



**Dipl.-Ing. Philipp Lengauer**  
Institut für Systemsoftware

Tel.: +43 732 2468-4362  
Fax: +43 732 2468-7138  
philipp.lengauer@jku.at

## Master's Thesis

### **Refinements and Extensions for Ant Tracks to support Multiple Garbage Collectors**

Lin, 10. Oktober 2014

Managed memory, i.e., garbage collected memory, has gained wide-spread use because it relieves the programmer from freeing heap objects manually and allows for fast allocations. However, these advantages come at the cost of a difficult to understand memory manager and garbage collector. Therefore, searching for memory-related performance degradations is a tedious task because the reasons for slow allocations, a large garbage collection pause, or a high garbage collection frequency might not be obvious.

In related work (see *Accurate and Efficient Object Tracing for Java Applications*, Lengauer et al., ICPE'15 (submitted for review), and *Post-Mortem Memory Profiling for Java Applications*, Bitto, Master Thesis JKU) a rigorous tracing mechanism is built into the Hotspot Java Virtual Machine to track all memory allocations and GC operations. The generated trace can then be used to reproduce and analyze the heap layout offline. A major goal of this mechanism was to keep the run time overhead as low as possible. However, it has still some performance bottlenecks and is only implemented for a single garbage collector whereas the JVM supports several others for different usage scenarios.

The goal of this thesis is to extend the tracing mechanism to support all garbage collectors (Concurrent Mark & Sweep, G1, Mark & Sweep, Parallel New, Serial with special focus on the first two). This also includes defining new efficient event types suited for a specific GC as well as to extend the offline analysis to handle the new events. The performance overhead must be kept as low as possible (at least comparable to the already implemented GC). Furthermore, binary on-the-fly compression of the trace must be investigated to reduce the trace size and potentially reduce the run time overhead due to less IO activity.

Supervisor: Dipl.-Ing. Philipp Lengauer

Student: Johannes Pichler