様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ６ 月　 ３ 日現在

機関番号：１３９０３
研究種目：挑戦的萌芽研究
研究期間：2013～2015
課題番号：２５５４００１９

研究課題名（和文）ガーベジコレクションをハードウェア支援する高速プロセッサ構成方式

研究課題名（英文）Hardware-Support for Common Garbage Collections

研究代表者
津邑　公暁（TSUMURA, Tomoaki）

名古屋工業大学・工学（系）研究科（研究院）・准教授

研究者番号：００３３５２３３

交付決定額（研究期間全体）：（直接経費）　　2,800,000円

研究成果の概要（和文）：GCは主にアルゴリズム面で改良がなされてきたが，GC実行時のレスポンス低下など，重要な問題の根本的解決には未だ至っていない．そこで本研究では，多くのGCアルゴリズムに共通する処理をハードウェア支援によって高速化することを目指した。まず，多くのGCがコールスタックを起点としてオブジェクトを探索する点に着目し，これを高速化するハードウェア支援手法を提案した。また，DalvikVM におけるマーク処理に冗長性がある点に着目し，GC処理時間内の多くを占める，クラスオブジェクトへのマーク処理の重複を防ぐ手法を提案した。

研究成果の概要（英文）：Many mobile systems have to achieve both high performance and low memory usage, and the total performance of the wide range of platforms now can be affected by the effectiveness of Garbage Collection (GC). GC algorithms have been actively studied and improved, but they still have not reached any fundamental solution. We focused on the point that the objects on the call stack should be traced in many GC algorithms, and proposed a hardware support technique for speed up of this trace. To trace objects, it is needed to find pointers on the call stack. Hence, we installed tables for managing all pointers on the call stack. By referring these tables, the GC routine can detect pointers immediately. The result of the simulation experiment shows the proposed method leads to low GC latency.

研究分野：計算機アーキテクチャ

キーワード：ガベージコレクション　ハードウェア支援

１．研究開始当初の背景

スマートフォン等のモバイル機器においては，アプリケーション開発効率等の観点から，VM 環境（例：Android の Dalvik）や Web API を基幹に用いた環境（例：B2G (Firefox OS)，webOS）などが多く採用・検討されている．しかしこれらは高いプロセッサ処理性能を必要とする上，特に実行時間に占める GC の割合が非常に大きいことが知られており，ときに十数%～数十%に及ぶとも言われる．

また，Dalvik 等では GC 実行中に全てのプロセス/スレッドが停止するため，GC の性能はユーザ体感品質に大きな影響を与える．一般に小容量のメモリ空間しか持たないモバイル機器においてメモリ管理の重要性は非常に高く，本研究はこれらの問題を高速かつ効率的な GC の実現により解決しようとするものである．

２．研究の目的

本研究では，高速な GC を可能とする専用ハードウェア，その専用ハードウェアを使用するための命令セットアーキテクチャ拡張，およびこの拡張命令セットをプログラム中から容易に使用可能とするための GC ライブラリの 3 つを実現することを目的とする．

GC の代表的なアルゴリズムはいずれも，特定の項目を検索するためのメモリ空間の走査が大きなオーバヘッドとなることを事前調査済であり，専用の表を用いてこれを高速化する手法を中心として実装を行う．

アルゴリズム面からの GC 高速化はこれまでも多く研究されているが，本研究では GC 処理をハードウェア的に支援可能なプロセッサを提案することで，大きな高速化をもたらすことを目標とする点が大きく異なる．

ハードウェアによる GC 支援には，わずかながら既存研究が存在する．しかし，これらはいずれも特定言語においてバリア同期のみを高速化するものである．これに対し本研究は，GC の代表的なアルゴリズムで必要となるメモリ空間の走査自体を，専用機構追加および命令セット拡張によってハードウェアサポートすることで高速実行することを目指す点が大きく異なる．

処理時間の多くを占める GC の高速化により，ユーザ体感品質の向上のみならず，モバイルプロセッサの高クロック化に歯止めをかけ，動作消費電力を抑制し，バッテリ持続時間の問題解決も期待できる．

３．研究の方法

まず，主にモバイル端末における代表的アプリケーションをシミュレータ実行し，トレースを取得しつつ実行時間の詳細な解析を行った．シミュレーションには，フルシステムシミュレータである gem5 を用い，ターゲットアーキテクチャとしては ARM プロセッサを想定した．

次にその解析結果から，GC アルゴリズム内のボトルネックとなる部分を洗い出し，それを重点的に高速化可能な機構を検討した．特に Mark&Sweep アルゴリズムを中心に高速化手法を設計した．さらにその機構をシミュレータに実装し，代表的アプリケーションの高速化率を評価した．

Mark&Sweep は，ヒープ領域内の到達可能オブジェクトをマークして回り，その後，マークされなかったオブジェクトを回収する手法であるが，ルート領域の走査コストが非常に大きい．これはマーク操作において，ヒープ領域へのポインタ捜索が大きなボトルネックとなるためである．特に，現在一般的となったマルチスレッド環境では，走査すべきコールスタックの増加や，マークスタック溢れによる影響などから，これが顕著となる．そこで，ヒープ領域へのポインタを，別途連想的に検索可能なバッファメモリを用いて高速に検索する手法を，設計・実装する．

４．研究成果

① 多くの GC アルゴリズムの高速化を目的として，GC の基本的な構成処理要素であるルートからのポインタ探索に着目し，これを高速化するハードウェア支援手法を提案した．ルートの一つであるコールスタック上には，ポインタだけでなく int 型変数などの値も含まれているため，GC 実行時にはこの中のポインタのみを判別するための処理が必要となる．そこで，プロセッサのハードウェアを拡張し，コールスタック上の全てのポインタを管理する専用表を追加することで，従来のポインタ判別コストの削減を図った．シミュレータを用いた評価により，提案手法による削減サイクル数，および表の操作にかかるコストを計算し，両者の比較を行ったところ，提案手法を適用することで GC 実行サイクル数の削減につながることを確認した．

② GC アルゴリズムにおいてオブジェクト間のポインタを辿ってメモリ上のオブジェクトを探索する必要がある点に着目し，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．オブジェクトの多くはクラスメソッド等の情報を持つクラスオブジェクトを参照している．そのため，クラスオブジェクトは複数のオブジェクトから参照される可能性が高く，クラスオブジェクトへのマーク処理の多くが重複してしまっている．そこで，一度探索したクラスオブジェクトへのポインタを管理する専用の表を用いることで，

クラスオブジェクトへのマーク処理の重複を防ぐ。また，オブジェクトへの参照を辿る際には，その参照が参照元オブジェクト内のどこに存在するかを計算によって求める必要があるが，この参照が存在する位置はオブジェクトのクラスごとに共通している．そこで，クラスオブジェクトの値と併せて，子オブジェクトへの参照の位置を表で管理することで，オブジェクト間の参照の探索コストを削減し GC の高速化を実現する．シミュレーションによる評価の結果，提案手法による GC の高速化が，スループット及び GC による停止時間を改善させることを確認した．

③ 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．この手法では，マーク済みのオブジェクトを記憶するための専用表をプロセッサに追加し，GC 実行時にこれを参照することで冗長なマーク処理を省略する．これにより，従来の冗長なマーク処理に要していたコストを削減し，GC の高速化を実現した．提案手法の有効性を確認するため，シミュレーションによる評価を行った．その結果，既存の Mark & Sweep と比較して，最大 12.4%の GC 実行サイクル数が削減できることが分かった．また，Concurrent GC では一部のベンチマークプログラムにおいてスループットが悪化したり，停止時間が長くなってしまうのに対し，提案手法ではそのような性能悪化を抑制できており，手法の有効性を確認した．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 4 件）

① 井手上慶，里見優樹，津邑公暁，松尾啓志：GC 実行時のポインタ判別コストを削減するハードウェア支援手法の検討，信学技報，Vol.113，No.169，pp.19‐24 (Jul. 2013)
② 里見優樹，井手上慶，津邑公暁，松尾啓志：GC におけるポインタ探索高速化のためのハードウェア支援手法，情処研報，Vol.2013-ARC-207, No.27, pp.1‐9 (Dec. 2013)
③ K. Ideue, Y. Satomi, T. Tsumura, H. Matsuo: Hardware-Supported Pointer Detection for common Garbage Collections, Proc. 1st Int'l Symp. on Computing and Networking (CANDAR'13), REGULAR PAPER, pp.134–140 (Dec. 2013) [A.R.: ~34.5%]
④ 井手上慶，河村慎二，津邑公暁：GC 実行時の高速なコンパクションを可能にするハードウェア支援手法の検討，情処研報，Vol.2014-ARC-212，No.1，pp.1‐9（Oct. 2014）

〔学会発表〕（計 7 件）

① 井手上慶，里見優樹，津邑公暁，松尾啓志：GC 実行時のポインタ判別コストを削減するハードウェア支援手法の検討，2013 年並列／分散／協調処理に関する『北九州』サマー・ワークショップ (SWoPP2013), 2013.08.01～08.03，北九州国際会議場（北九州市）
② 井手上慶，津邑公暁：ポインタ判別処理に着目したハードウェア支援による GC の高速化，第 5 回メニーコアアーキテクチャ研究会，2013.09.20～09.21，時之栖ホテル BrushUp（御殿場市）
③ 里見優樹，井手上慶，津邑公暁，松尾啓志：GC におけるポインタ探索高速化のためのハードウェア支援手法，第21回ハイパフォーマンスコンピューティングとアーキテクチャの評価に関する北海道ワークショップ(HOKKE-21)，2013.12.16～12.17，北海道大学学術交流センター（札幌市）
④ K. Ideue, Y. Satomi, T. Tsumura, H. Matsuo: Hardware-Supported Pointer Detection for common Garbage Collections, The 1st Int'l Symp. on Computing and Networking (CANDAR'13), 2013.12.04～12.06, ひめぎんホール（松山市）
⑤ 河村慎二，津邑公暁: GC のハードウェア支援に関する研究，第 6 回メニーコアアーキテクチャ研究会，2014.09.28～09.29，時之栖ホテル BrushUp（御殿場市）
⑥ 井手上慶，河村慎二，津邑公暁：GC 実行時の高速なコンパクションを可能にするハードウェア支援手法の検討，情報処理学会 第 204 回計算機アーキテクチャ研究発表会，2014.10.06～10.07，ホテルニューツルタ（別府市）
⑦ 廣田杏珠，津邑公暁: ハードウェア支援による GC 高速化手法に関する研究，第 7 回メニーコアアーキテクチャ研究会，2015.09.25 ～ 09.26，時之栖ホテル BrushUp（御殿場市）

〔図書〕（計 0 件）

〔産業財産権〕
○出願状況（計 0 件）

○取得状況（計 0 件）

〔その他〕
ホームページ等

６．研究組織

(1)研究代表者
津邑 公暁（TSUMURA, Tomoaki　　）
名古屋工業大学・大学院工学研究科・准教授
研究者番号：００３３５２３３