Projects

Corso di Sistemi Distribuiti e Cloud Computing
A.A. 2017/18

Valeria Cardellini
Project choice and deadline

• Send me an email by January 27 2018 with the following info:
  – Group members (names and emails)
  – Chosen project

• A maximum number of available slots for each project (see course site for details)
  – Assignment with a FIFO discipline

• Communicate promptly and motivate any change to the group

• The project is valid only for A.A. 2017/18
Project delivery

• When to deliver
  – By **September 22 2018**
  – About one week before the project presentation
  – No prefixed dates for the presentation

• What to deliver
  – URL pointing to cloud storage or code repository containing:
    project code, report and (if pertaining) datasets of experimental results
  – Paper copy of the report (only the report, no code)
  – Write the report possibly as a scientific paper (**maximum 10 pages** using ACM or IEEE format)
Project presentation

• All the group members discuss their project in the same date

• What to present
  – Prepare some slides
  – Prepare a live demo of the project
  – Each group member discusses a part of the project
  – Maximum 10 minutes per member
    • I will check the time and interrupt you
  – Questions during or at the end of the presentation
Common requirements for all projects

• You can choose the programming language
• You can use support libraries and tools to develop your project (of course they should not overlap with the project goals!)
  – Be careful: their use must be properly mentioned in the project report
• System/service with configurable parameters (no hard-coded!)
  – Through a configuration file/service
• You must test all the functionalities of your developed system/service and present and discuss the testing results in the project report
Common requirements for all projects (2)

- System/service supports multiple, autonomous entities contending for shared resources
- System/service supports real-time updates to some form of shared state
- System/service state should be distributed across multiple client or server nodes
  - The only allowed centralized service can be one that supports users logging on, adding or removing clients or servers, and other housekeeping tasks
- System/service scalability and elasticity
- System/service fault tolerance, in particular system/service continues operation even if one of the participant nodes crashes (optionally, it recovers the state of a crashed node so that it can resume operation)
Cloud services

• You can use Amazon Web Services (AWS) through AWS Educate https://aws.amazon.com/education/awseducate/
  – 100 $ (at least up to the end of 2018), renewable each year
  – Additional 10 $ with the student developer pack from GitHub https://education.github.com/pack
  – Plus AWS Free Tier for 12 months (unless you have already registered for an AWS account)
  – Check the list of available services!

• Alternatively, you can use Google Cloud Platform https://cloud.google.com/free/
  – 300 $ for 12 months
  – Plus Always Free (unless you have not previously signed up for free trial)
  – Check the list of available services!
Projects overview

- Project 1: CINI Smart Cities University Challenge
  - 3/4 students per team
- Project 2: Fog Middleware
  - 3 students per team
- Project 3: Your Own Replicated Service
  - 3 students per team
- Project 4: Replicated Data Storage
  - 2 students per team
- Project 5: Your Own Microservice Application with Decentralized Data Management
  - 1/2 students per team
- Project 6: Your Own Serverless Computing Application using Lambda Functions
  - 1/2 students per team
Partecipazione alla seconda edizione del CINI Smart Cities University Challenge

Scadenza per la consegna: 8 luglio 2018

Selezione di un team vincitore che parteciperà alla competizione nazionale che si terrà a settembre 2018

Entro fine luglio il team vincitore presenterà via skype il progetto a tutti i tutor delle università partecipanti

Quale progetto? Sistema di urban analytics per il monitoraggio e controllo del traffico urbano

Permette di rilevare lo stato di congestione di incroci stradali e di controllare la temporizzazione dei semafori
Progetto 1

• Analisi di flusso di dati continuo da:
  – Sensori collocati in prossimità degli incroci
  – Sensori su mezzi pubblici
  – Smartphone di utenti registrati al servizio

• Sistema di monitoraggio:
  – Fornisce informazioni real-time sul numero dei veicoli che attraversano l’incrocio e sul tempo di attraversamento dell’incrocio

• Sistema di controllo:
  – Gestisce la durata dei semafori
Projects 2-6

- Project 2: Fog Middleware
- Project 3: Your Own Replicated Service
- Project 4: Replicated Data Storage
- Project 5: Your Own Microservice Application with Decentralized Data Management
- Project 6: Your Own Serverless Computing Application using Lambda Functions