CS485 Selected Topics: Intro GPU Cluster Programming - Course Syllabus

MPI+CUDA: Programming a Cluster of CUDA-capable machines

Spring 2024

- Class Web page: http://web.njit.edu/~sohna/cs485 and http://canvas.njit.edu
- **About the course**: A project course. As such, lectures will be given for the first 10 weeks or less depending on the progress and pace of the course. You will learn how to program a cluster of Cuda-capable Linux-based computers to solve a single problem at a time.
- Instructor: Andrew Sohn, GITC 4209, (973)596-2315, email: sohna _at_ njit _dot_ edu
- Office Hours: Tue 10:30-11:30 am, Thur 2:30-3:30 pm, by appointment if necessary. If you want to see me outside the office hours, send me an email.
- **Teaching assistant:** No one qualifies for this course.
- **Class time and location**: See the registrar's page https://uisnetpr01.njit.edu/courseschedule
- **Prerequisites**: CS288 Intensive Programming in Linux and CS350 Intro Computer Systems
- **Read the following warnings carefully to make an informed decision on whether this course is for you:**
    - This course is difficult and time consuming because you are programming not just only a cluster of machines but a cluster of Cuda-capable machines. This topic is the current state of the art for harnessing generative AI. As such, you should be prepared to spend at least two hours a day for this course.
    - You must be proficient in Linux, C, Bash and some C++. Otherwise, this course is not for you.
    - If you received other than B+ or A in the two prerequisite courses CS288 and CS350, you will find this class very difficult and may not be able to keep up with the pace. I strongly discourage you to take this course if your preparation is weak. In this course, I won't repeat the topics I used to teach in CS288 and currently teach in CS350.
- **The goal of the course**: Learn how to program a cluster of Cuda-capable distributed-memory Linux computers. Specifically, there are two architectural and one programming models you will learn:
    - MPI Message Passing Interface for programming a cluster of distributed-memory Linux machines. MPI is the standard for high performance computing/parallel computing. The distributed-memory architectural model is called Multiple Instruction Multiple Data (MIMD).
    - CUDA Compute Unified Device Architecture for programming Nvidia GPUs within a single Linux box. The architectural model is called Single Instruction Multiple Data (SIMD).
    - SPMD Single Program Multiple Data for programming a cluster of Cuda-capable distributed-memory machines.
- **Outcome**: Towards the end of the semester, each team of two students presents a working MPI+Cuda program that runs on a cluster of at least two Cuda-capable Linux machines. The metrics for performance is improvement over using one to many machines with and without Cuda-capable GPUs. Specifically, each team will demonstrate by measuring and comparing the execution times of
    - Version 1: a plain serial C version on a single host machine with no MPI, no Cuda.
    - Version 2: an MPI-only version on a cluster of at least two machines. No Cuda is involved here.
    - Version 3: a Cuda version on a single machine.
    - Version 4: an MPI and Cuda version on a cluster of at least two machines.
    - Version 5: an *optional* Cuda-aware MPI version on a cluster of at least two machines if you are ambitious. Note that this is the current state of the art in high performance computing/parallel computing that enable generative AI and its variants.
- **Textbooks required**:
    - MPI: A Message Passing Interface Standard v3.1, mpi-forum.org, 2015 - free

- Programming Massively Parallel Processors - A Hands-on Approach, Wen-mei W. Hwu, David B. Kirk, and Izzat El Hajj, 4th Ed., Morgan Kauffman (Elsevier), 2023.

- **Course materials**:
  - MPI tutorial: Lawrence Livermore National Laboratory: https://hpc-tutorials.llnl.gov/mpi/
  - MPI lecture notes: http://wgropp.cs.illinois.edu/courses/cs598-s15
  - Cuda toolkit: TBA
  - Cuda lecture notes: https://www.elsevier.com/books-and-journals/book-companion/9780323912310
    Recordings: https://www.youtube.com/@pmpp-book/playlists
- **Grading**:
  - Attendance (10%)
  - Programming Project in multiple versions (30%)
  - In-class midterm (30%): **4-5:15 pm, Thur, 2/22/2023.**
  - In-class final exam (30%): Date and Time TBD, See the registrar's page.
- **Requirements:** As of today Tue, 1/16/2024, a proposal has been submitted for setting up a laboratory housing a system of eight clusters each of which has four Cuda-capable Linux boxes connected through a 10G switch. All eight clusters will be connected through an 80G to 100G switch. However, it's unlikely the lab will be ready for Spring 2024. That leaves you no choice but to get a Cuda-capable GPU on your own. So here is what you have to do if you want to stay in the course.
  - Secure access to a Cuda-capable Linux box for programming in Cuda
  - A switch/router to connect two Cuda-capable Linux boxes for programming MPI. You have a router/switch at home but getting a $10-$20 4-port switch/router would be handy. A team of two can instantly build a cluster of 2-4 Linux boxes on the go, except the cluster is not Cuda-capable unless you carry a Cuda-capable laptop. You'll learn more on this when we meet. Few people know how to program a cluster of distributed-memory machines for solving single compute intensive problems, let alone a cluster of Cuda-capable machines.
  - You will be updated on the progress of establishing a lab with eight clusters of 32 Cuda-capable machines as they become available.
- **Setting up a cluster on your own:**
  - On day 1, install Fedora 37, not 38, nor 39. Make sure gcc 12, not 13. See fedoraproject.org.
  - On day 1, Install Open MPI on Fedora 37. I will show you in class how to set up a cluster of Linux boxes with MPI. Again, you have to be proficient in Linux, Bash, C, etc. If you are struggling to fiugre out what commands to use, this class is not for you. I won't explain to you the commands you were supposed to learn in CS288.
  - As soon as possible, install Cuda toolkit 12, dated July 25, 2023
- **Exam-related:**
  - There will be no make-up exam(s). You must plan your semester accordingly, especially if you work.
  - No show for midterm or final will be an automatic failure in the course.
- **Academic Integrity:** I am required to post this on the course syllabus.
  "Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf. Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu"

- **Project Timeline**
  - Weeks 1-2: Setup a cluster of 2 to 4 machines for MPI programming. Find team mates, max 4 members per group. Test run an MPI program to see if the setup works. If you have access to a cluster of 2 Cuda-capable machines, you may work alone.
  - Week 3: Submit a one-page proposal describing what project your team will work on, its scope in terms of versions, timeline, individual responsibilities, and evaluation plan (see Outcome above). Check the textbooks and Cuda toolkit 12 for potential topics. Topics must be approved by the instructor. A proposal template will be sent out. If you don't pick, I will pick one for you.
  - Week 4: Version 1 due: Implement skeleton MPI code on a cluster - No GPU Cuda yet
  - Weeks 5-6: Version 2 due: MPI draft but working version - No GPU Cuda yet
  - Weeks 7-8: Version 3 due: Include skeleton Cuda code to expand the working MPI version
  - Weeks 9-10: Version 4 due: Debug and complete MPI+Cuda version. All four versions must work by now.
  - Weeks 11-12: No lectures. Individual team discussion on your project. Pre-arrangement is required for individual/team discussion.
  - Weeks 13-14: No lectures. In-class in-person project presentation. Everyone is required to attend.
- **Lecture Schedule by Week (will most likely change based on the class pace)**
  1. Preparatory steps
     - Parallel computing/High Performance Computing - solving a **single** problem using a cluster of distributed-memory machines.
     - Architectural models - Multiple Instruction Multiple Data (MIMD) and Single Instruction Multiple Data (SIMD)
     - Programming models - Single Program Multiple Data (SPMD)
     - Setting up MPI on Fedora 37 using gcc 12
     - Installing Cuda toolkit 12
     2. MPI point to point communication - blocking send(), recv(), and probe(); if time permits, nonblocking isend(), irecv() and iprobe()
  3. MPI collective communication - scatter(), gather(), barrier(), broadcast(), scan(); if time permits, nonblocking collective functions
  4. MPI one-sided communication - put(), get(), accumulate(), compare_and_swap(), fetch_and_op()
  5. Cuda: Intro to SIMD way of thinking - Ch.3 Multidimensional grids and data: host, device, grids, blocks, threads, matrix multiplication
  6. Cuda: Ch.4 GPU architecture - Compute architecture and scheduling: streaming multiprocessors, block scheduling, warps, control divergence, latency tolerance
     Cuda: Ch.5 Memory architecture and data locality - host, per-grid global, per-thread local, per-block shared, rad-only per-grid constant, per-thread registers
     Midterm, 4-5:15 pm, Thur, 2/22/2023.
  7. Cuda: Basic patterns: Ch.7 Convolution, Ch.9 Histogram, Ch.10 Reduction
  8. Cuda: Basic patterns: Ch.11 Prefix sum - scan, Ch. 12 Merge
  9. Cuda: Advanced patterns: Ch. 13 Radix sorting, Ch. 14 Graph traversal - breadth first search
  10. Cuda: Advanced patterns: Ch. 16 Deep learning (convolutional neural networks)
  11. Team discussion - no lectures
  12. Team discussion - no lectures
  13. In-class presentation - no lectures
  14. In-class presentation - no lectures
  15. **Final exam (week15):** See the registrar's page: http://www.njit.edu/registrar