

論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	High Performance Processor Design for Virtual Machine-Based Applications
著者(和文)	ThongkaewSurachai
Author(English)	Surachai Thongkaew
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第10014号, 授与年月日:2015年9月25日, 学位の種別:課程博士, 審査員:一色 剛,國枝 博昭,上野 修一,高橋 篤司,原 祐子,伊藤 和人
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第10014号, Conferred date:2015/9/25, Degree Type:Course 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

報告番号	甲第 号			学位申請者氏名		Surachai Thongkaew	
論文審査 審査員		氏 名	職 名		氏 名	職 名	
	主査	一色 剛	教授	審査員	原 祐子	准教授	
	審査員	國枝 博昭	教授		伊藤和人(埼玉大学)	教授	
		上野 修一	教授				
		高橋 篤司	教授				

論文審査の要旨 (2000 字程度)

本論文は、High Performance Processor Design for Virtual Machine-Based Applications (仮想マシン用アプリケーションのための高性能プロセッサ設計) と題し、英文 5 章から構成されている。

第一章 Introduction (緒言) では、近年様々な命令セットアーキテクチャに基づく計算機システムや高性能モバイル端末が開発される中、これら多様なハードウェアプラットフォームに依存しない形でソフトウェアの互換性・拡張性を担保するための「仮想マシン」(virtual machine)ソフトウェア実行環境と、その仮想マシンをベースとした様々なアプリケーションの開発が進んでいることに着目し、これら仮想マシン実行環境を実現するためのソフトウェア・ハードウェア上の技術課題について述べている。その中で、仮想マシン上で定義される「バイトコード」(標準機械語命令) を実行するための一連の処理 (命令フェッチ、デコード、オペランドフェッチ、演算実行、演算結果保存、プログラムカウンタ更新) がソフトウェアで実装されているために、仮想マシン上のアプリケーション実行速度の大幅低減の要因となっており、このバイトコード処理の効率化を図るためのソフトウェア・ハードウェア処理機構が必要であることを指摘している。このような背景の下で、本論文は仮想マシンのバイトコード処理の効率化を実現するための新たなソフトウェア・ハードウェア処理機構と、これら処理機構を既存プロセッサアーキテクチャの機能拡張で実現する実装方式を提案し、提案の仮想マシン処理機構の優位性を実証することが本論文の目的であると述べている。

第二章 Related Works (関連研究) では、仮想マシンの処理効率化技術に関する既存の研究成果を概説している。「バイトコード専用プロセッサ方式」(picoJava など) では、バイトコードそのものをプロセッサの命令セットとして直接実行するハードウェア機構を実装しており、単純なバイトコード命令を効率的に実行できる反面、OS を含む仮想マシン以外のソフトウェア群を実行できないため、システム構築のためには他のホストプロセッサと併用搭載が必要となることが欠点であることを指摘している。「アーキテクチャ拡張方式」(ARM Jazelle など) では、既存プロセッサアーキテクチャに一部のバイトコードを直接実行する専用デコード機能を追加することで、単純なバイトコード命令の効率的実行と、仮想マシン以外のソフトウェア群を既存プロセッサ部で実行する機能を共有できるため、仮想マシンの処理効率化技術としては最も有力であると述べている。ただし、既存のアーキテクチャ拡張方式では、複雑なバイトコード命令は従来通りのソフトウェア実装が必要である点と、仮想マシンレジスタをメモリ上に配置することによる大量のメモリアクセスが課題であると指摘している。

第三章 Dalvik Bytecode Acceleration Using Fetch/Decode Hardware Extension (フェッチ・デコード機能拡張による Dalvik バイトコード高速化手法) では、提案する仮想マシン処理機構の第一の方式として、Android にも搭載されている Dalvik 仮想マシンを例に、バイトコードの命令フェッチとデコード機能を既存プロセッサ上に機能拡張する実装方法を提案している。このプロセッサ機能拡張では、バイトコード用プログラムカウンタ、バイトコード命令バッファを追加することでバイトコードフェッチ機能を実装し、さらに、バイトコードデコード回路とオペランドアドレス計算回路を追加することでバイトコードデコード機能を実装している。これらの機能拡張により、バイトコード処理系における命令フェッチ、命令デコード、オペランドアドレス計算及びプログラムカウンタ更新の各ソフトウェア実装部分を削除するソフトウェア最適化が可能になり、その結果 1.8 倍～2.7 倍の処理速度向上を実験により示している。このソフトウェア最適化は、複雑なバイトコード命令にも適用可能なため、従来のアーキテクチャ拡張方式よりも広い範囲での命令処理速度向上効果を発揮できるとしている。またこれらの機能拡張に伴う回路面積の増加率は元のプロセッサに比べて 43%であることが示されている。

第四章 Register-Based Virtual Machine Acceleration Using Hardware Extension with Hybrid Execution (機能拡張と複合実行機構によるレジスタベース仮想マシン高速化手法) では、提案する仮

想マシン処理機構の第二の方式として、前述のフェッチ・デコード機能拡張方式に加え、単純なバイトコードを直接実行する「複合実行機構」（バイトコードとネイティブ命令コード両方を実行する機構）と仮想マシンのレジスタをプロセッサの実レジスタで実装する方式を提案している。そこでは、バイトコード種別判定回路を追加することで、単純なバイトコードを既存プロセッサの演算回路で直接実行する複合実行機構を実現し、さらに仮想マシンレジスタをプロセッサの実レジスタに置き換えるためのオペランドフェッチ回路を追加することで、従来メモリ上に配置されていた仮想マシンレジスタのオペランドフェッチに伴う大量のメモリアクセスを削減することに成功している。これらの機能拡張により、単純なバイトコード実行を直接プロセッサで実行できるようになり、その結果 2.2 倍～16.9 倍の処理速度向上を実験により示している。一方、機能拡張に伴う回路面積の元プロセッサに対する増加率は、既存プロセッサと同じ 32 個の実レジスタの場合 61%であり、256 個の実レジスタの場合 447% となることが示されている。また、256 個のレジスタを必要とする命令発生頻度は SPEC jvm ベンチマークではわずか 2%であり、32 個の実レジスタによるプロセッサ構成で十分な速度向上が見込めることが示されている。

第五章 Conclusion（結論）では、本論文で得られた結果をまとめている。

以上を要するに、本論文は仮想マシン処理機構をベースとしたアプリケーション実行環境を対象として、従来手法よりも処理速度向上効果の適用範囲が広く、メモリアクセス頻度を大幅に低減する新たなソフトウェア・ハードウェア処理機構と、これを既存プロセッサアーキテクチャの機能拡張で実現する実装方式とを考案したものであり、学術上および工業上貢献することが大きい。よって、我々は本論文が博士（学術）の学位論文として十分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。