Sistemas Distribuídos: Class Information
3º MIEIC

Pedro F. Souto (pfs@fe.up.pt)

February 13, 2019

Staff

Lectures

- Pedro Ferreira do Souto (`pfs@fe.up.pt`)

Labs

- Hélder Fernandes Castro (`hcastro@fe.up.pt`)
- Pedro Alves Nogueira (`pnogueira@fe.up.pt`)
- Pedro Ferreira do Souto (`pfs@fe.up.pt`)

Context

Distributed Application(Def.) An application with two or more
 processes:

 1. executing on different computers
 2. communicating via messages
    - with a no negligible delay (wrt computation)

- Most applications nowadays are distributed
- Virtually any **intellectually interesting** application nowadays is
  distributed.
    - Example of an interesting **non-**distributed application?

Objectives

1. Understand the foundations of distributed computing;
2. Be able to:
   ► design and implement simple distributed applications;
   ► analyse distributed solutions and evaluate their fitness to the problem at hand.

Caveat Actually, we'll focus on the concepts related to "cloud-based computing", not so much to the "Internet-of-Things"

Prerequisites

- ▶ Operating Systems – concurrency
- ▶ Computer Networks
- ▶ Programming in C/C++/**Java**

Syllabus: Part 1/3 - Communication and Processing

Introduction

Networking (Review)

Communication Paradigms

- ► Messages
- ► Remote invocation

Processing

Syllabus: Part 2/3 - Foundations

Names and Localization

Security

Sinchronization

Replication and Consistency

Fault Tolerance

Syllabus: Part 3/3 - Applications(?)

Distributed Filesystems
Data Intensive Computing
Web-based Applications
Peer-to-peer Systems

Supporting Material

Textbook
Tanenbaum, A. e van Steen, M.
Distributed Systems: Principles and Paradigms, 3rd Ed. (2017)
(available for free upon request)

Small fun "book" focused on the data center
Distributed Systems: for fun and profit.

Java Documentation E.g.:
Java 8.0 API

(Mini-)Projects

1. Simple distributed storage application
   - ▶ Groups of 2 students
   - ▶ Due date: April 7 @ 20:00 (Sunday)
   - ▶ Demo: on first lab class after due date (i.e. on April 8)
2. Your own project
   - ▶ Must use non-blocking API
   - ▶ Groups of 4 students
   - ▶ Proposal: April 14 @ 20:00 (Sunday)
   - ▶ Due date: May 26 @ 20:00 (Sunday)
   - ▶ Demo: on first lab class after due date (i.e. last week)

Note 0 Both projects must be implemented in Java

Note 1 Both projects and their reports must be submitted via SVN/Redmine

Note 2 Both projects have the same weight.

(Mini-)Projects: Grading

- ▶ Grading is individual
- ▶ We grade each project assuming the expected number of group members (2 and 4 respectively)
- ▶ To that grade we apply a **contribution factor** computed from the contribution using a piecewise linear function:

  First project

  - ▶ "Breaking points": 33%, 50%
  - ▶ Factor: 0 at 0%, .85 at 33%, 1 at 50% and 1.10 at 100%

  Second project

  - ▶ "Breaking point": 25%
  - ▶ Factor: 0 at 0%, 1 at 25%, 1.15 at 100%

  A ceiling based on the complexity will also be applied

Do you want to keep your previous year's project grade?
  Fill this form no later than **February 17th** (i.e. by the end of this week)

Exam and Class Participation

Exam

- ▶ In Moodle
  - ▶ True/False questions
  - ▶ Multiple-choice questions
  - ▶ Open questions
- ▶ Closed books **with cheat sheet**
  - ▶ A4 (both sides)
  - ▶ **Handwritten by yourself**

  Failure to comply means your exam will be **nullified**.
- ▶ Study by the book, not by the transparencies (at least mine)

Class Participation

- ▶ Including participation Moodle's forum

Final Grade

Ordinary students

$$G = min(min(F, P) + 3, 0.45P + 0.1C + 0.45F)$$

where:

G course final grade
P average of the grades in both projects ($P \geq 8$)
C class participation
F final exam grade ($F \geq 8$)

The final grade cannot exceed in more than 3 points (in 20) the minimum of the grades in the final exam and in the project

Special status students

$$G = min(min(F, P) + 3, 0.5P + 0.5F)$$

Academic Integrity

- ▶ UP, FEUP and we take academic integrity very seriously
  - ▶ Check out the Declaração de Princípios sobre a Integridade Académica na UP
  - ▶ We believe that the majority of you follow the rules
- ▶ You are allowed to discuss the projects
  - ▶ For each project, there will be a discussion forum on Moodle
- ▶ **But** all code submitted should be either:
  - ▶ Developed by the group members
  - ▶ Or authorized by me, **and** due credit should be given both in the report and in the source file.
- ▶ We will use tools to automatically detect common code
  - ▶ **All** groups with similar code will be penalized
  - ▶ You may still help your colleagues, but you **cannot share code nor read the code of someonelse's project**

  That the projects are identical to those of last year is no excuse

- ▶ The penalty may range:

  From a zero in that project

  To failing the course

Thank you!
Questions?

Announcements

Classes start 10 minutes after the hour

- ► 11:10 on Wednesdays
- ► 8:40 on Fridays

Labs start next week

Course material available on Moodle's course page

Important dates:

	Due date
1st project	April 7 @ 20:00 (Sunday)
Proposal of 2nd project	April 14 @ 20:00 (Sunday)
2nd project	May 26 @ 20:00 (Sunday)

Demos in the first lab class after due dates.