

Course introduction

Andrea Marin

a.a. 2024/2025

Università Ca' Foscari Venezia

MSc in Computer Science

Software performance and scalability

General notes

- **Teacher:**

- Andrea Marin
- Email: `marin@unive.it`

- **Scheduling of the lectures:**

- Check the Moodle web course `moodle.unive.it`
- Password: *none*

- **Meeting with the teacher**

- Thursday 15:30 - 16:30
- Any other day/time but send an email to fix the appointment



- **Henry H. Liu:** Software performance and scalability. Wiley & Son, Ed. 2009
- **Mor Harchol-Balter:** Performance Modeling and Design of Computer Systems: Queueing Theory in Action. Cambridge press 2021
- Notes by teacher



- Two options:
 - Option 1: lab + oral examination
 - Option 2: written exam + oral examination
- Bonus: Assignments will be given. Deliver the solution (originally done). You may get +2 on the final marking. Markings of the exercises will be done at the end of the course, while corrections along the course



What do we test?

The student must. . .

- know the basic notions about the scalability properties of software systems
- apply models to determine the performance of a system and propose the changes to its configuration accordingly
- know the theory behind this process and solve exercises involving theoretical notions

Examples will be provided during the course



Goals of the course

Modern computer science challenges software developers to produce product that work either on systems equipped with low-capacity computational units (such as smart-phones) or with very powerful collections of computational units (such as data centres) that have to treat huge amount of data. In both cases, the software performance becomes a key-aspect in the system engineering.

The course aims at giving the elements for the design of software architectures that satisfy some performance constraints on response time, resource usage, throughput, scalability.



Structure of the course

- The course is organised into three parts:
 - Part 1: scheduling (How do we serve requests?)
 - Part 2: performance of distributed systems (who stops my ability to scale up?)
 - Part 3: dispatching (who is going to serve my requests?)

