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)	要約
Type(English)	Outline

Thesis Outline

Chapter 1: Introduction

In order to understand the evolution of computer programming languages from the past until now, a brief history of programming language is firstly described in this Chapter. Then, two kinds of programming language implementation (compiled language and interpreted language) included the advantages and disadvantages of each implementation are described to give an idea of the previous programming languages framework. The modern evolution, "Virtual Machine," which lead to new concept of programming languages (platform-independent) and the typical Virtual Machines, such as Java Virtual Machine and Dalvik Virtual Machine are described respectively in this Chapter. After that, the drawbacks of Virtual Machine and thesis motivations are described and discussed for understanding the objective and contribution of this thesis. The Introduction Chapter is outlined as the following sections.

1.1 Background of Programming Language
1.2 Programming Language Implementation
1.3 Background of Virtual Machine
1.4 Java Virtual Machine
1.5 Dalvik Virtual Machine
1.6 Discussion and Motivation
1.7 Thesis Objective and Contribution

Chapter 2: Related Works

This Chapter covers the works associated with improving performance of both the Java Virtual Machine (stack based) and Dalvik Virtual Machine (register based). The similarity of bytecode execution enhancement in both Virtual Machines is acceleration the speed of bytecode interpretation at runtime phase. The different methodologies of related works showed

various implementations, which can be classified and outlined into 4 groups as the following sections.

2.1 Software Acceleration
2.2 Dedicated Bytecode Processor and Co-processor
2.3 Multi-Core Processor
2.4 Architectural Extension

Chapter 3: Dalvik Bytecode Acceleration Using Fetch/Decode Hardware Extension

After the analysis of strengths and weaknesses of related works in Chapter 2, the architectural extension approach was chosen to explore in our thesis. In this Chapter, we introduce new architecture and software optimization techniques for a practical solution called "Fetch/Decode Hardware Extension". The implementations, experiments and conclusion are outlined as following section.

3.1 Proposed Solution and Architecture
3.2 The Experiments
3.3 Conclusion

Chapter 4: Register-Based Virtual Machine Acceleration Using Hardware Extension with Hybrid Execution

From the first proposed solution (in Chapter 3), the strength of our first technique is the optimized software handler, which can accelerate complex bytecodes, but the drawback is the limited acceleration of simple bytecodes. Hence, in this Chapter we propose an alternative methodology, the "Hardware Extension with Hybrid Execution" to further enhance the interpretation performance of both simple and complex bytecodes and focus on Register-based model. The details of second proposed solutions, such as hardware implementation, the experiments, the application to portable systems, the energy consumption and conclusion are outlined as following sections.

Chapter 5: Conclusion

Both of our proposed methodologies (“Fetch/Decode Hardware Extension”, “Hardware Extension with Hybrid Execution”) were summarized in this Chapter. The conclusion described particular techniques and results of each proposed methodology, such as the dedicate design of hardware and software to solve the Virtual Machine drawback, the mode switching overhead elimination, the proposed achievements and the proposed hardware consumptions.