CS499: High Performance Computing (Independent Study)

Last changed on September 19, 2023

Please email mfricke@unm.edu if you have questions about the syllabus.

Recorded lectures are posted on Monday nights.

Lectures and Labs will be held in the CARC workshop room (1601 Central Ave).

Contents

1 Description

This course is a junior level introduction to high performance computing (HPC). Topics include basics of linux and its administration, building a high performance computing cluster, optimisation of compiled code for different CPU and GPU architectures, deployment and use of industry standard HPC tools such as spack, SLURM, FreeIPA, and LMOD.

This syllabus is intended to make course expectations and policies clear so that you, the student, have the best chance possible of meeting those expectations. If anything is unclear please let me know so I can update the syllabus accordingly.

This syllabus is likely to be revised throughout the semester.

2 Instructors

Instructor: **Prof. Matthew Fricke**
Email: mfricke@unm.edu

Website: http://www.cs.unm.edu/~mfricke
Office Hours. Zoom: https://unm.zoom.us/j/263992587
Wednesdays, 10:00-11:00am
Thursdays, 2:00-3:00pm

3 Format

This is a three credit hour course consisting of two 75 minute lectures per week to be held on Wednesday and Friday at 2:30pm.

3.1 Classroom and Workstations

Classes will be held in the CARC workshop room. This workshop room is equiped with Linux workstations that you will be expected to use for in class assignments. You are welcome to being a laptop to class, but due to the variety of possible OS configurations I will not be able to help you carry out the in class assignments on your laptop.

Adjacent to the workshop room is the Roadrunner room. The Roadrunner room houses the class clusters. The Roadrunner room also houses sensitive equipment for CARC. You will not be permitted access to the Roadrunner room unless I am present. After the first 2 weeks physical access should become unnecessary.

3.2 Curriculum Format

Classes will be taught using a mixture of lectures and hands on activities. The material you will learn is divided into two areas: 1) Linux and HPC Cluster Management and 2) High performance application management and benchmarking. Material on each topic will be presented each week. At first you will conduct your benchmarking on the UNM CARC clusters, but once your own clusters are deployed you will include them in your application deployment and benchmarking tasks.

3.3 Individual Homework

During the course you will be assigned homework to complete individually. You will have one week to complete your homework assignments.

3.4 Team Projects

You will be randomly assigned to a team for each project. Team membership will change for each project. Random assignment is essential for determining final grades. Requests to work with particular students will not be entertained. Projects will be assigned on Fridays. Your team will be given two weeks to complete your project with a progress update due on the first Friday.

3.5 Presentations

During the last week of the semester and the final exam week teams present the report from one of their projects. The audience will be HPC experts from industry, academia, and the national labs.

4 Resources

4.1 Recorded Lectures

Lectures are posted under this YouTube playlist. These lectures are a supplement to the in person lectures not a substitute.

4.2 Slack Workspace

The slack workspace serves several purposes. 1) I will make announcements about the class on slack. There is no guarantee that I will also make those announcements in class. I may also make announcements in class that are not repeated on slack. You are responsible for both. 2) Slack allows class-wide discussion. Feel free to post questions about the material presented in class, the code you are writing for your assignments, and logistic questions related to the class.

4.3 Center for Advanced Research Computing (CARC)

CARC is the UNM high performance computing center. All students will be provided with a CARC account so you can compare the performance of the code you build on different hardware.

4.4 Class Cluster

Students will be assigned to teams. Each team will be responsible for assembling and configuring their own small cluster using parts provided by CARC. The cluster will be housed in the Roadrunner room at CARC but teams will be able to access their assigned cluster remotely. These clusters will be used over the course of the semester.

4.5 Online Resources

https://github.com/hpc/cluster-school/blob/master/Chapters/

5 Schedule

This is the planned schedule, any changes will be recorded here.

For UNM semester dates and deadlines are here:
`http://registrar.unm.edu/semester-deadline-dates/spring-2020.html`

5.1 Week 1 (Aug 21 - 25): Intro, Devices, and Filesystems

Lecture Slides

General

- Course Introduction
- Syllabus Review
- BASH Shell and commonly used commands (directory traversal, transferring data, reading text files, grep, tail, etc)

System Administration

- Devices
- Filesystems and Partitioning

Applications

- VecAdd

Assignments

- Homework 1 “Installing Linux at Home” Assigned - Due Feb 5th.
- Project 1: VecAdd

5.2 Week 2 (Aug 28 - Sept 1): Bootstrapping and MPI

System Administration

- The Boot Process
- BIOS and UEFI
- Bootstrap
- GRUB

Applications

- MPI Overview
- Parallel calculation of PI. Benchmarking parallel vs serial versions.

Assignments

- Homework 1 Due Feb 5th.
- Project 1 Status Report

5.3 Week 3 (Sept 4 - 8): Building a Cluster and Caching

System Administration

- Power Connections
- Network Connections
- Configure BIOS/UEFI
- Installing Linux on the Head Node

Applications

- Matrix Multiplication (caching and locality)

Assignments

- Project 1 (VecAdd) Due
- Project 2 (MatMul) Assigned

Feb 7 - Last day to DROP without "W" grade and receive a 100% tuition refund on LoboWeb

5.4 Week 4 (Sept 11 - 15): IPMI, Networking, and Scaling

System Administration

- IPMI
- Networking Overview

Applications

- Scaling up to many CPUs and Nodes (HPL), Ahmdal's Law

Assignments

- Project 2 (MatMul) Status Report

5.5 Week 5 (Sept 18 - 22): The Kernel

System Administration

- The Linux Kernel

Applications

- Scaling up to many CPUs and Nodes (HPCG), Strong and Weak Scaling

Assignments

- Project 2 (MatMul) Due
- Project 3 (Benchmarking and Optimising HPL and HPCG) Assigned

5.6 Week 6 (Sept 25 - 29): User space, HPL

System Administration

- User Environment
- Services, Systemd, Logs
- Account Management

Applications

- Scaling up to many CPUs and Nodes (HPL)

Assignments

- Project 3 (Benchmarking and Optimising HPL) Status Report

5.7 Week 7 (Oct 2 - 6): Warewulf and HPCG

System Administration

- Provisioning Cluster Nodes with Warewulf

Applications

- Scaling up to many CPUs and Nodes (HPCG)

Assignments

- Project 3 (Benchmarking and Optimising HPL) Due
- Project 4 (Benchmarking and Optimising HPCG) Assigned

5.8 Week 8 (Oct 9 - 11): SLURM

System Administration

- SLURM

5.9 Fall Break (Oct 12 - 13)

5.10 Week 9 (Oct 16 - 20): Processes, Resources, and HPCG

System Administration

- Processes and Resource Utilisation

Applications

- Scaling up to many CPUs and Nodes (HPCG)

Assignments

- Project 4 (Benchmarking and Optimising HPCG) Status Report

5.11 Week 10 (Oct 30 - Nov 3): SPACK, LMOD, and AI

System Administration

- SPACK and LMOD

Applications

- Machine Learning and HPC

Assignments

- Project 4 (Benchmarking and Optimising HPCG) Due

- Project 5 (Benchmarking and Optimising ML Perf) Assigned

5.12 Week 11 (Nov 6 - 10): Security and JupyterHub

System Administration

- System Security (encryption and firewalls)
- Network Services (SSH, HTTPS)
- JupyterHub

Applications

- Machine Learning

Assignments

- Project 5 (Benchmarking and Optimising ML Perf) Status Report

April 17 - Last day to DROP without Dean's Permission on LoboWEB

5.13 Week 12 (Nov 13 - 17): SC2023

SUPERCOMPUTING STUDENT CLUSTER COMPETITION

5.14 Week 13 (Nov 20 - 22): RAID, Sci Computing

System Administration

- RAID

Applications

- Survey of Common HPC Scientific Computing Applications

Assignments

- Project 6 (Benchmarking and Optimising MPAS) Status Report

5.15 Thanksgiving Break (Nov 23 - 24)

5.16 Week 14 (Nov 27 - Dec 1): Parallel Filesystems

System Administration

- Parallel File Systems

Applications

- Survey of Common HPC Scientific Computing Applications

Assignments

- Project 6 (Benchmarking and Optimising MPAS) Due

5.17 Week 15 (Dec 4 - 8): Project Presentations

Career Engagement

- Presentations
- Ben Schwaller (Sandia Labs) - Dec 6th.

May 8 - Last day to DROP with Dean's Permission with form.

5.18 Final Exam Week (Dec 11 - 15): Project Presentations

Career Engagement

- Presentations

Sunday, May 17th, 2:30pm - Upload of Final Grades Expected by UNM.

6 Assignments and Grading

I use a 10 point fractional grading scale:

A+ 98-100	A 93-97	A- 90-92	B+ 87-89	B 83-86	B- 80-82	C+ 77-79	I Incomplete
C 73-76	C- 70-72	D+ 67-69	D 63-66	D- 60-62	F < 60		W Withdrew

A+, A, and A- indicate excellent performance.
B+, B, and B- indicate good performance.
C+, C, and C- indicate satisfactory performance.
D+, D, and D- indicate less than satisfactory performance.
F indicates unsatisfactory performance.
W indicates withdrawal from the course.

The course grade is comprised of the following:

Homework: 30% (6 homeworks 5% each)
Projects: 70% (7 projects 10% each)

6.1 Homework 1

1. Download VirtualBox
2. If you have a PC you may need to enable virtualisation in your computer BIOS. The exact option varies by motherboard manufacturer.
3. Download Rocky 8 ISO
4. Install VirtualBox

5. Choose Expert Mode do not select the ISO in in VirualBox or it will do the install for you and you will have no control over the partition scheme.
6. Create a 20 GB virtual hard disk
7. Select Linux and Redhat 8 as the type of system (shouldn't really matter)
8. After you create the virtual machine go to settings and change the boot order so the optical disk is before the harddisk (that setting is under system: boot order)
9. Start the virtual computer. It should fail to boot and prompt you to select an ISO to load in the virtual DVD drive.
10. You should now see the Rocky linux install bootstrap loader menu
11. select scaled mode from the machine menu to make it easy to resize the virtual screen. On mac left control key releases the mouse back to the host system. On windows it is right ctrl.
12. Follow this guide when installing Rocky: Example Configuration Guide
13. Boot linux and login as root.
14. Run "lshw -C volume", submit the result to me via email.
15. Repeat the installation but this time select LVM instead of standard when partitioning.
16. Run "lshw -C volume", submit the result to me via email.
17. Run "pvdisplay" and include the output in your submission email.
18. The email submission for this homework should include 3 output files: lshw for the standard partitioning scheme, lshw for LVM, and pvdisplay for LVM. Deliverable: email me your results before midnight on Friday September the 1st, 2023.

6.2 Project 1: VecAdd

Compile and Run Vec Add on CARC CPUs and GPUs. Produce plots showing how performance scales with additional CPUs. Compare 2 CPUs to 2 GPUs on Xena.

Clone the VecAdd code from this repo: https://github.com/gmfricke/vecaddmpi

Create the following 6 plots in a Juptyer Notebook using MatPlotLib as shown in class. Use markdown in the notebook to answer the questions below.

Use this command to create a conda environment for plotting:

conda create –name plotting matplotlib seaborn ipykernel

And this notebook as a template: Notebook

On Wheeler compile and run the CPU vecadd code with 3 through 16 CPUs.

1. Plot the "All compute processes finished" time in the output on the y-axis and the number of CPUs on the x-axis.
2. Plot the "Received results from all compute processes" time on the y-axis and the number of CPUs on the x-axis

3. Plot the "Total time" time on the y-axis and the number of CPUs on the x-axis

On Hopper compile and run the CPU vecadd code with 3, 8, 16, 24, 32, 40, 48, 56, and 64 CPUS.

1. Plot the "All compute processes finished" time on the y-axis and the number of CPUs on the x-axis.
2. Plot the "Received results from all compute processes" time on the y-axis and the number of CPUs on the x-axis
3. Plot the "Total time" time on the y-axis and the number of CPUs on the x-axis

Do you see any trends in the plots or not? How does the time to compute the partial results compare to the time it takes to recieve those results as the number of CPUs increases? Is that relationship the same on Hopper and Wheeler?

On Xena report the "All compute processes finished" time for the GPU version and the CPU versions. How do the GPU and CPU vec add times compare? What are your ideas about why one is better (or not) than the other.

Deliverable: email me your jupyter notebook before midnight on Friday September the 8th, 2023.

6.3 Project 2: HPL

High-Performance Linpack (HPL) The HPL benchmark solves a (random) dense linear system in double precision arithmetic. It is often used to measure the peak performance of a computer or that of a high-performance computing (HPC) cluster. The ranking of the top 500 supercomputers in the world is determined by their performances with the HPL benchmark.

Read this paper written by some of the HPL developers: The LINPACK Benchmark: past, present and future. Don't worry about understanding everything in the paper, concentrate on the big picture.

The HPL algorithm is changeable and can be tuned via 16 parameters:

- NB: Block size
- P: Rows in process grid
- Q: Columns in process grid
- Depth: Lookahead depth
- Bcasts: Panel broadcasting method
- Pfacts: Panel factorization method
- Rfacts: Recursive factorization method
- Pmap: Process mapping
- threshold: for matrix validity test
- Ndiv: Panels in recursion
- Nbmin: Recursion stopping criteria
- Swap: Swap algorithm

- L1, U: to store triangle of panel
- Align: Memory alignment

You will find this HPL benchmarking tutorial useful HPL Tuning, since it describes the various parameters in HPL.dat. Those parameters can be optimised to make HPL as fast as possible for the hardware on which it runs.

You will load modules appropriate for the CARC cluster you are using. Use srun and SLURM instead of manually specifying the compute nodes to use.

This project is divided into two parts.

In part one you will explore the NB (block size), P and Q. Use the provided Jupyter Notebook to present your results: Notebook.

1. Clone this repository: HPL project repository into your home directory at CARC.
2. On the Hopper cluster use the parameter sweep SLURM script provided in HPL/hopper/gcc-11/param_sweep to explore block size values between 16 and 1024. Use at least 30 factor levels (30 different values of block size).
3. Plot the results of your block size experiments using the provided Jupyter Notebook. The y-axis will be throughput in Gigaflops and the x-axis values for block size.
4. Identify the value of block size that yields the highest throughput.
5. Modify the provided code using block size as an example so that you can perform a parameter sweep of P and Q just like you did for block size.
6. Using the best block size from above vary P and Q and report your findings in a 3D plot using the provided Jupyter Notebook. Factor levels for P and Q; 1, 2, 3, 4, 5, 6, 7, 8. So a total of 64 pairs.
7. Use this online calculator and block sizes to generate an optimised HPL.dat file. HPL.dat Calculator. Report the throughput of this optimised HPL.dat file.
8. Repeat steps 1 through 7 for Wheeler.

Suggestions for which block sizes, P, and Q should do well. We will allocate 16 GB of RAM for our experiments. A reasonable problem size has already been selected for 16 GB so do not change the problem size.

What block size NB should I use? HPL uses the block size NB for the data distribution as well as for the computational granularity. From a data distribution point of view, the smallest NB, the better the load balance. You definitely want to stay away from very large values of NB. From a computation point of view, a too small value of NB may limit the computational performance by a large factor because almost no data reuse will occur in the highest level of the memory hierarchy. The number of messages will also increase. Efficient matrix-multiply routines are often internally blocked. Small multiples of this blocking factor are likely to be good block sizes for HPL. The bottom line is that "good" block sizes are almost always in the [32 .. 256] interval. The best values depend on the computation / communication performance ratio of your system.

To a much less extent, the problem size matters as well. Say for example, you emperically found that 44 was a good block size with respect to performance. 88 or 132 are likely to give slightly better results for large problem sizes because of a slighlty higher flop rate.

What process grid (P x Q) should I use? This depends on the physical interconnection network you have. Assuming a mesh or a switch HPL "likes" a 1:k ratio with k in [1..3]. In other words, P and Q should be approximately equal, with Q slightly larger than P. Examples: 2 x 2, 2 x 4, 2 x 5, 3 x 4, 4 x 4, 4 x 6, 5 x 6, 4 x 8 ... If you are running on a simple Ethernet network, there is only one wire through which all the messages are exchanged. On such a network, the performance and scalability of HPL is strongly limited and very flat process grids are likely to be the best choices: 1 x 4, 1 x 8, 2 x 4. From https://icl.utk.edu/hpl/faq/index.html

6.4 Project 3: HPCG

HPC Conjugate Gradient (HPCG) The HPCG benchmark uses a preconditioned conjugate gradient (PCG) algorithm to measure the performance of HPC platforms with respect to frequently observed but challenging patterns of computing, communication, and memory access. While HPL provides an optimistic performance target for applications, HPCG can be considered as a lower bound on performance. Many of the top 500 supercomputers also provide their HPCG performance as a reference.

Read more: https://www.hpcg-benchmark.org/

6.5 Project 4: ML Perf

MLPerf Inference Machine Learning (ML) is increasingly being used in many scientific domains for making groundbreaking innovations. MLPerf Inference is a benchmark suite for measuring how fast systems can run models in a variety of deployment scenarios. The key motivations behind this benchmark is to measure ML-system performance in an architecture-neutral, representative, and reproducible manner.

Read more: https://mlcommons.org/en/inference-datacenter-30/

6.6 Project 5: MPAS

MPAS (Atmosphere Core) The Model for Prediction Across Scales—Atmosphere (MPAS-A) is an atmospheric simulation model for use in climate, regional climate, and weather research. MPAS-A supports global and limited-area domains with horizontal resolution from O(100) km down to O(1) km or less, and it employs unstructured meshes known as centroidal Voronoi tessellations (CVTs). The model consists of a dynamical core, which handles the resolved-scale equations of motion, as well as parameterizations of additional physical processes. MPAS-A is developed by the National Center for Atmospheric Research (NCAR), and it shares software infrastructure that was co-developed with the Los Alamos National Laboratory.

Key software characteristics of MPAS-A: Runs on hardware as limited as a Raspberry Pi or as powerful as the largest systems on the Top500 list Primarily Fortran 2008 code, with some C Parallelization with MPI and OpenMP by horizontal domain decomposition Support (in a separate code branch) for executing parts of the model on GPUs via OpenACC

Homepage in NCAR's MMM Lab: https://www.mmm.ucar.edu/models/mpas

Source code repository: https://github.com/MPAS-Dev/MPAS-Model

6.7 Project 6: 3DMHD

3DMHD (Three-Dimensional Magneto Hydro Dynamic) This is a numerical simulation, written in Fortran with MPI, to study the descent of cold and dense plumes in a stratified layer. Such simulations are important to understanding dynamics of plume development in regards to thermal and magnetic forces inside of stars.

6.8 Late Projects

Late homework and projects will not be accepted unless accompanied by an absence note provided by the Dean of Students Office for the period beginning on or before the *day the assignment was due* until the *day before* the assignment is actually submitted. I do this because allowing late submissions without good cause results in more stress for students than having strict deadlines.

7 Attendance

- Please see the UNM policy on class attendance: Class Attendance Policy
- Class attendance will not be recorded.

8 Cheating

8.1 UNM Policy

- See the UNM policy on academic dishonesty: Academic Dishonesty

8.2 Why it's a bad idea

Cheating is harmful to you, the University, and your fellow students:

- The University warrants that receipt of grades and a degree accurately reflects the knowledge and skills of the recipient. Cheating undermines the value of the degrees and grades awarded by the University.

- Cheating makes assignments and exams look easier than they really are. This encourages instructors to make assignments and exams harder than they would otherwise.
- The desire to cheat indicates that there is an issue with developing the desired skills that needs to be resolved. Please approach lab instructors or the professor to discuss problems with the material and how to honestly do better in the class.
- Cheating indicates that the material is not being sufficiently mastered, this will likely result in difficulties when you take classes in the future that require this course.
- Trading your integrity for a better grade is something that will stay with you for the rest of your life, whether or not you are discovered.
- If you are caught cheating you will receive a failing grade for the course. However, the University may take further action including dismissal from the University.

8.3 What I do allow

You may use any resources to complete the projects including learning from other students and taking information from the internet. I encourage learning from your class mates.

8.4 What I don't allow

Quizzes will be used to test whether you have developed an individual understanding HPC fundamentals covered in the course. No collaboration or exam aids such as notes or devices may be used during the exam. Failure to follow this policy will result in an F on the exam or quiz, possibly the course, and may result in disciplinary action by the University. You may not copy quiz answers from other students.

9 UNM Resources for Students

9.1 Student Health and Counselling (SHAC)

"SHAC provides quality health and counseling services to all UNM students to foster student success."

https://shac.unm.edu/

505.277.3136 (24-hr number)

9.2 Center for Academic Program Support

https://caps.unm.edu/index.php

"CAPS is the University of New Mexico's learning center. We are a free service that provides academic support for undergraduate students at UNM through peer-tutoring. Our tutors are UNM students who are trained to create comfortable and welcoming spaces for students to learn and thrive in. We are passionate about helping students achieve individual academic success at UNM through peer tutoring and interactions."

9.3 Accessibility Resource Center

https://arc.unm.edu/

In accordance with University Policy 2310 and the Americans with Disabilities Act (ADA), academic accommodations may be made for any student who notifies the instructor of the need for an accommodation. It is imperative that you take the initiative to bring such needs to the instructor's attention, as I am not legally permitted to inquire. Students who may require assistance in emergency evacuations should contact the instructor as to the most appropriate procedures to follow. Contact Accessibility Resource Center at 277-3506 for additional information.

If you need an accommodation based on how course requirement interact with the impact of a disability, you should contact me to arrange an appointment as soon as possible. At the appointment we can discuss the course format and requirements, anticipate the need for adjustments and explore potential accommodations. I rely on the Disability Services Office for assistance in developing strategies and verifying accommodation needs. If you have not previously contacted them I encourage you to do so.

9.4 Equal Educational Opportunities

Our classroom and our university should always be spaces of mutual respect, kindness, and support, without fear of discrimination, harassment, or violence. Should you ever need assistance or have concerns about incidents that violate this principle, please let me know and/or access the resources available to you on campus, especially the LoboRESPECT Advocacy Center and the support services listed on its website (http://loborespect.unm.edu/).

Several Federal civil rights laws prohibit discrimination in programs or activities that receive Federal funds from the Department of Education. These laws prohibit discrimination on the basis of race, colour, and national origin (Title VI of the Civil Rights Act of 1964), sex (Title IX of the Education Amendments of 1972), age (Age Discrimination Act of 1975) and disability (The Americans with Disabilities Act of 2008). The Office for Civil Rights enforces these rights: https://www2.ed.gov/about/offices/list/ocr/know.html.

Please note that because I and the student lab instructors are considered "responsible employees" by the Department of Education, we are required to report any disclosure of gender discrimination (including sexual harassment, sexual misconduct, and sexual violence) to the university's Title IX coordinator. For more information on the campus policy regarding sexual misconduct, please see: UNM Policy.

UNM's Title IX Coordinator, Angela Catena, can be reached at acatena@unm.edu.