# グリーンコンピューティングのための低消費電力マルチコア技術

早稲田大学 理工学術院 基幹理工学部 情報理工学科
IEEE Computer Society 理事　　笠原　博徳

抄録

本稿では、グリーンコンピューティングのための低消費電力マルチコアプロセッサのアーキテクチャ、ソフトウェア、応用について紹介する。マルチコアは、半導体の集積度向上と共に問題となる消費電力を低く抑えつつ、処理性能を向上させるための技術としてスマートフォン、ゲームからPC、クラウドサーバ、スーパーコンピュータに至る多くのIT機器で採用されている。このようなマルチコアは、各プロセッサコアの動作周波数を低く抑えることにより消費電力を低く抑え、プロセッサを複数並列動作させることにより性能向上を目指そうとするものであり、複数のプロセッサを効率良く動作させることができる並列ソフトウェアの開発がキーとなる。ここではこの並列ソフトウェアを短期間・低コストで開発するための自動並列化コンパイラとその電力削減方式、生成した並列プログラムをマルチプラットフォームで実行するためのAPIとそれらの応用事例も紹介する。

## 1. はじめに

マルチコアプロセッサは、従来主流であった動作周波数の向上によるプロセッサの処理性能向上が、消費電力と冷却の面から困難になったため、消費電力を抑えつつ処理性能を向上させる方式として注目を集めている。現在、マルチコアは携帯電話、スマートフォン、ゲーム、カーナビからパーソナルコンピュータ、クラウドサーバ、スーパーコンピュータまでの多くの情報機器を構成する主要技術となりつつある。

しかし、このマルチコアによる低消費電力化では、周波数に比例し、動作電圧の二乗に比例して増大する消費電力を、周波数と電圧を低く抑えることにより下げようとするものであるため、1プロセッサコア当たりの処理性能の低下は避けられない。この性能低下を補いつつさらに性能を向上させるため、チップ上に複数のプロセッサを集積し、その並列動作によりチップ全体の処理性能を向上させようとするものである。マルチコア上で従来よりも高い性能を得るためには、ソフトウェアの並列化、すなわちプログラムからの並列性の抽出と、並列に動作可能なプログラム部分のプロセッサコアへの効果的な割り当てが必須となる。しかしこのプログラムの並列化作業は難しく、熟練したプログラマでも1アプリケーションの並列化に数週間から数ヶ月を要する場合がある。この問題を解決するためには、逐次プログラムを自動的に並列化する自動並列化コンパイラが必要となる。この目的のために筆者等はOSCAR自動並列化コンパイラを開発しており、このコンパイラでは並列化と共に、各プロセッサの周波数及び電圧の制御と、リーク電力を抑えるため使用されていないプロセッサコアの電源遮断（パワーゲーティング）を行う自動電力削減機能も世界で始めて実現している。さらにこのコンパイラにより並列化及び電力最適化されたプログラムを、異なるメーカで開発された種々のマルチコア、さらに集積度を増やしたメニーコアプロセッサあるいは共有メモリ型サーバで動作させることを可能とするAPI（Application Programming Interface）も、NEDOプロジェクトなどの支援により国内企業と共に開発を行ったのでこれについても紹介する。また、アーキテクチャとコンパイラの協調、OSCARコンパイラの自動並列化及び電力削減、APIによるソフトウェアポータビリティを確認するために、情報家電で要求されるマルチメディアアプリケーション、自動車におけるエンジン制御、医療における重粒子線ガン治療装置、地震波動伝搬計算を始めとした多くの科学技術計算への適用とその評価が行われているため、そのうちのいくつかについて紹介する。

## 2. グリーンコンピューティングシステム研究

早稲田大学では、図1に示すように、経済産業省「2009年度産業技術研究開発施設整備費補助金」先端イノベーション拠点整備事業の補助を受け、東京メトロ東西線早稲田駅横に2011年5月に環境に優しい低消費電力マルチコア／メニーコアプロセッサを研究開発するための研究所、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。ここでは、産官学連携で、太陽電池で駆動可能な冷却ファン不要の超低消費電力マルチコア／メニーコア

**経済産業省**
**「2009年度産業技術研究開発施設整備費補助金」**
**先端イノベーション拠点整備事業**

**〈目標〉**

太陽電池で駆動可能で冷却ファンが不要な超低消費電力・高性能マルチコア／メニーコアプロセッサ*のハードウェア，ソフトウェア，応用技術の研究開発

*1チップ上に多数のプロセッサコアを集積する次世代マルチコアプロセッサ

**〈産学連携〉**

日立，富士通，ルネサス，NEC，トヨタ，デンソー，オリンパス，三菱電機（重粒子線ガン治療）等

**〈波及効果〉**

超低消費電力メニーコア
- $CO_2$排出量削減
- サーバ国際競争力強化
- 我が国の産業利益を支える情報家電，自動車等の高付加価値化

**2011グッドデザイン賞受賞**

図1　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概要
2011年4月13日竣工，2011年5月13日開所（記念シンポジウム）

プロセッサのハードウェア、ソフトウェア、応用技術の研究開発を行っている。初年度は、NEC、オリンパス、デンソー、トヨタ、日立、富士通、ルネサスエレクトロニクス（五十音順）が連携研究室をセンターに設置し、また三菱電機は研究室を設置しない形で、密な共同研究を開始している。屋上にはサーバ給電用太陽光発電装置が設置されており、発生電力をサーバ室に直接給電できると共に見える化を行い、太陽光電力でサーバを動作させるための種々のデータの取得を行っている。サーバ室は、コンパイラ及び応用技術等のソフトウェア研究開発用に市販の共有メモリ型マルチプロセッササーバを設置するスペースと、研究開発するマルチコア／メニーコアとそれをベースとしたサーバを設置するスペースが用意されており、研究開発用スペースには直流給電、電池なども設置されている。図2は、見える化の一例で、2012年4月2日の太陽光発電量（赤バー）と商用サーバ日立SR16000（8コアマルチコアPower7 ベース128コアSMP）と富士通M9000（4コアSparc64 Ⅶベース256コアSMP）の消費電力（青バー）を示す。太陽光サーバ給電装置は最大40KWhの発電能力であるが、新宿地区の4月2日は32.5KWhがピークであり、サーバは常時25KWhを消費していることが分かる。昨年春からの集計では太陽光発電は平均して7KWh程度の発電量であったので、既存の上記サーバ2台を太陽光で動作させるためには消費電力を1／4以下に抑える必要があることが分かる。本研究のチップ消費電力の目標は消費電力1／100以下であるので、このようなサーバができれば200TFLOPS以上の能力をもったサーバを総量的には太陽光電力で駆動できる可能性がある。

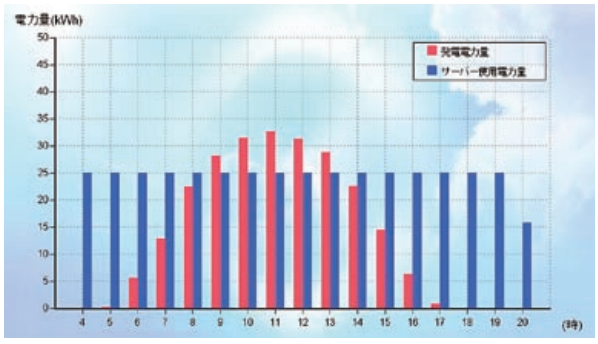



図2　電力量の1日の変化
グリーンコンピューティングシステム研究開発センター
2012.4.2（晴れ）太陽光電力とサーバ消費電力

また、研究開発するマルチコア／メニーコアチップの応用としては、図3に示すような分野を想定している。図左上から見てみると、次世代自動車のエンジン制御、カメラ画像からの歩行者認識、他車認識などの情報系とエンジン、ブレーキ制御などの制御系を統合しより安全、快適、低燃費の自動車（電気自動車を含め）の開発、より解像度が高く操作性の良い次世代カメラ、充電が1週間に1度で良く太陽光充電も可能な低消費電力スマートフォン、冷却ファンがなくホコリがたたず静音で手術室でも使用可能な医療サーバ、太陽光等再生可能電力で一部をまかなえるクラウドサーバ、地震計からの揺れを感知するとスーパーリアルタイムで各地域の津波の高さを推定し避難誘導を可能とするスーパーリアルタイムスパコンなどを想定している。低消費電力コンピューティングにより環境負荷を軽減し、低消費電力、高ソフトウェア生産性で付加価値の高い


図3　産官学連携研究開発・実用化（波及効果）

産業製品を創出し産業競争力を守り、低消費電力高性能で病気・災害から生命を守ることができればと考え当該分野の産官学連携研究を推進している。

## 3. ソフトウェア協調型マルチコアプロセッサ [1]－[6]

マルチコアプロセッサの開発においては、チップ上に集積するプロセッサ数が、パーソナルコンピュータあるいはサーバ用のインテル、AMD、IBM、富士通のチップが16プロセッサ程度まで、また組込用のARM、ルネサスエレクトロニクスからの低消費電力マルチコアはホモジニアスマルチコアで8コア程度と、集積コア数が多くないため、ほとんどがSMP（Symmetric Multi-Processor：主記憶共有型マルチプロセッサ）方式のアーキテクチャとなっている。また32コア以上集積したメニーコアでも、Tilela社の64コアあるいは100コア集積のチップはL2キャッシュ共有のSMP方式となっており、Intelの48コア集積のSCCは分散メモリアーキテクチャで分散メモリ型アーキテクチャも出始めている。ただし、SMPは自動並列化コンパイラが利用可能であり比較的並列化が容易なOpenMPを使用できるが、分散メモリの場合にはMPIを用いた人手によるプログラム並列化が前提となる。

また、従来のマルチプロセッサの開発では、ハードウェアの開発特にクロック周波数の高いプロセッサとそれを接続したアーキテクチャ設計が主流で、ソフトウェアは後からそのハードウェアをうまく使いこなすように開発して欲しいというスタイルが一般的であった。しかしマルチプロセッサでは、メモリ階層利用の最適化、プロセッサ間通信及び同期オーバーヘッドの最小化等が難しく、常に与えられたハードウェアを高速で動作させるプログラムを短期間で開発できるとは限らない。このような経験から、筆者らのグループは1980年代中旬より、並列化コンパイラによる階層的並列化、メモリ最適化、プロセッサ間データ転送及び同期オーバーヘッドを最小化しやすいマルチプロセッサアーキテクチャ、すなわちソフトウェア・ハードウェア協調型のマルチプロセッサアーキテクチャOSCAR（Optimally Scheduled Advanced Multiprocessor）アーキテクチャ（図4参照）を開発している。

OSCARマルチコアアーキテクチャでは、各プロセッサにより共有されるオフチップ及びオンチップの集中型の共有メモリと、サーバ用SMPのように各プロセッサ用プライベートキャッシュを持たせることができると共に、我が国が得意とするハードリアルタイム制御にも対応できるように、自分しかアクセスしないデータを格納するローカルデータメモリと他のプロセッサからも直接アクセスできる分散共有メモリ、またそれらのプロセッサ近接メモリにアクセスが必要となる以前に集中共有メモリからデータをロードしたり、他プロセッサが必要とする共有データを当該プロセッサ上の分散共有メモリにストアしたりするためのデータ転送ユニットDTU（高機能DMAコントローラ）を持たせている所に特徴がある。さらにOSCARマルチコアアーキテクチャでは、各プロセッサコア及びコア内のプロセッサ、DTU、各種メモリとメモリ内バンクなどを別々に周波数制御及びクロック停止（クロックゲーティング）、電源遮断（パワーゲーティング）するための周波数電圧制御レジスタFVRも用意されている[6]。2005年から2007

年度にかけて筆者がプロジェクトリーダとして行ったNEDO" リアルタイム情報家電用マルチコア "プロジェクトでは、委員会参加のIT／半導体企業6社、日立、富士通、ルネサス、東芝、パナソニック、NEC（順不同）と共に、このOSCARアーキテクチャをマルチコア用並列API（Application Programming Interface）の標準アーキテクチャと定め、APIの開発を行った。さらに、この標準アーキテクチャに基づき、図5、図6に示すように、SH4Aコアを90nmテクノロジーで9ミリ角のチップ上に8コア集積したRP2を2008年に開発した。OSCARアーキテクチャはシンプルな作りやすい構成のため、ルネサスが、最初の4コアチップRP1（2007年開発）、2個目の上記RP2チップとも、設計からチップの完成まで9ヶ月程度で一発完動の状態で完成させた。両方のチップともISSCCのプロセッサセッションで発表すると共に、チップ発表時にはOSCARコンパイラによりマルチメディアアプリケーションが並列化されて動作しているという短期間でのハードウェア・ソフトウェア開発を実現した。特にRP2では図6に示すように、オフチップ集中共有メモリDDR2、分散共有メモリURAM、ローカルデータメモリDLRAM、データキャッシュD-cache、オンチップ集中共有メモリCSM、各コア毎のデータ転送ユニット、階層並列をサポートするための任意グループでのバリア同期機構[5]CCN BAR、4コアまでのハードウェアキャッシュコヒーレンス機構2セット、各コア毎の周波数制御及びパワーゲーティング制御とチップ全体での電圧制御がロジック系とメモリ系独立に制御できる電力制御用レジスタを実装した[1]–[6]。周波数は600MHz、300MHz（1／2）、150MHz(1／4)、75MHz(1／8)、0MHz（クロックゲーティング）の5段階を1クロックで変えられ、電圧は1.4V、1.2V、1.0Vの3段階で変えられ、5μsで電源遮断を30μsで復帰できるソフトウェア制御可能な構成とした。

このチップはグリーンITの代表的

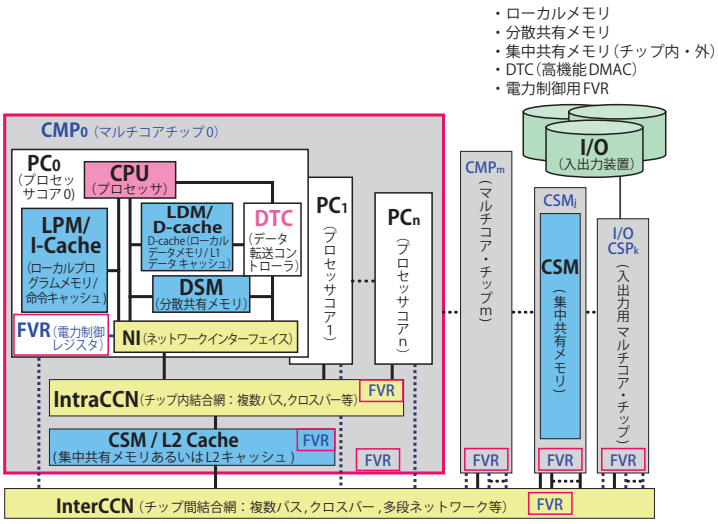
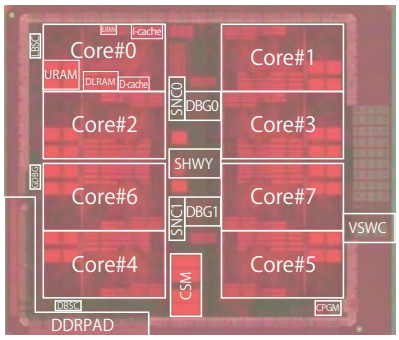



図4　OSCAR API標準的マルチコアOSCARメモリアーキテクチャ
(Optically Scheduled Advanced Multiprocessor)

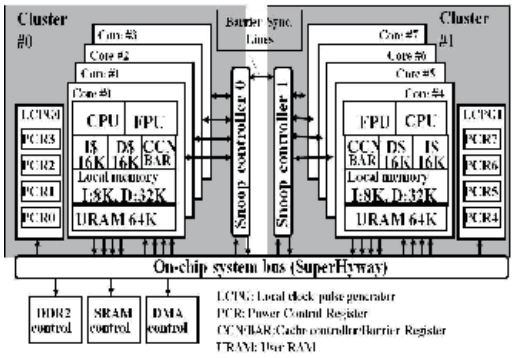



8コア集積マルチコア LSI チップ写真

| | |
|---|---|
| プロセス | 90nm CMOS, 8層メタル, 3種Vth |
| チップサイズ | 104.8mm²（10.61mm x 9.88mm） |
| 電源電圧 | 1.0V-1.4V（コア）, 1.8/3.3V（I/O） |
| 動作周波数 | 600MHz |
| CPU性能 | 8640 MIPS（Dhrystone 2.1） |
| FPU性能 | 33.6 GFLOPS |
| 低電力制御 | ・CPU毎に独立した周波数変更<br>・CPUコアのクロックを停止するスリープモード<br>・CPUコアの一部のクロックを停止するがキャッシュコヒーレンシ維持可能なライトスリープモード<br>・CPUコアの電源供給を停止するフル電源遮断モード<br>・URAM以外のCPUコアの電源供給を停止するレジューム電源遮断モード |

ISSCC08発表：ISSCC08 論文番号4.5, M.ITO, et al., "An 8640 MIPS SoC with Independent Power-off Control of 8 CPUs and 8 RAMs by an Automatic Parallelizing Compiler"

図5　早稲田OSCARコンパイラ協調型アーキテクチャ
ホモジニアスマルチコアRP2　SH4A8コア搭載

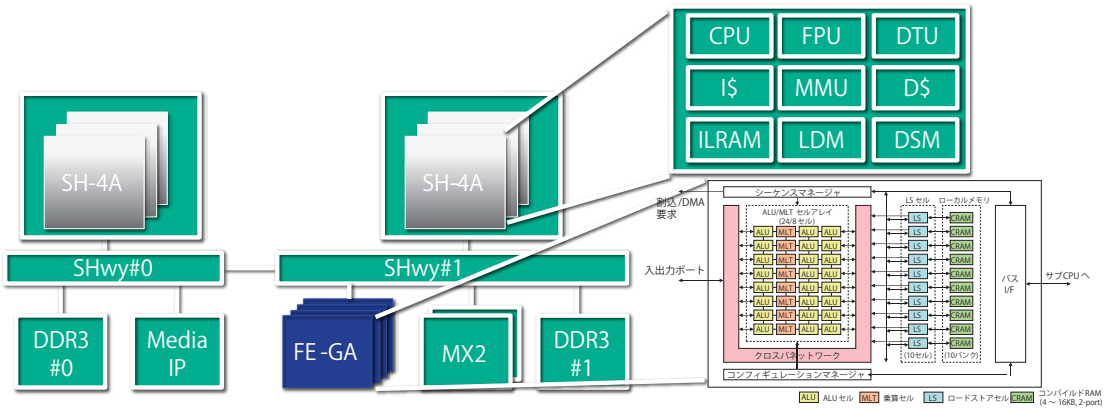



| | |
|---|---|
| Process Technology | 90nm, 8-layer, triple-Vth, CMOS |
| Chip Size | 104.8mm² (10.61mm x 9.88mm) |
| CPU Core Size | 6.6mm² (3.36mm x 1.96mm) |
| Supply Voltage | 1.0V–1.4V (internal), 1.8/3.3V (I/O) |
| Clock frequency | 600MHz, 300MHz, 150MHz, 75MHz |
| Power Domains | 17 (8 CPUs, 8 URAMs, common) |

"An 8640 MIPS SoC with Independent Power-off Control of 8 CPU and 8 RAMS
by an Automatic Parallelizing Compiler", IEEE ISSCC2008, Masayuki Ito, Toshihiro Hattori, Yutaka Yoshida, Kiyoshi Hayase, Tomoichi Hayashi, Osamu Nishii, Yoshihiko Yasu, Atsushi Hasegawa, Masashi Takada, Masaki Ito, Hiroyuki Mizuno, Kunio Uchiyama, Toshihiko Odaka, Jun Shirako, Masayoshi Mase, Keiji Kimura, Hironori Kasahara

低コスト化のため5コア以上ハード・コヒーレンシ制御機能無し→ソフト・コヒーレンシ制御必要

図6　8コアホモジニアスマルチコアRP2のアーキテクチャ




・8 Renesas SH-4As as CPUs, 4 Hitachi FE-GAs dynamically reconfigurable processor as accelerators
・I$:32KB/core, D$:32KB/core
・Frequency/Voltage State: 648MHz@1.3V, 324MHz@1.16V, 162MHz@1.0V

"A 45nm 37.3GOPS/W heterogeneous multi-core SoC", IEEE ISSCC2010
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, H. Maejima,

図7　15コアヘテロジニアスマルチコアプロセッサRP-X（45nm ローパワーテクノロジー使用）

技術として2008年4月10日の第74回総合科学技術会議で当時の福田総理を含め関係閣僚に紹介された。

また、引き続き2006年度から2009年度まで行ったNEDO "情報家電用ヘテロジニアスマルチコア" プロジェクトでは、上記SH4Aプロセッサコア8個と3種類のアクセラレータ7個（4つのDRP（Dynamic Reconfigurable Processor）FEGAコア、2つの画像認識エンジンMX2、1つのコーデックエンジンVPU5)、計15コアを45nmの低電力プロセスで集積したヘテロジニアスマルチコアRPX（図7）を2010年に開発しISSCCにて発表した。このチップではリーク電力がほとんどないプロセスを利用できたため648MHz、324MHz、162MHz、81MHz、0MHzのコア別周波数制御、1.3V、1.16V、1.0Vのチップ一括電圧制御のみを実装した。

## 4. OSCAR自動並列化及び電力削減コンパイラ[7]－[10]

OSCAR自動並列化コンパイラプロジェクトは1983年に開始し、30年近い長期に渡り開発を続け、現在、科学技術計算用のFortran及び組込用のCの自動並列化を行うことができると共に、世界で唯一自動電力制御が行えるコンパイラとなっている。OSCARコンパイラの並列処理方式の特徴は、1980年代後半に実現したマルチグレイン並列化と1990年後半にローカルメモリ最適利用のために開発し、2000年より開始したNEDOアドバンスト並列化コンパイラでキャッシュ最適化用に改良して実用レベルに高めたデータローカライゼーション技術である。

マルチグレイン並列化は図8に示すように、市販並列化コンパイラが利用しているループ並列化技術（ループ内の

**プロセッサ高速化における3大技術課題の解消**

**1.半導体集積度向上（使用可能トランジスタ数増大）に対する速度向上率の鈍化**

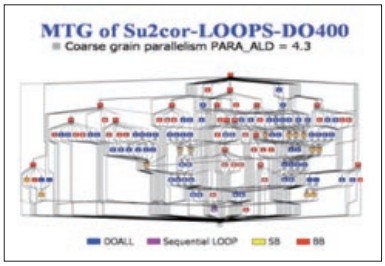
粗粒度タスク並列化,ループ並列化,近細粒度並列化によりプログラム全域の並列性を利用するマルチグレイン並列化機能により,従来の命令レベル並列性より大きな並列性を抽出し,複数マルチコアで速度向上

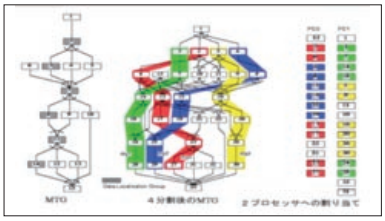
**2.メモリウオール問題**

コンパイラによるローカルメモリへのデータ分割配置,DMAコントローラによるタスク実行とオーバーラップしたデータ転送によりメモリアクセス・データ転送オーバーヘッド最小化

**3.消費電力増大による速度向上の鈍化**

コンパイラによる低消費電力制御機能を用いたアプリケーション内でのきめ細かい周波数・電圧制御・電源遮断により消費電力低減



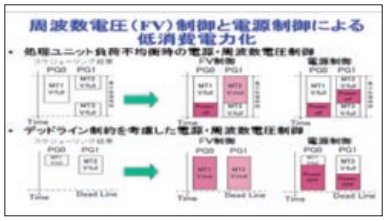



図8　OSCARマルチコア用コンパイラの特徴的技術

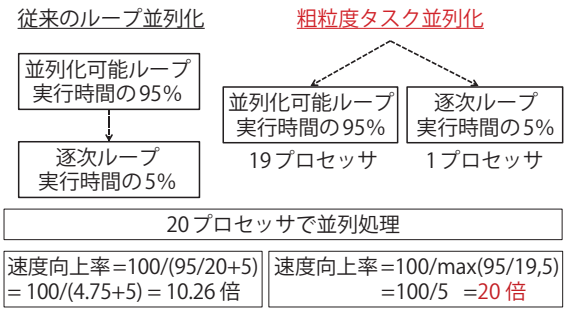



図9　粗粒度タスク並列：ループ並列の限界を越えるために

繰り返し、たとえばi=1、100の100回の繰り返しをプロセッサ間で並列実行する）に加え、ループ、関数（あるいはサブルーティン）、基本ブロック（代入文及び条件分岐文からなるブロック）を粗粒度タスクと定義しそれらの間の制御依存（条件分岐に伴って生じる依存関係）及びデータ依存（データの定義及び使用によって生じる依存関係）を解析し条件分岐を越えた並列性を利用する粗粒度タスク並列性と、基本ブロック内の文単位の近細粒度並列性の、プログラムの複数粒度の並列性を利用する独自の方式である。この粗粒度並列化を行うことにより、アムダールの法則で知られる逐次部分があるとプロセッサ数の増加と共にスケーラブルな速度向上が得られないという状況を改善することができる。たとえば図9の左のように、95%の処理コストを持つ並列化ループと5%のコストを持つ逐次ループがあるとすると、従来のループ並列化では95%ループを全プロセッサ（図では20プロセッサ）で処理し、その後逐次ループを1プロセッサで処理することにより、20プロセッサで最高10.26倍の速度向上を得ることができる。これに対し粗粒度タスク並列化では、並列ループと逐次ループ間の並列性を検出し、並列ループを19プロセッサで並列処理し、逐次ループを残りの1プロセッサで同時処理することにより最高20倍の速度向上を得ることができる。図8の右上のSPEC95のSu2corプログラム中の9重ネスト目のループボディ部の粗粒度タスクグラフ（タスク間の並列性を表すグラフ）を見ると分かるように、青色のブロックで示す並列ループと赤ブロックで示す逐次の基本ブロック及びピンクで示す逐次ループが同時実行することができると解析されており、従来のループ並列に比べ、非常に大きな並列性を抽出できることが分かる。

また、ループ間などの並列性を解析した後、図8右中央の左の網掛けされた6つのタスクが、実線で示すデータ依存エッジで結ばれたタスクグラフのように、ループ間にデータ依存があることが分かると、OSCARコンパイラでは依存があるループ間でのキャッシュのグローバル最適化を試みる[7]。具体的には、各ループでアクセスされる配列を調査し、図中の緑、赤、青、黄に色分けされたループに分割し、同一の色を持つ分割ループは同一の配列部分にアクセスするように調整することにより、同一色の分割ループを同一プロセッサに割り当てると同一色の6つのループ間では全ての配列データがキャッシュ上で再利用されるようになる。またこのローカライズ技術は、現在さらに進み、任意のサイズのローカルメモリあるいは分散共有メモリが与えられた時に、DMA（DTU）[4]を用いアクセスされる前に前記プロセッサ近接のローカルあるいは分散共有メモリに事前ロードし、プログラム全域で再利用したり、送付先のメモリがいっぱいの場合には送付先プロセッサのDTUがメモリからの掃き出し優先順位にしたがってデータを共有メモリ等へ掃き出したことを同期フラグで知らされたら、自動的に空いたメモリにデータを転送したり、将来再利用されるデータであるがしばらくの間使用されずメモリの領域を開ける必要がある場合には、CPUによるタスク実行の裏側でDTUが当該データを集中共有メモリに待避し、使用時までに再ロードするようなローカルメモリ管理、データ転送技術へと進化している[9]。

さらにコンパイラは、タスクのプロセッサへのスケジュール結果を解析し、自動的に各コアの周波数電圧制御、パワーゲーティング制御により電力を削減することができるようになっている。図10を用いて基本的な概念を説明すると、最速実行モードの際にはタスクグラフのクリティカルパス上のタスク集合はフルスピード（100%の周波数）で実行される必要があり、図10上図ではプロセッサ0に割り当てられたMT1とプロセッサ1に割り当てられたMT3が最大周波数で実行される。しかし、MT3はMT1の計算結果を必要とするので、MT2の実行後プロ

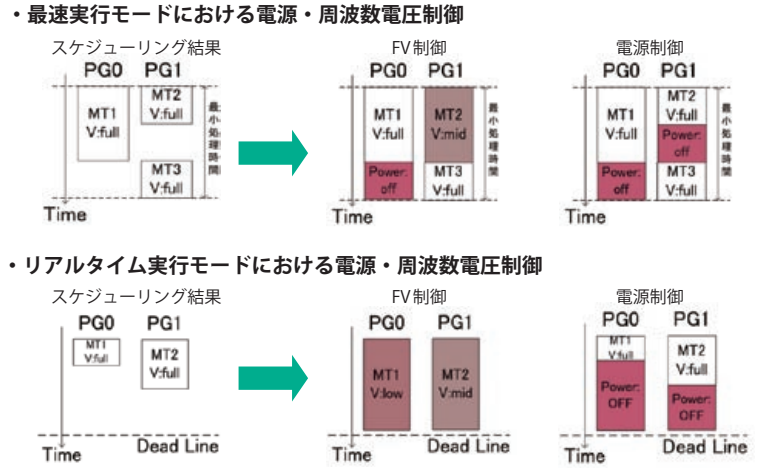



図10　周波数電圧（FV）制御と電源制御による低消費電力化

セッサ1は同期のためのビジーウエイト状態となり、プロセッサ0がMT1の出力データをプロセッサ1上の分散共有メモリにストアし、MT1の終了を通知する同期フラグを分散共有メモリ上にセットするのを待つ。この際、ビジーウエイトは電力を消費してしまうので、OSCARコンパイラはMT2を低周波数・低電力で実行するか、もしくは電源遮断するのに十分な待ち時間があればプロセッサ1の電源を一時遮断する。また図10下に示すリアルタイム実行モードでは、処理終了後デッドラインまでの待ち時間に電力を消費するのを避けるため、下中央のようにMT1を1／4の周波数、MT2を1／2の周波数及びそれらに合った電圧で実行することにより電力を下げるか、右図のようにクロックゲーティングあるいはパワーゲーティングによりプロセッサを停止させる。この時、コンパイラは電源状態遷移オーバーヘッドも考慮してヒューリスティック的に適切な電力モードを自動選択する[6]。

## 5. OSCAR APIとコンパイルフロー

4.で述べたOSCARコンパイラによりマルチグレイン並列化、メモリ最適化、電力最小化された並列プログラムは、通常のコンパイラと同様SPARC等用の並列バイナリープログラムを生成できるほか、OpenMPを用いたCあるいはFortran並列プログラムとして生成され、OpenMPコンパイラを持つ任意のマルチプロセッササーバ上で実行することができる。しかし、我が国が得意とする組込製品では、ハードリアルタイム処理（デッドラインを確実に守らなければならない処理）が必要となり、この場合、ミスヒットするとメモリアクセス時間が長くなりデッドラインを守れない可能性のあるキャッシュメモリは使用できない。そこで、データの転送をソフトウェアで明示的に行い動作を厳密に制御できるローカルメモリ、分散共有メモリ及びそのデータ転送のためのDMA（前述のDTU）の利用が必要となる。OpenMPでは、分散共有メモリ、DMA、また電力制御、グループバリア、時間管理、アクセラレータの利用などの機能が無いため、これらの機能を用意したOSCAR　API（図11）をNEDOプロジェクトの支援も受けつつ、IT／半導体企業と共に早稲田大学OSCAR API委員会にて作成した。第一期委員会 早大、日立、NEC、富士通研、ルネサス、東芝、パナソニック（2005－2008）によりホモジニアスマルチコア用OSCAR API V1.0を策定し、第二期委員会（2009－2010）により早大、日立、NEC、富士通、ルネサス、東芝のメンバーで、アクセラレータも集積したヘテロジニアスマルチコアのサポート及びメニーコアにおけるキャッシュメモリのソフトウェアコヒーレント制御のための拡張を検討し、第三期委員会（2011－）早大、名大、東邦大、日立、ガイオ・テクノロジー、三菱スペース・ソフトウェア、NEC、イーソル、ルネサスソリューションズ、東芝、三菱電機、オリンパス、富士通研、ルネサスエレクトロニクス、キャッツ、東芝、セミコンダクター、デンソーで、上記拡張を含めたOSCAR API V 2.0を策定した。2012年春公開予定のOSCAR API V 2.0（図12）では、逐

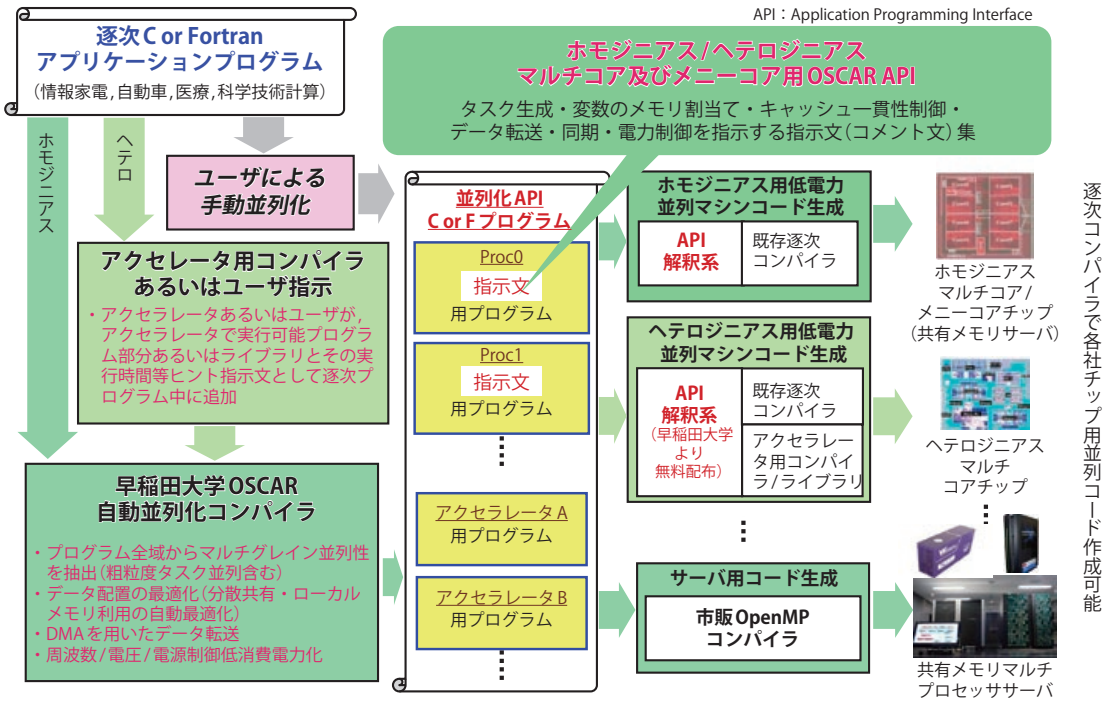



図11　（ホモジニアス/ヘテロジニアス）マルチコア・メニーコア用プログラム開発

□ Parallel Execution API
・parallel sections *
・flush *
・critical *
・execution

□ Memory Mapping API
・threadprivate *
・distributedshared
・onchipshared

□ Synchronization API
・groupbarrier

□ Timer API
・get_current_time

* Directives from OpenMP

□ Data transfer API
・dma_transfer
・dma_contiguous_parameter
・dma_stide_parameter
・dma_flag_check
・dma_flag_send

□ Power control API
・fvcontrol
・get_fvstatus

□ Heterogeneous API
・accelerator_task_entry

□ Cache control API
・cache_writeback
・cache_selfinvalidate
・complete_memop

◆ Hint Directive
・accelerator_task
・oscar_comment

図12　OSCARヘテロジニアスAPI
http://www.kasahara.cs.waseda.ac.jp/api/regist_en.html

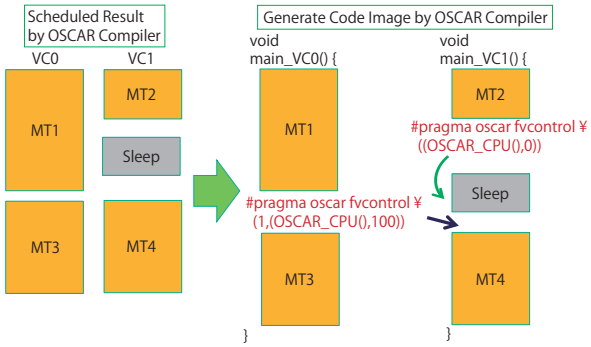



図13　OSCAR APIを用いた低電力制御

次CあるいはFortranプログラムをコンパイラが自動並列化・電力最小化し、各社の命令セットが異なるホモジニアスマルチコア及びヘテロジニアスマルチコア、さらにはキャッシュコヒーレンス用ハードウェアを持たないメニーコア上でも動作させることを可能としている。

OSCAR APIの特徴は、各プロセッサ毎に別々のスレッドを用意し、各スレッドに埋め込まれたディレクティブは早大が無料配布するAPI解釈系により各プロセッサ用のライブラリコール（プロセッサ企業側でDMA、電力制御などのライブラリを用意することが前提）に変換されるので、プロセッサメーカ側は通常の逐次CあるいはFortranコンパイラのみを用意すれば、並列バイナリが簡単に手に入る。これによりOpenMPコンパイラや並列化コンパイラの開発が不要になると共に、非常に短期間でコンパイラを用いた並列実行が可能となる。この機能により、OpenMPコンパイラが用意されているサーバでも、OpenMP機能を使わず逐次コンパイラのみを用いて並列実行を行うことができる。また、共有メモリをもつマルチプロセッサであれば、他のプロセッサへの移行も簡単に行え、ルネサスのSuperHプロセッサ・V850プロセッサ、ARM MPCore、富士通FRV・Sparc、Intel、AMD、IBMの任意のプロセッサで簡単に動作させることができる。また、ヘテロジニアスマルチコアでは、アクセレータ用のコンパイラが開発メーカ等から提供されていれば、それらのコンパイラと協調して自動並列化が可能であるのと、C等のソースコードがなくアクセレータ用のライブラリが用意されている場合にはユーザがライブラリでの実行にかかる時間と入出力データの情報をコメント文として逐次CあるいはFortranプログラムに追加すれば、OSCARコンパイラがアクセレータと汎用プロセッサ間での負荷分散とそれらの間のDMA等も用いた通信コードを自動生成する[10]。また、他の特徴としては、図12に示すようにディレクティブ数を非常に少なく抑えたシンプルな構成のため、各マルチコア上で最初に動作させるのが簡単かつ短時間で行え、またユーザにとっても覚えやすく、分かりやすいという特徴がある。たとえば図13に示す例のように、Cプログラムにおいて、プロセッサ1がスリープする時には自CPUを0%の周波数とすることを指定し、プロセッサ0がプロセッサ1を起こす場合には、図中の（1, CPU, 100）のようにプロセッサ1のCPUを100%で動かすと指定すれば簡単に電力制御ができる。電源を遮断したいときには（CPU, －1）と指定すれば遮断することができる。このAPI仕様はhttp://www.kasahara.cs.waseda.ac.jp/で無料公開されている。

## 6. OSCARコンパイラとAPIの応用事例

図14は、Fortranプログラムを自動並列化してインテル4コアマルチコア及びIBM　Power6マルチコアベース32コアSMPサーバ上でSPEC CFPベンチマークに対する性能を評価した例を示している。図中、青のバーは、インテル及びIBMコンパイラで、それぞれ各アプリケーションを4コア、32コア用自動並列化して動作させた時、1コアより何倍スピードアップできたかを示している。インテルコンパイラでは、4コアの利用で、2アプリケーションに対して2倍程度の速度向上ができているが他のアプリケーションではあまり高速化ができていないことが分かる。またIBMコンパイラでも、32コアの利用で、10倍強の速度向上ができているのが1アプリケーション、5倍強が1アプリケーションで他は大きな速度向上が得られていないことが分かる。それに対しOSCARコンパイラでOSCAR APIで並列化したFortranプログラムを生成し、各社のコンパイラでバイナリに落とすと、同一プログラムが赤のバーのように、インテルプロセッサで平均2倍、IBMプロセッサで平均3倍程度高速化できることが分かる。また、NEC／ARMの4コアMPcoreプロセッサ上でFortranとC（マルチメディアコード）をOSCARコンパイラで自動並列化した所、ARM用には市販の並列化コンパイラがないた

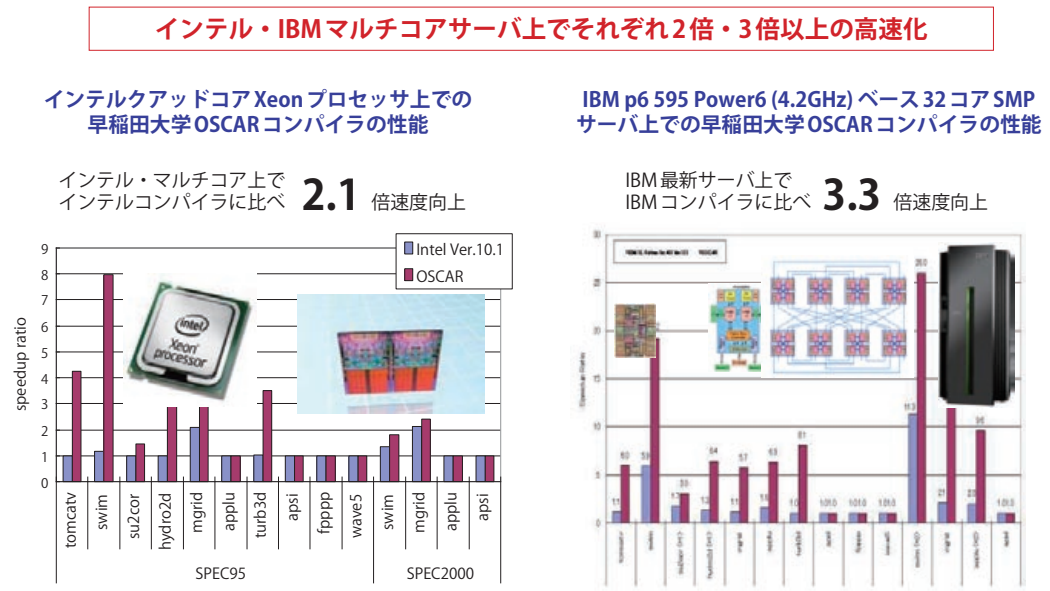



図14　現状：世界最高性能のOSCARコンパイラの性能

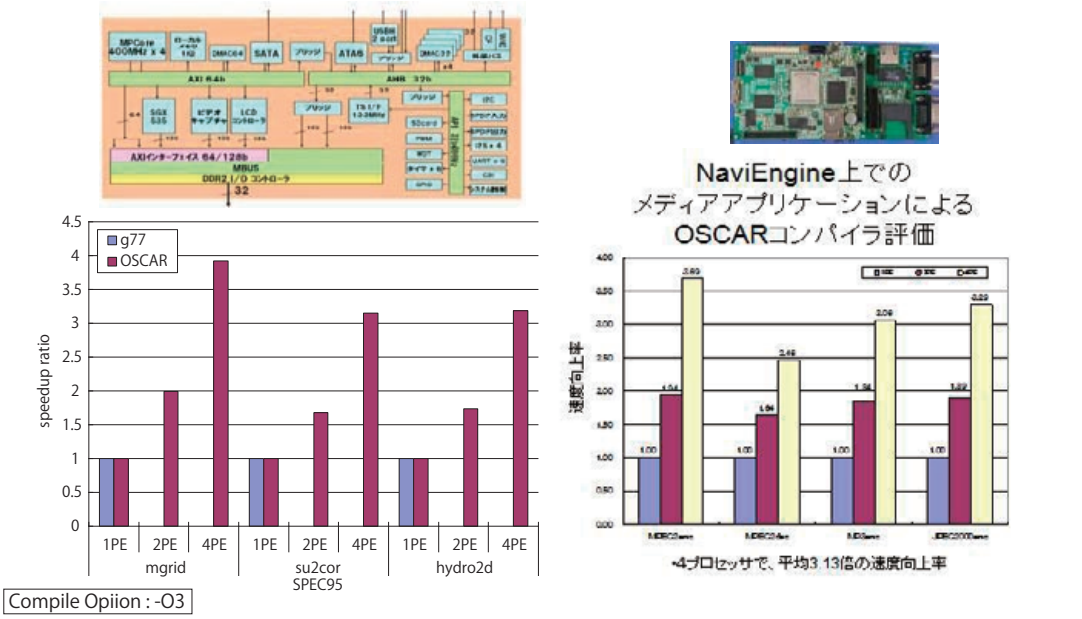



図15　NECナビエンジン（ARM-NEC MPcore）上でのOSCARコンパイラの性能

め比較できないが、1コアに比べFortranコードで3.4倍、Cで3.1倍の速度向上が得られた。

また、RP2上での電力削減例を図16に示す。図ではデジタルテレビで使用されるMPEG2デコードの8コア上でのリアルタイム並列実行において、電力制御をしないと5.7W消費する計算が、コンパイラによる電力制御を行うと1.5Wで行え、電力を1／4程度にまで削減できていることが分かる。

また図17は、ヘテロジニアスマルチコアRPX上でSH4Aプロセッサと FEGA（前述のDRP）アクセラレータを用いて、動画像の変化部分を検出するオプティカルフロー計算を並列化すると、SH4A 1コアと比べ、ホモジニアスな8コア部分で5.4倍の速度向上が得られ、さらにFEGAアクセレータを4コア同時動作させると32.7倍高速化できることを示している。またこのオプティカルフローのリアルタイム処理時の電力削減を行うと、図18に示すように電力制御しない場合に1.76Wを要した計算が0.54Wで実行できることが分かる。

また、このRPXを用いてLinuxベースWebサーバを試作した所、図19に示すように628MHz 8コア動作時でほ

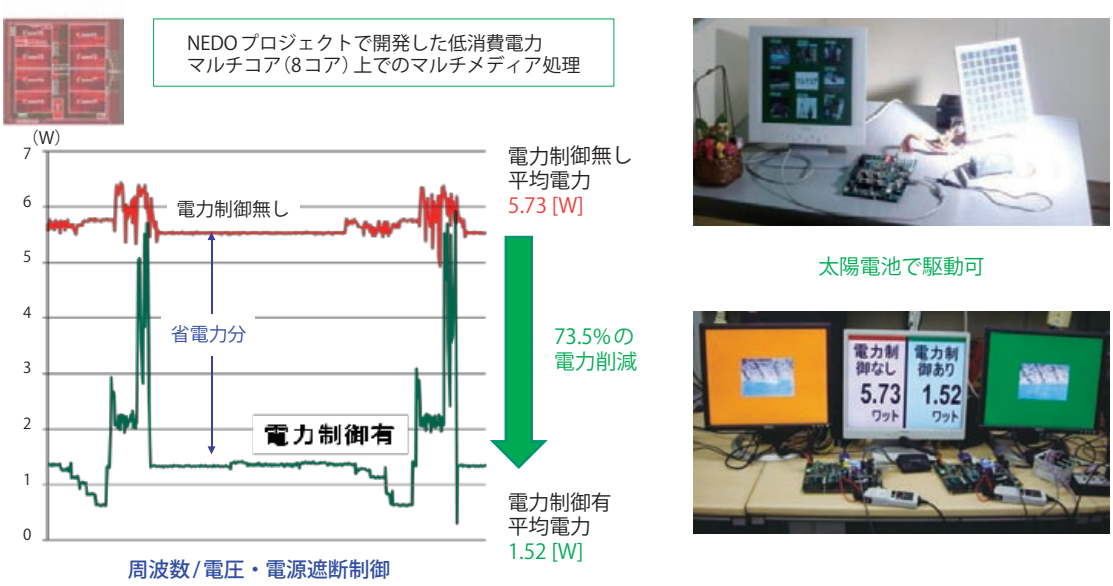



図16　現状：世界唯一コンパイラによる消費電力削減に成功

**-画像動作追従のためのオプティカルフロー演算のコンパイラ自動並列化（世界初）-**

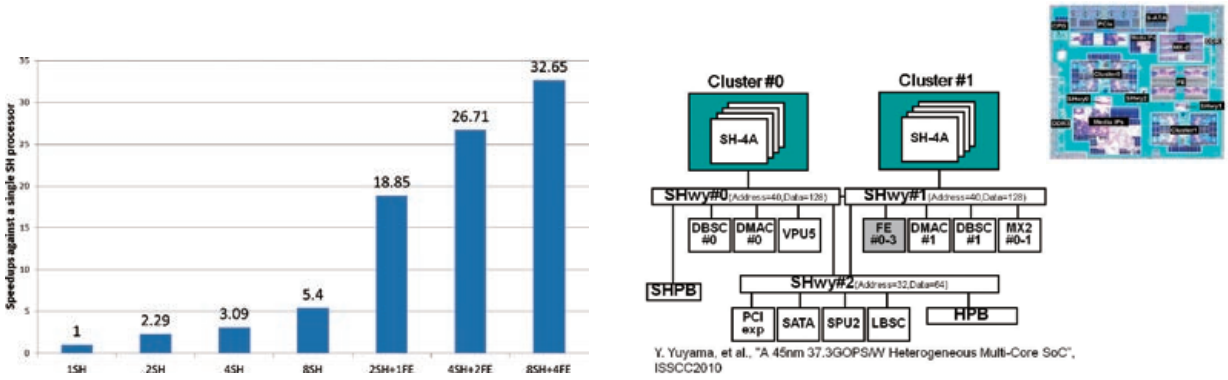



図17　ヘテロジニアスマルチコアRP-X上での処理性能
NEDO情報家電用ヘテロジニアスマルチコアプロジェクト（2006-09）

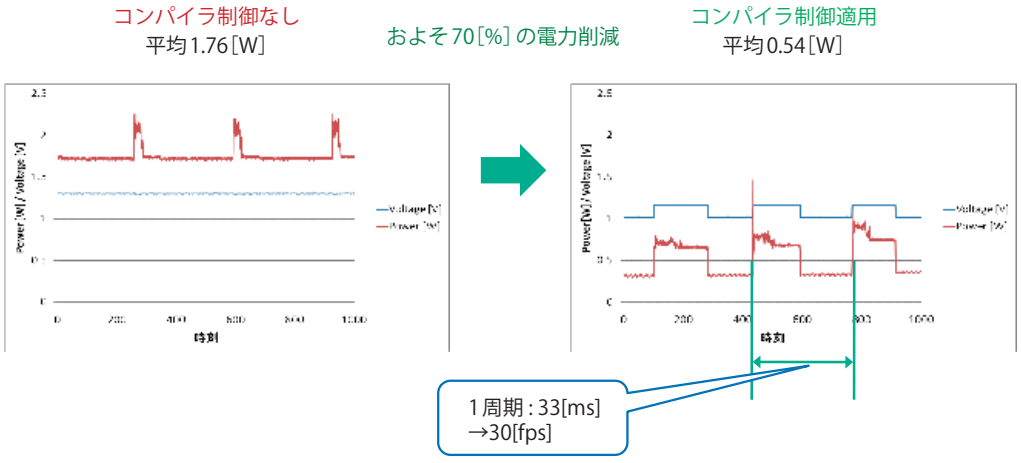
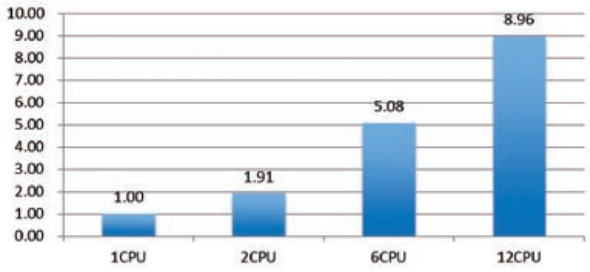



図18　オプティカルフロー（ライブラリ利用）に対するリアルタイム処理時の消費電力制御

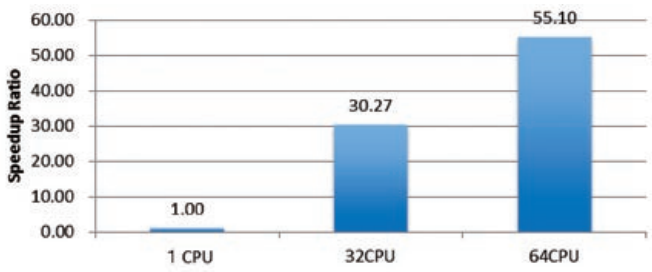
ぼ1Wで動作することが分かる。現在http://www.kasahara.cs.waseda.ac.jp/のサービスはこのサーバで行われておりリアルタイムの消費電力が表示できるようになっている。これにより24時間動作するWebサーバの電力を通常のサーバと比べ数十分の1に削減できることが確かめられた。

最後に、重粒子線ガン治療計算のOSCARコンパイラとAPIを用いた時の並列処理性能について図20に示す。この計算はCで記述され、従来OpenMPを用い手動並列化した場合、16コア使用しても1コアの2.5倍の速度向上しか得られていなかった計算であるが、OSCARコンパイラを用いることによりインテルプロセッサベース12コアSMPで約9倍の速度向上、IBM Power7ベース64コアサーバで55倍の速度向上が得られ、1日当たりに治療できる患者数の増大、それに伴う治療費の軽減が可能となることが確かめられた。


図19　組込マルチコアRPX利用低消費電力Webサーバ




8.9times speedup by 12 processors
Intel Xeon X5670 2.93GHz 12 core SMP
(Hitachi HA8000)




55 times speedup by 64 processors
IBM Power 7　64 core SMP
(Hitachi SR16000)

National Institute of
Radiological Sciences
(NIRS)

図20　重粒子線ガン治療装置 線量計算自動並列化性能（従来手動で16コアで2.5倍）

## 7. まとめ

本稿では、グリーンコンピューティングのために重要な低消費電力のマルチコアプロセッサのアーキテクチャ及びコンパイラ・API等のソフトウェア技術、これらをマルチメデイア処理、科学技術計算、医療などへ応用した時の、速度向上、電力削減の例を示した。低消費電力のマルチコアは今後、スマートフォン等の情報家電、自動車、医療、サーバ、エクサフロップススパコンなど多くの情報機器で使用されていくと考えられる。低消費電力化でクラウドサーバ、スパコンによる環境負荷を軽減し、1週間に一度の充電かつ災害時には太陽光電力などで動作できるスマートフォンのような高付加価値製品を開発したり、病気・災害から人命を守れるスパコンなどその重要性は益々高まっていくものと考えられる。

参考文献


[1]「マルチプロセッサ」、日本国特許 第4304347号、May. 15. 2009.
[2]「マルチプロセッサ」、日本国特許 第4784792号、Jul.22.2011.
[3]「マルチプロセッサ及びマルチプロセッサシステム」、日本国特許 第4784842号、Jul. 22. 2011.
[4]「プロセッサ及びデータ転送ユニット」、日本国特許第4476267号、Mar.19.2010
[5] "MULTIPROCESSOR SYSTEM AND METHOD OF SYNCHRONIZATION FOR MULTIPROCESSOR SYSTEM"、8108660 (US Patent)、Jan. 31. 2012.
[6]「マルチプロセッサシステム及びマルチグレイン並列化コンパイラ」、日本国特許第4082706号、Feb. 22. 2008.
[7]「コンパイル方法、コンパイラ、およびコンパイル装置」、日本国特許第4177681号、Aug. 29. 2008
[8] "LOCAL MEMORY MANAGEMENT、INFORMATION-PROCESSING DEVICE、PROGRAM CREATION METHOD AND PROGRAM"、2459802 (GB Patent)、Jan. 04. 2012.
[9] "LOCAL MEMORY MANAGEMENT、INFORMATION-PROCESSING DEVICE、PROGRAM CREATIONMETHOD AND PROGRAM"、2478874 (GB Patent)、Dec. 28. 2011.
[10]「ヘテロジニアスマルチプロセッサ向けグローバルコンパイラ」、日本国特許 第4784827号、Jul.22.2011.



## profile

**笠原 博徳**（かさはら ひろのり）

1985年 早稲田大学博士課程了 工学博士
カリフォルニア大学バークレー客員研究員
1986年 早大理工専任講師
1988年 助教授
1997年 教授、現在 理工学術院情報理工学科
1989年～1990年 イリノイ大学Center for Supercomputing R&D客員研究員
2009年よりIEEE Computer Society 理事。
IFAC World Congress Young Author Prize,情報処理学会坂井記念特別賞, STARC（半導体理工学研究センター共同研究賞, LSI・オブ・ザ・イヤー 2008 準グランプリ, Intel Asia Academic Forum Best Research Award, IEEE Computer Society Golden Core Member Award受賞。査読付論文 189件, 招待講演104件,シンポジウム論文 29件, 研究会論文134件, 全国大会論文 154件,特許 32件（内16件既取得）, 新聞・Web記事・TV等メディア掲載 443件等。経済産業省・NEDO：情報家電用マルチコア及びアドバンスト並列化コンパイラ・グリーンコンピューティングシステム等プロジェクトリーダ、NEDOコンピュータ戦略（ロードマップ）委員長,「グリーンネットワーク・システムプロジェクト（グリーンITプロジェクト）」技術委員長、内閣府：スーパーコンピュータ戦略委員会, 政府調達苦情検討委員等、文部科学省：地球シミュレータ（ES）評価委員、情報科学技術委員,HPCI計画推進委員,次世代スパコン（京）概念設計評価委員・中間評価委員, ES2導入技術アドバイザイリー委員長, IEEE,情報処理学会, ACM等の国際会議プログラム委員、高校生科学技術チャレンジ審査委員等歴任。