

科学研究費助成事業 研究成果報告書

平成 26 年 6 月 3 日現在

機関番号：12102

研究種目：基盤研究(C)

研究期間：2011～2013

課題番号：23500032

研究課題名(和文)並列処理に基づいたOSカーネルの再構成

研究課題名(英文)Reconstructing an OS kernel based on parallel processing

研究代表者

追川 修一(Oikawa, Shuichi)

筑波大学・システム情報系・准教授

研究者番号：00271271

交付決定額(研究期間全体)：(直接経費) 3,900,000円、(間接経費) 1,170,000円

研究成果の概要(和文)：本研究は、プロセッサの将来的なトレンドとなっているメニーコアプロセッサおよびSIMDユニットを、オペレーティングシステム(OS)カーネル内での処理にも活用することができる形態にOSカーネルのソフトウェアを再構成可能にすることを目的とした。これまではOSカーネル自体の実行環境を維持するため、プロセッサを制御する必要があることから特定のプログラミング言語が使用されることが多く、またプロセッサの並列処理機能を有効に活用できなかった。本研究では、OSカーネルを実現する環境の柔軟性を高めることを可能にした基盤実行環境を開発した。そして、その上で、OSカーネルにおける処理の並列性を高める研究を行った。

研究成果の概要(英文)：This work investigated the reconstruction of the software of an operating system (OS) kernel in order to enable the utilization of many core processors and SIMD units inside the kernel. It is required for the existing OS kernels to maintain their execution environment by themselves, and such a restriction forced the use of specific programming languages and became the barrier to utilize the parallel processing functions of processors. This work constructed the base execution environment that improves the flexibility of the OS kernel implementation. Upon the constructed base execution environment, the researches to utilize more parallel processing features inside the OS kernel were conducted.

研究分野：オペレーティングシステム

科研費の分科・細目：情報学・ソフトウェア

キーワード：オペレーティングシステム 計算機システム

1. 研究開始当初の背景

プロセッサの進化は、高性能化と省電力化を両立させるために、高クロック化による性能向上からマルチコア・マルチスレッド化にシフトし、実際に出荷される汎用製品レベルのコア数も年々増えている。研究開始当初の時点で、Intel は 6 コア 12 スレッド、AMD は 12 コアの製品が出ていたが、現時点では Intel は 15 コア 30 スレッド、AMD は 16 コアまで増加している。組込み分野では、研究開始当初はマルチコアのプロセッサは限定的な利用に留まっていたが、現時点では 4~8 コアのプロセッサが幅広く用いられるようになってきている。さらに、データ並列処理を高速化するためのプロセッサ機能として、SIMD ユニットの搭載も広まっている。

一方、ソフトウェアにおいては、アプリケーションレベルではスレッドライブラリや並行プログラミング言語の利用により、マルチコア、メニーコアへの対応が進んでいる。SIMD ユニットには、ライブラリやフレームワークでの対応が行われている。また、並行プログラミング機能が言語レベルで組み込まれたオブジェクト指向言語では、スレッドの生成をオブジェクトの生成と同一化でき、またオブジェクト単位での排他制御が可能であることから、比較的容易にマルチスレッド対応のアプリケーションを作成することができるようになっている。

しかしながら、オペレーティングシステム(OS)カーネルにおいては、対称型マルチプロセッサ(SMP)への対応から実質的に進歩が見られない。SMP 対応の OS カーネルはマルチスレッドカーネルと呼ばれるが、これはアプリケーションへのカーネルレベルでのスレッドの提供、および複数スレッドがカーネル内を実行できることを意味する。カーネルレベルでスレッドを提供することにより、複数あるプロセッサのそれぞれでスレッドを実行可能になる。また、これらのスレッドがシステムコールを発行した際に、複数スレッドがカーネル内を実行可能であることで、システムコールの処理を並列に行なうことができる。

このような OS カーネルにおけるマルチプロセッサの利用形態では、メニーコアや SIMD ユニットに十分に対応しきれていない。あるスレッドがシステムコールを発行した時、1 つのプロセッサのみがそのシステムコールを処理し、他にアイドル状態のコアや SIMD ユニットがあったとしても何ら性能向上に寄与しないからである。また、処理の基本は逐次処理のままである。

これまでに研究されてきた OS カーネルのメニーコア対応は、メニーコアを活用することによる本質的な性能向上を目的とはしていない。複数の論理コンピュータにパーティション分割(論理分割)することによるカーネル内の資源競合の抑制、アプリケーション

のマルチスレッド化を支援する OS など、既存のソフトウェア資産を生かしつつオーバーヘッドを減らそうとする、保守的なアプローチが取られてきた。

2. 研究の目的

本研究は、プロセッサの将来的なトレンドとなっているメニーコアプロセッサおよび SIMD ユニットの、アプリケーションだけでなく、オペレーティングシステム(OS)カーネル内での処理にも活用することができるように、並列処理を活用可能な形態に OS カーネルのソフトウェアを再構成することを目的とする。本研究では、再構成を可能にするため、OS カーネルを実現する環境の柔軟性を高める。即ち、これまでは OS カーネル自体の実行環境を維持するため、プロセッサを制御する必要があることから、特定のプログラミング言語が使用されることが多かったが、様々な記述言語を用いることができる基盤実行環境を開発する。その上で、OS カーネルにおける処理の並列性を高める研究を行う。

3. 研究の方法

本研究では、それ自体の実行環境を維持する必要のない OS カーネルの記述を可能にする基盤実行環境を開発する。これまで、型安全であったり並行プログラミング機能が言語レベルで組み込まれていたりする高レベルな言語により OS カーネルを記述する研究開発は、その高レベル言語で書かれたプログラムを実機上で直接動作させることに多くの労力が費やされてしまっていた。

本研究では、OS カーネル実現の柔軟性を高める基盤実行環境を、既存の汎用 OS である Linux 上に構築する。この基盤実行環境上で実行される OS カーネルは、Linux のユーザプロセスとして実行されるものとする。従って、実機を直接制御し、それ自体の実行環境を維持する必要はなくなり、Linux 上のプログラムとして OS カーネルを記述することができるようになる。

OS カーネルは本来プロセッサを含むハードウェアを制御するためのものである。Linux 上のプログラムとして高レベル言語によって記述されていたとしても、OS カーネルとして機能するためには、プロセッサの制御を行う必要がある。そのために、単一システム内ではあるが Linux カーネルにより制御されないプロセッサを残しておく。そのプロセッサ上で小さな制御カーネルを実行させる。この制御カーネルを、Linux カーネル上でユーザプロセスとして実行する OS カーネルから制御する。この組み合わせにより、間接的ではあるが、ユーザプロセスとして実行する OS カーネルからのプロセッサの制御を実現する。

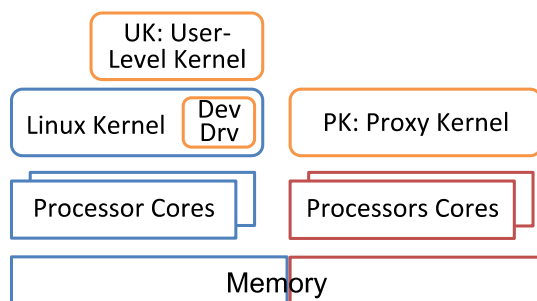
基盤実行環境が機能することを実証するために、基盤実行環境上で、ユーザプロセスとして実行する OS カーネルを実現する。基盤実行環境上で実行される OS カーネルは、Linux 上のプログラムとして実行されるため、既存の OS カーネルはそのままでは実行できない。Linux 上のプログラムとして実行できるように、プロセッサの制御はプロセッサ上で実行される制御プログラムを通して間接的に行うように、既存の OS カーネルを変更する。

基盤実行環境を用いて、処理の並列性を高める研究を行う。そのために、機能分割による処理の並列化、命令レベルでの並列処理機能を導入する。

4. 研究成果

(1) 特定の言語を前提とせずに OS カーネルの記述を可能にする基盤実行環境

開発した基盤実行環境は、Proxy Kernel (PK), User-Level Kernel (UK), デバイスドライバから構成される。以下に、基盤実行環境を図示する。



UK が、基盤実行環境におけるカーネル本体であり、Linux 上のユーザプロセスとして実行する。ユーザプロセスであるため、通常のカーネルと異なり、UK 自体の実行環境を維持する必要がなくなり、従ってランタイム環境を必要とする言語でも、UK は記述可能になる。UK はユーザプロセスであり、また PK が管理するプロセッサコアとは別のコア上で実行される。

PK は、プロセッサコアを直接制御するための単純なカーネルである。PK は、単純な資源管理のみを行い、複雑な管理や制御は UK が行う。そのため、PK は単純化が可能であり、プロセッサ資源を消費しないため、プロセッサの利用効率の向上が可能である。

UK と PK は別々のコア上で実行されるが、互いに通信する必要がある。そのため、Linux にデバイスドライバを導入する必要がある。デバイスドライバは、Linux から PK を起動するためにも用いられる。

(2) 基盤実行環境上における OS カーネルの実現

(1)で述べた基盤実行環境上で動作する UK の開発を行った。既存の小型 UNIX カーネルである Xv6 を変更し、UK として実装した。UK の特徴である、UK 自体の実行環境を維持する必要がない点から、そのために必要な実装が削除されたため、PK との通信機能を追加しても、全体としては簡素化されている。

性能評価から、getpid システムコールのような非常に単純なコストが少ない処理に関しては、UK/PK 間の通信コストの影響を受けるものの、fork や exec のような動的資源管理をとまなう複雑な処理では、UK/PK の分割により、TLB のフラッシュが不要になり、またキャッシュの局所性が向上するなど、プロセッサの利用効率が高まる。そのため、性能が向上することがわかった。通信コストの影響を受ける単純な処理に関しては、PK にその機能を移動することで、単一のカーネルよりも、処理の並列化、プロセッサの利用効率の向上により、全体の処理性能を高めることができています。

また、Java 言語による OS カーネル実装についての研究も行った。一般的に OS カーネルの実装に用いられる C 言語と比較し、Java 言語は、強い型付けによるデータ型の間違いによる実行時エラーの回避、ゴミ集め機能 (GC) が組み込まれていることによるメモリリークの回避、オブジェクト指向プログラミングによる拡張性の向上、といった利点がある。さらに、並行プログラミング機能が言語レベルで組み込まれていることから、マルチスレッド化に容易に対応できるようになっている。Xv6 に相当する機能を持つ OS カーネルを実装し、その評価を行った。

(3) 命令レベルでの並列処理機能の活用

Xv6 ベースの UK に、命令レベルでの並列処理機能である SIMD ユニットの導入する研究を行った。SIMD ユニットの導入はデータ並列処理を高速化するために、近年、汎用プロセッサだけでなく、組み込み向けのプロセッサを含め、多くのプロセッサに導入されている。SIMD ユニットの導入は、多倍長のレジスタを提供し、そこに置いた複数のデータに対し、同一の処理を行うものである。

UK の fork および exec 処理をプロファイラを用いて解析したところ、メモリコピーが処理の多くを占めていることがわかった。SIMD ユニットの導入は、メモリコピーの高速化に有効であるため、SIMD ユニットの導入によりコピー処理を行うように変更した。その結果、fork で 27%、fork+exec で 13%の性能向上を実現することができた。

(4) RTOS への並列処理の導入

スマートフォンやタブレット等に用いられている組み込みプロセッサのマルチコア化が進んでいるが、そのマルチコア化は、汎

用のプロセッサとは異なっている面がある。汎用プロセッサでは、同質のコアを増加させるが、組み込みプロセッサでは、高性能・高消費電力なコア、低性能・低消費電力なコアといった、異なる特性のコアを組み合わせることが多い。低性能・低消費電力なコアでは、小型のリアルタイム OS (RTOS) を動作させることが多いが、Linux のような汎用 OS と比較し、RTOS は機能が制限されるという欠点があった。

このような RTOS に、基盤実行環境を適用することで、機能向上を容易にし、また並列処理の導入により、性能向上が可能になることがわかった。即ち、RTOS を基盤実行環境における PK とみなし、UK により RTOS を補完する。これにより、RTOS 内で完結する処理についてはリアルタイム性を確保しながら、複雑な処理は UK で実行することで、並列処理により性能向上が可能である。

(5) 多くのデバイスを有する実際的なシステムでのデバイス処理の並列性の実現

スマートフォンやタブレット等のモバイルデバイスに用いられている組み込みプロセッサのマルチコア化、および RTOS への並列処理の導入に関連し、デバイス処理の並列性の実現について研究を行った。

デバイス処理の並列性は、センサ等のデバイスを多く持つモバイルデバイスにおける負荷分散に有効である。モバイルデバイスでは、柔軟なデバイスとのインタラクションを可能にするためのデバイスが数多く使用されるようになってきている。また、複数デバイスが複合的に使用されることもある。従って、デバイス処理を並列化し、負荷分散を図ることは重要な課題である。そこで、モバイルデバイスにおいて、デバイス処理を機能分割し、様々なデバイスに対応可能にするフレームワークを構築した。

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)

[雑誌論文](計 2 件)

Naoto Ogawa, Shuichi Oikawa: Device driver development of an external human interaction device for the Android, International Journal of Networked and Distributed Computing, Vol. 1, No. 3, pp. 159-166, 2013, 査読有. DOI:10.2991/ijndc.2013.1.3.4
Shuichi Oikawa: Extensible RTOS Architecture for Heterogeneous Multiprocessing with Proxy Kernels, ACIS International Journal of Computer & Information Science, Vol. 13, No. 1, pp. 1-7, 2012, 査読有. <http://www.acisinternational.org/ijcis.html>

[学会発表](計 15 件)

Shuichi Oikawa: Decoupling Dynamic Resource Management Functions from RTOS for Heterogeneous Multi-Core Systems, 2nd IEEE Global Conference on Consumer Electronics (GCCE 2013), October 1-4, 2013, Makuhari Messe (Chiba, Japan).

Naoto Ogawa, Shuichi Oikawa: Implementing the device driver of external human interaction device for the Android, 12th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2013), June 16-20, 2013, Toki Messe (Niigata, Japan).

Gaku Nakagawa, Shuichi Oikawa: An Architecture of Operating System utilizing Non-volatile Main Memory and Heterogeneous Multi-Core, 12th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2013), June 16-20, 2013, Toki Messe (Niigata, Japan).

Naoto Ogawa, Shuichi Oikawa: Measuring the Effects of Human Interaction Devices for Android, 1st IEEE Global Conference on Consumer Electronics (GCCE 2012), October 4, 2012, Makuhari Messe (Chiba, Japan).

Gaku Nakagawa, Shuichi Oikawa: Consideration on a Power-Saving and Reliable Embedded System, 1st IEEE Global Conference on Consumer Electronics (GCCE 2012), October 4, 2012, Makuhari Messe (Chiba, Japan).
Shogo Saito, Shuichi Oikawa: Enabling an OS Kernel for Large Data with a SIMD Unit, 1st IEEE Global Conference on Consumer Electronics (GCCE 2012), October 4, 2012, Makuhari Messe (Chiba, Japan).

Shuichi Oikawa: Delegating the Kernel Functions to an Application Program in UV6, 2012 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC 2012), August 15, 2012, Hong Kong Polytechnic University (Hong Kong, China).

Shuichi Oikawa: Architecting RTOS for Extensibility by Heterogeneous Multiprocessing with Proxy Kernels, 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2012), August 10, 2012, Campus Plaza Kyoto (Kyoto, Japan).

Shuichi Oikawa, Gaku Nakagawa, Naoto

Ogawa, Shougo Saito: Extending RTOS Functionalities: an Approach for Embedded Heterogeneous Multi-Core Systems, 7th International Conference on Computing in the Global Information Technology (ICCGI 2012), June 26, 2012, NOVOTEL VENEZIA MESTRE CASTELLANA (VENEZIA MESTRE, ITALY).

Shogo Saito, Shuichi Oikawa: Improving Data Processing in the OS Kernel with a SIMD Unit, 2012 International Conference on Future Information Technology and Management Science & Engineering, April 12, 2012, Regal Riverside Hotel (Hong Kong).

Gaku Nakagawa, Shuichi Oikawa: Consideration on an Operating System for Heterogeneous Multi-Core Architecture, 2012 International Conference on Future Information Technology and Management Science & Engineering, April 12, 2012, Regal Riverside Hotel (Hong Kong).

Shuichi Oikawa: UV6: an OS Kernel as an Application Program without Virtualization, 27th ISCA International Conference on Computers and Their Applications (CATA-2012), March 12, 2012, Imperial Palace Hotel (Las Vegas, USA).

齋藤奨悟, 追川修一: OS カーネル内における SIMD アーキテクチャ活用の検討, 第 120 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, 2012 年 2 月 28 日, 七島信用組合 (八丈島).

中川岳, 追川修一: ヘテロジニアスマルチコアを統合管理するオペレーティングシステム構成法の検討, 第 120 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, 2012 年 2 月 28 日, 七島信用組合 (八丈島).

Shogo Saito, Shuichi Oikawa: Utilization of a SIMD unit in the OS Kernel, International Workshop on Advances in Networking and Computing, 2nd International Conference on Networking and Computing (IC-NC 2011), December 1, 2011, Osaka University (Osaka, Japan).

6. 研究組織

(1) 研究代表者

追川 修一 (OIKAWA SHUICHI)

筑波大学・システム情報系・准教授

研究者番号: 00271271