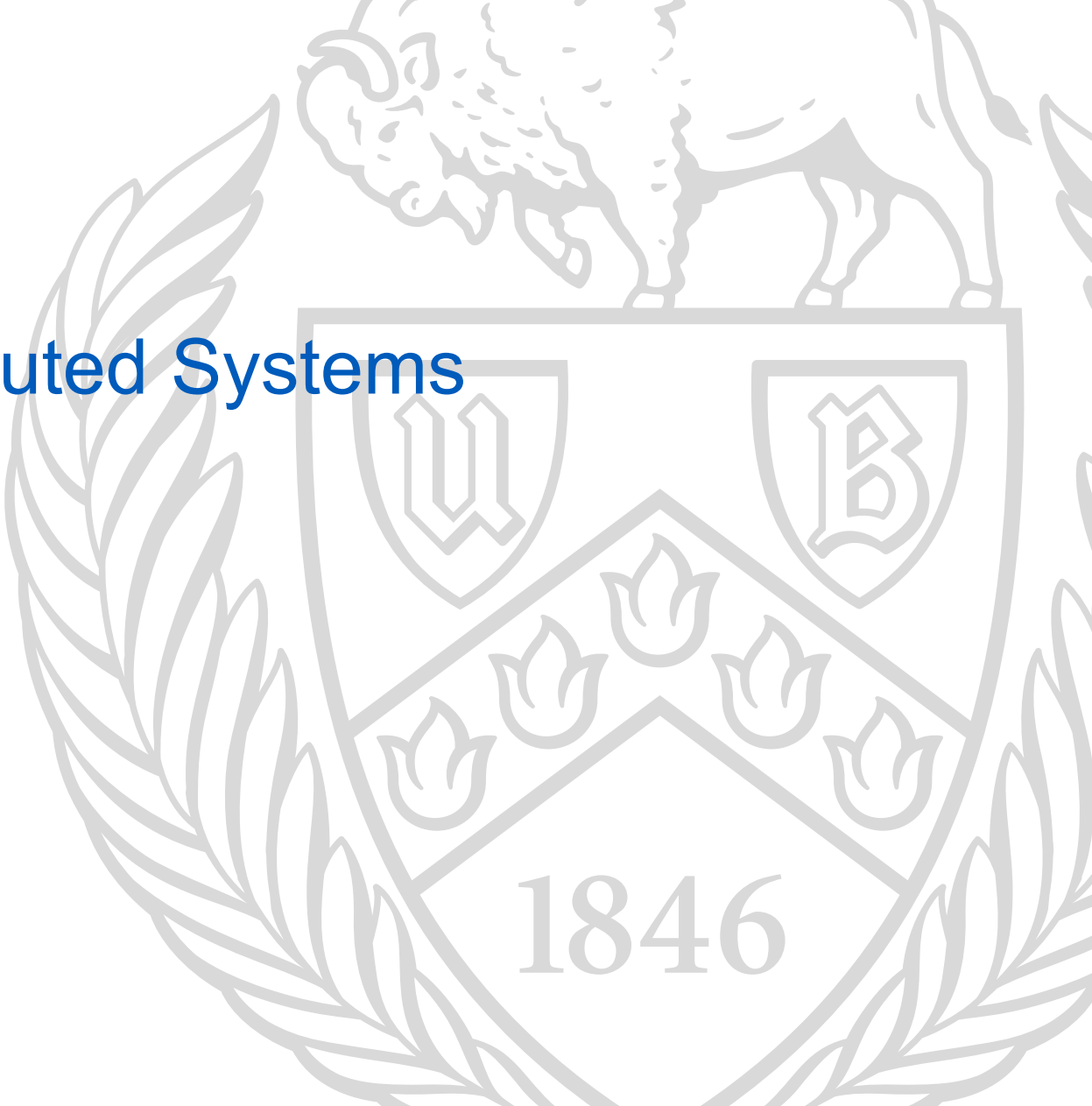
Introduction

CSE 486/586: Distributed Systems

Ethan Blanton

Computer Science and Engineering
University at Buffalo



Welcome to CSE 486

My name is Ethan Blanton

Contacting me:

Email:	eblanton@buffalo.edu
Office:	Davis 334
Office Hours:	Mon 13:00-13:50, Wed 09:00-09:50

Syllabus and slides are on the web page, OH info is on Piazza.

Distributed Systems

Distributed systems are:

- multiple processes
- possibly spread out over different networked components
- communicating by passing messages

Many modern software services are distributed systems.

Distributed Systems are Hard

What makes them hard?
In a word: failures.

Computers fail, networks fail, software fails.

Failures can be incidental (*e.g.*, power failures).

Failures can be orchestrated (*e.g.*, a virus or worm).

Add to this challenges of scale and concurrency.

Failures

Large systems are unreliable:

Backblaze reported a 1.55% drive failure rate in Q3 2024 [6].

Facebook says “In general, we always expect some degree of hardware failure in our data centers” [7].

Microsoft Azure reports that “Last year [written Dec 2020] alone, there were hundreds of routing outages or incidents, such as route hijacking and leaks.” [8]

High-profile service failures are a steady drumbeat in the news. [9] [10] [11] [12]

Scale

Distributed systems can be mind-bogglingly huge:

YouTube serves over a billion hours a day [13].

The Facebook Scuba Trailer service processes more than 180 GB per second of streaming data [14].

Slack produces 100s of Terabytes of observability data (debugging information) per day [15].

Concurrency

All of this happens concurrently:

Over 500 hours of video are uploaded every minute to YouTube servers [13].

The Facebook Scuba Trailer runs on over 2,000 machines with a total of over 96,000 cores [14].

Expectations

For this course, I expect that you:

- Will be respectful to me, TAs, classmates
- Attend every lecture
- Adhere strictly to the academic integrity policy
- Will seek assistance early if necessary
- Meet prereqs; among other things:
  - Have significant experience programming
  - Have a basic understanding of OS interfaces and networking
  - Can rapidly learn (or already know) Go

Most of all, behave as adults and strive to maximize your and your classmates' learning experience in this course.

Communication

We will communicate via:

- Piazza (our main tool; you should already be added)
- UB Email (check yours regularly)
- UBlearns (gradebook, Panopto videos, *etc.*)

Ensure that you have access to all of these things.

Having a Pleasant Semester

I intend for this course to be fun and rewarding.

You'll get out of it what you put in; no more, no less.

If you're willing to put in the time, I'm willing to help.

Basic Academic Integrity Policies

TL;DR: If you cheat, you will fail.

No generative AI is permitted!

You must be familiar with the official policies [1] [2] [3] [4].

- All assignments and exams in this course are individual.
- Collaboration with other students is forbidden unless explicitly stated otherwise.
- Only course-supplied resources are allowed (more later).

The default sanction in this course is failure in the course.

Amnesty Policy

If you commit an AI violation and wish to retract it, you can.

However, it must be retracted before I notice it.

Retracted assignments:

- Receive a zero in the gradebook
- Are not processed as an AI violation

Retractions must be in writing and explain the violation.

(See your syllabus.)

Allowable Resources

The only resources that are allowed for this course are:

- The lecture slides I provide
- Material covered in lecture
- Required or recommended readings from lecture
- Documentation on go.dev
- Distributed Computing by Kshemkalyani and Singhal [5]

Resources other than these are not allowable, and use of other resources may result in AI proceedings.

Example Problematic Resources

Examples of resources that may land you in hot water:

- VS Code default-on LLM tools
- Stack Overflow
- GitHub
- Geeks for Geeks
- Chegg, Course Hero, *etc.*
- ChatGPT, Copilot, Cursor, *etc.*
- Your work from previous semesters

This is not an exhaustive list!

Asking is Always OK

It is always OK to ask whether a resource is allowable.

It is always OK to ask course staff for assistance.

When in doubt, ask.

Programming Projects

A significant portion of your course grade will be projects.

- They are individual projects.
- Projects must be implemented in Go.
- You will receive detailed specifications.
- You will have to do your own testing.

Programming Background

You will write thousands of lines of Go for this course.

You will have to manage your time wisely.

If you are not prepared for this, resign early.

Project Assistance

Your student assistants will be your primary source of project help.

To get the most out of your SAs, do:

- try the obvious things first,
- create minimal examples to show problems, and
- consult the documentation.

To avoid wasting SA time and failing to get help, don't:

- ask for help before you've tried to understand the problem
- start at the last minute.

Office hours are (budgetarily) limited this semester!

Today's Assignments

As soon as possible:

- Read the syllabus [4]
- Complete https://go.dev/tour

Next Time …

A model of Distributed Systems

Bibliography

Required Readings

[1] University at Buffalo. University at Buffalo Academic Integrity Policies. 2026.

[2] UB CSE. Department of Computer Science and Engineering Academic Integrity Policy. 2026.

[3] Ethan Blanton. Ethan's Academic Integrity Policy. 2026.

[4] Ethan Blanton. CSE 486/586 Syllabus. January 2026.

[5] Ajay D. Kshemkalyani and Mukesh Singhal. *Distributed Computing: Principles, Algorithms, and Systems*. Cambridge University Press, 2008.

Optional Readings

[6] Drive Stats Team. Backblaze Drive Stats for Q3 2025. November 2025.

[7] Fred Lin, Harish Dattatraya Dixit, and Sriram Sankar. How Facebook keeps its large-scale infrastructure hardware up and running. December 2020.

[8] Albert Greenberg. Microsoft introduces steps to improve internet routing security. December 2020.

[9] Joe Mutascio. "'Instagram is getting too raggedy': Users react as more than 30K outages reported". In: *The Indianapolis Star* (20 March 2025). visited on 2026-01-13.

[10] Dan Goodin. *A single point of failure triggered the Amazon outage affecting millions*. 24 October 2025. visited on 2026-01-13.

[11] Jonathan Vanian. *Azure outage: Microsoft still working on fix, says recovery expected in several hours*. 29 October 2025. visited on 2026-01-13.

[12] Victor Mather. "Cloudflare Says It Has Resolved Outage That Disrupted Parts of the Internet". In: *The New York Times* (18 November 2025). visited on 2026-01-13.

[13] Google. YouTube for Press, viewed 2021-01-18. no date.

[14] Yuan Mei, Luwei Cheng, Vanish Talwar, Michael Y. Levin, Gabriela Jacques Silva, Nikhil Simha, Anirban Banerjee, Brian Smith, Tim Williamson, Serhat Yilmaz, Weitao Chen, and Guoqiang Jerry Chen. "Turbine: Facebook's Service Management Platform for Stream Processing". In: *IEEE International Conference on Data Engineering*. April 2020.

[15] Suman Karumuri, Franco Solleza, Stan Zdonik, and Nesime Tatbul. “Cloud Observability: A MELTing Pot for Petabytes of Heterogeneous Time Series”. In: *11th Annual Conference on Innovative Data Systems Research*. January 2021.

Copyright

To retrieve a copy of this material, or related materials, see https://www.cse.buffalo.edu/~eblanton/.