

階層型キャッシュシステムにおける高効率なブロック配置法

Huh, Younsuk・請園 智玲・田中 清史 (北陸先端科学技術大学院大学)

1 はじめに

現在主流となっている多くのプロセッサは、階層化されたキャッシュの管理方法として Inclusion Property が採用されている。Inclusion Property 方式は上位キャッシュのブロックのコピーを下位キャッシュで持つため、プロセッサ間のコヒーレンス処理が単純となる利点があるが [1]、同一のブロックのコピーが上位、下位キャッシュで複数存在するため、メモリ資源を浪費する欠点がある [2]。一方、Exclusive Property 方式では、省メモリ資源の特性を持つ代わりにコヒーレンス維持のオーバーヘッドが存在する。

プログラム実行において参照されるデータは異なる局所性を持つが、その違いによって配置すべきキャッシュ階層が異なる。本報告ではその違いに着眼し、適切なブロック配置により解決する方法を提案する。

2 局所性に基づくブロック配置

本報告ではキャッシュのデータの参照を以下のパターンに分類し、適切なキャッシュ階層に配置する手法を提案する。

1. 長期的に高い局所性をもつデータ
長期的に高頻度で参照されるデータを含むブロックは、上位キャッシュで置換された後も、近い将来再度参照される可能性が高い。このようなブロックは上位キャッシュと下位キャッシュ両方に配置する。
2. 長期的に低い局所性をもつデータ
長期的な再利用性があるが短期的には参照頻度が低い（すなわち、参照間隔が大きい）データからなるブロックは、最下位キャッシュにのみ配置することにより、上位キャッシュ内の局所性の高いブロックを置き換えることを防ぐ。ただし、ブロック内の空間的局所性に対処するために、当該ブロックを後述するバッファに格納する。
3. 短期的に高い局所性を持つデータ
短期的に高頻度で参照されるが長期的には再利用性が低いデータからなるブロックは、上位キャッシュにのみ配置することにより、下位キャッシュ内の再利用性のあるブロックを置き換えることを防ぐ。
4. 時間的局所性のないデータ
再利用性の無いデータからなるブロックは、上位、下位キャッシュに配置しないことにより、局所性の高いブロックを置き換えることを防ぐ。ただし、ブロック内の空間的局所性に対処するために、当該ブロックを後述するバッファに格納する。
5. プロセッサ間共有データ
複数のプロセッサで共有されるデータは、一般的に下位キャッシュ間のスヌーピング機構により一貫性が維持される。Exclusion Property 方式では、上位キャッシュもスヌーピングの対象とする必要があるため、一貫性維持処理のオーバーヘッドが大きくなる。したがって、共有データを含むブロックは参照パターンによらず下位キャッシュに配置することにより、オーバーヘッドを削減する方針を採る。上位キャッシュに配置するか、あるいはバッファに配置するかは参照パターンに依存する。（すなわち、上記の 1～4 のいずれかに該当する。）

3 提案手法

本提案で想定するキャッシュシステムにはバッファが存在する。バッファは L1 キャッシュと同列参照される位置に存

在し、プログラムの要求する空間的な局所性のみを持つ参照列に対応する機構であることから数ブロックほどの極小さいもので十分である。バッファは L1 キャッシュと並列参照し、L1 キャッシュと同等かそれ以上早く参照が完了するサイズを想定する。

提案するキャッシュシステムを扱うロード／ストア命令を 4 つのカテゴリに分けた。L1 キャッシュとバッファは第一階層のキャッシュとして扱われ、同時に参照される。第一階層キャッシュがミスした場合に L2 キャッシュが参照される。各ロード／ストア命令は第一階層キャッシュヒット時は同じ動作をする。第一階層キャッシュがミスした後、L2 キャッシュのヒット／ミス時に各ロード／ストア命令の動作が異なる。各ロード／ストア命令の L2 キャッシュのヒット／ミス時の処理を以下に示す。

1. ld / st with L1・L2
ブロックを L1 キャッシュと L2 キャッシュ両方に配置する命令。通常のプロセッサが備えるメモリアクセス命令と同様の動作を行う。L2 ヒット時は当該ブロックを L1 キャッシュへロードする。L2 ミス時は当該ブロックを L2 と L1 キャッシュへロードする。
2. ld / st with L2・B
ブロックを L2 キャッシュとバッファに配置する命令。L2 ヒット時は当該ブロックをバッファへロードする。L2 ミス時は当該ブロックを L2 とバッファへロードする。
3. ld / st with L1
ブロックを L1 キャッシュにのみ配置する命令。L2 ヒット時は当該ブロックを L1 キャッシュへロードする。L2 ミス時は当該ブロックを L1 へのみロードする。
4. ld / st with B
ブロックをバッファにのみ配置する命令。L2 ヒット時は当該ブロックをバッファへロードする。L2 ミス時は当該ブロックをバッファへのみロードする。

以上のメモリアクセス命令により、指定キャッシュへの配置の最適化を行う手段を導入することができる。

4 まとめ

本報告では、Inclusion Property 方式のメモリ資源の浪費と Exclusion Property のコヒーレンス処理のオーバーヘッドの問題に対して、キャッシュブロックの適切な配置により解決するキャッシュメモリシステムとそれを使う命令の提案を行った。今後は導入した命令をバイナリに適用する方法を検討する。命令の適用に関して、現在はシミュレーションベースの事前実行で取得したメモリアクセストレースを利用したバイナリ変換手法を検討中である。トレースを用いて命令毎の参照パターンを解析することにより、各命令の適用位置を特定することが可能になると考える。

参考文献

- [1] J. Baer, et al., "On The Inclusion Properties for Multi-Level Cache Hierarchies", Proc. of the 15th ISCA, pp. 73-80, 1988.
- [2] M.Zahrán, et al., "Non-Inclusion Property in Multi-level Caches Revisited", International Journal of Computers and their Applications, VOL 14; No. 2, pp. 99-108, 2007.