



**JOHANNES KEPLER
UNIVERSITY LINZ**

Bachelor's Thesis

Utilizing Database-based Storage to Reduce Memory Overhead in AntTracks

Student: Felix Batusic

SKZ/Matr.Nr.: 1455064

Email: felix@batusic.net

Advisor: Dipl.-Ing. Markus Weninger, BSc.

Start date: Jan 2018

Dipl.-Ing. Markus Weninger
Institute for System Software

P +43-732-2468-4361

F +43-732-2468-4345

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.

A potential risk of memory analysis tools, especially when analyzing heap states of applications that ran out of memory on the same machine, is to also run out of memory during analysis. For each heap object, information such as type, allocation site, references, etc. has to be reconstructed and stored in memory. AntTracks currently receives extended support for object graph analysis, which requires us to store additional information about the references between objects (e.g., dominator tree relation).

The goal of this thesis is to refactor AntTracks's current data structure that represents a heap state (i.e., the heap objects live at a certain point in time in the traced application). Currently, the data structures stores the data completely in memory, mostly using arrays of Java basic data types (e.g., *int[]*).

Instead of in-memory arrays, the information about the heap objects (e.g., type, allocation site, pointers, ...) reconstructed for a certain point in time should be stored in a database and queried from there on demand. Since the DB can decide when to store data to secondary storage, e.g., the HDD, we assume this approach to reduce AntTracks's memory overhead.

It of the student's tasks is to analyze which type of database (e.g., relational DB, NOSQL DB, graph-based DB, ...) suits the problem best.

A potential risk when using this technique is performance degradation, since data may have to be transferred between AntTracks Analyzer and the database as well as between RAM and the secondary data storage, and vice-versa. Therefore, the thesis should also include an evaluation on how utilizing database-based storage influences AntTracks's memory utilization and heap state analysis performance.

The final version of the written thesis must be submitted not later than XX.YY.ZZZZ.

**JOHANNES KEPLER
UNIVERSITY LINZ**

Altenberger Str. 69

4040 Linz, Austria

www.jku.at

DVR 0093696