MSDS: High-Performance Computing and Data Science

Contact Information

Instructor	Email	Phone	Office	Office Hours
Dr. Robert Kalescky	rkalescky@smu.edu	214-768-3070	Heroy Hall 303	By appointment

Dates and Times

Apr 25, 2022 - Aug 4, 2022 TuTh 6:30PM - 8:00PM

Course Overview

An introduction to high-performance computing (HPC) applied to data science, enabling accelerated data analysis and exploration of large data sets at scale. The course provides a practical introduction to HPC systems, job scheduling, and parallelization of data analysis and machine learning tasks using tools such as the Rapids framework, Dask, and Horovod. Students will learn to effectively manage large datasets with efficient file formats, define reproducible software environments using containers and version control systems, and execute parallelized workflows by simultaneously using multiple HPC compute nodes, CPUs, and GPUs.

The course will be taught using Unix shell scripting, Python, and C++ where appropriate to demonstrate various aspects of high-performance data analysis and machine learning workflows.

Course Prerequisites

Students taking this course must be enrolled in the SMU Masters of Data Science program and have completed MSDS 7331.

Learning Outcomes

At the end of the course students will be able to:

- Identify the hardware components of cluster supercomputers.
- Use Unix shell commands for managing data and computational workflows.
- Build and manage optimized software stacks.
- Execute workflows via a cluster resource scheduler.
- Understand the performance and programability trade-offs of interpreted and compiled languages.
- Use versions control systems for managing software development and collaboration.
- Use hardware and language appropriate tools for software debugging and performance analysis.
- Be familiar with common high-performance data file formats.

- Understand methods for efficiently organizing data during workflows.
- Be able to identify opportunities for parallel computation and improving computational efficiency.
- Be able to decompose common workflows for parallel computation at the socket, node, accelerator, and cluster levels.

Materials

All of the materials listed are freely available online via the links or through the SMU Libraries.

Required

(1) Boehm, M.; Kumar, A.; Yang, J., *Data Management in Machine Learning Systems*; Synthesis Lectures on Data Management, # 57; Morgan & Claypool: San Rafael, California, 2019, https://doi.org/10.2200/S00895ED1V01Y201901DTM057.

(2) Sutton, R. S.; Barto, A. G., *Reinforcement Learning, An Introduction*; The MIT Press: Cambridge, Massachusetts, 2018, http://incompleteideas.net/book/the-book.html.

(3) Victor Eijkhout with Robert van de Geijn and Edmond Chow, *Introduction to High Performance Scientific Computing*; lulu.com: 2011, https://bitbucket.org/VictorEijkhout/scientific-computing-public.

Supporting

(1) Rodriguez, A. Deep Learning Systems: Algorithms, Compilers, and Processors for Large-Scale Production. *Synthesis Lectures on Computer Architecture* **2020**, *15*, 1–265, https://doi.org/10.2200/S01046ED1V01Y202009CAC053.

(2) Mittal, S.; Vaishay, S. A Survey of Techniques for Optimizing Deep Learning on GPUs. *Journal of Systems Architecture* **2019**, *99*, 101635, https://www.sciencedirect.com/science/article/pii/S1383762119302656.

(3) Prasad, S. K.; Gupta, A.; Rosenberg, A. L.; Sussman, A.; Weems Jr., C. C., *Topics in Parallel and Distributed Computing: Introducing Concurrency in Undergraduate Courses*; Elsevier Science & Technology: San Francisco, 2015, https://smu.primo.exlibrisgroup.com/permalink/01SMU_INST/12013t3/cdi_proquest_ebookcentral_EBC4003865.

Syllabus

Date	Topics
Introduction	
Week 1	Course Introduction and Overview

Date	Topics
Week 1	High-Performance Computing (HPC) and Data Science
HPC Hardware	
Week 2	Cluster Supercomputers
Week 2	Nodes
Week 2	Accelerators
Week 2	Networks
Week 2	Storage
HPC Environments	
Week 3	Unix
Week 3	Software
Week 4	Containers
Week 5	Jobs and Scheduling
Programming Primer	
Week 6	Comparative Introduction to Python and C++
Week 7	Build Systems
Week 7	Programming Best Practices
Parallel Debugging and Performance Analysis	
Week 8	Debugging
Week 9	Profiling
Week 9	Benchmarking
Managing Data	
Week 10	Data File Formats
Week 11	Data Structures
Parallel Programming	
Week 12	Vectorization
Week 13	Shared Memory Parallelization
Week 14	Distributed Memory Parallelization
Week 15	Accelerator Programming

Technology Requirements

As the course will be held remotely and HPC compute resources used for teaching and coursework are also remote, the course requires a computer capable of the following:

- Attending classes and group meetings via video teleconferencing.
- Accessing the course materials.
- Accessing remote HPC compute resources via SSH and HTTP(S).

Evaluation

Scripting and Programming Languages

All programming assignments completed by students must be done using the appropriate choice of Unix shell scripting, Python, or C++.

Assignment Category	Grade Portion
Class Attendance and Participation	15%
Homework	20%
Labs	30%
Project	35%

Table 3: Assignment category final grade weights.

Homework

Homework assignments will be given throughout the semester. These assignments will include responses to reading, programming, and data analysis assignments. Homework assignments are to be completed individually. However, discussions amongst students and the instructor are encouraged.

Labs

Lab assignments will be given for each major section of the course. The lab assignments are to be completed individually. Labs will undergo peer review by an assigned student prior to submission.

Projects

The course will have one individual project with portions to be completed throughout the semester. Completed portions will undergo peer review by an assigned student along with a meeting with the instructor prior to submission.

Grading

All assignments will receive a grade of [100–0]. All assignments within an assignment category will be averaged. The averaged category values will be scaled by the values given in Table 4 and summed. This value will be used to determine the final grade in the course as detailed in Table 4.

Late assignments will have their maximum value reduced by 20% each day the assignment is late, i.e. after five days the student will receive a zero for the assignment.

Academic Integrity

Academic integrity during the course will be taken very seriously. Violations, including plagiarism, will be immediately reported to the SMU Honor Council.

University Policies

Disability Accommodations Students needing academic accommodations for a disability must first register with Disability Accommodations & Success Strategies (DASS).

Total Percentage	Grade
[100–93]	A
(93–90]	A-
(90–88]	B+
(88–83]	B
(83–80]	B-
(80–78]	C+
(78–73]	C
(73–70]	C-
(70–60]	D
< 60	F

Table 4: Final cumulative grade required for each letter grade.

Students can call 214- 768-1470 or visit: http://www.smu.edu/Provost/ALEC/DASS to begin the process. Once registered, students should then schedule an appointment with the professor as early in the semester as possible, present a DASS Accommodation Letter, and make appropriate arrangements. Please note that accommodations are not retroactive and require advance notice to implement.

Religious Observance Religiously observant students wishing to be absent on holidays that require missing class should notify their professors in writing at the beginning of the semester, and should discuss with them, in advance, acceptable ways of making up any work missed because of the absence. (See University Policy No. 1.9.)

Excused Absences for University Extracurricular Activities Students participating in an officially sanctioned, scheduled University extracurricular activity should be given the opportunity to make up class assignments or other graded assignments missed as a result of their participation. It is the responsibility of the student to make arrangements with the instructor prior to any missed scheduled examination or other missed assignment for making up the work.