



The Performance Analysis of Garbage Collection on Two Architectures

Yong Uk Lee
yonguks@gist.ac.kr
Programming Systems Laboratory

1. Introduction

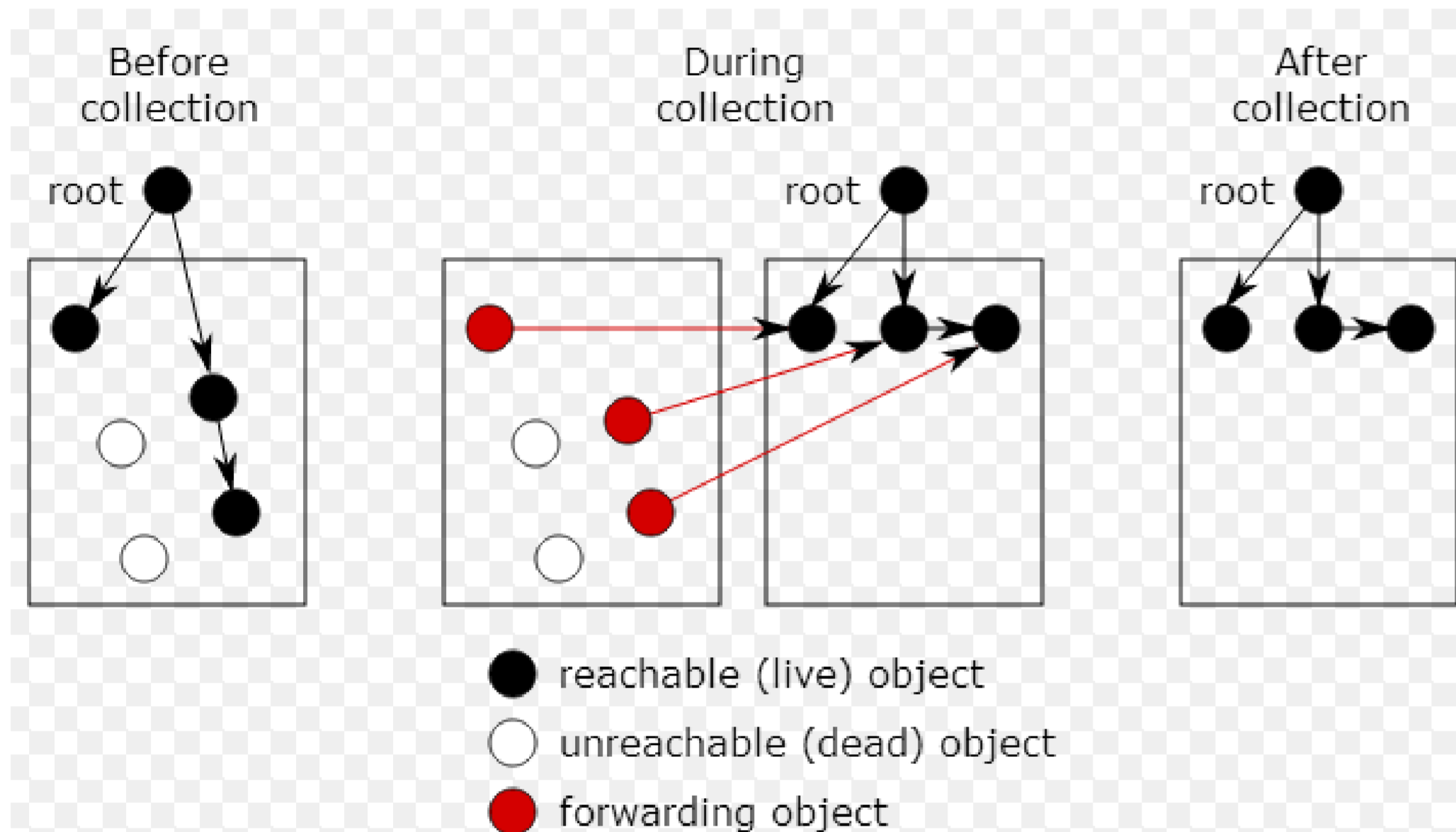
In computer science, garbage collection (GC) is a form of automatic memory management. The garbage collector, referred also to collector, attempts to reclaim garbage or memory occupied by objects that are no longer in use by the program.

Garbage collection was first proposed by John McCarthy around 1959. Garbage collectors for languages like ML and Java, which exert tight control over how applications create and use pointers, can maintain an exact representation of the reachability graph, and thus can reclaim all garbage.

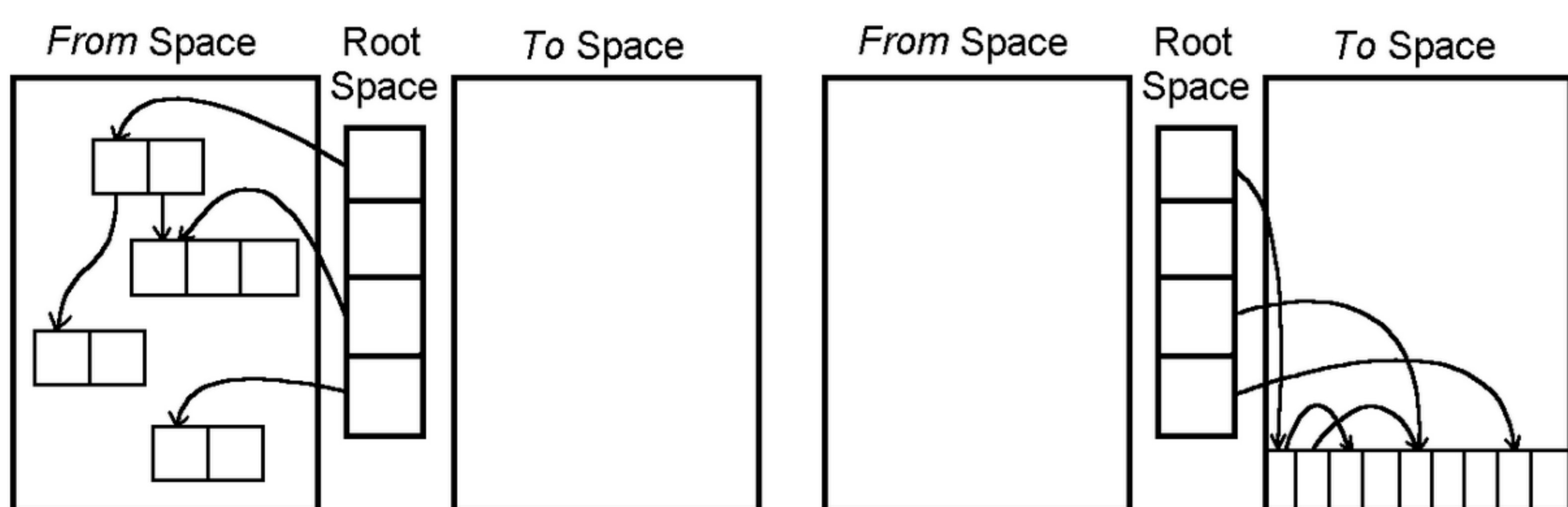
We explore and quantify garbage collection behavior for three whole heap collectors (MarkSweep, SemiSpace, RefCount) and generational counterparts (GenMS, GenCopy, GenRC). In order to analyze the performance of garbage collectors, we carried out an experiment of garbage collection using DaCapo-9.12-bash benchmark on two architectures which are AMD Opteron and Intel core i3. As a result, it is implied that the way they match program characteristics depending on the resource of hardware and the scale of programs.

2. Collectors

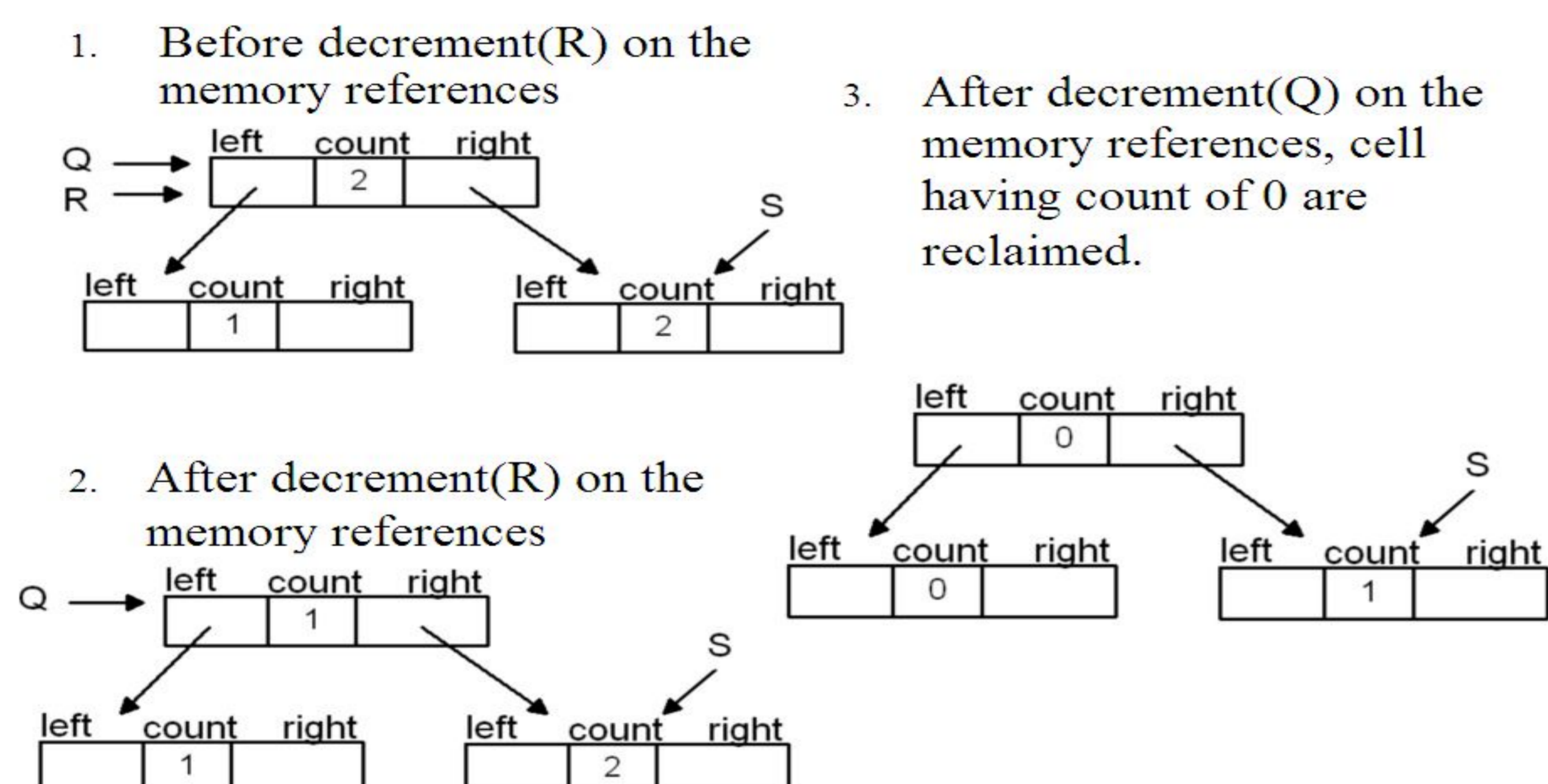
MarkSweep



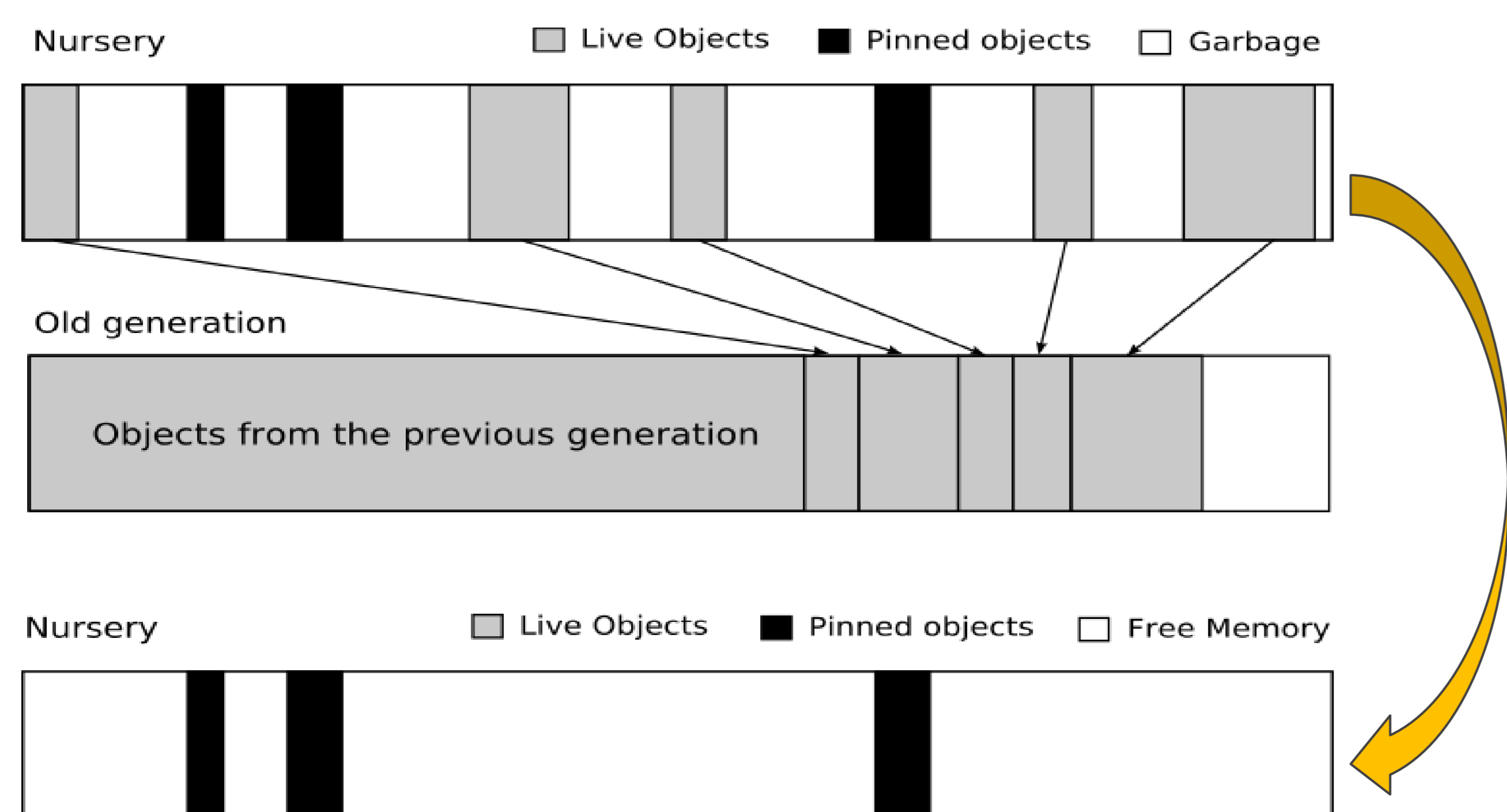
SemiSpace



Reference Counting

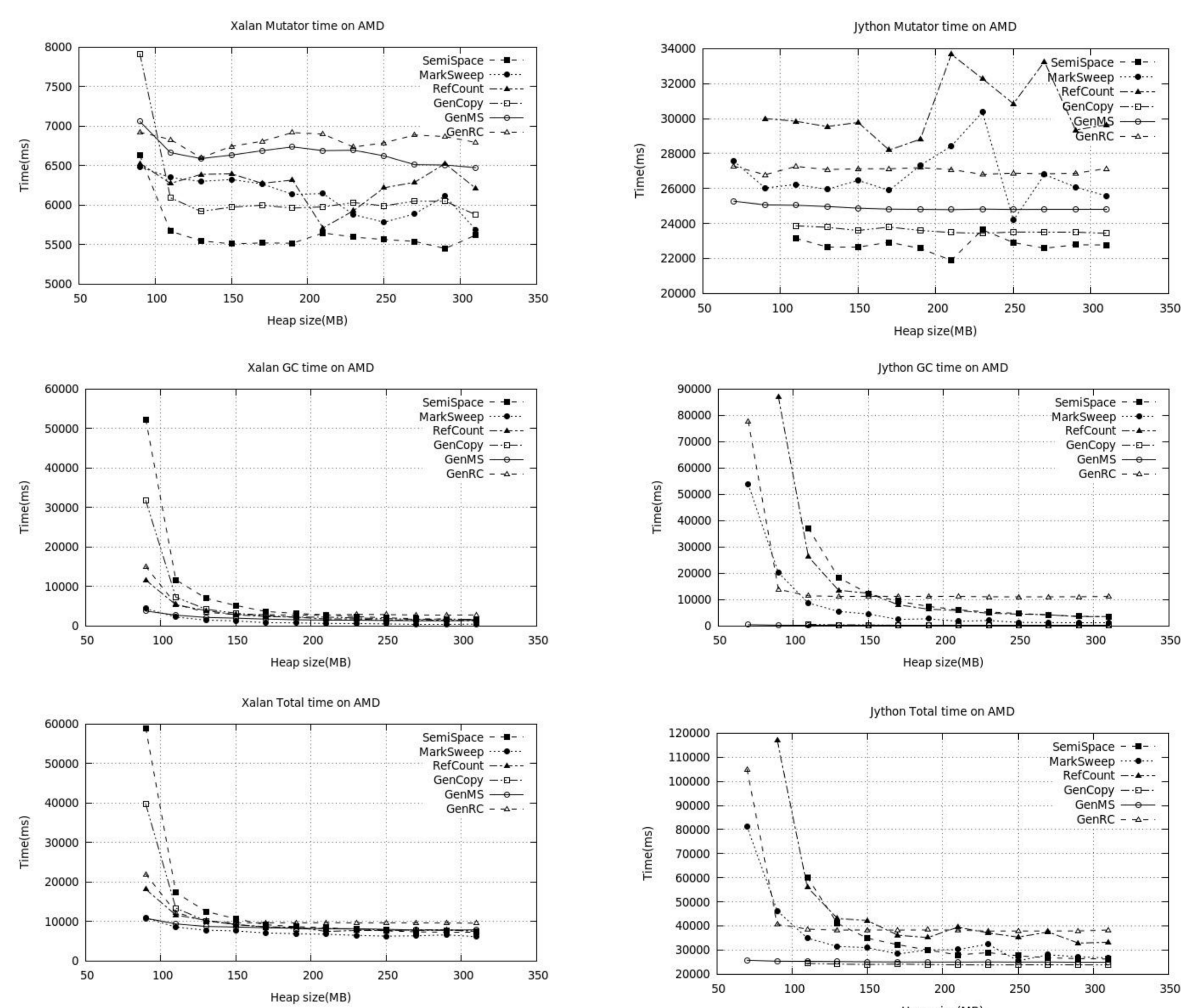


Generational collector



3. Result

AMD Opteron



Intel core i3

