Distributed Systems

CS425/ECE428

Today's agenda

- Introductions
- Course overview
- Logistics

Instructors

Radhika Mittal
Assistant Professor, ECE & CS
257 Coordinated Science Lab

Nikita Borisov
Professor, ECE & CS
460 Coordinated Science Lab

Teaching Assistants

Qingrong Chen
2nd year MS, CS

Mahir Morshed
1st year MS, ECE

Junli Wu
1st year MCS, CS

Today's agenda

- Introductions
- Course overview
- Logistics

Today's agenda

- Introductions
- Course overview
- Logistics

Examples of distributed systems

- World Wide Web
- A cluster of nodes on the cloud (AWS, Azure, GCP)
- Multi-player games
- BitTorrent
- Online banking
- ……..

What is a distributed system?

Hardware or software **components** located at **networked** computers communicate or **coordinate** their actions only by **passing messages**.

- *Your textbook*
(Coulouris, Dollimore, Kindberg, Blair)

What is a distributed system?

A collection of **autonomous computing elements**, connected by a **network**, which appear to its users as a **single coherent system**.

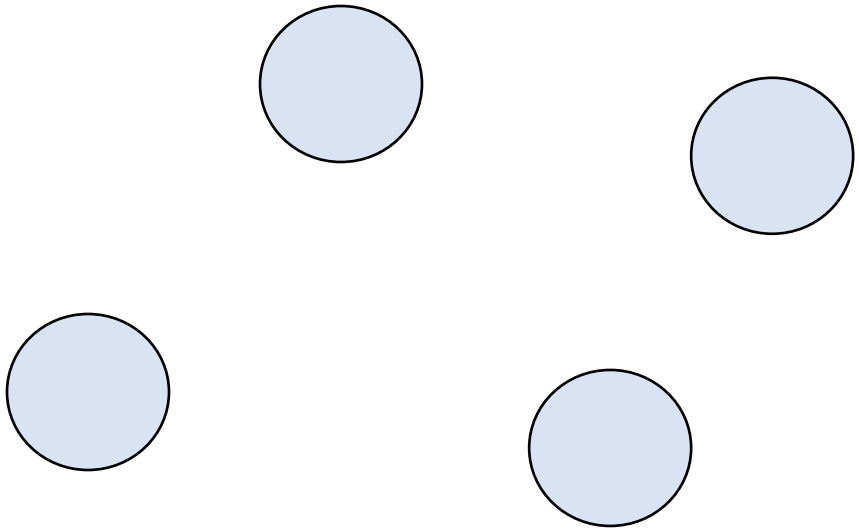
- *Steen and Tanenbaum*

What is a distributed system?

A system in which **components** located on **networked** computers communicate and **coordinate** their actions by **passing messages**. The components interact with each other in order to achieve a **common goal**.

- *Wikipedia*

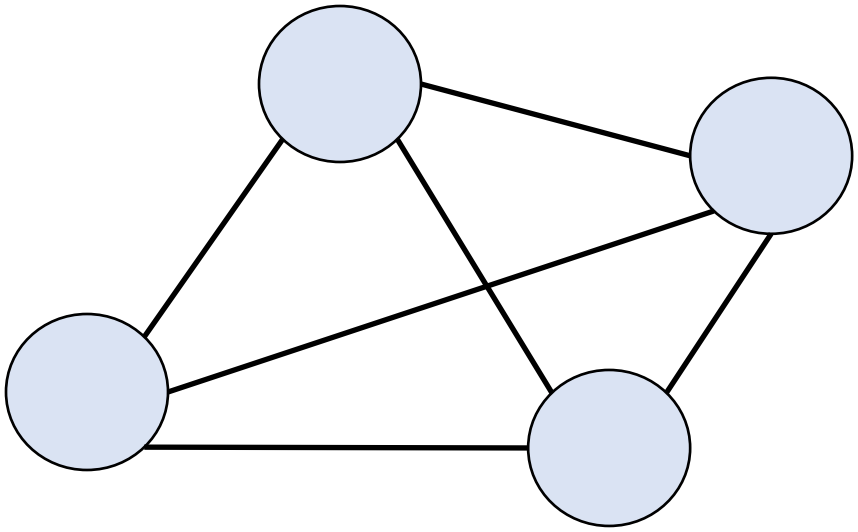
What is a distributed system?



Independent components or elements

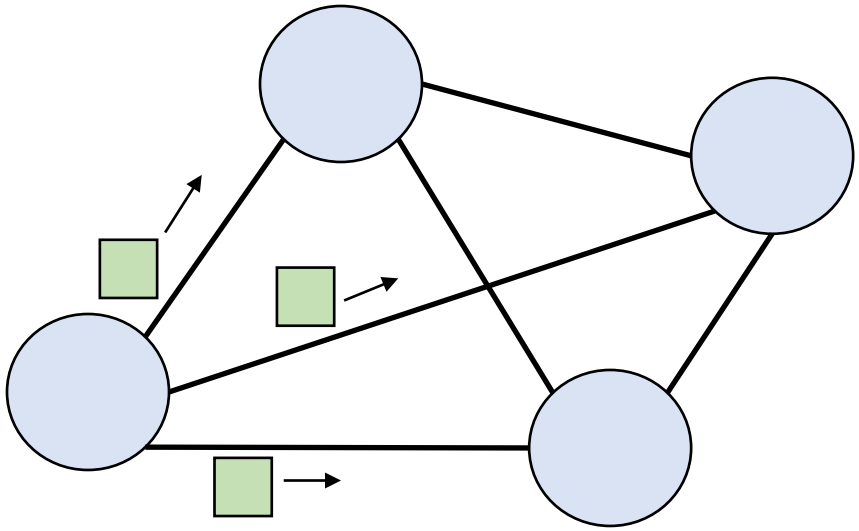
(software processes or any piece of hardware used to run a process, store data, etc)

What is a distributed system?



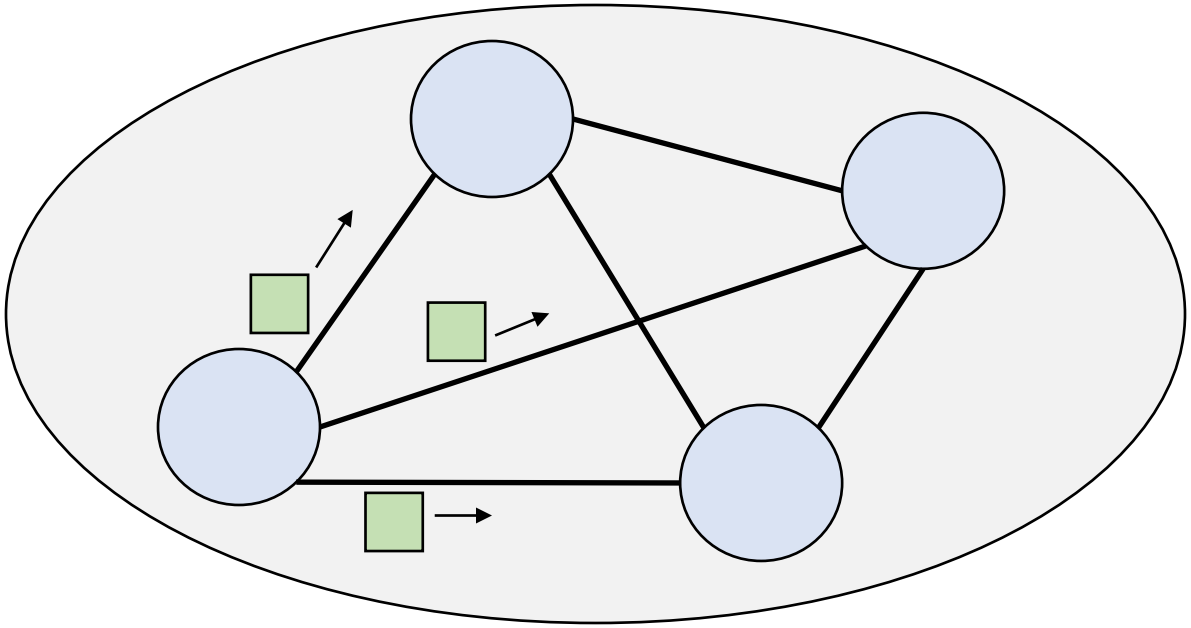
Independent components or elements that are connected by a network.

What is a distributed system?



Independent components or elements that are **connected by a network** and communicate by **passing messages**.

What is a distributed system?



Independent components or elements that are **connected by a network** and communicate by **passing messages** to achieve a common goal, appearing as a single coherent system.

What is a distributed system?

A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.

- *Leslie Lamport*

Why distributed systems?

- Nature of the application
  - *Multiplayer games, P2P file sharing, client requesting a service.*
- Availability despite unreliable components
  - *A service shouldn't fail when one computer does.*
- Conquer geographic separation
  - *A web request in India is faster served by a server in India than by a server in US.*
- Scale up capacity
  - *More CPU cycles, more memory, more storage, etc.*
- Customize computers for specific tasks
  - *E.g. for storage, email, backup.*

Example: scaling up Facebook

- 2004: Facebook started on a single server
  - Web server front end to assemble each user's page.
  - Database to store posts, friend lists, etc.
- 2008: 100M users
- 2010: 500M users
- 2012: 1B users
- 2019: 2.5B users

How do we scale up?

Example: scaling up Facebook

- One server running both webserver and DB

- Two servers: webserver, DB
  - *System is offline 2x as often!*

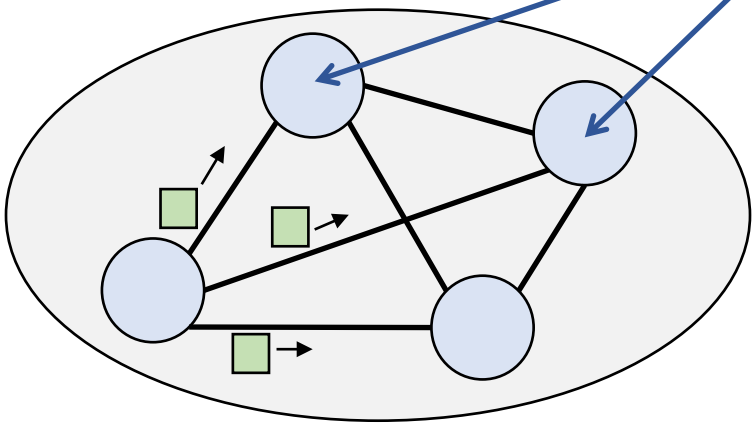
- Server pair for each social community
  - *E.g., school or college*
  - *What if server fails?*
  - *What if friends cross servers?*

Example: scaling up Facebook

- Scalable number of front-end web servers.
  - Stateless: if crash can reconnect user to another server.
  - Use various policies to map users to front-ends.

- Scalable number of back-end database servers.
  - Run carefully designed distributed systems code.
  - If crash, system remains available.

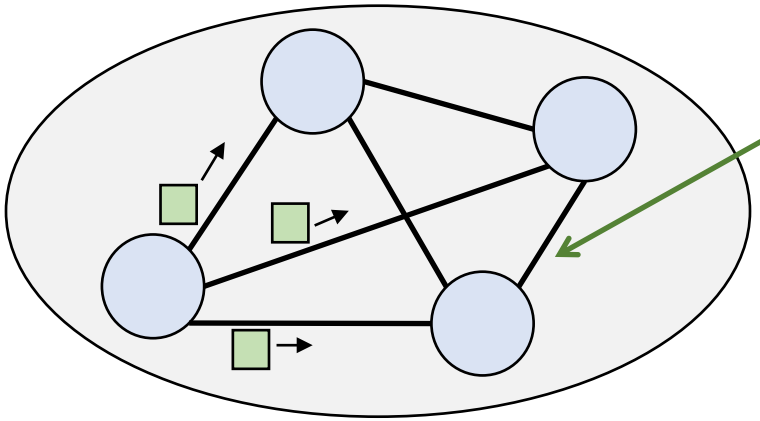
Challenging properties



Multiple computers

- Concurrent execution.
- Independent failure.
- Autonomous administration.
- Heterogeneous.
- Large numbers.

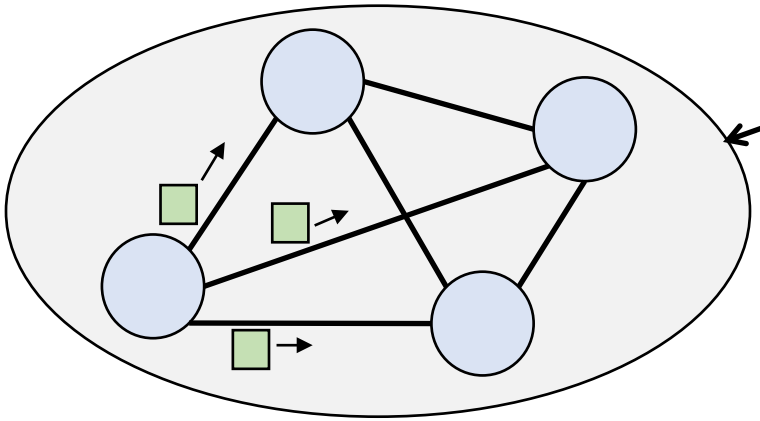
Challenging properties



Networked communication

- Asynchronous
- Unreliable
- Insecure

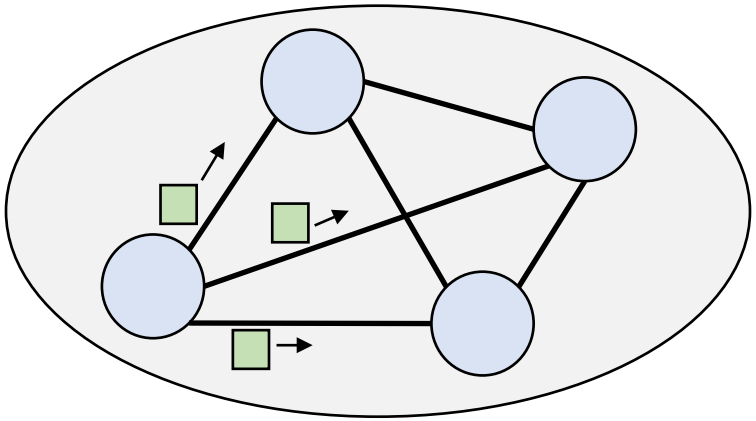
Challenging properties



Common goal

- Consistency
- Transparency

Challenging properties



Multiple computers

- Concurrent execution.
- Independent failure.
- Autonomous administration.
- Heterogeneous.
- Large numbers.

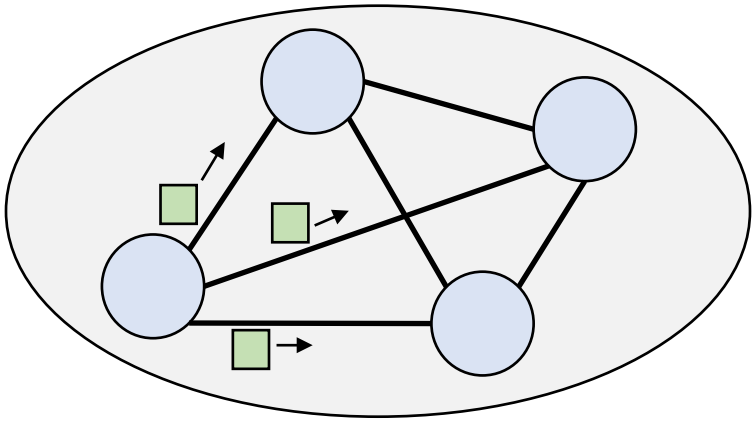
Networked communication

- Asynchronous
- Unreliable
- Insecure

Common goal

- Consistency
- Transparency

Challenging properties



Multiple computers

- Concurrent execution.
- Independent failure.
- Autonomous administration.
- Heterogeneous.
- Large numbers.

Networked communication

- Asynchronous
- Unreliable
- Insecure

Common goal

- Consistency
- Transparency

Rest of the course

- **Distributed system concepts and algorithms**
  - How can failures be detected?
  - How do we reason about timing and event ordering?
  - How do concurrent processes share a common resource?
  - How do they elect a "leader" process to do a special task?
  - How do they agree on a value? Can we always get them to agree?
  - How to handle distributed concurrent transactions?
  - ….
- **Real-world case studies**
  - Blockchains
  - Distributed key-value stores
  - Distributed file servers
  - …

Today's agenda

- Introductions
- Course overview
- Logistics

Sources of information

- Course website
  - Homeworks, MPs
  - Lecture schedule, readings, and slides
- CampusWire
  - Announcements, questions, clarifications

Course Staff

Radhika Mittal

Nikita Borisov

Qingrong Chen

Mahir Morshed

Junli Wu

Office Hours details will be made available on the website and on CampusWire shortly.

Books

- *Distributed Systems: Concepts and Design*, Coulouris et al., 5th edition.
  - Earlier editions may be acceptable.
  - Your responsibility to find correct reading sections.

- Other texts
  - *Distributed Systems: An Algorithmic Approach*, Ghosh
  - *Distributed Systems: Principles and Paradigms,* Tanenbaum & Steen
  - *Distributed Algorithms*, Lynch

Grade components

- **Homeworks**
  - 6 homeworks in total.
  - Approx every 2 weeks.
  - Will be submitted using Gradescope.
  - Must be **typed** (hand-written diagrams are fine).
  - Must be done **individually**.

Grade components

- Homeworks

- MPs (only for 4 credit version)
  - 4 mini projects.
  - First (warm-up) MP will be released on Friday!
  - Groups of up to 2
    - Need to fill up a form to activate VM clusters.
  - Supported languages: Python, Go, C/C++, Java

Grade components

- Homeworks
- MPs (only for 4 credit version)
- Exams
  - Two midterms
    - Tentative dates and times:
      - March 2nd, Mon, 7-9pm
      - April 6th, Mon, 7-9pm
  - Comprehensive final.

Grade components

- Homeworks
- MPs (only for 4 credit version)
- Exams
- CampusWire participation

Grade distribution

	3-credit	4-credit
Homework	33%	16% (drop 2 worst HWs)
Midterms	33%	25%
Final	33%	25%
MPs	N/A	33%
Participation	1%	1%

Integrity

- Academic integrity violations have serious consequences.
  - Min: 0% on assignment
  - Max: expulsion
  - All cases are reported to CS, your college, and senate committee.

- Note: any sharing of code outside group is forbidden.

Laptop/screen policy

- Research shows that:
  - Laptop use has a negative impact on student learning retention / performance.
  - Laptop use has a negative impact on ***other students*** in course.

- Policy
  - Laptops / iPads are strongly discouraged everywhere
  - If you feel you **must** use such a device, sit in side seats or back row
  - Enforcement will be lax in first two weeks.

Lecture Summary

- *Distributed Systems* properties
  - Multiple computers
  - Networked communication
  - Common goal

- Distributed systems are fundamentally needed, and are challenging to build.

- Course goals: concepts, designs, case studies

Acknowledgements

- Prof. Arvind Krishnamurthy
- Prof. Nikita Borisov
    - Prof. Jennifer Hou
    - Prof. Mehdi Harandi
    - Prof. Klara Nahrstedt
    - Prof. Indranil Gupta
    - Prof. Nitin Vaidya
    - Prof. Sayan Mitra

Questions?