

HPC & Parallel Programming

Introduction

Kun Suo

Computer Science, Kennesaw State University

<https://kevinsuo.github.io/>

Outline

- Why study HPC & parallel programming?
- What to learn?
- Course structure
- Course policy
- An example of HPC & parallel programming



Self Introduction

- Kun Suo, Ph.D.
 - Homepage, <https://kevinsuo.github.io/>
 - Research interests:
 - Cloud computing and virtualization;
 - Parallel and Distributed Computation, containers and kubernetes;
 - Software defined network (SDN) and network function virtualization (NFV)
 - Big data systems and machine learning systems
 - Projects you may be interested in:
 - Several projects in Cloud & Data & Edge
 - <https://kevinsuo.github.io/code-lab.html>



Now it's your turn

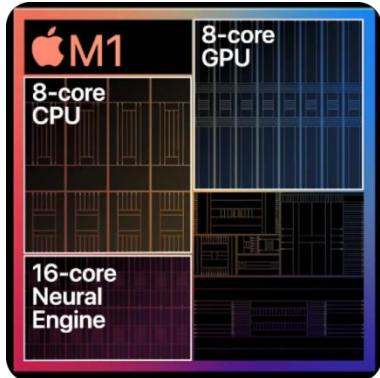
- Name, program/year, where from
- Your interests in Computer Science
- Have you ever used or heard of parallel and distributed system? Can you name some of them? What do you expect from this course?

<https://www2.eecs.berkeley.edu/Research/Areas/CS/>

If you are in the online course, introduce yourself in D2L,
Discussions → Self-Introduction



Example of HPC & Parallel Programming



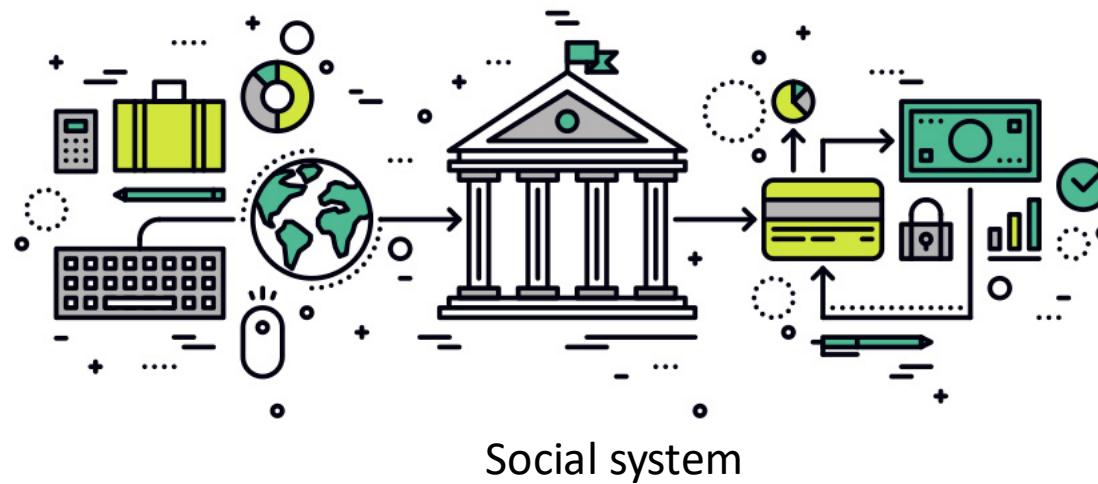
personal computer



internet



cloud



An Example of Parallel Computing

- CPU v.s. GPU

Single core

[NVIDIA: Adam and Jamie explain parallel processing on GPU's \(youtube.com\)](https://www.youtube.com/watch?v=KuXWzXWzXW)



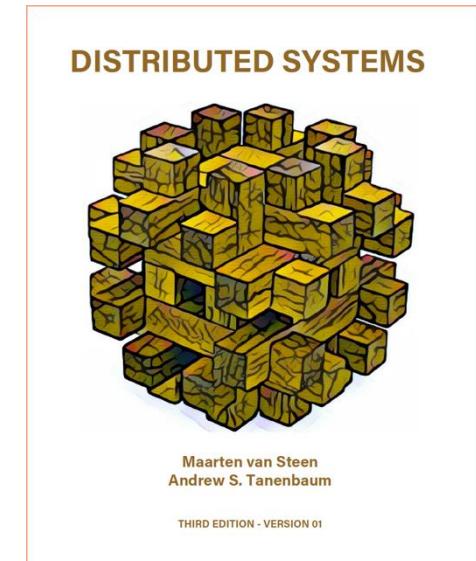
Course Information

- Instructor: Dr. Kun Suo
- Office: J-3230
- Email: ksuo@kennesaw.edu
 - Only reply to e-mails that are sent from KSU student email accounts and title the course number [CS4522]
- Office Hours:
 - Email or Microsoft Teams
 - By appointment
- Course Materials
 - Homework assignments, lecture slides, and other materials will be posted in the webpage (<https://kevinsuo.github.io/teaching.html>) and D2L.



Reference Book

- “Distributed Systems 3rd edition (2017)” by M. van Steen and A.S. Tanenbaum:
 - ISBN-13: 978-1543057386
 - You can get a digital copy of this book for free: <https://www.distributed-systems.net/index.php/books/ds3/>



Prerequisites

- Computer basics that are supposed to be covered in *(CS 3502) Operating Systems, (CS 3503) Computer Organization and Architecture* course, *(CS 4504) Parallel System* course.
- C programming (code reading, kernel development and debugging). ([Famous projects in C](#))
- Linux command line environment (compiling, Makefile, debugging, simple shell programming).



For C and Linux beginners

- C tutorial
 - <https://www.tutorialspoint.com/cprogramming/>
 - <https://www.learn-c.org>
 - <https://www.cprogramming.com/tutorial/c-tutorial.html>
- Linux tutorial
 - <https://ryanstutorials.net/linuxtutorial/>
 - <http://www.ee.surrey.ac.uk/Teaching/Unix/>
 - <https://www.tutorialspoint.com/unix/>



Project Environment

- Recommend project environment (local)

- VirtualBox + Ubuntu + Linux

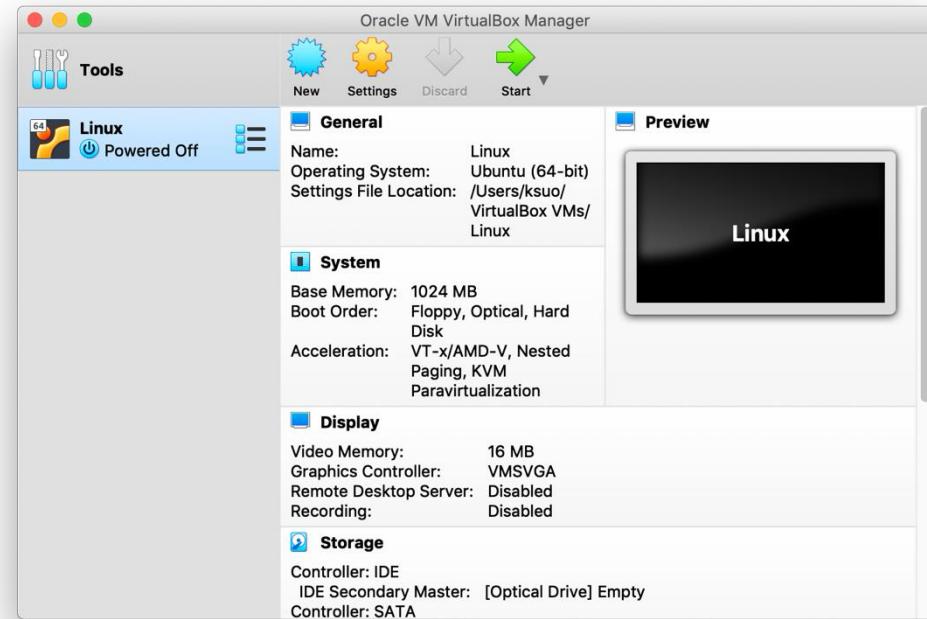
Virtual machine

VM OS

VM OS Kernel

<https://www.virtualbox.org/>

<https://ubuntu.com/download/desktop>



Project Environment

- Recommend project environment (local)
 - VirtualBox + Ubuntu + Linux
- New to VirtualBox?
 - <https://oracle-base.com/articles/vm/virtualbox-creating-a-new-vm>
 - https://www.youtube.com/watch?v=sB_5fqysi4
 - <https://youtu.be/GDoCrfPma2k> (MacOS)
- You can access to VMs in KSU data centers (cloud) through <https://cseview.kennesaw.edu/>,
 - username: administration; password: linuxadmin



Why study HPC & Parallel Programming?

- Most computer systems today are a certain form of HPC/parallel/distributed systems
 - Internet, datacenters, super computers, mobile devices
 - Most of the applications are parallel or even distributed apps (example: debug decompress file app, [link](#), starts at 2:33)
- To learn useful techniques to build large systems
 - A system with 10,000 nodes is different from one with 100 nodes
- How to deal with imperfections
 - Machines can fail; network is slow; topology is not flat



What to learn

- HPC & Parallel Programming:
 - Parallel hardware
 - Matrix multiplication optimization
 - Pthread programming
 - MPI programming
 - OpenMP programming
 - GPU programming



Expected Outcomes

- Familiar with popular parallel programming libraries (Pthread, OpenMP, MPI, GPU)
- Familiar with fundamentals of program optimization
- The ability to
 - Evaluate the performance of parallel and HPC systems
 - Write simple parallel and HPC programs
 - Understand the tradeoffs in program design



Course Structure

- Lectures
 - Time/Location
 - D2L/Course website
- Projects
 - 5 programming assignments
 - 1 paper/project presentation
- Exams (open books)
 - Midterm: online D2L, TBA.
 - Final: online D2L, TBA



Course Policy

- Grading scale

Percentage	Grade
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
Below 60	F



Grading Policy (cont.)

- Grading percentage
 - Projects (x5): 50%
 - Presentation: 10%, including project or paper presentation
 - Midterm: 20%
 - Final exam: 20%

Late submission policy: late submission will **not be accepted** and **no credits**



Academic Integrity

- Academic dishonesty

- Cheating
 - Plagiarism
 - Collusion
 - The submission for credit of any work or materials that are attributable in whole or in part to another person
 - Taking an examination for another person
 - Any act designed to give unfair advantage to a student or the attempt to commit
- [https://scai.kennesaw.edu/KSU Codes of Conduct 2019-2020.pdf](https://scai.kennesaw.edu/KSU_Codes_of_Conduct_2019-2020.pdf)
- Receiving, attempting to receive, knowingly giving or attempting to give unauthorized assistance...
- Do not upload course documents to 3rd party website without author's permission



Where to go for help ?

- Ask questions in class
- Ask questions outside class
 - Classmates and friends
- Attend office hours
 - Send Dr. Kun Suo emails or leave message on teams
- Search on the web
 - Stand on the shoulder of giants

