Main activities:

- Producing/gathering experimental data to answer the various formulated RQs.

Contexte et atouts du poste

Most of today's software is highly-configurable in order to fit constraints, functional and performance requirements of users. For instance, the Linux kernel offers 15000 configurations options and this is certainly the key to its success: configurations allow users to embed Linux variants in a multitude of devices with different levels of performance, security, etc. Configurations, though beneficial and needed for many organizations, also come with challenges for developers: How to ensure that, throughout the continuous evolution of a project, all or at least a subset of configurations build well?

The goal of this thesis is to develop what we call incremental build of configurations. Given a base configuration, we want to modify it (through the re-setting of some options values) and then build it without starting from scratch. Similarly, we aim to build a given set of configurations without starting from scratch each time. The promise is to dramatically reduce the cost of building large configurations, a stressing topic when you think about the environmental and financial costs that companies and public organizations should have to bear.

**Society relies on software, but building software has an enormous cost: this project aims to mitigate this trend**

The usual compilation and build process works quite well when small modifications are made (e.g., modification of one source file), but building several configurations involve large modifications that span numerous source files. There are two extremes: (1) small modifications, with very low cost since the incremental compilation is fast (2) large modifications, with high cost since almost everything should be recompiled. In-between, we want to find a good trade off between diversity of the configurations and cost of compiling them.

Mission confiée

 Assignments:
- With the help of DiverSE team (including CR Djamel Khelladi, Prof. Mathieu Acher, Prof. Jean-Marc Jézéquel), the recruited person will be taken to investigate smart and efficient building of software variants on several software subjects, such as (but not only) Linux.
- The goal is to develop what we call incremental build of configurations. Given a base configuration, we want to modify it (through the re-setting of some options values) and then build it without starting from scratch. Similarly, we aim to build a given set of configurations without starting from scratch each time. The promise is to dramatically reduce the cost of building large configurations.

Bibliography:


Collaboration:

The recruited person will be in connection with other researchers, PhDs, Postdocs, interns who work on teh research axis of software variability.

Responsibilities:

- We want to explore the configuration space in a smart, efficient way. The person recruited is expected to be involved in designing experiments and analyzing results. The are at least four research questions:
  - RQ1: Is incremental compilation of configurations safe? (ie do we obtain the same exact binary than with a standard compilation?)
  - RQ2: What's the gain of applying incremental compilation? (gain: time needed to compile eg the Linux kernels)
  - RQ3: Can we explore a diverse set of configurations with incremental compilations?
  - RQ4: Is there a compilation strategy that reduces the cost of compilation without trading diversity?

Principal activités

Main activities:
- Producing/gathering experimental data to answer the various formulated RQs.
Several subject systems will be considered, with different languages, compilers, and build properties. The candidate will start with the configuration space of Linux. She/he will instrument incremental compilation on top of TuxML, a tool dedicated to the large scale build of configurations. The ultimate goal is to integrate our idea in mainstream testing infrastructure (e.g. KernelCI), for exploring further configurations at lower cost. She/he will then consider other systems, like JHipster or Chromium.

- The outcome of this research is to formulate the foundations of incremental build, invent new algorithms integrated into mainstream compilers and build systems, and assess the solution on widely used software projects.
- Communicate and disseminate results.

**Compétences**

Technical skills and level required: automated software engineering, advanced programming, Linux kernel

Languages: Python, C, Bash

Relational skills: ability to communicate and share results

Other valued appreciated: interests in software and open-source projects

**Avantages**

- Subsidized meals
- Partial reimbursement of public transport costs

**Rémunération**

Monthly gross salary amounting to 1982 euros for the first and second years and 2085 euros for the third year.