C-002

メニーコアにおける柔軟なデータ供給支援の検討と予備評価
Preliminary Evaluation on Adaptable Data Access Support for Manycore

高橋 朝英† 小林 良太郎† 吉瀬 謙二‡
Tomohide Takahashi Ryotaro Kobayashi Kenji Kise

1. はじめに

近年，発熱や配線遅延等の問題から複数のコアを搭載したマルチコアが主流になっている．また，集積技術の発展により 1 つのチップに搭載されるコアの数は増加してきており，将来，数十から数百のコアを搭載したメニーコアが主流になると考えられている [1]．その一方で，プロセッサは性能向上，低消費電力，信頼性など様々な問題に直面している．これらの問題を打破するために，複数の機能を提供する Multifunction メニーコアの実現が重要になってくる [2]．しかしながら，ハードウェアだけに頼る手法では，ハードウェアコストを増加させ，柔軟性を低下させる．我々はシステムソフトウェアを使用することによって，各コアに柔軟性を持たせるメニーコアの研究を行っており，本文では，メニーコアの複数の機能の 1 つである柔軟なデータ供給支援手法の提案をする．この手法は，キャッシュプログラムを実行し [3]，他コアに対してキャッシュとして提供するキャッシュコアに基づいている [4]．キャッシュコアは，キャッシュパフォーマンスやデータアクセスパターン，タスクの優先度などに応じて，キャッシュ方式を動的に変更する新しい機能を持っている．現在，この手法の実装を進めている．ここでは，ダイレクトマップ (DM) 方式，ダイレクトマップと Victim キャッシュ使用した方式 (DM + Victim)，そして 2 ウェイセットアソシアティブ (2-way) 方式などのソフトウェアで実装した複数のソフトウェアキャッシュ方式の予備評価結果を示す．

2. 提案手法

2.1 概要

図 1 を用いて，提案手法におけるキャッシュ方式の動的切り替えについて説明する．キャッシュコアは，自コアのリソースを他コアに提供し，計算の支援を行うコアである．例えば，キャッシュコアは，アプリケーション実行時の L1 キャッシュや L2 キャッシュのヒット率が低下した場合に複数のキャッシュ方式の中から最適なキャッシュ方式を選択し，実行中にキャッシュ方式の切替えを行うことで，性能の向上を図る．キャッシュ方式の切替えを行う手法として，計算を止めるとオーバヘッドが大きくなるため，L1 または L2 キャッシュのみにダイレクトアクセスするようにする，等の方法が考えられる．最適なキャッシュ方式の選択を行うため，今回は複数のキャッシュ方式をソフトウェアキャッシュとして実装し，各方式の評価を行った．

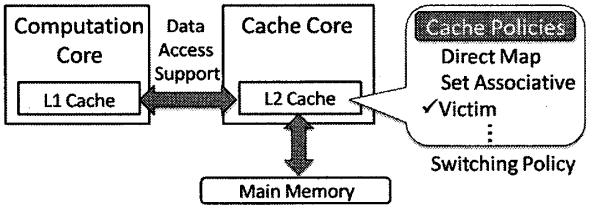

図 1: キャッシュ方式の動的切り替え

2.2 キャッシュ方式

DM 方式は [4] で使用されているものと同じもの使用する．図 2(a) と (b) は，それぞれ 2-way 方式と DM+Victim 方式を示す．どの方式とも，アプリケーションを実行するコアのローカルメモリの一部をキャッシュとして使用し，リプレイス方式は，ライトバック方式を使用する．

DM+Victim 方式は，L1 キャッシュにダイレクトマップ方式を使用し，L2 キャッシュに Victim キャッシュを使用する方式である．Victim キャッシュは，キャッシュから追い出されたデータがメインメモリ上にある他のデータよりも使用される可能性が高い事に基づいたキャッシュ方式であり，ダイレクトマップから追い出されたデータは，Victim キャッシュに書き込まれる．DM+Victim 方式は 2 つの方式を組み合わせることで，ダイレクトマップで発生しやすいキャッシュスラッシングが起きた場合でも極端なヒット率の低下を防ぐことを狙いとする．また，ダイレクトマップは一般的に 2 のべき乗のサイズで確保するが，Victim キャッシュのエントリ数を変更することで，限られたローカルメモリの容量を有効に利用することができる．

データを参照する場合は，L1 キャッシュであるダイレクトマップを先に参照し，見つからなかった場合，次に L2 キャッシュである Victim キャッシュを参照する．Victim キャッシュでもデータが見つからなかった場合，メインメモリからローカルメモリに DMA 転送によってデータを取得する．この際，Victim キャッシュの探索を開始する前に DMA 転送を発行することで，キャッシュミスを起こした場合の DMA 転送時間の隠蔽を図る．

2-way 方式は，1 次キャッシュとして 2 ウェイセットアソシアティブ方式を使用することで，ダイレクトマップで発生しうるキャッシュスラッシングを防ぐとともに，連想度をダイレクトマップよりも上げることで，キャッシュ

†豊橋技術科学大学大学院工学研究科
Graduate School of Engineering, Toyohashi University of Technology
‡東京工業大学大学院情報理工学研究科
Graduate School of Information Science and Engineering, Tokyo Institute of Technology

ヒット率の上昇を狙う．連想度が高いほどヒット率は上昇するが，今回は探索時間などを考慮し，2 ウェイセットアソシアティブとした．データアクセスの際は，way0 をまず参照し，次に way1 の参照を行う．キャッシュミスを起こした場合は，DMA 転送によってメインメモリからローカルメモリへデータの転送を行う．キャッシュを参照する際，ソフトウェアで逐次的に検索するため，way0 は way1 にアクセスするよりも速くアクセスすることができる．そのため，way0 は最近アクセスされたデータを保持する．way0 に最近アクセスされたデータを保持するための way0 と way1 の置き換えは，DMA 転送中に行うことで隠蔽を図る．また，DM+Victim 方式と同様に探索を開始する前に当該データのデータ転送を発行することで，キャッシュミスが発生した際の DMA 転送のレイテンシを抑える．

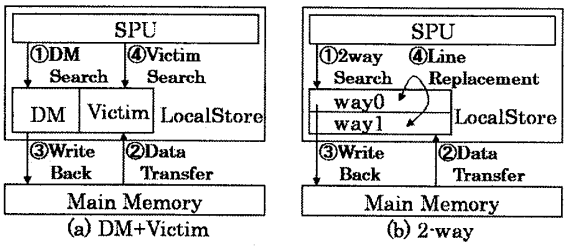

図 2: 実装したソフトウェアキャッシュ

3. 評価

評価はクイックソート (Qsort)，マージソート (Msort)，N クイーン (Nqueen)，ダイクストラ法 (Dijkstra)，姫野ベンチマーク (Himeno)，高速フーリエ変換 (FFT)，行列積 (MM) の 7 種類のベンチマークプログラムを用いて行った．各ベンチマークプログラムとソフトウェアで実装された各キャッシュ方式を Cell/B.E. 上で同時に実行し，その実行時間を計測した．ダイレクトマップおよび 2-way 方式のキャッシュサイズは 2KB，Victim キャッシュのキャッシュは 12 エントリ 3KB で測定を行った．

図 3 はソフトウェアキャッシュを使用しない場合に対して，DM 方式，DM+Victim 方式，2-way 方式の速度向上率を示したものである．FFT 以外で各方式はキャッシュを使用しない場合と比較して性能が改善している．

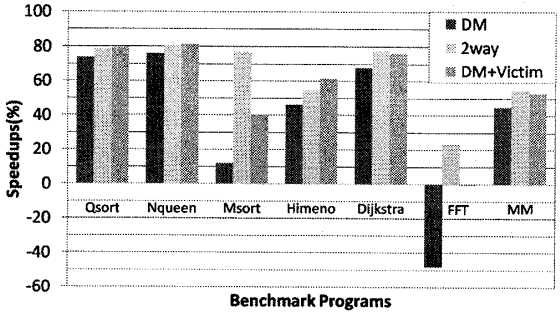

図 3: ベンチマーク実行結果

4. まとめ

Cell/B.E. 上で DM+Victim 方式と 2-way 方式を実装し，ベンチマークプログラムを実行した結果，2-way 方式では全てのベンチマークプログラムにおいて性能が向上し，その他の方式においても，FFT 以外で性能が向上した．FFT では，キャッシュスラッシングが発生しており，キャッシュヒット率が極端に低下したため，DM 方式で性能が低下したと考えられる．また，FFT において 2-way 方式及び DM+Victim 方式はキャッシュヒット率が DM 方式と比較し改善されたが，DM+Victim 方式では性能の向上が得られていない．これは，Victim キャッシュの探索では，キャッシュエントリの先頭から逐次的に検索しており，キャッシュヒット率向上の効果がキャッシュ探索時間によって打ち消されたためと考えられる．

さらに，これらの実行結果は，実行するプログラムによってキャッシュ方式を変更することで，性能が向上することを示している．プログラムに対して最適なキャッシュ方式を動的に選択することができれば，各アプリケーションプログラムに対して最大限の性能向上を図ることができる．

今後の課題として，各ソフトウェアキャッシュ方式の効果のさらなる調査と動的キャッシュ制御機構の検討を行う．加えて，柔軟なデータ供給支援の実現において重要な技術になるであろう複数のキャッシュコアの融合の検討を行う．

謝辞

本研究の一部は，文部科学省科学研究費補助金若手研究 (B) 課題番号 21700057，柏森情報科学振興財団研究助成，及び，中山隼雄科学技術文化財団研究助成の支援により行った．

参考文献

[1] S. K. Borkar, et al., “Platform 2015: Intel Processor and Platform Evolution for the Next Decade,” Technology@Intel Magazine, 2005.

[2] 小林 良太郎ほか, “多機能メニーコアを実現するアーキテクチャ技術 Feature-Packing の構想,” 情報処理学会研究報告 2007-ARC-175, pp.11-15, 2007.

[3] 佐藤 芳紀ほか, “Cell Broadband Engine への SPE ソフトウェアデータキャッシュの実装,” 情報処理学会研究報告 2008-HPC-10, pp.13-18, 2008.

[4] 森谷 章ほか, “メニーコアプロセッサに向けたデータ供給を支援する多機能キャッシュコア,” 先進的計算基盤システムシンポジウム SACSIS2008, pp.421-430, 2008.