# メモリアクセスパターンの局所性に基づくキャッシュメモリ構成方式の検討

金指和幸[*1], 中村友洋[*2], 吉瀬謙二[*2], 田中英彦[*2]
[*1]富士通 (株), [*2]東京大学工学部

## 1　はじめに

近年のプロセッサの処理速度の高速化に伴い、主記憶へのアクセスの遅延がシステムのボトルネックになっている。これを改善するためにキャッシュメモリが用いられるが、キャッシュミス時に外部の主記憶アクセスにかかる遅延時間が非常に大きくそのミス率が問題となる。そのキャッシュミスを低減するために SPARC シミュレータを利用したトレース・ドリブン・シミュレーションにより、キャッシュ・アクセス・パターンの解析を行ない、その局所性を利用したキャッシュ構成方式について検討する。

## 2　従来のキャッシュ構成

キャッシュミスの原因を調査すると一般的に知られているように、大きく分けて初期参照ミス、競合ミスの２つに分けることができる。その初期参照ミスと競合ミスの発生の割合を見ると競合ミスの方が多い。(表 1)

そこで、アクセス速度が速いダイレクトマップを使用した場合に、競合ミスを削減するために miss cache,vitium cache[1] が提案されている。

1. miss cache 方式はキャッシュミスしたデータを一度バッファに格納しておき、そのデータにヒットした場合にのみキャッシュへと登録する。

2. vitium cache 方式はキャッシュミスによって追い出されたデータをバッファに保管しておく。

この２つの方式のどちらかを使用すればバッファのエントリの数だけのミスについて競合ミスを回避することができる。

しかし、競合ミスを回避できる範囲がエントリの数に依存しておりエントリを越える先のミスとの競合ミスには効果がない。

そこで上記の方式で回避できなかった競合ミスのアドレスパターンについて調べると、発生したミスとその次のミスでアドレス値が近いことに気づく。これはプログラム自体が持つアクセスの特徴であり常に存在するが、多量の競合ミスの発生によりそのような特徴を持つアクセスパターンが分断されていた。その競合ミスの減少によってアクセスの特徴が見えるようになったと考えられる。このような空間的ミスの改善方法にはラインサイズの増加やプリフェッチの技術がある。

そこで今回は miss cache を利用した競合ミスの削減とラインサイズ増加のよるミスの削減を同時に行なう時のパフォーマンスの改善について検討する。

Preliminary Evaluation of Cache Memory for Memory access pattern
Kazuyuki KANAZASHI[*1], Tomohiro NAKAMURA[*2], Kenji KISE[*2],Hidehiko TANAKA[*2]
[*1]Fujitsu Limited
[*2]Faculty of Engineering, University of Tokyo

## 3　検討するキャッシュの構成

今回検討するキャッシュ構成を図 1 に示す。前述の miss cache のラインサイズを大きくしたものである。small cache には main cache の 1 ライン分ごとにヒット判定ビットを持たせる。

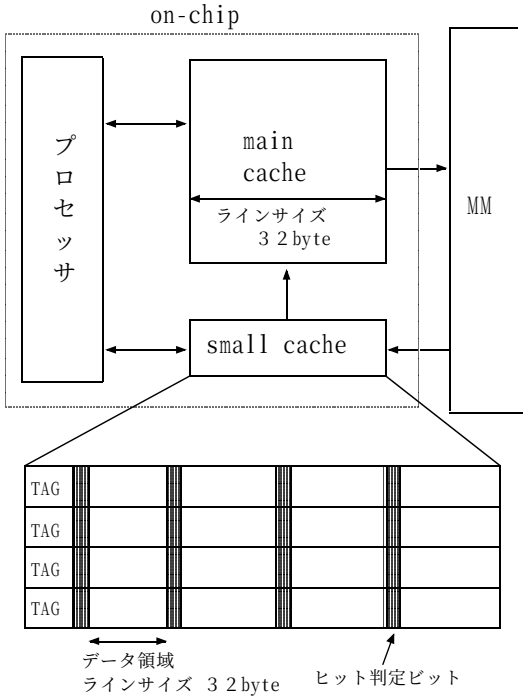


図 1: 構成図

キャッシュ動作について説明する。main cache にヒットであれば従来通りアクセスする。

main cache をミスし、small cache でヒットした場合には small cache からプロセッサへデータを渡し、使用されたことを示すヒット判定ビットをセットする。

main cache と small cache の両方にキャッシュミスが発生すると、メインメモリから small cache へとミスしたアドレスのデータがまず送られる。その時に small cache の 4 ライン中のどの領域に要求されたデータが置

かれたかを知るために該当するラインのヒット判定ビットをセットする。

さらに次のクロックで両方ともミスした場合には複数個用意した別のタグへと書き込む。すべてのタグが埋まった場合はランダムまたはＬＲＵによって追い出す。その時に small cache 中で使用された領域をヒットビットにて判断し、使用されているものについてはメインキャッシュへと書き込み、それ以外は捨てる。この動作はメインメモリにアクセスしている間 ( メインメモリのアドレス引き) に行なえる。この処理の終了後にミスしたアドレスのデータを small cache へと転送する。

## 4　実験

### 4.1　内容

SPARC シミュレータから得られるトレースデータを元に上記で示したキャッシュ構成でのキャッシュミス率の特性を調べる。サンプルとして使用したプログラムを下に示す。

表 1

| プログラム | 初期参照ミス | 競合ミス |
|---|---|---|
| *compress* | 13494 | 898737 |
| *espresso(bca)* | 1962 | 101238 |
| *espresso(cps)* | 2284 | 83992 |
| *espresso(ti)* | 1993 | 106020 |
| *espresso(tial)* | 1503 | 107046 |
| *sc* | 475 | 3561 |
| *jhd* | 596 | 4809 |
| *lha* | 731 | 124436 |

### 4.2　結果

データキャッシュにおいて、容量８ K バイト、ラインサイズ 32 バイトのダイレクトマップと、タグ 4、ライン領域 4 を追加した今回の構成との比較をした結果 (ダイレクトマップを 1 とした場合の割合) を図 2 に示す。

プログラムによっては約 30 %にミス率を削減できている。

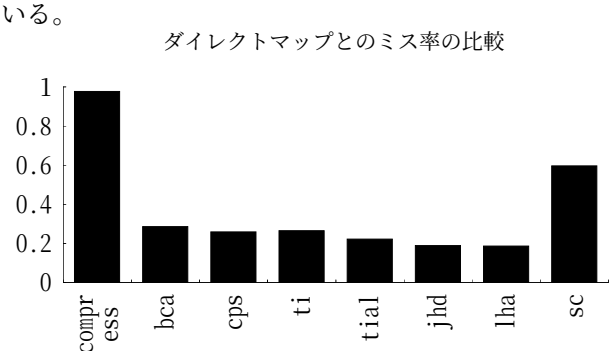


図 2: キャッシュミス率の減少の割合

次に、ダイレクトマップの構成に比べ、vitium cache と今回の構成でのキャッシュミスの削減率について図 3 に示す。

今回注目した競合ミスについて、ダイレクトマップとの比較では図 3 に示すように競合ミスを削減できることがわかる。また、victim cache のみの場合と比較しても、より効果的に競合ミスを削減できている。

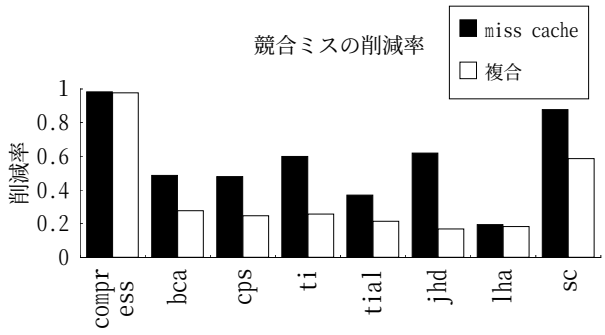


図 3: 競合ミス削減率

次に、ミスを起こす命令の中で、あるラインに注目しそのラインでキャッシュミスを起こすロード、ストア命令の距離 ( ミスを起こす命令がその間にいくつあるか) を図 4 に示す。

sc,compress で効果が小さいのは、図 4 からわかるように競合ミスの生じる命令間の距離が長く small cache が効かないためであると考えられる。

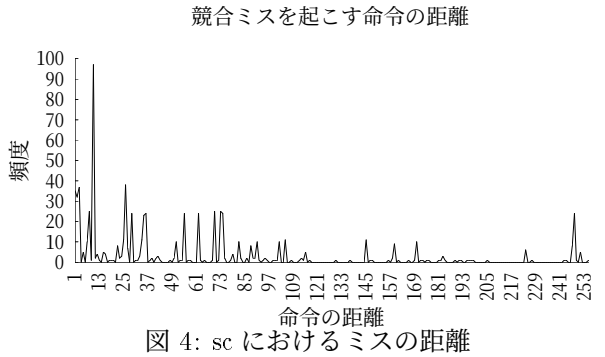


図 4: sc における ミスの距離

## 5　まとめ

ダイレクトマップとのミス率を比較すると、プログラムによるがこのような構成をとることによって 16 分の 1 のコスト上昇で約半分のミス率になることが確認できる。

今後の課題はプログラム compress,sc に見られるような同じラインを使用するキャッシュミスが遠い場合にどのようにして競合ミスを避けていくかが問題になっていく。

今回は考慮しなかったがライン長の増加によるオーバヘッドについても考慮する必要がある。

**謝辞**

本研究の一部は文部省科学研究費 (一般研究 (B) 課題番号 07458052「 大規模データパスプロセッサの研究」) による。

**参考文献**

[1] N.P.Jouppi , 'Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers' , 17th ISCA , 1990 , pp.364-373