Distributed Computing

A-01. Introduction to the Course

Welcome!

- This course will be shared between all 1st year students
- Different organization per curriculum
  - SSE: 6 credits (October-~November)
  - DSE: 9 credits (October-December)

Teams Channel

- We have a **Teams channel** (join it with code **bqrxc9d**) for lesson recordings
  - Please **do not** ask for permission to access videos, just join the channel
- No real advantage in following the live lesson, they are mostly there for the recording
- We'll only record this year if the content substantially changes

Learning Difficulties

- Italy recognizes **learning difficulties**, for example:
  - DSAs (*Disturbo Specifico dell'Apprendimento*: specific learning disorder)
  - BES (*Bisogno Educativo Speciale*: special educational need)
- You have the right to additional time during the exam
- If you want to share your difficulty (even anonymously), I'll try to make the lessons better for you

What Is a Distributed System?

- A collection of **autonomous computing elements** (**nodes**) that appear to its users as a **single coherent system** **(van Steen & Tanenbaum)**
  - Colored text are links… Please follow them to know more!
- Distributed systems are **everywhere**: pretty much anything you see on the Internet is one…
- Also, a single multi-core computer can be seen as a distributed system; distributed system techniques can be and are applied even there **(Bauman et al.)**

Distributed Computing is Hard

The Eight fallacies of distributed computing (whitepaper, tech talk):

1) The network is reliable
2) Latency is zero
3) Bandwidth is infinite
4) The network is secure
5) Topology doesn’t change
6) There is one administrator
7) Transport cost is zero
8) The network is homogeneous

Do I Need A Distributed System?

- **Availability**: if one computer (or 10) break down, my website/DB/fancy Ethereum app will still work
- **Performance**: a single-machine implementation won't be able to handle the load/won't be fast enough
- **Decentralization**: I don't want my system to be controlled by a single entity

- And all these things are related in non-trivial ways… **We'll see!**

Making a System Coherent

- We've seen that distributed systems are about making disparate machines **coherent**
- To be coherent, nodes need to **collaborate**
  - We need **synchronization** (there is no **global clock**)
  - We need to **manage group membership** & **authorizations**
  - We need to deal with **node failures**

Distributed Systems: a Huge Topic

- Many courses go in **depth**: you have a (small enough) topic and they teach you everything about it
- Here, we'll go in **breadth**: we'll introduce several topics at a high level, and go deeper on a few
  - Idea: giving you starting points for learning on your own, including after university

How We See the Course

- Security Software and Engineering
  - You'll be among the ones **designing** and **securing** these systems
  - You'll be the "mechanics" who will be tinkering with these systems
- Data Science and Engineering
  - You'll be **using** these systems
  - You'll be the "pilots" who need to know their systems to use them well

Organization of the Course

- Part A (6 CFU, **everybody**):
  - Distributed systems in general
  - Lessons held by Matteo Dell'Amico
- Part B (3 CFU, **DSE**):
  - Big Data Engines
  - Lessons held by Giorgio Delzanno

Exams: SSE

- Written exam (most likely on Aulaweb), with questions about the program of the course
- You'll have **assignments** based on a simulator you have to complete
  - Questions like "in this scenario, which are the best design choices"?
  - Alone or in groups of two people
- You'll **tinker** with the simulation and write reports answering the question
- There will be a peer review process: you'll be evaluated based on it

Exams: DSE

- You'll have **exercises** on Apache Spark to do during the year
- A written final examination

Part A: Some Topics We'll Touch

- Making systems **consistent**
  - **Consensus** mechanisms
- What **queueing theory** tells us
  - Effects of sharing load between servers
- Handling data efficiently
  - Modeling systems with data **replication**
  - **Erasure coding**: the gifts of coding theory
- Introduction to decentralized systems
- Introduction to big data engines