Profiling Sharing Opportunities in Java

Michael J. Steindorfer  Jurgen J. Vinju

Goal: Predict potential benefits of “maximal sharing” optimization (runtime and memory) a priori.

Maximal sharing a.k.a. hash-consing enforces: x.equals(y) ↔ x == y

Problem:
- Maximal sharing is a cross-cutting concern and engineering intensive, which makes experiments expensive.
- Effectiveness of maximal sharing is dependent on redundancy present in programs and the amount of deep equality comparisons.

Solution:
- Object redundancy profiling with fingerprinting along program run.
- Calculating object lifetime overlaps to predict memory gains.
- Validation of prediction accuracy by comparing against a maximal sharing implementation.
- Predicting impact on equality checks and reference comparisons by simulating maximal sharing hash table data structures.

Conclusion: We accurately predict the impact of maximal sharing on memory and equality comparisons a priori.