JOHANNES KEPLER
UNIVERSITY LINZ

Dipl.-Ing. Markus Weninger
Institute for System Software

P +43-732-2468-4361
F +43-732-2468-4345
markus.weninger@jku.at

Master's Thesis
Memory Leak Investigation using Graph-based Visualization

Student: Alexander Lang
SKZ/Matr.Nr.: 921 / 01255360
Email: al.lang@gmx.at
Advisor: Dipl.-Ing. Markus Weninger, BSc.
Start date: 01 October 2018

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.

A reconstructed heap in AntTracks contains various information about every object that has been live at the given point in time: Address, type, allocation site, the address of all referenced objects, and so on. Furthermore, AntTracks can detect data structures in such reconstructed heaps. To analyze a heap state, the objects can be arbitrarily grouped using AntTracks's classification system. For example, all objects could be first grouped by their types, and then grouped by their allocation sites. Such a multi-level grouping results in a hierarchical classification tree.
AntTracks uses this classification system throughout its various analyses and visualizes the resulting classification trees in a TreeTableView control. While this TreeTableView control is well suited to visualize these hierarchical trees, it cannot be used to visualize arbitrary graphs.

The goal of this thesis is to integrate a visualization system into AntTracks that allows to visualize arbitrary graphs. Graphs are a reoccurring pattern in memory monitoring data: objects that reference each other, data structures that contain other objects, just to name a few.
The visualization system should not be restricted for a certain use case but should be usable for arbitrary graphs. Nodes in the graphs should have customizable properties such as text, color and shape. Every node may have internal nodes, i.e., objects that could be displayed inside that node ("nested graph structures"), and successor nodes. It should be possible to label edges. The graph should not always be fully shown, but the user should be able to expand / collapse each node. Memory efficiency is of high importance, thus only visible parts of the chart should exist as objects in memory. Algorithms that automatically expand and highlight certain nodes (such as "Show paths to GC roots") should also be included, alongside other general features such as panning and zooming.
AntTracks is developed in JavaFX, but unfortunately, no free production-ready graph library exists for it at the moment. Thus, the project should be implemented in JavaScript, using well-known JavaScript frameworks such as vue.js or d3.js. The thesis should contain an evaluation on the different tools and approaches that could be used for (1) communication between JavaScript and Java, (2) visualization, and (3) layouting.
The thesis should also contain an evaluation of the approach that presents its applicability to investigate the root cause of a memory leak.

The final version of the written thesis must be submitted not later than 01.10.2019.