

Trace Cache を用いた命令キャッシュ改善手法の提案

木町 貴一[†]

茨城大学大学院 理工学研究科

岡本 秀輔[‡]

茨城大学大学院 理工学研究科

概要

Trace Cache が、高いフェッチバンド幅をもたらすキャッシュ機構として近年のプロセッサで利用されるようになってきた。Trace Cache には“トレース”を単位として命令データが格納されるが、一度 Trace Cache からトレースが追い出されると、そのトレースを構成する基本ブロック情報は失われてしまう。そこで、Trace Cache から追い出されたトレースの基本ブロック情報に着目し、命令キャッシュの制御にフィードバックすることで、命令のプリフェッチや置き換えに役立つ手法を提案する。

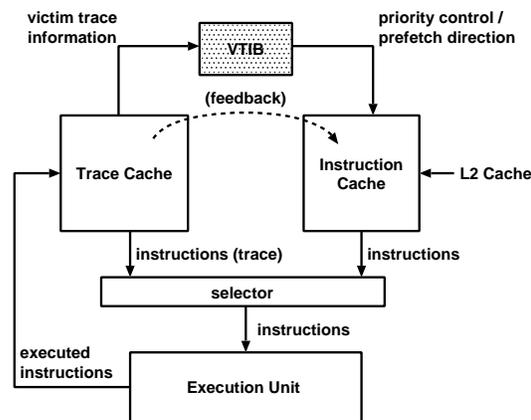


図 1: 提案手法の概要図

1 はじめに

Trace Cache [1] が、高いフェッチバンド幅をもたらすキャッシュ機構として近年のプロセッサで利用されるようになってきた。Trace Cache には、実行された命令列がその流れに沿った“トレース”の単位で格納され、また、“トレース”の単位で供給される。

文献 [1-3] での Trace Cache の構成においては、Trace Cache はそれ単独で従来の命令キャッシュに置き換わって利用されるのではなく、命令キャッシュと併用する形を採用している。プロセッサコアへ命令供給を行うとき、Trace Cache にヒットしない場合は従来同様に命令キャッシュからの個別のキャッシュラインごとに命令が供給される。一方、供給すべき命令列がトレースの形で利用可能な場合には Trace Cache から供給される。命令供給に際して、命令キャッシュではキャッシュライン内の分岐命令により供給数が制限を受けるのに対し、Trace Cache では分岐命令を跨るトレースの単位で供給可能であることから、有用性が認められてきた。

2 問題提起

過去の研究から、キャッシュサイズは相対的に Trace Cache を大きく、命令キャッシュを小さくする構成がよい結果が出ている。CPU 全体でのリソースの配分構成の関係上、搭載できるキャッシュのサイズは限られているので、Trace Cache のサイズを大きくするならば、命令キャッシュは相対的に小さくせざるを得ないだろう。Trace Cache がヒットし、ここから命令供給を行えるならそれにこしたことはないが、ミスの場合は依然として命令キャッシュに頼ることとなる。Trace Cache のサイズ分命令キャッシュのサイズを小さく抑えつつも、命令キャッシュの精度を極端に落とすことは許しがたい。

また、Trace Cache と命令キャッシュを併用する構成では、両者で同じデータを保持する可能性がある。これはキャッシュ利用効率に無駄が生じることになる。Trace Cache から追い出された命令は、後の利用に際して命令キャッシュ内に存在する方が良いだろう。一方、Trace Cache に存在する命令は、命令キャッシュには無くても良いかもしれない。Trace Cache にあるデータは命令キャッシュにある必要性は少なく、逆に Trace

“Improvement I-Cache using Trace Cache”

[†]Takaichi Kimachi; Graduate School of Science and Engineering, Ibaraki University

[‡]Shusuke Okamoto; Graduate School of Science and Engineering, Ibaraki University

Cache に無いデータが命令キャッシュに存在する必要性は高くなるだろう。

この問題を改善する方法として、Trace Cache と命令キャッシュに含まれる命令データの関係に着目し、トレースに含まれる情報から命令キャッシュを改善するフィードバック情報を提供することを考える。

3 手法

Trace Cache にはトレースを単位として命令データが格納されるが、一度 Trace Cache からトレースが追い出されると、そのトレースを構成する基本ブロック情報は失われてしまう。そこで、Trace Cache から追い出されたトレースの基本ブロック情報に着目し、命令キャッシュの制御にフィードバックする手法を提案する。

3.1 Victim Trace Info. Buf. (VTIB)

Trace Cache には命令データがトレースとして格納されている。しかし、Trace Cache ラインが置き換えられるときには、対象のトレースに含まれる命令の基本ブロック構成に関する情報も失われてしまう。そこで、トレースデータが Trace Cache から追い出される際には、そのトレースの構成情報を小さなバッファに残しておく。これを Victim Trace Information Buffer (VTIB) と呼ぶ。

VTIB はトレースの基本ブロック情報を保持するバッファで、キューとして構成する。基本ブロック情報とは、(1) ブロック先頭の命令のアドレス、(2) ブロック末尾の命令のアドレス (or 命令数)、で構成される。命令データそのものは含まないので、バッファサイズは小さく抑えられると考える。

VTIB からトレースの基本ブロック情報を取り出し、それに含まれる基本ブロックに関する命令データが命令キャッシュ内に存在するかどうかをチェックする。この判定結果により、以下の 2 つの処理を試みる。

3.2 命令キャッシュへの優先度制御

命令キャッシュが n -way のセット連想性を持つ場合、そのセット内ラインの置き換え優先度制御に VTIB の情報を用いる。VTIB と命令キャッシュの判定結果から、該当データが命令キャッシュ内に存在するなら、これ

は重複が起こっていることになるので、その命令キャッシュブロックの優先度を下げ、置き換えが起こりやすいように仕向ける。置き換えポリシーとして LRU (疑似 LRU) の利用を想定すると、その制御はリスト or エージングによってなされる。そこで、リストでの実装の場合には MRU よりに移動させて優先度を上げる。エージングでの実装では、age カウンタの値を下げる処理を行う。

3.3 命令キャッシュへのプリフェッチ

VTIB と命令キャッシュの判定結果から、命令キャッシュ内に該当データ無い場合には、命令キャッシュへのプリフェッチを行うことを考える。VTIB の情報からは、該当の命令データは一度は実行された実績があるにも関わらず、Trace Cache にも命令キャッシュには無いということが分かる。そこで、その命令データを 2 次キャッシュ等から命令キャッシュへ予めプリフェッチしておくことで、一度実行された実績のある命令は、Trace Cache が命令キャッシュのどちらか一方には存在しているように仕向け、その命令が再度必要になる場合に備える。

4 おわりに

本論文では、Trace Cache から得られるトレースの基本ブロックの情報を活用した、命令キャッシュの改善手法を提案した。今後、シミュレータによる提案手法の実装を行い、詳細なシステムの評価を通して、有効性を探っていきたい。

参考文献

- 1) E.Rotenberg, S.Bennett, and J.Smith; Trace Cache: A Low Latency Approach to High Bandwidth Instruction Fetching, Proceedings of the 29th Annual International Symposium on Microarchitecture (Nov 1996).
- 2) E.Rotenberg, S.Bennett, and J.Smith; A trace cache microarchitecture and evaluation. IEEE Trans. on Computers, 48(2) (Feb 1999).
- 3) D.Friendly, S.Patel and Y.Patt. Alternative Fetch and Issue Policies for the Trace Cache Fetch Mechanism. Proceedings of the 30th Annual International Symposium on Microarchitecture (Dec 1997).