資　　料

結合ミックスによるCPUの性能評価*

金沢正憲**　北川　一**　萩原　宏***　三輪　修****

Abstract

Instruction mixes, such as Gibson mix, are often used to evaluate the performance of computers, but these mixes are not sufficient for the computers with capability of concurrent operations in CPU and the use of buffer memory. We here propose the concatenate instruction mix which refers to the transition probabilities of instructions while executing scientific programs. We have evaluated the FACOM 230 Model 60 CPU and Model 75 CPU by means of the test program which was generated on the basis of the concatenate instruction mix. The relation between NFP (Not-Found-Probability) in the buffer memory and the average execution time per instruction is also shown.

1. まえがき

CPUの性能評価を行なう場合，Gibson mixなどの「ミックス」がよく取上げられている．この方法はいくつかの短所を持っているにもかかわらず，CPUの性能を1つの数値で表わせるがために，現在も比較的よく使用されている．ところが，最近の計算機のCPUには，先行制御の機能やバッファ・メモリ(Cacheメモリ)を持っているものがあり，これらの場合，一つ一つの命令の実行時間を排他的に決定することは難題である．この問題の解決へのアプローチとして，命令の出現順序について着目し，「結合ミックス (concatenate instruction mix)」なるものを考えた．「結合ミックス」の命令の遷移状況を示す値に基づく命令の系列（テスト・プログラム）を作成する．これはプログラムとしての意味はないが，現実のプログラムに近い命令順序をもっている．このテスト・プログラムを実際に計算機で走らせ，所要時間を測定し平均命令実行時間を算出して，この値によってCPUの性能の比較評価を行なおうとするものである．

2. 結合ミックス

「結合ミックス」とは，「ミックス」を拡張し，命令の出現確率だけではなくて，ある命令の次にどの命令が出現するかという遷移状況も示すものである．即ち，命令 I_j が実行された後に，命令 I_i が出現する条件付き確率 $P(I_i/I_j)$ を考えた．われわれは，種々のプログラムをトレースし*，**Table 1** に示すような「結合ミックス」を得た．**Table 1** には，このトレースによる従来のミックス（京大ミックス）とGibson mixも示してある．この場合の個々の命令のクラス分けは，Gibson mixのそれにほぼ従っている．(**Table 2** を参照.)

「結合ミックス」の荷重は，プログラムの動特性を比較的よく示していると思われる．例えば，クラス9(固定小数の除算)のあとにクラス14(その他)が多いのは，商と余りのレジスタ間の入換え命令が該当しているし，クラス8（固定小数の乗算）のあとにクラス10（シフト）が多いのは，乗算された結果が一語に収まるかどうかを調べるために行なうシフト命令が該当

---

* An Evaluation of CPU Performance by means of the Concatenate Instruction Mix, by Masanori Kanazawa, Hajime Kitagawa (Data Processing Center, Kyoto University), Hiroshi Hagiwara (Faculty of Engineering, Kyoto University) and Osamu Miwa (Computer Engineering Dept., Fujitsu Co., Ltd.)

** 京都大学大型計算機センター
*** 京都大学工学部
**** 富士通(株)電算機技術部

* 数値計算を主体としたFORTRANジョブをトレースした．トレースされた総ステップ数は，コンパイル時と実行時を合わせて約680万ステップである．

Table 1 Kyoto University concatenate instruction mix (%)

I_j \ I_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	京大ミックス	Gibson Mix
1	50.97	7.14	5.22	6.12	2.07	3.06	0.95	2.94	0.08	0.86	0.08	0.00	17.59	2.91	47.74%	31.2%
2	76.98	0.33	2.25	16.92	0.0	0.0	0.0	0.56	0.0	0.87	0.05	0.0	2.03	0.0	5.22	6.1
3	34.01	1.45	10.89	38.39	0.0	0.01	0.00	0.0	0.0	0.83	7.90	0.14	6.33	0.05	3.98	3.8
4	53.10	0.76	4.37	14.00	1.13	0.14	0.0	0.02	0.03	0.45	0.06	0.37	25.40	0.16	13.05	16.6
5	69.00	1.68	0.04	11.48	14.52	0.82	2.12	0.0	0.0	0.0	0.0	0.0	0.34	0.0	1.82	6.9
6	71.88	0.0	0.03	0.25	24.51	3.07	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.0	1.55	3.8
7	92.71	0.0	0.07	0.30	6.62	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.30	0.0	0.49	1.5
8	6.23	2.52	0.0	0.00	0.0	0.0	0.0	0.50	0.0	89.91	0.0	0.0	0.0	0.84	1.46	0.6
9	49.47	9.57	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.49	0.0	0.0	0.30	30.17	0.08	0.2
10	18.65	67.27	0.35	1.53	0.0	0.0	0.0	0.94	1.59	5.65	0.01	0.0	3.98	0.02	2.03	4.4
11	87.02	0.15	0.22	6.79	0.0	1.07	0.0	0.0	2.00	0.76	1.08	0.0	0.91	0.0	0.37	1.6
12	36.02	0.0	0.06	43.76	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.07	12.09	0.0	0.07	5.3
13	29.35	0.75	1.79	26.75	0.03	0.04	0.01	0.00	0.0	0.19	0.0	0.05	40.98	0.15	20.65	18.0
14	91.81	2.95	0.27	4.05	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.00	0.91	0.0	1.48	——

(注) 0.0 は真に 0 であり，0.00 は 0.005 以下を意味する．この区別を付けるため下 2 桁まで求めている．

Table 2 Classes of instructions

クラス	タ イ プ
1	置数格納（インデックス・レジスタも含む）
2	固定小数）加減算
3	比 較
4	分 岐
5	浮動小数の加減算
6	浮動小数の乗算
7	浮動小数の除算
8	固定小数の乗算
9	固定小数の除算
10	シフト
11	論理演算
12	レジスタを使用しないもの
13	インデックス関係（置数と格納は除く）
14	その他

している．このように，「結合ミックス」は，命令の出現順序を考慮しているので，順序によって実行時間が変ってくるような先行制御の機能を有する CPU の性能評価に対して，有効な方法の一つであると考えられる．

3. 結合ミックスを用いたCPUの評価の一例

われわれは，**Table 1** で求めた「結合ミックス」に基いて，平均命令時間を測定するためのテスト・プログラムを作成し，FACOM 230 モデル 60 とモデル 75* の CPU 性能比較テストを行なった．

テスト・プログラムを作るために，**Table 1** をさらに細分（オペランドのフェッチ回数による分類など）するとともに，条件付分岐などでは条件の成立・不成立に関するデータも収集した．このようなデータに基いて，乱数を使用することにより，テスト・プログラムを作成した．そして，このプログラムの走行時間を実行命令ステップ数で割ることにより，平均命令実行時間を求めた．

また，モデル 75 はバッファ・メモリを持っているので，NFP (Not Found Probability：プロセッサに必要な情報が，バッファ・メモリ上に見つからない確率）を考慮に入れるため，テスト・プログラムに特殊な命令(Forget All 命令：バッファ・メモリ上の情報をすべて無効にする）を付加え，NFP と平均命令実行時間との関係を測定した．これらの結果を **Fig. 1** に示す．**Fig. 1**(a) から，NFP の値が小さいとき先行制御が有効に働くのに対して，NFP の値が大きく

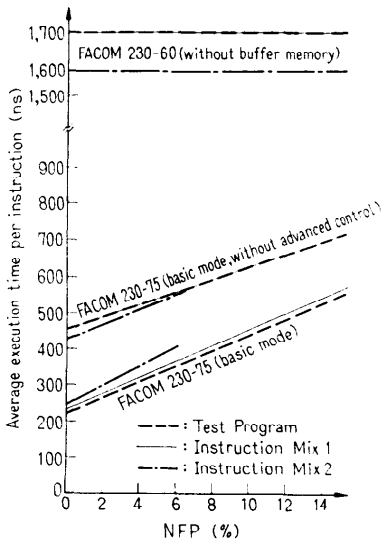

Fig. 1(a) Averge execution time per instruction by means of the test program

* モデル 75 は，モデル 60 に対して，上向きの互換性があり，語構成などは同一である．

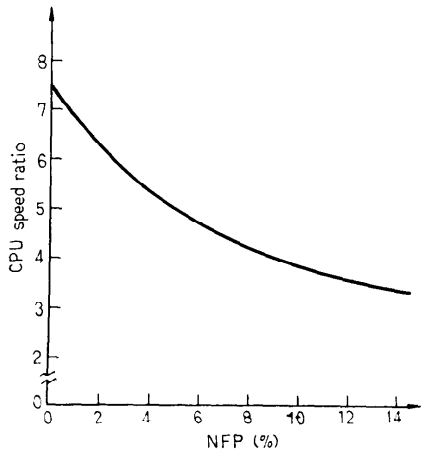

Fig. 1(b) CPU Speed Ratio of Model 75 to Model 60 by means of the test programs

なるとともに先行制御の効果の減少が見られる．「なお，図中には，「結合ミックス」によるテスト・プログラムの結果の他に，「京大ミックス」による命令の出現確率だけから作成したプログラムによる結果*(Instruction Mix 1) と，「京大ミックス」の各クラスに属する代表的な命令を荷重に対応して並べたプログラムによる結果 (Instruction Mix 2) も示した．出現確率だけから作成したプログラムでは，起こりえないような命令出現の遷移が生じるのに対して，「結合ミックス」によるテスト・プログラムでは，命令出現の遷移が考慮されているため，現実的なプログラムに近くなっており，その結果，より実際的な評価になっていると考えられる．

一方，その後，制御プログラムのもとでジョブ処理(44ジョブ，多重度6のマルチプログラミング処理)を行ない，モデル75の CPU(バッファ・メモリ4k語，マシン・サイクル 90 ns)の速さと，モデル60の CPU の速さをジョブ・ステップ単位で比較したのが，**Fig. 2** である．また，ジョブ・ステップ毎の CPU 時間の和をとって比較すれば，約6.5倍である．即ち，実行時間の長いジョブの方が，倍数が大きい．また，**Fig. 1** (b) と **Fig. 2** から，マルチプログラミング処理のもとでの NFP はおおよそ 2～4% でほとんど5% 以下であると考えられる．このジョブ処理のために使用したプログラム (制御プログラム，処理プログラム) は，

* 命令出現確率は，テスト・プログラムと同じであるが，遷移の状況が異なるために，先行制御機能の先行度が異なり，異なる結果が現われている．

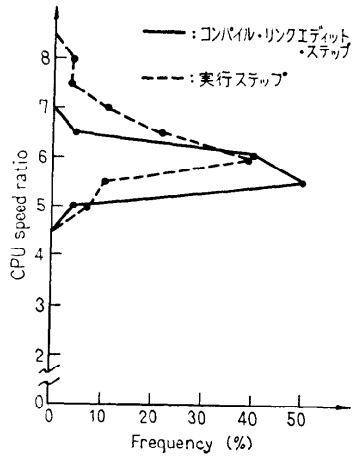

Fig. 2 Distribution of CPU Speed Ratio of Model 75 to Model 60 by means of FORTRAN programs

モデル60用のものであるために，モデル75の拡張命令*を使用していない．従って，拡張命令を使用すれば，さらに速度の比は増大する．

4. 終りに

CPU の性能比較の方法として，「ミックス」を拡張した「結合ミックス」を提案するとともに，それを用いた評価の一例を示した．従来，「ミックス」による評価においては，各クラスに属する命令の実行時間に荷重をかけて平均命令実行時間を算出することが行なわれているが，その方法によって，モデル60の平均命令実行時間を「京大ミックス」から算出すると，1.61 μs となる (ここでのテストプログラムによる測定では，1.68 μs)．一方，モデル75では，先行制御やバッファ・メモリの NFP による影響から一つ一つの命令実行時間を排他的および静的に決められないので，このような平均命令実行時間の算出法は実際的でないと考えられる．そこで，上に述べたようなテスト・プログラムによる測定法を採用した．

なお，実際の計算機の使用上から見れば，命令単位のミックスではなくて，問題向き言語レベルのステートメント単位のミックスとして「ステートメント・ミックス」とでもいうべきもの[2]を考えることも，CPU の性能評価に利用できると思われる．

* 例えば，サブルーチンに現われるようなインデックス・レジスタの退避や復旧を，1つの命令(Save Index 命令，Restore Index 命令) で行なうことができる．

参 考 文 献

1) 高橋義造：コンピュータ評価のための各種ミックス，情報処理，Vol. 13, No. 11, 1972, pp. 777—781.

2) Knuth, D. E.: An Empirical Study of FORTRAN Programs, Software-Practice and Experience, Vol. 1, 1971, pp. 105—133.

(昭和48年4月23日 受 付)
(昭和48年7月11日 再受付)