

Master Thesis / Project in Software Engineering

Graal Adaptive Compilation Framework

Student:

SKZ/Matr.Nr.: -- / -----

Email:

Advisor: Dipl.-Ing. David Leopoldseder

Start date: xx.xx.xxxx

Dipl.-Ing. David Leopoldseder
Institute for System Software

T +43 732 2468 4356
F +43 732 2468 4345
david.leopoldseder@jku.at

Secretary:
Birgit Kranzl
Ext 4341
birgit.kranzl@jku.at

Graal [1] is a just-in-time compiler for the JVM platform that is itself written in Java. Due to the modular design of Graal the addition of novel optimizations to the existing compiler is easy and straightforward. However, tuning such a compiler optimization for a non-trivial use case is a complex and error prone task typically done manually by experienced engineers.

Adaptive compilation [1, 2] is an approach that allows to optimize compiler parameters for a given workload by evaluating a final cost function on consecutive executions of an experiment.

The goal of this thesis is to implement an optimization test framework, that allows one to easily test existing and novel optimizations, e.g., compiler optimization phases, on a set of micro benchmarks. The test framework should allow an optimization to test the global impact of a local transformation decision by executing the experiment under test multiple times and evaluating the cost function on it. This way, the impact of a transformation can be simulated by executing it. This gives insight on the impact of a single, local, transformation within a global optimization pass. In contrast to adaptive compilation, which requires multiple, consecutive, executions of an experiment an online system capturing similar information is a desired research goal.

The goals of this thesis are:

- Implementation of a limited form of adaptive compilation in Graal that allows one to finish a given compilation, evaluating a cost function and returning the result.
- Design of a small set (5-10) of micro-benchmarks that are pure in a sense that they can be re-executed without global side effects. Those benchmarks will be used to simulate the impact of a transformation (dynamic cost function).
- Design of a small set of dynamic and static cost functions for the adaptive compilation using existing cost models of the Graal compiler.
- Evaluation of the test framework, using an existing optimization such as loop unrolling to test the capabilities of the adaptive compilation.

Modalities

The progress of the project should be discussed at least every two weeks with the advisor. A time schedule and a milestone plan must be set up within the first 3 weeks. It should be continuously refined and monitored to make sure that the thesis will be completed in time. The final version of the thesis must be submitted not later than xx.xx.xxxx.

[1] <http://openjdk.java.net/projects/graal>

[2] Cooper, Keith D., Devika Subramanian, and Linda Torczon. "Adaptive optimizing compilers for the 21st century." *The Journal of Supercomputing* 23.1 (2001): 7-22.

[3] Triantafyllis, Spyridon, et al. "Compiler optimization-space exploration." *Code Generation and Optimization, 2003. CGO 2003. International Symposium on*. IEEE, 2003.