

第 11 回課題レポート

1522068 西原翔

2024 年 12 月 11 日

課題 1

BHT Simulator を使用して、予測精度を測定し、結果を考察せよ。配列データの並び方、BHT エントリ数や履歴サイズ、および予測の初期値を変えた結果はどうなるか？

ランダムに並んだ配列のときの予測精度の結果は表 1 のようになった。

bne の命令アドレスははじめの if による 0x00400024, 最後の while による 0x00400044 であった。これよりエントリ数が 8, 16, 32 のときのインデックスが表 1 のようになるのがわかる。

ヒット率に関して、与えられた配列がランダムであるため if 文での分岐予測がうまくいかず、精度が半分程度になっている。一方 while での分岐ははじめのカウンタの更新の分と最後の分岐以外ではずっと同じ動きをするため、分岐履歴表による予測の精度が高くなっている。エントリ数が 8 であると、これら 2 つの分岐が合わさってしまっているため精度が悪くなっているのがわかる。

表 1: ランダムに並んだ配列のときの結果 (class1.asm)

エントリ数	カウンタのビット数	使われたインデックス番号	ヒット率
8	1	1	34/64
8	2	1	46/64
16	1	1, 9	30/32, 16/32
16	2	1, 9	29/32, 18/32
32	1	9, 17	16/32, 30/32
32	2	9, 17	18/32, 29/32

降順に並んだ配列のときの予測精度の結果は表 2 のようになった。

class2.asm は与えられる配列以外は class1.asm と同じコードであるため、分岐履歴表で使われるインデックスも表 2 のものと同じになった。

ヒット率に関して、与えられた配列が降順であり、はじめの分岐予測と配列の値が 16 をまたぐとき以外は分岐履歴と同じ分岐になるため、エントリ数が 16, 32 では精度がよくなっているのがわかる。while での分岐は class1.asm と同じ条件であるため、精度も同じようになっている。

表 2: ランダムに並んだ配列のときの結果 (class2.asm)

エントリ数	カウンタのビット数	使われたインデックス番号	ヒット率
8	1	1	32/64
8	2	1	46/64
16	1	1, 9	30/32, 31/32
16	2	1, 9	29/32, 30/32
32	1	9, 17	31/32, 30/32
32	2	9, 17	30/32, 29/32

課題 2

Data Cache Simulator を使用して、キャッシュのヒット数やミス数を測定し、結果を考察せよ。配置方式、ブロック数、ブロック・サイズを変えた結果はどうなるか？

結果は表 3, 表 4, 表 5 のようになった。いずれの方式であっても参照する配列がランダムになっているため、キャッシュサイズを十分上げたときのヒット率は同じようになった。ただ、キャッシュサイズに対するヒット率は Direct Mapping、2-Way Associative 方式、Fully Associative 方式の順によくなっているのがわかる。Fully Associative 方式はアドレスのオフセットを除くすべての部分がタグになっているため、利用効率は高くなっている。しかし Direct Mapping 方式はインデックスのみからキャッシュの配置位置を決めるため利用効率が悪く、結果としてヒット率が悪くなっている。2-Way Associative 方式はこれら二つの中間のような性質をもつのでこの順になっている。

このように短い処理であれば Fully Associative 方式でよいように見える。しかし、実際にはキャッシュのなかからデータを探索する必要があるためそのようにはいかない。

表 3: Direct Mapping 方式でのヒット率 (class2.asm)

ブロック数	キャッシュサイズ						
	1	2	4	8	16	32	64
1	0%	0 %	0%	0%	0%	0%	48%
2	0%	13 %	39%	17%	33%	47%	97%
4	0%	36 %	45%	39%	59%	95%	97%
8	0%	39 %	53%	63%	94%	95%	97%
16	0%	41 %	66%	88%	94%	95%	97%
32	0%	50 %	75%	88%	94%	95%	97%

表 4: Fully Associative 方式でのヒット率 (class2.asm)

ブロック数	キャッシュサイズ						
	1	2	4	8	16	32	64
1	0%	0 %	0%	0%	0%	0%	48%
2	0%	34 %	55%	64%	70%	72%	97%
4	0%	44 %	73%	88%	94%	95%	97%
8	0%	50 %	75%	88%	94%	95%	97%
16	0%	50 %	75%	88%	94%	95%	97%
32	0%	50 %	75%	88%	94%	95%	97%

表 5: 2-Way Set Associative 方式でのヒット率 (class2.asm)

ブロック数	キャッシュサイズ						
	1	2	4	8	16	32	64
2	0%	34 %	55%	64%	70%	72%	97%
4	0%	39 %	67%	73%	84%	95%	97%
8	0%	45 %	67%	84%	94%	95%	97%
16	0%	45 %	73%	88%	94%	95%	97%
32	0%	48 %	75%	88%	94%	95%	97%

課題 3

スカラ・プロセッサを用いて、あるプログラムを実行する。以下の設問に答えよ。

- (1) 命令キャッシュのミス率が 2 %, 主記憶へのミス・ペナルティが 300 クロック・サイクル、プロセッサの CPI は 2 である。この命令キャッシュによる実行時間 t_1 と、ミスをしない完全なキャッシュによる実行時間 t_2 の実行時間比 t_1/t_2 を求めよ。
- (2) データ・キャッシュのミス率が 5 %, 主記憶へのアクセス時間が 200 ns、プロセッサの CPI は 2, クロック周波数は 2 GHz である。プログラム中のロード・ストア命令の出現頻度は 30 % である。このデータ・キャッシュによる実行時間 t_1 と、ミスをしない完全なキャッシュによる実行時間 t_2 の実行時間比 t_1/t_2 を求めよ。
- (3) 分岐予測器の予測ミス率が 5 %, 予測ミス・ペナルティが 4 クロック・サイクル、プロセッサの CPI は 1 である。プログラム中の分岐命令の出現頻度は 20 % である。この分岐予測器による実行時間 t_1 と、ミスをしない完全な分岐予測器による実行時間 t_2 の実行時間比 t_1/t_2 を求めよ。

(1)

実行命令数を I , クロックサイクル時間を C としたときに、ミスをしていない完全なキャッシュによる実行時間 t_2 は

$$t_2 = 2 \times I \times C$$

となる。そして命令キャッシュミスがあるときの実行時間 t_1 は t_2 に加え、ミス・ペナルティによる時間が加わるため

$$t_1 = t_2 + 2 \times I \times C \times 2\% \times 300$$

となる。よって実行時間比 t_1/t_2 は

$$\frac{t_1}{t_2} = 4$$

というように 4 倍遅くなる。

(2)

実行命令数を I としたときに、ミスをしていない完全なキャッシュによる実行時間 t_2 は

$$t_2 = \frac{2 \times I}{2 \text{ GHz}} = I \text{ ns}$$

となる。そしてデータ・キャッシュミスがあるときの実行時間 t_1 は t_2 に加え、ミス・ペナルティによる時間が加わるため

$$t_1 = t_2 + I \times 30\% \times 5\% \times 200 \text{ ns} = t_2 + 3I \text{ ns}$$

となる。よって実行時間比 t_1/t_2 は

$$\frac{t_1}{t_2} = 4$$

というように 4 倍遅くなる。

(3)

実行命令数を I , クロックサイクル時間を C としたときに、ミスをしていない完全な分岐予測器による実行時間 t_2 は

$$t_2 = IC$$

となる。そして、分岐予測のミスがあるときの実行時間 t_1 は t_2 に加え、ミス・ペナルティによる時間が加わるため

$$t_1 = t_2 + IC \times 20\% \times 5\% \times 4$$

となる。よって実行時間比 t_1/t_2 は

$$\frac{t_1}{t_2} = 1.04$$

というように 1.04 倍遅くなる。