



**JOHANNES KEPLER  
UNIVERSITY LINZ**

Master's Thesis

**Pointer Tracing Support for Hotspot's G1 Garbage Collector in AntTracks**

Student: Christian Niederer, BSc.

SKZ/Matr.Nr.: 01355042

Email: [chris.niederer@outlook.com](mailto:chris.niederer@outlook.com)

Advisor: Dipl.-Ing. Markus Weninger

Start date: Sept 2017

**Dipl.-Ing. Markus Weninger**  
Institute for System Software

P +43-732-2468-4361

F +43-732-2468-4345

[markus.weninger@jku.at](mailto:markus.weninger@jku.at)

---

AntTracks comprises a modified Java VM based on the Hotspot VM, i.e., AntTracks VM, and an offline post-processing analysis tool.

The VM's aim is to allow tracking of an application's entire life cycle by writing information about certain events to a trace file. This events include object allocations, object movements by the garbage collector, pointers between the objects and so on.

Such an event trace can then be analyzed in the offline post-processing tool. Based on the information parsed from the trace file the tool is able to reconstruct the heap for any garbage collection point.

Currently, while trace files without pointer information can be created using various garbage collectors, tracing pointer information using AntTracks VM is yet only supported by the ParallelOld garbage collector. The goal of this thesis is to extend the AntTracks VM to also support the collection of pointer information when using the G1 garbage collector. Consequently, the GC's implementation must be extended to log the according events to the trace file, so that the tool can reconstruct the heap including pointer information based on the trace file. To ensure low runtime overhead, once a stable pointer tracing mechanism exists, the student has to focus on performance improvements such as reducing the number of logged events by detecting and ignoring redundant information. The post-processing tool currently offers basic support for parsing pointer information, yet without any focus on performance and memory overhead. Therefore, also the parsing and storage of pointer information in the tool should be optimized.

If changes to the trace event format seem necessary to the student, these changes have to be discussed with the supervisor beforehand.

The thesis must include both: A functional evaluation based on well-known benchmarks showing that the tracing and parsing works, as well as an quantitative analysis showing the tracing and parsing overhead (in regard to memory and run time).

The final version of the written thesis must be submitted not later than 01.09.2018.

**JOHANNES KEPLER  
UNIVERSITY LINZ**

Altenberger Str. 69

4040 Linz, Austria

[www.jku.at](http://www.jku.at)

DVR 0093696