ECEC 413/622: Parallel Computer Architecture
Course Syllabus and Policies

1 Schedule

Term: Winter

Credits: 3.00

Contact Hours: 3 lecture hours per week.

2 Instructor Information

Instructor Name: Nagarajan Kandasamy

Office: 313D Bossone Phone: 215.895.1996 Email: kandasamy@drexel.edu

Office Hours: T 1-2 or by appointment

Teaching Assistant: Arghavan Mohammadhassani

Email: am4774@drexel.edu

Office hours: TBD.

3 Student Learning Information

- **Brief description of the content of the course (Course Catalog Description):** Students will develop parallel algorithms for shared-memory parallel computers using pthreads and OpenMP programming interfaces. The course will also teach students how to develop parallel algorithms for the graphics processing unit (GPU) and implement them using the CUDA programming interface. Students will be introduced to the architecture of a modern GPU and to the CUDA programming interface; and will gain proficiency in general purpose programming on the GPU.

- **Pre-requisites or Co-requisites:** The course will assume that students have taken an undergraduate-level computer architecture class; for example ECEC 355/412 if you are a Drexel student, or equivalent. Students must also be proficient in C programming.

- **Course Purpose within a Program of Study:** This course teaches students the skills necessary to take a computational problem and parallelize it in a way suitable for execution on multi-core CPUs and GPUs. Well-designed parallel code can provide significant execution-time speedup as well as energy efficiency over its serial counterpart.

- **Statement of Expected Learning:** This course consists of three 1-hour lectures per week as well as numerous take-home programming assignments.

- **Course Outcomes:** Upon completion of this course, students will be able to:

    1. Develop parallel programs using the pthreads library and OpenMP language features.

2. Understand CUDA threads and GPU memory hierarchy; and develop high-performance programs within the C/CUDA framework.
3. Consider the performance bottlenecks in parallel programs and optimize performance.

- **Student Outcomes (ABET):**

  (1) Ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.

  (5) Ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.

  (6) Ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

  (7) Ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

- **Drexel Student Learning Priorities:**

  - **Creative and Critical Thinking:** Uses divergent (e.g., generation of novel ideas, thinking out of the box, brainstorming) and convergent thinking (e.g., critical thinking, evaluation of ideas, quantitative and qualitative analysis, scientific reasoning) to generate novel and relevant ideas, strategies, approaches or products.
  - **Technology Use:** Make appropriate use of technologies to communicate, collaborate, solve problems, make decisions, and conduct research, as well as foster creativity and life-long learning.

4 Course Material

- Required textbook: D. Kirk and W-M. Hu, *Programming Massively Parallel Processors: A Hands-on Approach,* 3rd Edition Morgan Kaufmann, 2017.
- Lecture notes and code examples will be provided by the instructor.
- Reference textbooks:
  - S. Cook, *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs,* Morgan Kaufmann, 2013.
  - B. Chapman, G. Jost, and R. Van Der Pas, *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation),* MIT Press, 2007.
  - P. Pacheco, *Introduction to Parallel Programming,* Morgan Kaufmann, 2011.
  - Michael Kerrisk, *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*, No Starch Press, San Francisco, 2010.
  - Robert Love, *Linux System Programming*, 2nd Edition, O'Reilly Media, 2013.
  - D. Butenhof, *Programming with POSIX Threads,* Addison-Wesley, 1997.
- Students who have not taken ECEC 355 at Drexel should read through: *Computer Organization and Design: The Hardware/Software Interface,* by Patterson and Hennessy, focusing on datapath design, pipelines, and cache design. Older editions such as the 3rd edition are acceptable.

5 Course Schedule

The following outlines course coverage by term week:

- **Week 1:** Introduction; rise of multi-core machines; types of parallelism; shared memory and distributed memory machines; viability of large-scale parallelization; Amdahl's law and Gustafson's laws; writing parallel programs

  Reading: Lecture notes and Chapter 1 of textbook.

- **Week 2:** Introduction to pthreads; thread creation and management; race conditions; hardware support for mutex; semaphores; examples of data-parallel programs written using pthreads

  Reading: Lecture notes.

- **Week 3:** Developing multi-threaded programs using OpenMP; examples of data-parallel programs written using OpenMP; loop optimizations

  Reading: Lecture notes.

- **Week 4:** GPU micro-architecture; thread organization; thread scheduling; developing simple GPU programs

  Reading: Textbook chapters 2 and 3.

- **Week 5:** GPU memory hierarchy; memory-access efficiency; use of shared memory; constant memory; texture memory

  Reading: Textbook chapter 4.

- **Week 6:** Performance considerations when developing GPU code; memory-transfer considerations

  Reading: Textbook chapter 5.

- **Week 7:** Numerical considerations

  Reading: Textbook chapter 6.

- **Week 8:** Parallel patterns for reduction, convolution, prefix sum, and histogram computation

  Reading: Textbook chapters 7, 8, and 9.

- **Week 9:** Other CUDA features: unified memory; CUDA BLAS library; CUDA streams

  Reading: Lecture notes.

- **Time permitting:** Developing vector code using SSE/AVX units on the CPU.

6 Assignments, Assessments, and Graded Activities

- **Grade Breakdown:** Grading is based on take-home programming assignments.
  - 100% Programming assignments
    - 30%: Three Pthread programming assignments.
    - 30%: Three OpenMP programming assignments.
    - 40%: Four CUDA programming assignments.

- **Grading Scale:** The mapping between the percent grade and the final letter grade will be assigned as follows (may be curved):

  A: 90–100, A-: 85–89, B+: 80–84, B: 75–79, B-: 70–74, C+: 65–69, C: 60–64, C-: 55–59, D+: 50–54, D: 45–49, F: 00–44

7 Academic Policies

- **Course-Specific Policies**

  - **Homework Assignment Policy:** Homework is always to be submitted on or before the date on which it is due. Homework submitted after the due date will not be accepted. Homework must be submitted using Black Board Learn unless otherwise instructed. Assignments must never be submitted via e-mail.
  - **Examination Policy:** All exams in the course are closed-textbook and closed-"reading assignment material." Use of other books or any other material (such as solutions to homework assignments), however, is not permitted. Use of cell phones, laptops, tablets, PDAs, or any other device capable of wireless communication is prohibited. Exams will cover material discussed in the lectures, homework assignments, or textbook sections given as reading assignments. For example, the exams may include questions on material covered in class lectures or homework but not specifically covered in the textbook. Similarly, the exams may include any material covered in a section of the textbook given as a reading assignment but not specifically covered in the lectures or homework.
  - **Absentee Policy:** Absence from examinations will be excused only under extraordinary circumstances such as medical or family emergencies. A missed examination without prior approval and without legitimate reason will be graded at zero points. An absence will be excused only if the student is able to provide legitimate documentation (such as a physician note). An absence from an examination with prior approval will require the student to take an alternate exam at a later time. Special examinations will not be held earlier or on later dates to accommodate, for example, flight schedules for overseas vacations.

- **University Academic Policies:**

  - **Missed Classes:** Absence from class will be based on the University's absence policy. Please review the link here:

    http://drexel.edu/provost/policies/absence/.
  - **Academic Integrity, Plagiarism and Cheating Policy:** Each student is expected to complete all assignments independently unless otherwise explicitly instructed. It is unacceptable to copy another student's work or solutions from any other source. Submitted assignments will be checked for plagiarism using Stanford's MOSS plagiarism detection system (https://theory.stanford.edu/~aiken/moss/). Violators of this policy will be reported to the Office of Student Conduct and Community Standards (SCCS). Academic integrity violations could result in failure for the course or the assignment among other sanctions determined by the instructor. A second violation of the academic integrity policy will likely result in suspension. Please review the University's policy regarding academic integrity at:

http://drexel.edu/provost/policies/academic-integrity/ and
http://drexel.edu/studentlife/community_standards/studentHandbook/

- **Office of Equality and Diversity - Disability Resources:** Students requesting accommodations due to a disability at Drexel University need to request a current Accommodations Verification Letter (AVL) in the ClockWork database before accommodations can be made. These requests are received by Disability Resources (DR), who then issues the AVL to the appropriate contacts. For additional information, visit the DR website at drexel.edu/oed/disabilityResources/overview/ or contact DR for more information by phone at 215.895.1401, or by email at disability@drexel.edu.
- **Course Drop Policy:** Please review the University's policy at:
  http://drexel.edu/provost/policies/course-add-drop/
- **Course Withdrawal Policy:** Please review the University policy at:
  http://drexel.edu/provost/policies/course-withdrawal/
- **Course Change Policy:** The instructor reserves the right to modify the course, as necessary, during the term: including policies, evaluations, due dates, course content, schedule, assignments or requirements. All changes will be communicated in lecture and/or via the course page on Black Board Learn.
- **Weather, Emergencies, and University Closing:** University closing or delayed opening information will be posted on www.drexel.edu. In the event of the need to close or delay the daily opening of a campus, the University will provide notice via Web, telephone, and the DrexelALERT system. Closing or delayed opening information will be announced at 215-895-MELT (6358). The University determines whether to close or delay opening due to inclement weather, not the instructor. Therefore, please do not contact the instructor for this information.