

## 実アプリケーションを用いた チップマルチベクトルプロセッサの消費エネルギー評価

永岡 龍一<sup>†1</sup> 佐藤 義永<sup>†1</sup> 撫佐 昭裕<sup>†2</sup>  
江川 隆輔<sup>†3,†4</sup> 滝沢 寛之<sup>†1,†4</sup> 小林 広明<sup>†3,†4</sup>

ベクトル型スーパーコンピュータは高精度・大規模なシミュレーションを可能とする一方で、高実行効率を支える高いメモリバンド幅や大容量のメモリに要する消費電力が問題となっている。したがって、今後のベクトル型スーパーコンピュータの設計では、高性能化だけでなく、低消費電力化の実現も求められている。高性能かつ低消費電力なベクトル処理を実現するアーキテクチャとしてチップマルチベクトルプロセッサ (CMVP) が提案されている。しかし、これまで消費エネルギーの観点から CMVP の評価はなされていない。そこで本稿では、CMVP の電力モデルを検討し、CMVP におけるベクトルキャッシュの有効性を実アプリケーションにより評価する。

### Energy Consumption of a Chip Multi-Vector Processor Using Real Applications

RYUICHI NAGAOKA,<sup>†1</sup> YOSHIEI SATO,<sup>†1</sup> AKIHIRO MUSA,<sup>†2</sup>  
RYUSUKE EGAWA,<sup>†3,†4</sup> HIROYUKI TAKIZAWA<sup>†1,†4</sup>  
and HIROAKI KOBAYASHI<sup>†3,†4</sup>

High performance computing using vector supercomputers has been shown to be effective for scientific simulations. However, a memory system of vector supercomputers requires the high-energy consumption to keep a high-memory bandwidth. To achieve high sustained performance and low energy consumption, a chip multi-vector processor (CMVP) has been proposed. However, a CMVP has not been evaluated from the point of view of energy consumption. Therefore, we evaluate the energy consumption of a CMVP. First, we establish an energy consumption model of a CMVP to analyze the energy consumption. Then, we evaluate the energy consumption to compare the several designs of varying hardware parameters.

### 1. はじめに

ベクトル型スーパーコンピュータは、その高い実効性能から、科学技術計算分野で広く利用されており、今後さらなる高性能化が期待されている。ベクトル型スーパーコンピュータは、極めて高いメモリバンド幅を有しているが、大規模、かつ高速なメモリシステムは大きな消費電力を必要とする。また実装面積の制約によりベクトルプロセッサとオフチップ間の配線長は長く、データ転送により高い電力を消費する。したがって、ベクトル型スーパーコンピュータにおいて、高性能かつ低消費電力を実現するためには高い実効メモリバンド幅を維持したままで、メモリシステムの消費電力を抑制する必要がある。

高い実効性能を実現するためには、高い演算性能と、演算性能に対する十分なデータ転送性能が必要である。ベクトル型スーパーコンピュータの高い演算性能を実現するアーキテクチャとして、チップマルチベクトルプロセッサ (CMVP)<sup>1)</sup> が提案されている。CMVP では、複数のコアにより複数のスレッドを並列に処理することで、高い実効演算性能を実現する。また、CMVP では高いデータ転送性能を実現するために、ベクトルキャッシュと呼ばれるオンチップメモリ<sup>2)</sup> を搭載している。ベクトルキャッシュを利用することにより、データ転送性能を向上させると同時に、消費電力の大きいオフチップメインメモリへのアクセスを削減することができる。このことから、メモリシステムを含むシステム全体の消費電力の削減が期待できる。そこで本報告では、CMVP におけるベクトルキャッシュの有効性を消費電力と性能の観点から明らかにする。

本報告ではまず、CMVP のメモリシステムの消費エネルギーをモデル化する。次に、実アプリケーションに対する CMVP の消費エネルギーをモデルに基づいて定量的に評価し、CMVP におけるベクトルキャッシュの有効性を消費エネルギーの観点から明らかにする。

### 2. チップマルチベクトルプロセッサおよびメモリシステム

本章では、まず本研究で想定する CMVP およびそのメモリシステムのモデルについて述

---

<sup>†1</sup> 東北大学情報科学研究科

Graduates School of Information Sciences, Tohoku University

<sup>†2</sup> 日本電気株式会社

NEC Corporation

<sup>†3</sup> 東北大学サイバーサイエンスセンター

Cyberscience Center, Tohoku University

<sup>†4</sup> JST CREST

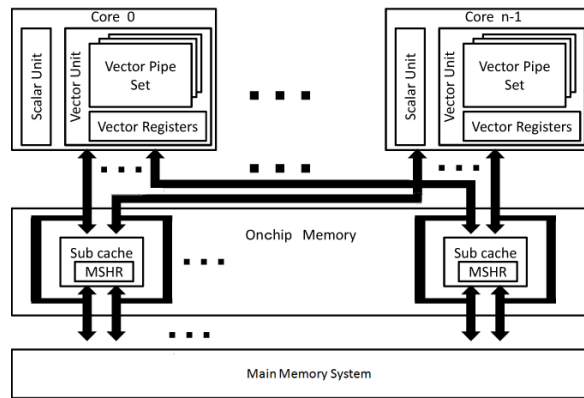


図 1 チップマルチコアベクトルプロセッサとメモリシステムのモデル

べ、その消費エネルギーモデルを定義する。ベクトルキャッシュを有する CMVP とメモリシステムの構成を図 1 に示す。モデルを構成する要素は大きくベクトルコア、ベクトルキャッシュ、ミスステータスハンドリングレジスタ (MSHR)<sup>3)</sup>、そしてメインメモリシステムに分類される。以下に各構成要素について述べる。

### 2.1 ベクトルコア

CMVP では、多数のコアがオフチップメインメモリ (以下メインメモリと表記) 、およびベクトルキャッシュを共有する。コアはベクトルユニットとスカラユニットから構成される。スカラユニットでは命令のフェッチやデコード、ベクトルユニットへのベクトル命令の供給、ベクトル化できない演算命令や分岐命令等の処理を行う。ベクトルユニットはベクトルレジスタとパイプラインセットからなる。ベクトルレジスタはベクトルデータの格納、およびベクトルパイプラインへのベクトルデータ供給を行う。複数のベクトルパイプラインセットは同時動作を可能としており、個々のベクトルパイプラインセットは論理演算・乗算・加算/シフト演算・除算を並列処理する。

ベクトル型スーパーコンピュータが対象とする科学技術計算のアプリケーションは、一般的に多重ループを数多く含んでいる。CMVP では、最外ループを分割してマルチスレッド化し、複数のベクトルコアで並列処理を行うことで、高い実効演算性能を実現する。

### 2.2 ベクトルキャッシュ

ベクトルキャッシュは、ベクトルプロセッサ上に搭載する高速動作が可能なメモリであり、

ソフトウェア制御により任意のデータを格納することができる。また、ベクトルキャッシュは独立動作が可能な複数のサブキャッシュから成り、各サブキャッシュからベクトルレジスタへ並列にデータを供給することで高いメモリバンド幅を有している。メモリアクセス時には、通常メインメモリからベクトルレジスタにデータが転送される。しかし、ベクトルキャッシュに必要なデータが格納されている場合には、直接ベクトルキャッシュからベクトルレジスタへデータ転送を行う。

メモリアクセスの対象となるデータがすでにベクトルキャッシュに格納されている場合には、ベクトルキャッシュからベクトルレジスタに高いメモリバンド幅でデータを供給することができ、実効メモリバンド幅が向上する。またベクトルキャッシュは、実効メモリバンド幅の向上だけでなく、メインメモリアクセスをベクトルキャッシュアクセスに置き換えることで、メモリのアクセスによる消費電力の削減が期待できる。さらに、実行時間を短縮することにより、システム全体の消費電力の削減が期待できる。

### 2.3 MS HR

MSHR は、メモリアクセス情報を保持するためのレジスタ機構であり、冗長なメモリアクセスを削減し、実効メモリバンド幅を向上させる。メモリアドレス、命令アドレス、フォーマット情報、有効判定ビットを MS HR に保持する。ロード命令が発行され、そのデータがベクトルレジスタに到着する前に、再びロード命令が発行された場合、後続のロード命令に対して、MSHR では先行するロード命令のメモリアドレスの有効判定ビット、およびメモリアドレスを参照し、後続のロード命令と同じデータが転送中か否かを確認する。後続のロード命令と同じデータが転送中ではない場合、後続のロード命令の情報を MSHR に書き込み、メインメモリにアクセスする。後続のロード命令と同じデータが転送中の場合、メモリアクセスを行わず、代わりに、後続のロード命令の命令アドレスを MSHR に保持する。データの到着時に MSHR 内の命令アドレスを参照し、データをベクトルレジスタに転送する。これにより、転送中のデータと重複する冗長なデータ転送を削減し、実効メモリバンド幅を向上させることができる。また MSHR は、実効メモリバンド幅の向上だけでなく、冗長なメモリアクセスを削減することで、メインメモリのアクセスに要する消費電力の削減も期待できる。さらに、冗長なメモリアクセスの削減によって実行時間が短縮されるため、システム全体の消費電力の削減も期待できる。

### 2.4 メインメモリシステム

本報告で想定するメインメモリシステムは、ベクトルプロセッサの外部にあるメインメモリと、メモリネットワーク、およびメモリコントローラから構成される。メインメモリは実

装の制約により、プロセッサの近くに配置することができず、プロセッサとメモリ間の距離は非常に長くなる。そのため、データ転送に大きなエネルギーを消費する。また、大容量のメインメモリは、データの読み出しや書き込みに大きなエネルギーを消費する。さらに、近年の半導体加工技術の進歩に伴い、単位面積あたりに実装されるトランジスタ数が増加し、メモリシステムの静的エネルギーも大きなものとなっている<sup>4)</sup>。

前述のとおり、ベクトルキャッシュとMSHRによりメモリシステムの消費エネルギーを削減できる可能性がある。しかしながら、ベクトルキャッシュやMSHR自体もその動作や静的電力によってエネルギーを消費するため、それらを考慮した消費エネルギー評価が必要である。

これまで、これらベクトルキャッシュとMSHRは性能の観点からの議論は行われている<sup>2)</sup>が、消費エネルギーの観点からは評価は行われていない。そこで本報告では、システム全体の消費エネルギーをモデル化し、定量的に評価する。

### 3. CMVP とメモリシステムの消費エネルギーモデル

本章では、前章で述べたCMVPとメモリシステムの動作に基づき、CMVPとメモリシステムの消費エネルギーモデルを構築する。本消費エネルギーモデルでは、メモリシステムとCMVPの構成ごとに消費エネルギーを定式化する。各構成要素の消費エネルギーを合算することによってシステム全体の消費エネルギーを得ることができる。

はじめに、コアの消費エネルギーのモデルについて述べる。本モデルにおいて、コアはプログラムの実行中に常に動作すると仮定し、動的電力と静的電力を合わせてコアの消費電力  $P_{core}$  とする。コアの消費エネルギー  $E_{core}$  は、 $P_{core}$ 、コア数  $N_{core}$ 、および実行時間  $T$  より、次式で表す。

$$E_{core} = N_{core} \times P_{core} \times T \quad (1)$$

コア数が増加することで  $N_{core}$  が増加する一方で、ピーク演算性能が向上することで  $T$  が削減される。したがって、高い演算性能を必要とするアプリケーションでは、コア数が増加しても実行時間が短縮されるため、 $E_{core}$  の増加は抑えられる。

次に、メモリシステムの消費エネルギーのモデルについて述べる。想定するメモリシステムの消費エネルギーは、メインメモリへのアクセスによる動的エネルギーと、読み書き動作と無関係に生じるメインメモリの静的エネルギーの二つに分類される。本モデルでは、メモリシステムの消費エネルギー  $E_{mem}$  を、一回のメモリアクセスにおける動的エネルギー (読み出し  $E_r$ 、書き込み  $E_w$ ) と、静的エネルギー (静的電力  $P_{leak}$  と  $T$  の積) の合計により算出する。メインメモリの読み出し回数  $N_r$  と書き込み回数  $N_w$  より、 $E_{mem}$  を次式で表す。

$$E_{mem} = N_r \times E_r + N_w \times E_w + P_{leak} \times T \quad (2)$$

次に、ベクトルキャッシュとMSHRの消費エネルギーのモデルについて述べる。ベクトルキャッシュの消費エネルギー  $E_{onchip}$  は、ベクトルキャッシュ自身の動的エネルギー (読み出し  $E'_r$ 、書き込み  $E'_w$ ) と静的エネルギー  $P'_{leak} \times T$  から成る。ベクトルキャッシュの読み出し回数  $N'_r$  と書き込み回数  $N'_w$  より、 $E_{onchip}$  を次式で表す。なお、ベクトルプロセッサ・ベクトルキャッシュ間のデータ転送により消費するエネルギー、およびベクトルキャッシュへのデータ参照に必要なエネルギーは、メモリシステムやコアの消費エネルギーと比較して微小であることから、本モデルでは省略する。

$$E_{onchip} = N'_r \times E'_r + N'_w \times E'_w + P'_{leak} \times T \quad (3)$$

また、MSHRの消費エネルギー  $E_{mshr}$  は、MSHR自身の動的エネルギー (データ情報の保持  $E_{mshr-hold}$ 、アドレスの保持  $E_{mshr-hit}$ ) と静的エネルギー  $P_{mshr-leak} \times T$  から成る。MSHRのデータの保持回数  $N_{mshr-hold}$  とMSHRのアドレス保持回数  $N_{mshr-hit}$  より、 $E_{mshr}$  は次式で表す。なお、MSHRへのデータ照合の消費エネルギーはメモリシステム・コアの消費エネルギーと比較して微小であることから、本モデルでは省略する。

$$E_{mshr} = N_{mshr-hold} \times E_{mshr-hold} + N_{mshr-hit} \times E_{mshr-hit} + P_{mshr-leak} \times T \quad (4)$$

式(2)より、ベクトルキャッシュとMSHRの利用によって  $N_r$  が  $N'_r$  に置き換わることで、 $E_{mem}$  が削減されることが分かる。ベクトルキャッシュの容量が増加し、より多くの  $N_r$  が  $N'_r$  に置き換わることで、より  $E_{mem}$  が削減される。一方で、式(3)より、ベクトルキャッシュの利用により  $N_r$  が  $N'_r$  に置き換わり、 $E_{onchip}$  が増加する。また、式(4)より、MSHRの利用により  $N_{mshr-hold}$  と  $N_{mshr-hit}$  が増加し、 $E_{mshr}$  が増加する。加えて、式(1)~(4)より、ベクトルキャッシュの利用によりデータ転送性能が向上し  $T$  が短くなることで、 $E_{core}$ 、 $E_{mem}$ 、 $E_{onchip}$ 、 $E_{mshr}$  が削減されることが分かる。

CMVPの消費エネルギー  $E_{total}$  は以下の式で表すことができる。

$$E_{total} = E_{mem} + E_{onchip} + E_{mshr} + E_{core} \\ = N_{mshr-hold} \times E_{mshr-hold} + N_{mshr-hit} \times E_{mshr-hit} \\ + N_r \times E_r + N'_r \times E'_r + N_w \times E_w + N'_w \times E'_w \\ + (P'_{leak} + P_{leak} + P_{mshr-leak} + N_{core} \times P_{core}) \times T \quad (5)$$

ベクトルキャッシュの容量を変更することにより、 $N_r$ 、 $N'_r$ 、 $N'_w$ 、 $T$  が変化する。また、コア数を変更することで、 $N_{core}$ 、 $T$  が変化する。

表 1 シミュレータのパラメータ

パラメータ名	設定値
コア数	1-16
ベクトルキャッシュ	SRAM
ベクトルキャッシュの容量	512KB-32MB
ラインサイズ	8B
置換ポリシー	LRU
ウェイ数	2
メモリバンド幅 (ベクトルキャッシュ・コア間)	64GB/s/core
メモリバンド幅 (メインメモリ・ベクトルキャッシュ間)	64GB/s
プロセス	90nm
MSHR エントリ数	8192

## 4. 評価

### 4.1 評価環境

3章で述べた消費エネルギーのモデルを用いて、ベクトルキャッシュの容量、コア数がCMVPの消費エネルギーにどのような影響を与えるかを評価する。本報告で想定するCMVPは、ベクトル型スーパーコンピュータであるNECのSX-8<sup>5)</sup>を基に、チップ内に複数のコア、ベクトルキャッシュ、MSHRを搭載したものと<sup>2)</sup>ベクトルキャッシュおよびMSHRの一回のアクセスにおける動的エネルギー( $E'_r, E'_w, E_{mshr-hold}, E_{mshr-hit}$ )と静的電力( $P'_{leak}, P_{mshr-leak}$ )は、CACTI6.5<sup>6)</sup>により算出する。

メインメモリ、ベクトルキャッシュとMSHRそれぞれへのアクセス回数( $N_r, N'_r, N_w, N'_w, N_{mshr-hold}, N_{mshr-hit}$ )と実行時間は、NECのベクトルプロセッサ用トレース駆動型シミュレータを用い、表1に示すパラメータに基づいて算出される。また、アプリケーション中の多重ループにおいて、最外ループをマルチスレッド化し複数のコアにより並列処理を行う。

本評価では、東北大学サイバーサイエンスセンターで利用されている各研究分野の代表的な実アプリケーションを用いる。表2に評価アプリケーションを示す。

### 4.2 消費エネルギー評価

本節では、ベクトルキャッシュの容量とコア数を変更し、実アプリケーションの実行に要する消費エネルギーを評価する。

図2に、ベクトルキャッシュの容量変更時の各アプリケーションの消費エネルギーの評価結果を示す。縦軸に消費エネルギー、横軸に各アプリケーションのベクトルキャッシュの容量、およびコア数を示す。消費エネルギーは、シングルコアでベクトルキャッシュを利用しない場合の値で正規化する。評価結果より、ベクトルキャッシュの容量が小さい場合には、容量の増加に伴って消費エネルギーが減少することが分かる。しかし、ベクトルキャッシュの容量が一定以上大きい場合には、容量の増加に伴って消費エネルギーが増加することが分かる。

ベクトルキャッシュの容量が増加するにつれ、ベクトルキャッシュに格納可能なデータが増え、ベクトルキャッシュから転送可能なデータ量が増加する。その結果、メインメモリからベクトルキャッシュへのアクセスの置き換え回数が増加し、データ転送にかかる消費エネルギーが抑えられる。しかし、ベクトルキャッシュの容量が再利用可能なデータを格納するために必要な容量に達すると、ベクトルキャッシュのヒット率が上限に達し、メモリシステムの消費エネルギーが削減されない。その結果、ベクトルキャッシュの容量の増加分だけ消費エネルギーが増加し、消費エネルギーが削減から増加に転じる。図3に、図2の16コアの場合のベクトルキャッシュのヒット率を示す。縦軸にベクトルキャッシュのヒット率、横軸に各アプリケーションを示す。図3より、8MBと32MBのヒット率の差は0%~約2.1%であり、ヒット率が上限に達していることが確認できる。また、コア数の増加に伴ってベクトルキャッシュに接続されるポート数も多くなるため、回路や配線長が増加することによってベクトルキャッシュの消費エネルギーが増加する。

次に、コア数と消費エネルギーの関係について考察する。図2より、コア数が少ない場合には、コア数の増加に伴って消費エネルギーが減少することが分かる。これは、コア数を増やして実行時間を短縮したことによる消費エネルギー削減が、コア数増加による消費エネルギーの増加を上回ったためである。一方、コア数がある程度多くなると、コア数の増加に伴って消費エネルギーも増加する。これは、コア数が十分に多くなったために、それ以上コア数を増やしても性能が向上せず、コア数の増加の分だけ消費エネルギーが増えるためである。

最後に、各アプリケーションの消費エネルギーを比較する。図2では、ベクトルキャッシュの容量増加による消費エネルギーの削減量が、SFHTで最大約13%、PRFで最大約53%、APFAで最大約54%、GPRで最大約54%、PBMで最大約40%であり、SFHTの消費エネルギー削減効果が低いことが分かる。そこで次節では、消費エネルギー削減傾向が低いアプリケーションとしてSFHT、および消費エネルギー削減傾向が高いアプリケーションの代表としてAPFAの傾向について分析する。



表 2 評価アプリケーション

研究分野	名前	シミュレーションの詳細	手法	問題サイズ	ベクトル長	ベクトル化率
熱・流体力学解析	PRF <sup>7)</sup>	水素と空気の予混合火炎	DNS	513 × 513	179.0	99.3%
	SFHT <sup>8)9)</sup>	飛行機表面の熱伝導と乱流剥離	SMAC	711 × 91 × 221	192.9	99.4%
電磁解析	GPR <sup>10)11)</sup>	地雷探査用アレイアンテナの探査性能	FDTD	50 × 750 × 50	245.1	99.7%
	APFA <sup>12)</sup>	フェルミアンテナの指向性の評価	FDTD	612 × 105 × 505	255.5	99.9%
地球科学	PBM <sup>13)</sup>	地震によるプレート境界のすべり	Friction Law	32400 × 32400	255.5	99.5%

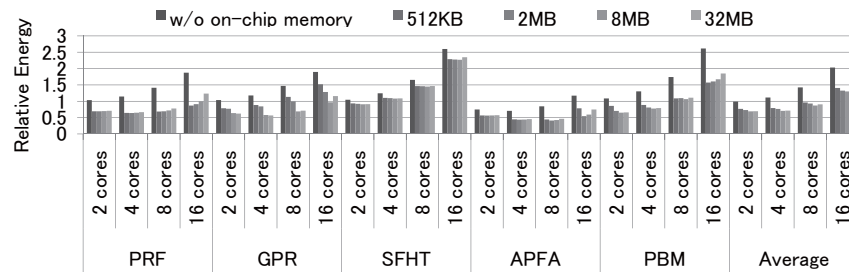


図 2 実アプリケーションの消費エネルギーの算出結果

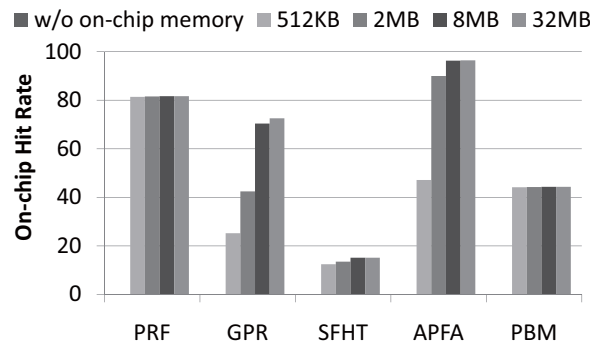


図 3 ベクトルキャッシュのヒット率 (16 コア)

#### 4.2.1 SFHT の消費エネルギー削減傾向の分析

図 4 に、SFHT の主要カーネルのプログラムを示す。本評価では、3 重ループの最外ループが並列化されている。参照の局所性のあるデータは三次元配列の Phi である。このデータをベクトルキャッシュに格納することによって、ベクトルキャッシュのヒット率は約 15% となる。

図 5 に、図 2 における SFHT の消費エネルギーの内訳を示す。この時、ベクトルキャッシュの容量は 8MB である。図 5 より、ベクトルキャッシュを利用しない場合の各コアのメモリシステムの静的エネルギーが、シングルコアのメモリシステムの静的エネルギーと比較して約 12% ~ 約 16% 削減されており、コア数の増加に伴う静的エネルギーの削減量の増加が少ないことが分かる。

SFHT の主要カーネルにおけるベクトル演算命令数とベクトルロード命令数の比は約 1.0 であり、ベクトルロード命令数の割合が多いため、実行時間のうちデータ転送処理時間を占める割合が高い。よって、コア数の増加によりピーク演算性能が向上しても、その演算性能に見合ったデータ転送性能を持っていないために、実行時間が短縮されない。

また、ベクトルキャッシュがない場合と比較して、ベクトルキャッシュ利用時の読み出しの動的エネルギーの削減量は約 15% である。さらに、ベクトルキャッシュを利用しない場合とベクトルキャッシュ利用時の各コアのメモリシステムの静的エネルギーを比較すると、SFHT では削減量が最大の 16 コアで約 16%、削減量が最小の 2 コアで約 14% とほぼ一定である。SFHT は実行時間のうちデータ転送処理時間の占める割合が高いため、ベクトルキャッシュにより実効メモリバンド幅が向上することで実行時間が短縮する。その結果、静的エネルギーが減少する。しかし、ベクトルキャッシュにより実効メモリバンド幅を向上しても、演算性能に対するメモリバンド幅が十分ではないため、コア数増加により実行時間が短縮する。その結果、静的エネルギーが減少する。

#### 4.2.2 APFA の消費エネルギー削減傾向の分析

図 6 に、APFA の主要カーネルのプログラムを示す。本評価では、APFA の主要カーネルは最外ループで並列化されている。すべての配列において参照の局所性があり、これらのデータをベクトルキャッシュに格納することによって、ベクトルキャッシュのヒット率は約 96% となる。

図 7 に、図 2 における APFA の消費エネルギーの内訳を示す。なお、ベクトルキャッシュの容量は 8MB である。図 7 より、ベクトルキャッシュを利用しない場合、コア数の増加に伴っ

```

do k=KST(1),NK,2
do j=JST(1),NJ,2
do i=IST(1),NI,2
res=(Phi(i+1,j,k)*AE(i,j,k)+AN(i,j,k)
& *Phi(i,j+1,k)+AS(i,j,k)*Phi(i,j-1,k)
& +AT(i,j,k)*Phi(i,j,k+1)+AB(i,j,k)*Phi(i,j,k-1)
& +AP(i,j,k)*Phi(i,j,k)+AW(i,j,k)
& *Phi(i-1,j,k)-DIV(i,j,k))/AP(i,j,k)
res=res*RGNC(i,j,k)
Phi(i,j,k)=Phi(i,j,k)-res

```

図4 SFHTの主要カーネルプログラム

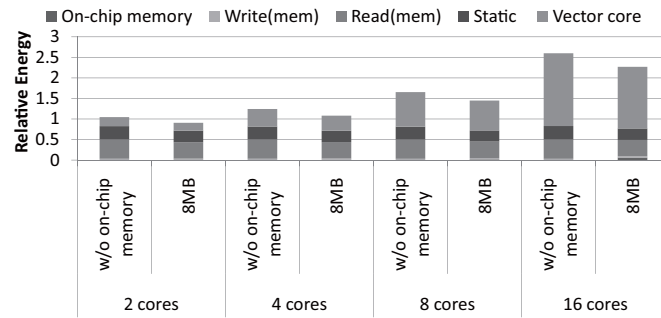


図5 SFHTの消費エネルギーの内訳

て、メモリシステムの静的エネルギーが減少することが分かる。シングルコアのメモリシステムの静的エネルギーと比較して、2コアで48%、4コアで66%、8コアで72%、16コアで75%静的エネルギーが削減されており、コア数の増加に伴いメモリシステムの静的エネルギーが少ないことが分かる。

APFAの主要カーネルにおけるベクトル演算命令数とベクトルロード命令数の比は約4.7であり、ベクトル演算命令数の割合が多いため、実行時間のうち演算処理時間の占める割合が高い。このため、コア数の増加によって、ピーク演算性能の向上に伴って実行時間が短縮する。その結果、コア数の増加に伴って、実行時間がより短縮されメモリシステムの静的エネルギーが減少する。

また、図7より、ベクトルキャッシュ利用時のメモリの読み出しに要する静的エネルギーは、ベクトルキャッシュを利用しない場合と比較して約95%削減されていることが分かる。さら

```

do ithxyz=1,nag*3
do kio=ki1,ki2,jcbs
do if=1,nf
do ki=kio,min(kio+jcbs,ki2)
CC1=CDEXP(A*m1(ki,ithxyz))*(X1(ki,ithxyz)+B*X2(ki,ithxyz)+C*X3(ki,ithxyz))
CC2=CDEXP(D*m2(ki,ithxyz))*(Y1(ki,ithxyz)+E*Y2(ki,ithxyz)+F*Y3(ki,ithxyz))
CUX(if,ithxyz)=CUX(if,ithxyz)+DXZC*PIC*(EZ1(ki)*CC1(ki)-EZ2(ki)*CC2(ki))
CUZ(if,ithxyz)=CUZ(if,ithxyz)-DXZC*PIC*(EX1(ki)*CC1(ki)-EX2(ki)*CC2(ki))
CWX(if,ithxyz)=CWX(if,ithxyz)-DXZC*PIC*(HZ1(ki)*CC1(ki)-HZ2(ki)*CC2(ki))
CWZ(if,ithxyz)=CWZ(if,ithxyz)+DXZC*PIC*(HX1(ki)*CC1(ki)-HX2(ki)*CC2(ki))

```

図6 APFAの主要カーネルプログラム

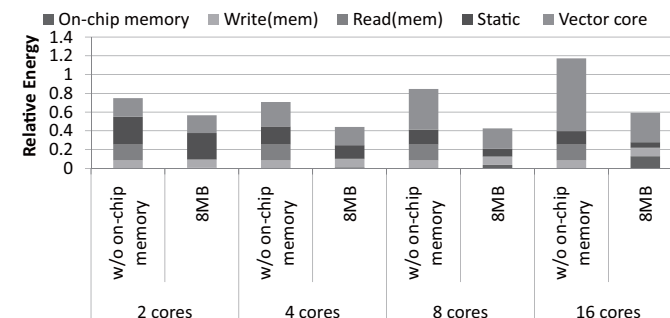


図7 APFAの消費エネルギーの内訳

に、16コアの時に静的エネルギーが約60%削減され、コア数の増加に伴ってメモリシステムの静的エネルギーの削減量が大きくなる事が分かる。

ピーク演算性能が高い場合、ベクトル演算命令数とベクトルロード命令数の比が高いAPFAにおいてもメモリバンド幅が不足する。このため、コア数が多い場合には、ベクトルキャッシュを利用することによって実効メモリバンド幅が大幅に向上する。その結果、コア数が多い場合に実行時間がより短縮され、メモリシステムの静的エネルギーが大きく削減される。

#### 4.2.3 消費エネルギー削減傾向の比較

ベクトルキャッシュのヒット率は、APFAが約95%、SFHTが約15%であり、APFAの方がより多くメインメモリをベクトルキャッシュアクセスに置き換えている。しかし、2コアにおけるメモリシステムの静的エネルギー削減量は、APFA約5%、SFHT約15%であり、APFAがSFHTよりも静的エネルギーの削減量が少ない。しかし、2コアではAPFAの方が

より多くメインメモリをベクトルキャッシュアクセスに置き換えているにもかかわらず、実行時間の短縮は SFHT の方が高い。

APFA の主要カーネルでは演算が支配的であるため、ベクトルキャッシュの効果は小さい。一方、SFHT における主要カーネルではメモリアクセスが支配的であるため、ベクトルキャッシュの効果が大きい。そのため 2 コアでは、ベクトルキャッシュのヒット率の高い APFA は、ベクトルキャッシュのヒット率の低い SFHT より、メモリシステムの静的エネルギーの削減量が低いものとなる。

4.2.1 節～4.2.3 節より、ベクトルキャッシュの容量を増加しヒット率を増加させることで、メモリシステムの動的エネルギーやシステム全体の静的エネルギー、コアの消費エネルギーが削減されることが明らかになった。一方で、ベクトルキャッシュの容量やコア数の増加に伴って、ベクトルキャッシュの消費エネルギー増加が顕著になる。さらに、高い演算性能を必要とするアプリケーションでは、コア数が多いほど、ベクトルキャッシュの消費エネルギー削減効果が高いことが明らかになった。また、高いメモリバンド幅を必要とするアプリケーションでは、コア数によらず、ベクトルキャッシュの消費エネルギー削減効果が高いことが示された。

## 5. ま と め

本報告では、CMVP とメモリシステムの電力モデルを構築した。このモデルを用いて、コア数、ベクトルキャッシュの容量を変更する場合の CMVP とメモリシステムの消費エネルギーを実アプリケーションにより定量的に評価した。また、コア数、ベクトルキャッシュの容量を変化させた場合の、実アプリケーション実行時の性能の評価を行った。評価の結果、ベクトルキャッシュの容量を増加しヒット率を増加させることで、メモリシステムの動的エネルギーやシステム全体の静的エネルギー、コアの消費エネルギーが削減されることが明らかになった。また、アプリケーション毎に、コア数やベクトルキャッシュの容量の最適値が異なることが明らかになった。したがって、CMVP の性能と電力効率のさらなる向上を実現するためには、アプリケーションに応じた動的構成型のアーキテクチャが有効であると考えられる。

本報告では実総面積を考慮せずに CMVP の構成の検討を行っている。チップ上に搭載可能なハードウェアリソースは限られているため、今後は、実装面積を考慮した詳細な評価を行う予定である。さらに、アプリケーションに応じて、動的に適切な CMVP を構成する機構の研究に取り組む予定である。

## 6. 謝 辞

本報告では評価に際し、東北大学サイバーサイエンスセンターのスーパーコンピュータシステムを利用した。本報告の一部は、科学研究費補助金基盤研究 (B) (22700044) , および科学研究費補助金若手研究 (B) (22300013) の助成を受けたものである。また本報告の一部は、科学技術振興事業団 JST の戦略的基礎研究推進事業 CREST の助成を受けたものである。

## 参 考 文 献

- 1) Akihiro Musa, Yoshiei Sato, Takashi Soga, Koki Okabe, Ryusuke Egawa, Hiroyuki Takizawa, and Hiroaki Kobayashi, "A Shared Cache for a Chip Multi Vector Processor," Proceedings of the 9th MEDEA Workshop, pp. 24-29, 2008.
- 2) Akihiro Musa, Yoshiei Sato, Takashi Soga, Ryusuke Egawa, Hiroyuki Takizawa, Koki Okabe, Hiroaki Kobayashi, "Effects of MSHR and Prefetch Mechanisms on an On-Chip Cache of the Vector Architecture," International Symposium on Parallel and Distributed Processing with Applications. pp.335-342 2008.
- 3) David Kroft, "Lockup-Free Instruction Fetch/Prefetch Cache Organization," In The 8th International Symposium on Computer Architecture, pp. 81-87, 1981.
- 4) J. Abam Butts and Gurindar S. Sohi, "A Static Power Model for Architects," Proceedings of The 33rd Annual IEEE/ACM International Symposium on Microarchitecture, pp. 191-201, 2000.
- 5) 篠原真史, 古澤一昭, 西垣泰洋, 多賀谷聡, 鈴木栄司, "SX-8 のハードウェア技術 (1)," NEC 技報, Vol. 58, No. 4, pp. 11-14, 2005.
- 6) Naveen Muralimanohar, Rajeev Balasubramonian, and Norman P. Jouppi. CACTI6.5. HP Laboratories HPL-2009-85, 2009.
- 7) Kazuya Tsuboi and Gorou Masuya, "Direct Numerical Simulations For Instabilities Of Remixed Planar Flames," In The Fourth Asia-Pacific Conference On Combustion, pp. 136-139, 2003.
- 8) Madoka Nakajima, Hideki Yanaoka, Hiroyuki Yoshikawa and Terukazu Ota, "Numerical Simulation of Three-Dimensional Separated Flow and Heat Transfer around Staggered Surface-Mounted Rectangular Blocks in a Channel" Numerical Heat Transfer (PartA), No. 47, pp. 691-708, 2005.
- 9) Hideki Yanaoka, Hiroyuki Yoshikawa, and Terukazu Ota, "Direct Numerical Simulation of Turbulent Separated Flow and Heat Transfer over a Blunt Flat Plate" Journal of Heat Transfer, No. 125, pp. 779-787, 2003.
- 10) Takeo Kobayashi, Xuan Feng, and Motoyuki Sato, "FDTD simulation on array

- antenna SAR-GPR for land mine detection” In Proceedings of SSR2003: 1st International Symposium on Systems and Human Science, pp. 279-283, 2003.
- 11) Motoyuki Sato, Takeo Kobayashi, Zhaofa zeng, Guangyou Fang, and Xuan Feng, “High Resolution GPR System for Landmine Detection” In International Conference Requirements and Technologies for the Detection, Removal and Neutralization of Landmines and UXO, pp. 548-553, 2003.
  - 12) Yukiko Takagi, Hiroyasu Sato, Yoshihiko Wagatsuma, Koji Mizuno, and Kunio Sawaya, “Study of High Gain and Broadband Antipodal Fermi Antenna with Corrugation” In 2004 International Symposium on Antennas and Propagation, Vol. 1, pp. 69-72, 2004.
  - 13) Keisuke Ariyoshi, Toru Matsuzawa, and Akira Hasegawa, “The Key Frictional Parameters Controlling Spatial Variation in the Speed of Postseismic Slip Propagation on a Subduction Plate Boundary” Earth and Planetary Science Letters, No. 256, pp. 136-146, 2007.