, Rague [30] has adopted a similar concept of teaching PDC in lower level CS courses using a visualization tool to promote parallel thinking. There are other CS1 [31], CS2 [32][33], and a few higher level CS courses that have been used to introduce PDC concepts to CS majors and non-majors [34][35].

Overall, authors in some of the literature [36][32][28][29][10][11][37] explain that students struggle in understanding the theoretical concepts, as well as the parallel programming due to the complexity imposed by the inherent nature of PDC. Thus, indicating that there could be better approaches to teaching PDC topics to CS undergraduate students rather than simply introducing the topic via traditional lecture methods and/or programming-based teaching. As an alternative teaching technique, Moore et.al. present the use of a flipped classroom approach for teaching concurrency and parallelism in split level CS course [37]. An assessment using a multi-year (courses offered in the spring semester of 2014 and 2015) report comparing the achievement of learning outcomes and comparison of course evaluation is presented in their work. Furthermore, unplugged activities [38] have been popularly used for teaching important CS concepts to primary-aged students. Neeman et.al [39] also adopted a similar concept of using unplugged activities for introducing PDC concepts to non-major CS students. Kitchen et.al. [40], and Bogaerts [28] have used a game playing approach to introduce PDC concepts, where students play the role of processors, computational cores, and others. The concept of using an unplugged activity, where students act as processes has also been adopted in a tutorial style data structures course (CS2) [41]. To the best of our knowledge, there appears to be a gap in the literature about understanding students interest and perspectives of learning PDC topics. In order to bridge this gap, measuring student engagement towards employing in-class lectures, or unplugged activities, could help enforce the PDC education to grow in an active learning environment. Student engagement has been a topic of interest in CS education. Morgan et.al. [42] provide deeper insights on how CS academics view student engagement.

In this work, a collaborative research approach for teaching the basic concepts of PDC using active learning modules in existing undergraduate level CS courses (CS0, CS1, CS2, and CS3) at four different universities, is presented. Wiggins

et.al's, ASPECT survey tool [2] has been used to measure student engagement after implementing the teaching modules in class. The module activities were developed by the authors in this paper at the NSF CDER training program in July 2018.

III. Research Plan and Design

The research plan for the work presented herein stemmed from a team-based collaboration at the NSF CDER early adopter summer training program in July 2018. The main focus of this research work follows the direction towards achieving the goal set by the NSF CDER workshop, which is to ensure that every computer science student is exposed to PDC topics early in their undergraduate study. The authors of this work conducted weekly meetings throughout the semester of Fall 2018 to identify the most appropriate plan for implementing their research effort towards introducing PDC topics to students in undergraduate level CS courses at their respective universities. The research planning and design phase were broken down into a number of tasks described in detail in the following subsections.

A. Determining Relevant Courses

One of the key challenges in our research has been to identify the commonality among the undergraduate CS courses that the authors taught during the Fall 2018 semester at their respective universities. The NSF CDER early adopter summer training program focused on developing classroom activities for introducing PDC topics in CS0, CS1, or CS2 level courses. In addition, we also employed the same set of activities in a CS3 level course, at the University of Richmond, which will be used in our future study to evaluate any significant difference in student engagement in an upper level CS course. To remain on track with the learning from the training program, the authors decided to implement the same two unplugged activities (discussed in Subsection III-B) in one or more of their undergraduate level CS courses. Using the same set of teaching modules at different universities facilitated in analyzing the utility and the generality of the PDC modules across a range of courses in a traditional CS undergraduate curriculum. The CS courses in our study consisted of two CS0 level courses, titled CMSC 105-01: Elementary Programming, and CSE 1002: Introduction to Computer Science and Software Engineering, one CS1 level course, titled CSCI 2911: Computer Science I, two CS2 level courses titled CSCI 2912: Computer Science II, and CS 311: Data Structures and Analysis of Algorithms, and a CS3 level course titled, CMSC 395: Digital Image Processing.

B. Developing Unplugged Activities as PDC Teaching Modules

As a part of our training at the NSF CDER early adopter summer workshop, the authors in this work were grouped as a team to develop a few unplugged activities to integrate PDC fundamentals in existing beginner level CS courses. The authors decided to use two of these activities for introducing PDC topics, such as uniprocessor versus multiprocessor environments, Amdahl's law, parallel sorting, and others, in their respective courses. The activities are described in more detail as follows.

(I) *More Processors are Not Always the Best (MP)*: this activity is a composition of four different phases of workload distribution. It involves the participation of two or more students for the four different execution scenarios. In all the phases, each participant is given their own sheet of paper that represents their own share of local memory. The four activity phases are described as follows.
(i) Uniprocessor Sequential phase: Participant 1 acts as a processor to perform Task 1, which is to write a sentence (more processors are not always the best) letter by letter, followed by Task 2, which is assigning a numeric value $(0, 1, \cdots, 32)$ as an index to each character in the written sentence. Participant 2 times the tasks performed by Participant 1. The sequential time is calculated as the time elapsed between allocation of work to participant 1, and the completion of Task 2.
(ii) Uniprocessor Multi-Tasking Phase: Participant 1 acts as a processor to perform Task 1, which is writing the same sentence (as in the sequential phase) character by character, interleaved with Task 2, which is assigning a numeric value $(0, 1, \cdots, 32)$ as an index to each character in the written sentence. Participant 2 times the tasks performed by Participant 1 and compares the performance of this phase with that of the sequential phase.
(iii) Two Processor Parallel Phase: Participant 1 acts as a parallel processor to perform Task 1, which is writing the same sentence as in the prior activity phases. At the same time Participant 2 acts as a second parallel processor to perform Task 2, which is assigning a numeric value $(0, 1, \cdots, 32)$ as an index to each character written by Participant 1. The two processors are writing on their own sheets of paper. Participant 3 times the tasks performed by parallel participants 1 and 2 and compares the performance of all the phases executed thus far.
(iv) Multi-Processor Parallel Phase: Participant 1 acts as a parallel processor executing the first half of Task 1, which is writing a partial sentence (more processors are). Participant 2 acts as another parallel processor executing the second half of Task 1, which is writing a partial sentence (not always the best). Participant 3 acts as another parallel processor executing the first half of Task 2, which is assigning a numeric value as an index (starting at 0) to each character in the partial sentence written by Participant 1. Participant 4 acts as another parallel processor executing the second half of Task 2, which is assigning a numeric value (continues from participant 3) as an index to each character in the partial sentence written by Participant 2. Participant 5 times

the tasks performed by parallel participants and compares the performance of all the phases executed thus far.

At the end of the four phases of the MP activity, class time is allocated for reflecting on the learning of the PDC concepts implemented via this activity. The students observe and understand the benefit of parallel processing in phase 3, which in most cases yielded a faster execution of parallel tasks. However, in the last phase a performance degradation in terms of increased parallel execution time is observed. As a part of students reflection, the performance degradation in the last activity phase is explained using Amdahls law as the underlying PDC concept.

(II) *Penny Sorting Exercise (PE)*: in this activity, the problem is to find the oldest penny from a pile of up to 50 pennies in the quickest time. A number of bags containing 50 pennies each were created by the instructors (the authors in this work) to allocate to each group of students in their classrooms. Each group consisted of five to six students. The activity involves sorting through the pennies in a sequential mode, followed by various parallel variations. The different sequential and parallel execution phases are described as follows.
(i) Phase 1 (sequential): in each group, Student 1 performs sorting through a single bag of 50 pennies to find the oldest penny, Student 2 times the work of Student 1, and the other students act as observers.
(ii) Phase 2 (parallelize it): in each group, Students 1 and Student 2 are given equal number of pennies (a bag of 25 pennies per student) and are required to find the oldest penny in their allocated pile. Student 3 records the processing times for students 1 and 2. Parallel Time is calculated as the time of the slowest student among 1 and 2.
(iii) Phase 3 (parallelize it-2): in each group, from a bag of 50 pennies, the instructor allocates two pennies to students 1, 2, 3, and 4, and the remaining pennies to student 5. At a set time, sorting for the oldest penny begins for all groups. Within a group, students are required to communicate with each other to produce the oldest penny for their respective groups. Student 6 records time for each of the worker students. Parallel time is calculated as the time of the slowest student.

At the end of the activity, part of the class time is used as reflection time to discuss the PDC topics, such as scheduling and load balancing, learned in this activity.

C. Employing the Unplugged Activities in Class for Collecting Research Results

The main goal of employing the two unplugged activities as active learning PDC teaching modules was to collect data in the form of student surveys to measure student engagement. The authors used the pre-validated ASPECT survey [2] as the data collection instrument for this research work. Student engagement has been used as the evaluation metric for measuring the effectiveness of implementing the PDC modules in the form of two unplugged activities in undergraduate level CS course. A more detailed description of the ASPECT survey and its usage for data collection, along with a more detailed analysis of results is given in the following section.

IV. Research and Analysis

Undergraduate CS students from four different universities in CS0, CS1, CS2, and CS3 level courses were introduced to two unplugged PDC activities and were surveyed to evaluate student engagement. Being the first collaborative effort towards implementing active learning PDC modules, this research is limited to a single group post-test analysis. The Wiggin's et. al. ASPECT survey was used to measure student engagement after each unplugged activity [2]. The ASPECT survey includes 16 questions (presented in Table VI) measuring student engagement with questions related to three constructs (value of the activity, instructor contribution, and personal effort). Demographic questions, such as gender, age, and class standing (freshman, sophomore, junior, and senior) were added to the survey to determine if there were differences between the demographic data and the constructs of student engagement. Students in these courses were asked to participate in the study and were given extra credit for doing so. For the PE activity, there were a total of 98 participants from the author's respective universities. For the MP activity there were a total of 102 participants from the same universities. General descriptive statistics are presented in Subsection IV-A for both activity groups. In addition, a detailed analysis is presented for each activity (PE and MP) in Subsections IV-C1 and IV-C2.

A. Descriptive Statistics

Tables I, II, and III present the frequencies of the primary demographic variables (gender, age group, and class standing) used in the analysis.

Gender	PE	MP
Male	73	77
Female	25	25
Total	98	102

Table I: Gender Frequency during PE and MP Activities

Age	PE	MP
18-22	80	85
23-27	11	9
28-32	3	4
33 and over	4	4
Total	98	102

Table II: Age Frequency during PE and MP Activities

Class Standing	PE	MP
Freshman	38	29
Sophomore	18	29
Junior	25	26
Senior	17	18
Total	98	102

Table III: Frequency of Class Standing during PE and MP Activities

Construct	PE	MP
Instrument	0.964	0.952
Value of the Activity	0.938	0.917
Instructor Contribution	0.938	0.917
Personal Effort	0.921	0.865

Table IV: Reliability Analysis of the ASPECT Instrument

B. Measure Validation and Reliability

To measure the validity of the ASPECT instrument a confirmatory factor analysis with varimax rotation of student responses was conducted. The results confirmed the findings in Wiggin's, et al., 2017 paper [2]. The three constructs (value of the activity, instructor contribution and personal effort) were confirmed towards an accurate measure of student engagement. In addition, the measure was tested for reliability and internal consistency using Cronbach's Alpha. Both the measure and each of the constructs were tested. The results of the reliability analysis are outlined in Table IV.

C. Activity Analysis

The intent of the study was to determine if there were any statistically significant differences between the demographic factors gender, age, and class standing and the three student engagement constructs (value of the activity, instructor contribution, and personal effort). The preferred method to test this would be a three-way Analysis of Variance (ANOVA) which would allow testing the main effects of the three variables while holding the others constant along with testing for multi-variate interactions. However, due to the small number of cases in each of the variable cells it was decided to use a one-way and two-way ANOVA to test age and class standing on each construct while using a t-test on gender and each construct. Table VII summarizes whether there was a statistical difference found between each demographic factor and student engagement construct. A more detailed analysis is discussed below.

1) PE Activity Analysis: The results of the two-way ANOVA indicated a significant difference in student perception of the value of group activity construct based on the age demographic factor. A least significant difference (LSD) test was used to determine specific group differences. The differences found were between age groups 18-22 and 23-27 (18-22, N (sample size) = 79, Mean = 37.71, 23-27, N = 11, Mean = 43.91), where the 23-27 age group provided a higher rating to the value of group activity. The two-way ANOVA test of the value of group activity and class standing showed no significant differences. The t-test of differences between gender and value of group activity also showed no significant differences. The results of the ANOVA to determine the differences between instructor contribution and age showed a significant difference. An LSD Post Hoc was conducted to determine the specific differences. The results showed that age groups 18-22 and 23-27 were significantly different (18-22, $N = 80$, Mean = 21.78, 23-27, $N = 11$, Mean = 24.55), where the 23-27 age group provided a higher rating to the instructor contribution. There were no significant differences found with any of the demographic factors and the personal effort construct.

2) MP Activity Analysis: a two-way ANOVA was used to test the differences between age, class standing, and instructor contribution and a significant main effect was found between class standing with high statistical power (0.781). An LSD Post Hoc test was conducted to determine the specific factors. The one-way ANOVA was used to

Questions	Answer Choices
Gender	Male / Female
Age	18-20 / 23-27 / 28-32 / 33 and over
School Attending	Hawaii Pacific University / Mississippi State University / University of Southern Indiana / University of Richmond
Class Standing	Freshen / Sophomore / Junior / Senior
Major	Computer Science / Engineering / Other
Course you are taking	CS 311 / CSE 1002 / CSCI 2911 / CSMC 105 / CSMC 395

Table V: List of Questions Added to the ASPECT Survey

Survey Questions
Value of the Activity
Explaining the material to my group improved my understanding of it.
Having the material explained to me by my group members improved my understanding of the material.
Group discussion during the"MP or PE" activity contributed to my understanding of parallel computing fundamentals.
I had fun during today's "MP or PE" activity.
Overall, the other members of my group made valuable contributions during the"MP or PE" activity.
I would prefer to take a class that includes this group activity over one that does not include this "MP or PE" group activity.
I am confident in my understanding of the material presented during today's "MP or PE" activity.
The "MP or PE" activity increased my understanding of the parallel computing fundamentals.
The "MP or PE" activity stimulated my interest in parallel computing.
Personal Effort
I made a valuable contribution to my group today.
I was focused during today's "MP or PE" activity.
I worked hard during today's "MP or PE" activity.
Instructor Contribution
The instructor seemed prepared for the "MP or PE" activity.
The instructor put a good deal of effort into my learning for today's class.
The instructor's enthusiasm made me more interested in the "MP or PE" activity.
The instructor and/or TAs were available to answer questions during the "MP or PE" activity.

Table VI: Survey Questions in the Pre-Validated Wiggin's et. al ASPECT Survey (measured in Likert scale, where 1 represents strongly disagree and 5 represents strongly agree)

test the differences between class standing and the three constructs. Significant differences were found between class standing and both the value of group activity construct and the instructor contribution construct. Post Hoc testing was done using the Tamhane's T2 multiple comparison test that is used to determine which of three or more means differ from one another. Post Hoc testing showed significant difference between the value of group activity construct, and the freshman and junior class standing factors. Juniors provided a higher rating to the value of the group activity construct (freshmen, $N = 29$, $M = 35.65$, junior, $N = 26$, $M = 40.73$). There was also a significant difference found between the instructor contribution construct and class standing, where juniors provided higher rating to the instructor contribution construct (freshman, $N = 29$, $M = 16.59$, junior, $N = 26$, $M = 19.19$). In addition, there was a significant difference between sophomores and juniors, where juniors rated the instructor contribution higher than sophomores (sophomore, $N = 29$, $M = 17.35$, junior, $N = 26$, $M = 19.19$). There was no significant difference found between the personal effort construct and any of the demographic factors. The analysis is summarized in Table VII, which illustrates the results of the statistical tests between each demographic factor and each student engagement construct for both activities.

Key Takeaways: the researchers acknowledge that the small sample size of the study may have affected the statistical analysis and a larger sample size may yield different results. Further studies are underway to evaluate the impact of demographic factors on the student engagement constructs with a larger data set for a more in-depth and sound statistical analysis. With respect to the PE activity, the statistical differences found (the value of the activity and the instructor contribution construct was rated higher by the 23-27 age group than the younger 18-22 age group) may imply that the younger group has limited college and professional experience and may not realize the importance of learning complex concepts in stages whereas, the slightly older students may have had more college and professional experience, thus realizing the value of learning in stages. The same reasoning may apply to the findings of the MP activity, where juniors rated the value of the activity higher than the freshmen students. Similar qualitative reasoning may be used when explaining why the 23-27 group rated the instructor contribution higher than the younger age group. The older age group may not have been exposed to active learning activities during their K-12 education. Whereas, the active learning approach has become more prevalent in K-12 education in the recent years and the younger group may have been more familiar to this learning approach, thus not considering it as valuable as the older age group.

	Demographic Factors		
Student Engagement Constructs	**Gender** (Male/Female)	**Age** (18-22, 23-27, 28-32, 33-over)	**Class Standing** (Freshmen, Sophomore, Junior, Senior)
Value of Activity	MP - No difference PE - No difference	MP - No difference **PE - Difference*** (18-22, 23-27)	**MP - Difference*** (Freshmen, Junior) PE - No difference
Instructor Contribution	MP - No difference PE - No difference	MP - No difference **PE - Difference*** (18-22, 23-27)	**MP - Difference*** (Freshmen, Junior) (Sophomore, Junior) PE - No difference
Personal Effort	MP - No difference PE - No difference	MP - No difference PE - No difference	MP - No difference PE - No difference
Activities: MP PE ***Statistical significant difference**			

Table VII: Summary of Statistical Analysis of Student Engagement Constructs and Factors

V. Conclusion and Future Work

In this paper, a first collaborative effort between four different universities towards establishing the proof of concept of employing active learning PDC teaching modules in existing undergraduate level CS courses, has been presented. Student engagement has been identified as the main metric for evaluation. The pre-validated ASPECT survey from the Wiggin's et. al. study [2] has been used to collect data from students from four different universities, who participated in the two unplugged PDC activities. The students' responses were analyzed statistically using t-tests, one-way ANOVA, and two-way ANOVA to determine whether there were any differences, and if any then to discover specific differences between gender, age, and class standing with respect to constructs used to define student engagement (value of activity, instructor contribution, and personal effort). Based on the analysis presented in Section IV, it can be concluded that gender has no significant effect on the value of the activity, instructor contribution, and personal effort across both activities. Age showed a significant effect on instructor contribution in the PE activity. For the MP activity, age and class standing showed a signicant effect on the value of the activity and instructor contribution.

The analysis in this work is limited to a single group post-test with a small sample size. In future work, the authors plan to continue collecting more data as well as extend the analysis to two group pre-test post-test analysis. The authors also plan to continue their research on developing additional unplugged and programming-based teaching modules to introduce PDC topics in existing undergraduate CS courses at their respective universities to improve student engagement. Based on the work presented here, which is a proof of concept for employing unplugged activities to increase student engagement towards learning PDC concepts, the authors also plan to include more qualitative data analysis with a larger data sample and add other measures (such as instructor and student expectation) for a broader assessment of student engagement. In addition, for a comprehensive comparative analysis, the authors also plan to use the developed active learning PDC teaching modules to measure learning effectiveness in undergraduate level CS courses at

the authors respective universities.

ACKNOWLEDGMENT

The authors would like to thank the organizing and the program committee 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 generous support and guidance through the initial phase of our research implementation. The authors would also like to thank iPDC Summer Institute at Tennessee Technological Institute for their training and support. In addition, the authors would like to thank Dr. Brett A. Becker, Assistant Professor at the School of Computer Science at University College, Dublin, Ireland, for his valuable contribution in providing us his ideas on the penny activity and Dr. Ken Rossi, Assistant Professor, College of Business at Hawaii Pacific University for his statistical consultation.

REFERENCES

[1] A. f. C. M. A. Joint Task Force on Computing Curricula and I. C. Society, *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. New York, NY, USA: ACM, 2013, 999133.

[2] 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, 2017.

[3] B. S. Bloom *et al.*, "Taxonomy of educational objectives. vol. 1: Cognitive domain," *New York: McKay*, pp. 20–24, 1956.

[4] M. D. Cohen, "The power of parallel thinking," *Journal of Economic Behavior & Organization*, vol. 2, no. 4, pp. 285–306, 1981.

[5] J. Mullen, C. Byun, V. Gadepally, S. Samsi, A. Reuther, and J. Kepner, "Learning by doing, high performance computing education in the mooc era," *Journal of Parallel and Distributed Computing*, vol. 105, pp. 105–115, 2017.

[6] H. Lin, "Teaching parallel and distributed computing using a cluster computing portal," in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*. IEEE, 2013, pp. 1312–1317.

[7] J. Liu, "20 years of teaching parallel processing to computer science seniors," in *Proceedings of the Workshop on Education for High Performance Computing*. IEEE Press, 2016, pp. 7–13.

[8] J. Eckroth", "A course on big data analytics," *Journal of Parallel and Distributed Computing*, vol. 118, pp. 166 – 176, 2018.

[9] A. Yazici, A. Mishra, and Z. Karakaya, "Teaching parallel computing concepts using real-life applications," *International Journal of Engineering Education*, vol. 32, no. 2, pp. 772–781, 2016.

[10] E. Saule, "Experiences on teaching parallel and distributed computing for undergraduates," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2018.

[11] 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 & PhD Forum (IPDPSW), 2013 IEEE 27th International*. IEEE, 2013, pp. 1318–1323.

[12] A. Marowka, "Think parallel: Teaching parallel programming today," *IEEE Distributed Systems Online*, vol. 9, no. 8, 2008.

[13] A. Shafi, A. Akhtar, A. Javed, and B. Carpenter, "Teaching parallel programming using java," in *Proceedings of the Workshop on Education for High-Performance Computing*, ser. EduHPC '14, 2014, pp. 56–63.

[14] E. Cesar, A. Cortés, A. Espinosa, T. Margalef, J. C. Moure, A. Sikora, and R. Suppi, "Introducing computational thinking, parallel programming and performance engineering in interdisciplinary studies," *Journal of Parallel and Distributed Computing*, vol. 105, pp. 116–126, 2017.

[15] 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.

[16] 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.

[17] 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 & PhD Forum (IPDPSW), 2012 IEEE 26th International*. IEEE, 2012, pp. 1279–1282.

[18] 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: guidelines for an undergraduate core curriculum," *CSI Journal of Computing*, vol. 1, no. 2, pp. 82–95, 2012.

[19] 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.

[20] D. J. John and S. J. Thomas, "Parallel and distributed computing across the computer science curriculum," in *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*. IEEE, 2014, pp. 1085–1090.

[21] W. B. Gardner, "Should we be teaching parallel programming?" in *Proceedings of the 22Nd Western Canadian Conference on Computing Education*, ser. WCCCE '17, 2017.

[22] 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.

[23] 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.

[24] ——, *Topics in Parallel and Distributed Computing: Enhancing the Undergraduate Curriculum: Performance, Concurrency, and Programming on Modern Platforms*, 2nd ed. Springer International Publishing, 2018.

[25] D. Valentine, "Monte carlo simulations: Parallelism in cs1/cs2."

[26] 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.

[27] S. Bogaerts, "Hands-on exploration of parallelism for absolute beginners with scratch," in *2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*. IEEE, 2013, pp. 1263–1268.

[28] S. A. Bogaerts, "Limited time and experience: Parallelism in cs1," in *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*. IEEE, 2014, pp. 1071–1078.

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

[30] B. Rague, "Teaching parallel thinking to the next generation of programmers," *Journal of Education, Informatics and Cybernetics*, vol. 1, no. 1, pp. 43–48, 2009.

[31] K. B. Bruce, A. Danyluk, and T. Murtagh, "Introducing concurrency in cs 1," in *Proceedings of the 41st ACM technical symposium on Computer science education*. ACM, 2010, pp. 224–228.

[32] 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.

[33] M. Grossman, M. Aziz, H. Chi, A. Tibrewal, S. Imam, and V. Sarkar, "Pedagogy and tools for teaching parallel computing at the sophomore undergraduate level," *Journal of Parallel and Distributed Computing*, vol. 105, pp. 18–30, 2017.

[34] R. Brown, E. Shoop, J. Adams, C. Clifton, M. Gardner, M. Haupt, and P. Hinsbeeck, "Strategies for preparing computer science students for the multicore world," in *Proceedings of the 2010 ITiCSE working group reports*. ACM, 2010, pp. 97–115.

[35] 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.

[36] 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.

[37] 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.

[38] T. Bell, I. H. Witten, and M. Fellows, "Cs unplugged: an enrichment and extension programme for primary-aged students.(2015)," *Retrieved October*, vol. 18, p. 2017, 2015.

[39] H. Neeman, L. Lee, J. Mullen, and G. Newman, "Analogies for teaching parallel computing to inexperienced programmers," in *ACM SIGCSE Bulletin*, vol. 38, no. 4. ACM, 2006, pp. 64–67.

[40] A. T. Kitchen, N. C. Schaller, and P. T. Tymann, "Game playing as a technique for teaching parallel computing concepts," *ACM SIGCSE Bulletin*, vol. 24, no. 3, pp. 35–38, 1992.

[41] B. R. Maxim, G. Bachelis, D. James, and Q. Stout, "Introducing parallel algorithms in undergraduate computer science courses (tutorial session)," *ACM SIGCSE Bulletin*, vol. 22, no. 1, p. 255, 1990.

[42] M. Morgan, M. Butler, N. Thota, and J. Sinclair, "How cs academics view student engagement," in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE 2018, 2018.