

# 般事務処理用システムの評価について

石上孝雄 宍戸建夫 山口幸男 若村泰代  
(富士通株式会社 システム部)

## 1. ま え が き

計算機システムの性能評価については、これまでに何種類もの方法が発表されている。<sup>(1)</sup>しかし、それぞれの方法に一長一短があるため、評価を行う時の目的、使用上の容易性、コスト等から評価方法を選択したり、または数種類の方法を組合せて、使用しているのが普通である。

Fmix (FACOM-mix) と呼ぶ性能評価方法は、次のような目的と特長を持っている。

- (1) インストラクション・ミックスのように簡単に算出できるものであること。
- (2) テスト・プログラムを実行したように、ハードウェアとソフトウェアを含めた総合評価、すなわち、実際の処理形態に近い形で評価できるものであること。
- (3) 数種類のデータを基礎値として、あとは基礎値の組合せで算出する。その基礎値は、システム完成前は、サンプルコーディングにより、また、システム完成後は実測により求められるものであること。

現在、処理形態を3種類に分類し、次のFmixを設定している。Fmix-B (Business ; 事務処理用)、Fmix-S (Scientific ; 科学技術計算用)、Fmix-R (Real time ; オンラインリアルタイム用)である。

ここではFmix-Bについて報告する。

一般に事務処理では、CPUの命令速度の他に、入出力装置(I/O)の性能と使い方が、計算機システムとしての性能に、大きく影響する。また、最近ではマルチプログラム処理をするのが普通である。Fmix-Bでは、これらの事務処理の特徴を考慮し、次の順序で設定を行なった。

- (1) 一般的な事務処理プログラムを調査し、統計的に分析する。
- (2) 事務処理用モデルジョブを設定する
- (3) 基礎値を設定し、シングル処理の時の性能値を算出するルールを決定する。
- (4) 入出力装置の処理の補正值、多重処理(マルチプログラム)を行う時の補正值を決定する
- (5) 全体の算出ルールを整理する。

現在、Fmix-Bにより、事務処理を行う際の計算機システムの性能評価を行なっているが、主に、相対比較、システムの改善案のピックアップ等に有効である。

## 2. 事務処理プログラム

### 2.1 処理パターン

一般に事務処理といっても、その内容は給与計算、在庫管理、販売管理など多くの業務に分けられる。しかし、その処理パターンを分析してみると、トランザクションデータを入力するときのコンバート処理、チェックやマスターファイルの更新などのメンテナンス処理と SORT 処理が繰り返されて、最終的なアウトプットを得ている場合が大部分である。結局、事務計算の処理パターンは、業務の種類とは特に関係せず、インプット処理、SORT 処理、メンテナンス処理、リスト処理に集約することができる(図1)

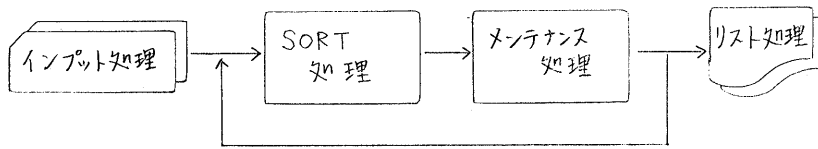


図1. 事務計算の処理パターン

この処理パターンを事務処理のモデルと考えると、このモデルでの仕事量は次のように考えることができる。

CPU の使用量に着目すると、3つの要素、すなわち、処理プログラムの処理手続部分 (COBOL の PROCEDURE DIVISION)、SORT 処理の部分、制御プログラムの走行部分に分けられる。

更に、事務処理を前提にすれば、次のように置きかえて考えることができる。

- (1) プログラムの処理手続部分  
→ COBOL のダイナミックな走行ステップ数
- (2) 制御プログラムの走行部分  
→ ダイナミックなファイル入出力の回数
- (3) SORT 部分  
→ SORT 件数

すなわち、事務処理に於ける CPU の使用量を、1件の入力 (トランザクションデータ) に対する、

COBOL のダイナミックな走行ステップ数  
ダイナミックなファイル入出力の回数  
SORT 件数

を求めて、それに代表させようとするものである。

### 2.2 調査と分析

代表的な中型機ユーザーを選定し、実際使用している代表的な事務処理業務から、120本のプログラムを選択し、次の調査を行った。選択されたプログラムは、各ユーザーでの処理時間全体のなかで、上位にあるものである。

- (1) COBOL プログラムのダイナミックな走行ステップ数
- (2) ファイルへのダイナミックなアクセス回数

- (3) 各ファイルのレコード長, ブロック長, ファイル編成, アクセスモード
- (4) リスト出力行数
- (5) SORTのキイ長, SORTの入出力となるファイルのレコード長, ブロック長
- (6) その他(機器構成, オブジェクトプログラムの大きさなど)

この調査はアンケート調査と, 特にこの目的のために向進した. ダイナミック・トレースとよって行なった. ダイナミック・トレースは各プログラムの実行時に付ける, ファイルへのアクセス情況, 実行スタートメント(COBOL)の詳細な分布情報等入手することができる.

調査結果は, ぼう大なものとなったが統計的に処理した. 例えば COBOLの総実行ステップは  $1.5 \times 10^8$  ステップ, またファイルへのアクセス回数は統計で  $5.7 \times 10^6$  回 などとなっている. また月次処理のプログラム, 日次処理のプログラムがあるが, 補正を行い, すべて月単位の処理量に置きかえた.

調査結果は次の通りである. いずれも1件の入力(トランザクションデータ)に対しての数値である

- COBOL走行ステップ数 1278 ステートメント
- ファイルへのアクセス回数
  - |||編成ファイル
    - Read / Write 40~46 回
  - |||インデックスファイル
    - Read 0.8~1 回
    - Rewrite 0.1 回
- SORT件数 3 件
- 出力(リスト)行数 4 行

結果に中をもたせているのは, ハードウェアの機器構成によって, 次のような補正を可能にするためである.

- (1) メモリサイズが大きくなれば, ジョブステップが統合される. 結果として, 中間ファイルが減少し, 全体のファイルアクセス回数が減少する.
- (2) バッファサイズが大きくなると, ファイルへのアクセス回数が減少する.

### 3. Fmix-B の算出

#### 3.1 Fmix-Bの基礎値

最初、Fmix-Bの算出に必要な基礎値を求める。基礎値は次の3項目である。

- (1) COBOL 1ステートメントの平均実行時間
- (2) ファイルへのアクセス1回当りの実行時間
- (3) SORT 1件当りの実行時間

基礎値の把握は、システム完成前は、サンプルコーディングにより、また、システム完成後は実測により求める。

#### [COBOL 1ステートメントの平均実行時間]

平均実行時間は、従来のインストラクション・ミックスと同様な方法によって求める。COBOL プログラムにおける命令ステートメントごとのダイナミックな出現頻度分布を、先に説明したダイナミック・トレースによって求め、重み付けを下記の通り決定した。

COBOLステートメント		頻度(%)
MOVE	X(2) → X(2)	7.8
"	X(64) → X(64)	10.0
"	S9(4) → S9(4)	8.0
"	S9(4)COM → S9(4)COM	5.5
"	S9(7)COM → -Z, ZZZ, ZZZ	2.1
IF	X(6) X(6)	18.5
"	S9(8) S9(8)	5.5
"	S9(4)COM S9(4)	8.3
ADD	S9(6) + S9(4) → S9(6)	12.0
SUB	S9(6) - S9(5) → S9(6)	0.7
MULT	S9(7) × S9(11)V9 → S9(11)V9	0.8
DIV	S9(7) / S9(5)V9 → S9(7)V9	0.5
GO TO		13.4
PERFORM	A THRU B	2.8
その他		4.1
計		100.0

表1 COBOLステートメント・ミックス

#### [ファイルへのアクセス1回当りの実行時間]

ファイルのレコード長、ブロック化定数等は次の値を使用している。順編成ファイルの場合、レコード長 220 バイト、ブロック化定数 4~12 である。また、順インデックス編成ファイルの場合、レコード長 140 バイト、ブロック化定数 4~10、シンボリックキーの長さ 10 バイトである。

この実行時間を求める方法には、いくつかの工夫が必要である。サンプルコーディングの場合は OS の構造から、標準的な手法を求めてあり、それにあては

めて算出する。また、実測の場合は、誤差を無視できる程度に、処理件数を大きくし、その平均を求める。

### [ SORT 1件当りの実行時間 ]

SORT を行うファイルのレコード長は 100 バイト、ブロック化定数は 10~18, キーの長さ 14 バイト, としている。また SORT を行う主記憶領域を 15~60 KB として実行時間を求める。

サンプルコーディングの場合は、基本的な命令、すなわち MOVE, COMPARE, GET, PUT の 4 つの命令に置きかえて求める。

### 3.2 シングル処理の処理時間

シングル処理の処理時間として、1 件の入力 (トランザクションデータ) に対する処理時間を求めると、次の通りである

シングル処理の処理時間 =

$$(T_{\text{SORT}} \times 3) + \text{MAX} \left\{ T_{\text{COBOL}} \times 1278 + T_{\text{AM1}} \times a + T_{\text{AM2}} \times b + T_{\text{AM3}} \times 0.1, \right. \\ \left. \text{I/O 時間} \times \text{I/O 時間補正係数} \right\}$$

$T_{\text{SORT}}$  ; SORT 1 件当りの実行時間

$T_{\text{COBOL}}$  ; COBOL ステートメントの平均実行時間

$T_{\text{AM1}}$  ; 順編成ファイル Read/Write 時間

$T_{\text{AM2}}$  ; 順インデックスファイル Read 時間

$T_{\text{AM3}}$  ; " Rewrite 時間

$a$  ; 40~46

$b$  ; 0.8~1

上記  $a, b$  で、いずれの値を採用するかは、機器構成の差、すなわち、大型、中型、小型のいずれの計算機システムであるかによって決定する。

I/O 時間を求めて、CPU の処理時間を求めた値と比較し、その大きい方を処理時間とする。I/O 補正係数は、標準として接続される I/O の種類及び台数を考慮して、理論的に決定する。

### 3.3 Fmix-B の値

Fmix-B の基本的な考え方は、一般の事務処理 (バッチ処理) を実行したときの計算機システムの能力、特に CPU 能力を求めることである。

前項の「シングル処理の処理時間」のみでは、複数種類の計算機システムに對して、1 件の入力についてどれだけの処理時間がかかるか、相対比較をすることができても、CPU の能力を十分に、使いきっているか否かは不明である。

そのため、CPU 能力の最大値を求めるため、次の補正を行う。

(1) シングル処理に於ける処理時間 (プロセス時間) と CPU 時間を求めて、

CPU 占有率を算出する

(2) CPU に余裕がある場合、多重処理を行うことができるが、いくつかの

制限がある。

- (3) 主要な制限は、OSの制限、メモリ容量の制限、入出力装置(I/O)の制限である。
- (4) この制限内で多重処理を行う場合は、整合関係が生ずるため、特合セ理論による補正を行う。

### 3.4 Fmix-B の事例

中型機～大型機である3機種(モデルA, B, C)のFmix-Bの結果は次の通りである。

項目		A	B	C		
基礎 値	COBOL 1スタートメント (ms)	37	22	8		
	SORT 実行時間	CPU時間 (ms)	2.12	1.36	0.82	
		プロセス時間 (ms)	6.58	5.25	4.15	
	ファイル アクセス 実行時間 (ms)	順編成	Read/Write	1.34	0.72	0.46
			Read	18.22	9.76	5.43
		編成	Rewrite	6.65	3.75	2.10
	シングル処理 時間 (ms)	CPU時間	134.2	69.2	35.6	
プロセス時間		198.7	129.1	85.3		
CPU占有率(%)		67.5	53.6	41.7		
最大多重度(多重度決定要因)		2 (CPU)	2 (CPU)	2 (I/O)		
多重効率		1.10	1.25	1.21		
Fmix-B		180.6	103.2	70.5		
能力比(モデルBを1として)		0.57	1	1.46		

表2 Fmix-B の例

Fmix-Bは計算機システムを事務処理用に使用した場合の、能力の相対比較に使用することができる。更に、各項目の比較をすることによって、ハードウェア、ソフトウェアの問題点をピックアップすることができる。

現在、各種補正係数の検定、単純化、新アーキテクチャ(バッファメモリ、マルチCPU、VS等)の導入等を検討している。

#### 4. 参考文献

- (1) システムの性能評価法特集号, 情報処理, Vol. 13, No 11 (1972)
- (2) 山本欣子; コンピュータ・システムの評価(1), (2),  
bit Vol. 5, No 11, 12 (1973)