CSE 224: OVERVIEW AND INTRODUCTION

George Porter
Jan 10, 2023
ATTRIBUTION

• These slides are released under an Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0) Creative Commons license
TODO

1. Go to Canvas and take the “Onboarding Survey”
2. Start project 0 (due Jan 17)
WELCOME!
CSE 224: NETWORKED SYSTEMS

- Add networking support to software
  - Between two computers
  - Between computer and datacenter (“The Cloud”)

- Develop software that is:
  - Scalable (handles 100s of M to 1+ billion users)
  - Fault-tolerant (survives failures)
  - Evolvable (how to update services without making them unavailable to end users)
MODERN SOFTWARE INCREASINGLY NETWORKED

Endhost / Front-end
(phone, laptop, game console, electric vehicle, ...)

Networked services / Backend
(cloud computing)

Application protocols
NETWORKED SERVICES DRIVEN BY DATA

Data + Amazon.com = Product Recommendations

Data + Spotify = Custom Stations

Data + Google = Personalized Search
DATA-DRIVEN, PER-USER CUSTOMIZATION + ML

$\text{Data} + \text{amazon.com} = \text{Product Recommendations}$
MAJOR THEMES OF THE COURSE

• Programming abstractions for communicating over the Internet through various network protocols
• Naming and indexing to find services and connect clients with servers (or clients with other clients)
• Managing scale; scale-out design
• Replicating and updating “mutable” data over the network
• Replicating and caching “immutable” data over the network (think Netflix, Disney+, Youtube, etc)
• Accessing and managing networked storage
• Managing fault tolerance
HOW CAN YOU WRITE SOFTWARE THAT WORKS DESPITE ADVANCEMENTS IN UNDERLYING TECHNOLOGY?

Think about the first computer you remember using...
- Can we find the oldest example here in class today? The most recent example?

Think about the first network you used (modem? Fiber optics? Mobile network?)
- Can we find the oldest example here in class today? The most recent example?

Discuss with the 3-4 people nearest you for 2 minutes and let’s find out!
THINK ABOUT HOW ONLINE NETWORK SERVICES HAVE CHANGED OVER THE PAST 20-ISH YEARS...
Try Netflix for FREE Today!

Rent all the DVD movies you want.
For 20 bucks a month. No late fees.

Super Selection!
Create a list online of all the movies you want to see

Free & Fast Home Delivery
The movies you select arrive via first-class mail in 2-4 days.

No Due Dates or Late Fees
Keep each DVD as long as you want. Have up to 3 movies on hand.

Free Shipping!
Return one DVD in its prepaid envelope and get another DVD from your list.

World's Largest Selection

$19.95 PER MONTH
One Flat Fee!
It's just 20 bucks a month. There are no late fees, no hidden charges, no commitments. If you have any questions, call 1-888-638-3549.
TWITTER (2007)

A global community of friends and strangers answering one simple question: What are you doing? Answer on your phone, IM, or right here on the web!

Explore Twitter

Look at what these people are doing right now...

claudiof @rumors a minha pergunta nao era pa ter a correta, ha eca faxes whois e ves que sim. era filas... "e lido? consolas?" less than 5 seconds ago from twitter

mseling I've started the bad habit of forgetting to eat breakfast and lunch. No wonder I'm always tired. less than 5 seconds ago from twitter

shinshin @sharon dg it's on its way via email. Not online so can't send a link. less than 5 seconds ago from web in search of sharon

groovesalad Mushroom Nation - Helsinki (Pancake Mix) less than 5 seconds ago from web

2628 weary thuds less than 5 seconds ago from twitter

trockenjew Billions of dollars spent on dirty contractors in iraq. http://twitter.com/tykko less than 5 seconds ago from twitter

Becks725 mein bruder ist jetzt auch bei twitter... sein nickname ist Jesus!!! Bis dienten ihr ihren? less than 19 seconds ago from web

leviad End of my blog http://leviad.canalblog.com less than 19 seconds ago from web

rawaki El diseño gráfico es la peor de las parojas y la mejor de las amantes, así pasa... ¿Porqué siempre nos gustan las cabronas? zapper less than 19 seconds ago from web

© 2007 Twitter About Us Contact Blog Developer API Help Jobs Terms Privacy
Invalid File

File not found

OK
Welcome to Thefacebook!

Thefacebook is an online directory that connects people through social networks at colleges.

We have opened up Thefacebook for popular consumption at Harvard University.

You can use Thefacebook to:
- Search for people at your school
- Find out who are in your classes
- Look up your friends' friends
- See a visualization of your social network

To get started, click below to register. If you have already registered, you can log in.

Register  Login
When was the web protocol created? When was the first graphical web browser released?

1. 1968 / 1972
2. 1974 / 1976
3. 1989 / 1993
4. 2001 / 2002
THE DEPLOYMENT OF “THE WEB”
RISE OF THE WEB

RISE OF THE WEB


Web Created
WHERE DO NETWORK SERVICES EXECUTE?
THE FIRST WEB SERVER (NEXT WORKSTATION, 1991)
THE RISE OF THE “DATACENTER” (AKA CLOUD COMPUTING)

1989
Web Created

1993

1997

2001

2005

2009

2013

Google’s 1st cluster
DATACENTERS:
THE HOME OF ALL THIS COMPUTING AND STORAGE

Google

Microsoft

Google

Facebook
HARDWARE HAS EVOLVED AS WELL. STARTING WITH CPUS...
TO GPUS...
TO PROGRAMMABLE FPGAS...
TO CUSTOM DESIGNED CHIPS

Google
Tensor Processing Unit
YouTube is now building its own video-transcoding chips

Google throws custom silicon at YouTube’s massive video-transcoding workload.

RON AMADEO - 4/22/2021, 11:24 AM
CLUSTERS OF CUSTOM ASICS FOR AI/MACHINE LEARNING

Source: google.com
FULL CLOUD NATIVE LANDSCAPE
CLOUD NATIVE LANDSCAPE IN A 10-WEEK QUARTER
BACKEND DEVELOPMENT ROADMAP

Internet
- How does the internet work?
- What is HTTP?
- Browsers and how they work?
  - HTML
  - Terminal Usage
  - How OSs work in General
  - Process Management
  - Threads and Concurrency
  - Basic Terminal Commands: grep, awk, sed, ls, curl, wget, tail, head, less, find, ssh, kill

Basic Frontend Knowledge
- CSS
- JavaScript
- Memory Management
- Interprocess Communication
- I/O Management
- POSIX Basics: atd, atdout, atderr, pipes
- Basic Networking Concepts

OS and General Knowledge
- DNS and how it works?
- What is Domain Name?
- What is hosting?
- Rust
- Go
- Learn a Language
  - Make sure to learn it's quirks. Core detail about it's runtime e.g. concurrency, memory model etc.
  - Java
  - C#
  - PHP

Version Control Systems
- What are they and why you should use one
  - Basic Usage of Git

Repo hosting services
- Create account and Learn to use GitHub
  - GitHub
  - Gitlab
  - Bitbucket
BACKEND DEVELOPMENT ROADMAP

Graph Databases
- Neo4j
- GraphQL
- WebSockets
- Web Servers
- Relay Modern

Building for Scale
General topics that you should learn and care about for the sustainability of the product.

- Graceful Degradation
- Throttling
- Backpressure
- Loadshifting
- Circuit Breaker
- Instrumentation
- Monitoring
- Telemetry
- Understand the Diff.
- Keep Learning

Migration Strategies
Horizontal vs Vertical Scaling

Building with Observability in mind
Metrics logging and other observable items that could help you in debugging and solving the issues when things go wrong
THE ENVIRONMENTAL IMPACT OF CLOUD COMPUTING

• Carbon/energy footprint:
  • 1-2% of global energy consumption\(^1\)
  • 140 billion kWh (50 power plants)\(^2\)
  • 100 metric tons of carbon pollution per year\(^2\)

1. LBNL, 2013
2. NRDC report
Google’s energy footprint

**RENEWABLE ENERGY PURCHASING COMPARED WITH TOTAL ELECTRICITY USE**

- Total electricity consumption
- Renewable energy %

<table>
<thead>
<tr>
<th>Year</th>
<th>Total electricity consumption</th>
<th>Renewable energy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>2.0 TWh</td>
<td>34%</td>
</tr>
<tr>
<td>2012</td>
<td>3.5 TWh</td>
<td>35%</td>
</tr>
<tr>
<td>2013</td>
<td>3.7 TWh</td>
<td>37%</td>
</tr>
<tr>
<td>2014</td>
<td>4.8 TWh</td>
<td>48%</td>
</tr>
<tr>
<td>2015</td>
<td>6.1 TWh</td>
<td>61%</td>
</tr>
<tr>
<td>2016</td>
<td>10.0 TWh</td>
<td>100%</td>
</tr>
<tr>
<td>2017</td>
<td>10.0 TWh</td>
<td>100%</td>
</tr>
<tr>
<td>2018</td>
<td>12.2 TWh</td>
<td>100%</td>
</tr>
<tr>
<td>2019</td>
<td>12.2 TWh</td>
<td>100%</td>
</tr>
</tbody>
</table>
• Network primitives are designed to scale

• Techniques we learn are directly applicable to global-scale services like Google, Facebook, ...

• Your projects will be tested in small scale
  • Yet could scale immensely with minimal to no modifications
HOW TO BUILD SUCH LARGE SYSTEMS?
HOW TO BUILD SUCH LARGE SYSTEMS?

- Systems...
- Built on top of abstractions...
- Built on software...
- Built on hardware...

We will cover the software abstractions to enable you to write networked software.
IT’S NOT JUST WEBSITES AND SOCIAL MEDIA THOUGH!
CSE 224 VS \{221,222A,223B\}

- 224: Graduate Networked Systems
  - How to program networked software
  - Socket programming, RPC, protocol design and implementation, consensus and consistency, security, TLS, ...
  - Designed as a broad survey of systems thinking
  - Learn through hands-on, programming-based projects

- 224 Target audience:
  - MS “comps” students and BS/MS students
  - Non-systems MS “thesis” and non-systems Ph.D. students

- Note:
  - Cannot receive credit for both 124 and 224

- Research-focused depth sequence
  - 221: Operating Systems
  - 222A: Networking
  - 223B: Distributed systems theory
  - Deep dives into peer-reviewed literature
  - Learn through close readings and in-class discussion of 4 research papers per week

- 221/222A/223B Target audience:
  - Systems MS “thesis” and Systems Ph.D students
THE CHALLENGE OF NETWORKING

• CS undergraduate curricula includes:
  • Algorithms
  • Programming languages
  • Architecture
  • Data structures
  • Etc...
• How does the network change each of these areas?
RESOURCES

- Website
  - https://canvas.ucsd.edu/courses/43955
  - Gradebook, links to assignments + deadlines, PDFs of lecture slides, in-class demos and exercises
- Piazza discussion board (linked off Canvas)
- Github (for managing your projects)
- Gradescope (for submitting your projects)
- Two books
- TA discussion section (1x week)
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Linfang He</td>
<td>Amanda Tomlinson</td>
</tr>
<tr>
<td>Saketh Khandavalli</td>
<td>Abhishek Vijeev</td>
</tr>
</tbody>
</table>
CLASS MEETINGS

• Mostly putting the material that you read into context

• Live coding demos, activities, some “mini lectures” on algorithms, protocols, etc.

• You are responsible for everything that happens during class
  • Will podcast, but can’t guarantee that system works flawlessly

• Will be asking for feedback on what works and what doesn’t work a lot during the class
Free if accessed through the UCSD library

Free if accessed through the UCSD library
PROGRAMMING SKILLS FOR THIS CLASS

• We’ll be using the “Go” language
  • golang.org
  • Designed at Google in 2007
  • Goals: improve programming productivity in an era of multicore, networked machines, and large codebases
  • Kernighan (of ‘C’ fame) co-created

• Why?
  • Simple, readable, no mem allocation (similar to Python)
  • High-performance networking
  • Concurrency/parallelism
  • Static typing and efficient runtime
  • Industry-quality and deployed at massive scale
1. Pre-lecture review question sets [5%]

2. Projects [60%]
   1. [5%] Single-node sort
   2. [10%] Distributed network sort
   3. [15%] Build your own web server
   4. [10%] SurfStore “Dropbox clone”
   5. [5%] Scaled-out SurfStore backend
   6. [15%] Fault-tolerant SurfStore backend

3. Exams [35%]
   1. [15%] Midterm (Thu Feb 9 during class time)
   2. [20%] Final exam (Tue Mar 21, 3-6pm)
## Default UCSD Grading Scheme

<table>
<thead>
<tr>
<th>Name</th>
<th>eGrades</th>
<th>Range:</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>&gt; 100.0%</td>
<td>97.0%</td>
<td>97.0%</td>
</tr>
<tr>
<td>A</td>
<td>&lt; 97.0%</td>
<td>94.0%</td>
<td>94.0%</td>
</tr>
<tr>
<td>A-</td>
<td>&lt; 94.0%</td>
<td>90.0%</td>
<td>90.0%</td>
</tr>
<tr>
<td>B+</td>
<td>&lt; 90.0%</td>
<td>87.0%</td>
<td>87.0%</td>
</tr>
<tr>
<td>B</td>
<td>&lt; 87.0%</td>
<td>84.0%</td>
<td>84.0%</td>
</tr>
<tr>
<td>B-</td>
<td>&lt; 84.0%</td>
<td>80.0%</td>
<td>80.0%</td>
</tr>
<tr>
<td>C+</td>
<td>&lt; 80.0%</td>
<td>77.0%</td>
<td>77.0%</td>
</tr>
<tr>
<td>C</td>
<td>&lt; 77.0%</td>
<td>74.0%</td>
<td>74.0%</td>
</tr>
<tr>
<td>C-</td>
<td>&lt; 74.0%</td>
<td>70.0%</td>
<td>70.0%</td>
</tr>
<tr>
<td>D</td>
<td>&lt; 70.0%</td>
<td>60.0%</td>
<td>60.0%</td>
</tr>
<tr>
<td>F</td>
<td>&lt; 60.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
COURSE AT A GLANCE

• Basics of networking, sockets API, DNS
• Remote procedure calls w/ Google RPC (gRPC)
• Distributed storage as an application
• Scale-out techniques and methods
• Replicating immutable state via CDNs
• Replicating mutable state with two-phase commit and replicated state machines (+ deep dive on the RAFT protocol)
In this course you’ll learn some things in class (mostly tested via the exams), and you’ll learn some things by *doing*—*working on the projects*.

We’ll cover the big themes and high-level ideas in class, but you’ll be learning a lot of the details in the projects.

That’s why the projects are NOT designed to be done in one long session or all-nighter—you should work on them a bit each day so you can research what you need to complete them, or to talk to the TAs/myself, etc.

Start early—start often!
TAKE THE ONBOARDING SURVEY

• Required by UCSD for some reason or another regarding Federal financial aid

• BUT also, there is a very important question for those of you who need this class to graduate and plan to graduate this term...

  • I’m going to export more of you this term, not less

  • While I’m not going to enforce it, you really need to be physically attending class every time, not just relying on podcasts
• You can use the lab computers in the building, or the “ieng6” servers that can be accessed via ssh

• ssh <username>@ieng6.ucsd.edu
TODO

1. Go to Canvas and take the “Onboarding Survey”
2. Start project 0 (due Jan 17)