T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

# 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）,<br>学位授与機関:東京工業大学,<br>報告番号:甲第10014号,<br>授与年月日:2015年9月25日,<br>学位の種別:課程博士,<br>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:,<br>Conferring organization: Tokyo Institute of Technology,<br>Report number:甲第10014号,<br>Conferred date:2015/9/25,<br>Degree Type:Course doctor,<br>Examiner:,,,,, |
| 学位種別(和文) | 博士論文 |
| Category(English) | Doctoral Thesis |
| 種別(和文) | 論文要旨 |
| Type(English) | Summary |

Powered by T2R2 (Science Tokyo Research Repository)

（博士課程）
Doctoral Program

# 論 文 要 旨

THESIS SUMMARY

| | | | |
|---|---|---|---|
| 専攻：<br>Department of | Communications and Integrated Systems 専攻 | 申請学位（専攻分野）：<br>Academic Degree Requested | 博士 (Philosophy)<br>Doctor of |
| 学生氏名：<br>Student's Name | Thongkaew Surachai | 指導教員（主）：<br>Academic Advisor(main) | Tsuyoshi Isshiki |
| | | 指導教員（副）：<br>Academic Advisor(sub) | |

要旨（英文 800 語程度）
Thesis Summary (approx.800 English Words )


The drawback of Process Virtual Machines (such as Java VM and Dalvik VM) is interpretation of bytecode software handlers, with the result that bytecodes execution is slower than compiled programming language execution. The architectural extension approach was chosen to exploration and new architecture with software optimization techniques was introduced for practical solutions. We proposed 2 methodologies to solve the drawback of Process Virtual Machines and improve speed up of bytedode execution as follows:

"Fetch & Decode Hardware Extension": to improve the performance of Dalvik Virtual Machine. The Fetch & Decode Hardware Extension is a specially designed hardware component to fetch and decode Dalvik bytecode directly. Dalvik bytecode fetch logic (DFE) fetches operation on frequently used 2-byte and 4-byte bytecodes with in one cycle and Dalvik bytecode decode logic (DDC) calculates the corresponding software handler address and preloads source/destination operand address (Vreg indices) to physical registers. While the core computations within the virtual registers are processed by the optimized Dalvik bytecode software handler.

"Hardware Extension with Hybrid Execution": to further enhance the performance of Process Virtual Machines interpretation and focus on Register-based architecture. This new technique provides an additional decoder, which can classify bytecodes into either simple or complex instructions. With "Hybrid Execution," the simple instructions will be directly executed on hardware of native processor. The complex instructions will be emulated by the "extra optimized bytecode software handler" of native processor. In order to eliminate the overheads of retrieving and storing operands on memory, we utilize the physical registers instead of (low address) virtual registers. In addition, the combination of 3 techniques: Delay scheduling, Mode predictor Hardware and Branch/goto controller can eliminate all of the switching mode overheads between native mode and bytecode mode as well.

Both of our methodologies enhance the efficiency of bytecode interpretation by using hybrid technique, which is the hardware extension into the existing processor to operate with handler software perfectly. The achievement of "Fetch & Decode Hardware Extension" was the Dalvik hardware extension enabled existing processor interpreted Dalvik bytecode to the maximum speed up to 2.7x @200 MHz by consuming the additional area of 0.03 $mm^2$ and power increment of 0.23 mW. The achievement of "Hardware Extension with Hybrid Execution" was the hardware extension enabled existing processor interpreted bytecode to the maximum speed up to 16.9x @200 MHz by consuming the additional area of 0.04 $mm^2$ and power increment of 0.24 mW.

From the above methodologies, the proposed hardware can speed up only the execution of frequently used bytecodes, which access 4-bit indices registers (V0-V15) such as move VA, VB ; const/4 VA, #B ; add-int/2addr VA, VB and so on. Therefore, we also proposed the optional methodology to support the other bytecodes, which access 8-bit indices registers (V0-V255). The optional methodology is the extension 256 registers (ER0-ER255) to the proposed hybrid execution hardware. Then the operands accessing methodologies need to be adjusted as follows:

- Utilize extended physical registers (ER0-ER255) instead of 8-bit indices virtual register (V0-V255).
- The existing physical registers (R16-R31) are utilized for computation in software handler and processing in Virtual Machine.
- The 16-bit indices virtual registers (V256-V65,535) are still mapped to the memory.

As the results of optional proposed methodology, the bytecodes which access 8-bit indices registers such as const/16 VAA, #BBBB ; if-eqz VAA, #BBBB ; add-int VAA, VBB, VCC and so on, can be speeded up as well. The optional proposed methodology enabled proposed hybrid execution processor to the maximum speed up to 16.9x @200 MHz by consuming the additional area of 0.31 $mm^2$ and power increment of 28.94 mW.

Three proposed solutions (2 proposed and 1 optional proposed solutions) can be utilized as the following suitability.

- The first proposed solution: Fetch/Decode hardware extension is a simple hardware extension, which consumes small area, power and energy. It can be applied simply to the other processors and achieved the higher speed than the original processor with a bit of hardware extension.
- The second proposed solution: hardware extension with hybrid execution accomplished the highest speed and consumed a few area, power and energy. Even though, the extension of hybrid hardware to the other processors will have more complexities, but it will perform the high efficiency with the small consumptions.
- The optional proposed solution: the second proposed solution with the 256 registers extension consumed a lot of power, area and energy. But it can perform a 16-bit index registers accessing with the highest efficiency. Thus, it is suitable for running huge applications on unlimited hardware resources systems.

Moreover, the proposed techniques are simple to apply to other processors. The application can be done with just in 2 steps. The first step is the hardware extension on the existing processor and the second step is the software optimization in the opcode handler.

備考 ： 論文要旨は、和文 2000 字と英文 300 語を 1 部ずつ提出するか、もしくは英文 800 語を 1 部提出してください。
Note : Thesis Summary should be submitted in either a copy of 2000 Japanese Characters and 300 Words (English) or 1copy of 800 Words (English).

注意：論文要旨は、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。
Attention: Thesis Summary will be published on Tokyo Tech Research Repository Website (T2R2).