JOHANNES KEPLER
UNIVERSITÄT LINZ

o.Univ.-Prof. Dr.
Hanspeter Mössenböck
Institute for System Software

T +43 732 2468 4340
F +43 732 2468 4345
hanspeter.moessenboeck@jku.at

Secretary:
Karin Gusenbauer
Ext 4342
karin.gusenbauer@jku.at

Master's Thesis

On-the-fly Analysis of Just-in-time Compiled Code to Identify Anti-Patterns

Student:	Simon Schuster
Supervisors:	Prof. Hanspeter Mössenböck DI Sebastian Kloibhofer
Supervisor (Dynatrace):	Dr. Philipp Lengauer
Start:	1. Oktober 2020

Dynatrace's *OneAgent* can inject into and monitor a wide variety of technologies. In the case of Java, it does so by attaching itself to the Java Virtual Machine Tool Interface (JVMTI) and instrumenting application code. The ultimate goal is to monitor requests, identify the root cause for slow requests and other bottle necks.

One of these bottlenecks could be caused by badly written code, impeding the just-in-time compiler and preventing aggressive optimizations. By means of the JVMTI, the *OneAgent* should listen to the *CompiledMethodLoad* hook that provides access to the newly compiled code. Using the compiled code, one can identify inlined methods, safepoints, and other optimizations.

The goal of this thesis is to evaluate the feasibility (by building a prototype) of extending the *OneAgent* to automatically search for missing optimizations, e.g., methods that have not be inlined, allocations that have not been removed or badly placed safepoints. This information must then be used to analyze the original method and to identify whether the method contains something that impedes the optimization in question.

As a limitation, the implementation only has to support Java 11 and Hotspot-based JVMs on x86 64bit. Presenting the results in the Dynatrace UI is not in the scope of this thesis.

Finally, both the performance as well as the functionality must be evaluated extensively. Although the prototype is restricted to a very limited set of JVMs, the thesis must include a brief evaluation of whether the prototype is also expected to work on other JVMs as well as a brief evaluation if similar concepts could be applied to other JVMs.

The progress of the thesis should be discussed on regular basis with the supervisors. A time schedule with milestones must be presented 3 weeks after the work starts. This schedule will be continuously refined as the work progresses. The final thesis should be submitted not later than September 30, 2021.

JOHANNES KEPLER
UNIVERSITÄT LINZ
Altenberger Straße 69
4040 Linz, Österreich
www.jku.at
DVR 0093696