

MorphCache: A Reconfigurable Adaptive Multi-level Cache Hierarchy

著者: Shekhar Srikantaiah, Emre Kultursay, Tao Zhang, Mahmut Kandemir, Mary Jane Irwin, and Yuan Xie

出典: 2011 IEEE 17th International Symposium on High Performance Computer Architecture, pp.231-242, 2011.

発表者: 高性能コンピューティング講座 本多・近藤研究室 1353006 金 東賢

1 はじめに

1つのチップに複数のコアを搭載したチップマルチプロセッサ (CMP) ではプロセスの並列実行数の増加に伴い、CPU 全体ではシングルコアに比べ多くのメモリアクセスが発生する。そのため、あるコアの使用しているデータが他のコアによって追い出されてしまうキャッシュ競合によるキャッシュミスの増加が問題となっている。その解決手法として多くの手法が提案されている。しかし、従来手法ではキャッシュ構成は静的に決められており、アプリケーションの特性や、実行サイクルごとに最適なキャッシュ構成が異なることを考慮していない。

そこで本研究ではキャッシュ構成をアプリケーションの実行サイクルごとに変更できるように設計し、アプリケーションの特性や実行サイクルごとに最適なキャッシュ構成を提供する MorphCache 機構を提案し、スループットの向上を目指す。

2 キャッシュ構成の問題点

CMP では、キャッシュ階層内の構成は大きく分けて複数のコアからアクセスできる共有型と1つのコアのみがアクセスできる非共有型に分けられる。共有型のメリットとしてコア間でキャッシュ容量を柔軟に分配できるため、あるコアの使用していない領域を他のコアが使用できることが挙げられる。非共有型のメリットとして、他のコアからのアクセスが無い場合、あるコアの使用しているデータが他のコアによって追い出されてしまうキャッシュ競合が起きないことなどが挙げられる。

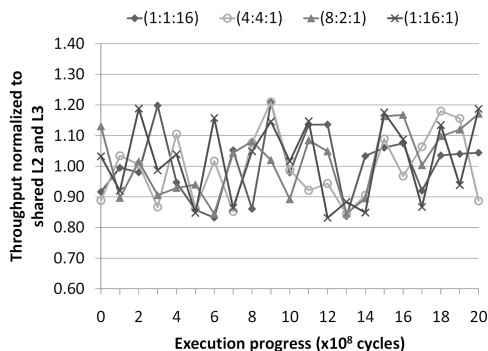


図 1: 実行サイクルごとのスループット

このようなキャッシュ構成は処理性能に大きく影響し、適切な構成にすることができればキャッシュミスが減らし、スループットの向上につながる。予備実験として図1にキャッシュ構成を変更させ、実行サイクルごとにスループットを測定したものを示す。図1よりキャッシュ構成によりスループットが変化していることが分かる。また、実行サイクルごとにも同じ現象が見受けられる。このことから、実行サイクルごとに最適なキャッシュ構成を提供できればスループットの向上につながると考えられる。

3 提案手法: MorphCache

本研究では、アプリケーションの実行サイクルごとに共有型や非共有型への変更を動的に行い、最適なキャッシュ構成を提供する MorphCache 機構を提案する。MorphCache 機構の概要図を図2に示す。MorphCache 機構ではキャッシュ階層をコア数で分割したキャッシュスライスを作成し、実行サイクルごとにスライスを共有型、非共有型へ変更することでキャッシュ構成の再構成を行う。キャッシュ構成の再構成には2つの指標を用いる。

1つ目の指標はキャッシュスライスの使用率である。キャッシュスライスの使用率が高い場合はキャッシュ容量を多く割り当て、使用率が低い場合はキャッシュ容量の割り当てを減らすことでキャッシュミス削減する。本研究では使用率の高いキャッシュスライスと低いスライスを1つのスライスに統合し共有型とすることで、使用率の低いスライスから使用されていない容量を使用率の高いスライスに割り当てる。

2つ目の指標はスライス間での同一アドレスデータの利用率である。同一アドレスデータが複数のスライス間に存在する場合、重複データによりキャッシュ容量を無駄に使用してしまう。この場合、同一アドレスデータが存在するスライス同士を統合し、共有型とすることで、重複データを削減することができる。

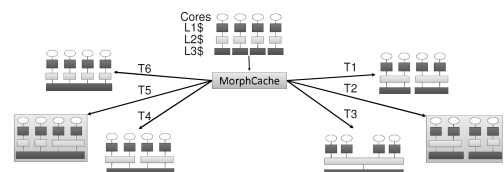


図 2: MorphCache

しかし、2つの指標を同時に知る機構は存在しないため、本研究では新たな推定機構である Active Cache Footprint(ACF) を提案する。

3.1 Active Cache Footprint(ACF)

キャッシュスライスの使用率、コア間の同一アドレスデータの利用率を推定する機構として Active Cache Footprint(ACF) を提案する。ACF ではキャッシュスライス内のデータへアクセス情報を記録するビット列を用意し、このビット列からキャッシュ構成の再構成に必要な指標の推定を行う。

キャッシュスライスの使用率は、データへのアクセス情報を参照し異なるデータへのアクセス数が基準値を超えているかどうかで判定する。同一アドレスデータの利用率は、2つのスライス間のアクセス情報を比較し、同一アドレスデータへのアクセス数が基準値を超えているかどうかで判定する。

アクセス情報の記録にはデータのタグ情報を使用し、記録されているタグ情報の量やスライス間で同じタグ情報がどれだけ存在するかで判断する。しかし、タグ情報をそのまま記録するには多くのハードウェア容量が必要である。そこで、本研究ではハッシュ関数を用いタグ情報をハッシュ値へ圧縮し、ハードウェア容量を削減すると共に効率的にタグ情報を記録した [1]。ハッシュ関数にはタグ情報とハードウェア上に用意されたある配列 n ビットと XOR をとり、 n ビットを生成する関数を使用した。 n は任意に決定できるが、小さいとハッシュ競合が起りやすくなるため、実験から適切な n を求めた。ハッシュ値は 1 ビットの配列である Active Cache Footprint(ACFV) に記録した。ACFV を図 3 に示す。

なお、本研究では2つの指標の推定の際、すでに使用されていないデータを考慮し、不適切なキャッシュ構成に再構成されてしまうことを防ぐため、キャッシュ構成の再構成を行う 30 億サイクルごとに ACFV をリセットした。

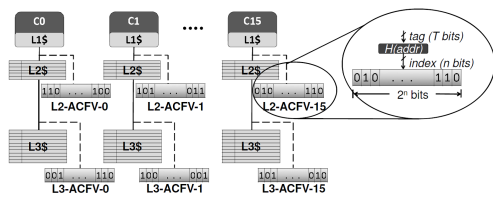


図 3: ACFV

3.2 キャッシュ再構成ポリシー

キャッシュ構成の再構成を行うかどうかの判断基準は実験から算出し、ACFV の 1 のビットの数の比率が 60% 以上の場合キャッシュスライスの使用率が高い、30% 以下の場合使用率が低いとした。同一アドレスデータの利用率は 2つの ACFV を比較し、対応する 2つのビットが両方も 1 の数の割合が多い場合、同一アドレスデータの利用率が高いとした。

また、キャッシュスライスを統合する組み合わせは、キャッシュの使用率の 60% 以上のスライスと 30% 以下のスライスの組み合わせ、キャッシュ使用率が 60% 以上で同一データの利用率が高いスライス同士の組み合わせとした。

4 評価

実験評価には 16 コアスーパースカラプロセッサを想定し、シングルスレッドアプリケーション、マルチスレッドアプリケーションを動作させ評価を行った。L2、L3 キャッシュはそれぞれ 16 スライスあり 1 スライス当たり 256kB、1MB とした。またキャッシュ構成を (16:1:1) で正規化し、静的にキャッシュ構成を変換させた場合と MorphCache のスループットを比較した。キャッシュ構成の再構成は 30 億サイクルごとに行った。

図 4 にシングルスレッドアプリケーションを実行した際の結果を示す。16 コアに SPEC CPU2006 から異なるアプリケーションを 1 つ割り当て実験を行った。実験結果より (16:1:1)、(1:1:16)、(4:4:1)、(8:2:1)、(1:16:1) からそれぞれ 29.9%、29.3%、19.9%、18.8%、27.9% 向上した。

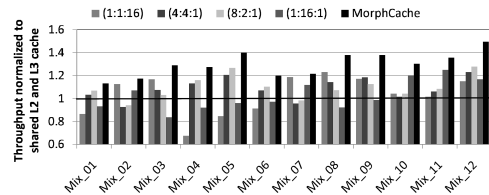


図 4: シングルスレッドアプリケーションでの比較

次に図 5 にマルチスレッドアプリケーションでの測定結果を示す。PERSEC から 1 つのアプリケーションを割り当て 16 コアで並列処理を行い評価を行った。実験結果より (16:1:1)、(1:1:16)、(4:4:1)、(8:2:1)、(1:16:1) からそれぞれ 25.6%、30.4%、12.3%、7.5%、8.5% 向上した。

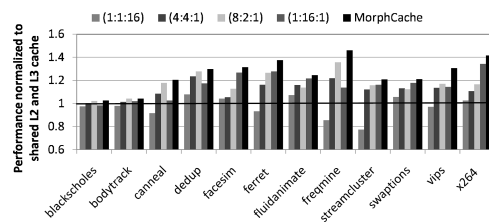


図 5: マルチスレッドアプリケーションでの比較

5 おわりに

本研究では、キャッシュ構成を動的に再構成する MorphCache 機構を提案した。評価結果より静的キャッシュ構成に比べ高いスループットが得られ、提案手法の有用性を示した。

参考文献

- [1] M. V. Ramakrishna, *et al.* "Efficient hardware hashing functions for high performance computers", IEEE Trans on Computer, Vol.46, No.12, 1997.