COMP428/6281: Parallel Programming

Course Outline
Fall 2019

Professor in charge	Dr. Dhrubajyoti Goswami, P. Eng. E-mail: goswami@cs.concordia.ca Web : http://www.cs.concordia.ca/~goswami/ Office : EV 3.145 Tel : (514) 848-2424 Ext. 7882
Tutors and Lab Demonstrators	TBA
Marker	TBA

Course Description

Pre-requisites: COMP 346 or COMP 5461

Parallel programming techniques as a natural extension to sequential programming. Overview of parallel programming architectures and models. Parallel programming issues: locality, granularity, scheduling, data decomposition and distribution, load balancing, communication and synchronization, determinacy and non-determinacy, cost and performance. Techniques and tools for message-passing parallel programming.

Objectives

This is a course on parallel programming. There will be two distinct components of the course: theory and practice. The practical part will be carried out in a Beowulf cluster running Linux and OpenMPI (http://www.open-mpi.org/). The programming language on the cluster is C/C++. More details about the course contents are in the following.

Textbook

Introduction to Parallel Computing (Second Edition), by Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar. Addison Wesley. ISBN 0-201-64865-2

Additional references

There will be some additional references including research article(s) which will be mentioned in class and via Moodle.

Tentative Course Coverage

The first part of the course will cover selected topics from Chapters 1 to 12 of the textbook. Some of the chapters will be covered in detail depending on their relevance to the course description. The second part of the course will follow research paper(s) on selected important topics, e.g., automatic parallelization, dependency analysis. The course coverage may be adjusted based on course progress throughout the semester.

Workload and Grading (COMP428)

It has the following three components:

1. **Assignments: 30%**

There will be three assignments, each with theory and/or programming components and to be done individually unless otherwise stated. The programming components will use C/C++ and MPI on a Beowulf cluster. The last programming assignment will be considered as a project. **You must pass the assignments to pass the course.**

2. **Midterm**: **20%**

There will be one mid term exam. The midterm will cover all material presented in the lectures, the textbook and other lecture material, and in the assignments, up to and including the lecture preceding the exam**.** The exam will include multiple choice questions, questions with detailed answers, and True/False questions requiring short justifications. There will be no substitute midterm.

3. **Final exam: 50%**

The final exam will be scheduled by the University. The exam will cover material from the entire semester, including lectures, textbook and other lecture material, and assignments. The exam will include multiple choice questions, questions with detailed answers, and True/False questions requiring short justifications. **You must pass in the final exam and pass in the total marks obtained to pass the course.**

Workload and Grading (COMP6281)

It has the following four components:

1. **Assignments: 20%**

Refer to the above description for COMP428. **You must pass the assignments to pass the course.**

2. **Midterm**: **20%**

Refer to the above description for COMP428.

3. **Literature Survey and Presentation: 20%**

The graduate part of the course will have an additional literature survey, a brief report, and a presentation. The topic area will be based on some important research domain of today and will be decided about the time of the Midterm.

4. **Final exam: 40%**

Refer to the previous description for COMP428. **You must pass in the final exam and pass in the total marks obtained to pass the course.**

The following apply to both COMP428 and COMP6281: usually a score of 50% is required to pass. However, it is important to note that the grading of the course will be done based on the relative percentages assigned to the assignments and the exams. For reasons of fairness, we may choose to scale up/down the marks in a particular exam or assignment to ensure that all aspects of the course receive a fair weight. Finally, there are no pre-set cutoff points for the final grades; the cutoff points will be decided based on an assessment of difficulty level, class performance, fairness, and instructor’s wisdom from teaching and grading the course in the past.

Important note: In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change

Tutorials and labs

Please refer to the departmental/University course web page for tutorial/lab timings.

CEAB/CIPS Graduate Attributes

As part of either the Computer Science or Software Engineering program curriculum, the content of this course includes material and exercises related to the teaching and evaluation of graduate attributes. Graduate attributes are skills that have been identified by the Canadian Engineering Accreditation Board (CEAB) and the Canadian Information Processing Society (CIPS) as being central to the formation of engineers, computer scientists and information technology professionals. As such, the accreditation criteria for the Software Engineering and Computer Science programs dictate that graduate attributes are taught and evaluated as part of the courses. The following is the list of graduate attributes covered in this course, along with a description of how these attributes are incorporated in the course:

1. ***Knowledge base:*** *Parallel programming techniques as a natural extension to sequential programming. Overview of parallel programming architectures and models. Parallel programming issues: locality, granularity, scheduling, data decomposition and distribution, load balancing, communication and synchronization, determinacy and non-determinacy, cost and performance. Techniques and tools for message-passing parallel programming.*
2. **Design:** *Develop efficient, scalable applications that solve data management and/or computational challenges within high volume, resource-intensive domains.*
3. ***Use of Engineering tools:*** *Use various tools for parallel programming including explicit parallel programming tools.*

Moodle and other resources

You will be able to access the course material and general discussion for this course via Moodle, which you can access through this link www.myconcordia.ca. More details about the Beowulf cluster will be later communicated in class, tutorials, labs, and via Moodle. The following is a link to official MPI documentation: http://www.mpi-forum.org/docs/.

All assignment submissions are electronic through the electronic assignment submission system (https://fis.encs.concordia.ca/eas).

Academic Misconduct

Any form of academic misconduct will not be tolerated and will be handled as per University guidelines. You are required to be familiar with the following information:

http://www.concordia.ca/students/academic-integrity.html