

Master's Thesis

Unification and Automation of Time-Based Heap Analysis Methods

Student: Elias Gander

SKZ/Matr.Nr.: 921 / 01456336

Email: elias.gander@jku.at

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

Start date: 15 October 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, i.e., AntTracks Analyzer.

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, AntTracks supports various types of time-based heap analysis methods, i.e., algorithms that perform analyses over a period of time. For example, AntTracks can visualize object groups (e.g., objects of the same type that have been allocated at the same allocation site) that accumulate over time, inspect data structures that grow over time, and similar methods.

The time window in which these methods operate has to be manually chosen by the user.

Currently, all these methods are separated from each other. If multiple time-based analyses should be performed, the trace has to be parsed multiple times, leading to redundant calculations and redundant data, thus imposing unnecessary overhead in terms of computation time and memory.

The goal of this thesis is to create a universal time-based analysis feature that combines all existing methods and allows AntTracks developers to easily add new ones. A single, modifiable parser for time-based analyses should be developed, replacing AntTracks's task-specific parsers. It should be possible to collect the information that is currently collected by the different parsers in multiple passes during a single parsing pass. The information that is currently stored in different locations should be unified, checked for redundant information and made accessible through a common interface.

Using the new parser and its combined information, a new analysis feature should be added. AntTracks's current time-bases analyses focus on objects that survived or have been born during a given time window. This analysis should detect hotspots in the monitored application that generate a lot of short-lived temporary objects. High-frequent allocations of short-living objects can lead to unnecessary GC overhead and should be circumvented if possible.

Another task of this thesis is to automate the selection of time windows for analysis. Currently, the user has to select a time window he/she thinks may be of interest. Using heuristics, e.g., looking for a monotone increasing object count and keeping track of local minima, AntTracks should be able to suggest time windows of interest to the user.

The last mandatory point of this work is a feature to display a human-readable summary of the time-based analysis. The different methods should be able to report objects or objects groups of interest that could then be analyzed in more detail. For example, ever-growing data structures could be reported.

The final version of the written thesis must be submitted not later than 15.10.2019.