CS 475 / 575 -- *Parallel Programming*

Ecampus Edition

Spring Quarter 2018

http://cs.oregonstate.edu/~mjb/cs575e





Handouts Projects Grades VHR Piazza

This page was last updated: April 2, 2018

Things to Know About:

- Important 3/27/2018 Please go here and fill out the class web form. This seeds my email list, my grading spreadsheet, and helps me set Office Hours.

- Quiz #1 Quiz #1 consists of you answering the "Why Are You Here? Give me a quick sentence on why you are taking this course:" question.

- The most recent gcc/g++ is version 7.30 . It supports OpenMP 4.0. On OSU Linux machines, it is hiding at **/usr/local/common/gcc-7.3.0.**
  Just using "gcc" or "g++" gets you version 4.8.5 instead.

- As students, you can get the Intel compilers for your own machine for free. Go to https://software.intel.com/en-us/qualify-for-free-software/student to learn more.

- The instructions for using the Intel compilers (icc and icpc) on the OSU machines flip and flop are here.

- Want to see if your Windows system supports OpenCL? Grab these two files:
  - PrintInfo.exe
  - printinfo.cl

  (You might have to change permissions on PrintInfo.exe to make it executable.)

Then double-click on PrintInfo.exe -- it will print out the OpenCL characteristics of your system. If it does all of that, you have OpenCL.
The source is in printinfo.cpp and cl.h

- Want to see if your Linux system supports OpenCL? Grab these two files:
  - printinfo
  - printinfo.cl

  Then execute printinfo -- it will print out the OpenCL characteristics of your system. If it does all of that, you have OpenCL.
  The source is in printinfo.cpp and cl.h
  Go here to see what the rabbit machine produced when it ran *printinfo*.

Important Dates

April 2	Spring Quarter classes start
Thursday, May 3	Test #1 opens at 10:00 AM PDT
Sunday, May 6	Test #1 closes at 10:00 AM PDT
May 13	Mothers Day
May 28	Memorial Day
Thursday, June 14	Test #2 opens at 10:00 AM PDT
Sunday, June 17	Test #2 closes at 10:00 AM PDT
June 15	EECS Graduation Celebration, 3:30
June 16	OSU Graduation

Note that Test #1 and Test #2 will require a proctor. Ecampus recommends that you complete your proctor requests during week #2 of the term,

---

Here are Some Background Resources for You

Over the last couple of months, various faculty have been putting together mini-tutorials on topics that we (apparently) expect you to magically know, without us ever actually showing you. Here they are. We hope they help.

- Video on C/C++ Debugging
- Video on C/C++ pointers
- Video on C/C++ pointers as applied to strings
- Notes on Solving Problems and Debugging
- Notes on Learning *vim*
- Walking through the Microsoft Visual Studio download/install process from Microsoft Imagine
- Using Kaltura to record turn-in videos for class assignments.

What We Will Be Doing This Quarter

The goals of this course are to leave you "**career-ready**" (i.e., both **work-ready** and **research-ready**) for tasks that require desktop parallelism, both on a CPU and on a GPU.

CS 475/575 topics include:

- Parallel computing: types, limitations
- Moore's Law
- Amdahl's Law
- OpenMP
- Synchronization issues in parallel computing
- Cache issues in parallel computing
- SIMD on CPUs
- GPU programming
- OpenCL and CUDA: data parallel + SIMD
- Xeon Phi

Prerequisites

This course will use C/C++ for its programming. Already being comfortable with function calls, arrays, for loops, structures, arrays of structures, structures of arrays, pointers, trees, and linked lists is important. It is strongly suggested that you not use this class as an opportunity to learn C for the first time.

Many of the assignments can be done on Linux systems. It would be good if you already know how to use the Linux command line and know at least one Linux-based editor (vim is good.

Some knowledge about computer architecture (e.g., cores, cache) would be a plus, but not critical.

Learning Outcomes

On completion of the course, students will have demonstrated the ability to:

1. Explain the clockspeed limitations of computing using physics and Moore's Law
2. Explain the limitations of parallel computing using Amdahl's Law
3. Demonstrate "parallel thinking" in program design
4. Explain the difference between TLP, DLP, and SIMD
5. Demonstrate the ability to program parallel algorithms in TLP, DLP, and SIMD.
6. Characterize what types of problems are best able to be parallelized
7. Characterize different parallel programming patterns and what types of problems they best address
8. Chacacterize how cache issues affect parallel performance
9. Demonstrate the proper use of synchronization to avoid race conditions and deadlock
10. Characterize the benefits of using a CPU versus using a GPU for parallel programming
11. Characterize the benefits of using a GPU versus using a CPU for parallel programming

In addition, those taking this course as CS 575 will also have deminstrated the ability to:

1. Read a parallel-programming-related research paper and write a 5-page analysis paper of it. (I will make some of these available for you, or you can propose your own. It has to be a real research paper, though.)

Professor

The class is being taught by Professor Mike Bailey.

Office:	Kelley 2117
Phone:	541-737-2542
E-mail:	mjb@cs.oregonstate.edu
URL:	http://cs.oregonstate.edu/~mjb
WebEx Room for Office Hours:	https://oregonstate.webex.com/meet/mjboregonstate.edu

Prof. Bailey Office Hours (tentative):

I will hold live Office Hours via my WebEx room. It's site is: https://oregonstate.webex.com/meet/mjboregonstate.edu , but it's only open when I am there.

My Office Hours times are:

Sundays	5:30 - 7:00 PM PDT
Wednesdays	5:30 - 7:00 PM PDT
Fridays	5:30 - 7:00 PM PDT

TAs

The TAs for this course are:

Wenbo Hou
houw@oregonstate.edu

Matthew Meyn
meynm@oregonstate.edu

Paris Kalathas
kalathap@oregonstate.edu

Yifan Shen
shenyif@oregonstate.edu

The Virtual Hand Raise (VHR)

I recognize that it sometimes takes a certain amount of courage to ask a question in class. But, the worst thing of all is to not ask! So, this class also uses something called the *Virtual Hand Raise*. Click here to get into it. It will allow you to send me a question or comment, *completely anonymously*. I will answer questions submitted this way by email to the class or in class.

Textbook

There is no required textbook for this course. We will use notes that will be made available on the class web site.

There is a free Parallel Programming book available. Click here to get to it. I'm not sure how much it helps for this class, but it is certainly interesting to look through.

Grading

CS 475/575 will be graded on a fill-the-bucket basis. There will be 8 projects, 10 quizzes, and two tests. You get to keep all the points you earn.

In addition, the CS 575 people have an extra 100-point project.
The CS 475 people automatically get 100 points for this project, even though they didn't have to do it.

Your final grade will be based on your overall class point total. Based on an available point total of **1100**, grade cutoffs will be no higher than:

Grade cutoffs will be no higher than this.

Project Turn-In Procedures

- Your project turnins will all be electronic.
- Your electronic turnin will be done at http://engr.oregonstate.edu/teach and will consist of:
  1. All source files (.cpp, .cl)
  2. A report in PDF format.

  Electronic submissions are due at 23:59:59 on the listed due date.

- Your PDF report will include:
  1. A title area, including your name, your email address, the project number, and project name.
  2. Any tables or graphs to show your results.
  3. An explanation of what you did and *why it worked the way it did*. Your submission will not be considered valid unless you at least attempt to explain why it works the way it does.

  You can turn in all of your files, if you want, in a .zip file *except the PDF report*. Please leave the PDF report out of the .zip file. (I have a script that vacuums up all your PDF files so I can read them as one big PDF. I can get your project grades back to you faster this way.)

- Your project will be graded and the score posted to the class web page.
  If you want to know why you did not receive full credit, send me an email asking about it, or see me during Office Hours.

Bonus Days and Late Assignments

Projects are due at 23:59:59 on the listed due date, with the following exception:

Each of you has been granted **five** Bonus Days, which are no-questions-asked one-day extensions which may be applied to any project, subject to the following rules:

1. No more than 2 Bonus Days may be applied to any one project

We will grade project assignments when they are two days past due:

1. If your assigment is late, but you have enough Bonus Days left to expend on it, then we will just grade it as normal, and adjust your expended Bonus Days Late number on your grade sheet.
2. If you don't have enough Bonus Days Left to expend on it, your grade is a 0.
3. If your project is 3 or more days late, your grade is a 0, no matter how many Bonus Days you have left.

Downloadable Files

None yet.

Class Rules

- **Computer Security:** We take computer security very seriously. Please use intelligently-chosen passwords and protect them. Anyone caught abusing the system or causing deliberate damage will be asked to drop the class and the matter will be turned over to the Dean's Office.

- **Working Together:** It is OK to work together by sharing information and teaching each other, but each person's projects *must be original work with no sharing of code*.

- **Cheating:** You are expected to do your own work. Sharing code is considered cheating. Anyone caught cheating will fail this class, and the matter will be turned over to the Dean's Office.

Students With Disabilities

Accommodations for students with disabilities are determined and approved by Disability Access Services (DAS). If you, as a student, believe you are eligible for accommodations but have not obtained approval please contact DAS immediately at 541-737-4098 or at http://ds.oregonstate.edu.. DAS notifies students and faculty members of approved academic accommodations and coordinates implementation of those accommodations. While not required, students and faculty members are encouraged to discuss details of the implementation of individual accommodations.

Religious Holidays

Oregon State University strives to respect all religious practices. If you have religious holidays that are in conflict with any of the requirements of this class, please see me immediately so that we can make alternative arrangements.

Other Useful Online Parallel Programming Information

- OpenMP:
  - Industry-wide OpenMP information: http://www.openmp.org

  - OpenMP 4.0 Quick Reference Card: http://openmp.org/mp-documents/OpenMP-4.0-C.pdf
  - OpenMP 4.5 Quick Reference Card: http://www.openmp.org/mp-documents/OpenMP-4.5-1115-CPP-web.pdf

- OpenCL:
  - Industry-wide OpenCL Information: http://www.khronos.org/opencl
  - OpenCl Reference (~man) Pages http://www.khronos.org/registry/cl/sdk/1.0/docs/man/xhtml
  - OpenCL 1.1 Quick Reference Card: http://cs.oregonstate.edu/~mjb/cs575/opencl-1-1-quick-reference-card.pdf
  - OpenCL 1.2 Quick Reference Card: http://cs.oregonstate.edu/~mjb/cs575/opencl-1-2-quick-reference-card.pdf
  - OpenCL 2.0 Quick Reference Card: https://www.khronos.org/files/opencl20-quick-reference-card.pdf
  - NVIDIA's OpenCL example code: http://developer.nvidia.com/opencl-sdk-code-samples
  - Prof. Bailey's OpenCL status error code printer: errorcodes.cpp

- Xeon Phi:
  - Videos of a Xeon Phi workshop put on at OSU by Colfax International: https://www.youtube.com/playlist?list=PLkjf1FcmOKlghScT9ajvG5SwHEQ_yhjL2

- SSE:
  - Good explanation of SSE arithmetic: http://www.tommesani.com/SSEArithmetic.html

- Online copy of the book: Ian Foster, *Designing and Building Parallel Programs*, Addison-Wesley, 1995. http://www.mcs.anl.gov/~itf/dbpp/

- Tutorials from Lawrence Livermore National Labs:
  - Parallel computing: https://computing.llnl.gov/tutorials/parallel_comp/

- *vi* Quick Reference Cards
  - http://tnerual.eriogerg.free.fr/vimqrc.pdf
  - http://pangea.stanford.edu/computing/unix/editing/viquickref.pdf

- Association of Computing Machinery http://www.acm.org
  - Special Interest Group on High Performance Computing http://www.sighpc.org
  - Special Interest Group on Computer Graphics http://www.siggraph.org