CSCE569 Parallel Computing, Spring 2018

Department of Computer Science and Engineering, University of South Carolina

---

Syllabus: Basic Information | Textbooks | Description | Objectives | Delivery Structure | Grading Scheme | Policies and Procedures | Course Flyer

Schedule, Lecture Notes and Assignments | Resources for Learning and Exercise | Test Topics and Sample Questions

Hardware Resources and Development | Acknowledgement

Basic Information

- Website: https://passlab.github.io/CSCE569/
- Meeting Time: 9:40 AM - 10:55AM Monday and Wednesday; Jan 16, 2018 - Apr 30, 2018. Check University Academic Calendar for other important dates and deadlines.
- Class Room: 2A15, Swearingen Engineer Center, 301 Main St, Columbia, SC 29208
- Instructor: Yonghong Yan, http://cse.sc.edu/~yanyh, yanyh@cse.sc.edu
    - Office: Room 2211, Storey Innovation Center (Horizon II), 550 Assembly St, Columbia, SC 29201
    - Tel: 803-777-7361
    - Office Hours: 11:00AM - 12:30PM Monday Wednesday in classroom or my office, or by appointment

---

Textbooks

- **Required**: Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, Introduction to Parallel Computing (2nd Edition), PDF, Amazon, cover theory, MPI and OpenMP introduction
- **Recommended**: John Cheng, Max Grossman, and Ty McKercher, Professional CUDA C Programming, 1st Edition 2014, PDF, Amazon.
- **Reference book for OpenMP**: Barbara Chapman, Gabriele Jost, and Ruud van der Pas, Using OpenMP: Portable Shared Memory Parallel Programming, 2007, PDF, Amazon.
- **Reference book for MPI**: Choose from Recommended Books for MPI

Course Description

Official Course Catalog Description

Architecture and interconnection of parallel computers; parallel programming models and applications; issues in high-performance computing; programming of parallel computers. Prerequisites: knowledge of programming in a high-level language; MATH 526 or 544 3.000 Credit hours 3.000 Lecture hours

Outline

Fundamentals and programming practices for parallel computing on parallel computing systems including general purpose multicore CPU, Graphic Processing Unit (GPU) manycores and high performance computing (HPC) clusters. **Topics include**:

- Principles of parallel algorithm design and performance analysis;
- Processor, memory and interconnection architectures of multicore CPU, NVIDIA GPU manycores, and HPC clusters;
- Parallel programming using OpenMP shared memory and threading model, CUDA threading and offloading model, and MPI distributed memory model;
- Parallelization and performance optimization techniques using computation kernels;

The course content includes a significant amount of programming exercises to create performant code using C/C++ programming language on Linux environment.

Overview

In high performance computing (HPC) systems and enterprise systems, parallel and distributed computing have been the approaches for several decades to deliver the performance needed for large-scale scientific and engineering simulation, and for big data analysis and machine learning. Today, parallel computing capability are available in the computing devices we use daily. Multicores CPUs are widely used in laptop, desktop, smartpad and phone, and some of them have manycore GPUs. In this course, we will study the processor, memory and interconnection architectures of modern CPU, GPU and HPC clusters, learn to design high performance parallel algorithms, and develop parallel programs using OpenMP, CUDA and MPI programming models. The course content includes ~40% theory and fundamentals, and ~60% programming and exercises.

Prerequisites, and Prior Programming Skills and Knowledge (Required or Good to Have):

- Prerequisites: MATH 526 or 544 is the hard requirement for the course.
- Minimum requirements include good reasoning and analytical skills and familiarity with C/C++ programming, e.g. the use of pointer, array, struct, function pointer, and library for memory allocation and de-allocation (malloc and free).
- Familiarity with Linux environment will be important for the assignments, including the use of SSH to access a remote Linux machine, and the use of vim/Emacs editor and GNU compiler (gcc) from Linux terminal.
- Knowledge of computer architecture (memory hierarchy, cache, virtual address) and data structures will be necessary for preforming well for the class.
- Knowledge of programming languages and compilers, and operating systems will also help.

[Survey for prerequisites, and prior programming skills and knowledge]

Learning Objectives

Following completion of the course, students should be able to:

1. Describe benefits and applications of parallel computing.
2. Explain architectures of multicore CPU, GPUs and HPC clusters, including the key concepts in parallel computer architectures, e.g. shared memory system, distributed system, NUMA and cache coherence, interconnection
3. Understand principles for parallel and concurrent program design, e.g. decomposition of works, task and data parallelism, processor mapping, mutual exclusion, locks.
4. Write parallel program using OpenMP, CUDA, MPI programming models.
5. Perform analysis and optimization of parallel program.

Course Delivery Structure

There will be two 75-minute lectures per week, which constitutes 100% of the course lecture delivery. There will be no separate labs, and there are no plans for distance learning.

Course Exams and Assignments

There are two exams and four assignments. The two exams are close books/notes. The contribution of each to your final grades are shown below. Questions with bonus points may be given.

- Two exams (40%):
  - **Midterm (15%, March 7th Wednesday during class)**
  - **Final Exam (25%, May 2nd Wednesday, 9:00AM - 11:30AM)**
- Four assignments (60%):
  1. Sequential implementation (10%)
  2. OpenMP (10%)
  3. MPI and Hybrid MPI/OpenMP/GPU (20%)
  4. GPU and CUDA (20%)

Grading Scheme

- Two exams: 40% (15% + 25%)
- Four assignments: 60% (10% + 10% + 20% + 20%)

Mapping Between Letter Grade and Percentage Grade:

Letter	Percentage
A	90-100
B+	86-90
B	80-86
C+	76-80
C	70-76
D+	66-70
D	60-66
F	0-60

Schedule, Topics, Lecture Notes and Assignments

- Assignments need to be submitted to https://dropbox.cse.sc.edu. Email or hard-copy will NOT be considered.
- Assignment due at 11:55PM on the due date, cut-off date is 72 hours after the due date/time. Check Policies and Procedures for details.
- **Topics, Lectures Notes and Assignments will be updated as the class progresses.**

Week	Date	Week date	Class	Topics and Lecture Notes	Resources	Assignment
1	01/15	Monday		MKL Jr. Day		
	01/17	Wednesday	1	Introduction	Test Topics and Sample Questions	
2	01/22	Monday	2	Introduction		
	01/24	Wednesday	3	Review of C programming, Compiler, Makefile, Linux and SSH access	Linux and C, sum.c, axpy.c, mm.c, matvec.c	Assignment 1, Due 02/09 Friday
3	01/29	Monday	4	OpenMP Parallel	omp_hello.c, mm_openmp.c, sum_openmp.c	
	01/31	Wednesday	5	OpenMP Worksharing and Reduction		
4	02/05	Monday	6	OpenMP Data Environment, Tasking and Synchronization		
	02/07	Wednesday	7	OpenMP Performance Optimization		
5	02/12	Monday	8	Parallel Algorithm Design 01		Assignment 2, Due 03/02
	02/14	Wednesday	9	Dense Matrices and Decomposition	matmul-decompose.c	
6	02/19	Monday	10	Parallel Algorithm Design 02		
	02/21	Wednesday	11	Distributed Memory Systems		
7	02/26	Monday	12	Message Passing and MPI	MPI Exercises	
	02/28	Wednesday	13	MPI and Practices		
8	03/05	Monday	14	Parallel Program Metrics and Analysis		Assignment 3, Due 03/28
	03/07	Wednesday	15	**Midterm Exam**	Test Topics and Sample Questions	
9	03/12	Monday		*Spring Break*		
	03/14	Wednesday		*Spring Break*		

Week	Date	Week date	Class	Topics and Lecture Notes	Resources	Assignment
10	03/19	Monday	16	Parallel Program Scalability		
	03/21	Wednesday	17	C Function Pointer, PThread 1	pthread examples	
11	03/26	Monday	18	PThread 2 and Mutual Exclusion		
	03/28	Wednesday	19	Cilk/Cilkplus		
12	04/02	Monday	20	Cilk/Cilkplus and OpenMP Tasking		Assignment 4, Due 04/18
	04/04	Wednesday	21	Parallel Architecture and Thread Level Parallelism, Data Level Parallelism		
13	04/09	Monday	22	Memory Hierarchy and Cache Coherence		
	04/11	Wednesday	23	Manycore GPUs and CUDA		
14	04/16	Monday	24	CUDA Threading and Memory		
	04/18	Wednesday	25	CUDA Threading and Memory		
15	04/23	Monday	26	GPU/CUDA Streaming, Library and Tuning		
	04/25	Wednesday	27	OpenMP/OpenACC for offloading		
16	04/30	Monday	28	Hybrid MPI/OpenMP/CUDA		
	05/02	Wednesday		**Final Exam (9:00AM - 11:30AM)**	Test Topics and Sample Questions	

Acknowledgement

We greatly appreciate National Science Foundation's XSEDE program and Pittsburgh Supercomputing Center for supporting students' access to Bridges Supercomputer.

Policies and Procedures

Homework and Project

The homework and project exercises are chiefly for your own benefit. You may collaborate and consult outside sources freely when doing the homework, but you must tell me who you are collaborating with. Please remember, however, that the best way to master the material is to try the exercises on your own, then submit your own work to get feedback on it. (You will get credit even if you do not solve the problem, as long as you make an honest effort.)

Due Date and Late Policy

Each assignment has a due date and a cutoff date. Assignment due 11:55PM on the date, cut-off date is 72 hours after the due date/time. You have a total of 5 "slip days" throughout the semester that you can use at your discretion to turn in programming assignments past the posted due date. Slip days are used in whole day increments. Once your slip days are consumed, late programming assignments will be penalized at 50% per day. Assignment submission will not be accepted after the cutoff date.

Class Policies

Reading and lectures: The students are expected to read all assigned material before the lecture begins and review the material after the lecture.

Attendance Policy:

There is no required attendance policy, but if for some reason you cannot attend class, you are responsible for any material covered during your absence. Late arrivals must enter the classroom quietly and discreetly.

Exams:

Exams are given in class and are close-book/close-notes. No make-up exams will be given except under extreme circumstances (such as severe illness or death in the immediate family) in which case you must give me notice well before the exam if at all possible.

Academic Honesty:

Examination work and assignment are expected to be the sole effort of the student submitting the work. If a student collaborates with anyone on the homework (whether or not a student), all collaborators must be listed at the top of the first page. Students are expected to follow the Code of Student Academic Responsibility. Every instance of a suspected violation will be reported. Students found guilty of violations of the Code will receive the grade of F for the course in addition to whatever disciplinary sanctions are applied by the University.

Proper Use of Computing Resources:

Students are expected to be aware of the university policy on use of computing resources, including the Student Guidelines for Responsible Computing, as well as the college and departmental policies on proper use of computing resources. Every instance of a suspected violation will be reported.

Students With Disabilities:

Any student with a documented disability should contact the Office of Student Disability Services at 803-777-6142 to make arrangements for appropriate accommodations.