Journal of Engineering Technology (ISSN 0747-9964)
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 117-124

ORIGINAL ARTICLE

Incorporating Parallel Computing Education at Undergraduate Level in Developing Economies

Muhammad Kashif Hanif[1*], Muhammad Umer Sarwar[1], Ramzan Talib[1], Hassan Ajmal[1], Muhammad Asif[2]

[1]Department of Computer Science, Government College University, Faisalabad, Pakistan
[2]Department of Computer Science, National Textile University, Faisalabad, Pakistan
*Corresponding Author: mkashifhanif@gcuf.edu.pk

Abstract: Introduction: Modern ICT infrastructure such as parallel and cluster computing centres are vital for transforming developing economies. The process starts from developing capable human resources. We present one of very first attempts to incorporate parallel computing in CS education in Pakistan. The experience of teaching parallel computing course at undergraduate level is described that can provide valuable insights for other universities to follow. **Methodology:** This work provides a qualitative survey of first batch students that provides insightful feedback about the course materials, difficulty level, and effectiveness. **Results and Conclusion:** Students in developing countries do not even have a vague idea about the potential and opportunities in parallel computing. The main reasons are unavailability of faculty or reluctance to teach advanced courses as well as lack of expensive hardware resources. The goal of the course was to introduce students to this exciting field and increase their knowledge and skill in parallel computing techniques to solve complex problems. We offer details of syllabus and courses which can be included at undergraduate level in this paper. The contents were designed by considering the syllabus of big data and computational biology courses world-wide. Students have shown keen interest in the course and willingness to take advance course.

Keywords: Computing Education, Pedagogy, Parallel Computing, Graphics Processing Unit, CUDA.

1. Introduction

Developing economies can be transformed into developed economies within short span by fulfilling the technology gap. Developing countries cannot afford to invest major share of resources on costly technology and infrastructure. However, capitalization on human resource is promising venue for such economies. Collaboration of developing and developed countries hinges on capable human resource. To this end, there is need of education paradigm shift at grass root level. Obsolete educational syllabus in developing countries should be transformed to include new state of the art technologies and blend fundamental but out-of-date concepts that are not relevant to current era.

In the context of computing education, universities should embrace parallel computing and related advanced technologies [1, 2]. For this purpose, parallel computing course was offered in 8th semester to undergraduate students as an elective subject at National Textile University, Pakistan in spring 2016 semester. This course is a first step in long journey of the educational transformations in Pakistan. We have designed and developed this course to meet the future computing trends and human resource demand in this field.

Similar course will be offered in at Government College University, Faisalabad, Pakistan in fall 2017 semester.

The effectiveness and benefits of this course cannot be measured in short time. However, we have designed a qualitative survey to assess student's satisfaction level and learning outcomes of this course. We received overwhelming response from students to include this subject and its advanced contents into the curriculum. One important issue in this respect is when to introduce this course during undergraduate degree program. The survey results provide useful insights to find out the best time to learn this subject in order to achieve maximum benefits. Moreover, we also discover interesting correlation between core computing subjects and parallel computing. In results section, we share striking outcomes based on student's feedback that can be very helpful in future replication of this course for collaborators and academic executives.

The rest of the paper is organized in different sections. In the next section, we present the course design, andragogical approach, and survey design. In sections 3, the results of the survey were discussed. We conclude the outcomes in section 4.

2. Course design

Undergraduate computer science curriculum has programming courses which are taught using the serial programming languages. In current era of technology, parallel programming skills are becoming increasingly important [1], [2], [3], [4]. However, majority of students in developing countries are unaware of parallel computing paradigm. Parallel computing concepts are generally considered difficult to learn and master. The proposed course contents should be designed to address aforementioned issues by keeping in mind students background and programming ability. The goal of the course is to introduce concepts related to parallel computing.

In this course, we have made special effort to develop a pedagogical approach that covers both theoretical and practical components of the field at introductory level. First, we needed to make theoretical base about parallel computing that familiarize students with basic concepts. Then, the criteria to classify parallel computing platforms and to evaluate the performance were discussed followed by detailed techniques to map algorithms on parallel hardware. In last quarter of course, we addressed the Graphics Processing Unit (GPU) architecture and programming model culminating at discussion of Compute Unified Device Architecture (CUDA) [5], OpenCL [6], and OpenACC [7]. Table 1 provides details of topic wise contents of the course.

Modern GPUs are widely used as a parallel processor to perform computation intensive tasks. Computational Biology, Big Data, Machine Learning, and other areas are taking advantage of computational power of GPUs to attain high performance [5], [7], [9], [10]. The programming model for NVIDIA GPUs is CUDA [5]. CUDA enables programmers to write programs with minimal extensions to traditional programming languages. The trend to use GPU for parallel processing is expected to continue in coming years [2], [3], [11], [12]. For these reasons, we decided to use GPU for practical part. Another reason is, Department of Computer Science at National Textile University, Pakistan is a GPU education centre designated by NVIDIA Inc.

To better understand the concepts, we agreed to employ an integrative example problem which should be used throughout the course. We opted Floyd-Warshall algorithm [13], [14] as an example. The major reasons to select Floyd-Warshall algorithm were:

- It is widely used in many applications to find the shortest path and
- Students had already learned this algorithm in previous course.

This algorithm was first implemented using sequential program. As the course progressed, we formulated a parallel version [15], [16], [17], [18]. This was achieved by removing data dependencies. The parallel version was implemented on GPU using fine and coarse grained decomposition [16].

The biggest challenge faced by students was to make transition from sequential programming to parallel programming. However, students had shown keen interest in parallel computing with the course progression by demonstrating increased capability in advanced concepts.

Table 1. Parallel Computing curriculum for undergraduate students

Week	Topic
1	Introduction to parallel computing
2	Parallel computing classification
3	Dependence analysis in parallel computing
4	Physical organization of parallel platform
5	Performance analysis
6	Principles of parallel algorithm design
7	Mapping parallel algorithm techniques
8	Parallel algorithm models
9	Graph algorithm
10	Dynamic programming
11	GPU computing
12	Kernel based parallel programming
13	CUDA memory model
14	Performance consideration
15	Introduction to OpenACC
16	Introduction to OpenCL

Journal of Engineering Technology (ISSN 0747-9964)
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 117-124
ORIGINAL ARTICLE

2.1 Evaluation Survey

In order to learn about the effectiveness and correlation with the other courses, a survey was designed consisting of closed, mostly multiple choice or Likert-style questions [19]. The survey had three parts. The first part had general questions about the respondent's interest and participation in the course. The second part was intended to find the relationship with other courses and how the learning experience of these courses could help to progress in parallel computing course. The last part was planned to get student opinion for the course contents and difficulty level.

3 RESULTS AND DISCUSSION

The post course evaluation and feedback survey was conducted in spring semester 2015 from participating students. They were ensured to maintain the confidentiality of their information to foster their honesty and to reduce their hesitation. The survey based evaluation will help in redesigning the course for subsequent offering. The total number of responses to the survey was 44 of which approximately 81% were male and 19% were female. The survey results depict majority of students wants to study the advanced topics of parallel computing (77%). However there a difference of opinion exists about the schedule of course offer during undergrad program. Most of the students, 66%, wanted to study this course in 6th or 7th semester as presented in Figure 1. The reason behind is, this schedule suits well to the students who wish to conduct their Final Year Project in parallel computing field. Secondly, until this time students have already taken all the prerequisite courses.

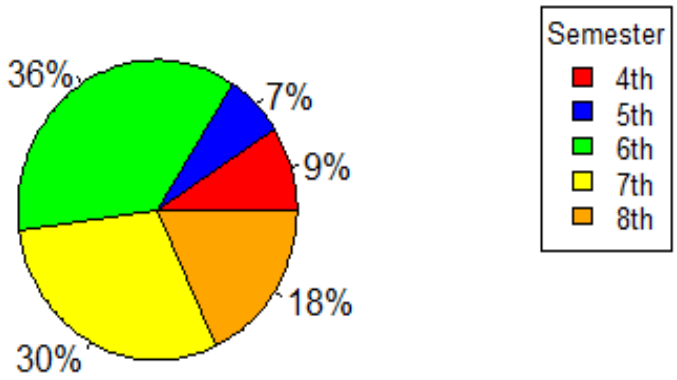

Figure. 1. Student opinion when parallel computing course should be offered.

At the time of introducing this course, main apprehension was whether it will be appreciated by the students or not. The survey assessed student acceptability by how much stimulating and engaging they found this course on a Likert scale. Their response was very encouraging as 68% students found it very interesting and 82% participated actively during the class (Figure 2). This reflects a good first initiative in the region.

Even though it was offered as an elective course, there are strong conceptual connections with other core computing courses that should be kept in view while designing the contents. Our survey evaluated students view of such dependencies by asking respondents about most related courses that helped them during this course. Students reported above 80% relevance with Problem Solving Using C, Linear Algebra, Computer

Journal of Engineering Technology (ISSN 0747-9964)
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 117-124
ORIGINAL ARTICLE

Architecture, Data Structures and Design and Analysis of Algorithms courses. However, Software Engineering course was observed as least relevant. Figure 3 depicts more detail on this question.

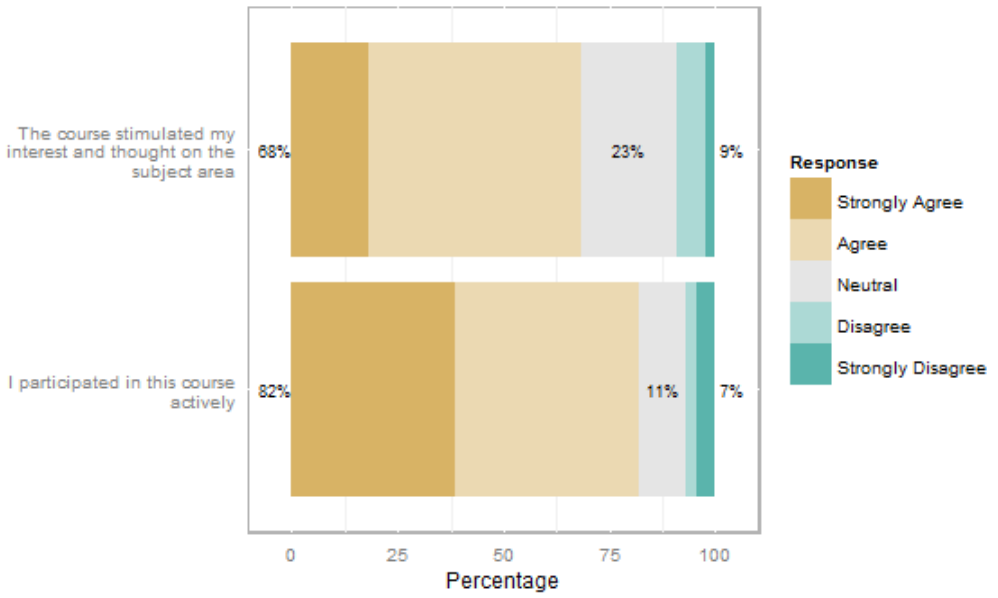

Figure. 2. Student interest and participation in the course.

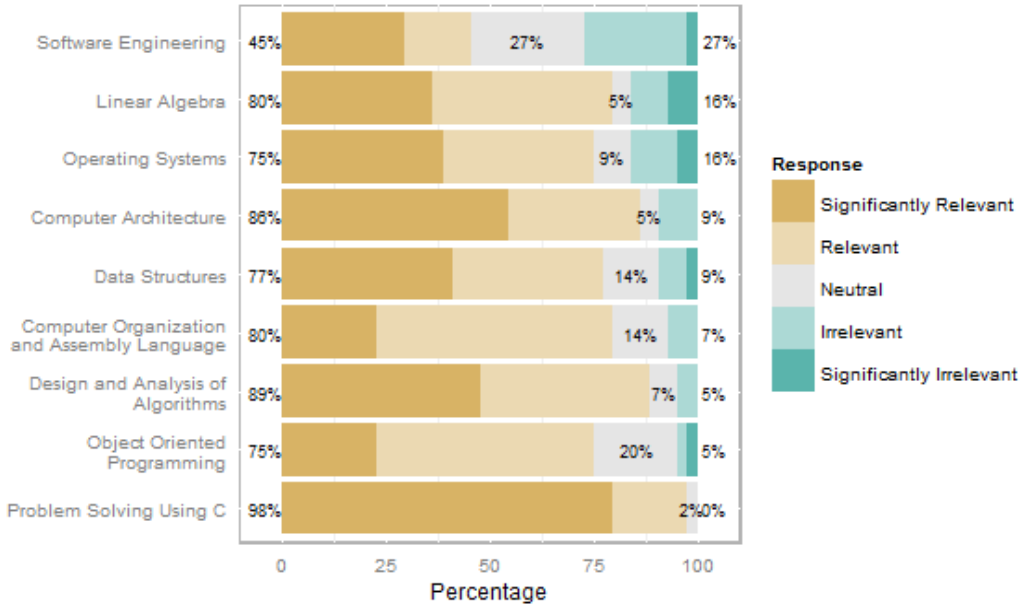

Figure. 3. Courses related to parallel computing.

Journal of Engineering Technology (ISSN 0747-9964)
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 117-124
ORIGINAL ARTICLE

The questionnaire also judges the difficulty of different topics. The purpose was to get insight from student's perspective to adapt the time allocated to each topic. Figure 4 summarizes all responses where each topic is listed on Y-Axis along with perceived difficulty level. X-axis show aggregated response on percentage scale. As expected, Mapping Parallel Algorithm Techniques and Graph Algorithms were noted as most difficult topics. Whereas, Parallel Computer Classification and Dependencies Analysis were observed as easily understandable by, respectively, 73% and 68% participants. Rest of the topics received mixed response. Students face problems learning a new programming model, especially in implementation related topics. We recommend allocating more time for the implementation specific topics to facilitate better understanding of the concepts.

A similar question was asked to know the opinion of students about future course contents. It can be observed from summary graph of responses, Figure 5, that majority of the participants agrees with current course design. However, there are some topic such as Graph algorithms and Dynamic Programming that were perceived less important by some of the students. The reason behind this opinion may be involvement of mathematical intensive operations that are prerequisite to fully appreciate the importance of these topics. Similarly OpenCL was noted by approximately 32% respondents as less relevant. This may be due to different programming model than CUDA.

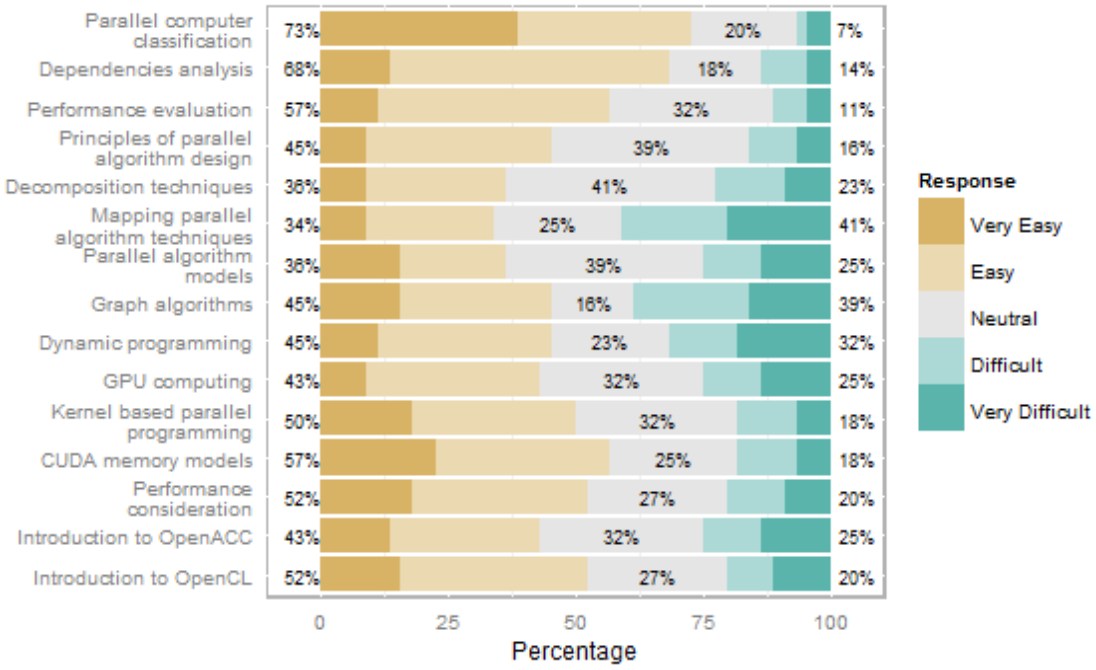

Figure. 4. Different course topics difficulty level.

Journal of Engineering Technology (ISSN 0747-9964)
Volume 6, Special Issue on Technology Innovations and Applications
Oct. 2017, PP. 117-124

ORIGINAL ARTICLE

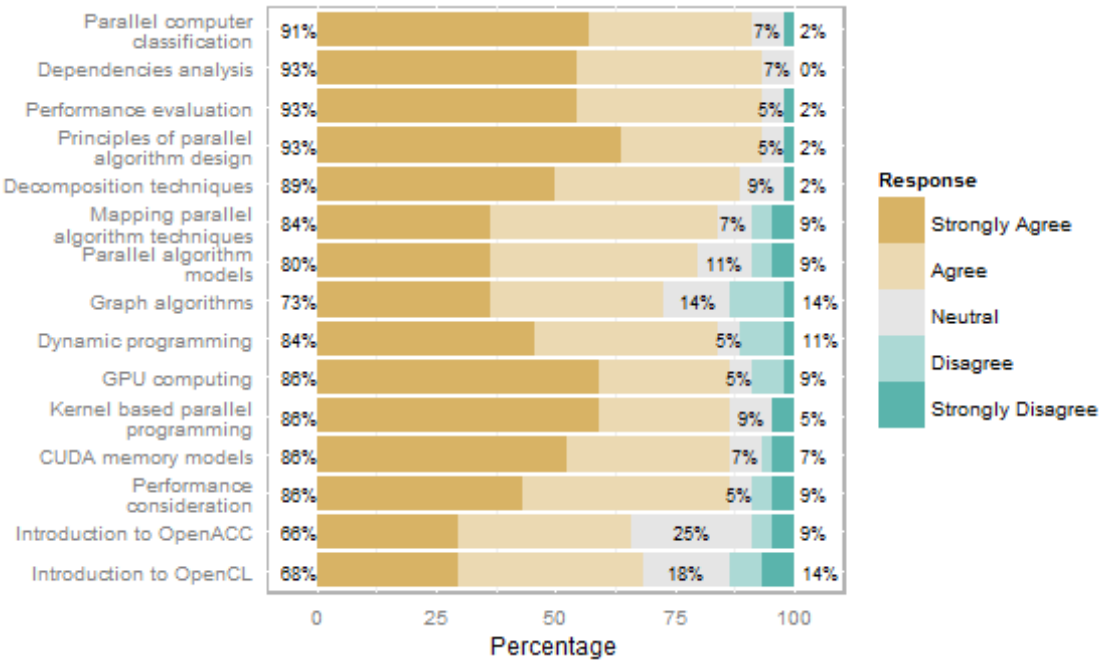

Figure. 5. Student opinion about course contents for future.

4 Conclusion

The importance of parallel computing training is evident from its being used in myriad of technologies e.g. Super Computers, Grid, Big Data Analytics. Shortage of expert human resources in this field is a serious bottleneck in wide scale adoption of this technology especially in developing world. Countries like Pakistan can greatly benefit by preparing computing professional who can contribute in global market for this technology. We attempted to introduce Parallel Computing at undergrad level as an elective course. Our pedagogical approach focused on practical work and hands on training of the students to learn the core concepts. The lessons learned from conducting this course are; i) best time to offer this course in four year degree program is between 6 and 7 semester. ii) It is difficult to thoroughly cover both theoretical and practical aspects in one course. Therefore contents should be balanced and divided in a sequel of two courses. iii) Students cannot readily apply their skills if proper hardware resources are not available. School planning on running such courses should procure recommended computing resources to develop required learning outcomes.

References

[1]. Yazici. A, Mishra. A, Karakaya. Z, "Teaching parallel computing concepts using real-life applications", International Journal of Engineering Education, 2016; 32(2):772-781.

[2]. Navarro. C, Hitschfeld-Kahler. N, Mateu. L, "A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures", Communications in Computational Physics, 2014; 15(2):285-329.

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

[4]. Mullen. J, Byun. C, Gadepally. V, Samsi. S, Reuther. A, Kepner. J, "Learning by doing, High Performance Computing education in the MOOC era", Journal of Parallel and Distributed Computing, 2017; 105:105-115.

[5]. Nvidia, "NVIDIA CUDA compute unified device architecture programming guide", NVIDIA Corporation, 2015.

[6]. Munshi. A, Gaster. B, Mattson. TG, Ginsburg. D, "OpenCL programming guide", Pearson Education, 2011.

[7]. CAPS Enterprise, Cray Inc., NVIDIA, the Portland Group, "The OpenACC application programming interface", 2011.

[8]. Che. S, Boyer. M, Meng. J, Tarjan. D, Sheaffer. JW, Skadron. K, "A performance study of general-purpose applications on graphics processors using CUDA", Journal of parallel and distributed computing 2008; 68(10):1370-1380.

[9]. Chen. Y, Qiao. Z, Davis. S, Jiang. H, Li. KC, "Pipelined multi-GPU MapReduce for big-data processing", in Computer and Information Science, 2013; 231-246.

[10]. Xiao. S, Aji. AM, Feng. WC, "On the robust mapping of dynamic programming onto a graphics processing unit", in 15th International Conference on Parallel and Distributed Systems (ICPADS), 2009; 26-33.

[11]. Asanovic. K, Bodik. R, Demmel. J, Keaveny. T, Keutzer. K, Kubiatowicz. J, Morgan. N, Patterson. D, Sen. K, Wawrzynek. J, Wessel. D, "A view of the parallel computing landscape", Communications of the ACM 2009; 52(10):56-67.

[12]. Keckler. SW, Dally. WJ, Khailany. B, Garland. M, Glasco. D, "GPUs and the future of parallel computing", IEEE Micro 2011; 31(5):7-17.

[13]. Floyd. RW, "Algorithm 97: shortest path", Communications of the ACM 1962; 5(6):345.

[14]. Warshall. S, "A theorem on boolean matrices", Journal of the ACM 1962; 9(1):11-12.

[15]. Grama. A, "Introduction to parallel computing", Pearson Education, 2003.

[16]. Hanif. MK, "Mapping dynamic programming algorithms on graphics processing units", PhD thesis, Institut für Rechnertechnologie, Technische Universität Hamburg-Harburg, 2014.

[17]. Romani, F, "Shortest-path problem is not harder than matrix multiplication", Information Processing Letters 1980; 11(3):134-136.

[18]. Torgasin. S, Zimmermann. KH, "An all-pairs shortest path algorithm for bipartite graphs", Central European Journal of Computer Science 2013; 3(4):149-157.

[19]. Likert. R, "A technique for the measurement of attitudes", Archives of psychology, 1932.