科学研究費助成事業　　研究成果報告書

平成　２６　年　　５　月　２１　日現在

機関番号：　１１３０１
研究種目：　挑戦的萌芽研究
研究期間：　2012～2013
課題番号：　２４６５００１８

研究課題名（和文）アプリケーション適応型動的超多階層メモリアーキテクチャの開発

研究課題名（英文）Application-Aware Highly Hierarchical Memory Architecture

研究代表者
小林　広明（Kobayashi, Hiroaki）

東北大学・サイバーサイエンスセンター・教授

研究者番号：４０２０５４８０

交付決定額（研究期間全体）：（直接経費）　　2,900,000円、（間接経費）　　870,000円

研究成果の概要（和文）：　本研究の目的は、アプリケーションが求めるメモリ機能・性能からアーキテクチャ設計を見直し、多階層・アプリケーション適応型オンチップメモリアーキテクチャ、及びその利用技術を確立することを目的としている。
　本研究では、マイクロプロセッサの高性能化・低消費電力化に向けて、キャッシュメモリを考慮した効率的な資源管理に取り組んだ。このような資源管理は、キャッシュメモリ上で発生するスレッド間資源競合の回避や、キャッシュメモリ資源の効率的な利用を可能とし、マイクロプロセッサの性能向上・消費電力の削減を可能とする。

研究成果の概要（英文）：The objective of this study is to establish a novel on-chip memory architecture that can provide necessary memory resources to running applications under the consideration of their behaviors and requirements regarding a memory subsystem on a multi-core processor.
In this study, we have developed a cache-resource management mechanism to realize energy-efficient high performance execution of multi-threaded applications on a multi-core processor. In cooperation with developed hardware functions of cache resizing and partitioning to reduce cache conflicts and maximize the efficiency of cache utilization, this mechanism can extract the potential of multi-core processors with a low-power consumption.

研究分野：　総合領域

科研費の分科・細目：　情報学・計算機システム・ネットワーク

キーワード：　キャッシュメモリ　コンピュータアーキテクチャ　キャッシュパーティショニング　スレッドスケジューリング

１．研究開始当初の背景

シンプルなプロセッサコアを多数集積させたマルチコアプロセッサアーキテクチャは、単一コアプロセッサの限界を打破し、10億超トランジスタ時代における基本マイクロアーキテクチャとして高性能汎用プロセッサから組み込み用プロセッサにいたるまで、広範囲な適用が期待されている。しかしながら、これら大量のコアの集積によりCPU演算パフォーマンスが向上できる一方、チップあたりの外部とのデータ転送能力(ピンバンド幅）の制限はますます深刻になり、それをカバーするオンチップメモリのデータ供給性能がプロセッサの実効性能を大きく左右することから、マルチコアに対応した革新的なオンチップメモリアーキテクチャが求められている。

２．研究の目的

本研究では、メニーコアプロセッサ時代の革新的メモリシステム実現を目指して、アプリケーションが求めるメモリ機能・性能からアーキテクチャ設計を見直し、多階層・アプリケーション適応型オンチップメモリアーキテクチャ、およびその利用技術を確立することを目的としている。

３．研究の方法

本研究では、マイクロプロセッサの高性能化・低消費電力化に向けて、個々のアプリケーションの実行ワーキングセットを考慮してキャッシュメモリの効率的な資源管理に取り組んだ。このような資源管理は、キャッシュメモリ上で発生するスレッド間資源競合の回避や、キャッシュメモリ資源の効率的な利用を可能とし、マイクロプロセッサの性能向上・消費電力の削減を可能とする。

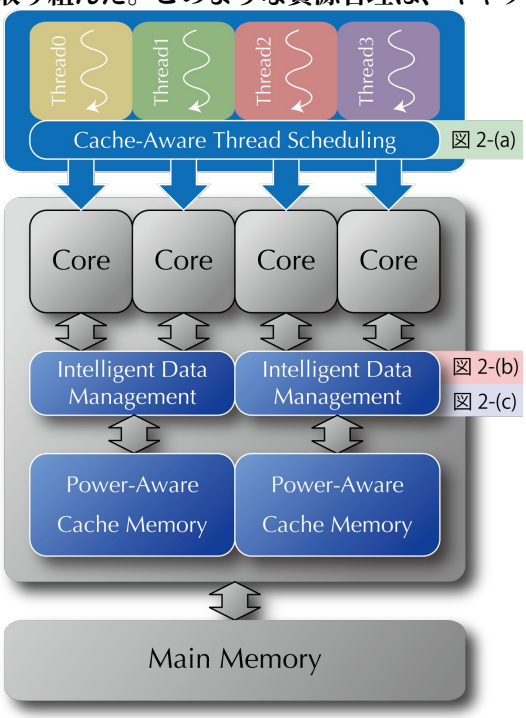

図　１　ターゲットアーキテクチャ

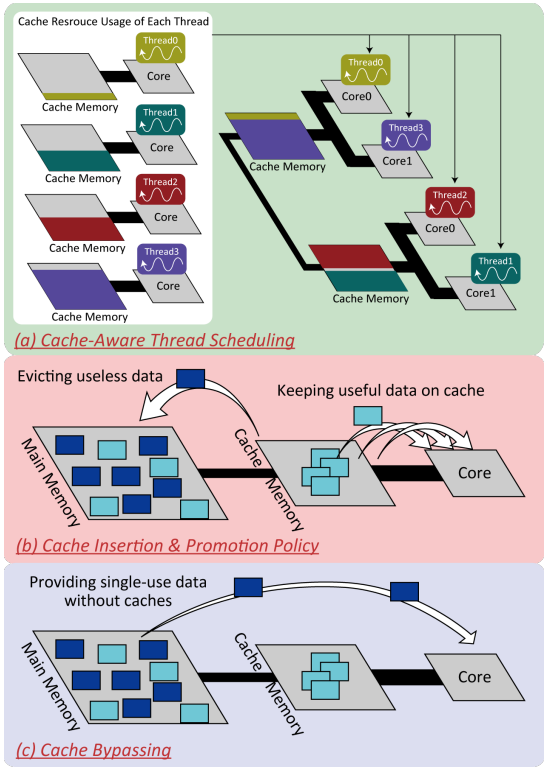

図　２　提案する資源制御機構

図１に本研究で設計したターゲットアーキテクチャを示す。本研究は、マイクロプロセッサに複数のコア・共有キャッシュが搭載されている構成を想定しており、近年のマルチコアプロセッサとしては一般的な構成である。このようなマイクロプロセッサ上で、コアで実行されるスレッドのキャッシュ利用状況を考慮しつつ、スレッドとコアの関係を決定するスレッドスケジューリング(図2(a))に取り組み、複数スレッドでキャッシュを共有する場合の利用効率の向上を図った。また、再利用されないデータの早期の追い出しを行うデータ管理ポリシ（図 2(b))や、再利用されないデータのキャッシュへの保存を抑制するキャッシュバイパス機構(図2(c))についても取り組み、スレッド実行時のキャッシュ資源利用の効率化をはかった。

４．研究成果

(1) キャッシュを考慮したスレッドスケジューリング

マルチコアプロセッサにおける共有キャッシュの利用効率向上を目的に、キャッシュパーティショニング統合型スレッドスケジューリング手法を提案した[雑誌論文１]。本手法はマルチコアプロセッサにおける共有キャッシュ資源競合の解消により、性能向上

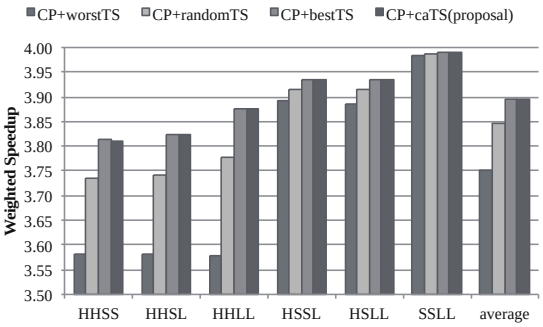

図 3　スケジューリング時のプロセッサ全体のスループット

を実現することが可能である。

共有キャッシュ資源競合には主に 2 つのタイプがある。1 つ目の競合は Inter-Thread KickOut (ITKO)と呼ばれるもので、キャッシュを共有している一方のスレッドの有用なデータが、他方のスレッドのデータによって頻繁に追い出される現象である。この現象により、追い出されたデータのスレッドでミスが頻発するため、性能低下が発生する。2 つ目の競合はキャッシュ資源量の不足によるものである。スレッドが高性能で実行されるために必要となるキャッシュ資源量が多い場合、複数のスレッドが同じキャッシュを共有すると、必要とする資源量に対してキャッシュメモリの提供可能な資源量が不足する。このとき、スレッドが必要なデータをキャッシュに保存することができず、キャッシュミスが発生する。この結果、キャッシュを共有するスレッドで性能低下が発生する。

これら 2 つの競合に対し、本手法では、キャッシュパーティショニング機構を用い、スレッドに対してキャッシュ資源を排他的に割り当てることにより ITKO を防ぐ。また、スレッドが要求するキャッシュ資源量についての情報をキャッシュパーティショニング機構により得た上で、スレッドが利用可能なキャッシュ資源量を最大限確保できるようコアにスレッドをスケジューリングする。図 3 はマイクロプロセッサ全体のスループットの評価結果を表しており、縦軸はスループット、横軸はベンチマークを示す。図 3 より、他のスケジューリング手法と比較して提案手法(グラフ中の ”CP+caTS(proposal)”)では全体のスループットが最大 8%向上した。

また、図 4 は個別スレッドのスループットの評価結果を示しており、縦軸はスループット、横軸は各スレッドを示している。図 4 より、他のスケジューリング手法に比べて提案手法では個別スレッドのスループットが最大 12%改善した。以上より、提案手法は、マルチコアプロセッサの性能向上を実現できることが示された。

(2) キャッシュデータ管理ポリシ

スレッド実行時のキャッシュ資源利用の

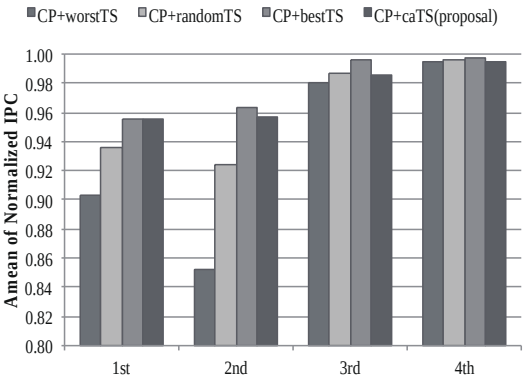

図 4　個別スレッドのスループット評価

効率化を目的に、再利用されないデータを考慮したデータ管理ポリシを提案した[雑誌論文 2、 3]。本ポリシは、再利用されないデータをキャッシュメモリから早期に追い出すことを可能とし、スレッドが必要とするキャッシュ資源量を削減することが可能である。

一般的なキャッシュメモリは、再利用されるデータが保存されることを前提に、LRU 置換ポリシを用いてデータを管理している。LRU 置換ポリシでは参照局所性の原則に基づき、最近アクセスされた順番に応じてデータに高い優先度を割り当てつつ、優先度の高いデータをキャッシュ中に維持しようとする。

しかし、近年ではアプリケーションの応用範囲の広がりや、扱うデータセットの増大により、キャッシュに保存しても性能向上が得られないデータが増加しつつある。このようなデータは、キャッシュに保存されたとしても一度も再利用されない、もしくはキャッシュに保存されている間は再利用されない。このような状況下で LRU 置換ポリシを用いると、再利用されないデータであっても高い優先度が割り当てられる。このため、再利用されないデータがキャッシュを長時間占有し、性能向上に貢献しない不要なエネルギを消費してしまう。また、再利用されないデータを高い優先度で保存することによって、本来再利用されるはずの低い優先度のデータを追い出し、性能低下が発生する。

そこで、提案ポリシはキャッシュメモリに保存されるブロックに対して比較的低い優先度を設定可能とした。これにより、再利用

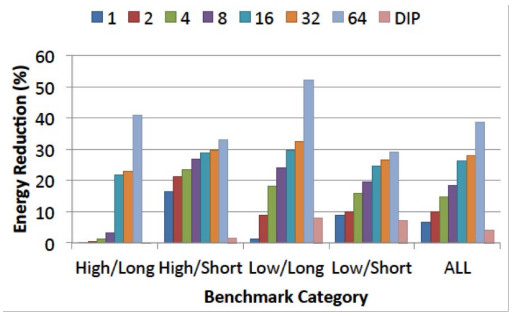

図 5　提案ポリシ適用時のエネルギ削減率

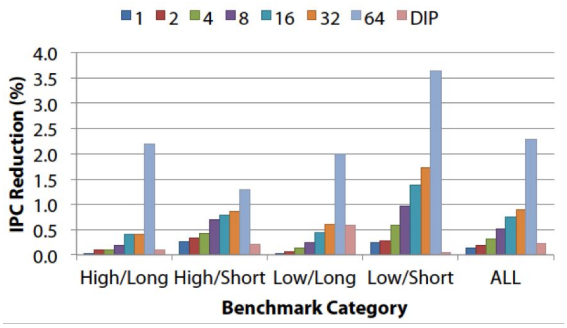

図 6　提案ポリシ適用時の性能低下率

されるブロックを維持しつつ、再利用されないブロックを早期に追い出すことを可能とした。その結果、性能低下を抑制しつつ、エネルギを削減することが可能である。

本手法を省電力化キャッシュパーティショニング機構と併用した場合の消費エネルギの評価結果を図 5 に示す。図 5 の縦軸はエネルギ削減率、横軸はベンチマークカテゴリを示す。図 5 より、代表的なパラメータ設定の提案手法(グラフ中 ”32 ”)では、エネルギを平均で 27%削減することができた、

また、図 6 に性能低下率を示す。図 6 の縦軸は IPC 低下率、横軸はベンチマークカテゴリを示す。図 6 から提案手法による IPC 低下率は 1%程度であった。よって、提案手法は性能低下を抑制しつつエネルギを削減することが可能であり、キャッシュメモリの利用効率を向上できることが示された。

さらに、上記の手法を拡張し、キャッシュメモリ上で一度再利用されたデータもその後は再利用されなくなる場合を考慮し、アクセスを受けても優先度を増加させないデータ管理ポリシも考案した[学会発表 2]。

(3)　キャッシュバイパス機構

スレッド実行時のキャッシュ資源利用の効率化を目的に、再利用されないデータの保存を抑制するキャッシュバイパス機構を提案した[学会発表 1、 3]。本手法は前節で述べた再利用されないブロックの削減を、データ管理ポリシの限界を超えて推し進める機構である。

前節で述べたデータ管理ポリシでは、キャッシュに保存されるデータを比較的低い優先度に設定することにより、再利用されないデータの早期追い出しをはかるとともに、低い優先度にある再利用されるデータの追い出しを回避している。しかし、データが再利用される際の優先度が非常に低い場合には、保存されるデータの優先度を低くすると、アクセスされないままデータが追い出されてしまう。このような場合、データ管理ポリシは、データを保存する際の優先度を低くすることができない。このため、データ管理ポリシによってキャッシュにデータを保存する際に優先度を変更するだけでは、再利用されないブロックを十分に削減できない場合がある。

そこで、本機構では、再利用されないデータの保存自体を抑制するキャッシュバイパスを行う。本機構は、再利用されないデータのキャッシュ中の占有割合をモニタリングし、再利用されないデータのキャッシュの占有割合が多いほど、保存されるブロックの大部分は再利用されないデータであるとみなす。そして、再利用されないデータの保存が非常に多い場合には、キャッシュに保存されるはずのデータをバイパスする。また、本機構によるバイパスは保存されるデータに対して一定の確率で行われるため、個別のデータに対して再利用されるかの判断は行わない。このため、キャッシュバイパスの実現に必要なハードウェアコストを抑制することができる。

評価結果として、図 7 にキャッシュメモリの省電力化パーティショニング機構と併用した場合のキャッシュの消費エネルギの評価結果示す。図 7 の縦軸は消費エネルギ、横軸はベンチマークカテゴリを示す。図 7 より、ベースライン(baseline)と比較して提案機構 (proposal(3bit)) ではエネルギを平均 5.9%削減できた。ベンチマーク毎では最大で 46%のエネルギ削減が得られるものもあった。

また、図 8 は提案機構を適用した場合の性能評価結果を表しており、縦軸は正規化 IPC、横軸はベンチマークカテゴリを示している。図 8 より、ベースラインと比較した場合の性能低下は平均 1%以内に抑制できる。

以上の結果から、提案機構は性能低下を抑制しつつエネルギ削減を実現し、キャッシュ

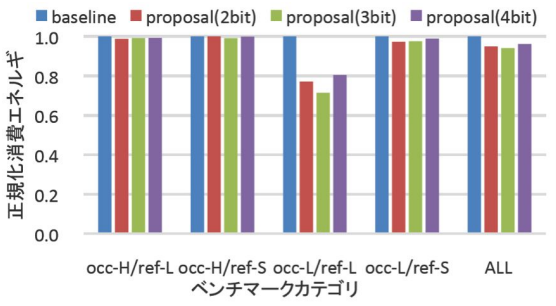

図 7　提案バイパス機構を適用した場合のエネルギ削減率

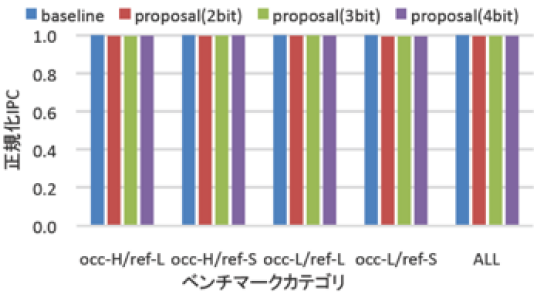

図 8　提案バイパス機構を適用した場合の IPC 評価

の利用効率を向上できることが示された。

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計　3 件）

[1] Masayuki Sato, Ryusuke Egawa, Hiroyuki Takizawa, Hiroaki Kobayashi, "A Capacity-Aware Thread Scheduling Method Combined with Cache Partitioning to Reduce Inter-Thread Cache Conflicts," IEICE Transaction on Information and Systems, Vol. E96-D, pp.2047--2054, September 2013. （査読あり）

[2] Masayuki Sato, Yusuke Tobo, Ryusuke Egawa, Hiroyuki Takizawa, Hiroaki Kobayashi, "A Capacity-Efficient Insertion Policy for Dynamic Cache Resizing Mechanisms," Proceedings of ACM International Conference on Computing Frontiers, pp.265--267, May 2012. [DOI: 10.1145/2212908.2212949] （査読あり）

[3] Masayuki Sato, Yusuke Tobo, Ryusuke Egawa, Hiroyuki Takizawa, Hiroaki Kobayashi, "A Flexible Insertion Policy for Dynamic Cache Resizing Mechanisms," In Proceedings of IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips XVI), pp.1--3, April 2013. [DOI: 10.1109/CoolChips.2013.6547923] （査読あり）

〔学会発表〕（計　3 件）

[1] 高井 拓実, 佐藤 雅之, 江川 隆輔, 滝沢 寛之, 小林 広明, "ブロックバイパス機構によるキャッシュのエネルギ効率化に関する研究," 並列/分散/協調処理に関するサマー・ワークショップ (SWoPP2013), 北九州市, 7 月 31-8 月 2 日, 2013.

[2] 東方 雄亮, 佐藤 雅之, 江川 隆輔, 滝沢 寛之, 小林 広明, "ウェイ適応型キャッシュの高エネルギ効率化のためのデッドブロック早期追い出しポリシ,"先進的計算基盤シンポジウム SACSIS2012, 神戸, 5 月 16-18 日, 2012.

[3] Takumi Takai, Yusuke Tobo, Masayuki Sato, Ryusuke Egawa, Hiroyuki Takizawa, Hiroaki Kobayashi, "A Bypass Mechanism for Way-Adaptable Caches," COOLChips XV, Yokohama, April 18-20, 2012.

〔その他〕
ホームページ等
http://www.sc.isc.tohoku.ac.jp/

６．研究組織
(1)研究代表者
小林　広明（KOBAYASHI　HIROAKI）
東北大学・サイバーサイエンスセンター・教授
研究者番号：40205480

(2)連携研究者
滝沢　寛之（TAKIZAWA　HIROYUKI）
東北大学・大学院情報科学研究科・准教授
研究者番号：70323996

江川　隆輔（EGAWA　RYUSUKE）
東北大学・サイバーサイエンスセンター・准教授
研究者番号：80374990