Analysis and Lesson Learned in Teaching Parallel and Distributed Computing for Undergraduate Computer Science Course at Universiti Putra Malaysia

Nor Asilah Wati Abdul Hamid, Masnida Hussin
Communication Technology and Network Department,
Universiti Putra Malaysia, UPM Serdang,
Serdang Selangor, Malaysia.
{asil, masnida}@upm.edu.my

Abstract—Faculty of Computer Sciences and Information Technology, Universiti Putra Malaysia has introduced the Parallel and Distributed Computing courses since 2005 to the undergraduate Computer Sciences Course. This work summarizes our experiences in teaching and embedding the TCPP curriculum in parallel and distributed computing course.

Keywords-Universiti Putra Malaysia, Fakulti Sains Komputer & Teknologi Maklumat, Parallel and Distributed Computing, Early Adopter

I. INTRODUCTION

In October 1998, the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia was officially established to meet the demands for additional academic programmes and to keep pace with rapid developments in the respective field.

In August 2014, we received an award from the NSF/IEEE TCCP Early Adopters Program [1]. We proposed to embed several chapters from TCPP curriculum to our Distributed and Parallel Computing (SKR 3202) course. This work documents our evaluation analysis based from the exams and assignments provided to the students.

II. DISTRIBUTED AND PARALLEL COMPUTING, SKR3202

This course expected the student will capable to describe and apply concepts of distributed and parallel computing. This course is compulsory and act as introductory course for parallel and high performance computing to the undergraduate students. The content of this course covers the foundations of parallel computing, memory architectures, parallel programming model and design of parallel programs. Mainly, we refer to [2] and [3] for slides, notes and examples for this course.

III. PERFORMANCE EVALUATION AND ANALYSIS

In line with the Curriculum Initiative's, this work adapt the Programming Topics, which considered the following sub-topics; Parallel Programming Paradigms, Semantics/correctness issues and performance issues. The expectation from the adaptation on the Curriculum Initiative is to enable the students to “apply” (A) and “comprehend” (C) the following sub-topics:-

i. Parallel Programming paradigms
   ▪ Shared memory (A), Compiler directives/pragmas (C), Libraries (C)
   ▪ Distributed Memory (C), Message passing (A), PGAS Language (A), Hybrid (A)
   ▪ SPMD (C), Data parallel (A), Parallel loop(C)

ii. Semantics and Correctness
   ▪ Synchronisation (A), Concurrency defects (C)
   ▪ Deadlocks (A), Critical Regions (A)

iii. Performance issues
   ▪ Load Balancing (C)
   ▪ Performance monitoring tools
   ▪ Performance metrics (C) (Speedup (C), Efficiency(C)

In this section we explain our analysis based from the students exam results and assignments. The numbers of students involve in this analysis were eighteen (18) and during this analysis they were in semester four (4). Refer to Table I, we only plot several items that represent each sub-topics that we manage to highlight during the 14 weeks lectures. The other items that we do not mention in the table are considered students only know or comprehend (C).

The results and analysis in Table 1 were evaluated and collected based from the evaluations which are assignments (20%), Test 1 (20%), Test 2 (20%) and final examination (40%). For example, for sub-topics hybrid, we consider all the students are comprehended due to the students only understand the concept without being evaluated in practical or technical questions. While for sub-topics distributed memory (implemented with OpenMPI), all the students were evaluated with technical questions and hands-on assignments. So, for those unable to perform the OpenMPI program and answer the technical question, we consider they only comprehend with those sub-topics.

Furthermore, from Table I it shows that most of the students can apply (A) distributed memory paradigm and OpenMPI as the library, while some can apply (A) shared memory paradigm and OpenMP as the library. However, none of the student can apply (A) Hybrid and Data Parallel...
sub-topics. Moreover, most of the students only comprehend (C) and unable to fully apply the semantics and correctness sub-topics in their parallel programs. In performance issues almost half of the students in the class manage to apply (A) the load balancing concept in their parallel programs and problems. However, only a few students can apply (A) the usage of performance monitoring tools. Finally, more than half of the class manage to apply (A) and fully understand the performance metrics which is speedup and efficiency.

We can conclude that, the main problems of the students unable to apply for certain sub-topics are due to the following problems;

- Limited basic knowledge in operating system and computer architecture.
- Limited time in class and too many topics to cover.
- The students refuse to learn many parallel programming libraries/ language in short period of time.
- Laboratory class is not provided.
- This is an introductory course for parallel, so the students need more time to adapt the basic concept of parallel computing.

### TABLE I. PERCENTAGE OF COMPREHEND (C) AND APPLY (A) FOR TCPP EMBEDDED TOPICS

<table>
<thead>
<tr>
<th>Topics</th>
<th>Sub Topics</th>
<th>Comprehend (C)</th>
<th>Apply (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Programming Paradigm</td>
<td>Shared Memory (OpenMP)</td>
<td>72%</td>
<td>38%</td>
</tr>
<tr>
<td></td>
<td>Distributed Memory (OpenMPI)</td>
<td>32%</td>
<td>78%</td>
</tr>
<tr>
<td></td>
<td>Hybrid</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>Data Parallel</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Semantics and Correctness</td>
<td>Synchronization</td>
<td>78%</td>
<td>32%</td>
</tr>
<tr>
<td></td>
<td>Dead Lock and Critical regions</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Performance Issues</td>
<td>Load Balancing</td>
<td>55%</td>
<td>45%</td>
</tr>
<tr>
<td></td>
<td>Performance Monitoring Tools</td>
<td>89%</td>
<td>11%</td>
</tr>
<tr>
<td></td>
<td>Performance Metrics (Speedup, Efficiency)</td>
<td>32%</td>
<td>78%</td>
</tr>
</tbody>
</table>

### IV. EXPERIENCES AND LESSONS LEARNED

Additionally, we also discover several weaknesses which need attention and improvements. The following are some of lesson that we have learned during we conduct the course;

- Allow the students to execute several parallel programs such as Hello in week 7. By doing this, the students started to have more interest in understanding the structure of parallel program paradigm.
- Guide the students to execute several programs of OpenMP and OpenMPI which include communication between processors. Once the interest to know how those thing work, then explain the concept of communication, synchronization, load balancing and other issue related to parallel program.
- Practical work is essential. Students need to have more practical works particularly after week seven (7) and the percentage of assignment and projects should be increase.
- Frequent small practical assignments are essential to prepare the students for their projects. We observed that when students were working on the projects they spent most of their time dealing with basic concepts. By the time they dominated the technology and the requirements for the assignment they were close to deadline.
- Continuity of parallel programming courses is needed after the introduction to parallel computing course. The continuity from the introduction of parallel computing courses is crucially needed to support and provide more times to prepare the students with essential knowledge in parallel computing.

### V. CONCLUSION

As a conclusion, we found that the Curriculum Initiative is very useful and beneficial in helping faculty to improve and align with the other high ranking international universities related to parallel computing. Importantly, the Curriculum Initiative provides guidelines and exposure particularly for students with less exposure on parallel and distributed computing learning environments.

### ACKNOWLEDGMENT

The authors would like to thank faculty and authorities of Faculty Faculty of Computer Sciences and Information Technology, Universiti Putra Malaysia

### REFERENCES