# CSE113: Introduction to Parallel and Concurrent Programming

January 3, 2022

# Hello!



- Professor Tyler Sorensen (he/him)
  - Call me Tyler!

- **Faculty** at UC Santa Cruz Since Summer 2020
  - Second time teaching this class

- Previously
  - Post doc at Princeton
  - PhD Student at Imperial College London
  - BS/MS at University of Utah

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

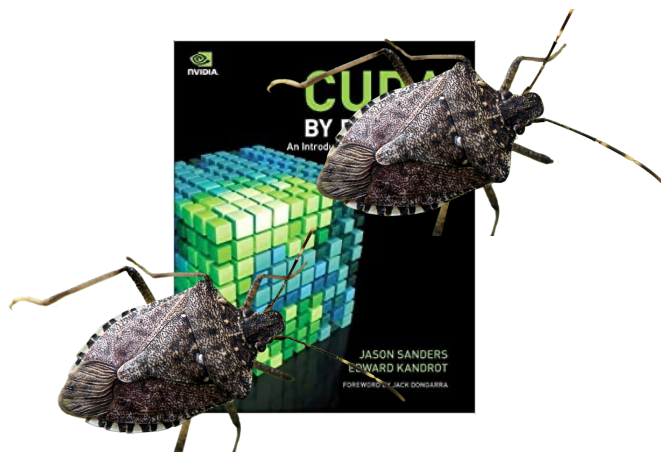# Research Interests

MS: Utah

PhD: London

Post Doc: Princeton







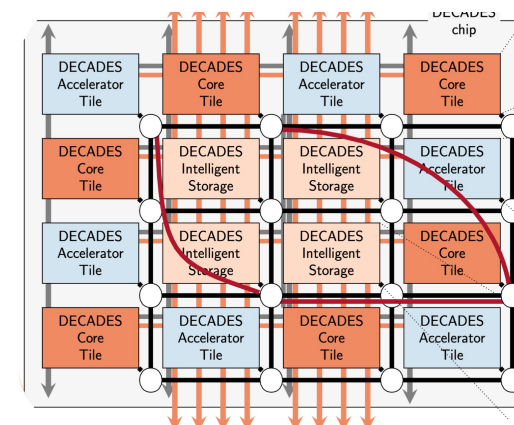Bugs in GPU programs

locks

barriers

new parallel architectures
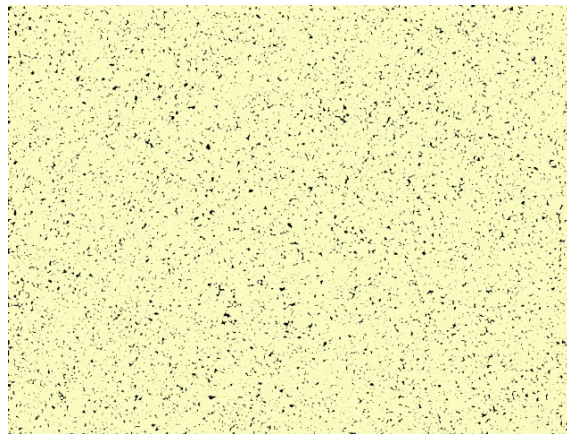






on

# Research Interests

Faculty at UCSC

individual Contributor to

parallel particle simulations

GPU memory model testing

(a)
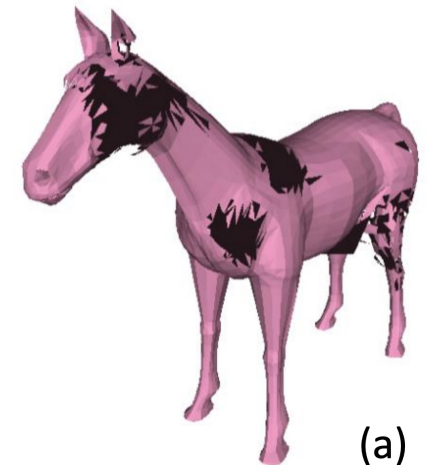
# Research Interests

- 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!**

# 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!*

# 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

# Teaching Staff Introductions

- Grad TA: Reese Levine
  - PhD student working on GPU programming models with me
  - 2$^{nd}$ time TA'ing this class

- Undergrad mentors/graders:
  - Sanya Srivastava
  - Tianhao Guo (Tim)

They are all awesome!

# Déjà vu….

- Current guidance is that we will be remote for the first 2 weeks

- After that, things will be "back to normal"™

- I understand this is not ideal and there are new challenges
  - living conditions may have changed
  - uncertainty going forward
  - unrelenting disheartening news

# We will persevere!

- We will be:
  - We will be understanding and accommodating with each other
  - We will be kind to each other
  - We will be organized and communicate with each other
  - We will adapt

- We will have:
  - a fun and productive quarter!

# Class Websites

- **Public**: https://sorensenucsc.github.io/CSE113-wi2022/index.html
  - Slides, schedule, resources

- **Private**: Canvas
  - Homeworks, grades, exams, messages, videos

- **Class forum:**
  - I will set up a piazza (unless there are other preferences?)

# 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

- **1:20 - 1:25 MWF: 65 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
  - 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!
  - Raise hand or use chat

# Class Format

- **1:20 - 1:25 MWF: 65 minutes**
  - I will try to be 15 minutes early and stay 15 minutes afterwards

- In person (hopefully in 2 weeks):
  - Engineer 2 194

- Masks make it a little harder to engage, please try anyways!

# Class Format

- For first 2 weeks: Lectures will be held synchronously
  - They will be recorded and hosted on Canvas
  - *No attendance will be recorded for the first two weeks*
  - However, I would really appreciate it if you attended and participated
  - We will try to have lectures uploaded no longer than 24 hours after the lecture.

- Zoom links can be found on Canvas

# 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.

# Class format

- Once things return to in-person (in two-weeks 🤞)

- This is an in-person synchronous class
    - I expect you to attend the synchronous lecture
    - Attendance is part of your grade (discussed later)
    - Please participate in class!

- But don't come to class sick! message us if you are not feeling well for an excused absence.

# Office Hours

- **My office hours:**
  - 3 - 5 PM on Thursdays
  - I will share a google sign up sheet (it will contain a zoom link)
  - Slots are 10 minutes
  - link will be posted in Canvas around noon that day
    - don't sign up before the Canvas announcement
    - don't sign up unless you have a question
    - sign up for 1 slot at a time
    - sometimes on the week that homework is due, we'll have a group session
  - Strict with timing to make sure it is fair!

  - No office hours this week!

# Office Hours

- Reese, Sanya and Tim will decide on theirs in the next few days
  - I'll let you know next class

# Asynchronous Discussion

- **Piazza**
  - Private message (to teaching staff) technical homework questions
  - Programming and framework questions (global)
  - Tech news (global)
  - Discussions on class material (global)

- You can also message on Canvas. Try not to email directly!

- Do not expect replies off-hours (after 5 pm, weekends, holidays)

*We will try to answer in 24 hours*
*Please try to help your peers!*

# Asynchronous Discussion

- **Additional forums**
  - You are welcome to create one yourselves
  - 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!

# Class Content

# Class Content

- **30** classes, split into
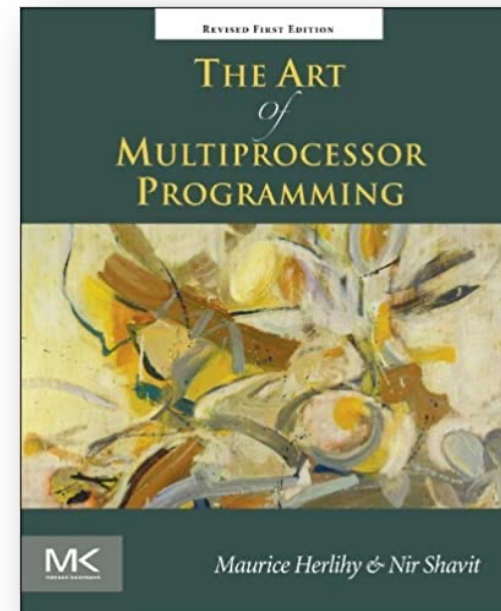
- **5** modules, so there are

- **6** 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 discuss how it can be used in 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 SIMT programming model, hierarchical execution, and different architectural considerations when optimizing programs.

# Class Content

- **Schedule:**

https://sorensenucsc.github.io/CSE113-wi2022/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
  - halfway through the module
  - due halfway through the next module
  - Each homework is worth 10% of your grade (total of 50%)

Do not expect replies off-hours (after 5 pm, weekends, holidays)

- We will try to make homeworks due at midnight. If this is an issue, we will move earlier (8 PM)

# Assignments and Tests

- **Format**:
  - Coding assignments in C/++ and Python (and some Javascript for module 5)
  - We will provide a docker image that you should be able to run locally.
  - *It must run on the docker to be graded*
  - The homework will specify constraints on the code format and submission format. It must adhere to this format to be graded!

- *It is recommended that you have access to a machine with at least 4 cores!*
  - There are some UCSC computing resources. We will provide a list on Canvas.

# Assignments and Tests

**Two tests: Final and Midterm**

- We did asynchronous tests last year. Let's plan on that again this year.

- Designed to take ~180 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 Feb 7, Due Feb 14
- Designed to take 3 hours
- Worth 10% of grade
- *Review slides and readings*

Do not expect replies off-hours (after 5 pm, weekends, holidays)

# Assignments and Tests

**Final**

- 1 Day (16 hours)
- Assigned 8 AM on March 17
- Due before 10 PM on March 17
- Designed to take 3 hours, we will monitor Piazza
- Scheduled time is 4 PM to 7 PM
- Worth 30% of grade
- *inclusive: slide material from all year*

- Do not expect help off-hours (before 9 am, after **7 pm**)

# 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.

*Please do not cheat! I like asynchronous tests but if we catch students cheating then I will have to move to synchronous tests!*

# 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.

# Discussing results

You cannot share code snippets or discuss coding solutions at a low-level.

However, unless otherwise specified: in the second week of the assignment, you can share results with your classmates.

You will have different machines and thus, your results may not align completely: it is interesting to think about why!

# Late policy

- Assignments:
  - 10% subtracted per day
  - will not be accepted after 3 days

- Tests:
  - Will not be accepted late

# Reviewing Grades

- For assignments and tests:
  - You have 1 week from when the grade is posted to discuss grades with teaching staff

# Assignments and Tests

**Grade Breakdown:**

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

# Attendance and Quizzes

- Small canvas "quiz" every day - take the quiz to get the daily points

- Quiz answers are not graded! only if you submit it

- If you do not have the ability to access Canvas during class, let me know
  - It should be possible on a smart phone…

- I will post the quiz for today right after class
  - It's a little longer than usual just to get to know everyone

# Attendance and Quizzes

- You can have 3 free absences

- More absences can be considered on a case-to-case basis
  - Don't come to class sick!

- First two-weeks I will not take attendance, but you are still required to take the quiz. There is a quiz for today I will post after class.

- Only submit the quiz if you attend! if you do not attend and submit the quiz, it is considered a breach of academic integrity

# 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.

# Website tour

# Final notes

- This class is impacted
  - We have a higher student/staff ratio than normal
  - This might affect office hours, grading time, etc.

- This class is "new"
  - Material is still being developed.
  - There may be issues on HWs and tests (please let us know if you find any!)
  - There may be schedule changes

We will do our best and make sure to stay organized and communicate clearly!

# Thank you!

- I know this class is a risk for you! I'm a new(-ish) professor teaching a class that hasn't been taught in CSE for awhile

- Thank you for giving it 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

- **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!!*