GCにおける冗長なマーク処理のハードウェア支援による高速化

河村 慎二*，津邑 公暁（名古屋工業大学）

Hardware Support for Excluding Redundant Marking in common Garbage Collections

Shinji Kawamura, Tomoaki Tsumura (Nagoya Institute of Technology)

1. はじめに

スマートフォンなど，小容量のメモリしか搭載されておらず，メモリ管理機能の重要性が高いモバイル機器の普及に伴い，ガベージ・コレクション（**Garbage Collection: GC**）の性能が与える影響範囲が拡大している．本研究では，多くの実行環境で用いられる代表的なGCアルゴリズムに共通して存在する構成処理要素に着目し，これをハードウェア支援によって高速化する手法を提案する．

2. ガベージ・コレクション

GCとは，プログラムが動的に確保したメモリ領域のうち，不要になった領域を自動的に解放する機能である．プログラム実行時に動的に生成されたオブジェクトはヒープ領域に配置され，そのオブジェクトを指すポインタはグローバル変数やコールスタック，レジスタなどの，アプリケーションから直接参照可能な領域に格納される．このような領域の集合をルート集合と呼び，これを起点としてポインタを辿ることで，ヒープ領域内のオブジェクトを参照することができる．また，ヒープ領域内に配置されたオブジェクトは他のオブジェクトへのポインタを保持することもある．他のオブジェクトから参照されているオブジェクトは，ルート集合から複数のそのようなオブジェクトを経由することで参照される．このようにルート集合から直接，あるいは間接的に参照可能なオブジェクトを生きているオブジェクトと呼ぶ．一方，ルート集合から参照不能なオブジェクトを死んでいるオブジェクトと呼ぶ．GCはこの死んでいるオブジェクトを破棄し，これに割り当てられていたメモリ領域を解放することで，不要なオブジェクトに割り当てられていたメモリ領域を再利用できるようにする．

3. 提案手法

本研究では，まずモバイル端末の実行環境として広く普及しているDalvikVM[1]に実装されているGCアルゴリズムであるMark & Sweepの動作を調査した．その結果，オブジェクト間に存在する参照関係を辿りオブジェクトを探索，およびマークする処理に多くの時間を要していることが分かった．また，既にマーク済みのオブジェクトに対して繰り返しマーク処理が適用されていることも分かった．そこで本稿では，マーク済みのオブジェクトを管理するための専用表をプロセッサに追加し，オブジェクトの探索時にこの表を参照することで，マーク済みのオブジェクトに対する冗長なマーク処理を省略する手法を提案する．これにより，GC実行時間の多くを占めるオブジェクトの探索処理を高速化し，GCの高速化を実現する．

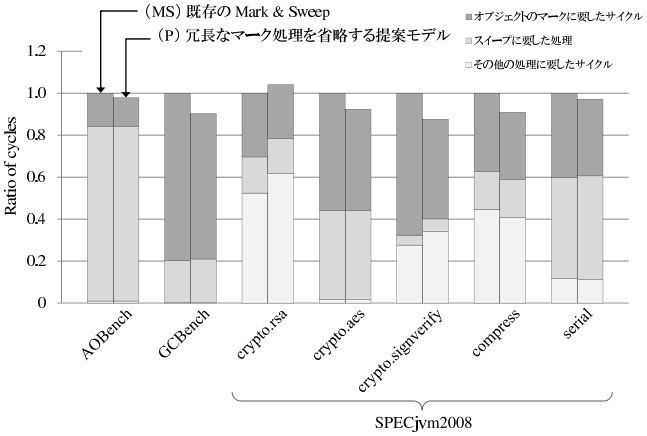

Fig.1　Evaluation Results

4. 評価

評価にはGCBench, AOBench, また汎用ベンチマークプログラムであるSPECjvm2008から5個の，計7個を使用し，既存のMark & Sweepと提案手法のGC実行サイクル数を計測した．評価結果をFig.1に示す．結果は，既存のMark & Sweepを実行するモデル(MS)，提案モデル(P)のGC実行サイクル数を示しており，既存のMark & Sweepを実行するモデル(MS)を1として正規化している．評価の結果，既存モデル(MS)と比較して提案モデル(P)はcrypto.rsaを除く全てのベンチマークプログラムでGCの実行サイクル数を削減できていることが分かる．これは，提案手法によって冗長なマーク処理が省略され，マークに要したサイクルが削減されたためである．本提案手法により，ベンチマークプログラム全体で，最大12.4%，平均5.7%のGC実行サイクル数を削減できることを確認した．

5. おわりに

本研究では，多くのGCアルゴリズムで必要となるオブジェクト探索処理をハードウェア支援によって高速化する手法を提案した．これにより，冗長なマーク処理を抑制し，オブジェクト探索処理の高速化を達成した．

文　献

(1) Bornstein, D.: Dalvik Virtual Machine Internals, *Google I/O 2008* (2008)