Vol.2012-ARC-201 No.12
2012/8/1

三段階対応の可変レベルキャッシュのマルチスレッドアプリケーションを用いた評価

渡部 功[1]　佐々木 敬泰[1]　大野 和彦[1]　近藤 利夫[1]

概要：現在，プロセッサには高性能と低消費エネルギーの両立が求められている．特に，回路の微細化にともないリークエネルギーが増加しているため，リークエネルギーを削減することが重要である．そこで我々は，キャッシュの消費エネルギーを削減する手法の一つとして可変レベルキャッシュを提案している．可変レベルキャッシュはキャッシュの容量をプログラムの特徴によって動的に変化させる手法で，キャッシュ容量が必要でないときには標準電圧で高速動作するアクティブ領域を順次半減させることで消費エネルギーを低減する．また，アクティブでなくなった領域の電源を遮断するのではなくスリープモードへと移行させ，1つ下位レベルの排他的キャッシュとして動作させることでアクティブ領域を減らした時の性能低下を抑制している．これまでに我々は可変レベルキャッシュのマルチプロセッサ環境での有効性を示した．また，従来の可変レベルキャッシュで問題となっていた書戻しオーバヘッドの削減手法も提案してきたが，マルチプロセッサ環境での評価はされていなかった．更に，これまでの評価ではマルチスレッドアプリケーションを用いた評価を行われていなかった．そこで本稿では，マルチプロセッサ環境において，提案されてきた性能向上手法を適用した上で，マルチプロセスおよびマルチスレッドアプリケーションを用いて我々が提案している可変レベルキャッシュの有効性について評価を行う．評価の結果，可変レベルキャッシュはマルチプロセッサ環境において通常キャッシュより平均41%電力遅延積が改善できることが明らかとなった．

キーワード：低電力・高性能キャッシュ，可変レベルキャッシュ，リーク電力

Evaluation of Three-level Variable Level Cache using Multithread applications

Ko Watanabe[1]　Takahiro Sasaki[1]　Kazuhiko Ohno[1]　Toshio Kondo[1]

Abstract: Power dissipation is a major concern not only for mobile computing but also high performance computing, and achieving both low energy and high performance at the same time is required. It is particularly important to reduce leakage energy consumed in a cache memory because power dissipation caused by leakage energy is dominant factor in deep submicron technologies and a cache memory consists of a large number of transistors. So, we propose Variable Level Cache to achieve both low energy consumption and high performance simultaneously. Variable Level Cache analyzes cache performance dynamically and if it detects that the current running program does not need so large capacity of cache memory, half of the active area is put into standby mode, and is treated as a lower level exclusive cache to prevent the performance degradation. Previously, we showed the effectiveness of Variable Level Cache on multiprocessor. Meanwhile, we also solved writeback overhead that was a serious problem of Variable Level Cache. However, we didn't evaluate that solution on multiprocessor. Therefore, in this paper, we evaluate Variable Level Cache with that solution on multiprocessor. In addition, we also evaluate it with using multi thread applications. According to the simulation results, the performance of Variable Level Cache is about 41% superior to that of normal cache in the energy-delay product on multiprocessor.

Keywords: Low-power and high-performance cache, Variable level cache, Leakage energy.

1. はじめに

現在，ノートパソコン，PDA，携帯電話などのモバイル端末の高性能化にともない消費エネルギーが増大し，バッテリによる駆動時間が短くなるという問題が発生している．そこで，モバイル端末の性能を落とすことなく低消費エネルギーを実現することが要求されている．

プロセッサで消費されるエネルギーは動的消費エネルギーと静的消費エネルギーに分けられる．動的消費エネルギーはトランジスタのスイッチングによって消費されるエネルギーである．一方，静的消費エネルギーはトランジスタの漏れ電流 (リーク電流) によって引き起こされ，トランジスタのスイッチングに関係なく消費されるエネルギーで，リークエネルギーともいう．近年，回路の微細化にともなって，動的消費エネルギーが削減される一方，リークエネルギーが増加している．リークエネルギーはトランジスタ数に比例するため，プロセッサの高性能化に伴って容量が増大したキャッシュシステムのリークエネルギー削減が重要である．

高性能と低消費エネルギーの両立を目指すキャッシュシステムは様々なものが提案されているが，その手法の一つとして我々は可変レベルキャッシュ [1] [2] を提案している．可変レベルキャッシュはキャッシュの容量をプログラムの特徴によって動的に変化させる手法で，キャッシュ容量が必要でないときには標準電圧で高速動作するアクティブ領域を順次半減させることで消費エネルギーを低減する．また，アクティブでなくなった領域の電源を遮断するのではなくスリープモードへと移行させ，1 つ下位レベルの排他的キャッシュとして動作させることでアクティブ領域を減らした時の性能低下を抑制している．これまでに我々は可変レベルキャッシュのマルチプロセッサ環境での有効性を示すと同時に，レベル切換時に発生するライン書戻しのオーバヘッド削減手法 [14] や切換可能レベルの拡張 [15] など，可変レベルキャッシュの性能を向上させる手法を提案してきた．しかしながら，マルチプロセッサ環境において，それらの性能向上手法を適用した場合の有効性については明らかになっていない．また，これまでの評価ではマルチスレッドアプリケーションを用いた評価を行えていなかった．そこで本稿では，マルチプロセッサ環境において性能向上手法を適用した上で，マルチプロセスおよびマルチスレッドアプリケーションを用いて可変レ ベルキャッシュの有効性について評価を行う．

2. 関連研究

これまでにキャッシュの様々なリークエネルギー削減手法が提案されてきた．これらの手法は通常状態と待機状態を切換える単位で，大きく 2 つに分離することができる．

1. ライン単位の状態切換
   1 つ目は，ライン単位で通常状態と待機状態を切り替えるものである．[3][4][5][6]

   代表的な研究として，Drowsy Cache[4][5] やウェイ予測キャッシュ [6] が挙げられる．Drowsy Cache は定期的に全てのキャッシュ・ラインへの電源電圧を下げ，アクセスが発生したラインに対してのみ電源電圧を回復する事でリークエネルギーを削減する手法である．ウェイ予測キャッシュは，アクセス開始前に最も最近アクセスされたウェイ情報 (MRU) を利用して，参照データが存在するウェイを予測し，選択的に活性化する事で低消費エネルギー化を行う手法である．

   これらのようなライン単位で切り換える手法は，キャッシュの状態を細かく切り換えることが可能である一方，センスアンプ等のライン以外にかかる電力を削減できないといった問題がある．また，1 つのセット内で通常状態のラインと待機状態のラインが混在する状態となる，すなわち連想度が落ちてしまうという問題がある．

2. バンク単位の状態切換
   もう 1 つは，DRI キャッシュ [7] のようにキャッシュシステム内を複数のバンクに分割し，バンク単位で通常状態と待機状態を切り換えるものである．このタイプは，ライン単位で切り換える手法のように状態の細かな切り換えが出来ない反面，待機状態時にセンスアンプ等のライン以外の電力削減が可能となる．

   バンク単位で切り換えを行う手法は，バンクを構成する際にセット単位，ウェイ単位のどちらでも可能であるが，本稿ではセット単位でバンクを構成する事を考える．ウェイ単位で切り換える事はライン単位で切り換えるのと同様に実効的に連想度を低下させてしまうが，セット単位の切り換えの場合，各セット内の連想度が通常状態と同じに保たれるという利点があるからである．次項で説明する DRI キャッシュにおいても同様に扱う．

2.1 DRI キャッシュ

2.1.1 DRI キャッシュの概要

図 1 に DRI キャッシュの概要図を示す．DRI キャッシュはある一定時間間隔 (interval) でキャッシュミス数をカウントする (miss counter)．そして，ミス数がある閾値 (miss-bound) より小さい場合には，キャッシュ・サイズを縮小しても性能には大きな影響を与えないと判断する．一方，ミス数が閾値よりも大きい場合にはキャッシュ・サイズを増大して性能低下を防ぐ．キャッシュ・サイズを減らす場合はその時の容量の半分にし，逆にキャッシュ・サイ

[1] 三重大学大学院工学研究科情報工学専攻
Graduate School of Engineering, Mie University

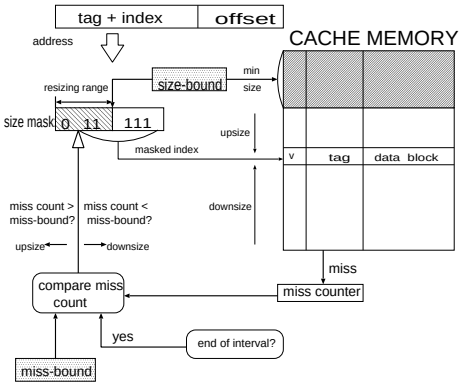

図 1　DRI のブロック図

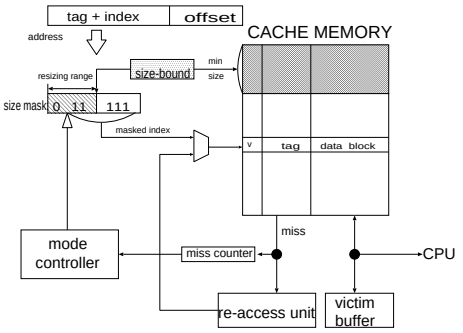

図 2　可変レベルキャッシュのブロック図

ズを増やす場合は倍にする．これを複数段階に分けて実装を行う事で，キャッシュ・サイズを必要に応じて変更する．例えば，動的に 256KB，128KB，64KB，32KB のキャッシュ・サイズに変更することができる．キャッシュラインへのアクセスは，アドレスにマスク (size mask) を掛ける事でキャッシュ・サイズの変化に対応させている．

このようにして DRI キャッシュでは，キャッシュ・サイズを縮小した場合の未使用領域の SRAM セルに対して電源電圧の供給を停止することによってリークエネルギーの削減をしている．また，DRI キャッシュは電源を切る部分のキャッシュを 1 つのバンクとして電源管理を行う事で，センスアンプやアドレスデコーダなどの，SRAM セル以外の回路の電源も切る事が出来るメリットがある．

2.1.2　DRI キャッシュの問題点

従来の DRI キャッシュでは，キャッシュサイズを縮小した場合，未使用領域の SRAM セルに対して電源電圧の供給を停止することでリークエネルギーを削減している．したがって，電源を落とした部分に格納されていたデータは破棄されるためキャッシュヒット率が低下するという問題がある．

更に，DRI キャッシュは命令キャッシュ用に開発された手法であり，当該手法をデータキャッシュに適用する場合，縮小する部分への電力の供給を完全に停止し，データを破壊するため，その部分にあるデータを下位の記憶層へと書戻す必要がある．また，キャッシュサイズによってデータの配置が異なるために，キャッシュサイズを増大させる場合には，現在キャッシュに入っているデータを下位の記憶層へと書戻さなければならない．DRI キャッシュをデータキャッシュに適用するとこれらの処理によって性能へ悪影響を与えるという問題がある．

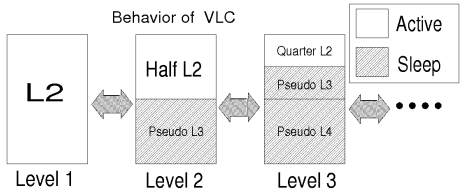

図 3　可変レベルキャッシュのレベル切換動作

3.　可変レベルキャッシュ

DRI キャッシュの上記の問題点を解決する手法として，我々はメモリのスタンバイモード (以降スリープモードと呼ぶ) を利用し，かつ待機状態への切換時の書戻しを抑制することで性能を向上させる可変レベルキャッシュを提案している．

3.1　可変レベルキャッシュの概要

可変レベルキャッシュ [1][2] は，DRI キャッシュと同様に，ある一定時間間隔でキャッシュミス数をカウントする事で，キャッシュへの要求性能を動的に判断し，キャッシュ容量を増減させる．しかし，DRI キャッシュのように単純に容量を減少させるだけでは，キャッシュミス回数が増加してしまう．そこで，容量の半分の電源供給を遮断するのではなく，スリープモードに落とす．スリープモードとは，電源の供給を完全に遮断するのではなく，データの内容が破壊されない程度に電源電圧を下げた状態の事をいう．スリープモードは電源供給を停止する場合よりはリークエネルギーの削減率が低くなるが，通常モードより大幅にリークエネルギーが削減可能で，データが保持できるというメリットがある．しかし，スリープモードとなっている領域へのアクセスは通常のアクセスより時間がかかるため，スリープモードになっている領域へアクセスが多発する場合，性能が悪化する恐れがある．そこで，スリープモードの領域を擬似的に 1 つ下位レベルの排他的キャッシュ（詳細は第 3.2 項で述べる）とすることでスリープモード領域へのキャッシュアクセスを減らし，性能低下を抑制している．可変レベルキャッシュの動作の概要を図 3 に示す．以後，容量および擬似的階層に関わらず，通常電圧で動作している領域をアクティブ領域，スリープモードになっている領域をまとめてスリープ領域と呼ぶ．以降では図 3 同様，全てのラインがアクティブの時を「Level 1」，アクティブ領域が全体容量の半分の時を「Level 2」, アクティブ領域の容量が全体の 1/4 で，疑似 L4 キャッシュまでの階層に分かれている時を「Level 3」と呼ぶ．

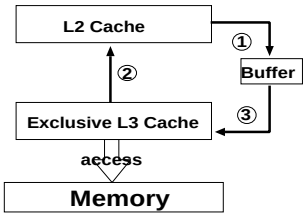

図 4　排他的キャッシュ

256KB の L2 キャッシュに可変レベルキャッシュを適用した場合，性能があまり必要でないと判断したときは Level 2 へ切替わり，容量の半分である 128KB はスリープモードへと移行し，擬似的な L3 キャッシュとして動作する．この時，この疑似 L3 キャッシュは排他的キャッシュとして動作させる．Level 2 においてもキャッシュの容量がまだ多いと判断したときは Level 3 へ切替わる．このとき，アクティブ領域と疑似 L3 キャッシュの容量は全体の 1/4 である 64KB，疑似 L4 キャッシュの容量は 128KB である．

図 2 に可変レベルキャッシュの詳細図を示す．可変レベルキャッシュは主に，DRI キャッシュの回路に再アクセスユニット (re-access unit) とバッファ (buffer) を加えた形で構成される．Level の変更はマスクのビットを Level に合わせて変更し，参照するインデックスを制限することで実現している．再アクセスユニットとバッファは共に Level 1 以外の状態で利用し，再アクセスユニットはアクティブ領域でキャッシュミスをした場合に，スリープ領域をアクセスするために利用される．バッファは排他的キャッシュとしてアクティブ領域のリプレイスデータとスリープ領域のデータとのスワップを行うときに用いる．

可変レベルキャッシュの問題点の一つとして，アクティブ領域を増やす場合（例えば Level 2 から Level 1 に切替わる時）にキャッシュメモリ内に本来とは異なるセットに存在するラインが発生してしまい，キャッシュメモリ内のデータをメモリへ書戻す必要がある．この書戻しオーバヘッドを削減する手法については第 3.3 項に記述する．

3.2　排他的キャッシュ

排他的キャッシュ (Exclusive cache) は AMD 社が開発したキャッシュアーキテクチャで，L1 キャッシュと L2 キャッシュのデータを排他的にすることでキャッシュメモリを有効に利用する手法である．

以降の説明で，キャッシュの構成を L1 キャッシュは 128KB，L2 キャッシュは 256KB と想定する．

従来のキャッシュでは，すべてのデータはまず L2 キャッシュに格納され，その後 L2 キャッシュから L1 キャッシュへとコピーされる．そのため，L1 と L2 の割当てがあっても全体的なキャッシュ・サイズは L2 キャッシュに相当する 256K バイトであるといえる．それに対し，排他的キャッシュでは，図 4 のように，L1 キャッシュから L2 キャッシュへと書戻されるデータを一度バッファに移し (1)，その後 L2 キャッシュから必要なデータをロードし (2)，最後にバッファのデータを L2 キャッシュに書戻している (3)．このように，データを L1 と L2 との間で交換する事 (排他的に処理する事) で全体的なキャッシュ・サイズが 128+256=384K バイトとなり，キャッシュサイズを有効に活用できる．

3.3　書戻しオーバヘッド削減手法

可変レベルキャッシュでは，Level 2 や Level 3 として動作している時の排他的キャッシュによるライン交換によって，アクティブ領域を増加した場合に本来とは異なるセットに存在するラインが発生する．我々はそのラインを「不良配置ライン」と呼んでいる．不良配置ラインが存在してしまうと，通常のアクセス方法では不良配置ラインへアクセスが出来なくなるため，不良配置ラインを下位メモリに書戻す必要がある．しかし，Level を切換える毎に書戻しが発生すると性能に大きな影響を及ぼすため，可変レベルキャッシュには不良配置ラインの書戻しを低減する手法を実装している．図 5 は Level 3 から Level 2 を介して Level 1 に切替わった直後の 4 セット，4 ウェイのキャッシュ内のライン状態を示している．ライン A0-1〜A0-4 は BankA 内に存在すべきライン，同様に B0-1〜B04, C01〜C1-4 はそれぞれ BankB, C に存在すべきラインである．また，四角で囲ったライン A0-2，A0-4，B0-3，C0-1 が不良配置ラインであり，存在すべきラインにいないため，このままでは不良配置ラインへアクセスが出来ない．そこで，不良配置ラインへのアクセスを可能にするために，各セット毎に切替え可能な Level の数だけ NoD カウンタを設ける．NoD カウンタは自セットの対応するセット内に存在する可能性がある不良配置ラインの数を記憶する．図 5 の例では可変レベルキャッシュは Level 3 まで対応しているため，NoD カウンタを 2 つ持っている．インデックス 00 において NoD 1 はインデックス 10，NoD2 はインデックス 01 のそれぞれセット内の不良配置ラインの数を記憶する．ライン A0-4 へアクセスが発生した場合の動作は以下の通りである．

ステップ 1: ライン A0-4 が本来存在すべきセットであるインデックス 00 へアクセスしミスヒットする．

ステップ 2: NoD1 を参照し，値が 0 ではないので，対応するセットであるインデックス 10 へアクセスしミスヒットとなる．

ステップ 3: NoD2 を参照し，値が 0 ではないので，対応するセットであるインデックス 01 へアクセスし、ヒットとなる．

NoD カウンタはセット毎に log(ウェイ数)+1　ビット必要であるが，タグメモリやデータメモリと比較してハード

ウェア規模は十分に小さいといえる (4 ウェイキャッシュの場合セット毎に 3 ビット必要).

3.4 従来の評価方法

我々はこれまでに，可変レベルキャッシュをマルチプロセッサ環境に適応した場合の有効性を示している [12]．また，その後，NoD カウンタを用いた不良配置ライン書戻しオーバーヘッドを削減する手法 [14]，その手法を改良した切換可能レベルの拡張 [15] をそれぞれ提案し，その有効性を示してきた．しかしながら，文献 [12] ではマルチプロセス，すなわち異なるプログラムをそれぞれのコアで実行した場合の評価のみを行っており，マルチスレッドアプリケーションを用いた評価を行えていない．また，文献 [14], [15] で提案した性能向上手法はシングルプロセッサ上でのみしか評価が行われておらず，マルチプロセッサ環境に適応した場合の有効性は明らかとなっていない．

そこで本稿では，マルチプロセッサ環境においてこれらの性能向上手法を適用した上で，マルチプロセス・マルチスレッドアプリケーションの両方を用いて可変レベルキャッシュの有効性を明らかとする．

4. 評価環境

4.1 評価対象

性能評価を行うに先立ち，本稿において議論する対象となる 3 つの手法について述べる．

4.1.1 従来のマルチプロセッサ対応型可変レベルキャッシュの手法

文献 [12] の可変レベルキャッシュは第 3.3 項で説明した書戻しオーバヘッド低減手法が提案される以前のものである．そのため，書戻し回数を抑えるためレベル切替え頻度を抑えるコントローラを実装していた．このコントローラでは全キャッシュアクセスの回数を数え，一定キャッシュアクセス毎に各プロセッサでキャッシュミス率を測定し，レベル切換判定を行う．各プロセッサのキャッシュミス率を出した後の状態遷移を表 1 示す．表のように全てのプロセッサでキャッシュミス率が閾値を下回った場合のみ Level を上げ，逆に全プロセッサで上回れば Level を下げることでレベル切替えの頻度を抑える．その他場合は状態を維持する．以降の評価において，このコントローラを採用し，文献 [14], [15] で提案している性能向上手法を適用していない可変レベルキャッシュを「Conventional」と呼ぶ．

表 1 状態遷移

Core1 \ Core2	閾値以下	閾値より上
閾値以下	Level Up	状態維持
閾値より上	状態維持	Level Down

表 2 シミュレーションのキャッシュに関するパラメータ

キャッシュ容量	
L1 i-cache	32KB(64B/entry, 1way, 512entry)
L1 d-cache	32KB(64B/entry, 1way, 512entry)
Shared L2 cache	512KB(64B/entry, 4way, 2048entry)
ヒット・レイテンシ	
L1 cache	1 cycle
L2 cache	10 cycle
主記憶	250 cycle
ウェイクアップ・オーバーヘッド	
レイテンシ	10cycle

4.1.2 Level 3 まで対応した可変レベルキャッシュ

我々はこれまでに可変レベルキャッシュの性能を向上するための手法をシングルプロセッサ環境において提案してきた．これらの手法はマルチプロセッサ環境においても可変レベルキャッシュの性能向上に寄与すると考えられる．そこで，我々の提案している性能向上手法のマルチプロセッサ環境での影響を明らかにするために，Conventional にそれらの性能向上手法を適用した可変レベルキャッシュについて評価を行う．この可変レベルキャッシュを以降の評価では「3-Level」と呼ぶ．この 3-Level は Conventional と同様，レベル切換頻度を抑制するコントローラを実装している．

4.2 高頻度レベル切換コントローラを組込んだ可変レベルキャッシュ

Conventional と 3-Level で用いられているコントローラは，Level 変更時に書戻しオーバーヘッドが発生することを想定して作られており，Level 切換頻度が抑制されている．しかしながら，第 3.3 項で記述したオーバーヘッド削減手法により Level 変更をより頻繁に行っても性能低下を低く抑えることができるため，より頻繁な Level 切換を行うことでより大きな電力削減が期待できる．そこで，一定間隔毎に共有 L2 キャッシュのミス率を算出し，閾値によってレベルを切換えるコントローラを 3-Level に組み込んだ可変レベルキャッシュの評価を行う．この可変レベルキャッシュを「Proposed」と呼ぶ．なお，上記三手法に組込むコントローラは全て同じ一定間隔毎にミス率を算出するものとする．

4.3 実験環境

本稿ではマルチスレッドアプリケーションに対応しているマルチプロセッサシミュレータ M5 [13] を改造し，M5 シミュレータのキャッシュアクセスに関するトレースデータを作成した．そのトレースデータを用いて動作するキャッシュシミュレータを作成し，各手法の評価を行った．

本実験において，可変レベルキャッシュは 512KB の共有 L2 キャッシュに実装し，また，CPU はデュアルコアと

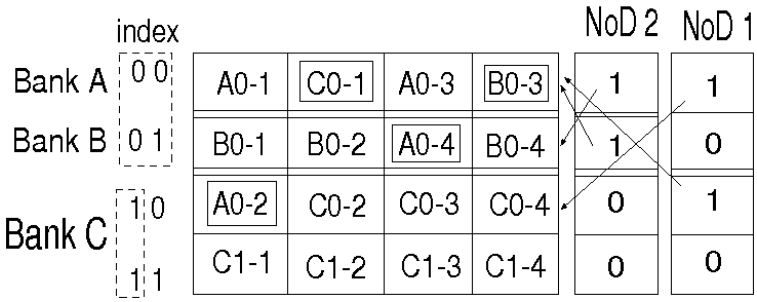

図 5 書戻しオーバヘッド低減手法

し，クロック周波数は 1GHz とした．

ベンチマークプログラムはマルチスレッド環境での評価として，SPRASH2 ベンチマークより CHOLESKY，FFT，LU，RADIX の計 4 種類を使用した．また，マルチタスク環境での評価として，SPEC2000[8] より，SPEC2000 から 164.gzip, 181.mcf, 183.equake, 188.ammp, 256.bzip2 の計 5 種類を使用し，デュアルプロセッサ環境で 2 種類のプログラムを組み合わせてシミュレーションを行った．また，キャッシュヒット率とエネルギー削減効率の関係性を調査するため，シングルコア環境においてヒット率の高いベンチマークとベンチマークをそれぞれ同一ベンチマークの組合わせをマルチタスクとして評価に用いた．紙面の都合上，評価結果の傾向が類似したものについては結果を省略している．

5. 評価結果

本稿では，可変レベルキャッシュについて，実行時間，消費エネルギーと電力遅延積について評価を行う．電力遅延積は高性能と低消費エネルギーの両立を指し示す指標の一つで，実行時間 × 消費エネルギーで計算される．実験によって得られたマルチスレッド，マルチタスク環境それぞれでの可変レベルキャッシュの実行時間を図 6，消費エネルギーを図 7，電力遅延積による評価結果を図 8 に示す．縦軸はそれぞれ通常のキャッシュの結果で正規化したものである．図中の Conventional，3-Level，Proposed はそれぞれ第 4.1 項で記述した手法である．

全体の傾向としては，Conventional，3-Level，Proposed の順で実行時間が増加しているが，その割合は小さい．反対に，消費エネルギーは同じ順で削減されており，実行時間の増加と比較するとその割合は大きく，電力遅延積も改善されている．

評価結果から，全てのベンチマークは 4 つの傾向に分かれた．一つ目の傾向としては，AmmpBzip2，AmmpEquake，AmmpMcf，Bzip2Mcf，CHOLESKY，FFT，LU の様な Proposed が大きくエネルギー削効果を向上させているものである．これらのベンチマークのほとんどにおいて 3-Level と Proposed の実行時間の増加がみられた．これはプログラムの一部が Level 3 で実行されたためである．Level 3 では Level 2 と比較して一部のデータアクセスにより多くのレイテンシを必要とする．また，Level 1 に戻った後の不良配置ラインへのアクセスに最大 3 回のアクセスが必要となるため，実行時間が Conventional よりも増加している．消費エネルギーでは，Proposed に組込んだ切換コントローラによって，プログラムの細かな要求の変化に素早く対応することが出来た結果，Proposed は Level 3 で動作する割合が大きくなり，エネルギー削減効果の向上に繋がった．Conventional，3-Level については，レベル切換頻度を抑制するコントローラの働きによって，Level 1 と Level 2 で動作する割合が大きいため，両者のエネルギー削減率に差はみられなかった．

二つ目の傾向としては，AmmpGzip，RADIX の様な Proposed のエネルギー削減効果が他の二手法を下回ったものである．これらのベンチマークの特徴としては，Proposoed の実行時間が 3-Level を上回っている点である．これは，Proposed のコントローラによって，Level 2 から Level 3 へ切替わった直後のレベル切換で，すぐに Level 2 に切替わる，といった余分なレベル切換を行ってしまうためである．Level 3 ではキャッシュ内に目的データが存在しない場合に，Level 2 よりもスリープモードの領域を起こす回数が増加する．Proposed は要求に沿わないレベルでの動作によって実行時間が増加してしまった．また，スリープ領域へのアクセス頻度が大きくなり，スリープ領域を起こすためのエネルギーや再アクセスにかかるエネルギーも増加する．これらの原因によって，Proposed が 3-Level に劣る結果となった．また，AmmpGzip においては Conventional の実行時間が最大となったが，これはレベル切換時の書戻しオーバヘッドによるものと考えられる．

三つ目の傾向としては，AmmpAmmp，EquakeEquake の様な 3-Level，Proposed の二手法に大きなエネルギー削減効果がみられたものである．これら Ammp，Equake のベンチマークはシングルコア環境での評価において，高いキャッシュヒット率を示したものであり，今回の評価では同一のベンチマークを組合わせて使用した．これらのベンチマークはマルチタスク環境においても高いヒット率を示

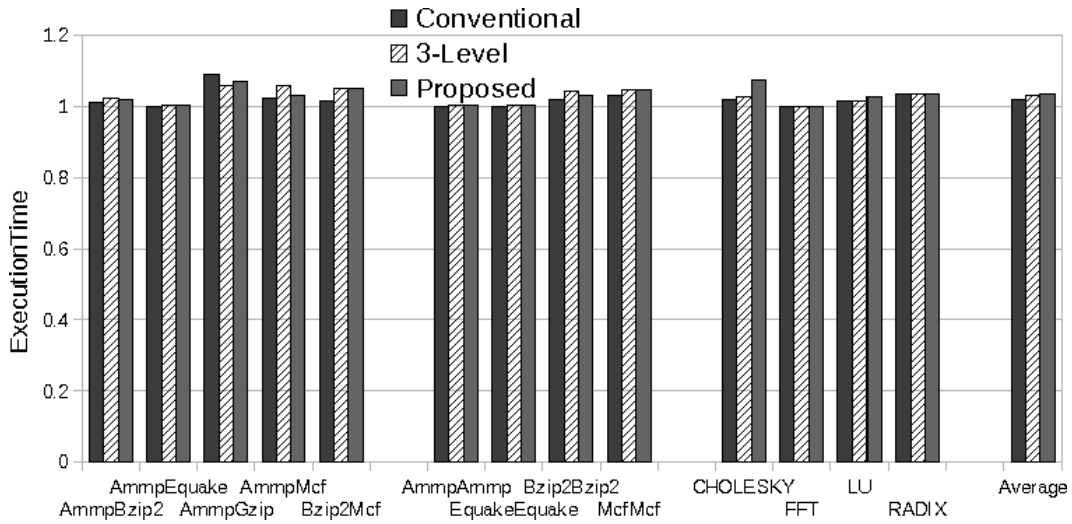

図 6　実行時間

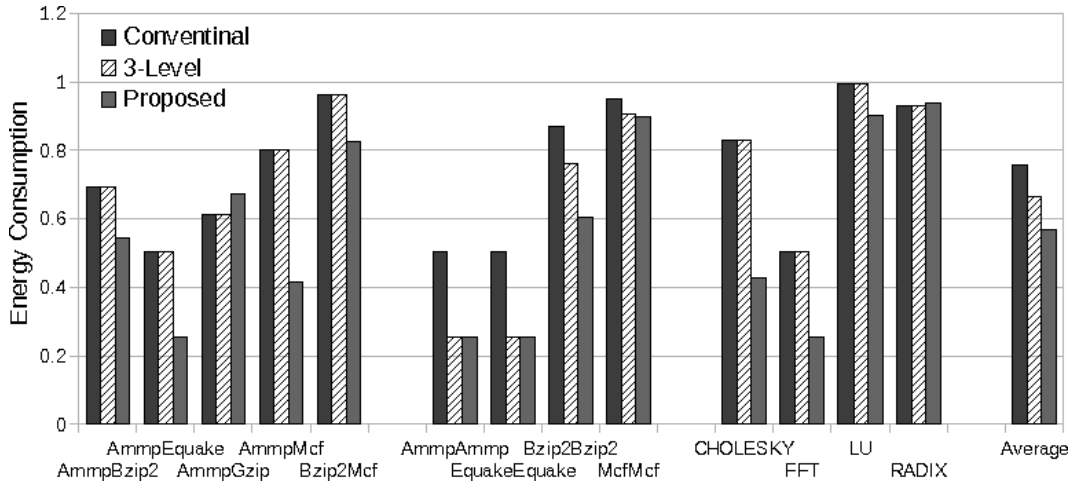

図 7　消費エネルギー

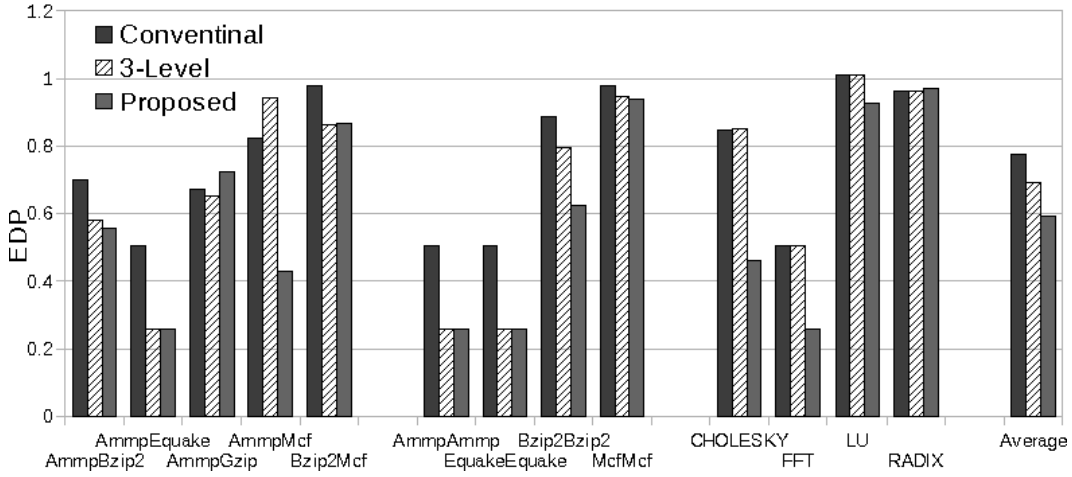

図 8　電力遅延積

した．今回使用した二種類のコントローラはそれぞれヒット率によるレベル判定を行うため，プログラムのほとんどをスリープ領域が最も大きくなる Level で動作し続ける結果となった．また，高ヒット率のため，スリープ領域へのアクセス回数が少なく，実行時間の増加もほとんどみられなかった．Conventional はスリープ領域を最大で全体容量の半分までにしかできないため，他の二手法よりもエネルギー削減効果が小さいものとなった．

四つ目の傾向としては，Bzip2Bzip2，McfMcf の様な Proposed，3-Level，Conventional の順にエネルギー削減効果が高かったものである．これらのベンチマークはシングルコア環境においてヒット率が低く，また，キャッシュアクセス頻度も少ないベンチマークの組み合わせである．Proposed はレベルが切替わる頻度が高いため，プログラムの要求に柔軟に対応できた結果，他手法よりも大きなエネルギー削減を行うことができた．また，Level 3 に対応している 3-Level もわずかではあるが，Level 3 で動作したため，Conventional よりも高いエネルギー削減を行うことができた．

最後に，性能評価の指標として電力遅延積を用いて評価を行った．電力遅延積において，Proposed は Conventional と比較して平均約 24%，3-Level と比較して平均約 15%性能向上することがわかった．また，通常のキャッシュと比較して平均約 41%性能向上することがわかった．

6. まとめ

本稿では，切換可能レベルを増やした可変レベルキャッシュをマルチプロセッサ環境へ適用し，マルチスレッド，マルチタスクの両環境での評価を行い，可変レベルキャッシュのマルチプロセッサ環境での有効性を示した．シミュレータによる評価の結果，通常のキャッシュよりも電力遅延積において平均約 41%の改善がみられた．

今後は実際にハードウェアを設計し，詳細な評価を行う．またはキャッシュパーテショニングなど，他の低電力キャッシュ手法との比較を行っていきたい．

参考文献

[1] 恩賀琢也, 佐々木 敬泰，大野 和彦，近藤 利夫，“キャッシュ階層動的切り替えによる低消費電力化”，情処学研報, 2007-ARC-174,pp.115-120. August 2007
[2] 松原 伸幸，佐々木 敬泰，大野 和彦，近藤 利夫，“高性能かつ低消費電力を実現する可変レベルキャッシュのモード切換アルゴリズムの改良と評価 “ ，信学会技報，CPSY2009-44, pp.7-12，December 2009
[3] S. Kaxiras, Z. Hu, and M. Martonosi, “Cache Decay: Exploiting Generational Behavior to Reduce CacheLeakage Power,” Proc. of the 28th Int. Symp. on Computer Architecture, pp.240-251, June 2001.
[4] K. Flautner, N.S. Kim, S. Martin, D. Blaauw, and T. Mudge, “Drowsy Cache: Simple Techniques for Reducing Leakage Power,” Proc. of the 29th Int. Symp on Computer Architecture, pp. 148-157, May 2002.
[5] N.S. Kim, K. Flautner, D. Blaauw, and T. Mudge, “Drowsy Instruction Caches; Leakage Power Reduction using Dynamic Voltage Scaling and Cache Sub-bank Prediction,” Proc. of the Int. Symp. on Microarchitecture, pp.219-230, November 2002.
[6] 田中秀和，井上弘士，モシニャガ・ワシリー，“低消費電力を目的とした適応型ウェイ予測キャッシュとその評価，” 信学会技報, VLD2004-139, ICD2004-235, pp.13-18, March 2005.
[7] S.H. Yang, M.D. Powell, B. Falsafi, K. Roy, and T.N. Vijaykumar, “An Integrated Circuit / Architecture Approach to Reducing Leakage in Deep-Submicron High-Performance I-Caches,” Proc. of the 7th Int. Symp. on High-Performance Computer Architecture, pp.147-157, February 2001.
[8] “SPEC -Standard Performance Evaluation Corporation-,” URL: http://www.spec.org/.
[9] 小宮礼子，井上弘士，モシニャガ・ワシリー，村上和彰，“キャッシュ・リーク電力削減アルゴリズムに関する定量的評価，” 第 17 回回路とシステム軽井沢ワークショップ論文集, pp.235-240, April 2004.
[10] 図子純平，冨山宏之，高田広章，井上弘士，“Drowsy キャッシュにおけるモード切替アルゴリズムの評価，” 情処学研報, 2006-ARC-170, pp.37-41, December 2006.
[11] CACTI 5.1 Shyamkumar Thoziyoor, Naveen Muralimanohar, Jung Ho Ahn and Norman P. Jouppi.
[12] 城田幸利，佐々木 敬泰，大野 和彦，近藤 利夫 “可変レベルキャッシュ用モード切換手法のマルチコア環境への適用と評価，” SWoPP2010，August 2010.
[13] Nathan L. Binkert，Ronald G. Dreslinski，Lisa R. Hsu，Kevin T. Lim，Ali G. Saidi，Steven K. Reinhardt “The M5 Simulator: Modeling Networked Systems，”
[14] 渡部 功，佐々木 敬泰，松原伸幸，大野 和彦，近藤 利夫 “モード切換オーバヘッドを低減した可変レベルキャッシュの提案と評価，” ACS，採録決定済，2012.
[15] Ko WATANABE, Takahiro SASAKI, Kazuhiko OHNO and Toshio KONDO “IMPROVEMENT OF WRITEBACK MECHANISM OF VARIABLE LEVEL CACHE，” ITC-CSCC2012，July 2012.