CS532 - High Performance Distributed Computing

Tuesday/Thursday 11:00am-12:15pm, Streibel Hall Room 115

This course introduces the basics of cluster and supercomputing with MPI (the message passing interface) and the basics of GPU computing with OpenCL.

News

- 2017-08-21 - Most account issues can be resolved by visiting the Computer Science Department Wiki (http://wiki.cs.und.edu).
- 2017-08-21 - You can get to the Computational Research Center (CRC)'s website here (http://crc.und.edu/). It has instructions for compiling, transferring files and running things on the large campus clusters.

Contact Information

Email and Phone:

tdesell@cs.und.edu (mailto:tdesell@cs.und.edu)
701-777-3477

Email is the best and most reliable way to contact me.

Office Hours:

Streibel Hall 220
To be decided and by appointment.

Address

Department of Computer Science
University of North Dakota
Streibel Hall Room 220
3950 Campus Road Stop 9015
Grand Forks, North Dakota 52802-9015

Date	Topic	Lecture Notes/Lab Files	Reading
2017-08-22	Course Information		
2017-08-24	Introduction to C++		
2017-08-29	Parallel Hardware - Part 1 *Chapters 2.1-2.3 (Pacheco)*	Parallel Hardware [pdf] (./lectures/parallel_hardware.pdf)	
2017-08-31	Parallel Hardware - Part 2 *Chapters 2.3, 2.6 (Pacheco)*		
2017-09-05	MPI - Part 1	MPI [pdf] (./lectures/mpi.pdf) mpi_hello.cxx (./files/mpi_hello.cxx) mpi_bcast_example.cxx (./files/mpi_bcast_example.cxx) mpi_scatter_gather_example.cxx (./files/mpi_scatter_gather_example.cxx) mpi_scatterv_gatherv_example.cxx (./files/mpi_scatterv_gatherv_example.cxx)	
2017-09-07	Homework 1 Introduction Lab		
2017-09-12	MPI - Part 2	mpi_reduce_example.cxx (./files/mpi_reduce_example.cxx) mpi_allgather_example.cxx (./files/mpi_allgather_example.cxx)	
2017-09-14	**Guest Lecture: David Apostal - Vectorizing a Geological Simulation**	auto-compiler-vect.pdf (./lectures/auto-compiler-vect.pdf) auto-compiler-vect.pptx (./lectures/auto-compiler-vect.pptx)	
2017-09-19	MPI - Part 3	using_hodor.pptx (./lectures/using_hodor.pptx)	
2017-09-21	Lab		
2017-09-26	MPI - part 4	mpi_process_ring.cxx (./files/mpi_process_ring.cxx) mpi_asynch_busy_process_ring.cxx (./files/mpi_asynch_busy_process_ring.cxx) mpi_asynch_process_ring.cxx (./files/mpi_asynch_process_ring.cxx) unordered_map_test.cxx (./files/unordered_map_test.cxx)	
2017-09-28	Lab	mpi_asynch_mw.cxx (./files/mpi_asynch_mw.cxx)	
2017-10-03	**Guest Lecture - David Apostal (Vectorization Pt. 2)**		
2017-10-05	Lab		
2017-10-10	**Test 1 Review**		
2017-10-12	**Test 1**		
2017-10-17	GPU Computing with CUDA - Part 1 (Basics)	CUDA 1 [pdf] (./lectures/cuda.pdf) vector_sum.readme (./files/vector_sum.readme) vector_sum_gpu.cu (./files/vector_sum_gpu.cu) vector_sum_gpu_v2.cu (./files/vector_sum_gpu_v2.cu) vector_sum_gpu_v3.cu (./files/vector_sum_gpu_v3.cu) cuda_device_query.cu (./files/cuda_device_query.cu)	
2017-10-19	**No Class - Travel (Potential Guest Lecture)**		
2017-10-24	**No Class - Travel (Potential Guest Lecture)**		
2017-10-26	**No Class - Travel (Potential Guest Lecture)**		
2017-10-31	GPU Computing with CUDA - Part 1 (continued)		
2017-11-02	Lab		
2017-11-07	GPU Computing with CUDA - Part 2 (Grids, Blocks, Warps, Threads)	CUDA 2 [pdf] (./lectures/cuda_2.pdf)	
2017-11-09	Lab		

Date	Topic	Lecture Notes/Lab Files	Reading
2017-11-14	GPU Computing with CUDA - Part 2 (continued)		
2017-11-16	Lab		
2017-11-21	GPU Computing with CUDA - Part 3 (Synchronization)	CUDA 3 [pdf] (./lectures/cuda_3.pdf)	
2017-11-23	**No Class - Thanksgiving Recess**		
2017-11-28	GPU Computing with OpenCL (Basics)	OpenCL 1 [pdf] (./lectures/opencl_1.pdf) opencl_matrix_add.zip (./files/opencl_matrix_add.zip)	
2017-11-30	Lab		
2017-12-05	Test 2 Review		
2017-12-07	Test 2		

Course Description

This course introduces the basics of cluster and supercomputing with MPI (the message passing interface) and the basics of GPU computing with OpenCL and OpenCL. This course will also review current research being done in the areas of high performance distributed computing and GPU computing.

Objectives

At the completion of this course, students should be proficient in GPU computing and programming in MPI; as well as knowledgeable of the current research being done in the area of high performance distributed computing and GPU computing. Main objectives include:

1. Introduce super/cluster computing with MPI.
2. Introduce GPU computing with OpenCL.
3. Examine a wide range of current topics in high performance distributed computing.
4. Participate in a research project to gain expertise in HPDC.

Outcomes

Students will gain knowledge/understanding of the following:

1. C++ for high performance computing
2. Vector Programming
3. MPI and super/cluster computing.
4. GPU computing with OpenCL.
5. Cluster and GPU architectures.

Students will acquire the ability to do the following:

1. Develop high performance programs.
2. Create and debug programs in MPI and OpenCL.
3. Run applications on clusters and GPUs.

Pre-Requisites

None.

Text

The following texts are suggested for this course:

- *An Introduction to Parallel Programming*, Peter S. Pacheco.
- *Programming Massively Parallel Processors*, David B. Kirk and Wen-mei W. Hwu.

Grading

The course grade will consist of one group research project, one research paper presentation, four programming assignments and one test. The grade will be calculated as follows:

- 80% - Programming Assignments:
  - 10% - Vector Programming Assignment

    - 10% - MPI Programming Assignment 1
    - 10% - MPI Programming Assignment 2
    - 10% - MPI Programming Assignment 3
    - 10% - OpenCL Programming Assignment 1
    - 10% - OpenCL Programming Assignment 2
    - 10% - OpenCL Programming Assignment 3
    - 10% - OpenCL Programming Assignment 4
- 10% - Test 1
- 10% - Test 2

There will be the following grade distribution:

- [90 - 100]: A
- [80 - 89.9999...]: B
- [70 - 79.9999...]: C
- [65 - 69.9999...]: D

Academic Integrity

The development of the individual problem solving skills needed for computer programming is one of the major objectives of this course. **Students are to work independently of each other in completing the programming assignments. Any exception to this rule will require documentation signed by me allowing the collaborative work.**

If you need help, you are welcome to consult with your instructor, your teaching assistant, or the staff of the department's Instructional Help Desk in 109A Streibel Hall. A submission of source code that you did not develop or homework assignments that was not your individual writing will be treated as plagiarism. These assignments will receive zero points and you may be referred to the Associate Dean of Student Life as a case of Scholastic Dishonesty.

Attendance

Class attendance and lab attendance are required. Any student missing more than 6 classes without a doctors excuse will fail the course. The classroom is the primary venue for course material, announcements, and other information relevant to the course. An on-line course management system may be used to make some information available to students, but this is intended to enhance, not replace, classroom interaction.

Homework and Lab Submission

Code for homeworks and labs **must** be commented and properly formatted (see different coding styles (http://en.wikipedia.org/wiki/Indent_style), I prefer 1TBS) or points will be taken away. The final homework submission must be submitted through Moodle (http://moodle.cs.und.edu/). Each homework will list its grading criteria.

Late assignments will have their grades penalized by 15% the first day, and 30% the second day. No assignments will be accepted more than two days late. Homeworks are to be done individually (see Academic Integrity) and may involve a significant amount of programming, so start them early.

Lab Policies

All lab assignments must be completed by the end of the lab session. Any exception will require proper excuse with permission granted before the end of the lab session. Partial credit may be given to incomplete work.

Any issues related to the machines in the computer labs can be sent to cslabs@cs.und.edu (mailto:cslabs.cs.und.edu).

Students with Disabilities

Upon request, the Computer Science Department will provide reasonable accommodations for students with disabilities as specified in the policies of the UND office of Disability Services for Students (DSS). You must contact your instructor to request and arrange accommodations.