

マルチコア・プロセッサにおける SimCell を用いた キャッシュコアの可能性の検討

森谷 章^{†1} 藤枝 直輝^{†1}
佐藤 真平^{†1} 吉瀬 謙二^{†1}

1. はじめに

近年、性能向上を目指してプロセッサのマルチコア化が進んでいる。このマルチコア・プロセッサにおいてアプリケーションを実行する時、データ供給の支援を行うコアを導入する。このコアでは資源をデータキャッシュとして利用するプログラムを実行させる。このコアをキャッシュコアと呼ぶ。計算を行うコア（計算コア）はキャッシュコアに対してデータアクセスを行い通信レイテンシの大きいメインメモリへのアクセスを減少させ、全体として処理速度を向上させる。

本研究ではメニーコア・プロセッサにおけるキャッシュコアの可能性を検討する。ハードウェアのキャッシュをもたない Cell/B.E. の SPE 上で計算コアとソフトウェアキャッシュのプログラムを実行するキャッシュコアを実装する。機能レベルの Cell/B.E. プロセッサシミュレータ SimCell¹⁾ を用いてコア間の通信レイテンシを変化させて評価し、キャッシュコアの可能性を明らかにする。

2. マルチコアにおけるキャッシュコア

Cell/B.E. の SPE は、ローカルストア (LS) と呼ばれる 256KB の専用メモリを持つ。データを処理するためには、ここに予め DMA 転送によりロードしておく必要がある。

メインメモリへのアクセスは大きなレイテンシを要し、SPE にとっては大きなオーバーヘッドとなる。また、複数のコアからメインメモリへアクセスが集中した場合にはさらに大きなレイテンシが生じ、各コアの実行に多大な悪影響を及ぼす恐れがある。

本研究ではプログラムを実行していないアイドル状態のコアをキャッシュコアとして利用することを提案する。通常のマルチコア・プロセッサにおいてコア間の通信レイテンシはメインメモリの通信レイテンシより小さい。したがって、計算コアがキャッシュコアへアクセスすることでデータアクセス時のレイテンシの軽減を目指す。加えて、キャッシュのヒット率を向上させることでメインメモリへのアクセスを減少させる。

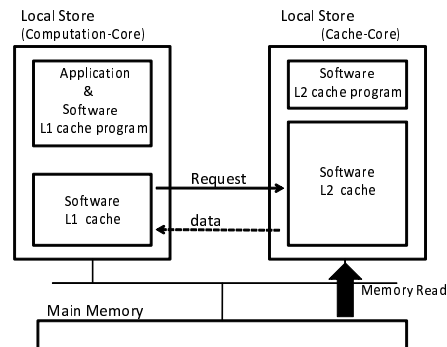


図 1 計算コアとキャッシュコア間のデータ読み出し時の流れ。

キャッシュコアはソフトウェアによって実現されるため、実行するアプリケーションに適した構成へと動的に変更できる。また、プリフェッチなどによる高速化が期待できる。

本稿では、まず単純なデータキャッシュの可能性を明らかにする。

図 1 に、キャッシュコアを想定した場合のデータ読み出し時の処理を示す。まず、計算コアが Read の要求をキャッシュコアに DMA 転送する。キャッシュコアは要求を受け取り、L2 キャッシュにヒットするとデータを L1 キャッシュに転送する。これら L1, L2 はソフトウェア実装のキャッシュである。ミスした場合はメインメモリからデータを読み出す。

3. SimCell を用いたキャッシュコアの検討

3.1 SimCell の仕様と検証プログラム

実験には機能レベルのシミュレータ SimCell Version 0.7.5¹⁾ を用いる。SimCell では DMA 転送によるメインメモリのリードとライト、コア間の通信レイテンシ、IPC などがそれぞれパラメータとして指定できる。

SimCell の想定する構成と実機の構成は異なるため、アプリケーションによって実行サイクル数に差が生じる。IPC を調節することである程度まで、差を縮めることができる。

また、設定できるレイテンシとは、DMA 転送命令を発行してからデータの転送が完了するまでのサイク

^{†1} 東京工業大学

Tokyo Institute of Technology

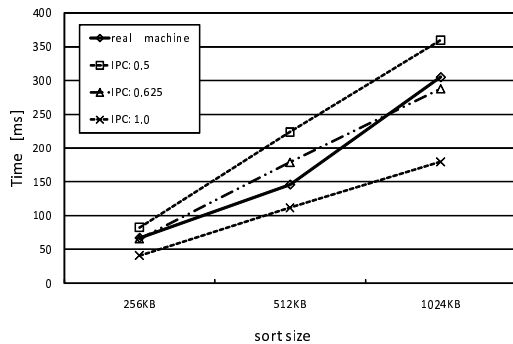


図2 実機 (Cell/B.E.) 及び SimCell によるクイックソートの実行時間。

ル数である。

キャッシュの効果を検証するプログラムは、クイックソートに計算コア上の L1 キャッシュのみを利用する版と、キャッシュコアを L2 キャッシュとして利用する版の 2 つを用いる。L1 キャッシュのみを使用する版では、データアクセスはコア内の L1 キャッシュを参照し、ミスが発生した場合はメインメモリを参照する。L2 キャッシュも用いる版ではデータアクセスは L1 キャッシュを参照し、ミスが発生した場合は別のコアの 32KB の L2 キャッシュを参照する。L2 キャッシュでミスが発生した場合はメインメモリを参照する。L1 キャッシュの容量は共通して 2KB である。

キャッシュコアのプログラムは処理のオーバーヘッドをなるべく小さくするため、ダイレクトマップ方式を採用し、リプレース方式はライトバックとする。なお、クイックソートを処理する計算コア数は 1 とする。

3.2 実験

L1 キャッシュのみを用いる版を用いて適切な IPC を調べる。図 2 に、SimCell で IPC のパラメータを変更して実行した場合と、実機で実行した場合の結果を示す。横軸はソートするデータのサイズを表し、縦軸は実行時間を表す。SimCell におけるメモリアクセスのレイテンシは文献 2) の値を参考に、リードを 700、ライトを 420 と設定する。

SimCell が算出する実行時間とは、処理に要したサイクル数を IPC を考慮して計算したものである。IPC が 0.625 とした時に実機に近い実行時間が得られた。従って、以降の実験において IPC は 0.625 とする。

次にキャッシュコアを利用する版を用いて、キャッシュコアの効果を検証する。図 3 に、コア間のレイテンシを変更して実行した結果を示す。横軸はコア間の通信レイテンシを表し、縦軸はキャッシュコアを使用しない版と使用する版の速度比を表す。

ソートサイズが小さい時には L1 キャッシュでのヒットが多く、キャッシュコアの効果が小さい。逆にソートサイズが大きい時には L1 キャッシュでのミスが多くなり、キャッシュコアの効果が大きくなる。ソート

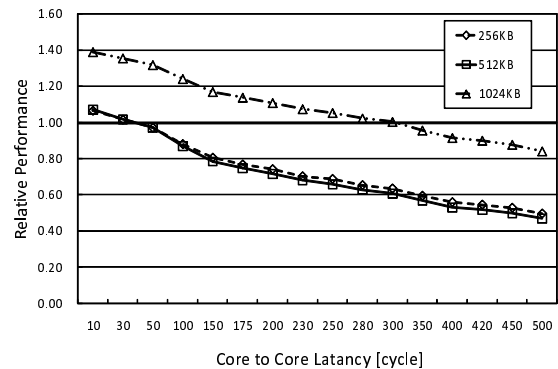


図3 ソフトウェア L1 キャッシュのみの性能に対するキャッシュコアを追加した場合の相対性能。

サイズが 1024KB の場合に、コア間レイテンシが 300 サイクル以下の場合にキャッシュコアによる速度向上を得る。コア間レイテンシが 0 の場合には、40%程度の速度向上を達成する。

4. おわりに

マルチコア・プロセッサにおけるデータ供給を支援するコアとして、キャッシュコアの可能性を検討した。

実験結果よりキャッシュコアが有効に働く可能性を確認できた。また、本稿ではデータを示していないが、キャッシュコアを使用しキャッシュヒット率を向上させるすることで、メインメモリへのアクセスの大幅な削減が見込める。

今後、多くのアプリケーションにおけるキャッシュコアの効果を調べることや、サイクルレベルのシミュレータを用いた精度の高い測定を行う必要がある。

加えてキャッシュコアを複数使用する場合の効果も検討する。

謝辞 本研究の一部は、科学研究費補助金若手研究 (B) 課題番号 18700042 「投機技術を積極的に利用するチップマルチプロセッサに関する研究」の助成による。

参考文献

- 1) 佐藤真平, 藤枝直輝, 田原慎也, 吉瀬謙二: 実用的かつコードのシンプルさを追求した Cell BE の機能レベルシミュレータ SimCell の設計と実装, コンピュータシステム・シンポジウム Com-Sys2007(2007).
- 2) Kistler, M., Perrone, M. and Petrini, F.: Cell Multiprocessor Communication Network: Built for Speed, *IEEE micro*, Vol.26, No.3, pp.10-23(2006).
- 3) 佐藤芳紀, 神酒 勤: 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, HPC-110, pp.13-18(2007).