C | A | U Christian-Albrechts-Universität zu Kiel Technische Fakultät

Christian-Albrechts-Universität zu Kiel, 24098 Kiel

Institut für Informatik
Software Engineering

Leitung:
Prof. Dr. Wilhelm Hasselbring

Hausanschrift:
Christian-Albrechts-Platz 4, 24118 Kiel

Postanschrift: 24098 Kiel

http://se.informatik.uni-kiel.de

Paketanschrift:
Christian-Albrechts-Platz 4, 24118 Kiel

Professor Jan Trlifaj
Vice Dean
Research and International Affairs Department
Charles University
Ke Karlovu 2072/3
121 16 Praha 2

Bearbeiter/in, Zeichen

Mail, Telefon, Fax
hasselbring@email.uni-kiel.de
tel +49(0)431-880-4664 / 3734 (Sek.)
fax +49(0)431-880-7617

Datum
Kiel, 09.11.2018

Review on the habilitation thesis of Lubomír Bulej

Dear Professor Trlifaj,

Lubomír Bulej, Ph.D., has applied for the position of associate professor at the Faculty of Mathematics and Physics, Charles University. As part of the promotion process, I review the submitted habilitation thesis.

The habilitation thesis is entitled "Performance Awareness and Observability on Modern Platforms". The thesis presents selected results of research on performance evaluation, performance modeling, and dynamic program analysis.

Relevance of the research topic: Performance is an important aspect of software systems, but unlike correctness, performance is largely invisible to developers because it is difficult to observe and reason about it. The complexity and performance of software is not caused by its core algorithms and data structures, which a single person or a small team could understand completely. The complexity of large software-intensive systems is caused by scale and its architecture, and the behavior is a result of interactions among entities that number in hundreds and thousands. Consequently, performance must be designed into the system from the start, and controlled throughout its construction; not just as an afterthought. Performance of computer systems is difficult to predict and to evaluate, which also applies to impact of any code changes intended to improve performance. However, performance is decisive quality property of software systems.

Research approach: The general goal of the research presented in this thesis is to make it easier to understand performance aspects of modern execution platforms, to make software running on those platforms more observable, and to make performance visible to developers such that it can be managed. Specific topics include performance testing and performance awareness, performance aspects of modern platforms, construction of dynamic program analyses, and observability of modern managed platforms. In all these areas, the ultimate goal of Dr. Bulej is to provide software developers and operators with methods and means to better understand the behavior of a software system at runtime, and thus contribute to making well-founded decisions during software development and operation. The addressed topics are:

- With a background of fifteen years of research, Dr. Bulej's research evolved from the original concept of ***regression benchmarking*** to the more general concepts of performance *testing* and performance *awareness*. In contrast to functional tests for correctness, performance tests are potentially more difficult to construct and execute, and their results are much more difficult to evaluate automatically, because the test conditions are not evaluated by the test code. Instead, evaluating the outcome of performance tests requires analyzing the data collected during execution of the test workload. To avoid misleading results due to interference, the test workload needs to be run multiple times, necessitating probabilistic rather than deterministic evaluation. Yet, automating statistical analysis and hypothesis testing on such data is difficult, because the underlying data distributions are typically unknown, often long-tailed and/or multi-modal, with conditional variance coming from multiple sources. To address these challenges, Dr. Bulej relies on Stochastic Performance Logic, a mathematical formalism for expressing performance requirements either in absolute terms, or as relations between the performance of multiple methods.

- To understand performance-related aspects of **modern execution platforms**, Dr. Bulej investigates the impact of partial load on performance and energy efficiency of different virtualization solutions, the performance impact of garbage collection and the de-optimization on managed platforms, as well as the effects of workloads resulting from programs written in different languages targeting a single managed execution platform:
    - Dr. Bulej presents Showstopper, a tool that addresses some of these challenges and enables systematic experimental evaluation with varying partial utilization of shared resources. Given arbitrary concurrent workloads, Showstopper accurately controls the utilization of a selected resource (processor utilization) by controlling the intensity of the workloads. This removes the need for implementing workload generation harnesses with configurable workload intensity, and makes it possible to examine system behavior at a particular utilization directly. The study of the impact of CPU pinning on energy efficiency received a **best paper award**.
    - Dr. Bulej shows that, given an almost-complete information about application behavior in form of allocation traces with object sizes, lifetimes, and reference updates, the simplified garbage collection model can fairly accurately predict frequency of minor collections in garbage collection. The prediction quality for minor collections remains stable across workloads, and across inputs ranging from full traces to probabilistic distributions of object sizes and lifetimes. Dr. Bulej shows that the ability to model real-world garbage collection performance hinges on our ability to obtain accurate allocation profiles, and that we need better approaches to observing application behavior on managed platforms. This paper received another **best paper award**.
    - A dynamic compiler can produce machine code based on assumptions about program behavior, which may be checked at runtime. Dr. Bulej presents a study of de-optimization behavior of code compiled by the Graal dynamic compiler and the behavior of the VM runtime in response to the de-optimizations via several benchmark suites.
    - Workload characterization uses various metrics to capture various qualitative and quantitative aspects of a workload, which can be used to gain insight into performance-related behavior of the combination of a particular workload and a particular execution platform. Dr. Bulej introduces a new set of dynamic metrics that are sensitive to differences in the workloads resulting from bytecode produced by different JVM languages, along with an easy-to-use toolchain to collect the metrics on a standard JVM.
- **Dynamic program analysis** tools are programs that observe the behavior of another program while it is executing, and report on the properties of that particular execution. This provides developers and software engineers with insights into the dynamics and behavior of software systems at runtime, which allows them to better understand, debug, optimize, or refactor such systems. To address the challenges, Dr. Bulej developed DiSL, a domain-specific language and framework designed specifically for instrumentation-based dynamic program analysis. DiSL relieves dynamic program analysis tool developers from having to deal with too many low-level details associated with program instrumentation, allowing them to focus on the dynamic analysis itself. DiSL received **two awards**: It has been accepted by the Technology Council of the OW2 Consortium as an incubator project, and by the SPEC Research Group for their repository of peer-reviewed tools for quantitative system evaluation and analysis. Concerning dynamic program analysis, the following contributions are made:

- Dr. Bulej presents an approach based on **state-oriented decomposition**, which enables the succinct description of different dynamic analyses. The key idea is to decompose the high-level analysis' requirements in terms of structures holding the accumulated state of an analysis, and the semantics with which these structures evolve in response to consumed events. The provided FRANC framework for composition of dynamic analyses demonstrates the feasibility of the state-oriented decomposition approach via an event-based publish-subscribe system with reusable abstractions for data collection and aggregation.
- Dr. Bulej addresses the observability of programs executing on **modern managed platforms**. Dynamic program analysis critically depends on the ability to observe the execution of a program. He contributes ShadowVM, a dynamic program analysis framework, which adopts an event-based programming model and realizes dynamic program analysis as a distributed event-processing system.
- The ShadowVM approach with the analysis split between an event producing instrumentation co-located with application code, and an event consuming analysis code executing in a dedicated virtual machine is a good match for the specifics of the **Android** platform. Because Android applications usually execute in a resource-constrained environment, Dr. Bulej executes the analysis VM on a remote system.
- State-of-the-art **dynamic compilers** perform optimizations based on profiling information gathered during program execution, and many optimization result in machine code that does not even perform certain operations present at the level of bytecode. The ShadowVM avoids perturbation by isolating the analysis and the observed application from each other, thus eliminating sharing of library-internal state. An approach to avoiding perturbation of dynamic compilation is contributed by Dr. Bulej. Additional contributions comprise case studies and tools, including an object allocation and lifetime profiler, a call-site polymorphism profiler, and a compiler-testing framework, which received a **distinguished paper award**.

Evaluation: The habilitation thesis consist of eight papers, all have been accepted and published at major peer-reviewed conferences and journals. Some conference papers received best paper awards. Besides papers, Dr. Bulej publishes open source software systems, some of which received awards. In summary, the thesis contributes new knowledge, methods and software for performance evaluation, performance modeling, and dynamic program analysis. This is a significant, original research achievement, which I highly appreciate.

Recommendation: Based on my review, I recommend appointing Dr. Bulej as an associate professor.

Please let me know if you require more information.

Yours sincerely