

低消費電力コンピューティングを実現するマルチコア技術

Multicore Technologies Realizing Low-power Computing

木村啓二 笠原博徳

Abstract

マルチコアプロセッサは、スマートフォン、パーソナルコンピュータ、自動車からクラウドサーバ、スーパーコンピュータに至るまで、各種の IT 機器で利用されている。これは、マルチコアでは半導体集積度の向上とともに性能向上を可能にしつつ消費電力を抑えることができるため、環境に優しい低消費電力コンピューティング、すなわちグリーンコンピューティングの実現のための最有力技術として採用されている。本稿では、この低消費電力マルチコアにおけるコンパイラを中心としたソフトウェアとハードウェアの協調及び各種組み込み応用について紹介する。

キーワード：グリーンコンピューティング、マルチコア、低消費電力、自動並列化コンパイラ

1. はじめに

コンピュータの性能が向上し、その応用分野が拡大するに伴い、我々の生活はより便利に、より豊かになっている。しかしながら、それに伴いコンピュータの消費電力とそれに伴う発熱が無視できなくなっており、これらがコンピュータを用いたシステムの構築を行う上で重要な要件となっている。すなわち、コンピュータに要求される高い計算能力を維持したままシステムの低消費電力かつ低エネルギー消費を実現するための、電力効率の高いコンピューティングが強く求められている。

例えば、データセンターやスーパーコンピュータでは、必要な計算能力を持つハードウェアを駆動するだけの電源供給設備の確保、並びに電力消費により生じる熱を冷却するだけの施設の設置が大きな課題となっている。具体的には、「京」スーパーコンピュータでは、計算機システムの総電力が最大 20 MW であり、冷却施設として 64 台の空調機を備えている⁽¹⁾。

また、スマートフォンなどのモバイル機器では、端末の大きさやバッテリーの充電間隔がシステムの利便性に与える影響は非常に大きい。そして現在、これらのシス

テムは要求性能を低消費電力で実現するために、複数の CPU コアを 1 チップ上に搭載したマルチコアで構成されている。

本稿では、筆者らが産官学連携で研究・開発した低消費電力マルチコアにより高電力効率コンピューティングを実現するためのコンパイラとハードウェアの協調技術を紹介する。

2. 低消費電力コンピューティングを実現するマルチコア電力制御技術

現在のマイクロプロセッサの消費電力は「動的消費電力」と「静的消費電力」の二つから構成される。動的消費電力 P_{dynamic} 及び静的消費電力 P_{static} は、それぞれ式 (1), (2) で表すことができる。

$$P_{\text{dynamic}} \propto afCV^2 \quad (1)$$

$$P_{\text{static}} = I_{\text{leak}}V \quad (2)$$

ここで、 a はスイッチング率、 f は動作周波数、 C は回路の負荷容量、 V は電源電圧、 I_{leak} は漏れ電流をそれぞれ表す。また一般に f と V には比例関係がある。

マイクロプロセッサの消費電力を制御するためには、式 (1), (2) の各パラメータを制御すればよい。すなわち、動作周波数と電源電圧を動的に制御する DVFS (Dynamic Voltage and Frequency Scaling)、クロック供給を停止するクロックゲーティング、及び電力供給を停

木村啓二 正員 早稲田大学理工学術院情報理工学科
E-mail keiji@waseda.jp
笠原博徳 正員 早稲田大学理工学術院情報理工学科
E-mail kasahara@waseda.jp
Keiji KIMURA and Hironori KASAHARA, Members (Faculty of Science and Engineering, Waseda University, Tokyo, 162-0042 Japan).
電子情報通信学会誌 Vol.97 No.2 pp.133-139 2014 年 2 月
©電子情報通信学会 2014

止するパワーゲーティングによって制御することができる。これらの制御により、要求性能を満たす範囲で低周波数かつ低電圧動作を行い、使用しないモジュールのクロック供給あるいは電源供給を停止することで消費電力を削減することができる。

ここで、マルチコアの並列処理による高電力効率なプログラム実行を考える。

今、あるプログラムが4コアのマルチコアによる並列処理で1コアの逐次実行に対して3.5倍の速度向上を得ることができるとする。このプログラムの要求性能が1コア実行時の実行時間だとすると、4コア動作時では1/3.5のスピード、すなわち1/3.5のクロック周波数で実行してよいことになる。前述のとおり、クロック周波数と電源電圧は比例するので、式(1)から $4 \times (1/3.5)^3 \approx 0.09$ となり、約1/10の動的消費電力で逐次実行と同等の処理ができることになる。

また、要求性能を最高性能としたときであっても、一般に並列処理では同期待ちの時間が存在し、これらの時間に対してクロックゲーティングやパワーゲーティングを適用することにより、並列処理による性能向上を維持したまま消費電力を削減することができる。

しかしながら、これらの電力状態制御には時間的及び消費エネルギー的なオーバーヘッドが存在する。また、ハードウェアの実装状況により、どの制御がどのハードウェアモジュール単位に対して適用可能かが異なる。例えば、現在のマルチコアプロセッサではコアごとの周波数制御が可能なものは存在するが、コアごとの電源電圧の制御が可能なものは少ない。そのため、制御時にはこれらのオーバーヘッドや制約条件を考慮する必要がある。

一方、これらのオーバーヘッド削減や制約緩和のためのハードウェアの研究及び開発も進められている。例えば、電源供給を行うレギュレータをチップ上に搭載することで、電圧制御オーバーヘッドの削減やコアごとの電圧制御が可能となる⁽²⁾。これらの技術により、より柔軟かつ効率の良い電力制御が可能となる。

3. 低消費電力コンピューティングのためのマルチコアソフトウェア技術

本章では、グリーンコンピューティングのためのマルチコアソフトウェア技術として、筆者らが開発しているOSCAR自動並列化コンパイラとOSCAR APIによる、マルチコア上での自動並列化・自動電力制御を紹介する。

OSCARコンパイラはCあるいはFortranプログラムを入力とし、これを自動並列化する。従来の並列化コンパイラがグループのみを並列化対象としていたのに対し、OSCARコンパイラはループ並列処理に加えて、ループやサブルーチン呼出し間の並列性を利用する粗粒度タスク並列処理、ステートメント間の並列性を利用する近細粒度並列処理を階層的に組み合わせてプログラム全域から並列性を抽出するマルチグレイン並列処理を特徴とする。また、ループ間の依存関係を考慮しつつ複数のループをキャッシュやローカルメモリのサイズを考慮しながら分割し、CPUコア近接メモリ上のデータ再利用性を抽出するデータローカリティ最適化、CPUとアクセラレータを混載したヘテロジニアスマルチコアに対する最適化、及び電力最適化の機能を有する。

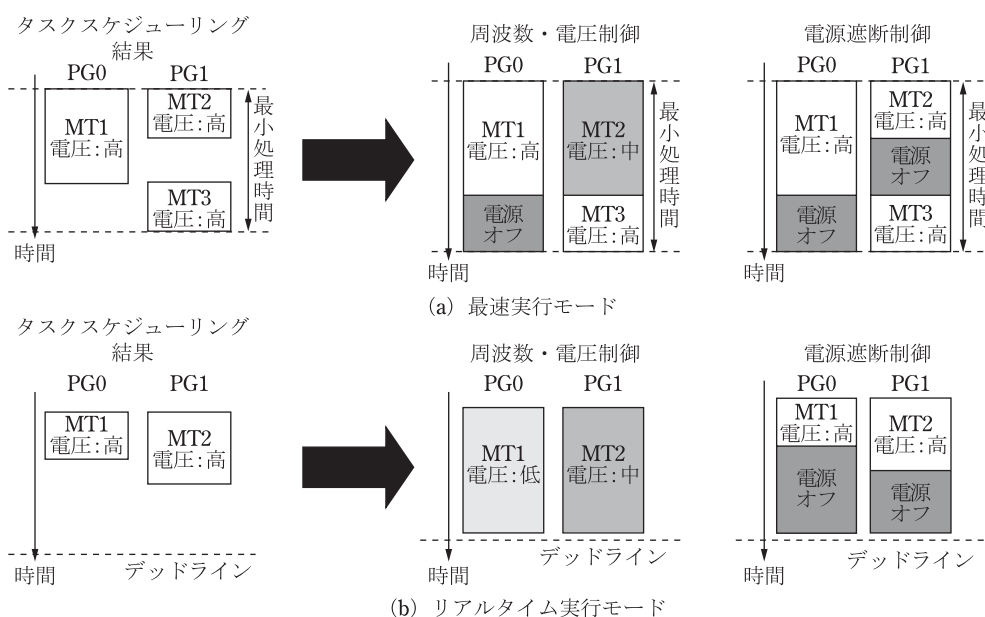


図1 OSCARコンパイラによる電力最適化 粗粒度タスクのスケジューリング結果に基づき、DVFS、クロックゲーティング、パワーゲーティングを適用する。

図1に OSCAR コンパイラによる電力最適化の概要を示す⁽³⁾。

最速実行モードは、実行時間最短の状態です電力を削減する。図1(a)ではコンパイラによる並列化により、タスク1 (MT1) がコア0 (PG0) に、タスク2, 3 (MT2, MT3) がコア1 (PG1) に割り当てられており、MT3はMT1とMT2の実行終了を待つようスケジューリングされた状態を表す。このとき、MT2の終了時刻がMT1よりも早い場合、PG1に空き時間が生じる。また、MT1の後にPG0に割り当てられているタスクがないため、ここにも空き時間が生じる。このような空き時間に対して、電力状態のオーバヘッドを考慮した上で、DVFS、クロックゲーティングあるいはパワーゲーティングを適用することにより、プログラムの消費エネルギーを最小化する。

リアルタイム実行モードは、あらかじめ設定されたデッドラインに間に合う範囲で電力を削減する。図1(b)では、MT1とMT2の終了後に、PG0、PG1にデッドライン時刻までにそれぞれ空き時間が生じる。このような空き時間に対して最速実行モードと同様に電力制御を行い、プログラムの消費エネルギーを最小化する。

これらの OSCAR コンパイラによる最適化を様々なマルチコアプラットフォームに適用するため、筆者らは産学協同で OSCAR API を開発した。OSCAR API は共有メモリ形マルチコアをベースに、ローカルメモリや分散共有メモリなどのメモリ構成や電力制御、ヘテロジニアスマルチコアなどの様々なマルチコアアーキテクチャに対

応している。OSCAR API は OpenMP からスレッド生成等の四つのコンパイラ指示文に、上記様々なアーキテクチャを利用するためのコンパイラ指示文を加えて構成されている^{(4),(5)}。OSCAR API は OSCAR コンパイラとターゲットマルチコア用ネイティブバックエンドの間のインタフェースとして利用される。

図2に OSCAR コンパイラと OSCAR API によるコンパイルフローを示す。まず、Fortran 若しくは C で記述されたプログラムを OSCAR コンパイラの入力とする。OSCAR コンパイラは前述の最適化を行った後、OSCAR API の指示文入りの並列化された Fortran 若しくは C プログラムを生成する。この生成コード中の OSCAR API 指示文は、OSCAR API 解釈系によりターゲットマルチコア用のランタイムライブラリに変換され、gcc 等のターゲットマルチコア用バックエンドコンパイラにより並列実行可能なバイナリが生成される。

図3に OSCAR コンパイラが生成する OSCAR API 指示文が挿入された電力最適化コードのイメージを示す。図では、コンパイラによるスケジューリングにより、コア0 (VC0) にタスク1, 3 (MT1, MT3)、コア1 (VC1) にタスク2, 4 (MT2, MT4) がそれぞれ割り当てられており、コア1はMT2終了後に空き時間があるためクロック停止状態 (sleep) となっている。OSCAR コンパイラによるコード生成の結果、MT2終了後に fvcontrol 指示文が挿入される。この指示文の第1引数は制御対象コア (省略時は自コア)、第2引数は制御対象モジュールを表し、第3引数はこの指示文で設定す

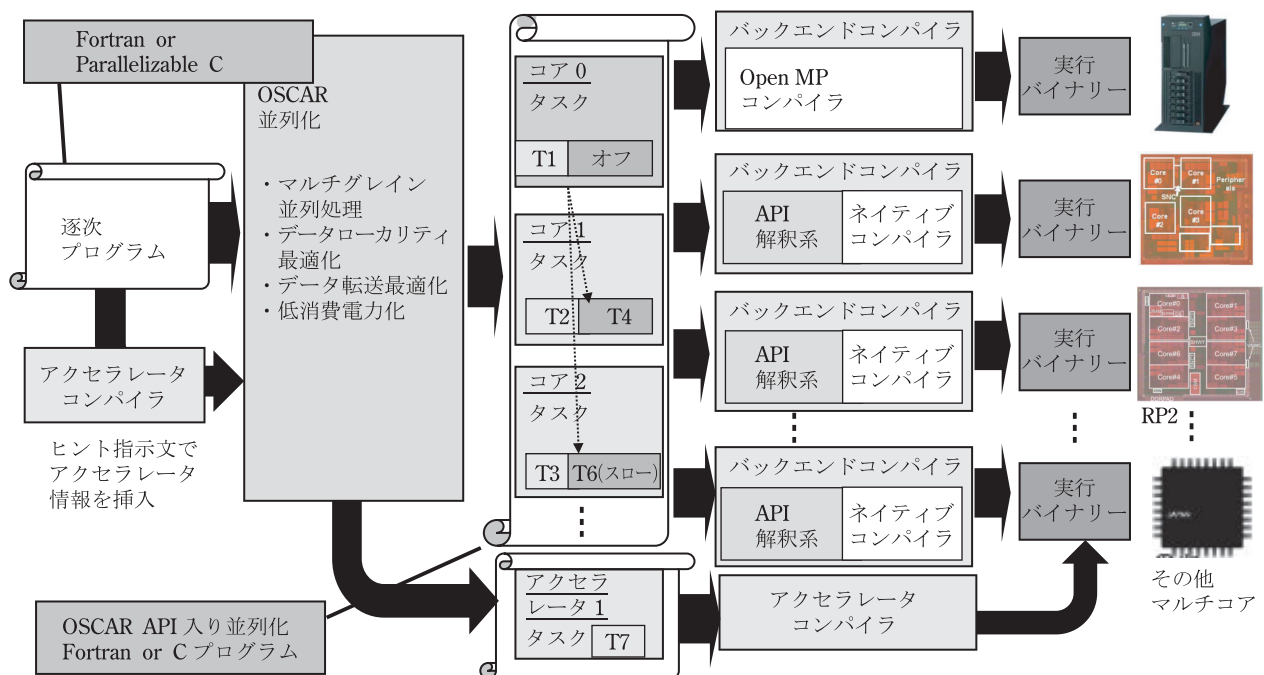


図2 OSCAR コンパイラと OSCAR API のコンパイルフロー 逐次プログラムを OSCAR コンパイラに入力すると OSCAR API 入りの並列化プログラムが生成される。その後、バックエンドコンパイラが実行バイナリを生成する。

る電力状態をそれぞれ表す。ここで、電力状態は、0 より大きな値の場合は、最大周波数を百分率で表した動作周波数、0 はクロックゲーティング、-1 はパワーゲーティングを表す。この例では、コア 1 の CPU のクロックゲーティングを指示している。その後、コア 0 の MT1 が終了後に fvcontrol 指示文により、コア 1 の CPU を最大周波数で動作するよう指示している。

4. 低消費電力コンピューティング マルチコア技術の実例

本章では、これまでに述べた OSCAR コンパイラによる並列化及び低消費電力化の適用例を紹介する。ターゲットのマルチコアとして低消費電力組込みマルチコア RP2 及び RP-X を用いた。

4.1 組込み低消費電力マルチコア RP2 及び RP-X

RP2 は経済産業省/NEDO プロジェクト「リアルタイム情報家電用マルチコア」により産官学連携で開発された、SH4A コアを 8 コア搭載したホモジニアスなマルチコアである⁽⁶⁾。電力制御の機能としては、コアごとの周波数制御、クロックゲーティング、及びパワーゲーティングが可能である。ただし、電源電圧はチップ一括で制御する必要がある。

RP-X は、経済産業省/NEDO プロジェクト「情報家電用ヘテロジニアスマルチコア」により産官学連携で開発された、8 コアの SH4A に加え、FE-GA、MX2、及び VPU5 の 3 種のアクセラレータを搭載した 15 コアのヘテロジニアスマルチコアである⁽⁷⁾。本章では、3 種のアクセラレータのうち、動的再構成可能アクセラレータである FE-GA を 4 コア使用した例を紹介する。図 4 に

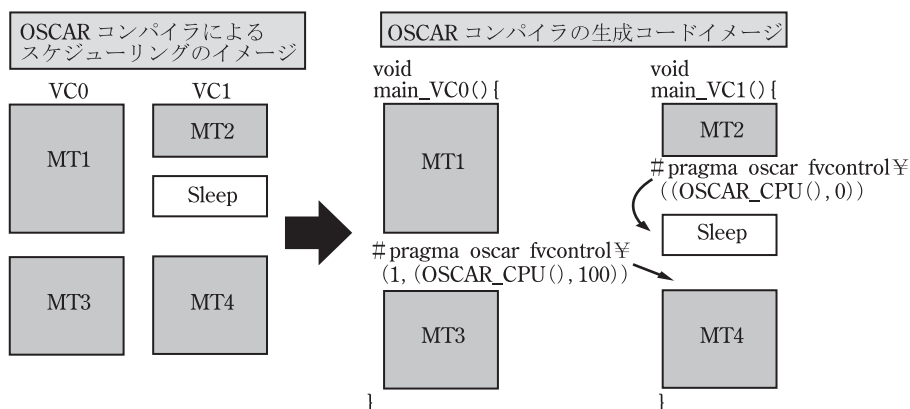


図 3 OSCAR コンパイラが生成する OSCAR API 入り並列化コードのイメージ fvcontrol 指示文により電力制御が行われている。

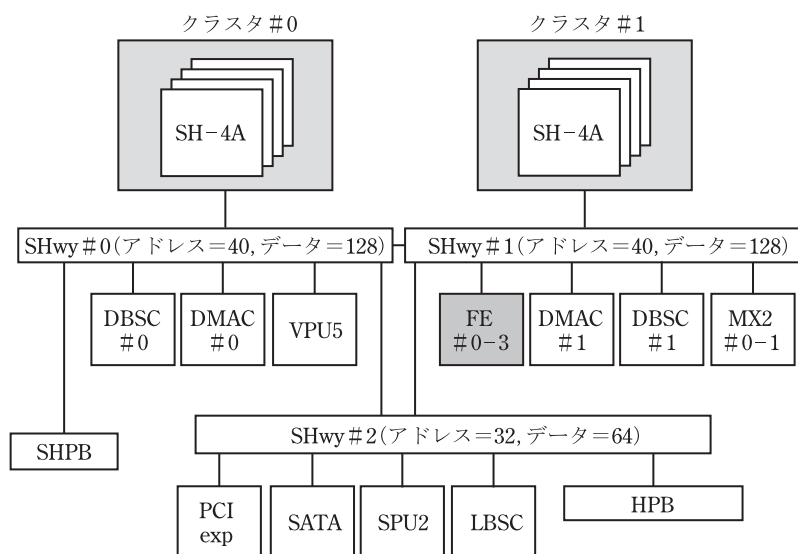


図 4 RP-X のブロック図 SH4A8 コアと FE-GA を含めたアクセラレータを 7 コア搭載する、計 15 コアの低消費電力ヘテロジニアスマルチコア。

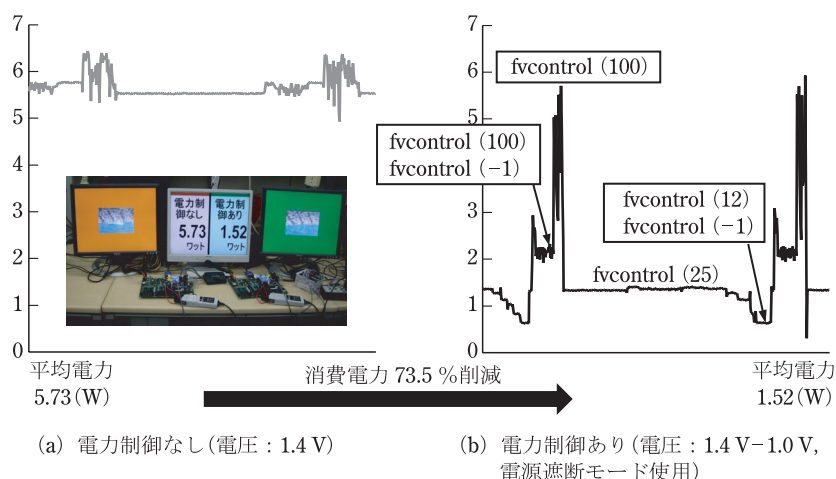


図5 RP2上でのMPEG-2動画画像再生並列実行時の電力波形 コンパイラによる電力制御により73.5%の電力削減。

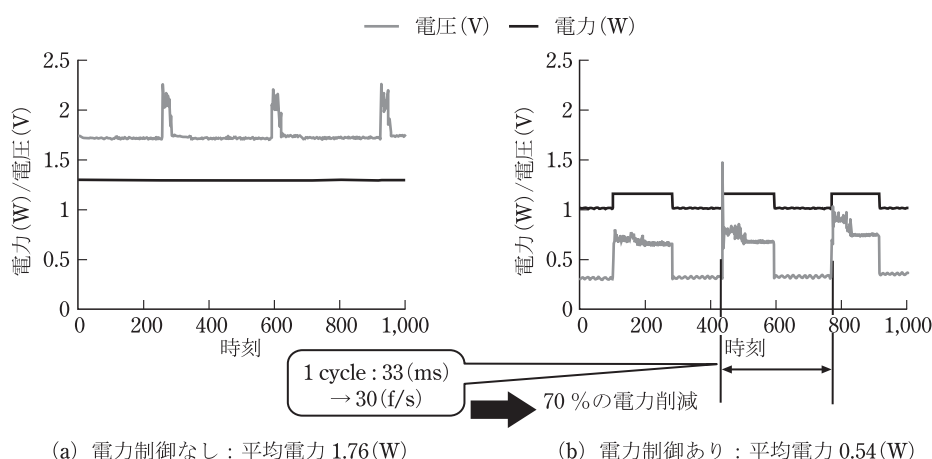


図6 RP-X上でのオプティカルフロー並列実行時の電力波形 ヘテロジニアスマルチコアに対するコンパイラによる電力制御により70%の電力削減。

RP-Xのブロック図を示す。

4.2 コンパイラによる低消費電力化

MPEG-2 動画画像再生を OSCAR コンパイラで RP2 の 8 コア用に並列化し、実行したときの電力波形のスナップショットを図5に示す。図(a)がコンパイラによる電力制御なし、図(b)が電力制御ありをそれぞれ示す。いずれの場合も、秒間 30 フレームの動画画像を再生しており、ピーク間が 1 画像再生のためのデッドライン時間である 33 (ms) である。電力制御なしの場合は、1 フレームの再生中、常にほぼ一定の電力を消費していることが分かる。一方、電力制御ありの場合は、1 フレームの再生中、そのときに実行している処理とデッドラインまでの残り時間に応じて、細かく電力制御を行う様子が分かる。電力制御の結果、73.5% の電力削減効果を得ることができた。

図6は秒間 30 フレームの動画画像に対するオプティカ

ルフロープログラム（物体追跡プログラム）を OSCAR コンパイラで RP-X の SH4A を 8 コア及び FE-GA を 4 コアの計 12 コア用に並列化し実行したときの電力波形を示す。図5と同様に(a)が電力制御なし、(b)が電力制御ありをそれぞれ表す。図6から、ヘテロジニアスマルチコアにおいても、コンパイラによる電力制御により70%の電力削減効果が得られていることが分かる。

更に、本技術を Android プラットホームに対して適用しその評価を行った。評価には、4 コアの ARM Cortex A9 を持つ Samsung Exynos4412 を搭載した評価ボード Odroid-X2 を用いた。本ボードに Android4.1.2 を搭載し、更に筆者らは低オーバヘッドでクロックゲーティングができるように Linux カーネルを修正した。本環境上で並列化及び低消費電力最適化された MPEG-2 動画画像再生プログラムを実行した。

図7に4コア使用時の電力波形を示す。(a)が電力制御なし、(b)が電力制御ありをそれぞれ示す。図7か

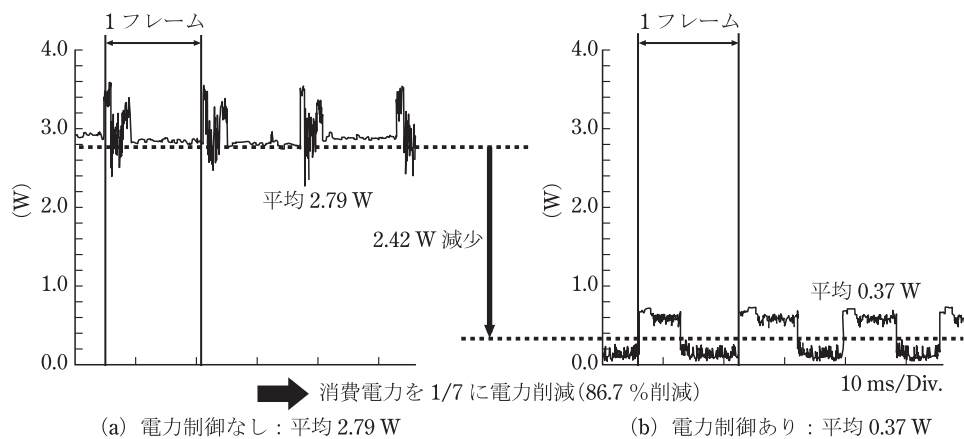


図 7 Android プラットホーム上での電力最適化 ARM Cortex A9 を 4 コア使用時に 86.7% の電力削減。



図 8 RP-X による Web サーバ 低消費電力マルチコアによる 1 W Web サーバを実運用中。



図 9 早稲田大学グリーンコンピューティングシステム研究開発センター (GCS) 外観

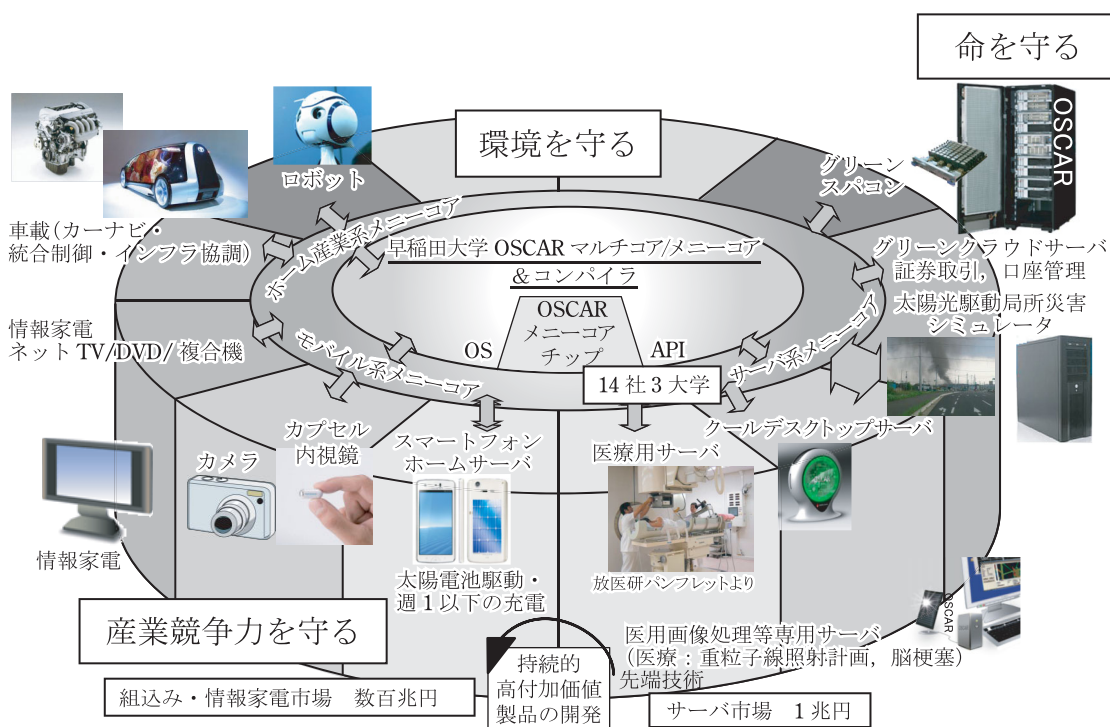


図 10 グリーンコンピューティングシステム研究開発センター (GCS) における産学連携研究

ら、Android プラットホーム上でも 86.7% の電力削減効果が得られることが確認できた⁽⁸⁾。

4.3 低消費電力 Web サーバ

RP2 や RP-X のような低消費電力マルチコアを利用して、低消費電力サーバを構築することも可能である。現在、早稲田大学笠原研究室の Web サーバは RP-X に Linux を搭載したシステムにより運用している⁽⁹⁾。研究室のページをアクセスすると、図 8 のようにアクセス時の消費電力が表示される。

5. 産官学連携研究開発拠点グリーンコンピューティングシステム研究開発センター

上記のような低消費電力マルチコア／メニーコアのハードウェア、コンパイラを中心としたソフトウェア、応用技術を産官学連携で研究開発及び人材育成を行うため、早稲田大学では、経済産業省「2009 年度先端イノベーション拠点整備事業」の支援を受け、2011 年 5 月に図 9 に示すグリーンコンピューティングシステム研究開発センターを開設した。

このセンターでは、図 10 に示すような各種製品化応用を産官学連携で研究開発することを目指しており、屋上の太陽光発電装置で駆動されるソーラーパワードサーバ（災害・病気から生命を守る）の開発、低消費電力・高性能・低ソフトウェア開発コスト・短開発期間による付加価値の高い産業製品（自動車、スマートフォン、カメラ、内視鏡、マルチコアプロセッサ、コンパイラ、並列化チューニングツール、OS、基地局、複合機）の創出（産業競争力を守る）、IT 機器の低消費電力化による環境負荷の低減（環境を守る）を目標に、富士通(株)、(株)日立製作所、日本電気(株)、ルネサスエレクトロニクス(株)、オリンパス(株)、トヨタ自動車(株)、(株)デンソー、理想科学工業(株)、イーソル(株)、(株)オスカートテクノロジー（順不同）等の企業と共同研究を行っている。

6. ま と め

本稿では、筆者らが取り組んでいる低消費電力コンピューティングを実現する低消費電力かつ高電力効率なマルチコア技術について解説した。

本技術は、コンパイラによる自動並列化及び自動電力最適化技術から構成されている。更に、本技術を低消費電力マルチコアに適用した例も提示し、コンパイラとマルチコアの協調動作により、リアルタイム動画像再生プログラム及び動画像処理プログラムにおいて約 7 割の電力削減が可能であることを示した。

現在、筆者らは産学連携により様々な実アプリケー

ションや商用プラットフォーム上で本技術の実証実験を行っている。

文 献

- (1) <http://www.aics.riken.jp/jp/k/facility.html>
- (2) W. Kim, M.S. Gupta, G.-Y. Wei, and D.M. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," IEEE 14th International Symposium on High Performance Computer Architecture (HPCA 2014), pp. 123-134, Salt Lake City, USA, Feb., 2008.
- (3) J. Shirako, M. Yoshida, N. Oshiyama, Y. Wada, H. Nakano, H. Shikano, K. Kimura, and H. Kasahara, "Performance evaluation of compiler controlled power saving scheme," Lect. Notes Comput. Sci., vol. 4759, pp. 480-493, Springer, Jan. 2008.
- (4) K. Kimura, M. Mase, H. Mikami, T. Miyamoto, J. Shirako, and H. Kasahara, "OSCAR API for real-time low-power multicores and its performance on multicores and SMP servers," Lect. Notes Comput. Sci., vol. 5898, pp. 188-202, Springer, 2010.
- (5) A. Hayashi, Y. Wada, T. Watanabe, T. Sekiguchi, M. Mase, J. Shirako, K. Kimura, and H. Kasahara, "Parallelizing compiler framework and API for power reduction and software productivity, of real-time heterogeneous multicores," Lect. Notes Comput. Sci., vol. 6548, pp. 184-198, Springer, 2011.
- (6) M. Ito, T. Hattori, Y. Yoshida, K. Hayase, T. Hayashi, O. Nishii, Y. Yasu, A. Hasegawa, M. Takada, M. Ito, H. Mizuno, K. Uchiyama, T. Odaka, J. Shirako, M. Mase, K. Kimura, and H. Kasahara, "An 8640 MIPS SoC with independent power-off control of 8 CPU and 8 RAMS by an automatic parallelizing compiler," IEEE International Solid State Circuits Conference (ISSCC2008), pp. 90-598, San Francisco, USA, Feb. 2008.
- (7) Y. Yuyama, M. Ito, Y. Kiyoshige, Y. Nitta, S. Matsui, O. Nishii, A. Hasegawa, M. Ishikawa, T. Yamada, J. Miyakoshi, K. Terada, T. Nojiri, M. Satoh, H. Mizuno, K. Uchiyama, Y. Wada, K. Kimura, H. Kasahara, and H. Maejima, "A 45 nm 37.3GOPS/W heterogeneous multi-core SoC," IEEE International Solid State Circuits Conference (ISSCC 2010), pp. 100-101, San Francisco, USA, Feb. 2010.
- (8) H. Yamamoto, T. Hirano, K. Muto, H. Mikami, T. Goto, D. Hillenbrand, M. Takamura, K. Kimura, and H. Kasahara, "OSCAR compiler controlled multicore power reduction on android platform," Proc. of the 26th International Workshop on Languages and Compilers for Parallel Computing (LCP2013), Sept. 2013.
- (9) <http://www.kasahara.cs.waseda.ac.jp/>

(平成 25 年 10 月 23 日受付 平成 25 年 11 月 8 日最終受付)



木村 啓二 (正員)

平 13 早大大学院理工学研究科電気工学専攻博士課程了、平 11 早大・理工・助手、平 16 同大学・理工・専任講師、平 17 同助教授、平 24 教授、現在に至る。マルチコアプロセッサのアーキテクチャ、コンパイラ、アプリケーションに関する研究に従事。



笠原 博徳 (正員)

昭 55 早大・理工・電気卒。昭 60 同博士了。工博。昭 61 同大学・理工・専任講師、平 9 同教授。現在、情報理工・教授。昭 62 IFAC World Congress Young Author Prize, 平 9 情報処井記念特別賞、平 22 IEEE Computer Society Golden Core Member Award 各受賞。IEEE Computer Society 理事、マルチコア STC 委員長。経産省 NEDO マルチコア及びコンパイラ等の PL、文科省情報科学技術委員等歴任。