CSCI 4850/5850: High-Performance Computing

Catalog Description:

High-performance computing (HPC) refers to a specialized use of processor features, multiple cores, memory, graphic cards, accelerators, clusters, supercomputers, and everything from software to hardware to speed up computations. This course introduces state-of-the-art HPC technologies and parallel programming skills for them.

Class Meeting Time/Location:

Time: Tue, Thu 12:45pm-2:00pm
Location: 115 Ritter Hall (Linux Classroom)

Instructor:

Tae Hyuk (Ted) Ahn, PhD
Assistant Professor
Department of Computer Science
Program of Bioinformatics and Computational Biology
Email: ted.ahn@slu.edu
Office: 305 Ritter Hall (Lab: 302 Ritter Hall)
Phone: (314) 977-3633
Office Hour: Tue 10:00-11:30am, Wed 12:00-1:30pm, or by appointment

Credits/Pre-requisites:

Credits: 3 Credits
Pre-requisites: CSCI 2100 (required); CSCI 3100 and CSCI 3500 recommended.
(Exceptions if non-cs major students have strong programming skills and computer/numerical concepts with enthusiasm for learning HPC)
Overall requirements: Students will be expected to have knowledge of algorithms and data structures, and linear algebra at an undergraduate level. Students are expected to be proficient in either C, C++, or FORTRAN. Students should also be familiar with the Unix environment.

Course Description

High-performance computing refers to a specialized use and parallel programming of supercomputers, computer clusters, and everything from software to hardware to speed up computations. This course introduces state-of-the-art technologies and programming skills for the HPC. Main topics include basic HPC hardware architectures, parallel programming languages, new trends of HPC, and more. The contents of the course provide a balance of theoretical and practical aspects in parallel computing. A student of this course is expected to develop the right skills to design parallel applications and to effectively use modern HPC platforms.

Topical Outline:

- Introduce High-Performance Computing (HPC)
- Profiling applications for run time and parallel speed-up
- Brief introduction of computer organization and HPC architecture
- Shared-memory programming with OpenMP
- Distributed-memory programming with MPI

- MATLAB Parallel Computing Toolbox
- GPU programming with OpenACC (optional: CUDA)
- Cloud Computing and Apache Spark

Student Learning Outcomes:

After successfully complete this course, students are expected to:

- Know fundamental concepts and new trends of high-performance computing
- Optimize software to take advantage of processor feature
- Design, implement, and analyze with OpenMP for shared-memory system
- Design, implement, and analyze with MPI for distributed-memory programming
- Parallelize an existing application using an appropriate parallel programming paradigm
- Utilize MATLAB parallel computing toolbox
- Understand the concept and implement simple GPU programs with OpenACC (or CUDA)
- Learn cutting-edge cloud computing techniques using Amazon AWS and Apache Spark

Course Textbook and Resources:

No textbook is required for this course. Assigned slides/materials will be posted on class website (blackboard). However, several reference books are recommended as below:

- An Introduction to Parallel Programming by Peter Pacheco (Morgan Kaufman, 2011, ISBN:978-0123742605).
- Introduction to High Performance Computing for Scientists and Engineers by Georg Hager and Gerhard Wellein, 1st Edition (Chapman & Hall/CRC Computational Science , ISBN-13: 978-1439811924)
- Introduction to Parallel Computing by Ananth Grama, George Karypis, Vipin Kumar, and Anshul Gupta (Addison-Wesley, 2nd edition, 2003, ISBN:978-0201648652).
- Principles of Parallel Programming by Calvin Lin, and Larry Snyder (Addison-Wesley, 2008, ISBN:978-0321487902).Bioinformatics and Functional Genomics 2nd or 3rd Edition by Jonathan Pevsner (Hoboken, N.J. : Wiley-Blackwell, c2009)

SLU CS Git will be used to

- Assign and submit homework

Email with Instructor

Face-to-face contact in class and in office hours is most desirable. Yet email is a convenient form of communication as well. I try to respond to email promptly, including at least once each evening when possible.

If your question involves your progress on a current programming assignment, my response will be more informative if you can point out the specific problem you have encountered, and if I am able to see all of your source code. Therefore I strongly suggest that you either attach all relevant files to the email or submit preliminary versions of such files through git system.

Computer and Cell Phone Policy

Computers will be an integral part of this course, both inside and outside of class. However, out of courtesy to both the instructor and other students, please do not use the lab computers for non-class related activity. In particular, you do not need to be using a computer unless an exercise or in class activity requiring them is in progress.

You are unlikely to need cell phones during the course of lecture. Please ensure that your cell phone is set to vibrate or silent during lecture, and do not send text messages of any kind.

Grading:

Homework Assignments (50%)

We expect there to be a total of 6-10 written and programming homework assignments during the course. Maximum points of assignment could be different.

Exams (50%)

- Take-home Midterm Exam (20%), Tue 03/06/2018 – Thu 03/08/2018 (tentative)
- Take-home Final Exam (30%), Thu 05/03/2018 – Wed 05/09/2018 (tentative)

Submission of all take-home exams is mandatory. If you must miss an exam submission due, you must contact the instructor and submit a written request for a makeup exam prior to the exam. Your written request should include detailed information about the conflict and provide documentation of the conflict and any relevant contact information. Legitimate conflicts and excuses are limited to death or near-death instances in the immediate family, a student's illness that requires immediate doctor's care, a University sponsored event (not club sports) and regularly scheduled religious obligations. Excuses that will **NOT** be considered include personal travel arrangements, non-University sponsored events, a conflicting appointment, a previous illness that interfered with your study time or an illness that does not prevent you from coming to the exam. Unexcused exam absences will be counted as zero in the calculation of your final grade.

Extra Credit

Both homework and programming assignments will generally include a small extra credit challenge. Please notice, however, that the actual extra credit given for these challenges is relatively insignificant. Students who are seriously concerned about improving their overall grade would be best advise to focus all efforts on doing as well as possible on the required work and in preparing for exams.

Grading Scale:

Final grades for the course will be based on the following scale. The instructor reserve the right to make adjustments to grades based on overall performance in the course. There will be **no opportunity for "extra credit" to improve grades that have already been earned**. Bargaining for grades will not be tolerated.

Letter Grade	F	D	C-	C	C+	B-	B	B+	A-	A
%	<60	≥60 <70	≥70 <73	≥73 <77	≥77 <80	≥80- <83	≥83 <87	≥87 <90	≥90 <93	≥93

Late Policies

Late homework or programming assignments will suffer a penalty of 20% for every day they are late. For example, homework which is submitted between midnight and 11:59pm next day will be worth at most 80% of the total credit. After five days of the due, submission will get zero point as the formula. For the lab session, 1 point reduction for every day if they are late.

Regrade Requests

I am happy to regrade any assignments, programs, quizzes or exam problems which you think were unfair or incorrect. Please bring me the original assignment, plus a written explanation of your question or complaint, within two weeks of the time the paper in question is graded and returned to you.

Academic Integrity and Honesty

Academic integrity is honesty, truthful and responsible conduct in all academic endeavors. The mission of Saint Louis University is "the pursuit of truth for the greater glory of God and for the service of humanity." Accordingly, all acts of falsehood demean and compromise the corporate endeavors of teaching, research, health care, and community service via which SLU embodies its mission. The University strives to prepare students for lives of personal and professional integrity, and therefore regards all breaches of academic integrity as matters of serious concern.

The governing University-level Academic Integrity Policy can be accessed on the Academic Polices website (https://www.slu.edu/online/current-students/academic-policies.php#). A more detailed policy statement is given by the College of Arts & Science, also applying to this course (http://www.slu.edu/arts-and-sciences/student-resources/academic-honesty.php)

In addition to those general statements, we wish to discuss our policy in the context of this course. When it comes to learning and understanding the general course material, you may certainly use other reference materials and you may have discussions with other students in this class or other people from outside of this class. This openness pertains to material from the text, practice problems, general syntax and use of the C++ language or other computing tools.

However, when it comes to work that is submitted for this course, you are not to use or to search for any direct or indirect assistance from unauthorized sources, including but not limited to:

- other students in this class
- past students, whether from this school or other schools
- other acquaintances
- other texts or books
- online information other than that referenced by course materials

Acceptable sources of information include consultations with the instructor, teaching assistants, or members of organized tutoring centers on campus, as well as any materials explicitly authorized in an assignment. Even in these cases, if you receive significant help you should make sure to document both the source of the help as well as the extent.

On certain programming assignments, we will explicitly allow students to work in pairs. In this case, conversations between partners is both permissible and required. Furthermore, both students are expected to contribute significantly to the development of the submitted work. It is unethical to allow a partner to "sign on" to a submission if that partner did not significantly contribute to the work.

Any violations of these policies will be dealt with seriously. Penalties will apply as well to a student who is aiding another student. Any such violations will result in a minimum penalty of a zero on the given assignment that cannot be dropped, and severe or repeated violations will result in an immediate failing

grade in the course. Furthermore all incidents will be reported in writing to the Department and/or the Dean, as per the College procedure.

Students with Special Needs-Disability Services

In recognition that people learn in a variety of ways and that learning is influenced by multiple factors (e.g., prior experience, study skills, learning disability), resources to support student success are available on campus. Students who think they might benefit from these resources can find out more about:

- Course-level support (e.g., faculty member, departmental resources, etc.) by asking your course instructor.
- University-level support (e.g., tutoring/writing services, Disability Services) by visiting the Student Success Center (BSC 331) or by going to www.slu.edu/success.

Students who believe that, due to a disability, they could benefit from academic accommodations are encouraged to contact Disability Services at 314-977-8885 or visit the Student Success Center. Confidentiality will be observed in all inquiries.

Course instructors support student accommodation requests when an approved letter from Disability Services has been received and when students discuss these accommodations with the instructor after receipt of the approved letter.

Course Content Disclaimer

In this course, students may be required to read text or view materials that they may consider offensive. The ideas expressed in any given text do not necessarily reflect the views of the instructor, the English Department, the Writing Program, or Saint Louis University. Course materials are selected for their historical and/or cultural relevance, or as an example of stylistic and/or rhetorical strategies and techniques. They are meant to be examined in the context of intellectual inquiry of the sort encountered at the university level.

Writing Center

I encourage you to take advantage of the writing services in the Student Success Center; getting feedback benefits writers at all skill levels. Trained writing consultants can help with any writing, multimedia project, or oral presentation. During the one-on-one consultations, you can work on everything from brainstorming and developing ideas to crafting strong sentences and documenting sources. These services do fill up, so please make an appointment! For more information, or to make, change, or cancel an appointment, call 977-3484 or visit.

Tentative schedules

Week	Dates	Topic
1	01/16/2018	Welcome and Introduction
	01/18/2018	Architecture
2	01/23/2018	Optimization
	01/25/2018	Performance Analysis
3	01/30/2018	Programming with Shared Memory using OpenMP
	02/01/2018	
4	02/06/2018	
	02/08/2018	
5	02/13/2018	
	02/15/2018	
6	02/20/2018	Programming with Distributed Memory using MPI
	02/22/2018	
7	02/28/2018	
	03/01/2018	
8	03/06/2018	
	03/08/2018	
9	03/13/2018	Spring Break (No Class)
	03/15/2018	Spring Break (No Class)
10	03/20/2018	Programming with Distributed Memory using MPI
	03/22/2018	
11	03/27/2018	Programming with GPU using OpenACC
	03/29/2018	Easter Break (No Class)
12	04/03/2018	Programming with GPU using OpenACC
	04/05/2018	
13	04/10/2018	MATLAB with Parallel Tool Kit
	04/12/2018	
14	04/17/2018	Cloud Computing
	04/19/2018	
15	04/24/2018	Apache Spark
	04/26/2018	
16	05/01/2018	Additional resources (CUDA, Python Multi-Processing, Boost C++ Parallel Programming)
	05/03/2018	