

# Introduction and Administrative Aspects

PAMS'19

Zsolt István

[zsolt.istvan@imdea.org](mailto:zsolt.istvan@imdea.org)

# Goals of the seminar

Building and measuring the performance of complex software systems:

- What do we want to measure?
- How to design experiments?
- How to analyze results?
- How to build a model of the system?
- How to optimize the system based on the model and experiments?

# Our path

- **Lectures** that cover theory and discuss examples
  - Understanding throughput and response time
  - Basic statistical treatment of data
  - Experimental design
  - Queueing theory
- **Project** to apply the ideas from the lectures
  - Java application with multi-threaded processing

# Questions? Need help?

- Seminar website: <https://zistvan.github.io/teaching/pams19>
- Documents such as project description, slide template, etc.
  - Lecture slides
  - Announcements at top of the page
- Questions over email (to me)
  - Add “[PAMS19]” to the subject line
  - Be concise and to the point
  - There can be delay in the system ;)

# Lecture dates


- Lectures are held Tuesdays from 7PM to 9PM at IMDEA Room 302
  - Possible to move earlier if works for all
- Oct. 14 (today)
  - Performance measurement basics
  - How to present results
- Oct. 15
  - Statistical methods, Experiment planning, Factorial designs
- Oct. 21 and 22
  - Queuing theory
- Nov. 5
  - Final presentations

# Project overview

## Word Count (used as a step in many machine learning pipelines)

- Clients submit documents, server gives back <word, occurrences> tuples
- Text is converted to lower case, only English characters are kept
- Words are separated by one or more spaces

This is an example  
text, and this TEXT  
is short!



Word	Occurrences
this	2
is	2
an	1
example	1
text	2
and	1
short	1

# Skeleton is provided

- Project in Java
  - Recommended to be run on Linux for scripting reasons
  - **Has to be run in the cloud (Amazon EC2) on two machines**
  - **Prerequisite! Will not have time to cover networking/multi-threading/diskIO in Java**
- Word Count Server
  - Basic operation
  - Accepting requests using Java NIO (non-blocking IO)
  - Implemented on a single thread
- Word Count Client
  - Generates a document based on a source
  - Sends the document to the server in a loop
  - Takes measurements

# What you will have to implement

1. Cleaning the document
  - The document is in HTML format
  - We only want to count words of text (not tags\*)
2. Add fine-grained statistics gathering to server
  - Measure how long each step takes (network packet reassembly, cleaning, counting, etc.)
  - Display statistics of the server
3. Make the server multi-threaded
  - Process requests from different clients on separate threads to take advantage of modern CPUs
  - Measure how this increases/impacts the behavior of the overall system



# Writing a report

- You will be given a report outline
  - Experiments to perform
  - Graphs to create
  - Don't exceed the maximum length!
- Most important: show that you understand the behavior of the system
  - Don't just list the results, put them in the context of the system, design decisions, implementation details, etc.
- Understanding of system behavior more important than raw performance!

# Preparing the final presentation

- Exercise in presenting and explaining your system to others
- Will be held in the last lecture slot\*
  - A slide “template” to fill out
  - Short, ~10-15 minutes with Q&A
- Understanding of system behavior more important than raw performance!

# Grading

- The seminar is worth 1.5 ECTS credits (~40h effort)
  - We'll have four “proper” lectures → 8h
  - Work on the project → ~32h
- Short quizzes before/after lectures – 25%
  - Starting next week
- Project report and code – 50%
  - Deadline for submitting report and code: 04.11.2018
- Project presentation – 25%
  - Deadline for slides: 04.11.2018