Unplugged Activities to Introduce Parallel Computing in Introductory Programming Classes: an Experience Report

Sheikh K. Ghafoor
sghafoor@tntech.edu
Tennessee Tech University

David W. Brown
dwbrown@tntech.edu
Tennessee Tech University

Mike Rogers
mrogers@tntech.edu
Tennessee Tech University

Thomas Hines
tmhines42@students.tntech.edu
Tennessee Tech University

ABSTRACT

Learning programming in early introductory classes is challenging for first year university students, and introducing parallel programming (PDC) in early classes along with traditional sequential programming is even more challenging. Unplugged activities may help alleviate some of the difficulties for students. Unplugged activities have been shown to increase student interest, and to enhance student understanding of CS programming concepts [1]. We have used unplugged activities to teach PDC concepts before introducing parallel programming. Our experiences show that using unplugged activities to introduce the PDC concepts reduce the barrier to learn parallel programming.

CCS CONCEPTS

• **Social and professional topics** → *CS1*; • **Computing methodologies**;

KEYWORDS

Unplugged Activities, Parallel and Distributed Computing, Introductory Programming

1 INTRODUCTION

Parallel and Distributed Computing (PDC) has become ubiquitous as modern computing devices from high-end servers to cell phones arrive with multiple cores as well as GPU capabilities. The changing landscape points to the need for every programmer to understand how parallelism and distributed computing affect problem solving. Early programming courses present complex concepts to novice students, and additionally they include added complexity with syntax, structures, and semantics of particular programming languages. Therefore, learning programming is challenging for early CS/CE students, as reported in the literature [2]. ACM has recommended Parallel and Distributed Computing as a required Knowledge Area for undergraduate Computer Science curricula because of the proliferation of computing devices with parallel capabilities and the need

for students to be able to program these devices. Unfortunately, introducing PDC concepts and associated programming skills add more complexity and is more challenging for many students. Unplugged activities, which are activities done without writing programs or using computers and which usually have physical activity with visual components, have been shown to engage students and convey complex concepts in a way that enhances student learning [3].

2 PRELIMINARY OBSERVATIONS

At Tennessee Tech University we have introduced PDC concepts in CS1 and CS2 classes using both unplugged and plugged activities. We have used two unplugged activities in CS1 and CS2 to help explain several PDC concepts before introducing hands-on programming exercises using OpenMP. One such unplugged activity is sorting colored candy, according to the color, first by one student and then by three students to illustrate the concept of concurrency, speedup, and sequential dependency. Another such activity was to find the youngest person in class, to illustrate the concept of multicore, inter-process communication, and synchronization. Details of many such activities is available on our website[4]. We evaluated the impact of using unplugged activities on ease of learning, through pre-post surveys and also looking at the grades on PDC related exams and lab assignments. Our evaluation indicates that using unplugged activities makes it easier to learn PDC concepts. We have seen a gain of 1.55 points on a 5 point scale when using unplugged exercises in addition to plugged laboratories compared to a gain of .88 points when using the plugged activities alone.

3 FUTURE WORK

We plan to develop more hands-on exercises that are applicable to upper level classes and additional evaluations to comprehensively study the impact of the activities on learning PDC concepts.

REFERENCES

[1] B. Rodriguez, C. Rader, and T. Camp, "Using Student Performance to Assess CS Unplugged Activities in a Classroom Environment," in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE 16*, 2016, pp. 95 – 100
[2] C. Watson and F. W. B. Li, "Failure rates in introductory programming revisited," in *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE 14, 2014*, pp. 39 – 44.
[3] T. Bell, J. Alexander, I. Freeman, and M. Grimley, "Computer science unplugged: School students doing real computing without computers," *New Zeal. J. Appl. Comput. Inf. Technol., vol. 13, no. 1*, pp. 20 – 29, 2009.
[4] "iPDC Modules", https://www.csc.tntech.edu/pdcincs/index.php/ipdc-modules/