JOHANNES KEPLER
UNIVERSITY LINZ

Master's Thesis

Analyzing Collection Staleness and Collection Anti-Patterns that Lead to Memory Problems

Dipl.-Ing. Dr. Markus Weninger, Bsc
Institute for System Software
T +43-732-2468-4361
markus.weninger@jku.at

Student: Markus Hirth
Advisor: Dipl.-Ing. Dr. Markus Weninger, BSc
Start date: October 2022

Modern programming languages such as Java are so-called *managed languages* that leverage garbage collection to free the programmer from the error-prone task of manually managing memory. While garbage collection prevents certain memory problems, anomalies such as memory leaks can still happen. Such anomalies are often connected to the inproper use of collections such as lists or maps.
For example, Xu and Rountev presented “Precise Memory Leak Detection for Java Software Using Container Profiling” (https://doi.org/10.1145/2491509.2491511, https://doi.org/10.1145/1368088.1368110) in which they present a technique on how to track accesses to containers, how to calculate the staleness of the container’s elements (i.e., when each element was accessed the last time) and how this information, together with information about the collection’s overall size and growth, can be used to calculate a leaking confidence factor. This leaking confidence factor tells how probable it is that a collection exhibits a memory growth problem.

In previous work (“Collecting Memory Monitoring Data using Aspect-oriented Programming” by Markus Hirth), we have shown that it is possible to collect information about collections (creation, additions, removes, accesses, size) in Java using aspect-oriented programming, opposed to the established way of using (native) agents and bytecode manipulation. This information is written to a trace file while the monitored application is running.

The goal of this master’s thesis is to utilize such trace files to analyze the staleness of collections and to inspect general anti-patterns that can lead to memory problems. As a first step, the student has to develop a parser that can process the trace files. Secondly, inspired by the work by Xu and Rountev, the student should develop data structures that allow us to reconstruct staleness information similar to the one presented in their papers – this should allow us to also calculate a leaking confidence factor. The student might also have to improve the trace collection with new event types and information for this. Once the subgoal of replicating the leaking confidence factor metric is achieved, the trace collection, the parser and the data structures should be extended to detect other memory anti-patterns. The student might find inspiration in “Patterns of Memory Inefficiency” by Chis et. al (https://doi.org/10.1007/978-3-642-22655-7_18) regarding possible patterns that might be detected using the trace data.
The written thesis should also contain an evaluation of the implemented approach, showing how the resulting tool can be used to detect memory problems in applications and how the tool helps users fixing these problems.

Modalities:
The progress of the project should be discussed at least every four weeks with the advisor. A time schedule and a milestone plan must be set up within the first 3 weeks and discussed with the advisor and the supervisor. It should be continuously refined and monitored to make sure that the thesis will be completed in time. The final version of the thesis must be submitted not later than 30.09.2023.