C-026

キャッシュ・ミス頻発命令が性能に与える影響
Effects of Troublesome Memory Access Instructions on Processor Performance

堂後 靖博†1　井上 弘士†2
Yasuhiro Dougo　Koji Inoue

1. はじめに

現在，拡大を続けるプロセッサー主記憶間性能差の隠蔽を目的として，多くのプロセッサにはオンチップ・キャッシュが搭載されている．しかしながら，依然としてオフチップ・メモリに対するアクセス・レイテンシは増加の一途をたどっており，更なるキャッシュ性能の向上が必要となる．そこで我々は，キャッシュ・ミス頻発命令（以下，Troublesome:TS 命令と呼ぶ）に着目したメモリ性能向上技術に関する研究を行っている．具体的には，プログラム実行中，キャッシュ・ミスを頻繁に引起こす命令（以下，Troublesome:TS 命令と呼ぶ）を動的に検出し，その特徴解析を行い，TS 命令専用最適化を施す事で高い性能向上を実現する．

このような動的最適化を実装するためには，最初の処理である TS 命令の検出が極めて重要となる．そこで本研究では，ベンチマーク・プログラムを用いたシミュレーションを行い，①TS 命令の存在とその出現率，②TS 命令の生存期間，ならびに，③そのアクセス・パタンを解析する．また，これらの結果に基づき，TS 命令が性能へ与える影響を考察する．

2. TS 命令の解析

本稿では，以下３つの項目について解析を行う．

- TS 命令の全キャッシュ・ミス中に占める割合
- TS 命令の生存期間
- TS 命令のメモリ・アクセス分布

以降，プログラム実行において，n 番目に多くのデータ・キャッシュ・ミスを引起したメモリ・アクセス命令を TSn と表記する．例えば，TS1 は最も多くのミスを引起したメモリ・アクセス命令であり，TS2 は 2 番目に多くのミスを引起した命令となる．

2.1 実験環境

TS 命令の特徴解析を行うため，SPECint95 ベンチマーク（入力には train を使用）を用いたシミュレーションを行った．プロセッサ・シミュレータとしては，SimpleScalar[2] ツールセットを使用している．ここで，L1 データ・キャッシュは 32K バイト，ラインサイズ 32 バイト，連想度は 4 とした．また L2 キャッシュは命令・データ統合キャッシュであり，その容量は 256K バイト，ラインサイズは 32 バイト，連想度は 4 である．その他のプロセッサ構成に関しては，文献[1]で示された default 値を用いている．

†1 福岡大学大学院工学研究科電子情報工学専攻
†2 福岡大学工学部電子情報工学科

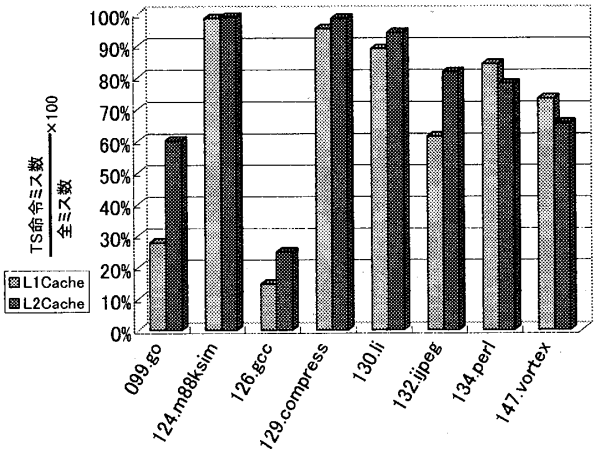

図 1：TS 命令出現頻度

2.2 TS 命令の存在

全キャッシュ・ミスにおいて，上位 10 個の TS 命令により生じたミスの割合を図 1 に示す．129.compress では，全ミスの約 90%が TS1～TS10 命令（つまり，最も多くデータ・キャッシュ・ミスを発生した上位 10 個のロード/ストア命令）によって占められる．実際には，L1 キャッシュでミスしたロード/ストア命令は 146 個存在していた．そのうち，全キャッシュ・ミスに占める割合は，TS1 命令が 23%，TS2 命令が 22%，TS3 命令が 21%であり，これら上位 3 個の TS 命令で全ミスの 66%を占めていることが分かった．この結果より，129.compress では TS 命令が存在すると考える．また，L2 データ・キャッシュに関しても L1 キャッシュと同様に TS 命令の存在を確認できる．これに対し，126.gcc では，上位 10 個の TS 命令が占める割合は L1 キャッシュで 15%程度，L2 キャッシュでは 25%程度と比較的低い．このベンチマークでは，実行中に L1 キャッシュ・ミスを発生したロード/ストア命令が 8403 個存在した．つまり，これはミスを引起す命令の局所性が低い事を意味する．

2.3 TS 命令の生存期間

第 2.2 節の実験結果より，多くのプログラムでは TS 命令が存在する事が分かった．しかしながら，TS 命令の生存期間が短い場合には，動的最適化の適用可能時間が短くなる．そこで，129.compress における TS 命令生存期間を調査した．なお，他のベンチマークに関しても同様の結果を得ている．実験結果を図 2 に示す．ここで，縦軸は TS1～TS10 命令を，横軸は実行サイクル数（1 万サイクルを 1 とする）を表している．図より，TS 命令は比較的長期間に渡って実行されている（ミスを引起している）事が分かる．これより，TS 命令を検出して専用最適化回路を実現した場合でも，その恩恵を受けるだけの十分長い期間 TS 命令は生存し続けることが期待できる．

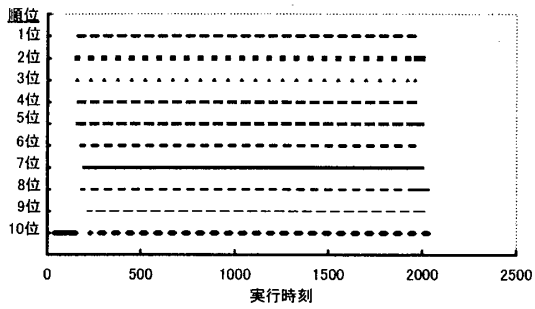

図 2:129.compress での TS 命令生存期間

2.4 TS 命令のアクセス・パタン

129.compress に関して，L1 キャッシュに関する TS 命令のメモリ・アクセス分布を図 3~5 に示す．ここで，縦軸は各 TS 命令がアクセスしたアドレスを，横軸は実行サイクル数（1 万サイクルを 1 とする）を表している．図 3 は 129.compress を実行した際の TS 1 命令におけるアクセス分布である．この図より，アクセスするアドレス範囲は非常に広いが，アクセスするアドレス幅がある程度一定間隔である．ここでは，このようなアクセス・パタンを均一分散型と呼ぶ．また，図 4 は TS2 命令に関するアクセス分布であるが，図 3 とは異なり，比較的広域に渡ってアクセスを行っている．このようなアクセス・パタンを広域分散型と呼ぶ．最後に，図 5 は TS4 命令のアクセス分布であるが，この場合は極めて局所的にアクセスが集中している．これを局所集中型と呼ぶ．

3 考察

第 2.2 節で示したように，多くのプログラム（図 1 において 099.go と 126.gcc を除く全てのベンチマーク）には TS 命令が存在する．また，第 2.3 節で示したように，その生存期間は比較的長い．したがって，上位 10 個の TS 命令に関するキャッシュ・ミスの回避，または，ミス・ペナルティの隠蔽を行うだけ，大きな性能向上を達成できると考える．また，第 2.4 節より，TS 命令の種類によってアクセス・パタンが大きく異なる事が分かった．よって，固定的な最適化手法ではなく，各 TS 命令の特徴に応じた手段を用いる方が有効であると考える．例えば，局所集中型 TS 命令に関しては小容量の TS 命令専用キャッシュを，広域分散型に関しては比較的大容量の専用キャッシュを搭載する等が考えられる．また，均一分散型に関しては，ストライド・プリフェッチなどが有効である．

4 おわりに

本稿では，TS 命令の特徴解析を行い，キャッシュ・ミス頻発命令に着目したメモリ高性能化技術の有効性を考察した．その結果，多くのプログラムには TS 命令が存在し，その生存期間も長い事が判明した．また，各 TS 命令は異なるアクセス・パタンを有するため，各 TS 命令専用最適化回路を動的に決定・適用する事が有効であると分かった．今後，TS 命令の動的検出ならびに最適化回路生成手法を確立し，

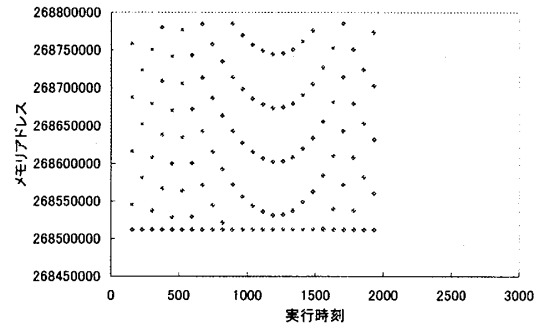

図 3： 均一分散型

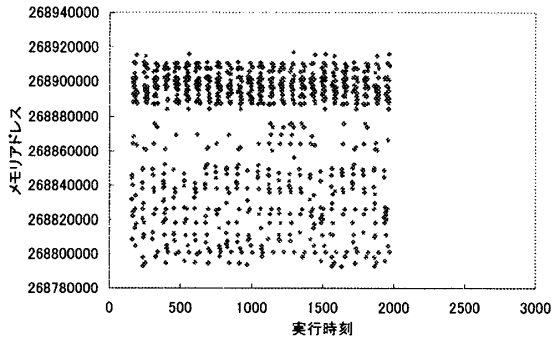

図 4： 広域分散型

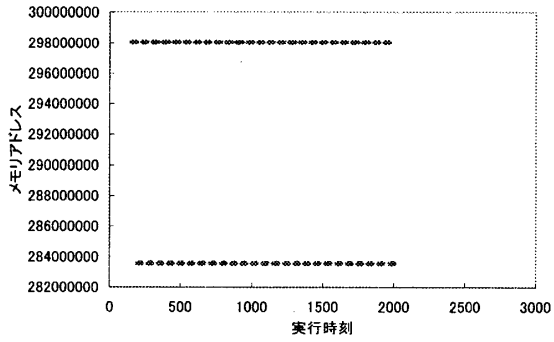

図 5： 局所集中型

実際のプログラム実行における性能向上率を評価する予定である．

謝辞

多くの有用なアドバイスを頂いた福岡大学モシニャガ・ワシリー教授に感謝します．なお，本研究は一部，科学研究費補助金(課題番号：14GS0218, 14702064）による．

参考文献

[1] D.Burger and T.M.Austin,"Univ. of Wisconsin-Madison Computer Sciences Department Technical Report #1342, June, 1997.

[2] SimpleScalar Tool Stes,http://www.simplescalar.com/.

[3] SPEC(Standard Performance Evaluation Corporation) http://www.specbench.org/.