Parallel Computing

Jinzhen Wang
jwang96@charlotte.edu

Department of Computer Science
UNC Charlotte

Outline

About Me

- Assistant Professor of Computer Science
- PhD graduate in 2023, from New Jersey Institute of Technology
- I do research on High-Performance Computing, with the focus on data compression, data management.
- Marvel fan, anyone?
- Coffffffffffee drinker, stop by for some.

Format

- ITCS 6145/8145 is Face-to-Face @ Burson 115
  - All students are required to attend the class In Person.
  - If you cannot attend the class due to any reasons, send me an email BEFORE the class.
- Instructor: Jinzhen Wang (jwang96@charlotte.edu)
  - Office Hours: Mon Wed 3:00pm - 4:00pm or by appointment
  - Woodward 210B
- TA: Khondoker Mumenin (kmumenin@charlotte.edu)
- Slack group chat: `https://join.slack.com/t/itcs61458145p-fcl7468/shared_invite/zt-2owrwcl8j-lypTLKX4beLgBHCPHbAoOQ`
  - DM me and TA when you have any questions.
  - Discuss with your peers.
- Important notifications will still be sent on Canvas and by emails.

Material

Textbook

- I am not quite following a given textbook.
- But I'll point you to external material (see last slides of each deck).

Class material

- Slides, Quizzes, Assignments, and Answers will be posted online
- Answers for the assignments will be posted online right after its due day

Typo hunt

- If you see a typo or something that does not make sense in a slide, let me know.
- That way I can fix it for the next class and also help lifting doubts.

Workload and Grading

- Workloads
  - One assignment per topic
    - Most assignments are programming-oriented
    - You need to run, test, debug, and benchmark them on actual HPC Cluster.
  - Three exams
    - 2 x Midterm, 1 x Final
- Grade Distribution
  - Assignments: 30%
  - In-class Quiz: 10%
  - Two Midterm: 30%
  - One Final Exam: 30%

Submission Guidelines

- **All** submissions should be on Canvas. Assignments submitted through email will not be graded.
- Assignments are usually due a week after they have been released. If the assignment due date is a date where we have a class, the assignment is due BEFORE class starts.
- Assignment can be submitted up to one week late; but a late submission penalty applies.
  - If an assignment is submitted L dayes late, then the maximum achievable grade for that assignment is $100 - 3 * L$.
- Throughout the semester, you have 2 credits throughout the semester for late submissions without penalty.

Important Dates

- Check the academic calendar
  - https://registrar.charlotte.edu/calendars-schedules/academic-year-fall-2024-summer-2025
- Some important dates:
  - September 2, 2024 Labor Day, No Class
  - October 14, 2024 Student Recess, No Class
  - November 11, 2024 Veteran's Day, No Class
  - November 18, November 20, Business Travel, No Class
  - December 4, 2024 Last day of class
  - December 5, 2024 Reading Day
  - December 11, 2024 Final Examinations
- https://ninercentral.charlotte.edu/wp-content/uploads/sites/803/2024/08/Fall-2024-FE-Table.pdf

Outline

What this class is about

- Module 1: Introduction to Parallel Computing
- Module 2: Parallel Computing Theory
- Module 3: Introduction to Shared Memory Programming
- Module 4: Pthread Programming
- Module 5: OpenMP Programming
- Module 6: CUDA Programming
- Module 7: Introduction to Distributed Memory Programming (well, if we still have time)
- Module 8: Distributed Memory Programming using MPI (again, if we still have time)

Expectation of Incoming Knowledge

Y'all know Computer Architecture, right?

- Programming in C or in C++
- Surviving in a UNIX environment
- Using git
- Basic Algorithm
  - Complexity (Big O, Big Theta)
  - Sorting (Mostly Merge Sort)
  - Some graphs

Academic Integrity

- Examination work and assignment are expected to be the sole effort of the student submitting the work. Students are expected to follow the UNC Charlotte Code of Student Academic Integrity for all class activities, assignments and tests. This includes following all of the instructions given by the course instructor, TAs, and other test proctors. Here is a link to the code: http://legal.uncc.edu/policies/up-407
- Prohibited behaviors include plagiarism, cheating, falsification, and complicity. All cases of potential academic misconduct will be reported to the Dean of Students Office.
- If any of the prohibited behaviors is caught, or reported and proved, the persons who cheat and those who help cheating fail the course.

Use of GenAI

The use of GenAI tools like: GitHub Copilot, ChatGPT, Gemini, Claude, Jasper AI, Character.ai, DALL-E, OpenAI API, Synthesia, Otter.a is permitted WITH DISCLOSURE.

But, you are responsible for any errors or information that is misrepresented or inaccurate (i.e. hallucinations) that generative AI tools produce when submitting work that includes AI-generated material.

Outline

Cluster Environment

Our educational cluster: `centaurus.urc.uncc.edu` hosted by URC.
User note:
`https://oneit.charlotte.edu/urc/educational-clusters/centaurus-user-notes/`

System Specifications

- 12 nodes + headnode
- Two Xeon E5-2667 v3 per node
- GPU Compute Nodes:
  - 2 nodes each with 16 cores and 4 Titan RTX GPUs/node
  - 1 node with 16 cores and 8 Titan V GPUs
  - 1 node with 8 cores and 8 GTX-1080ti GPUs

Do not run code on the head node!
There are about 40 students and 12 computing nodes. Do NOT wait for the last minute!

Cluster Environment

Running job on HPC cluster

Access by ssh: `ssh centaurus.urc.uncc.edu`
Need to be connected to the VPN
Submit: `sbatch script.sh`
You get a jobID.
Eventually, you will get an output and error file.
Check status: `squeue`

Another whole lecture on this later

Using Linux System on your machine

Why?

- To remove the load from the cluster
- To allow you to work from your laptop

Recommended Workflow

- Develop & Debug on a Linux system
- Upload to git
- Log on cluster
- Download from git
- Recompile & Submit job
- Plot results, push results back to git
- Look at it on your laptop.

Git

Versioning System

- We will use git; recommend GitHub.

In short

- Initial checkout with: `git clone` *`https://something`*
- Declare new file: `git add` *`somefile`*
- Commit (locally): `git commit -am "message"`
- Upload to server: `git push`
- Download from server: `git pull`

Structure

- Each student makes one repository.
- Make one directory per programming assignment.

Lost yet?

The activities in the first module are there to get you up to speed on:

- Using the Linux VM.
- Connecting to Centaurus.
- Setting up git to work with Centaurus.
- Running a hello world.

No parallel computing involved.
There will be plenty of office hours during the semester to make sure problems can be taken care of.

Outline

External

Git stuff:

- Basic: `https://git-scm.com/docs/gittutorial`
- Complete: `https://www.atlassian.com/git/tutorials/`

C crash course:

- `http://www.mattababy.org/~belmonte/Teaching/CCC/CrashCourseC.html`
- `http://www.cprogramming.com/tutorial/c-tutorial.html`
- `https://www.physics.drexel.edu/~valliere/General/C_basics/c_tutorial.html`

C++ crash course:

- `http://www.cplusplus.com/doc/tutorial/`

C++ advanced resources:

- All the "effective" books by Scott Meyers: Effective C++, More Effective C++, Effective STL, Effective Modern C++.

Unix:

- `http://www.ee.surrey.ac.uk/Teaching/Unix/`
- Makefiles crashcourse: `http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/`

Data Structure and Algorithms:

- Data Structures & Algorithm Analysis by Clifford A. Shaffer (Available online at : `https://people.cs.vt.edu/shaffer/Book/`)

Cluster:

- URC: `https://urc.uncc.edu/`
- Centaurus details: `https://oneit.charlotte.edu/urc/educational-clusters`