

機関番号：12102

研究種目：基盤研究(B)

研究期間：2007～2010

課題番号：19300004

研究課題名(和文) 組み込み非対称マルチコアシステムのための仮想基盤環境

研究課題名(英文) Virtual Platform Environment for Embedded Heterogeneous Multi-Core Systems

研究代表者

追川 修一(OIKAWA SHUICHI)

筑波大学・大学院システム情報工学研究科・准教授

研究者番号：00271271

研究成果の概要(和文)：

本研究は、組み込みプロセッサ向け仮想化環境の研究、および異なるプロセッサアーキテクチャを持つプロセッサの相互接続手法の研究の2点を研究の目的とした。研究成果として、組み込みシステムで広く使用されているARMプロセッサをターゲットとする仮想化ソフトウェアの開発手法を明らかにし、また実際に実装することで手法の正しさを検証した。また、OpenCLを拡張したHybrid OpenCLにより、効率的に複数のプロセッサを相互接続できることを明らかにした。

研究成果の概要(英文)：

The objectives of this research are (1) a research on a virtualization environment for an embedded processor, and (2) a research on an interconnection method for heterogeneous processors. The research results include a clarified and verified method to construct a virtualization software targeted for the ARM processor architecture, which is a widely used processor architecture for embedded systems, and also an interconnection runtime system based on OpenCL, called Hybrid OpenCL, which enables efficient interconnection between multiple processors.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2007年度	3,000,000	900,000	3,900,000
2008年度	2,500,000	750,000	3,250,000
2009年度	2,200,000	660,000	2,860,000
2010年度	1,600,000	480,000	2,080,000
総計	9,300,000	2,790,000	12,090,000

研究分野：システムソフトウェア

科研費の分科・細目：情報学・ソフトウェア

キーワード：ソフトウェア学, 計算機システム, オペレーティングシステム

1. 研究開始当初の背景

我々の身の回りには、製品にコンピュータが組み込まれた組み込みシステムが数多くある。これらの製品は、その比較的単純な目的を達成するために従来8bitマイコン程度の安価なプロセッサにより制御を行うものが主流であった。ところが近年、利便性を高

めるためにLCD等の画面でGUIが提供されるようになり、さらにインターネット接続や他機器とのネットワーク化による連携・協調動作が行われるようになると、製品の差別化を図るための高機能化が急速に進み、組み込みシステムのソフトウェアの規模、複雑さは増大の一途をたどっている。そのため、高速かつ高機能なプロセッサが使用されるように

なり、ソフトウェアの基盤となるオペレーティングシステム (OS) も、小型プロセッサ用のリアルタイム OS (RTOS) から、より高機能な Linux 等の汎用 OS への移行が進行中である。ソフトウェアの規模、複雑さの増大は、その十分な信頼性の確保を困難にしているため、基盤ソフトウェアレベルでの信頼性向上が求められてきており、次世代の高信頼性環境を目指している OpenTC や QUALCOMM 社の次世代携帯電話基盤ソフトウェアでは仮想マシンモニタ (VMM: Virtual Machine Monitor) 上で OS を動作させるようになってきている。

一方、プロセッサは高性能化と省電力化を両立させるために、単なる高クロック化による性能向上から、複数のプロセッサコアをワンチップ化したマルチコアプロセッサへとトレンドが移り変わりつつある。これは、組み込みシステム用プロセッサも例外ではなく、例えば ARM MPcore やルネサス SH-X3 などのワンチップに 4 コアを搭載した組み込みシステム用マルチコアプロセッサが開発されている。トレンドが高クロック化からマルチコア化へとシフトしたことにより、現在のコア数は 4 程度が一般的ではあるが、近い将来には 8 コアや 16 コア、32 コアのマルチコアプロセッサが一般的になると思われる。

マルチコア化が進みコア数が増大すると、現在のように均質なコアを複数搭載するのではなく、性能的に特色を持ったコアを組み合わせて搭載できるようになる。Sony、東芝、IBM による Cell プロセッサは 1 つの汎用的な用途のコア、および 8 つの計算に特化したコアからなる非対称マルチコアシステムの実例であり、また Intel も Many-Core 戦略として非対称マルチコアプロセッサアーキテクチャを発表している。将来的にはアプリケーションの用途に適したコアを組み合わせるようになると思われ、用途が比較的限定しやすい組み込みシステムではこの傾向はさらに強まると予想される。

2. 研究の目的

本研究「組み込み非対称マルチコアシステムのための仮想基盤環境」は、将来的なプロセッサのトレンドになっている非対称マルチコアプロセッサを活用するために必要となる機能を、組み込みシステムを対象として研究することを目的としている。そのため、主に次の 2 点を研究の目的とした。

(1) 組み込みプロセッサ向け仮想化環境の研究：複数の異なるプロセッサアーキテクチャからなる非対称マルチコアプロセッサを活用するためには、異なるアーキテクチャの差異を吸収する必要がある。そのため的手法として、本研究では仮想化環境を用いる。

しかしながら、仮想化環境はサーバが主な用途であったため、組み込み用途で幅広く使用されているプロセッサ用の仮想化環境の研究開発の事例はなかった。そのため、組み込み用途で幅広く使用されているプロセッサをターゲットとした仮想化環境の研究を行うことを目的の 1 つとした。

(2) 異なるプロセッサアーキテクチャを持つプロセッサの相互接続手法の研究：非対称マルチコアプロセッサアーキテクチャのプログラミングをサポートするためには、言語レベルでプロセッサの相互接続を実現する必要がある。そこで、各プロセッサの性能を最大限生かしつつ、複数の異なるプロセッサアーキテクチャを接続するために、言語レベルでプロセッサの相互接続を行う手法について研究開発を行うことを 2 つめの目的とした。言語レベルで、複数のプロセッサを接続する際に、ランタイムは仮想化したものとなる。そのような仮想化したランタイムの研究開発の研究開発を行うことも目的とした。

3. 研究の方法

2 つの研究目的について、それぞれ研究の方法について述べる。

(1) 組み込みプロセッサ向け仮想化環境の研究：研究開始時点で、組み込みプロセッサをターゲットとした仮想化ソフトウェアについての研究開発事例はなかった。そのため、開発当初から組み込みプロセッサをターゲットとすることはリスクを伴うと考えた。そこで、まずは仮想化の事例があるプロセッサについて、仮想化ソフトウェアの開発を行い、組み込みプロセッサをターゲットとした仮想化環境を構築するにあたり問題となる点について検討を行うこととした。この事例のあるプロセッサとしては x86 プロセッサを用いるものとした。開発開始当初、x86 プロセッサの仮想化についての研究はさかんに行われていたため、仮想化ソフトウェア開発のための情報を多く入手することができた。x86 プロセッサをターゲットとし、組み込みシステムを用途として想定し、軽量かつリアルタイム性を考慮した設計を行い、実装することにした。

すでに事例のあるプロセッサとして x86 プロセッサについて、仮想化ソフトウェアの開発を行った後に、組み込みプロセッサをターゲットとした仮想化ソフトウェアについての研究を行うこととした。ターゲットとしては、組み込みシステムで広く使用されている ARM プロセッサを使用することとした。仮想化ソフトウェアを実現するためには、ターゲットとするプロセッサについて、実行環境に影響を与えるセンシティブ命令が何であるか、それらセンシティブ命令はプロセッサが

特権モードである時のみ実行可能な、即ち、非特権モードでの実行時には例外を起こす特権センシティブ命令であるか、または非特権モードでも実行できてしまい、特権モードでの実行時とは異なる副作用を引き起こす非特権センシティブ命令であるかの分類をする必要がある。ARM プロセッサについては、この特権センシティブ命令および非特権センシティブ命令の分類が明らかになっていなかったため、この分類をすることから始める必要があった。また、分類が正しいことを検証するために、その分類をもとに構築した仮想化ソフトウェア上で Linux オペレーティングシステムを動作させることとした。

また、仮想化ソフトウェアの開発は 2 段階に分けて行った。まずは、仮想化ソフトウェアの実行環境としてプロセッサエミュレータを用いて開発を行った。これは、仮想化ソフトウェアのデバッグを容易にするためと、センシティブ命令の挙動を正確に把握するためである。プロセッサエミュレータ上で動作する仮想化ソフトウェアを実現し、Linux オペレーティングシステムを動作させることでその実装の正しさを検証した後に、実機上に仮想化ソフトウェアを実現し、詳細な性能評価を行うこととした。

(2) 異なるプロセッサアーキテクチャを持つプロセッサの相互接続手法の研究：異なるプロセッサアーキテクチャを持つプロセッサの相互接続を行うために、独自の仮想化レイヤについて研究を行う方針であったが、研究開始時にちょうど異種混在の計算資源を利用した並列コンピューティングのためのフレームワークとして OpenCL が制定、標準化され、また組み込み用のプロファイルも策定されているため、OpenCL をベースに研究を行うこととした。

OpenCL を用いることで、異なるプロセッサで行う処理を、言語レベルで記述可能になる。しかしながら、非対称マルチコアシステムに対応するためには、異なるプロセッサ同士を接続する仮想化されたランタイムが必要になる。ここでは、そのようなランタイムを持つ Hybrid 化された OpenCL について研究開発を行うこととした。

Hybrid 化された OpenCL (以下、Hybrid OpenCL) は、異なる複数のプロセッサ同士を接続することのできる OpenCL 実装である。そのような OpenCL 実装を実現し、動作を検証するために、ネットワーク化された環境を用い、その上で複数の OpenCL 実装を接続するランタイムを研究することとした。

4. 研究成果

研究の目的と方法を述べた、2 つの項目のそれぞれについて研究成果を述べる。最後に、

2 つの研究成果を統合した場合の展望について述べる。

(1) 組み込みプロセッサ向け仮想化環境の研究：まず、x86 プロセッサをターゲットとし、組み込みシステムを用途として想定し、軽量かつリアルタイム性を考慮した仮想化ソフトウェアの設計を行い、実装した。軽量化のために、モード遷移のオーバーヘッドを少なくするために x86 の 4 レベルのリングアーキテクチャを活用、割り込みや例外については仮想化ソフトウェアを経由することなく直接カーネルに通知、センシティブ命令については実際に使用されているエミュレーションの必要な命令のみデコーダを実装することで、オーバーヘッドの小さい仮想化ソフトウェアを実現することができた。さらに、軽量化のために、割り込みや例外については仮想化ソフトウェアを経由することなく直接カーネルに通知することにしたため、当初、仮想化ソフトウェアは全て割り込み禁止で実行することとしていた。これが、割り込み処理のリアルタイム性に影響することがわかったため、仮想化ソフトウェアを実行中であっても、カーネルに割り込みを通知できるようにする機構についても研究を行った。その結果、割り込み禁止で実行される命令数を大幅に削減することが出来た。

次に、組み込みシステムで広く使用されている ARM プロセッサをターゲットとし、仮想化ソフトウェアの設計を行い、実装した。仮想化ソフトウェアを実現するために、ARM プロセッサについて、特権センシティブ命令、非特権センシティブ命令が何であるかを明確に定義した。最も問題となるのは、特権モードでの実行と非特権モードでの実行で、異なる副作用を引き起こす非特権センシティブ命令の存在である。これらを明確にすることで、仮想化ソフトウェアで正しくセンシティブ命令をエミュレーションすることができるようになった。さらに、ARM プロセッサの特徴および x86 プロセッサとは大きく異なる点として、特権と非特権の 2 つしかないプロセッサモード、バンクレジスタ、ドメインによるメモリアクセス制御がある。特権と非特権の 2 つしかないプロセッサモード、およびバンクレジスタに関しては、これらを仮想化ソフトウェアにより仮想化し、仮想プロセッサモード、仮想バンクレジスタを提供することにより対応できることを示した。また、ドメインによるメモリアクセス制御を用いることにより、カーネル・ユーザ間のアクセス制御を効率的に実現できることを示した。開発には QEMU プロセッサエミュレータを使用した。開発した仮想化ソフトウェア上で Linux を動作させることにより、設計の正しさを示すことができた。

プロセッサエミュレータ上に実現した仮

想化ソフトウェアは、実機上に移植され、詳細な性能評価を行った。実機への移植にあたり、同じ ARM プロセッサではあるが異なるマイクロアーキテクチャを持つプロセッサをターゲットとしたため、異なるマイクロアーキテクチャでは異なる非特権センシティブ命令を持つことも明らかにすることが出来た。性能についてベンチマークプログラムを実行することでオーバヘッドを計測し、パフォーマンスモニタレジスタを活用したオーバヘッド要因の解析を行った。解析結果から、センシティブ命令が連続して実行されることが多く、これが性能に影響していることがわかった。そこで、これらを単一の命令に仮想化することで、性能向上が可能なことを明らかにした。

(2) 異なるプロセッサアーキテクチャを持つプロセッサの相互接続手法の研究：仮想化された Hybrid OpenCL ランタイムにより非対称マルチコアシステムの接続を可能にした。OpenCL は同一種類のデバイスのみ対象としているが、Hybrid OpenCL ではデバイスを仮想化したランタイムを開発し、透過的に接続可能にすることで、非対称なデバイスの取り扱いを可能にした。

Hybrid OpenCL 開発にあたり、OpenCL が前提とする単一マシン、即ち同一種類の均一なプロセッサ上での実行環境を、複数に分割することから開発を始めた。複数の非均一非対称な環境を模擬するため、ネットワークで接続された複数マシンからなる実行環境を用いた。まずは 1 対 1 で接続された 2 台のマシンを用い、1 台のホストマシンにおいてもとなる OpenCL アプリケーションを実行し、もう 1 台のリモートマシン上でその OpenCL アプリケーションが使用する OpenCL カーネルを実行できるように、ランタイムを分離した。そのランタイムの分離のためには、もともと OpenCL の仕様では想定されていなかった複数のリモートデバイスをリストアップし管理する機能、および計算リソースの実態とそれへのポインタを分離して管理する機能が必要であるとわかった。Hybrid OpenCL は、これらの機能を実現した。

さらに、1 台のホストマシンと複数台のリモートマシンからなる構成にも対応することで、OpenCL API の全てがエラーコードを返す仕様になっていることが、大きな通信コストを引き起こすことがわかった。また、模擬環境で用いたネットワークで複数マシンを接続した環境ではデータの送受信も大きなオーバヘッドとなることがわかった。OpenCL API については、ホスト側とリモート側で非同期に API を実行することでオーバヘッドを減少させる方法について検討を行った。また、データの送受信については共有メモリを持つことが出来れば、大きなオーバヘッドにな

らないことが検討された。

これら 2 つの研究成果を統合することで、非対称なプロセッサを単一のプログラミング環境で扱うことが出来るようになると考えられる。(1)の仮想化環境を用いることにより、異なる計算リソースを効率的に分離することが可能になる。その上で(2)のプロセッサの相互接続手法を用いることで、他方のプロセッサを透過的に利用することができるようになる。

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)

[学会発表] (計 10 件)

- ① Ryo Aoki, Shuichi Oikawa, Takashi Nakamura, Satoshi Miki: Hybrid OpenCL: Enhancing OpenCL for Distributed Processing. The 9th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA 2011), May 26, 2011, Busan Lotte Hotel, Busan, Korea.
- ② Ryo Aoki, Shuichi Oikawa, Ryoji Tsuchiyama, Takashi Nakamura: Hybrid OpenCL over High Speed Networks, IEEE Region 10 International Conference, November 23, 2010, Fukuoka International Congress Center, Fukuoka, Japan.
- ③ Ryo Aoki, Shuichi Oikawa, Ryoji Tsuchiyama, Takashi Nakamura: Improving Hybrid OpenCL Performance by High Speed Networks. International Workshop on Advances in Networking and Computing, 1st International Conference on Networking and Computing, November 19, 2010, University of Hiroshima, Hiroshima, Japan.
- ④ Ryo Aoki, Shuichi Oikawa, Ryoji Tsuchiyama, Takashi Nakamura: Hybrid OpenCL: Connecting Different OpenCL Implementations over Network. The 10th IEEE International Conference on Computer and Information Technology, July 1, 2010, University of Bradford, Bradford, UK.
- ⑤ Akihiro Suzuki, Shuichi Oikawa: Implementation of Virtual Machine Monitor for ARM Architecture. The 10th IEEE International Conference on Computer and Information Technology, June 30, 2010, University of Bradford, Bradford, UK.

- ⑥ 鈴木章浩, 追川修一, ARMアーキテクチャ用仮想マシンモニタの実装, 情報処理学会第 113 回システムソフトウェアとオペレーティング・システム研究会, 2010 年 1 月 27 日, 札幌コンベンションセンター (北海道).
- ⑦ 青木亮, 追川修一, 異なる OpenCL 実装を接続可能にする Hybrid OpenCL の構築, 情報処理学会第 113 回システムソフトウェアとオペレーティング・システム研究会, 2010 年 1 月 27 日, 札幌コンベンションセンター (北海道).
- ⑧ 青木亮, 追川修一, ヘテロジニアスマルチコアエミュレータにおけるコア間通信とアトミック命令の実装, 情報処理学会第 111 回システムソフトウェアとオペレーティング・システム研究会, 2009 年 4 月 23 日, 沖縄青年会館 (沖縄県).
- ⑨ Megumi Ito, Shuichi Oikawa: Improving Real-Time Performance of a Virtual Machine Monitor Based System. The 6th IFIP International Workshop on Software technologies for future Embedded and Ubiquitous Systems (SEUS 2008), October 2, 2008, Anacapri, Italy.
- ⑩ Megumi Ito, Shuichi Oikawa: Lightweight Shadow Paging for Efficient Memory Isolation in Gandalf VMM. The 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC 2008), May 7, 2008, Florida, USA.

6. 研究組織

(1) 研究代表者

追川 修一 (OIKAWA SHUICHI)

筑波大学・大学院システム情報工学研究科・准教授
研究者番号 : 00271271