# 1 Introduction

Modern software systems usually ship with a variety of options to select from in order to tailor them to customer's needs. Although a system's configuration is primarily chosen to meet functional requirements, the selection of features, of course, has effects on non-functional properties. The selection of features can directly influence the source code of a software system (compile-time variability) or the execution path of a software system (dynamic or load-time variability). Moreover, non-functional properties (e.g., performance or memory utilization) depend on the functionality offered, the respective implementation, program load and the resulting execution; that is, the choice of features also indirectly shapes non-functional properties. While some effects on non-functional properties only depend on a single feature selected, effects can also depend on a combination of features (feature interaction). Recent research studied

# 2 State of The Art

## 2.1 Preliminary Work and Context

1. Performance Measurement (measure-based, model-based)
2. Performance Regression and Root Cause Detection
3. Evolution of Configurable Systems
4. Performance Prediction of Configurable Systems (Genetic algorithms, Performance Influence Models)

## 2.2 Research Motivation

Aside from evolution of code and architecture, to what extent can variability influence performance? Why important: requirements evolve and so does variability, a better understanding of variability changes in the presence of existing architecture/code may lead to best practices in how to plan architectures (extensible, modular) or to unveil sub-optimal techniques used, e.g., regarding cross-cutting concerns or scattering [**?**]

# 3 Research Method

## 3.1 Experiment Setup

**Revision history.** We mine the revision history of software systems from public repositories, e.g, GitHub or Sourceforge. To keep the implementation effort on a feasible level, we focus on systems configurable at load-time since re-compilation for each revision times a number of variants needed for building a performance influence model will likely exceed the scope of this thesis.

**Variability History.** In addition to the mined revision history, we require a history of changes in the variability model. We do not expect this information to be explicitly documented. Therefore, we will have retrieve it manually from second-hand information including commit messages and release notes. If run-time parameters are used, inspection of parameter validation in the source code and respective changes might indicate changes in the variability model.

**Detecting variability changes** We are primarily interested in the co-variance of changes in the variability model and the system performance. Therefore, we limit our search space to those revisions $R$ where variability was changed. Then, for each revision $r \in R$, we compare the performance of the predecessor $pre(r)$ to $r$. Consequently, we will need to build $2 \times |R|$ performance influence models per system studied, whereby $|R|$ denotes the number of changes in the variability model.

**Variability-related performance regression.** We will build performance influence models using our previously derived feature models and SPLConqueror. As the performance influence model comprises a linear combination of terms representing features and feature interactions, we will measure changes in performance by comparing those performance influence models term-wise.

### 3.2 System Corpus

- For performance measuring, we cannot use unit tests shipped with the system since tests evolve as the system itself does. That is, we need a reusable and feasible test load, e.g., sample files for compression tools or sample queries for database systems.
- Our choice comprises systems that are configurable at load-time (cf. 3.1). Moreover, performance needs to be feasible to measure in terms of execution time and system size.
- Possible candidate systems may be *gzip*[1], *snappy*[2], *SQLite*[3] ...

## 4 Discussion

## 5 Related Work

## 6 Conclusion

1

2

3