Implementation of the Graal Baseline Compiler

Master thesis for ...
Matr.Nr.: ...
E-Mail: ...

Graal [1] is an effort to create a new just-in-time compiler for Java that is itself written in Java. It is based on a port of the HotSpot client compiler from C++ to Java.

Graal uses different internal representations of the program during compilation. The high-level representation (Graal IR) is graph-based, highly flexible and well suited for optimization and transformation. Eventually, Graal translates the high-level IR into a low-level representation (LIR), used for register allocation and code generation.

Building the Graal IR graph from Java Bytecode is costly and so is the translation to the low-level IR. The goal of the Graal Baseline Compiler is to generate the low-level IR directly from Bytecode without the detour of building the high-level graph.

The scope of this thesis is as follows:
- Write a Baseline Compiler for Graal.
- Apply test-driven development techniques.
- Continuously integrate your changes into upstream development.

Optional goals are:
- Test the implementation on a variety of non-trivial Java programs.
- Support parse-time optimizations (such as inlining, null-check elimination, etc.).
- Add instrumentation support to collect profiling information.

The work’s progress should be discussed with the supervisor at least every 2 weeks. Please note the guidelines of the Institute for System Software when preparing the written thesis.

Supervisor: Dipl.-Ing. Josef Eisl, Dr. Lukas Stadler