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 ２４ 日現在

機関番号：２６４０２
研究種目：基盤研究(C)（一般）
研究期間：2013～2015
課題番号：２５３３００８０

研究課題名（和文）マルチコア・メニーコア組込みシステム向け言語仮想機械のメモリ管理

研究課題名（英文）Memory Management for Managed Runtimes in Embedded Systems on Multi-Core and Many-Core Processors

研究代表者

鵜川　始陽（Ugawa, Tomoharu）

高知工科大学・工学部・准教授

研究者番号：５０４２３０１７

交付決定額（研究期間全体）：（直接経費）　　3,600,000円

研究成果の概要（和文）：本研究では，マルチコアやメニーコアプロセッサを使った組み込みシステムで動作する言語仮想機械に適したごみ集め(GC)の開発を行った．本研究では主に以下の成果を得た．(1)実時間アプリケーションに適した既存の並行コピーGCを実装し問題点を発見して改良した．(2)それを基にして新しい並行コンパクションGCを開発した．(3)GCの消費電力を削減する手法を開発した．(4)バグを発見するプログラム解析ツールを開発した．

研究成果の概要（英文）：In this research, we developed a garbage collection (GC) that is suitable for managed runtimes in embedded systems on multi-core and many core processors.The main results of this research are the following. (1) We implemented an existing concurrent copying GC, which is suitable for real-time applications. We identified problems on the GC algorithm and proposed solutions for them. (2) We developed a novel concurrent compacting GC based on the concurrent copying GC. (3) We developed a technique to reduce energy consumption by GC. We developed a program analysis tool for bug detection.

研究分野：プログラミング言語と処理系

キーワード：ガベージコレクション　プログラミング言語　マネージドランタイム　組込みシステム　プログラム解析

１．研究開始当初の背景

高性能と省電力を両立するために，組み込みシステムでもマルチコアプロセッサを搭載することは少なくない．今後はメニーコアプロセッサにシフトすると予想される．また，記憶階層も複雑になりキャッシュの重要性が増している．一方で，多様かつ大規模化したソフトウェアの生産性を高めるために，Java などの高級言語の利用が進んでいる．高級言語の実行環境である言語仮想機械(以下言語 VM)では，メモリ管理にごみ集め(以下 GC)が用いられる．しかし，組み込みアプリケーションに多い実時間アプリケーションは突然起動する GC とは相性が悪く，プログラマがプロセッの動作を完全に把握できる C++言語などが用いられる場合も多い．

研究開始当初は，スマートフォン用に開発された OS である Android が成功し，普及しつつある時期であった．Android は研究開始当初は Dalvik VM という言語 VM を使って，Java で記述されたアプリケーションを実行していた．初期の Android では，アプリケーションを止めて GC を行うストップザワールド方式だったが，アプリケーションを止めずに並行して GC を行う方式が主流になってきていた．そのため，実時間の応答が必要な，ユーザインタフェースを備えるアプリケーションや，組み込みシステムの制御への Android の応用が視野に入りつつあった．

２．研究の目的

本研究では，マルチコアプロセッサやメニーコアプロセッサを搭載した組み込みシステムの上での動作に適した，言語 VM のための GC を研究することを目的とする．開発する GC は，メニーコアプロセッサに適した並行および並列 GC，実時間性，低消費電力，メモリコンパクションの性質と機能を備えることを目指す．これにより，組み込みシステムのソフトウェアを高級言語で記述できる応用分野が広がり，ソフトウェア開発の生産性が向上すると期待できる．以下では課題となる性質と機能の詳細を説明する．

(1) マルチコア・メニーコア対応: マルチコアやメニーコア環境での GC は，複数のコアを有効利用するために，アプリケーションとは別の複数コアで GC を実行する並行 GC が必須である．単に並行 GC を行うだけでなく，同期や排他制御などを最小にしなければ高い性能は得られない．
(2) 実時間 GC: インタラクティブなアプリケーションや実時間アプリケーションでは，実時間 GC が必須である．最も時間がかかる実行経路での実行時間(worst case execution time; WCET)や必要なメモリ量の予測可能性も課題である．
(3) 省メモリ: 長期間連続して実行するシステムでは，空き領域の断片化を避けるためにコンパクションは必須である．また，GC が使用するデータ(メタデータ)や，GC が正しく動作するために予約するメモリも少ないことが望まれる．
(4) 省電力: バッテリ駆動デバイスでは，消費電力の削減は課題である．実行時間だけでなく消費電力の観点からも，キャッシュミスやアトミック命令の使用を削減することが望ましい．

３．研究の方法

並行コピーGC のひとつである Sapphire GC を基にして，並行コンパクションを行う GC を開発する．GC の正しさ性能の評価は以下の方法を用いる．

(1) モデル検査等の形式的手法を使う．
(2) GC の実装や評価の仕組みが揃っている研究用の Java VM である Jikes RVM に実装して評価する．
(3) Android に実装して評価する．

４．研究成果

(1)　本研究では，並行コピーGC のひとつである Sapphire GC のアルゴリズムを研究用の Jikes RVM に実装して，さらに，アルゴリズムを改良した．通常の並行 GC は，GC を開始するときや GC のフェーズが変わる時に，GC とアプリケーションの全てのスレッドが同期をとる必要があり，その時に数 10 ミリ秒から数 100 ミリ秒の間アプリケーションが停止してしまう．Sapphire GC では，全てのスレッドで同期をとる必要がなく，そのためアプリケーションが停止することがない，実時間アプリケーションに適した GC アルゴリズムである．

Sapphire GC は他の研究グループによって発表されていたアルゴリズムだが，本格的な実装は存在せず，十分なアルゴリズムの検証と評価がなされていなかった．本研究では，Sapphire GC を GC の研究で一般的に用いられる製品レベルの Java VM である Jikes RVM に実装し，動作の確認と性能評価をした．その結果，アルゴリズムの問題点を発見した．

①　まず，弱参照という Java の言語機能を実現するには，アプリケーションのスレッドを全て停止させる必要があった．これは，ある瞬間にアプリケーションが使っていない弱参照を全て特定して，一度に処理する必要があるからである．弱参照はあまり使用されない機能なので，全スレッドを停止させる実時間性に乏しい実装でも問題がないと考えられていた．しかし，本研究では，広く利用されているベンチマークプログラムでも，かなりの頻度で使用されていることを明らかにした．さらに，本研究では，アプリケーションを停止させずに弱参照

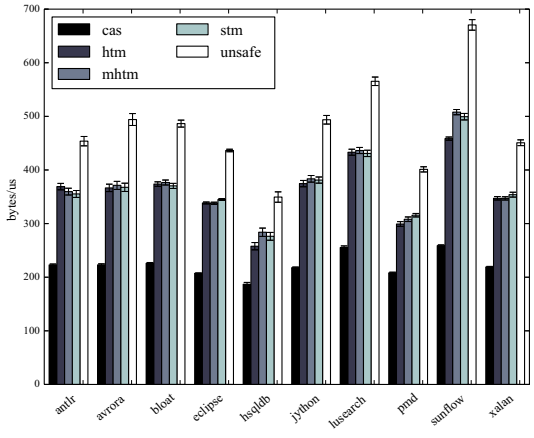

図 1 並行コピーのスループット

を少しずつ処理し，もしその途中でアプリケーションが弱参照を使えば，それを検出して弱参照の処理をやり直す手法を提案した．アプリケーションが弱参照を使ったのを検出して弱参照の処理をやり直すだけでは，やり直しを繰り返すだけでいつまでたっても弱参照の処理が終わらない可能性がある．本研究では，弱参照の処理をする直前に Sapphire GC で使われている Dijkstra スタイルの書込みバリアから，Yuasa スタイルの書込みバリアに切替えることで，弱参照の処理がいつか必ず終わるようにし，その性質をモデル検査を用いて検証した．この成果はメモリ管理に関する国際シンポジウム ISMM'14(雑誌論文(2))で発表した．

② 次に，並行コピーGC ではアプリケーションを動作させながらオブジェクトをコピーする必要があるが，Sapphire GC ではオブジェクトの全てのフィールドをアトミック命令を使ってコピーしていた．アトミック命令は最近の CPU が備える機構であるストアバッファの機能を抑制する．これによって，CPU がメモリにアクセスした際に生じる待ち時間に，依存がない命令を先行して実行する範囲が制限され，結果的にプログラムの実行が遅くなってしまう．さらに，消費電力も増えてしまう．本研究ではアトミック命令の使用数を大幅に削減したアルゴリズムを開発し，評価した．本研究のアルゴリズムはトランザクショナルメモリの考え方に基づいて，楽観的にコピーした後コピーが正しいかの確認をすることで．このアルゴリズムによりスループットが大幅に向上した．図 1 に実験結果を示す．図の縦軸はスループット(長い方がよい)である．図 1 の cas は従来のアルゴリズムのスループット，unsafe はアトミック命令を一切使わない理想的な場合の実行速度(正しく動作しない可能性がある)であり，理論上のスループットの上限である．本研究で開発したアルゴリズムの 3 つのバリエーションは htm，mhtm，stm であり，いずれも従来アルゴリズムに比べオーバヘッドが半分程度に削減されている(cas と unsafe の丁度中間付近のスループットが出ていることから分かる)．この成果はメモリ管理に関する国際シンポジウム ISMM'14(雑誌論文(2))で発表した．

③ さらに，GC のフェーズを進める時のプロトコルにおいても正しく動作しないケースが発見された．本研究では，GC のフェーズを進めるプロトコルの一般的なデザインパターンを開発し，プログラミング言語と処理系に関する国際シンポジウム APLAS'14(学会発表(6))で発表した．

(2) GC にかかる消費電力を削減することによって，プログラム全体の実行にかかる消費電力を削減する手法を開発した．GC はメモリの広い範囲を，規則性がない順序で 1 回ずつアクセスするという，キャッシュメモリと相性の悪いプログラムである．そのため，CPU が高速に処理しても，メモリアクセスに時間がかかり，待ち時間が増えてしまう．本研究では，GC の処理中だけ CPU の動作速度を抑制しても，GC にかかる時間はそれほど長くならず，動作速度を抑制による消費電力の削減の効果の方が大きい場合があるということを明らかにした．さらに，この成果を利用して，GC にかかる消費電力を削減する手法を開発した．この成果はプログラミングシンポジウム(学会発表(3)，(8))で発表した．また，特許も出願している．

(3) Sapphire GC のアルゴリズムをもとにして並行コンパクションのアルゴリズムを開発した．並行コンパクションのアルゴリズムは，GC 本体のアルゴリズムと，アプリケーションが GC に協力する仕組み(バリア)から構成されている．

① 本研究では，GC 本体のアルゴリズムを Android に試験実装して正しいことを確認した．試験実装はバリアを実装していないため，アプリケーションと並行に動作することはできないが，バリアを実装すれば並行に動作する見込が得られた．この成果はプログラミングおよびプログラミング言語ワークショップ(学会発表(1))で発表した．

② バリアのプログラム片は，アプリケーションから呼び出されてアプリケーションのコンテキストで実行される言語 VM のルーチン中の，Java のオブジェクトにアクセスする全ての箇所に正確に挿入する必要がある．しかし，本研究が対象とした言語 VM である ART は巨大なため，挿入する箇所を手作業で探すのは

難しい．本研究では，言語 VM のプログラムを静的に解析して，特定のパタンに合致する箇所を探すツール ASTgrep を開発した．ASTgrep の本研究での本来の目的はバリア挿入箇所を探すことであったが，ASTgrep はプログラムのバグを探すツールとしても有用であることが分かった．この結果は関数型と論理型プログラミングの国際シンポジウム FLOPS'16(学会発表(2))などで発表し，成果をまとめた論文は情報処理学会論文誌プログラミングに採録が決定している(雑誌論文 (1))．

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 3 件）

(1) 中村真也、鵜川始陽、馬谷誠二、規則違反コードの構造を反映した木パタンを用いるコード検査器、情報処理学会論文誌 プログラミング、情報処理学会、査読有、掲載予定(採録決定)

(2) Tomoharu Ugawa, Richard Jones, Carl Ritson, Reference Object Processing in On-The-Fly Garbage Collection, ACM SIGPLAN Notice - ISMM '14, ACM, Volume 49, Issue 11, pp. 59-69, 2014, 査読有、doi>10.1145/2775049.2602991

(3) Carl Ritson, Tomoharu Ugawa, Richard Jones, Exploring Garbage Collection with Haswell Hardware Transactional Memory, ACM SIGPLAN NOTICE - ISMM '14, ACM, Volume 49, Issue 11, pp. 105-115, 2014, 査読有、doi>10.1145/2775049.2602992

〔学会発表〕（計 8 件）

(1) 鵜川始陽、岩崎英哉、Android における部分コンパクションの実装、第 18 回プログラミングおよびプログラミング言語ワークショップ(PPL 2016)、ポスターセッション、2016 年 3 月 7 日、岡山県玉野市

(2) Tomoharu Ugawa, Seiji Umatani, Shinya Nakamura, A Source Code Checker Using Declarative Patterns to Represent Rule Violations, Thirteenth International Symposium on Functional and Logic Programming (FLOPS 2016), poster session, Kochi Japan, March 4-6, 2016

(3) 片岡崇史、鵜川始陽、並行 GC 中の CPU 周波数抑制による消費電力の削減、第 57 回プログラミングシンポジウム、ポスターセッション、2016 年 1 月 8 日、静岡県伊東市

(4) 竹内洋平、Java 仮想マシンの動作特性に注目した CPU 周波数制御の PC への適用、プログラミングおよびプログラミング言語ワークショップ(PPL 2015)、ポスターセッション、2015 年 3 月 5 日、愛媛県松山市

(5) 中野陽基、鵜川始陽、岩崎英哉、Android 上のごみ集めにおける停止時間の削減、プログラミングおよびプログラミング言語ワークショップ(PPL 2015)、ポスターセッション、2015 年 3 月 4 日、愛媛県松山市

(6) Tomoharu Ugawa, Carl Ritson, Richard Jones, An Implementation of On-The-Fly Copying Garbage Collection on Jikes RMV, 12th Asian Symposium on Programming Languages and Systems (APLAS 2014), poster session, Singapore, November 18. 2014, ポスター賞(first prize)受賞

(7) 中野陽基、鵜川始陽、Dalvik VM における並行メモリ割当ての実現、第 16 回プログラミングおよびプログラミング言語ワークショップ(PPL 2014)、ポスターセッション、2014 年 3 月、熊本県阿蘇市

(8) 橘田頼之、鵜川始陽、岩崎英哉、携帯端末における仮想機械での CPU 周波数抑制による消費電力の削減、第 55 回プログラミングシンポジウム、情報処理学会プログラミングシンポジウム予稿集、55 巻、pp. 75-81、2014 年 1 月 11 日、静岡県伊東市、http://id.nii.ac.jp/1001/00112989/

〔図書〕（計 1 件）

(1) Richard Jones(著), Antony Hosking(著), Eliot Moss(著), 前田敦司(訳)、鵜川始陽(訳)、小宮常康(訳)、翔泳社、ガベージコレクション 自動的メモリ管理を構成する理論と実装、2016, 560 ページ

〔産業財産権〕
○出願状況（計 1 件）

名称：電子機器、制御方法、及び、プログラム
発明者：鵜川始陽、橘田頼之、岩崎英哉
権利者：電気通信大学
種類：特許
番号：特願 2013-272936
出願年月日：2013 年 12 月 27 日

国内外の別： 国内

○取得状況（計　0　件）

〔その他〕
ホームページ等:
http://pl.info.kochi-tech.ac.jp/index.php/project_astgrep/
(ASTgrep の研究紹介ページ)

６．研究組織

(1) 研究代表者
高知工科大学・工学部・准教授
鵜川　始陽（UGAWA, Tomoharu）
研究者番号：５０４２３０１７

(2) 研究分担者
岩崎　英哉（IWASAKI, Hideya）
電気通信大学・情報理工学(系)研究科・教授
研究者番号：　９０２０３３７２
（平成 26 年度より研究分担者）