Scalable Web Systems

Introduction

Welcome to Scalable Web Systems

Instructor: Tim Richards

This is a new class! (well, sort of…)

The web has become a large and complex area for application development. Access to an abundance of open source languages, libraries, and frameworks has led to the quick and easy construction of a variety of applications with several moving parts working in coordination to present to the user the illusion of a single program. In reality, web applications are extremely difficult to get right. They involve a large collection of coordinated services, multiple databases, complicated user interfaces, security and performance issues, and ever changing 3rd party services, spread across physical and virtual machines. These complications are further stressed by the large number of concurrent users that access these applications every second. This course will investigate several well known web-based applications and the technology and software architecture used to scale these applications. We will also study a specific topic related to scalability in software design in the context of web application architecture. This course counts as a CS Elective toward the CS major (BA/BS) and as an Elective toward the INFORM major. Prerequisites: COMPSCI 326. 3 credits.

Statement of Inclusivity

The staff for this course support the UMass commitment to diversity, and welcome individuals regardless of age, background, citizenship, disability, sex, education, ethnicity, family status, gender, gender identity, geographical origin, language, military experience, political views, race, religion, sexual orientation, socioeconomic status, and work experience. In this course, each voice in the classroom has something of value to contribute. Please take care to respect the different experiences, beliefs and values expressed by students and staff involved in this course.

Introduction

Welcome to COMPSCI 497S Scalable Web Systems

This course will be fully remote and mostly asynchronous.

- The course is arranged in a weekly format where each week focuses on a particular topic or set of topics.
- Each week will include readings, videos, articles, and assignments
- This goal of this course is to study topics related to scalability in web systems/applications with a particular emphasis on microservices.
- We will study microservices, but we will draw on other areas to better understand the issues involved in scaling web software.
- This course is project-based and you will work with a team of students to design and implement a web application.

Although we hope we are able to stick to this format, it may require changes...

Asynchronous Teaching and Learning

What does it mean to **teach** asynchronously?

What does it mean to **learn** asynchronously?

Why did I decide to have this course fully asynchronous?

How can we make asynchronous learning be just as interactive and fun?

This is certainly odd times, but with a little bit of creativity I believe that we can all be successful in this remote environment.

Remote Communication

I have been teaching Data Structures online during the summer for 7 years. The most important aspect of the course is to provide as much communication opportunity as possible.

However, there are problems with synchronous communication:

- Different time zones - can’t attend a synchronous class
- Connectivity issues - can’t attend a synchronous class
- Privacy issues and comfort level of opening up your living space to other students and instructors

<u>These are **real** difficulties, so we must treat them seriously and with understanding</u>

Remote Communication

For the reasons just mentioned, I promote heavily an asynchronous communication style with a light mix of synchronous:

- Online discussion forums - we must be quick to respond (as best we can)!
- Online messaging - for synchronous virtual office hours (and direct communication between me and student-to-student)
- Zoom for those who can meet synchronously

It will also be important for everyone taking the course to be flexible as well as the myself to be flexible (or at least as best we can) in order to tap into everyone’s ability to succeed remotely.

I will certainly do our best to accommodate and be understanding

Prerequisites

COMPSCI 326 Web Programming is required to take this course.

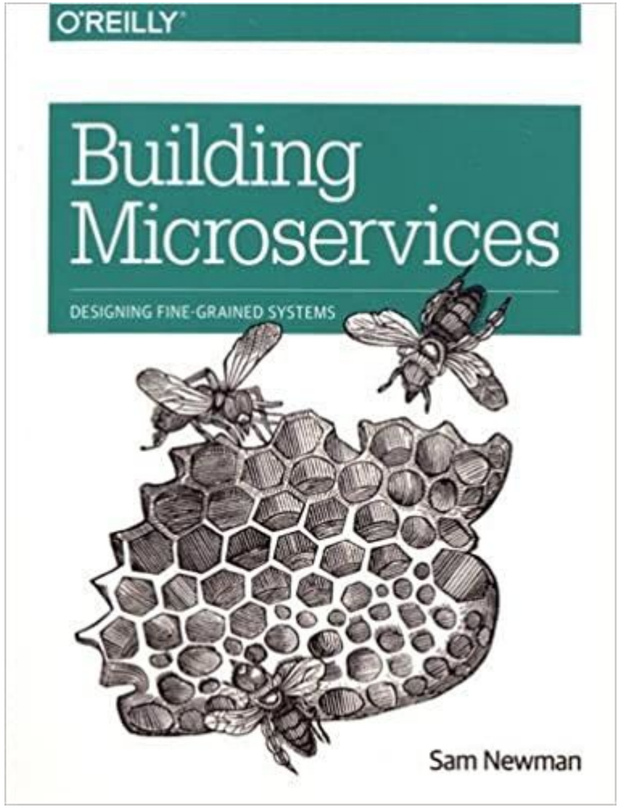
It is recommended to have (COMPSCI 220 Programming Methodology and COMPSCI 230 Computer Systems Principles) or 377 Operating Systems, however, this is not required but it will make your experience more rich. (e.g., background in web app deployment, basic security, network protocols, processes, client/server architecture, and general operating systems concepts)

Textbook

Textbook:

Building Microservices: Designing Fine-Grained Systems

This book is also available for free from the UMass Safari subscription.



Topics

- Week 1 - Microservices
- Week 2 - What is an evolutionary architect?
- Week 3 - How do you model services?
- Week 4 - Integration (1)
- Week 5 - Integration (2)
- Week 6 - Splitting the monolith (1)
- Week 7 - Splitting the monolith (2)
- Week 8 - Deployment
- Week 9 - Testing
- Week 10 - Monitoring
- Week 11 - Project
- Week 12 - Project
- Week 13 - Project

Topics - Week 1 - Microservices

- **Week 1 - Microservices**
- Week 2 - What is an evolutionary architect?
- Week 3 - How do you model services?
- Week 4 - Integration (1)
- Week 5 - Integration (2)
- Week 6 - Splitting the monolith (1)
- Week 7 - Splitting the monolith (2)
- Week 8 - Deployment
- Week 9 - Testing
- Week 10 - Monitoring
- Week 11 - Project
- Week 12 - Project
- Week 13 - Project

We will investigate what microservices are

Topics - Week 2 - What is an evolutionary architect?

- Week 1 - Microservices
- **Week 2 - What is an evolutionary architect?**
- Week 3 - How do you model services?
- Week 4 - Integration (1)
- Week 5 - Integration (2)
- Week 6 - Splitting the monolith (1)
- Week 7 - Splitting the monolith (2)
- Week 8 - Deployment
- Week 9 - Testing
- Week 10 - Monitoring
- Week 11 - Project
- Week 12 - Project
- Week 13 - Project

We will look at how to architect a system involving microservices
Different programming languages, idioms, technologies, etc.
How do we go about making these decisions?

Topics - Week 3 - How do you model services?

- Week 1 - Microservices
- Week 2 - What is an evolutionary architect?
- **Week 3 - How do you model services?**
- Week 4 - Integration (1)
- Week 5 - Integration (2)
- Week 6 - Splitting the monolith (1)
- Week 7 - Splitting the monolith (2)
- Week 8 - Deployment
- Week 9 - Testing
- Week 10 - Monitoring
- Week 11 - Project
- Week 12 - Project
- Week 13 - Project

How to think about the boundaries of microservices that will hopefully maximize the upsides and avoid some of the potential downsides.

Topics - Week 4/5 - Integration

- Week 1 - Microservices
- Week 2 - What is an evolutionary architect?
- Week 3 - How do you model services?
- **Week 4 - Integration (1)**
- **Week 5 - Integration (2)**
- Week 6 - Splitting the monolith (1)
- Week 7 - Splitting the monolith (2)
- Week 8 - Deployment
- Week 9 - Testing
- Week 10 - Monitoring
- Week 11 - Project
- Week 12 - Project
- Week 13 - Project

How do we integrate microservices?
What are the challenges?
What are the pitfalls?

Topics - Week 6/7 - Splitting the monolith

- Week 1 - Microservices
- Week 2 - What is an evolutionary architect?
- Week 3 - How do you model services?
- Week 4 - Integration (1)
- Week 5 - Integration (2)
- **Week 6 - Splitting the monolith (1)**
- **Week 7 - Splitting the monolith (2)**
- Week 8 - Deployment
- Week 9 - Testing
- Week 10 - Monitoring
- Week 11 - Project
- Week 12 - Project
- Week 13 - Project

This is about change.
How do we decompose large applications into smaller ones?
What are the right tools to do this?

Topics - Week 8 - Deployment

- Week 1 - Microservices
- Week 2 - What is an evolutionary architect?
- Week 3 - How do you model services?
- Week 4 - Integration (1)
- Week 5 - Integration (2)
- Week 6 - Splitting the monolith (1)
- Week 7 - Splitting the monolith (2)
- **Week 8 - Deployment**
- Week 9 - Testing
- Week 10 - Monitoring
- Week 11 - Project
- Week 12 - Project
- Week 13 - Project

Deploying a large application is straightforward (kind of)
With microservices it becomes more complicated.
So, how do you do it?
what are the traps?

Topics - Week 9 - Testing

- Week 1 - Microservices
- Week 2 - What is an evolutionary architect?
- Week 3 - How do you model services?
- Week 4 - Integration (1)
- Week 5 - Integration (2)
- Week 6 - Splitting the monolith (1)
- Week 7 - Splitting the monolith (2)
- Week 8 - Deployment
- **Week 9 - Testing**
- Week 10 - Monitoring
- Week 11 - Project
- Week 12 - Project
- Week 13 - Project

How do you test all of this stuff?
Unit tests, service tests, end-to-end tests, oh my!

Topics - Week 10 - Monitoring

How do you monitor any web system?
A single program is one thing, but 1000 services?
What happens when something goes wrong and where?

Assessment and Grading

Percentage	Component
30%	Assignments
40%	Project Assignments
15%	Team Meetings
15%	Project Presentation

Lots of your time in this course will be working on your project!

Course Software / Platforms

We will primarily be using the following platforms for course material, project submissions, and communication:

- Moodle - all material will be distributed on Moodle
- Piazza - all communication will be conducted on Piazza

Additional software:

- Zoom - for appointments and team meetings

Project

This course has a semester-long project. It will be a **topic of your choice** and **technology of your choice**, but key aspects of the project will connect directly with scalability and the material we cover.

- A team of 4-5 students
- Working throughout the entirety of the semester
- Weekly progress updates (team meetings)
- 4 project submissions over the semester
- 1 final project presentation (recorded demo and presentation)