

ヘルパースレッドによるキャッシュ置換ポリシーの実行時制御の研究

高性能コンピューティング学講座 本多・近藤研究室

1053002 井上 功一

主任指導教員：近藤正章

1 概要

プロセッサの高性能化は、最近では複数のプロセッサコアを1チップに搭載したマルチコアプロセッサによる並列処理を行うことで高性能化を図る手法が主流となっている。しかし、メインメモリとの通信がボトルネックとなりプロセッサとメモリ間の性能差拡大のため、プロセッサの性能はコア数の増加とともに線形に向上しない。そのため、プロセッサの性能向上のためにはメモリ性能の改善が極めて重要となる。

そこで本研究では、一部のプロセッサコアを演算用ではなく、メモリ性能向上用であるヘルパーコアとして活用することでプロセッサの性能向上を目指す。ヘルパーコアで演算用コアにおいて実行されているプログラムの特徴に応じてそのコアのキャッシュ置換ポリシーを最適化するヘルパースレッドを実行させ、メモリ性能の向上を図る。

2 研究背景

2.1 関連研究と本研究の位置づけ

マルチコアプロセッサにおいて一部のコアを用いてメモリ性能を向上させる関連研究として、未使用のコアにおいてプリフェッチ用スレッドを実行することでメモリアクセスレイテンシを隠蔽する手法 [1] や未使用コアのキャッシュを演算用コアに貸与する手法 [2] などがある。

本研究は様々なプログラムにおいてメモリ性能の向上を図るため、制約を受けやすいハードウェアによる手法ではなく、実行中のプログラムの特徴に沿って柔軟に置換ポリシーを設定可能なソフトウェアによる手法をとる。

2.2 キャッシュ置換ポリシー

キャッシュはメインメモリに比べ容量が少ないため、キャッシュ内に空き領域がなくなった際に新たなデータを格納するために置換ポリシーに沿って不要なデータを選択し追い出す。代表的な置換ポリシーとして Least Recently Used (LRU) や Most Recently Used (MRU), RANDOM などがある。キャッシュ置換ポリシーは格納されているデータのアクセス履歴情報などを用いて置換対象を決定する。

MRU

最も最近アクセスされたデータを置換対象とする。

LRU

最も最近アクセスされていないデータを置換対象とする。

RANDOM

ランダムにデータを選択し、それを置換対象とする。

3 提案手法

3.1 ヘルパースレッド

従来のマルチコアプロセッサでは全てのコアは演算用として利用されるが、本研究では図1のように一部のコアをメモリ性能向上に活用するヘルパーコアとする。そしてそのコアにおいて実際にメモリ性能を向上させる役割を果たすヘルパースレッドを実行させる。ヘルパースレッドは通常のプログラムの演算処理は行わず、演算用コアの性能向上を図るためだけに実行される。

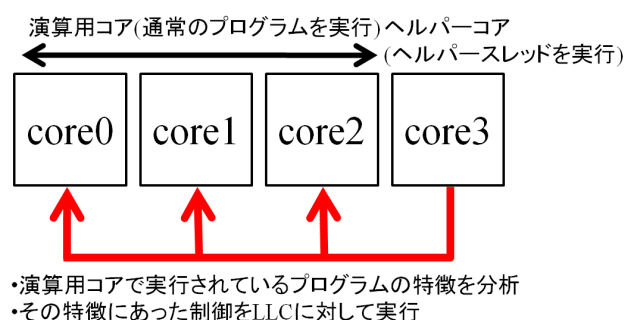


図1: 想定しているマルチコアプロセッサ構成

ヘルパースレッドは演算用コアで実行しているプログラムの特徴を解析し、その特徴にあった制御をキャッシュに対して行う。行う制御としては以下が考えられる。

- 共有キャッシュの占有率制御
- キャッシュ置換ポリシーの擬似的変更制御

3.2 想定するハードウェア構成

本研究で想定するプロセッサのキャッシュは命令キャッシュ(IL)、1次データキャッシュ(DL1)、2次データキャッシュ(DL2)、共有キャッシュ(LLC)で構成され、具体的なハードウェアの構成としては以下と想定している。

- LLCの置換ポリシーは初期設定としてLRUで構成する。
- 各コアのプログラムからLLCのLRU情報にアクセスできるものとする。

3.3 共有キャッシュの占有率制御

LLCは全コアでの共有資源であるため、プロセッサコア間の干渉が起こりやすい。その干渉の一部を解決するために本制御はキャッシュの時間的局所性に着目する。実行するプログラムによって扱うデータの再利用性は異なるが、再利用性の高いデータを扱うプログラムのLLC内での占有率を高めることで再利用性の高いデータが追い出されることを防ぐ。

3.4 キャッシュ置換ポリシーの擬似的変更制御

これまでの研究によって、全てのプログラムに対してLRUによる置換が最適であるわけではなく、MRUやRANDOMなど他の置換ポリシーが最適であるプログラムも存在することがわかっている。そこで本制御ではヘルパースレッドで実行中のプログラムの特徴を捉え、それにふさわしい置換ポリシーに変更することを行う。ハードウェア上ではLRUで構成されているため、何もしなければLRUの置換ポリシーで置換されるが、そのLRUで置換する際に用いるLRU情報をヘルパースレッドが操作することで他の置換ポリシーによる置換を擬似的に実現する。

4 進捗状況

現在はx86系CPUのシミュレータであるzesto x86 simulatorを用いて、SPEC2006ベンチマークを実行させ、その時のクロックあたりの命令実行数を性能指標とし、評価を行っている。

SPEC2006ベンチマークを様々な置換ポリシーを設定したCPUで実行させ、各置換ポリシーとプログラムの特徴との関係を明らかにしている。表1、表2から、LRUは多くのプログラムにおいてキャッシュミス率が低くなりやすいが、LLCへのアクセス回数が少ない場合、MRUの方がミス率が低くなる傾向であることがわかった。

表1: LLCへのアクセス回数の少ないプログラムを実行した際のキャッシュミス率 [%]

	LRU	MRU	RANDOM
IL1	0.354	1.797	0.446
DL1	0.345	2.834	0.444
DL2	0.051	0.573	0.065
LLC	0.046	0.040	0.045

表2: LLCへのアクセス回数の多いプログラムを実行した際のキャッシュミス率 [%]

	LRU	MRU	RANDOM
IL1	0.264	1.240	0.332
DL1	1.823	3.522	1.914
DL2	0.515	1.102	0.558
LLC	0.785	1.260	0.864

以上の結果から、ヘルパースレッドが演算用コアで実行中のプログラムの特徴を解析した際、LLCへのアクセス回数が少ない特徴を持つことがわかれば、その際はキャッシュ置換ポリシーをLRUから擬似的にMRUに変更させることでメモリ性能が向上し、プロセッサの性能も向上すると考えられる。

5 今後の課題

5.1 プログラムの特徴の解析

各コアにおいて実行しているプログラムの特徴の解析は事前実行による結果ではなく、実行時に解析を行うが、プログラムの特徴を正しく判断するために、解析を行う際のサイクル数の範囲とLLCへのアクセス回数の閾値などの適切な値を求める必要がある。

5.2 シミュレータへの実装

サイクル数の範囲とLLCへのアクセス回数の閾値を設定できたあとは、その値で解析を行うヘルパースレッドをzesto x86 simulatorで実装を行い、性能の向上率を評価する。

参考文献

- [1] I.Ganusov and M.Burtscher. Efficient Emulation of Hardware Prefetchers via Event-Driven Helper Threading. In Proc. PACT2006, pp.144-153, 2006.
- [2] 福本 尚人, 井上 弘士, 村上 和彰. 演算/メモリ性能バランスを考慮したマルチコア向けオンチップメモリ貸与法. HPCS2011, pp.130-139, 2011.