CAS781 High-Performance Scientific Computing

Department of Computing and Software
McMaster University
Winter 2022

Instructor

Ned Nedialkov

Lectures

Tuesday	16:30–18:00, ITB 222
Thursday	13:30–15:00, ITB 222

Course Description

This course gives a basic introduction to parallel architectures and programming (distributed, shared memory, and GPU). We shall study how algorithms interact with these architectures, performance analysis, understanding compiler optimizations, and techniques for writing efficient code. Selected numerical methods will be presented. Current trends in HPC, such as mixed-precision computations will be discussed. The focus will be on the C/C++ languages. Interoperability with other languages (e.g. Fortran) will be addressed.

Required Materials

- Victor Eijkhout. Introduction to High-Performance Scientific Computing

Prerequisites

A course in numerical methods/scientific computing, good C/C++ programming skills, Linux, basic knowledge of computer architecture.

Grading

- 45% three assignments
- 55% project, 15% presentation, 40% report

Topics

- Programming strategies and software optimizations for high-performance
- Performance analysis and tools
- Understanding what compilers can do
- Build tools (make, cmake)
- Computer arithmetic
- Parallel computing
  - Message-passing interface (MPI)
  - OpenMP
  - GPU computing: OpenACC
- High-performance linear algebra
- Mixed-precision computing
- Numerical libraries and tools for high-performance computing

Resources

- MPI
  - MPI standard
  - MPI: The Complete Reference
  - MPI tutorial
- Linux
  - Unix/Linux
  - Shell programmng

Reading

This schedule is tentative and subject to change.

1 Introduction. Single-processor computing

- Read pp. 10–72.

- Study makefiles. I recommend these slides and the first two chapters of Managing Projects with GNU Make.

2 Introduction to parallel computing

3 Vectorization

4 Computer arithmetic

5 MPI Basics

6 Collective communications

7 Speedup, efficiency, scalability Read also 2.2 Theoretical concepts.

8 Parallel program design. Tasks, critical path

9 Example of algorithm analysis

10 Distributed matrix multiply

11 OpenMP basics

12 OpenMP basics

13 Gauss elimination

14 Iterative refinement

15 Sparse matrices

16 Sparse Gauss elimination

17 OpenACC

18 OpenACC

Course Policies

Academic Integrity

You are expected to exhibit honesty and use ethical behaviour in all aspects of the learning process. Academic credentials you earn are rooted in principles of honesty and academic integrity. It is your responsibility to understand what constitutes academic dishonesty. Academic dishonesty is to knowingly act or fail to act in a way that results or could result in unearned academic credit or advantage. This behaviour can result in serious consequences, e.g. the grade of zero on an assignment, loss of credit with a notation on the transcript (notation reads: "Grade of F assigned for academic dishonesty"), and/or suspension or expulsion from the university. For information on the various types of academic dishonesty please refer to the Academic Integrity Policy. The following illustrates only three forms of academic dishonesty:

- plagiarism, e.g. the submission of work that is not one's own or for which other credit has been obtained
- improper collaboration in group work
- copying or using unauthorized aids in tests and examinations

Authenticity/Plagiarism detection

Some courses may use a web-based service (Turnitin.com) to reveal authenticity and ownership of student submitted work. For courses using such software, students will be expected to submit their work electronically either directly to Turnitin.com or via an online learning platform (e.g. Avenue to Learn, etc.) using plagiarism detection (a service supported by Turnitin.com) so it can be checked for academic dishonesty. Students who do not wish their work to be submitted through the plagiarism detection software must inform the Instructor before the assignment is due. No penalty will be assigned to a student who does not submit work to the plagiarism detection software. All submitted work is subject to normal verification that standards of academic integrity have been upheld (e.g., on-line search, other software, etc.). For more details about McMaster's use of please go to Turnitin.com.

Faculty Notices

"The Faculty of Engineering is concerned with ensuring an environment that is free of all discrimination. If there is a problem, individuals are reminded that they should contact the Department Chair, the Sexual Harassment Officer or the Human Rights Consultant, as the problem occurs."