社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2009-ARC-181／2009-EMB-11 (16)
2009/1/14

組み込みマルチコアにおける非同期遠隔手続き呼び出しを用いた省電力制御

山内 宏真† 鈴木 貴久‡ 伊藤 真紀子‡

†株式会社富士通研究所システムLSI開発研究所プロセッサソリューション開発部
〒211-8588 神奈川県川崎市中原区上小田中4-1-1
E-mail: †{yamauchi.h, suzuki.takahisa, maki-ito}@jp.fujitsu.com

あらまし 組み込みシステムへの高性能と低消費電力の要求に対して、近年マルチコアの適用が進んでいる。我々は、動的負荷分散スケジューラを搭載した非対称型マルチコア(Asymmetric Multicore Processor : AMP)に対して、非同期遠隔手続き呼び出し(ARPC)によってコア間通信を行うARPCプログラミングモデルを提案している。AMPではコア毎に特性が異なるため、ソフトウェア開発が困難な問題があるが、ARPCプログラミングモデルを用いることで、小コストで高性能を実現する並列アプリケーションの開発を容易にする。本論文では、ARPCプログラミングモデルに対してクロック制御を行う手法を提案する。本手法を用いることでAMPを用いた組み込み向けシステムにおいて低消費電力化を実現することを容易にする。

キーワード 非同期遠隔手続き呼び出し、動的負荷分散、省電力制御

Power Saving with Asynchronous Remote Procedure Call for Embedded Multi-core Processor

Hiromasa YAMAUCHI† Takahisa Suzuki‡ and Makiko Ito‡

† Processor Solution Development LAB. , Fujitsu Laboratories Ltd. 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki 211-8588, Japan
E-mail: †{yamauchi.h, suzuki.takahisa, maki-ito}@jp.fujitsu.com

Abstract Recently, multi-core processor system is increasing in the embedded system for high performance and low power consumption. We propose ARPC(Asynchronous Remote Procedure Call) programming model using dynmic load balance scheduler and ARPC. There is a problem to make parallel software because of the difference of characteristic of the core in AMP. ARPC make it easy to develop it with low cost. In this paper, we propose the clock control method with ARPC programming model. This method make it easy to save the power consumption in the embedded system with AMP.

Keyword ARPC, Dynamic load balancing, Power Saving

1. はじめに

組み込みシステムへの高性能と低消費電力の要求に対して、近年マルチコアによる並列処理の適用が進んでいる。

動作周波数を向上させずに演算性能を向上させることが出来る並列処理は組み込みシステムに求められる高性能と低電力に対する効果的な解決方法の1つであると考えられる。並列性にも様々な粒度のものがあるが、VLIWなどの命令レベル並列性やSIMDなどのデータレベル並列性は4程度が限界であると言われている。更なる性能向上のために、新たな並列性のとしてタスクなどの粗粒度レベルでの並列性が注目されている。マルチコアでは各コアに処理を割り当てることで粗粒度レベル並列性を効果的に活用することが出来る。商用プロセッサとして、富士通FR-V[1], ARM MPCore[2], IBM・ソニー・東芝 CELL[3]などが開発されている。

マルチコアには大きく分けて対称なシステムであるSMP(Symmetric Multicore processor)型と、非対称なシステムであるAMP(Asymmetric Multicore Processor)型の2つがある。組み込みシステムは、汎用プロセッサの他に画像処理などを目的にしたDSP(Digital Signal Processor)やハードウェアアクセラレータなどで構成されるAMP型をしている。

一方、電力を削減する技術の研究開発も数多くされている。動作していない回路に対してクロックを遮断するクロックゲーティングや、電源を遮断することでリーク電流を削減するパワーゲーティング、電源電圧や動作周波数を処理に合わせて動的に変化させるDVFS(Dynamic Voltage and Frequency Scaling)などが挙

げられる。

組み込みマルチコアに対してこれらの電力削減技術を適用し、より高性能・低消費電力を実現したいが、現状、制御手法は確立されていない。

2. 課題

AMP型のマルチコアでは、コア毎に実行可能な処理が異なり、各コアの特性を基に処理の割り当てを行うため、コアの数や種類を変えた場合には割り当てを変更する必要がある。またプログラム中に挿入された電力制御コードに関しても変更する必要がある。マルチコア構成の変更の度にアプリケーションの変更が必要となるため、ソフトウェア生産性において大きな影響を与える問題となっている。

このようなAMP型マルチコアにおける問題に対して、我々は動的負荷分散スケジューラをつかった非同期遠隔手続き呼び出し(ARPC)による並列処理を行うARPCプログラミングモデルを提案している[4]。ARPCプログラミングモデルでは、プログラムを機能単位に分割し、マルチコア上の各コアに対してスケジューラが動的に処理を割り当てる。このプログラミングモデルを用いた場合、小さいコストで高性能を実現する並列アプリケーションを開発することが出来る。

本論文では、ARPCプログラミングモデルにおいてスケジューラによって処理を割り当てられなかったコアに対してはクロックを遮断することで、低消費電力に関しても実現するという電力制御手法を提案する。

3. 関連研究

多くの低消費電力化技術が研究されており、OSと連動した電力制御手法やパフォーマンスカウンタによって取得した情報を基にした制御手法などが挙げられる。

OSにおける電力制御規格・モジュールとしてPC向けにACPI(Advanced Configuration and Power Interface)やcpu freq、組み込みLinux向けにDPM(Dynamic Power Management)などが提案されている。ACPIやcpu freqは一定時間idleになると電源遮断を行う。またDPMはアプリケーションごとに制御ポリシーを規定し、それに基づき省電力制御を行っている。

またOSによってタイムスライス毎にCPUのidle時間を測定し、各タスクに対してidle時間が長い場合には、電圧・周波数を低下させるDVFS制御を行う技術も提案されている[5]。

しかし、AMP型のマルチコアは従来、それぞれのコアに対して分散制御を行うものと想定されており、全体を統合して制御するOSというものは想定されていない。そのため、OSを基にした制御手法を適用することは難しい。

また現在のプロセッサにはパフォーマンスカウンタなどアプリケーション実行中の各種カウンタ情報を取得する機構が備わっている。この情報を基に性能予測を行い、電力制御を行う手法が研究されている[6][7]。

これらの手法はシングルコア環境での評価が行われているが、ARPCプログラミングモデルでは動的負荷分散スケジューラが実行時に割当先を決定するため、パフォーマンスカウンタ情報によって性能予測を基に制御を行う手法とは親和性が悪い。

またマルチコア向け並列化コンパイラを用いて、タスクスケジューリング情報を基に消費電力制御を行う技術が提案されている[8]。様々な粒度の並列性を用いて、各コアに対して最適な処理の割り当てを行うことで、コア数に比例した演算性能を実現している。また、タスクスケジューリング結果を基に各タスクの演算時間の見積もり、リアルタイム制約に対して余裕のある場合にDVFS制御やパワーゲーティングなどを適用し、電力削減を行っている。しかしスタティックスケジューリング時の電力制御を想定しているため、動的スケジューラによって制御を行うARPCプログラミングモデルとは異なるモデルであるため適用は難しい。

4. ARPCプログラミングモデルにおけるクロック制御

ARPCプログラミングモデルを構成するARPC技術および動的負荷分散スケジューラについて、ARPCプログラミングモデルに対してクロック制御を適用した電力制御について説明する。

4.1. 非同期遠隔手続き呼び出し

遠隔手続き呼び出し(RPC) [9]は、通常の関数呼び出しの形式で異なるマシンで実行するための技術であり、ネットワークの分野などで広く使われてきた。

クライアント/サーバアプリケーションを実装するに、RPCを用いると、ネットワーク通信に関する部分をプログラム中に記述する必要がないため、開発が容易になる。処理があたかも一つのアドレス空間で行われているように見せるために、ネットワークに関する異なるシステム間のフォーマットにおけるデータ変換や通信エラーの検知などの全ての処理がプログラマからは見えないところで行われる。また通常に関数呼び出し型のプログラミングと親和性が高いため、関数呼び出し型言語であるC言語で書かれることが多い組み込みソフトウェアの開発者になじみやすいと考えられる。

RPCでは、実際にはローカルにあるクライアントスタブを呼び、クライアントスタブはRPCランタイムライブラリを使って、サーバスタブと通信する。サーバスタブはリモートプロシージャを呼び出し、実行が終了するとサーバスタブはクライアントスタブに実行結果を渡し、クライアントスタブはクライアントに結果を渡す。

本手法では、ARPC という通信ライブラリを用いて、図 1 に示すように複数の遠隔手続きを並列に処理する制御を行っている。

PE をフロントエンドとバックエンドに分け、フロントエンド側の PE がバックエンド側の PE に対して処理を割り当て、バックエンド側の PE での実行後、結果を回収する。

関数呼び出し型プログラムでは、動画再生・グラフィック・音声再生などの"機能"を関数として実現しシステムを構成しているが、本手法ではこれらの機能を RPC の粒度としている。これらの機能を同時に実行できる場合、コア数を増やすと、それに沿って演算性能を向上させることが可能である。

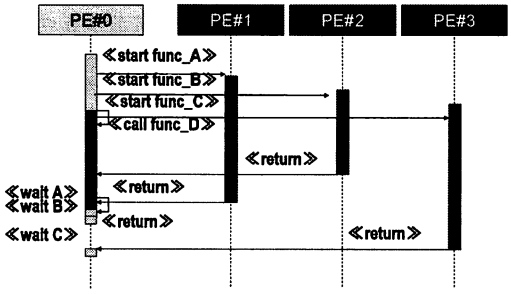


図 1非同期遠隔手続き呼び出し

4.2. 動的負荷分散スケジューラ

2 章で述べたように、AMP 型のマルチコアでは、各コアの特徴が異なるため、処理を割り当てるには、各コアに適した処理をプログラムから割り当てる必要がある。コアの数や種類が増えると人手による静的な割り当ては困難になる。この問題に対して、本手法では動的負荷分散スケジューラを用いている。図 2 に示すように、本手法では各関数に対して、どの PE(Processing Element)で実行するのかの割り振りを動的負荷分散スケジューラが決定する。機能の特性ごとにキューを用意することで、コア数や種類を変更した場合でも、アプリケーションを変更することなく、各キューの割り当て先を指定するスケジューラの一部を変更するのみでよい。

組み込みシステムにおいてハイエンドからローエンドまで様々な製品展開をする際には、コアの数が変更されることが考えられる。この際に製品ごとに機能の割り当てを変更するとなると、ソフトウェア生産性が大きく低下してしまうが、本手法を用いるとこの問題を解決することが出来る。また次世代製品の開発の際に、システム性能の向上のためにコアの種類が変更されることが考えられる。

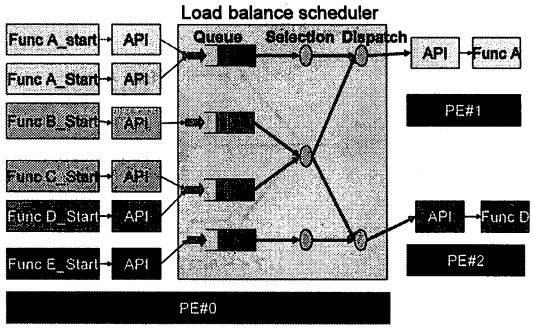


図 2 動的負荷分散スケジューラによる処理の割り当て

4.3. クロック制御を用いた電力削減

本論文で提案する ARPC プログラミングモデルにおける電力制御手法について説明する。

ARPC プログラミングモデルでは各コアへの処理の割り当てをスケジューラによって行っているが、本手法では処理の割り当てられなかったコアに対してスケジューラがクロック制御を行う機能を追加している。各 PE に対するクロック制御の処理フローを図 3 に示す。各 PE はスケジューラから割り当てられた関数の処理が終了すると、クロック停止可能状態へ遷移する。その後、スケジューラはクロック停止命令を出し、PE のクロックは停止する。その後、再度 PE に処理が割り当てられる際にクロックの供給が再開し、関数の処理を行う。

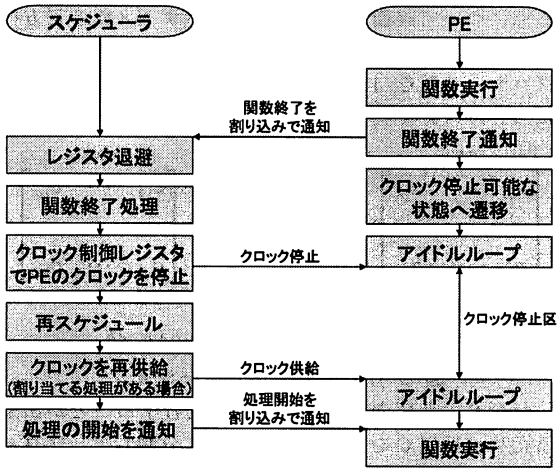


図 3 クロック制御処理フロー

このクロック制御を適用する例を図 4 に示す。2 つのリモートプロシージャ func_A, func_B があり、実行可能なプロセッサ PE#1, PE#2, PE#3 の 3 つがある場合、PE#3 には処理を割り当てられないため、クロックを停止する。

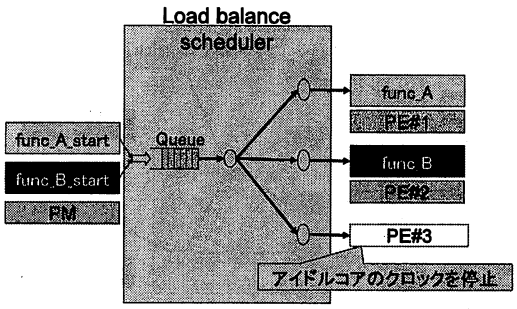


図 4 スケジューラによる電力制御

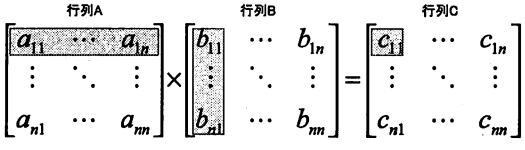
5. 評価実験

今回、マルチコアに対して提案手法を用いた際の演算性能と消費電力についての評価を行う。表 1 に示す FR1000 プロセッサに対して、本手法を用いての評価を行った。FR1000 は 1 クロックで 8 命令/28 演算を並列に実行可能である FR550 を 4 コア搭載したマルチコアプロセッサである。

表 1 FR1000 仕様

動作周波数	500MHz × 4
ピーク性能値 (整数演算 /浮動小数点演算 /Media)	8000MIPS 8GLOPS 32GOPS @500MHz
ローカルバス インターフェース	166MHz
ワーク RAM	512KB
低消費電力モード	Clock Gating

まず、提案手法がコア数に応じた性能向上を実現する点について評価を行う。図 5 に示す N×N の正方行列の掛け算を行う行列演算アプリケーションに対して、プロセッサ数を 1～4 個と変更させた場合の演算性能についての評価実験を行った。



$$C_{ij} = \sum_{k}^{n} a_{ik} \times b_{kj} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + \cdots + a_{in} \times b_{nj}$$

図 5 行列演算アプリケーション

行列 C の各要素の計算は並列に実行することが出来るため、行列 C を n 個のブロックに分割して、ブロック単位で並列処理を行っている。

n 個のブロックを各コアに割り当てることによる並列処理を行った実行結果を表 2 に示す。

表 2 行列演算実行結果

PE number	Cycle	Speedup ratio
1	1800461984	1
2	899554412	2.001504256
3	602039128	2.990606258
4	453814024	3.967400496

この結果は、本制御手法が遠隔手続き呼び出しを非同期に行うことで高い演算性能を実現できる制御手法であることを示している。

次に図 6 に示す MPEG2 デコード行いながら、複数の 3D オブジェクトを万有引力によって衝突させ、その様子を描画する動画アプリケーションに対して、クロック制御を適用した。図 7 に示すように、1 フレームごとに 3D オブジェクトの衝突計算を行い、それらの描画を重ね合わせている。図 8 に実行時の各 PE におけるガントチャートを示す。空白の部分は各コアに割り当てられた処理の回収後、次のフレームにおける処理が割り当てられるまでを示しており、クロック制御の適用が可能である。1 秒間に 30 フレームの処理を行うため、33[ms]が処理の制約時間となる。制約時間を守るように処理の回収を行っているが、MPEG2 デコードの処理は画像サイズやビットレートによって、3D 描画の処理は 3D オブジェクトの数によって負荷が変化する。網掛けの部分は、制約時間よりも前に処理が終了するためにクロック制御が適用可能な部分を示している。

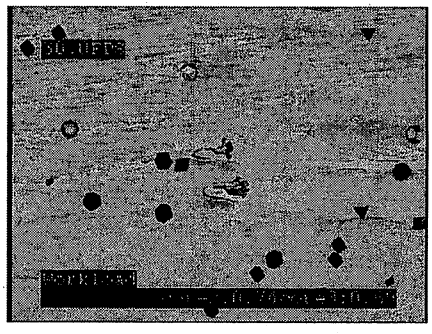


図 6 mpeg2 decoder + 3D object 描画アプリケーション

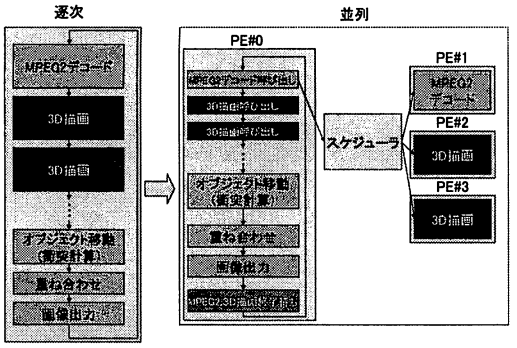


図 7 評価アプリケーションの処理フロー

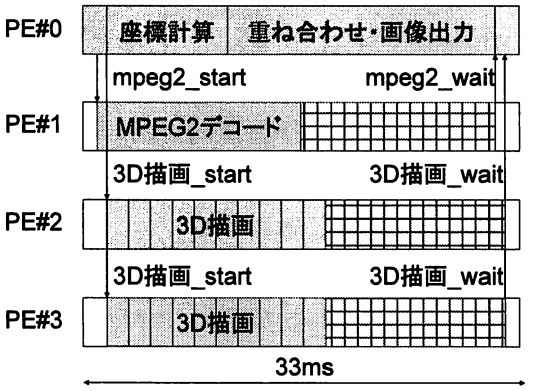


図 8 各 PE におけるガントチャート

図 9 にチップ上の消費電力を実測し、クロック制御 ON/OFF 時のそれぞれの電圧波形を示す。クロック制御を適用したときの電力が削減されることが確認できる。

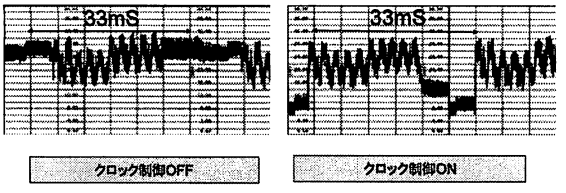


図 9 クロック制御による電力削減

本実験において、画像サイズを小さくしたり 3D オブジェクトの数を減らしたりした場合に、クロック制御適用箇所が大きくなり電力がより削減されるはずである。そのことを確認するため、3D オブジェクト数を様々な値に設定し、クロック制御を行ったときの電力を実測した。

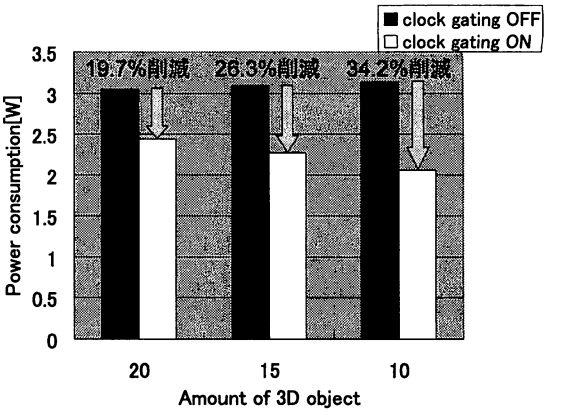


図 10 mpeg2 decode + 3D object 描画アプリケーションへの消費電力削減結果

3D オブジェクトの数を減らしていくと、3D オブジェクトの描画を行う PE#2, PE#3 の処理負荷が軽くなり、消費電力の削減効果が大きくなっていることが確認できる。

次に、この 3D オブジェクトの数を最小値 1，最大値 20 に設定し、1 フレーム処理毎にランダムに変化させ、クロック制御を適用した結果を図 11 に示す。

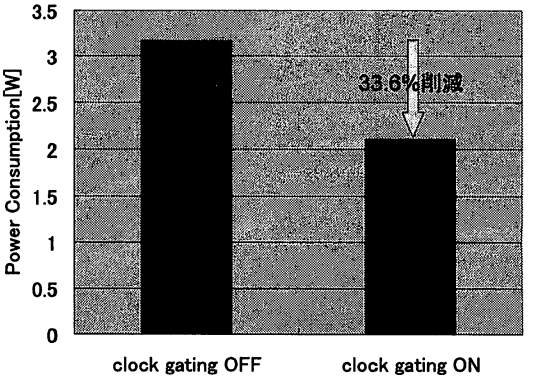


図 11 動的負荷変動アプリケーションに対する消費電力削減結果

この結果から動的に処理の負荷が変化するアプリケーションに対しても、消費電力制御が行われていることが確認できる。

6. まとめ

本論文では ARPC プログラミングに対してクロック制御を適用し、電力制御を行う手法を提案した。マルチコアにおいて、遠隔手続き呼び出し技術を並列に処理することによる、コア数に応じた性能向上を示している。また複数のアプリケーションの制御および実行時によって負荷の大きさが変わる処理に対するクロック制御を適用することで消費電力の削減に関しても効

果があることを示している。

本評価実験では、クロック制御による消費電力制御を行ったが、他の電力削減技術の適用および併用することで、より消費電力を削減できると考えられる。DVFS 制御やパワーゲーティングを用いた制御については、本手法における今後の課題である。

文　献

[1] T. Shiota et al., A 51.2GOPS, 1.0GB/s-DMA Single-Chip Multi-Processor Integrating Quadruple 8-Way VLIW Processors, ISSCC2005, ISSCC Dig. Tech.Papers, pp.18-19, 2005.

[2] Cornish, J.: "Balanced Energy Optimization", International Symposium on Low Power Electronics and Design, Newport, California, USA, 2004

[3] Pham, D. et al.: The Design and Implementation of a First-Generation CELL Processor, In Proceeding of the IEEE International Solid-State Circuits Conference, San Francisco, USA, 2005

[4] 須賀 敦浩, スケーラビリティを実現する組み込み用途向けマルチコアプラットフォーム, マイクロプロセッサ フォーラム ジャパン 2008

[5] 宮川大輔, 石川裕, "電力制御スケジューラのプロトタイプ実装", IPSJ, SigOS, Vol103, pp.109-115, 2006

[6] 浅井雅史, 池田佳路, 佐々木広, 近藤正章, 中村宏, "統計処理に基づくコンパイラ協調型 dvfs 手法", IPSJ, SigARC, Vol166, Jan.2006

[7] 金井遵, 佐々木広, 近藤正章, 中村宏, 天野英晴, 宇佐美公良, 並木美太郎, " 性能予測モデルの学習と実行時性能最適化機構を有する省電力化スケジューラ", IPSJ, SigACS, Vol49, 2008

[8] 白子準, 吉田宗弘, 押山直人, 和田康孝, 中野啓史, 鹿野裕明, 木村啓二, 笠原博徳, "マルチプロセッサにおけるコンパイラ制御低消費電力化手法", IPSJ, SigACS, Vol47, 2006