Partitioning and generational GC

Richard Jones, Antony Hosking and Eliot Moss

2012

Presented by Hadar Getzel

25-11-14

My chapters dealt with motivation for partitioning the heap, generational garbage collectors and other partitioned schemes.

In my lecture we have discussed several reasons and criterions for partitioning the heap such as partitioning by mobility, partitioning by size and partitioning for space. We discussed the benefits of using partitioning for garbage collection.

Next we discussed the generational garbage collection, including the main idea of generational GC (the generational hypothesis), age recording schemes, the beltway framework (and GC) and the abstract generational GC algorithm.

Finally we have discussed other partitioned schemes, presented the Treadmill GC for large objects spaces and the hybrid mark-sweep copying collector.

Contribution (things that weren't present in the book):

Slide 43: a video that demonstrates the generational garbage collection in action.

Slide 62: an example for executing the abstract generational garbage collection algorithm.

Discussion: several issues were brought up during the talk:

- The effectiveness of generational GC depends strongly on the lifetime demographics of the application program. The cost of more frequent collections of the nursery and of write barriers must be amortized by obtaining a much better than average pay-back from collecting young generations.
- If the overwhelming majority of objects do not die young — then generational collection will not be an appropriate solution.
- Generational collection improves only expected pauses times; eventually the collector must collect the full heap and generational collection on its own cannot solve the problem of the worst-case pause time, which may be excessive for large heaps.
- Partitioning for multi-thread purposes – allowing each thread to manage by a generational garbage collection scheme it's on heaplet, and another area detected for handling shared objects.