

論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Techniques for Mixed-mode Virtual Machines to Improve Overall System Performance
著者(和文)	緒方一則
Author(English)	Kazunori Ogata
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:乙第4128号, 授与年月日:2016年4月30日, 学位の種別:論文博士, 審査員:増原 英彦,南出 靖彦,渡邊 治,首藤 一幸,脇田 建
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:乙第4128号, Conferred date:2016/4/30, Degree Type:Thesis doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	審査の要旨
Type(English)	Exam Summary

論文審査の要旨及び審査員

(2000字程度)

報告番号	乙 第 号	学位申請者	緒方 一則	
論文審査員	氏 名	職 名	氏 名	職 名
	主査 増原英彦	教授	脇田建	准教授
	南出靖彦	教授		
	渡辺治	教授		
	首藤一幸	准教授		

本論文は「Techniques for Mixed-mode Virtual Machines to Improve Overall System Performance (インタープリタ - 動的コンパイラ併用型仮想マシンのシステム性能改善技術)」と題し英文7章からなる。

第1章は、論文の目的がインタープリタ - 動的コンパイラ併用型仮想マシン(以降併用型仮想マシン)の実運用環境における性能改善であるとした上で、言語処理系の様々な構成方式の中での併用型仮想マシンの位置付けと、その内部構造を説明している。さらに実運用環境においては、バイトコードインタープリタの性能、言語処理系自身が使用するデータによるオーバーヘッド、動的コンパイラのデバッグ情報によるオーバーヘッドが性能上の課題として残されていることを指摘している。

第2章は、本論文が対象とするJava言語について基本的な用語の定義を行い、Java言語処理系の実現方式の現状をまとめている。

第3章は、本論文が提案する3つの技術の背景を説明している。バイトコードインタープリタの実現については、代表的な手法としてスレッドコードがあり、その亜種が多く提案されていることを説明している。言語処理系自身が使用するデータの管理技術については、明示的な確保・解放によるメモリ管理ライブラリの基本的な仕組みを、OSによるメモリ管理との関連を含めて説明している。動的コンパイラのデバッグに関しては、一般的なプログラム、静的なコンパイラ、動的コンパイラそれぞれを対象とした場合のデバッグの特徴と、コンパイラのデバッグに一般的に用いられている診断記録方式について説明している。

第4章は、バイトコードインタープリタの効率改善方法を提案している。従来のバイトコードインタープリタはバイト単位で命令読み込みを行っているが、本章はワード単位で先読みを行う方法を提案している。さらにワード内の命令位置ごとにバイトコード命令処理部を特化したものを用意することで、非効率なバイト操作命令と分岐命令の遅延を削減する。標準的なベンチマークプログラムを用いた評価実験では、従来のインタープリタに比べて平均30.3%、最大で56.4%高速化を達成したことを示している。従来のバイトコードインタープリタの研究ではほとんど考慮されていなかったハードウェアレベルのメモリアクセスを意識することで、さらに効率の改善が可能であることを示した点で重要な提案となっている。

第5章は、Java言語処理系のメモリ使用状況の分析を行い、その改善方法を提案している。一般に仮想マシンは実行するプログラムが使用するデータ以外に、メタデータ、例えば実行時に読み込んだプログラム定義、実行時コンパイラの作業データ、コンパイルコードなどのためのメモリ領域を管理している。本章は実用規模プログラムの実行におけるメタデータの使用状況の実証的な分析を行い、得られた発見としてプログラム定義やコンパイルコードのほかに大きな部分を占めているのが解放済みの一時メモリ領域であることを報告している。さらにその領域は長時間に渡ってメモリを占有していること、その原因が動的コンパイラの作業領域に由来することを特定している。解決策として動的コンパイラがOSに強制的なメモリ解放を促す方法を提案し、実験によって使用メモリ領域を16~41%削減できることを示している。言語処理系のメタデータ使用に関する研究はほとんどない中で、この提案は実用的な大規模プログラムにおいて問題が発生していることと、その解決手法を示しており、重要なものである。

第6章は、動的コンパイラのデバッグを目的とした実行再現技術を提案している。動的コンパイラの誤りは、コンパイルされたコードの実行時まで異常が検出できないうえに、プログラム実行ごとに異なるコンパイルを行う可能性があるため、デバッグが難しい。従来はコンパイラの詳細な動作状況を記録す

る方法が用いられているが、その記録は数十から数百MBと大きなものになるため、実運用環境で使用することは現実的でなかった。これに対して本章は、言語処理系から動的コンパイラに対して与えられた入力記録と、異常検出時のメタデータだけでコンパイラの動作を再現できることを指摘し、再現に必要な最低限の入力の集合を特定している。実際に処理系を作成し、標準的なベンチマークプログラムで実験したところ、実行時に記録すべきデータ量を数十から数百KBと、実運用環境で常時動作させても問題がない水準に抑えられることを示した。一般的なプログラムの実行を記録・再現する手法はいくつか知られているものの、動的コンパイルの再現について実運用に耐える手法を実現した例はこれまでなく、独創的な提案となっている。

第7章は、論文のまとめとして、3つの技術についての貢献点を整理している。

以上をまとめると、本論文で提案されている3つの技術は、今日言語処理系の実現方式の主流となっている併用型仮想マシンが実運用環境において抱えている問題について、新しい手法によって改善できることを実証したものであり、理学上貢献するところ大である。よって本論文は博士(理学)の学位論文として十分な価値があるものと認める。

注意：「論文審査の要旨及び審査員」は、東工大リサーチリポジトリ(T2R2)にてインターネット公表されますので、公表可能な範囲の内容で作成してください。