COURSE SYLLABUS

CSC14120 – Parallel Programming

1. GENERAL INFORMATION

Course name:	Parallel Programming
Course name (in Vietnamese):	Lập trình song song
Course ID:	CSC14120
Knowledge block:	
Number of credits:	4
Credit hours for theory:	45
Credit hours for practice:	30
Credit hours for self-study:	90
Prerequisite:	
Prior-course:	
Instructors: Phạm Trọng Nghĩa	

2. COURSE DESCRIPTION

The course is designed to provide students with knowledge and skills of parallel programming on GPU (Graphics Processing Units, students will use free GPUs in Google Colab) with CUDA C/C++ (or CUDA for short); through this, students will develop a mindset of general parallel programming and will be able to quickly learn parallel programming on other hardware and programming languages. Concretely, students will learn about the differences between CPU and GPU, and how to write CUDA programs to take advantage of both CPU and GPU: sequential parts will run on CPU, massively parallel parts will run on GPU. Students will learn how to divide work for threads on GPU and coordinate these threads so that they will run in parallel and not interfere with each other. Students will also learn about optimization techniques to speed up CUDA programs further. Finally, student will be able to apply their learned knowledge and skills to parallelize and optimize an application which is slow if implemented sequentially.

3. COURSE GOALS

At the end of the course, students are able to

ID	Description	Program LOs
G1	Implement common tasks (vector addition, matrix multiplication, convolution, reduction, scan, histogram, sort, …) to run parallel on GPU using CUDA	1.2.1 1.3.1 1.7.3
G2	Apply knowledge of GPU parallel execution in CUDA to speed up a CUDA program	1.3.1 1.3.3 1.7.3 4.1
G3	Apply knowledge of GPU memories in CUDA to speed up a CUDA program	1.3.1 1.3.3 1.7.3 4.1
G4	Apply the optimization process to optimize a CUDA program	1.7.3 4.1
G5	Apply teamwork skills to complete final project	2.2

4. COURSE OUTCOMES

CO	Description	I/T/U
G1.1	Implement common tasks which is easy to parallelize (vector addition, matrix multiplication, convolution, …) to run parallel on GPU using CUDA	UT
G1.2	Implement common tasks which is not easy to parallelize (reduction, scan, histogram, sort, …) to run parallel on GPU using CUDA	UT
G2.1	With a kernel (the function will be executed in parallel on GPU by many threads): (i) identify possible problems when	T

	block size is changed (a block: a group of threads), and choose the optimal block size; (ii) identify whether the kernel has serious warp divergence problem, if yes then propose a solution and implement it	
G2.2	With a program: identify whether CUDA streams can be used to speed up the program, if yes then implement it	T
G3.1	With a kernel: identify whether CUDA memories can be used to speed up the kernel, if yes then implement it	T
G3.2	With a kernel: identify whether the kernel has the problem of accessing global/shared memory inefficiently, if yes then propose a solution and implement it	T
G4.1	With a program: identify which parts should be prioritized to optimize	T
G5.1	Apply teamwork skills to complete final project	U

5. TEACHING PLAN

THEORY

ID	Topic	Course outcomes	Teaching/Learning Activities (samples)	Assessments
1	Introduction to CUDA; example: vector addition, matrix multiplication, convolution, … (2 weeks)	G1.1	Lecturing, demo, Q&A	HW1, FP
2	GPU parallel execution in CUDA; example: reduction, vector addition, … (3 weeks)	G1.2 G2.1 G2.2	Lecturing, demo, Q&A	HW2, FP
3	Types of GPU memories in CUDA; example: reduction, convolution, …	G3.1	Lecturing,	HW3, FP

	(2 weeks)	G3.2	demo, Q&A	
4	Example: scan, histogram, sort (3 weeks)	G1.2 G2.1 G3.1 G3.2	Lecturing, demo, Q&A	HW4, FP
5	Optimizing a CUDA program; additional topics in parallel programming; final project (1 weeks)	G4.1 G5.1	Lecturing, demo, Q&A	FP

LABORATORY

ID	Topic	Course outcomes	Teaching/Learning Activities (samples)	Assessments
1	Coding environment setup		Q & A on Moodle	
2	HW1 guide	G1.1	Q & A on Moodle	HW1
3	HW2 guide	G1.2 G2.1 G2.2	Q & A on Moodle	HW2
4	HW3 guide	G3.1 G3.2	Q & A on Moodle	HW3
5	HW4 guide	G1.2 G2.1	Q & A on Moodle	HW4

		G3.1 G3.2		

6. ASSESSMENTS

ID	Topic	Description	Course outcomes	Ratio (%)
HW	**Homework**			**50%**
HW1	Introduction to CUDA	Implement CUDA programs to convert a RGB image to a grayscale one, and to blur an image	G1.1	12.5%
HW2	GPU parallel execution in CUDA	Implement a CUDA program to compute the sum of an array; apply knowledge of parallel execution in CUDA to analyze this program and then improve it	G1.2 G2.1 G2.2	12.5%
HW3	Types of GPU memories in CUDA	Implement a CUDA program to blur an image; apply knowledge of memories in CUDA to speed it up	G3.1 G3.2	12.5%
HW4	Parallelizing common tasks which is not easy to parallelize	Implement CUDA programs to do tasks: scan, histogram, sort	G1.2 G2.1 G3.1 G3.2	12.5%
FP	**Final Project**			**50%**

FP	Parallelizing and optimizing a slow application	Teacher can suggest applications, or let students choose their own applications	G1.1 G1.2 G2.1 G2.2 G3.1 G3.2 G4.1 G5.1	50%

7. RESOURCES

Textbooks

- Wen-Mei, W. Hwu, David B. Kirk, and Izzat El Hajj. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 2022.
- Cheng John, Max Grossman, and Ty McKercher. *Professional Cuda C Programming*. John Wiley & Sons, 2014.
- Lê Hoài Bắc, Vũ Thanh Hưng, Trần Trung Kiên. *Lập trình song song trên GPU*. NXB KH & KT, 2015

Others

- NVIDIA. *CUDA Toolkit Documentation*

8. GENERAL REGULATIONS & POLICIES

- All students are responsible for reading and following strictly the regulations and policies of the school and university.
- Students who are absent for more than 3 theory sessions are not allowed to take the exams.
- For any kind of cheating and plagiarism, students will be graded 0 for the course. The incident is then submitted to the school and university for further review.
- Students are encouraged to form study groups to discuss on the topics. However, individual work must be done and submitted on your own.