データ依存命令を対象としたデータ値予測

仲沢由香里　　山名早人
早稲田大学 理工学部 情報学科

1．はじめに

プロセッサの性能向上を妨げている要因の一つに，命令間に存在するデータ依存がある．データ依存は，プロセッサのパイプライン処理においてストールを引き起こす原因の一つである．近年，このデータ依存を解決するための投機実行として，データ値予測が考案されている．これは，生成されるデータの値を予測し,その予測値を用いて投機的に処理を進めていくことで,プロセッサ性能を向上させる技術である．

値予測は，予測した結果が正しければ速度向上が得られるが，誤っていた場合にはその結果を使用した全ての命令が生成した値を破棄し，再実行しなければならない．そのため，性能向上を得るには，できる限り多くの命令に対して正しく値予測をする必要がある．しかし，データ値予測は全レジスタ値生成命令が対象となるため，値履歴テーブルのエントリ数が多く，競合が発生しやすい．予測の対象となる命令の中には，予測する意味のない命令，つまりストールの原因とならない命令も含まれる．それらの命令を予測しないことで，値履歴テーブルの競合を減らし予測正解率を上げることができると考えられる．そこで本稿では，全命令に対して値予測を行う場合と，ストールの原因となる命令に対してのみ値予測を行う場合の予測正解率とミス率を比較した．

2．実験環境

本稿では，SimpleScalar tool set Version 3.0c/PISA[1]のsim-outorderを使用した．実験に用いたベンチマークはSPECint95の124.m88ksim,126.gcc,129.compress,130.li,132.ijpegで，プログラムはgcc2.7.2.3によって-O2 -funroll-loopsでコンパイルされている．入力セットはtestセットを用いる．値予測機構には，最終値予測機構[2]とストライド値予測機構[3]を用いる．各予測機構に対して,全レジスタ値生成命令をエントリ対象とする場合とストールの原因となる命令のみをエントリ対象とする場合を比較する．実験に用いた容量は1K(ストライド値予測機構のみ),2K,4K,8K(最終値予測機構のみ),無限エントリの各4種類である．

Data Value Prediction for data dependenced instructions
Yukari Nakazawa, Hayato Yamana
Department of Information and Computer Science, School of Science and Engineering, Waseda University

3．予測対象命令

実行時に，ストールの原因になると判明した命令を値履歴テーブルにエントリする．各ベンチマークにおいて，全レジスタ値生成命令数とストールの原因となった動的命令数を表1に示す．約60～80%の命令がストールの原因となることがわかる．

表１：ストールの原因となる命令数

	レジスタ値生成命令数	ストールの原因となる命令数	割合(%)
compress	1,363,519	792,107	58.09
gcc	819,511,474	544,909,043	66.49
ijpeg	440,554,936	363,866,912	82.59
li	573,182,562	412,841,981	72.03
m88ksim	294,513,778	207,149,642	70.34

4.実験結果

まず，本稿で用いる予測正解率とミス率という言葉を定義する．ストールの原因となる命令のうち，値予測機構で正しく値を予測できた命令の割合を予測正解率，間違った値を予測した命令の割合をミス率と定義する．一般に値予測機構には，予測の成功/失敗で増減する飽和カウンタ[1]を用意し，その値が閾値[2]以上の時だけ予測を行う．そのため，予測を行わない命令が存在し，予測正解率+予測ミス率が実際に値予測を行った割合となる．

表2,3に各データ値予測機構に対して①全レジスタ値生成命令をエントリする場合と，②ストールの原因となる命令のみエントリする場合の予測正解率を示す．各ベンチマークの上の行が①の予測正解率,下の行が②の予測正解率である．

最終値予測機構でもストライド予測機構でも容量が小さいほど①と②の予測正解率の差が大きいことがわかる．これは，容量が小さい予測機構ほど，予測対象命令の削減による競合緩和の効果が大きいためと考えられる．

また，エントリ数が無限の時に②の方がわずかに予測正解率が低いのは,②では命令がストールの原因となった時に初めて値履歴テーブルにエントリされるため，予測ができないことによる．例えば,ある静的命令が計10回実行され，3回目に初めてストールの原因となる場合,3回目に実行された時に値履

1,2 本稿では，飽和カウンタ＝2bit，閾値＝2 と設定

歴テーブルにエントリされるため，3回目の時は予測ができない．しかし，容量の小さい予測機構では，上記の問題によるデメリットよりも予測対象命令数削減による競合緩和のメリットの方が大きいことがわかる．

表2: 予測正解率(最終値予測機構)(%)

	対象	エントリ数			
		2K	4K	8K	無限
compress	①	32.28	36.38	39.22	41.20
	②	34.23	37.86	40.13	41.17
gcc	①	21.35	26.96	33.15	46.25
	②	23.75	29.90	35.90	46.25
ijpeg	①	16.42	17.05	17.34	18.07
	②	16.67	17.25	17.52	18.07
li	①	20.05	26.70	33.80	34.37
	②	21.87	28.38	34.04	34.37
m88ksim	①	23.80	42.51	51.97	60.15
	②	32.06	49.38	55.73	60.15

表3: 予測正解率(ストライド予測機構)(%)

	対象	エントリ数			
		1K	2K	4K	無限
compress	①	61.83	64.58	66.14	66.49
	②	63.23	65.41	66.26	66.45
gcc	①	31.83	37.70	42.56	49.56
	②	34.53	40.18	44.29	49.56
ijpeg	①	23.56	23.89	24.31	24.54
	②	23.70	24.00	24.33	24.53
li	①	33.09	40.63	40.67	41.08
	②	35.13	40.83	40.85	41.08
m88ksim	①	43.79	53.42	59.94	61.62
	②	50.65	57.23	60.95	61.62

次にミス率を比較する．表4と表5に各予測機構における5つのベンチマークのミス率平均を示す．予測正解率と同じく上が①のミス率，下が②のミス率である．無限エントリを除いて,①よりも②の方が0.01～0.26%高く，その差は容量が大きいほど小さい．予測対象命令数を削減することで，値履歴テーブルの競合が減少し，実際に値を予測できる命令数が増える．そのため，予測正解率と共にミス率も増加すると予想されたが，ミス率の増加は0.01～0.26%と少ないことがわかった．

表4: ミス率(最終値予測機構)(%)

	対象	エントリ数			
		2K	4K	8K	無限
All	①	2.14	2.76	3.04	3.26
	②	2.40	2.94	3.05	3.26

表5: ミス率(ストライド値予測機構)(%)

	対象	エントリ数			
		1K	2K	4K	無限
All	①	3.03	3.29	3.44	3.47
	②	3.18	3.36	3.45	3.47

5.おわりに

本稿では，最終値予測機構とストライド値予測機構に対して,全レジスタ値生成命令を予測対象とする場合と,実際にストールの原因となる命令のみを予測対象とした場合の予測正解率とミス率の比較を行った．その結果，ストールの原因となる命令のみを値予測の対象とすることで，対象命令を削減しない場合よりも，エントリ数2Kの最終値予測機構で平均0.25～8.26%予測正解率が向上した．この手法は,実行時にストールの原因となる命令を判別し，ストールの原因となった時に初めて値履歴テーブルにエントリするため，初めてエントリされた時に過去の履歴を有効利用できないという問題があるが，値予測機構の容量が小さい時には予測命令数を削減したことによるメリットの方が大きいことがわかった．

また，値予測を適用する命令を選択する手法に，プロファイルを用いて静的に選択する手法[4,5]があるが，それと比較して本手法はあらかじめプロファイルをとる必要がないという利点がある．

参考文献

[1] Doug Burger and Todd M. Austin, "The SimpleScalar Tool Set, Version 2.0", Technical Report CS-TR-97-1342, University of Wisconsin-Madison (1997).

[2] Mikko H. Lipasti and John Paul Shen, "Exceeding the Dataflow Limit via Value Prediction", Proc. of MICRO-29, pp. 226-237 (1996).

[3] Kai Wang and Manoj Franklin, "Highly Accurate Data Value Prediction using Hybrid Predictors", Proc. of MICRO-30, pp. 281-290 (1997).

[4] Freddy Gabbay and Avi Mendelson, "Can Program Profiling Support Value Prediction?", Prof. of MICRO-30, pp. 270-280 (1997).

[5] 飯塚大介，小沢年弘，坂井修一，田中英彦，"プロファイルを用いた値予測命令削減手法"，情処研報，HPC-82-22，pp. 125-136 (2000).