

## TOP-Forschungsprojekte 2017

### Pervolution: Performance-Evolution von hochkonfigurierbaren Software-Systemen

Professur: Intelligente Softwaresysteme  
Prof. Dr.-Ing. Norbert Siegmund  
Fakultät Medien

Laufzeit: 1. Juli 2017 bis 31. Juni 2020

Drittmittelgeber: DFG

Fördersumme: 289.602,00 Euro

#### Beschreibung:

Fast jedes komplexe Softwaresystem ist heutzutage konfigurierbar. Nutzer können Konfigurationsoptionen verwenden, um Systemvarianten auf ihre Anforderungen zuzuschneiden. Dabei ist insbesondere Performance eine Schlüsseleigenschaft. Oft ist jedoch nicht klar, welche Optionen welchen Einfluss auf Performance haben, da Interaktionen und der gewaltige Konfigurationsraum den Nutzer überfordern. Hinzu kommt, dass Systeme sich weiterentwickeln und weitere Optionen hinzugefügt oder verändert werden oder die Performance-Einflüsse von Optionen und deren Interaktionen sich unvorhersehbar über die Zeit ändern. So hat sich die Anzahl an Konfigurationsoptionen des Linux-Kerns (x86) zwischen den Jahren 1995 (3284 Optionen) und 2012 (6319 Optionen) nahezu verdoppelt. Der Apache Web Server vervierfachte gar die Anzahl an Optionen, von 150 in 1998 auf nahezu 600 in 2014. Das Bestimmen von Optionen, die kritisch für Performance sind, ist in solchen Szenarien nahezu unmöglich. Ohne geeignete Repräsentationen und Methoden für die Evolution der Performance-Einflüsse von Konfigurationsoptionen müssen Entwickler immer wieder von vorne anfangen, um Performance-Bugs oder Performance-optimale Konfigurationen zu finden und ein Verständnis des Systems aufzubauen.

Im Projekt Pervolution werden wir einen Ansatz entwickeln, der eine Performance-bewusste Evolution von komplexen, konfigurierbaren Systemen unterstützt, so dass Entwickler informierte Entscheidungen bezüglich der Konfigurierbarkeit und Evolution eines Systems und im Lichte sich ändernden Performance-Verhaltens treffen können. Dazu extrahieren wir aus der Analyse von Performance-Evolutionsmustern wahrscheinliche Performance-Änderungen sowie Prinzipien für Performance-Evolution an sich. Die Evolutionsmuster werden wiederum aus Performance-Einflussmodellen gewonnen, die die Performance von Optionen und Interaktionen über die Zeit darstellen. Technisch werden wir für jede Version eines Systems (Zeitdimension) verschiedene Systemvarianten (Konfigurationsdimension) analysieren. Hierdurch werden wir Performance-Verbesserungen als auch -Verschlechterungen durch den Vergleich der Einflüsse von Konfigurationsoptionen auf die Performance erkennen. Insbesondere stellen wir Beziehungen von Variabilität und Code-Änderungen zu den Einflüssen im Einflussmodell her und ermöglichen Nutzern diese Modelle zu studieren. Wir werden das Lernen von Performance-Einflussmodellen, welches derzeit nur auf Black-Box-Messungen basiert, um White-Box-Analysen erweitern und darüber hinaus das Sampling der Messungen durch eine neuartige Kombination aus statischen, variabilitätsgewahren

#### Kontakt:

Bauhaus-Universität Weimar  
Intelligente Softwaresysteme  
Prof. Dr.-Ing. Norbert Siegmund  
norbert.siegmund@uni-weimar.de

Besuchsadresse:  
Bauhausstraße 9a  
99423 Weimar  
03643 / 58 35 74

## TOP-Forschungsprojekte 2017

Analysen sowie Differenzanalysen auf Code-Ebene erweitern. Diese einzigartige Kombination von Techniken ermöglicht die exponentielle Komplexität des Problems beherrschbar zu machen und praktische Lösungen anzubieten. Pervolution ist einzigartig indem es beide Dimensionen der Komplexität, Variabilität und Zeit, explizit und gleichrangig betrachtet.

Almost every complex software system today is configurable. Configuration options allow users to tailor a system according to their requirements. A key non-functional requirement is performance. However, users and even domain experts often lack understanding which configuration options have an influence on performance and how and which combinations of options cause performance interactions. In addition, software systems evolve, introducing a further dimension of complexity. During evolution, developers add new functionality and need to understand which pieces of functionality—controlled by which configuration options—require maintenance, refactoring, and debugging. For example, the number of configuration options has nearly doubled in the Linux kernel (x86), starting from 3284 in release 12 to 6319 in release 32; the increase for the Apache Web server has been from 150 options in 1998 to nearly 600 options in 2014. Identifying the options that are critical for performance becomes infeasible in such scenarios. Without a proper representation of the evolution of the performance influence of configuration options, developers have to start from scratch again and again, to identify performance bugs, performance-optimal configurations, and to build an understanding of the performance behavior of the system.

In Pervolution, we will develop an approach to facilitate performance-aware evolution of complex, configurable systems, by tracking down evolutionary performance changes and by providing development guidelines based on extracted performance-evolution patterns. Our goal is to provide deep insights into the performance evolution of configuration options and their interactions, so that developers can reason about their decisions about the system's configurability during software evolution, in the light of the changed performance behavior. Technically, we will use performance-influence models that capture the influences of configuration options as well as of their interactions on performance, using a unique combination of machine-learning techniques, sampling heuristics, and experimental designs. For each version of the system (time dimension), we analyze performance influence models of different variants (configuration dimension). We will relate variability and code changes to the individual influences inside the model and allow users to reason about these models.

We will extend the prevailing black-box approach of learning performance-influence models by combining static, variability-aware code analyses and program differencing to spot performance-related changes across different system versions. This unique combination of techniques allows us to tame the exponential complexity of the problem and to provide practical solutions. A distinguishing feature of Pervolution is that it considers both dimensions of variability and time explicitly and equally.

Weitere Informationen:

<https://www.uni-weimar.de/en/media/chairs/computer-science-and-media/intelligent-software-systems/>

<http://www.infosun.fim.uni-passau.de/se/pervolution/>

### Kontakt:

Bauhaus-Universität Weimar  
Intelligente Softwaresysteme  
Prof. Dr.-Ing. Norbert Siegmund  
norbert.siegmund@uni-weimar.de

Besuchsadresse:  
Bauhausstraße 9a  
99423 Weimar  
03643 / 58 35 74