

細粒度分割キャッシュのためのロード命令毎の特徴を考慮した挿入位置調整手法

学籍番号 23413523 氏名 小田 遼亮

指導教員名 津邑 公暁 准教授

1 はじめに

これまでゲート遅延に対し、配線遅延が相対的に増大してきた事により、配線遅延がプロセッサ性能に与える影響が大きくなった事で、配線遅延を隠蔽するキャッシュメモリの重要性が高まってきた。この一方で、集積回路の微細化に伴うリーク電流の増大による消費電力及び発熱量の増大といった問題から、プロセッサの動作周波数の向上は困難になってきている。このため、現在では消費電力や発熱量の問題を解決しつつプロセッサあたりの処理能力を向上させる、マルチコアプロセッサが広く普及している。

さて、このようなマルチコアプロセッサではキャッシュを有効活用するため、複数のコアがキャッシュを共有するケースが多い。ここで、複数のコアによって共有されるキャッシュ上では、複数のプロセスがエントリを追い出し合う干渉が発生し、キャッシュ性能が著しく低下するケースがある。このため、干渉の発生を抑制する手法として、各コアに占有領域を与えるキャッシュ分割手法が提案されている。

このようなキャッシュ分割手法の 1 つである Vantage[1]では、多くの分割領域を管理できる。しかしながら、Vantage では多くの分割領域を効率的に管理する手法について深く考察されている一方、その追い出しアルゴリズムの選択については深く考察されていない。そこで、本研究では Vantage に適した追い出しアルゴリズムとして、エントリの挿入位置を動的に調整する手法を提案する。

2 既存の追い出しアルゴリズム改良手法

これまで提案されてきた追い出しアルゴリズムの改良として、キャッシュ上に配置されてから再参照された事のある既参照エントリを、再参照された事が無い未参照エントリよりも優先する手法が多く存在する。このような手法として、Jaleel ら[2]はエントリの挿入位置を調整する Re-Reference Interval Prediction (RRIP)を提案し、Khan ら[3]

はキャッシュ上に配置されてから再参照された事の無い未参照エントリの利用数を指定する Dynamic Segmentation を提案した。これらの手法はいずれも低いハードウェアコストで実現可能でありながら、非常に高い性能を引き出す事ができる。そこで、本提案手法ではこれらの手法に着目し、より高性能な追い出しアルゴリズムを提案する。

3 細粒度分割キャッシュにおける挿入位置調整

本研究では、既存の既参照エントリ優先方式の内、未参照エントリ利用数を指定する方式と挿入位置を指定する方式の、どちらがより高性能な追い出し方式であるかを予備実験により評価した。この結果から、未参照エントリ利用数を指定する方式は性能が著しく低下するケースがあったのに対して、挿入位置を指定する手法は性能を安定して向上させる事が確認できた。また、未参照エントリの利用数を指定する方式を Vantage 上を実現する場合には、その性能低下の回避が困難である。このため、本研究では Vantage 上での既参照エントリ優先方式として、挿入位置を調整する手法を提案する。

Vantage 上で挿入位置を指定するためには、挿入時のタイムスタンプを指定する必要がある。これに加えて、挿入位置を指定したモデルにおける実行時の性能を知るためには、サンプリングモニタを利用する必要がある。このモニタを利用して性能を観測するためには、優先される既参照エントリの割合を知る必要がある。そこで、本提案手法では挿入時に設定するタイムスタンプとして Insertpoint を用意する。そして、優先する既参照の目標値を設定し、Insertpoint の値がこの目標値に近づくよう一定間隔毎に調整する事により、指定した割合の既参照エントリを優先させる。

ここで、1つの Insertpoint を利用して指定する事のできる挿入位置は 1 箇所ではない。しかしながら、複数箇所からエントリの挿入位置を選択する事ができれば、更なる性能向上が可能であると考えら

れる。そこで、複数箇所からの挿入位置選択を実現するため、複数の Insertpoint を用意する。そして、各 Insertpoint に対応するモニタにおける性能を比較する事によって、最も性能を向上させる挿入位置を選択する。これにより、複数箇所からの挿入位置選択を実現する事が可能になる。

4 ロード命令毎の特徴を考慮した挿入位置調整

ロード命令毎に、アクセスするアドレスの数や、エンタリを再参照する間隔には特徴がある。このため、ロード命令毎のアクセスパターンに着目し、追い出し順を変化させれば、更にキャッシュ性能を向上させる事ができると考えられる。

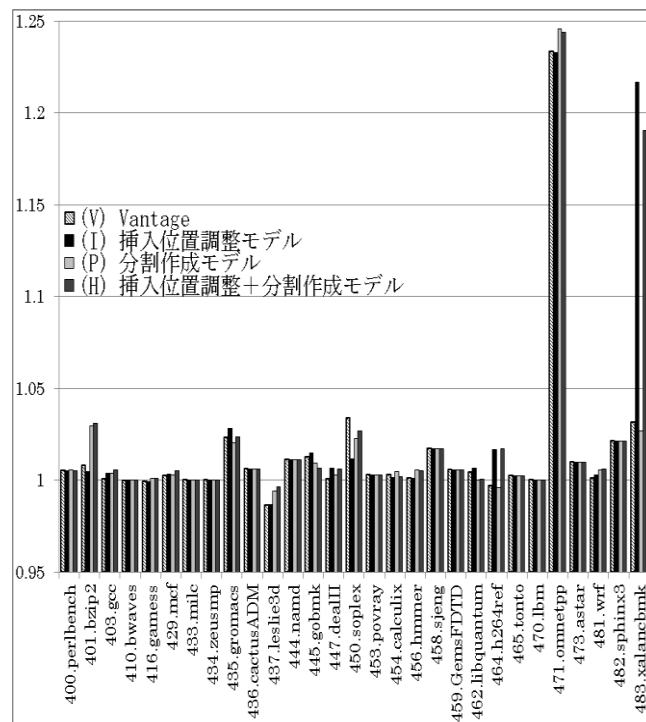
ここで、ロード命令毎のアクセスパターンを考慮する手法としては、浅見ら[4] が提案したロード命令単位での分割領域割り当て手法が存在している。この手法では、プロセス間だけでなく単一プロセス内でも干渉が発生する事に着目し、ロード命令単位で分割領域を割り当てる。そして、各ロード命令が必要とするエンタリをキャッシュ上で保護する事により、性能向上を図る。このように、浅見らの手法は単一プロセス内で発生する干渉の抑制を目的としているため、追い出しアルゴリズムの選択については考察しておらず、その追い出しアルゴリズムとしては LRU 方式を採用している。

しかしながら、ロード命令毎のアクセスパターンを考慮して追い出しアルゴリズムを変化させれば、更に性能を向上させる事ができると考えられる。そこで、本提案手法では前章で述べた追い出しアルゴリズムを、ロード命令単位での分割領域割り当て手法と組み合わせ、分割領域毎に別々の挿入位置を選択する手法を提案する。

5 評価

評価対象のプログラムとして SPEC CPU2006 の 29 プログラムを選択し、500M サイクルのスキップ後 500M サイクル実行した場合の IPC を既存の Vantage と比較した。

評価の結果、471 において全てのプログラムが大幅に性能向上した。これは、Vantage を利用した場合の連想度向上が非常に大きかったためである。また、483 では挿入位置調整モデル(I)が大きく性能を向上させている。これは、483 が既参照エンタリを優先する事で大きく性能が向上するワーキングセットの



非常に大きなプログラムであったためである。(P) は 401 や 464 で大きく性能を向上させた。そして、これらを組み合わせたモデル(H)は平均的に見て、その性能を最も向上させた。

6 おわりに

本研究では細粒度分割キャッシュ上に適した追い出しアルゴリズムを提案した。今後の課題としてハードウェアコストの削減が挙げられる。

7 参考文献

- [1] Sanchez, D. and Kozyrakis, C.: Vantage: Scalable and Efficient Fine-Grain Cache Partitioning, Proc. 38th Annual Int' l Symp. on Computer Architecture (ISCA), ACM, pp. 57-68 (2011).
- [2] Jaleel, A., Theobald, K. B., Steely, S. C. and Emer, J.: High performance cache replacement using re-reference interval prediction (RRIP), Proc. 37th Annual Int' l Symp. on Computer Architecture (ISCA), ACM, pp. 60-71 (2010).
- [3] Khan, S. M., Wang, Z. and Jimenez, D. A.: Decoupled dynamic cache segmentation, Proc. 18th IEEE Symp. on High Performance Computer Architecture (HPCA), IEEE Computer Society, pp. 1-12 (2012).
- [4] 浅見公輔, 倉田成己, 塩谷亮太, 三輪忍, 五島正裕, 坂井修一: 命令グループごとのキャッシュ・パーティショニングの予備評価, 情報処理学会研究報告 2012-ARC-201, No. 5, pp. 1-8 (2012).