

発表概要

Rubyにおけるライトバリアのないオブジェクトを考慮した 世代別インクリメンタルGCの実装

笹田 耕一^{1,a)} 松本 行弘¹

2014年11月10日発表

世代別 GC, およびインクリメンタル GC はよく知られた GC の改善手法であるが, 実装するためには正確なライトバリアが必須である. 一方, Ruby 処理系は保守的マーク&スイープ GC を用いることで, C 言語などで拡張コードを書くときに余計なコードを含む必要がなかったが, ライトバリアを挿入しておらず, これを必要とする GC アルゴリズムを導入することができなかった. 我々はこの問題を解決するために, ライトバリアに対応しているオブジェクトとしていないオブジェクトを区別して, 世代別インクリメンタル GC を実装する新しい方法を提案する. 本発表では提案する手法について説明し, Ruby への実装について述べる.

Implementation of Generational Incremental Garbage Collector with Write Barrier Unprotected Objects on Ruby

KOICHI SASADA^{1,a)} YUKIHIRO MATSUMOTO¹

Presented: November 10, 2014

Generational GC and incremental GC are well known algorithms to improve performance of garbage collector, and they require correct write barriers in the interpreter. On the other hands, Ruby interpreter uses conservative mark and sweep GC, and we can implement C extensions easily without any extra codes. However, Ruby interpreter does not have any write barriers and we can not implement GC algorithms which require correct write barriers. To solve this problem, we propose new technique to implement generational and incremental GC by separating all objects into write barrier protected objects and write barrier unprotected objects. In this presentation, we will describe new technique and an implementation on Ruby interpreter.

¹ Heroku, Inc.
Heroku, Inc., Chiyoda, Tokyo 100-7012, Japan
^{a)} kol@heroku.com