ECE 473/573
Cloud Computing and Cloud Native Systems
Lecture 01 Introduction

Professor Jia Wang
Department of Electrical and Computer Engineering
Illinois Institute of Technology

August 18, 2025

Outline

Administrative Issues

Introduction to Cloud Computing

Reading Assignment

- This lecture: Course Syllabus
- Next lecture: 1,6

Outline

Administrative Issues

Introduction to Cloud Computing

Instructor

- Professor Jia Wang
- E-Mail: jwang34@illinoistech.edu
- Office hours: TBD

Lectures

- Mon./Wed. 1:50 PM – 3:05 PM
- Hermann Hall MEZZANINE
- Course website: `https://wngjia.github.io/ece573-web/`

Textbooks

- Required Textbook
  - “Cloud Native Go: Building Reliable Services in Unreliable Environments”, Matthew Titmus, O’Reilly Media, 2021.
- Plus additional research papers

Useful Websites

- `https://www.youtube.com`
  - Find tutorials.
- `http://stackoverflow.com`
  - Learn how to communicate with professionals.

Prerequisite

- Computer programming
  - Branch and loop, function
  - Class
  - Searching and sorting
  - Debugging and troubleshooting
- Computer organization
  - Memory and pointer
  - Interrupts
  - I/O subsystems for storage and communication

Course Outline

Software system design for cloud

- Cloud computing architecture
  - Virtualization, containerization, and resource management
  - Distributed storage, communcation, and computing systems
- Cloud native system design and implementation
  - Scalability, resilience, manageability, observability
  - Go language introduction and common cloud native patterns

Course Objectives (ABET)

After completing this course, the student should be able to do the following:

1. Identify scenarios where cloud computing applies.
2. Understand virtualization and containerization techniques.
3. Explain common storage, communication, and computing patterns in cloud.
4. Describe cloud native system design techniques for security, scalability, resilience, manageability, and observability.
5. Utilize open-source software to build cloud native systems.

Homeworks/Projects

- 5 Homeworks
  - 2 points each for a total of 10 points
- 6 Projects
  - 20 points each for a total of 120 points
- Submit online in Canvas only.

ECE 473 Grading

- A: 90
- B: 80
- C: 60
- D: 55

ECE 573 Grading

- A: 115
- B: 100
- C: 80

Expectations and Late Policy

- This is a project based course and you are expected to practice your problem solving and troubleshooting skills.
- Learn how to manage deadlines. Do NOT expect to finish a project during the weekend right before its deadline.
- Late homeworks and projects will NOT be graded, unless
  - A request to extend the deadline is received by email 48 hours BEFORE the deadline.
  - With 48 hours of the deadline or after, the request should be accompanyed by extraordinary reasons with documented proof like docter's notes, or it will be rejected.
- Extraordinary reasons do NOT include
  - Lost of code or progress or report due to software, hardware, and networking failures.
  - AI assistants generate code that doesn't work.
- Resubmission/regrading of projects are not allowed.

Project Setup

- A recent Windows computer with 4 CPU cores, 16GB memory, and 512GB SSD.
  - Or access to a x64 Ubuntu server with 4 CPU cores, 8GB memory, and 100GB storage.
  - We are not able to support ARM-based computers, like Apple MacBooks and Raspberry Pi's, since many docker images don't support ARM.
  - We are not able to support computers that are more than 5 years old since virtualization would require substantial amount of computational and I/O power.
- Internet access is required to access common code and package repositories like GitHub and Docker Hub.

How to ~~survive~~ succeed in this course?

- Read: all instructions are in written.
  - Tutorials, source code, documents, and don't overlook command outputs.
- Communicate: we are very happy to solve any issue you may meet but you need to let us know what's wrong.
  - `https://stackoverflow.com/help/how-to-ask`
- Learn to use AI assistants.
  - `https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/`
- Feel free to explore new computer hardware and software but make sure they do not interfere with your schedule to meet deadlines.

Ethics (Very Seriously)

- Read “IIT Code of Academic Honesty” and “IEEE Code of Conduct” (posted on the course website).
    - Projects/homeworks should be done individually.
    - Discussions on homeworks/projects are encouraged.
    - Interactions with AI assistants (prompts and answers) should not be shared since they are considered as your own work.
    - Source code from the lectures and instructions in this course can be used directly.
    - Source code from other online sources not directly related to this course may be used with proper references.
- All other writings and code should be BY YOURSELF.
    - NEVER SHARE YOUR WRITINGS/CODE WITH OTHERS!
    - NEVER USE WRITINGS/CODE FROM OTHERS!
    - NEVER POST YOUR PROJECT CODE OR ASK FOR HELP DIRECTLY ONLINE!
- Please review our Academic Honesty Guidelines.

  `https://www.iit.edu/academic-affairs/academic-honesty-guidelines`

Outline

Administrative Issues

Introduction to Cloud Computing

Meeting Personal Computing Needs

- Buy a desktop or laptop computer that meets the need.
  - Or, source parts and assemble a computer by yourself.
- Things you have taken for granted.
  - A place to use the computer.
  - Power that is readily available.
  - Internet connection that is also available.

Meeting Business Computing Needs

- Needs are more diverse than those for personal computing.
- Demand of computing power beyond a single computer.
  - Demand may vary, e.g. peak vs off-peak times.
- Facility to hold many computer servers physically.
  - Space and cooling.
  - Power delivery.
  - Network bandwidth.
- Redundancy in case of power and network failures.
- Have multiple facilities across the world.
  - Better performance to serve users locally.
  - Better redundancy in case of extreme events.
- Will existing software running on a single computer adapt automatically with multiple servers?

Rise of Cloud Computing

- Supercomputers for high-performance computing (HPC)
  - Primarily target at R&D of national interest that requires substantial computational power.
  - Specially made computers that are highly-reliable.
  - Need special skills to develop HPC software.
  - Very high up-front cost, difficult to expand.
- 2004: Google MapReduce
  - A programming model and framework to simplify software development for parallel data processing.
  - Run on cluster of commodity computer servers.
    - Automatically handle server failures.
    - Allow to expand cluster with more servers as business grows.
- 2006: Amazon Web Services
  - Including Amazon S3 for storage and EC2 for computing.
  - Allow businesses to rent computing resources on demand.
    - Without high up-front cost and long time to build physical infrastructure.

Why Cloud Computing?

- An ecomonically viable solution for business computing needs.
- Economies of scale in cloud computing
    - Fixed costs for facilities are shared among customers.
    - Bulk purchasing power allows cloud providers to negotiate better deals with suppliers on servers etc.
    - Diverse range of customers smooth the demand of computing power over time and increase resource utilization.
    - Operation of large number of servers requires automated tools to reduce human errors, which in turn reduces labor costs.
- Open-source movement reduces cost of cloud software development
    - No software license fee
    - Encourage interoperability
    - Reduce cost to train new workforce

Categories of Cloud Services

- Infrastructure as a Service (IaaS)
  - Developers rent virtualized computing resources like virtual machines, storage, and networking components.
  - Allow developers to have complete control over software stack including networking, OS, library, and application.
- Platform as a Service (PaaS)
  - Developers rent pre-configured tools, libraries, and services like databases and message queues.
  - Allow developers to focus on core business logic while maintenance of common software components are outsourced.
- Software as a Service (SaaS)
  - End-users rent applications like email that can be accessed anywhere via a browser or a dedicated client.
  - Developers manage operation and maintenance of the software application to avoid compatibility issues with end-users' hardware and software systems.
  - End-users no longer need to worry about updates and upgrades other than their own devices.

Pay-per-Use vs. Subscription-based Pricing

- IaaS and PaaS are usually charged pay-per-use.
  - Cores/memory/storage are charged per quantity per hour.
  - Network traffics are charged per bytes depending on direction.
  - Database queries are charged per each depending on functions.
  - However, it may become very complicated to estimate the amount of usage and thus charge for actual applications.
- Virtual Private Servers (VPS) provide subscription-based IaaS
  - Pre-defined packages of cores, memory, storage, and networking resources for a fixed monthly or yearly fee.
  - A good option to try cloud compting for beginners.
- SaaS are usually subscription-based.
  - End-users have a clear understanding of the cost.
  - Providers have a predictable and steady revenue stream, making it easier to plan and allocate resources.
  - Perpetual licensing is not a good fit when software needs to be updated and upgraded often.
- Overall, pay-per-use pricing allows to optimize usage to reduce costs while subscription-based pricing is more predictable.

Summary

- Cloud computing provides an ecomonically viable solution for business computing needs.