情報処理学会論文誌　プログラミング　Vol. 3　No. 4　61 (Sep. 2010)

発表概要

世代別 Mostly-Copying GC の Ruby VM への実装と評価

永原　治[†1]　鵜川始陽[†2]　岩崎英哉[†2]

近年，Ruby による大規模アプリケーションの開発が行われるようになってきている．大規模なアプリケーションの特徴として，大量のデータを扱い長時間動作し続けるという特徴があげられる．既存の Ruby 処理系は保守的マークスイープ方式でごみ集め（GC）をしているが，このような特徴を持つアプリケーションでは，世代別コピー GC で GC の時間を短縮できることが知られている．しかし，Ruby 処理系には保守的 GC を前提に作られている部分があり，移動できないオブジェクトが多くある．そのため，通常のコピー GC は実現できない．そこで，移動できないオブジェクトが存在しても，コピー GC を実現できる Mostly-Copying GC を用いることにした．Mostly-Copying GC では，ヒープを固定長のブロックに分割する．我々の GC では，ブロックごとに世代を持たせる．マイナ GC では，新世代にある移動できるオブジェクトを旧世代に移動させるだけでなく，生存オブジェクトの多いブロックは，ブロックごと旧世代に含める．これにより移動できないオブジェクトも旧世代に移すことができる．

Implementation and Evaluation of Generational Mostly-copying GC in Ruby VM

Osamu Nagahara,[†1] Tomoharu Ugawa[†2] and Hideya Iwasaki[†2]

Recently, large-scale applications have come to be developed in Ruby. A large-scale application tends to treat a large amount of data and to keep working for a long time. While the Ruby VM has the conservative mark-sweep GC, it is known that the time spent on GC in such an application can be reduced by using the generational copying GC. However, we cannot implement the copying GC in the Ruby VM, because part of the VM depends on the conservative GC, and many objects cannot be moved. Therefore we decided to use the mostly-copying GC, which collects garbage by copying only movable objects. The mostly-copying GC divides the heap into equal sized blocks. In our garbage collector, each block has a generation. In minor GC, the collector does not only move movable objects in the new generation to the old generation, but also turns blocks into the old generation if they contain many live objects. This allows immovable objects to get promoted.

（平成 22 年 3 月 15 日発表）

†1 電気通信大学大学院電気通信学研究科情報工学専攻
Department of Computer Science, Graduate School of Electro-Communications, The University of Electro-Communications
†2 電気通信大学情報工学科
Department of Computer Science, The University of Electro-Communications