



JavaScript Backend for Graal

Master thesis for David Leopoldseder

Email: david.leo@web.de

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.

Substrate VM compiles Java Bytecode with the Graal Compiler ahead-of-time to native machine code. This project should extend the Substrate VM with a JavaScript backend.

The scope of this project is as follows:

- Evaluate the support Substrate provides for JavaScript.
- Determine which parts of the Substrate's amd64 backend can be reused to support code generation for high level languages.
- Inspect the Support of Graal IR for structured control flow code generation.
- Create the SSA based code generator that will generate high-level instructions from the high-level intermediate representation. Control flow needs to be reconstructed from Graal IR where possible, with a fallback mechanism for unstructured control flow.
- Add tests for the backend covering the Java ByteCode Specification.
- Evaluate and implement performance optimizations specific to JavaScript.
- Document the problems that were encountered and the changes in Substrate that were required to overcome them.
- Evaluate the performance of the JavaScript backend with standard Java benchmarks.

Explicit non-goals are:

- Code Quality of the generated JavaScript Code.
- Completeness with respect to the Java bytecode specification.

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: Dr. Lukas Stadler, Dipl. Ing. Matthias Grimmer

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