

CSE113: Introduction to Parallel and Concurrent Programming

March 30, 2021



<https://techwireasia.com/2020/06/japans-fugaku-is-the-worlds-fastest-supercomputer/>

<https://www.lenovo.com/>

<https://www.apple.com>

Hello!



- Professor Tyler Sorensen (he/him)
 - Call me Tyler!
- **Faculty** at UC Santa Cruz Since Summer 2020
 - This is the second class I'm teaching
- Previously
 - Post doc at Princeton
 - PhD Student at Imperial College London
 - BS/MS at University of Utah

<https://users.soe.ucsc.edu/~tsorensen/>

Research Interests

MS: Utah



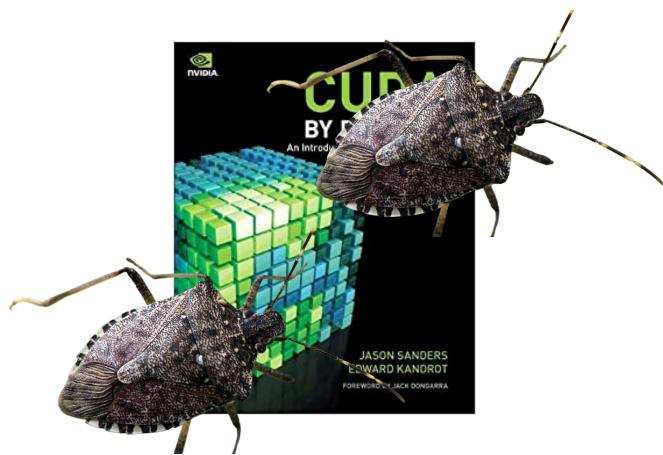
PhD: London



Post Doc: Princeton



Bugs in GPU programs



locks



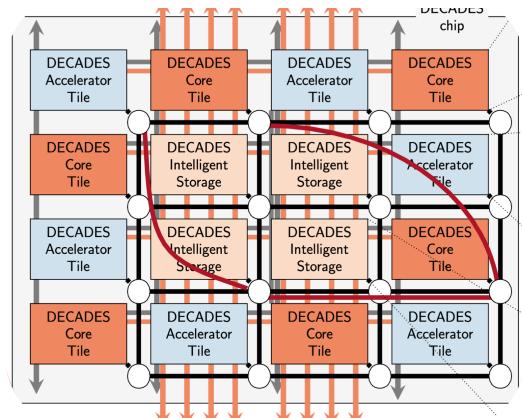
on



barriers



new parallel architectures



Research Interests

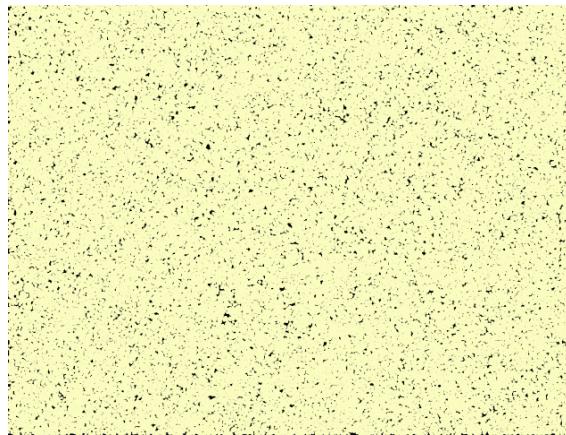
Faculty at UCSC



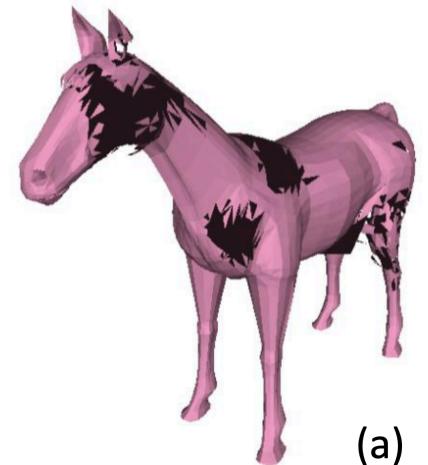
individual Contributor to



parallel n-body simulations



GPU memory model testing



(a)

Research Interests

- Parallel Programming!

Concurrency and Parallelism is everywhere!



Fujitsu SC at Riken (Japan)

7.6M cores



Consumer Laptop

2-16 cores



Mobile Phone

2-8 cores



Watches?

1 core

BUT

**still need to worry about
concurrency!**

<https://techwireasia.com/2020/06/japans-fugaku-is-the-worlds-fastest-supercomputer/>

<https://www.lenovo.com/>

<https://www.apple.com>

Concurrency and Parallelism is everywhere!

In many cases you won't know what hardware you are programming for



web apps



Android apps

You still need to worry about concurrency!

<https://techwireasia.com/2020/06/japans-fugaku-is-the-worlds-fastest-supercomputer/>

<https://www.lenovo.com/>

<https://www.apple.com>

Learning Objectives

- Foundations of concurrent/parallel computing
 - **Concepts**, not languages/frameworks!
 - Allows you to pick up future new languages and frameworks quickly
- Shared memory concurrency
 - Many concepts apply to other domains, but likely have different performance characteristics (e.g. distributed systems)
 - **Thread cooperation**, not embarrassingly parallel applications
- Think in concurrency
 - understand common synchronization idioms and their performance characterizations
 - efficiently (and safely) utilize modern systems

Wall of text warning for today's lecture!

- Class overviews are a bit dry, but important, esp. for remote instruction and classes this size!
- Future lectures will be more visually engaging I promise!

Today's Schedule

- Class Structure
- Class Contents
- Assignments, Tests, Grades

TA Introductions

- **Reese Levine**
 - PhD student working on GPU programming models with me
- **Gan Shen**
 - PhD student working on Programming Languages with Professor Kuper

They are awesome!

 Mesa >  mesa > Issues > #4475

Open

Opened 1 week ago by  Reese Levine

Relaxed atomic loads in while loops being optimized away

Describe the issue

Its been a tough year...

- Global Pandemic
- Political unrest

Special COVID Note: This year has been difficult for many different reasons. I imagine that it will stay difficult for some time. College is hard even in the best of times. Please take care of yourself; support each other; find time for the things you enjoy. Please email me if you don't feel like you are performing at the best of your ability and we can discuss various accomodations.

Class Websites

- **Public:** <https://sorensenucsc.github.io/CSE113-2021/>
 - Slides, schedule, resources, homeworks
- **Private:** Canvas
 - Grades, exams, discussions, messages, videos

Background

- **CSE 12: systems**
 - C/++ programming
 - Compilation
 - Basic unix command line
- **CSE 101: data-structures and algorithms**
 - Data structure specifications (Queues and Stacks)
 - Reasoning about algorithms (Space and time complexity)
- **CSE 120: architecture (recommended)**
 - Caches
 - Assembly Language basics
 - We will review the basics next lecture

If you do not have architecture, please consult the architecture reference on webpage!

Class Format

- **1:30 - 3:05 Tues & Thurs. 95 minutes**
 - I will try to be 15 minutes early and stay 15 minutes afterwards
- **Zoom details**
 - Global chat is enabled.
 - Reminder: Zoom chats are not private
 - TAs will monitor the chat for questions
 - Raising hands also works
 - Please mute if not talking
 - No requirement for video on, *but it is nice to see faces occasionally!*
- Please be **assertive** and **active** in chat!

Class Format

- Lectures will be recorded and hosted on Canvas.
 - We will try to have lectures uploaded no longer than 24 hours after the lecture.
- However: this is a **synchronous** class:
 - I expect you to attend the synchronous lecture
 - Attendance is part of your grade (discussed later)
 - Please participate in class!
- Zoom links can be found on Canvas

TA Moderators

- Gan will attend Tuesdays
- Reese will attend Thursdays
- Asynchronously answer questions in chat
- Moderate questions
 - raise hand
 - type question

Asynchronous Discussion

- **Canvas discussion forum**
 - Non homework technical questions
 - Programming and framework questions
 - Tech news (related to parallel programming)
 - etc.

*Forum will be monitored by me and the TAs (we will try to answer in 24 hours)
Please try to help your peers!*

- **Message me or TAs in using the mailing list**
 - cse113-spring21-teach-group@ucsc.edu
 - specific homework questions
 - grading questions
 - we may ask you to post to discussion forum
 - personal emails may not be answered

Asynchronous Discussion

- **No official slack or discord for the class**
 - moderating becomes difficult
- **you can set one up yourselves if you'd like**
 - Please make it open and available to all your classmates
 - Please provide sufficient moderation (e.g. be nice to each other!)
 - Please do not cheat
 - Please remember that anything that is not in Canvas may not be private
 - If there are issues, please let me or a TA know!

Office Hours

- **Office hours daily!**
 - Spread between AM and PM to accommodate many timezones
- **My office hours:**
 - 3 - 5 PM on Wednesdays
 - Zoom link in Canvas
 - Sign up sheet
 - link will be posted in Canvas 10 minutes before
 - Email (on canvas) for reservations (requires conversation topic)
 - Strict with timing to make sure it is fair!

Office Hours

- **Monday:** TA office hour from 10:30 AM - 11:30 AM
- **Tuesday:** TA office hour from 10:30 AM - 11:30 AM
- **Wednesday:** Instructor office hours from 3:00 PM - 5:00 PM
- **Thursday:** TA office hour from 4:00 PM - 5:00 PM
- **Friday:** TA office hour from 3:00 PM - 4:00 PM

Office Hours

- **Sign up sheet**

Office Hours

- No office hours for instructor or TAs this week
- Office hours for me are moved to Friday next week.

A Note on Privacy

- We will be using Zoom for remote interactions. You should know:
 - I will be recording lectures to host on Canvas for you to review. Things you do or say will be recorded. I doubt that this will be an issue, but if you want me to remove any part of the recording, please just let me know.
 - Zoom chats are not private. Please be respectful and kind and assume everyone can see what you are typing. If you are unmuted, people can hear what you are saying. If your camera is own, they can see into your home.
 - I will use best-practices to keep our Zoom lectures private to the class. Despite our best efforts, [Zoom has a checkered history w.r.t. privacy](#). However, Zoom is the best tool we have available and these risks are simply part of the reality we must deal with.

Class Content

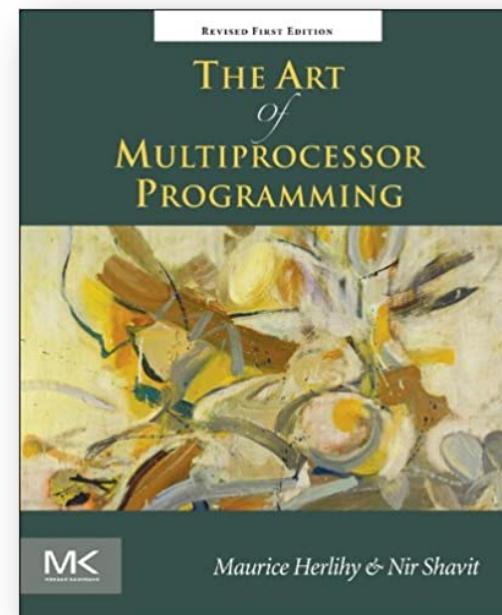
Class Content

- **20** classes, split into
- **5** modules, so there are
- **4** classes per module

- **Reference book:**

Available online from the library
Link on the webpage

Book uses Java: we will use C++



Class Content

- **Module 1: Introduction, Background and ILP** This module will introduce the class, provide an architectural refresher and discuss how modern hardware exploits parallelism within a thread (ILP).

Class Content

- **Module 2: Mutual Exclusion** This module will discuss the fundamental problem of mutual exclusion. We will discuss the theory behind mutual exclusion, how it is implemented in practice, and specialized mutual exclusion objects.

Class Content

- **Module 3: Concurrent Data Structures** This module will discuss concurrent objects and how to reason about them. We will discuss several implementations and conclude with a use-case about load balancing

Class Content

- **Module 4: Reasoning about Concurrency** This module will discuss how to reason about concurrent programs, including memory consistency and fairness.

Class Content

- **Module 5: Heterogenous Parallelism (GPGPU)** This module will discuss heterogenous programming, with a focus on GPGPU programming. We will discuss the SIMD programming model, hierarchical execution, and different architectural considerations when optimizing programs.

Class Content

- **Schedule:**

<https://sorensenucsc.github.io/CSE113-2021/schedule.html>

Readings are *suggested* and will be a useful reference for test studying and homeworks

Slides will be uploaded within 1 day of the lecture

Assignments and Tests

Assignments and Tests

- **Assignments:**
 - 1 assignment per module
 - posted the last day of the module
 - due at the the last day of the next module
 - Each homework is worth 10% of your grade (total of 50%)

Assignments and Tests

- **Format:**
 - Coding assignments in C/++ and Python
 - Using C/++ threads or OpenMP
 - These languages and frameworks are widely supported. You can run them completely locally, but we will not guarantee support on your local environment.
 - We will provide a docker image that you should be able to run locally
 - There are also UCSC computing resources. We will provide a list on Canvas.
- *Please make sure you have access to a machine with at least 4 cores!*

Assignments and Tests

Two tests: Final and Midterm

- Remote testing is difficult (as I'm sure you've all discovered)!
- I do not want to do anything invasive.
- Synchronous tests are difficult to moderate.
- We will do asynchronous tests
- Designed to take ~120 minutes
- As a warning: people take much longer on take-home tests than in-person tests!

Assignments and Tests

Midterm

- Assigned halfway through module 3
- One week
- Assigned April 29, Due May 6
- Designed to take 2 hours
- Worth 10% of grade
- *Review slides and readings*

Assignments and Tests

Final

- 1 Day
- Assigned sometime before Final Date
- Due before the end of Final Date
- Designed to take 2 hours
- Worth 30% of grade
- *Inclusive*

Assignments and Tests

What you can use for tests:

You are free to consult notes, books, or the internet. While the test is active, you are not allowed to discuss the test with another person (either in the class or online). For example, you *can* google concepts that are on the test. You *cannot* post a test question to stackoverflow.

Assignments and Tests

For assignments:

I expect submitted assignments to contain your own original work. You can refer to notes, slides, internet, etc. But do not blindly copy code.

Any part of your submission that is not your original work (e.g. code snippets from the internet) need a citation. My aim is to be lenient with cited code, but we may remove some points based on the extent. A few missing points is better than a referral for academic misconduct.

I prefer that you do not collaborate on homework with classmates. In the case that you do, please mention in the submission. Again, a few missing points is better than a misconduct referral.

Assignments and Tests

This class has a zero tolerance policy on cheating. Please don't do it. I would much rather get a hundred emails asking for help than have to refer anyone for academic misconduct.

Cheating harms you: this is the best chance in your career to take the time to really learn the class material. If you do not learn the material you will not be successful in a tech career.

Assignments and Tests

Grade Breakdown:

- 5 homeworks: 50%
- 1 midterm: 10%
- 1 final: 30%
- **attendance/quiz:** 10%

Attendance and Quizzes

- Small quiz every day - take the quiz to get the daily points
- Each day you attend and take the quiz is worth ~.5% of your grade.
- 2 days of unexcused absences.
- Email 1 day before missing to arrange an excused absence.

Attendance and Quizzes

My two absences:

Next week: April 1: Architecture Review

Next Thursday: April 8: Overview of instruction-level parallelism (ILP)

I will provide recorded lectures. You are responsible for watching them asynchronously. Please post any questions or comments to the canvas discussion.

Accessibility

UC Santa Cruz is committed to creating an academic environment that supports its diverse student body. If you are a student with a disability who requires accommodations to achieve equal access in this course, please submit your Accommodation Authorization Letter from the Disability Resource Center (DRC) to me by email, preferably within the first two weeks of the quarter. I would also like us to discuss ways we can ensure your full participation in the course. I encourage all students who may benefit from learning more about DRC services to contact DRC by phone at 831-459-2089 or by email at drc@ucsc.edu.

Thank you!

- I know this class is a risk for you! I'm a new professor teaching a class that hasn't been taught in CSE for awhile
- Thank you for giving me a chance!
- Your experiences and feedback will help shape this class for future students.
- Email is always open for comments about class material, HW assignments, etc.

Next Class

- Asynchronous: I will upload a video
- **Architecture/Compiler review:**
 - Why?
 - Parallel programming lives at the edge of the software/hardware interface.
We will need to understand architecture/compiler basics in order to program efficient and correct programs
 - *Good programming languages for parallel architectures is still an open problem!!*

Questions or comments?