Distributed Programming II
A.Y. 2016/17

Special Projects Proposals

This is the list of proposed special projects for DP II. If interested in applying, send an e-mail message to d1943@studenti.polito.it by November 20th, 2016 with subject “Special Project Application”. The message must include the following information:

- The id of the project(s) you are interested in (in case you specify more than one project, please sort them in order of preference)
- Your CV including the transcript of your exams and the marks you got

The special projects will be assigned by November 25th, 2015. Students who have been assigned a special project will have to confirm their decision of undertaking it and get the full documentation for their work from the teacher. When a special project is completed, the material produced for the project has to be submitted to the teacher and discussed with the teacher. Special projects must be completed by the deadline indicated in each project. Completing a special project exempts from the final test in the Lab and from the mandatory assignments (the exam terminates with the discussion of the special project).

1) Neo4JManager

This assignment consists of the following parts:

1a) Given a JSON format for representing NF-FGs (Network Function Forwarding Graphs), richer than the one that has been used for the DP2 assignments, create a corresponding XML data format (XML schema).

1b) Implement a library of Java classes for managing NF-FGs and for storing them into the NEO4J graph-oriented database (http://neo4j.com/). The library must expose methods to read and validate XML representations of NF-FGs, serialize NF-FGs to XML, upload NF-FGs into a NEO4J DB, and retrieve (using NEO4J) all the paths from one node to another node in a given NF-FG.

1c) Design and implement a RESTful web service that uses the library developed in the previous part. The web service must be able to receive an NF-FG using the previously defined XML format and to perform some operations on the received NF-FG. The operations to be implemented in the web service must allow a client to retrieve all the graphs available, create a new graph, delete an existing graph, compute all the paths from one node to another node in a given graph.

A documentation of the XML format and of the web service, a client for testing the service, and an ant script that automates compilation and running of the various programs and tests must also be
provided as part of the solution.

Deadline: the February exam session

2) gRPC interface for Verigraph

Verigraph (https://github.com/netgroup-polito/verigraph) is a RESTful web service (based on JSON) that can receive a description of an NF-FG (richer than the one used for the DP2 assignments) and of network function configurations and that can perform some reachability tests on the given NF-FG and related configurations.

gRPC (http://www.grpc.io/docs/tutorials/basic/java.html) is a new RPC system that has been proposed for micro services that need lightweight and very efficient communication.

This assignment consists of providing a gRPC interface to Verigraph. It can be divided into the following parts:

2a) study gRPC and the RESTful API of Verigraph

2b) define a gRPC interface with the same functionality as the one exposed by the Verigraph service.

2c) implement the designed interface (the implementation will simply forward requests to the RESTful API of Verigraph)

2d) implement a client for testing the gRPC service

A documentation of the gRPC interface and of the whole project and an ant script that automates compilation and running of the various programs and tests must also be provided as part of the solution.

Deadline: January 26

3) RESTful interface for Z3.

Z3 (https://github.com/Z3Prover/z3) is a Satisfiability Modulo Theory (SMT) solver developed by Microsoft, with a Java API. It can be used to check if a number of given logical formulas is collectively satisfiable or not.

The assignment consists of designing and implementing a RESTful web service that exposes a subset of the functionalities of Z3. In particular, the service must enable the uploading, updating and deleting of sets of formulas for Z3 and the execution of the satisfiability check for an uploaded set of formulas, with the reception of the corresponding result.
A documentation of the XML format for representing sets of formulas and of the web service, a client for testing the service, and an ant script that automates compilation and running of the various programs and tests must also be provided as part of the solution.

Deadline: the February exam session