To Infinity and Beyond: High Performance Garbage Collection Combining Reference
Counting and Immix

Rifat Shahriyar
Australian National University
rifat.shahriyar@anu.edu.au

Reference counting and tracing are the two fundamental approaches that have underpinned garbage collection since 1960. However, despite some compelling advantages, reference counting is almost completely ignored in implementations of high performance systems today. We take a detailed look at reference counting to understand its behavior and to improve its performance. We identify key design choices for reference counting and analyze how the behavior of a wide range of benchmarks might affect design decisions. We use insights gleaned from this analysis to introduce a number of optimizations that significantly improve the performance of reference counting. We find that an existing modern implementation of reference counting has an average 30% overhead compared to tracing, and that in combination, our optimizations are able to completely eliminate that overhead. This brings the performance of reference counting on par with that of a well-tuned mark-sweep collector.

Both mark-sweep and reference counting generally use very well tuned free list implementation. We compared our improved reference counting with Immix. Immix is a mark-region based tracing garbage collector that performs better than mark-sweep. Much of its performance advantage over mark-sweep is due to its heap organization. Our improved reference counting is 3% slower than Immix and 10% slower than sticky Immix. Sticky Immix should therefore be a good indicator of the performance of our improved reference counting projected onto the Immix heap organization. This is an exciting prospect because Sticky Immix is only 3% slower than Jikes RVM's production collector.

This work focuses on mapping our improved reference counting in unique heap organization of Immix to achieve a higher performing garbage collector. It opens up an opportunity to achieve a production level garbage collector using reference counting.