A Case Study: Incorporating Parallel and Distributed Computing into Computer Science Curriculum

Ali Abu-El Humos[1*] , Sungbum Hong[1], Tzusheng Pei[1] and Omar Aljawfi[1]

[1]Department of Electrical & Computer Engineering and Computer Science,

Jackson State University, Jackson, MS 39217, USA

[*]Corresponding Author Email: ali.a.humos@jsums.edu

ABSTRACT

Recent technology advances in parallel computing such as multicore CPUs, GPUs, and their driving software require a well-prepared workforce to support this demanding and fast changing industry. Parallel and Distributed Computing (PDC) education for computer science and computer engineering majors will play a major role in preparing well trained graduates to join this workforce. In this work, we share past and future plans to update the computer science curriculum at Jackson State University (JSU) with PDC modules. As part of this effort, some of the NSF/IEEE-TCPP curriculum initiative on PDC modules were integrated into department-wide core and elective courses offered in both fall and spring semesters. These courses were: CSC 119 Object Oriented Programming (core) [2, 4, 6, 9], CSC 216 Computer Architecture and Organization (core) [3, 5, 9], CSC 312 Advanced Computer Architecture (core) [3,5], CSC 325 Operating Systems (core) [6, 9], CSC 350 Organization of Programming Languages (core) [9], CSC 425 Parallel Computing (elective) [1, 2, 6] , CSC 499 Special Topics: Data Mining (elective) and UNIV 100 University Success course, which is a university-wide class offered for all JSU majors. In an effort to update the contents of the UNIV 100 course, some contemporary PDC topics and their essence in higher education were incorporated into this course. The inclusion of the PDC modules was gradual and light weighted in the lower level courses and
more aggressive in the higher-level courses to let the students easily grasp PDC concepts. Specific test questions, homework assignments and projects were developed to assess students' performance.

1. RESEARCH OBJECTIVES

• Update computer science curriculum at JSU according to the NSF/IEEE-TCPP Curriculum Initiative [11, 12]. This goal will be measured by the changes implemented into the core and elective computer science courses to incorporate PDC topics. These changes should positively impact the computer science ABET accreditation.
• Help computer science - and all JSU students in general-gain the PDC and High-Performance Computing (HPC) knowledge. Some short-term measures for this goal will be exam questions, assignments and class projects to assess students' knowledge of the PDC topics every semester. A long-term measure will be the number of CS students who joined internships or permanent jobs that require PDC knowledge.
• Engage faculty and students in best learning practices of PDC topics. This goal will be measured through a student-feedback survey where they indicate the PDC topic they enjoyed and learned the most, and the technique the instructor used to trigger their interest.
• Create research venues for both undergraduate and graduate students in the areas of PDC and (HPC). This goal will be measured by the number of research projects and publications students and faculty were able to produce in the area of PDC during the project duration. Another measure will be the number of workshops, conferences, and tutoring sessions students were engaged in.

1.1. Benefits to STEM Education and Research at JSU

• Computer science students will be learning the latest technology advancements in PDC.
• STEM students at JSU will learn how PDC can be applied to solve science and engineering problems in their discipline.
• All JSU students will learn an overview of PDC topics integrated into UNIV 100 course.
• This project is expected to create a multidisciplinary research environment, where computer science students and faculty will be engaged with the Computational Data Science and Engineering PhD (CDS&E) program at JSU. Many CDS&E research projects require the PDC knowledge our CS students will gain during this project.

1.2. Baseline data

Our research team was awarded the NSF/IEEE-TCPP Curriculum Initiative Early Adopters award in Fall 2014 and Fall 2015 consecutively [7, 8]. That provided our team with the opportunity to revisit some of the courses taught in the computer science program and to incorporate some PDC modules into these courses. The first Early Adopter award was targeting computer science students during Fall 2014 [7]. The feedback results (through assignments and surveys) showed that most students were highly interested and motivated to learn PDC topics. However, most students indicated that PDC topics need to be more emphasized and organized across the computer science curriculum. The second Early Adopter award was targeting non-computer science students during Fall 2015 [8]. The feedback results showed that students have no previous knowledge of PDC. However, more than 50% of the students expressed their interest to learn about this topic. Some of them did not mind pursuing a career that may require some PDC background. The majority of the students, though, were not sure how PDC can be related to their major field of study.

1.3. Evidence-Based Instructional Practices

The computer science program at JSU is updating its curriculum with PDC modules. As part of this effort, several courses will integrate modules of the NSF/IEEE-TCPP curriculum initiative on PDC into department-wide core and elective courses offered on both fall and spring semesters. These courses are: CSC 119 Object Oriented Programming (core), CSC 216 Computer Architecture and Organization (core), CSC 312 Advanced Computer Architecture (core), CSC 325 Operating Systems (core), CSC 350 Organization of Programming Languages (core), CSC 425 Parallel Computing (elective) and CSC 499 Special Topics: Data Mining (elective). The number of computer science freshmen, sophomores, juniors and seniors that will be affected by this project is around 125 students.

2. RESEARCH ACTIVITIES

2.1. Integrate PDC Modules into Computer Science Courses

Every participating faculty member will develop PDC modules to incorporate into his/her course. Assessment data will be collected during the fall and spring semesters. This data will be analyzed by the project evaluator and should result in changes to the PDC modules that will be taught on the following fall and spring semesters.

The following is the list of the courses where PDC modules will be implemented:

A. CSC 119 Object Oriented Programming: This is the second course in the computer science programming sequence and is required for all computer science majors. For the purpose of supporting parallel computing, Java has the Thread class and the Runnable interface, and it also provides rich primitives with the java.util.concurrent packages, which include the fork/join framework. Students can explore these features in Java for parallel computing.

B. CSC 216 Computer Architecture and Organization: This course covers the basic concepts of computer architecture, which includes machine level representations of data, computer arithmetic, instruction set architecture and assembly language, datapath and control, memory system, bus architectures and I/O devices. A new PDC module will be added to this course to introduce students to multicore processors and GPU hardware. Also, throughout the course, parallelism at different levels will be discussed.

C. ECE 412 Computer Architecture: This is the second course in the computer architecture sequence offered by the department. Advanced topics of PDC curriculum are covered, such as instruction level parallelism: pipelining and superscalar architectures, processor level parallelism: array processors, multiprocessor and multicomputer systems. Techniques to reduce instruction pipeline stalls, and set associative caches are analyzed. Quantitative approaches of computer performance are emphasized. A new PDC module will be added to this course to introduce students to OpenMP and Cuda GPU programming.

D. CSC 325 Operating Systems: This course introduces the major concepts of process communication and synchronization, protection, performance measurement, causes and evaluations of the problems associated with mutual exclusions and process synchronization among concurrent processes. It also, introduces and analyzes various operating systems in terms of processor, memory, device, information, and distributed systems management. A PDC module will be added to this course to extend process synchronization issues to parallel programming concepts. With this module, the course will provide students with parallel thread programming opportunities.

E. CSC 350 Organization of Programming Languages: This course is a study of organization and specification of programming languages. It covers several issues in language design, including typing regimens, data structure models, control structure models, abstraction, virtual machines, language translation, interpreters, compiler design, lexical analysis, parsing, symbol tables, declaration and storage management, code generation; and optimization techniques. In this course, after a brief review of the features in Java for supporting parallel computing (taught in CSC 119), parallel programming assignments will be given in order to gain hands-on experience. Generic concepts in parallel computing will be also introduced.

F. CSC 425 Parallel Computing: This is a newly developed elective course. It is a study of the hardware and software issues in parallel computing. Theoretical and practical issues of parallel processing, including a discussion of parallel architectures, parallel programming languages, and parallel algorithms. Programming on multiple parallel platforms in a higher-level parallel language. In this course, the students will learn how to write parallel programs on three different parallel architectures: i) shared memory model- thread programming; ii) Cluster- Message passing Computing; and iii) Multicore- GPU Programming.

G. CSC 499 Special Topics: Data Mining: This course offers an introduction to data mining. Hardware and Software architectures, Map-Reduce programming paradigm on distributed clusters of computers. Finding similar items and Mining data streams. Frequent item sets and Clustering will be covered. A new PDC module will be added to this course where students will learn map-reduce computation for cluster programming.

H. UNIV 100 University Success: This a university-wide course for all JSU freshmen. An ongoing effort has started at JSU to restructure and digitize this course. Some PDC topics will be added to this course and included in the digital course. These topics include: Distributed computing overview, Client/Server paradigm, peer-to-peer paradigm, parallel computing and Internet of Things overview.

2.2. Enhance the Computer Science Program Computing Capabilities with Cutting Edge GPU Hardware and PDC Software

40 PC Computer Lab for CSC 119, one Cluster that consists of 8 nodes of computers for CSC 215 for Hadoop and MPI, a Server with Tesla K40 Supported by Nvidia Corporation that has for CSC325 Operating Systems and CSC 425 Parallel Computing, and one server that consists of 4 computer nodes for CSC 360 Client-Server Computing. This equipment will be the seed for the new PDC lab.

2.3. Organize PDC Workshop for STEM Students and Faculty at JSU

This event will be held every semester. STEM students and faculty will be trained on GPU programming and MATLAB Parallel Computing toolbox and will discuss research projects where these technologies can be used.

3. TIMELINE

All involved faculty members will submit their assessment results as part of the Faculty Course Assessment Report (FCAR), which is prepared every semester. The FCAR will contain mapping between the PDC modules and the specific course outcomes. It also provides assessment data and feedback results from students. It should indicate which changes were implemented into the course and the recommendations to enhance the course quality. The FCAR data are included in the ABET report.

Table I shows a summary of the project timeline. Table II shows the project objectives and its measurement tools Timeline.

4. CONCLUSION AND FUTURE WORK

Currently, we are in the process of refining the PDC material that will be used in the aforementioned courses. A major challenge is to find some examples where PDC knowledge will relate to the student's field of study as well as their future careers. We will continue collect assessment data and feedback from both students and faculty members which will guide us develop a more comprehensive plan to integrate PDC topics into the CS curriculum.

5. REFERENCES

[1] J. Adams, "Early Adopter: Integration of Parallel Topics in the Undergraduate CS Curriculum at Calvin, 2011", *1st NSF/TCPP Workshop on Parallel and Distributed Computing Education (EduPar-11)*, Anchorage, Alaska, May 2011.

[2] J. Adams, "Multithreading Made Easy: Parallelism in Scratch", *42nd SIGCSE Technical Symposium on Computer Science Education*, Dallas, Texas, March 2011.

[3] R. Brown, E. Shoop, J. Adams, C. Clifton, M. Garnder, M. Haupt, and P. Hinsbeeck, "Strategies for Preparing Computer Science Students for the Multicore World," *in Proceedings of 15th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), 2010, pp. 99-115*.

[4] R. Brown and E. Shoop, "CSinParallel and Synergy for Rapid Incremental Addition of PDC Into CS Curricula," *in IEEE 26th International Parallel and Distributed Processing Symposium, EduPar Workshop, 2012, pp. 1329–1334*.

[5] Y. Chen, Y. Zhuang, and N. Lopez-Benitez. "Fall-11: Early Adoption of NSF/TCPP PDC Curriculum at Texas Tech University and Beyond". *In the 3rd NSF/TCPP Workshop on Parallel and Distributed Computing Education (EduPar-13), in conjunction with the 27th IEEE International Parallel & Distributed Processing Symposium (IPDPS'13)*, Boston, May 20th, 2013

[6] A. Danner and T. Newhall, "Integrating Parallel and Distributed Computing Topics into an Undergraduate CS Curriculum". *Workshop on Education for High-Performance Computing EduPar 13, in conjunction with the 27th IEEE International Parallel & Distributed Processing Symposium*, Cambridge Boston, Massachusetts USA, May 20, 2013.

[7] A. Humos, S. Hong, J. Jackson, X. Liang and T. Pei, "NSF/TCPP Early Adopter Experience at Jackson State University Department of Computer Science". *Workshop on Education for High-Performance Computing EduHPC 14, in conjunction with SC-14: The International Conference for High Performance Computing, Networking, Storage, and Analysis*, New Orleans, LA, November 16-21, 2014.

[8] A. Humos, S. Hong, G. Offiah, E. Al-Areqi and B. Aldrich, "Incorporate PDC Topics into University Level Digital Computer Principles Class at Jackson State University". *Workshop on Education for High-Performance Computing EduHPC 15, in conjunction with SC-15: The International Conference for High Performance Computing, Networking, Storage, and Analysis*, Austin, TX, November 15-20, 2015.

[9] B. Lu and T. Declue, "A Spiral Curriculum Teaching PDC in a Small College Environment". *Workshop on Education for High-Performance Computing EduPar 13, in conjunction with the 27th IEEE International Parallel & Distributed Processing Symposium*, Cambridge Boston, Massachusetts USA, May 20, 2013.

[10] McKinsey Global Institute, 2011. *Big data: The next frontier for innovation, competition, and productivity. http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation*.

[11] "NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing – Core Topics for Undergraduates." *[Online]. http://www.cs.gsu.edu/~tcpp/curriculum/*

[12] S. K. Prasad, A. Chtchelkanova, S. Das, F. Dehne, M. Gouda, A. Gupta, J. Jaja, K. Kant, A. La Salle, R. LeBlanc, M. Lumsdaine, D. Padua, M. Parashar, V. Prasanna, Y. Robert, A. Rosenberg, S. Sahni, B. Shirazi, A. Sussman, C. Weems, and J. Wu, "NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing: core topics for undergraduates," *in Proceedings of the 42nd ACM technical symposium on Computer science education, New York, NY, USA, 2011, pp. 617–618*.

Table I: Project Timeline Summary

Tasks	Year 1			Year 2		
	Summer	Fall	Spring	Summer	Fall	Spring
Prepare and/or refine PDC Material	x			x		
Integrate PDC Module into CSC 119		x	x		x	x
Integrate PDC Module into CSC 216		x	x		x	x
Integrate PDC Module into ECE 412		x	x		x	X
Integrate PDC Module into CSC 325		x	x		x	x
Integrate PDC Module into CSC 350			x			x
Integrate PDC Module into CSC 425			x			x
Integrate PDC Module into CSC 499		x			x	
Integrate PDC Module into UNIV 100		x	x		x	x
JSU STEM PDC Workshop		x	x		x	x
Assessment data gathering		x	x		x	x
Formative and Summative Assessment		x	x	x	x	x

Table II: Project Objectives and Corresponding Measurement Tools Timeline.

Objective	Measurement Tool	Timeline
Update computer science curriculum at JSU according to the NSF/IEEE-TCPP Curriculum Initiative.	Changes implemented into computer science core and elective courses to incorporate PDC topics.	End of every Fall and Spring Semester
Help computer science students - and JSU students in general-, gain the PDC and HPC knowledge.	*Exam questions, assignments and class projects. *Number of CS students who worked in internships or permanent jobs that require PDC knowledge.	*During Fall and Spring Semesters *Every Summer Semester
Engage faculty and students in best learning practices of PDC topics.	Student-feedback survey.	End of every Fall and Spring Semester
Create research venues for both undergraduate and graduate students in the areas of PDC and HPC.	Number of research projects publications, workshops, conferences, and tutoring sessions.	End of Spring