Topics in chaos engineering

Title: Preventing algorithmic DOS attacks with blackbox randomization.
(Topic already selected)
Supervisor: Martin Monperrus, CSC/Theoretical Computer Science Department
Description: The goal of this thesis is to study counter-measures to algorithmic denial of service attacks. An algorithmic DOS consists of a input specifically designed by the attacker to trigger the worst case execution of a program [1]. Black-box randomization consists of identifying and injecting randomization points in software, without any knowledge of the application domain and implementation choices. The goal of this thesis is to study the usage of black-box randomization for countering algorithmic DOS attacks. The student will devise and perform a scientific experiment in this context. She/he will read the literature, implement the required software for supporting the experiment, design the inclusion criteria for subjects and run the experiment on a scientific computing grid.
[1] Denial of Service via Algorithmic Complexity Attacks
[2] Correctness Attraction: A Study of Stability of Software Behavior Under Runtime Perturbation

Title: Preventing side-channel attacks with blackbox randomization.
(Topic already selected)
Supervisor: Martin Monperrus, CSC/Theoretical Computer Science Department
Description: The goal of this thesis is to study counter-measures to side-channel attacks [1]. Black-box randomization consists of identifying and injecting randomization points in software, without any knowledge of the application domain and implementation choices. The goal of this thesis is to study the usage of black-box randomization for countering side-channel attacks. The student will devise and perform a scientific experiment in this context. She/he will read the literature, implement the required software for supporting the experiment, design the inclusion criteria for subjects and run the experiment on a scientific computing grid.
[1] Side-channel attacks on cryptographic software
[2] Correctness Attraction: A Study of Stability of Software Behavior Under Runtime Perturbation

Title: Simulate chaos engineering of Internet-scale software in the NS3 simulator.
(Topic already selected)
Supervisor: Martin Monperrus, CSC/Theoretical Computer Science Department
Description: Chaos Engineering [1] is a new software engineering technique that aims at constantly assessing and measuring resilience of production software systems. In science, simulation is a powerful approach to explore and understand a given phenomenon. The goal of this thesis is to bridge chaos engineering and scientific simulation. The student will design and prototype a simulation of chaos engineering in the NS3 internet simulator [2]. She/he will read the literature and run the experiment on a scientific computing grid.

She/he will analyze the results of the experiment with the appropriate methodology and statistical approaches.
[1] Chaos Engineering
[2] Exception Handling Analysis and Transformation Using Fault Injection: Study of Resilience Against Unanticipated Exceptions

Topics in self-healing software

Title: Replacing human code with automatically synthesized code
Supervisor: Martin Monperrus, CSC/Theoretical Computer Science Department
Description: Can we replace manual error-handling with automatic error-handling? One area of software technology research consists of devising algorithms that automatically handle software errors at runtime. A software application that uses such a technique is said self-healing, or self-repairing. The goal of this thesis is to devise, prototype and perform an experiment to study the effectiveness of one automatic error-handling called “failure oblivious computing” [1]. The idea of the experiment is to first automatically remove all error-handling routines in a codebase, then to automatically add failure-oblivious code, and finally to run the test suite [2]. By measuring and analyzing the tests that pass with failure-oblivious code will enable us to understand the quality of the failure-oblivious model under consideration.
[1] Enhancing Server Availability and Security Through Failure-Oblivious Computing.
[2] Study of Resilience Against Unanticipated Exceptions

Topics in deep learning from automatic patch generation

Title: Deep-learning based prediction of code insertion for automatic program repair (Topic already selected)
Supervisor: Martin Monperrus, CSC/Theoretical Computer Science Department
Description: A lot of automatic bug fixing generation rely on inserting new code in the program under repair, this new code is called a “repair ingredient”. The student will set up a predictive learning algorithm that, given a repair ingredient, predicts where it should be inserted in the code base. The planned methodology is as follows: 1) set up a training and evaluation dataset based on diffs 2) evaluate the state-of-the-art approaches on this task 3) devise, implement and assess a new prediction algorithm. The student will perform the experiment by running it on a scientific computing grid.
[1] ASTOR: A Program Repair Library for Java
[2] Sorting and Transforming Program Repair Ingredients via Deep Learning Code Similarities

Title: Deep learning for metrics between program traces
Supervisor: Martin Monperrus, CSC/Theoretical Computer Science Department

Description: To reason about the correctness of an automatically generated patch, one needs to compare the execution of the buggy version and the patched version on the same input. For this, there is need to defined an appropriate distance metrics between program traces in Java. The student will 1) extend an existing tool to collect program traces 2) collect a dataset of program traces 3) devise and implement different techniques for computing a distance between two traces 4) compare the effectiveness of those techniques. The student will perform the experiment by running it on a scientific computing grid.

Core reading;

[1] Identifying Patch Correctness in Test-Based Automatic Program Repair

[2] Test Case Generation for Program Repair: A Study of Feasibility and Effectiveness

Optional reading

[3] A survey on metric learning for feature vectors and structured data