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

A new Mark-and-Copy GC for GraalVM Native Image

Student:	Christian Aistleitner
Advisor:	Prof. Hanspeter Mössenböck
Co-Advisor:	Christian Wirth, Peter Hofer, Christian Häubl
Begin:	February 1, 2023

GraalVM [1] Native Image [2] provides ahead-of-time compilation of Java code to optimized native binaries that run without a JDK. The garbage collector (GC) is written in Java too and, like the rest of the Java runtime system, part of the AOT compiled binary. The current GC algorithm is quite primitive: It is a simple generational stop-and-copy algorithm. Objects are allocated in the young generation. An incremental collection promotes live objects of the young generation directly into the old generation. This can cause short-lived objects that accidentally survive, to be promoted to the old generation. Promotion of short-lived objects to the old generation causes more frequent old generation collections and leads to larger GC pauses. For a full collection, every reachable object is copied, increasing the memory footprint during GC.

The goal of this project is to implement a new GC that provides high throughput and minimized memory footprint. The target applications of this GC are microservices and function-as-a-service frameworks that are mostly single threaded and need to run in, e.g., a Docker container with a very constrained memory size.

The basis of the new GC should follow established best practices: a mark-and-compact algorithm for the old generation, and a stop-and-copy algorithm with aging for the young generation. After the basic algorithm is implemented, we can research novel improvements that are tailor the algorithm to the specific use case. For example, we can exploit that the heap is not a contiguous memory region but separated into chunks. In the mark-and-compact algorithm we can only compact regions that have a significant number of dead objects.

The work is an evolutionary step that can build on existing code in old and new generation.

The scope of this thesis is as follows:

- Implement the new garbage collection algorithm, replacing the existing implementation.
- Test the approach on all available test suites and ensure full compatibility.
- Benchmark the new approach on our existing benchmark suites.
- Optionally, work on advanced features and provide new tests and benchmarks for them.
- Contribute the approach to the open-source repository [2] (requires signing OCA [3]).

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. The deadline for the written thesis is January 31, 2024.

References:

[1] https://www.github.com/oracle/graal
[2] https://github.com/oracle/graal/tree/master/substratevm
[3] https://oca.opensource.oracle.com/

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