SE 14-2
(1976. 3. 26)

科学技術用計算機システムの性能評価法

斉藤梅朗　（電気通信大学）

1 はじめに

計算機システムの性能は、初期の段階においては専らハードウェアの処理能力によって左右されておりその評価はハードウェアの速さ・信頼性・保守性等によっておこなわれていたが、現在のように多面的要素をもった計算機システムではソフトウェア：特にオペレーティングシステムとコンパイラが生成する目的プログラム：の性能を無視して評価をおこなうことはできない。

ひとくちに性能といっても量的・質的な面があり計算機システムを構成する多面的要素の中から適当な評価項目を定めて評価をおこなう必要がある。したがってその着目する評価項目によって種々の評価方法が提案[1]されている。

計算機システムのCPU速度の評価はその絶対量として Instruction mix という形であらわし、その代表的なものとして Gibson mix がよく知られている。比較的処理時間の長いジョブについては計算機システム間の速度比は Instruction mix の比に近い値になる。処理時間の短かいジョブではオペレーティングシステムの性能に大きく依存する。また Instruction mix では計算機固有の命令・アドレス方式・ワードサイズ等の特徴が生かされず、現在のような高度で複雑なアーキテクチャのもとでは Instruction mix が定められた背景から考えても充分な評価方法とはいえない[2]。

本稿ではジョブの処理過程をモデル化し科学技術計算の分野における計算機システムのCPU速度の相対評価量をジョブ処理時間の関数として導びき、現用の計算機システムの測定結果をもとにモデルの妥当性・有用性について検討した。

2 計算機システムのCPU速度比

一般に計算機システム間のCPU速度比のジョブ処理時間に対する依存性はジョブ処理過程を翻訳・結合・実行の3段階から成るとすれば Fig. 1 のようになる。$T_e \sim T_1$ の領域は Empty job[3] と処理時間の短かいジョブのもので、CPU時間のほとんどはオペレーティングシステムによるオーバーヘッドとして消費される。$T_1 \sim T_2$ の領域はCPU時間は実行のために消費されるが $T_e \sim T_1$ の領域のオーバーヘッドを回復するための過渡領域である。$T_2 \sim \infty$ の領域はハードウェアの速度・生成された目的プログラムの効率が反映される定常領域である。T_1, T_2 はオペレーティングシステムの構成、ハードウェアの設計思想によって著しく変化する。

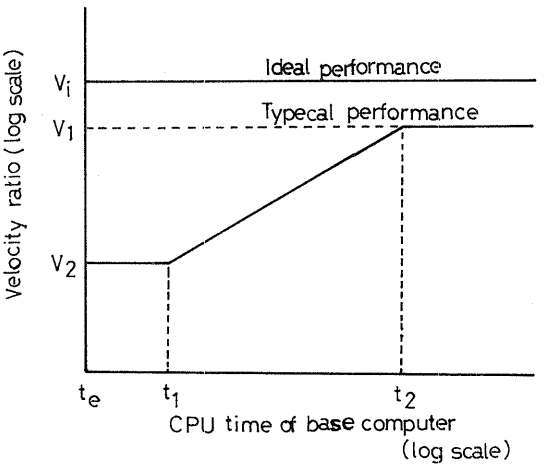

Fig.1 Time chart of velocity ratio

このようにCPU速度が一定とならない原因としては、

(1) 翻訳・結合と実行とでは命令の種類がちがう。

(2) CPU時間にはオペレーティングシステムのオーバーヘッドによる定数項があるためにCPU時間の少ないジョブは遅くなる。

(3) 科学技術演算命令は計算機システムが大規模になるほどほかの命令より速く設計されている。

が考えられる。

いま理想的な計算機システムを考え、ハードウェアの性能が100%ソフトウェアに反映されたとすれば、CPU速度比は一定(V_1)になる。したがって、$(V_1 - V_2)/V_1$ は生成された目的プログラムの効率をあらわし、$V_1 - V_2$ は前述した(1)(2)による速度低下ぶんである。

3. 評価モデル

このモデルであつかう計算機システムのジョブ処理過程は(1)ジョブの開始、(2)翻訳(3)結合(4)実行(5)ジョブの終了の5つのプロセスからなると仮定している。

計算機システムにジョブが投入されるとプロセス(1)から(5)まで順にプロセスの遷移がおこりプロセス内では種々のタスクが発生し消滅する。プロセス内でのタスクの動きをプロセスの時間経過に伴なう状態遷移と考えFig.2のようにジョブ処理過程をモデル化する。

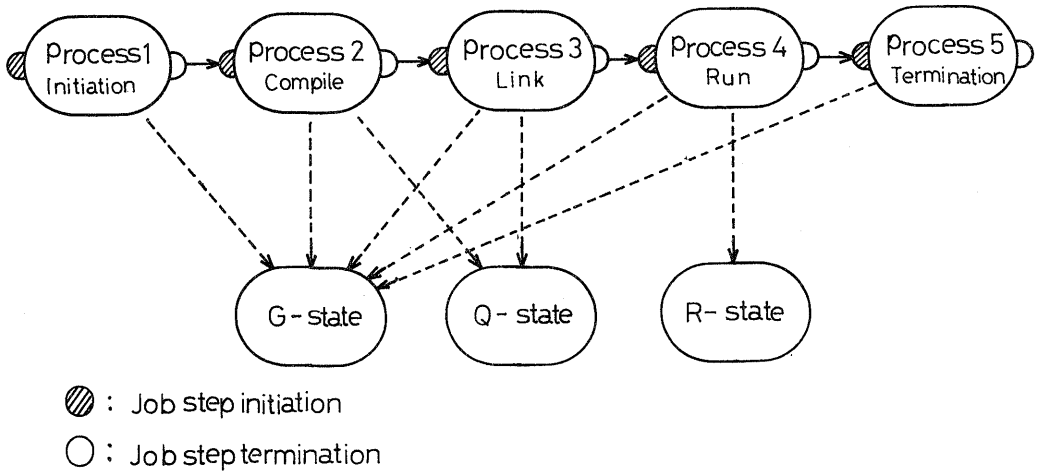

Fig.2 Process transition and the state

このモデルではプロセス内で発生するさまざまなタスクを次の3つの状態に分類し規定する。

(1) 状態G：プロセス(1)~(5)で発生したシステムタスクの集合.

(2) 状態Q：プロセス(2),(3)で発生したユーザタスクの集合.

(3) 状態R：プロセス(4)で発生したユーザタスクの集合.

ただし、ここでいうシステムタスクとはオペレーティングシステムの導入によ

って生じたオーバーヘッドとしてあつかわれる事象であり、ユーザタスクとはそれ以外の事象とする。状態Q・Rはともにユーザタスクであるが、性質の異なった機械命令が実行されているために分離したほうが解析しやすい。

それぞれの状態はその状態に属する事象の集合であるから、それぞれ次のようにあらわされる。

$$G = \{ g_1, g_2, g_3, \cdots\cdots, g_i \} \tag{1}$$

$$Q = \{ q_1, q_2, q_3, \cdots\cdots, q_j \} \tag{2}$$

$$R = \{ r_1, r_2, r_3, \cdots\cdots, r_k \} \tag{3}$$

したがってジョブの処理過程を状態Sとすれば

$$S = G \cup Q \cup R \tag{4}$$

となる。また状態Sが消費した時間はそれぞれの状態に含まれる事象が消費した時間の総和であるから状態Sの消費した時間をTとすれば(1)(2)(3)より

$$T = \int^{i} t_{g_a}\,dt + \int^{j} t_{g_b}\,dt + \int^{k} t_{r_c}\,dt = T_G + T_Q + T_R \tag{5}$$

となる。ただし、T_G, T_Q, T_Rはそれぞれの状態が消費した時間である。

異なる計算機システムの状態Sの消費した時間をそれぞれT_1, T_2とすると計算機システムの速度比 $Vr(t)$は(5)式より次のようになる。

$$Vr(t) = \frac{T_2}{T_1} = \frac{T_{G2}}{T_1} + \frac{T_{Q2}}{T_1} + \frac{T_{R2}}{T_1}$$

$$= G_{V(t)} + Q_{V(t)} + R_{V(t)} \tag{6}$$

上述したように$G_{V(t)}$はオペレーティングシステムによるオーバーヘッドとなる事象の集合であるから、オペレーティングシステムの性能をあらわす項と考える。$G_{V(t)}$の一般的な性質は、大きく複雑な機能をもったオペレーティングシステムほど単位機能あたりの値は大きくなる(Fig. 3参照)。

$Q_{V(t)}$は言語処理プログラム・リンケージエディタの性能に関する項で状態Qの消費する時間が増加するにつれて値は大きくなる(Fig. 4, 5 参照)。

$R_{V(t)}$は一般的にいわれているImstruction mix 比に相当する。

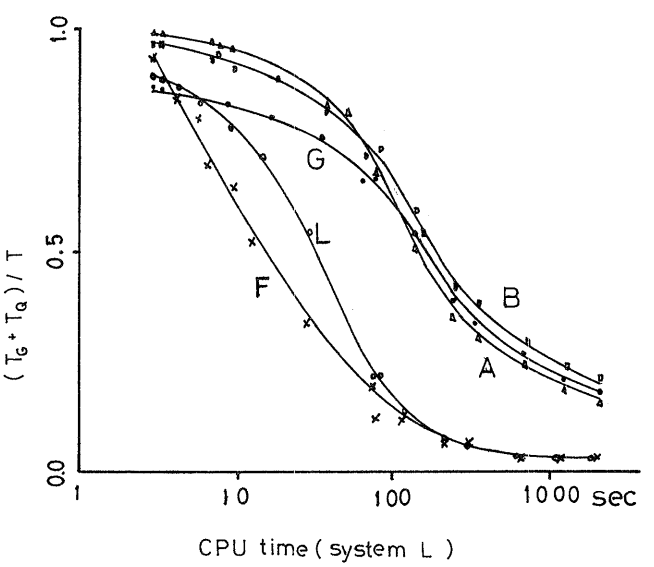

Fig.3 $(T_G + T_Q) / T$

項で計算機システムの科学技術演算命令の平均実行速度比 Iv をあらわす。

次に各項の処理時間の依存性について述べる。

$t = te$（Empty job の消費した時間）のとき $T_Q \simeq 0$，$T_R \simeq 0$ であるから．

$$V_{r(t)} \simeq G_{r(t)} \qquad (7)$$

$t = te + \Delta t = t'$ のとき Δt はすべて状態Qで消費されたとすると $T_R \simeq 0$ であるから．

$$V_{r(t)} \simeq G_{v(t')} + Q_{v(t)} \qquad (8)$$

$t \gg t'$ のときは $R_{v(t)} \gg G_{v(t)}$，$R_{v(t)} \gg Q_{v(t)}$ より

$$V_{r(t)} \simeq R_{v(t)} = const \qquad (9)$$

となる。

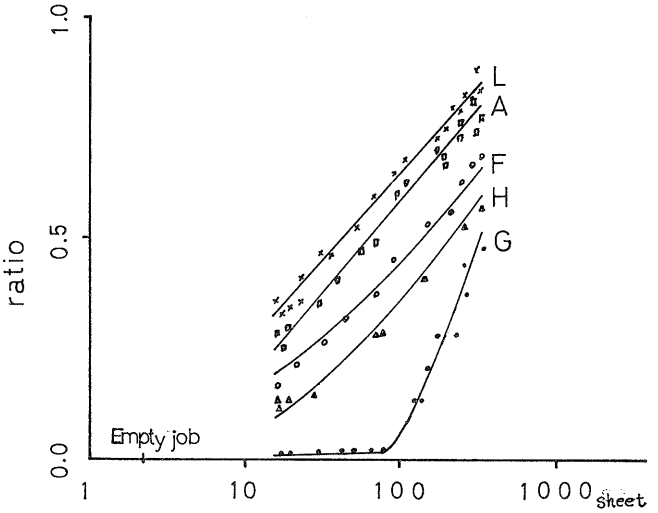

Fig.4 $T_Q / (T_Q + T_G)$

理想的な状態、つまり $S = R$ に対して状態G，Qはオーバーヘッドと考え $V_{r(t)}$ に対する影響について考察してみる。我々が状態G，Qを測定する場合の多くは T_G, T_Q を個別に測定するのではなくそれらが混然とした状態K（$K = G \cup Q$）を測定することしかできない。

いま $t = te$ のとき(7)式より $V_{r(te)} = K_{v(te)} = G_{v(te)}$、$t = t'$ のとき(8)式より $V_{r(t')} = K_{v(t')} = G_{v(t')} + Q_{v(t')}$ とすれば $K_{v(t)} > K_{v(te)}$ であるから $G_{v(te)} \simeq G_{v(t)}$ とみなせば、

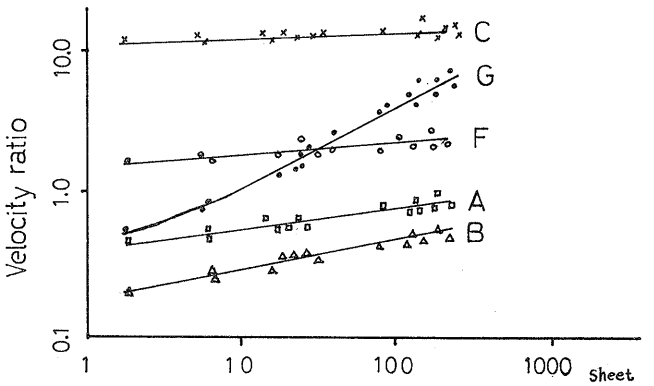

Fig.5 $(G_{V_1(t)} + Q_{V_1(t)}) / (G_{V_2(t)} + Q_{V_2(t)})$

$$Q_{v(t)} \geq K_{v(t)} \geq G_{v(t)} \qquad (10)$$

の関係が得られる。つまり状態Qの存在は T_Q が大きくなるにつれて $te \sim t'$ のあいだはオペレーティングシステムによって低下した速度比を改善する。したがって状態G，Qを状態Rに対するオーバーヘッドとして(10)式の関係から(6)は次のようになる。

$$V_{r(t)} = R_{v(t)} - (G_{v(t)} - Q_{v(t)}) \qquad (11)$$

次に(11)式の各項について解く。$R_{v(t)}$ は(9)式にしめすように $t \gg t'$ では $R_{v(t)} \simeq I_v$ となる。状態G，Qはプロセスの終了とともに消滅し T_G, T_Q は一定になる。したがって T に対する T_G, T_Q の占める割合いの時間変化を γ_G, γ_Q とすると．

$$d\gamma_G / dt = -\alpha \gamma_G \qquad (12)$$

$$d\gamma_Q / dt = -\beta \gamma_Q \qquad (13)$$

となる。これを $t=t_e$, $t=t'$ の境界条件を入れて解き(11)式に代入すると

$$V_{r(t)} = I_v - (K_0 e^{-\alpha t} - K_c e^{-\beta t}) \qquad (14)$$

が得られる。この結果、時間を含んだ計算機システムの性能評価式が得られ、パラメータを個々の計算機システムごとに定めることによって動的な相対評価が可能となる。

4 測定環境について

4.1 テストプログラム

ベンチマークに使用したプログラム群は (1) STOP/END job (2) Student job (3) User's job からなる。(3)のプログラムは電通大データステーションの利用者のジョブから平均的(CPU時間と出力行数の関係で)なものを選び出し、時間軸上に適当な間隔で配置されるようにした。

これらのプログラムはTable.1にしめすような内容のプログラムで約40本、ステートメント数で2～400枚、H8250の処理時間で4.5～3030秒のものである。

精　度	単精度，倍精度，複素単精度
解析手法	掃き出し法，ルンゲクッタ，ヤコビ法 Newton法，最小自乗法，Simpson，Gauss
方程式	連立方程式，微分方程式，ラプラス変換 楕円関数，積分，逐次近似式，固有値 高速フーリエ変換，行列式

Table.1 The character of test program

4.2 計算機システムと測定条件

測定対象とした計算機システムの諸条件をTable.2にしめす。これらのシステムは、(1)経時時計機構 (2) スタックドジョブの処理機能をもっている。

測定形態はシングルジョブストリームが望ましいが、超大型計算機システムはマルチジョブストリームでおこなった。言語処理プログラムの中には、目的プログラムの最適化機能をもったものがいくつかある。特にOS7では利用者が最適化の程度を指定できるが本測定では最適化機能を使用してないために実際の速度[4]に比べて遅くなっている。シングルジョブストリームの場合はテストプログラムをすべて入力キューにスタックしたのちに処理をはじめ、結果は出力キューにスタックした。

出力結果は原始プログラムリスト・リンケージマップ・計算結果である。

HITAC 8800 はマルチプロセッサ構成のシステムである。

System name	OS name	memory size	S&M	FORTRAN compiler name
MELCOM 7500	BPM_1-G00	25KW	S	EX. FORTRAN
MELCOM 7700	BPM_2-F00	25KW	S	EX. FORTRAN
HITAC 8800	OS7-V03	240KB	M	OS7 FORTRAN
HITAC 8700	OS7-V03	200KB	S	OS7 FORTRAN
HITAC 8450	EDOS-V10	102KB	S	FORT131
HITAC 8350	EDOS-V11	100KB	S	FORT131
HITAC 8250	NDOS-V06	92KB	S	NDOS FORTRAN
FACOM 230/45S	OSII/VS-7	50KW	S	FORTRANS
FACOM 230/38	OSII/VS-7	50KW	S	FORTRANS
NEAC 2200/700	MOD4 EX452	120KC	M	FORTRAN700
IBM 360/195	HASP R21G903	256KB	M	FORTRAN IV G LEVEL 21
UNIVAC 1106	EXEC/8-32	64KW	M	FORTRAN V

Table.2 The environment of computer system

5 測定結果の概念的検討

各計算機システムの測定結果をFig.6～15にしめす。計算機システムにはA～Lの略号をつけ測定値は計算機システムLを基準に求めた。Lは語・バイト向きの命令をそなえている。

(1) システムA，Bの検討

AとBは同一のOSで動作する。データは語、半語、バイト単位であつかうことができるが演算時間は語く半語くバイトであるからプロセス2,3のように字単位処理の多いプロセスでは不利である。

(2) システムC～Gの検討

CとD，EとFは同一のOSで動作する。C～GはLと設計思想に共通した部分が多く見られる。CとDは同一のOSで測定したがCはマルチジョブストリームのためオーバーヘッドが少し認められる[4-]。EとFも同一のOSで測定したので両者の差はInstruction mix比に相当する量があった。Gはプログラムの処理時間によって速度比によって大きな差が認められる。この原因としてはマルチジョブストリームによるオーバーヘッドが考えられる。Fig.12を見ると、CPU time 14～23 sec の測定値が他より大きくはずれている。シングルジョブストリームで測定した場合には点線のようになると推定される。

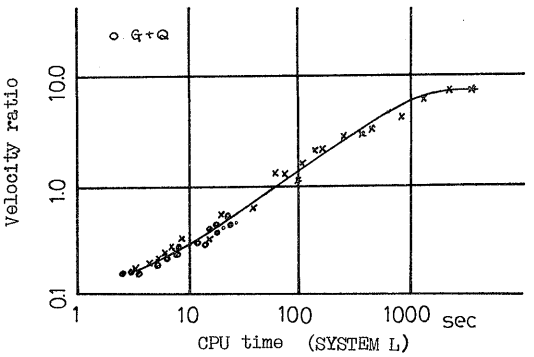

Fig.6 SYSTEM A

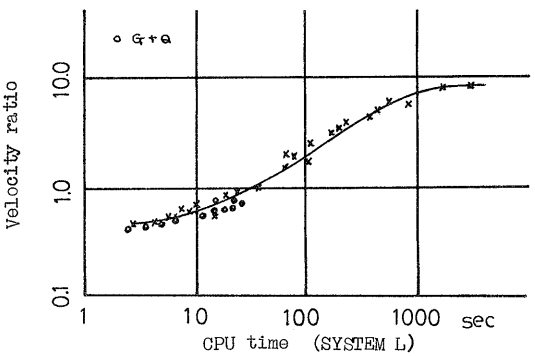

Fig.7 SYSTEM B

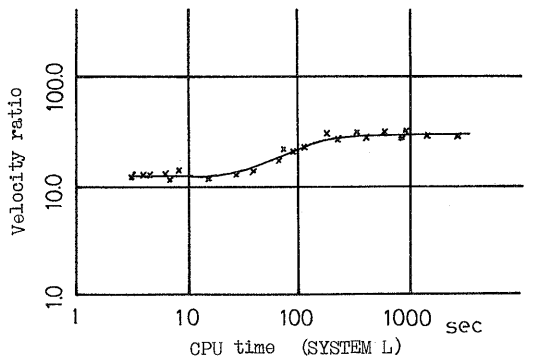

Fig.8 SYSTEM C

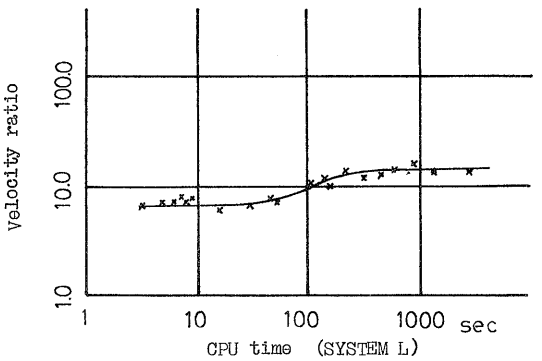

Fig.9 SYSTEM D

(3) システムH，Iの検討

HとIは他の計算機システムとは語長が異なるため解の収束判定を伴なうプログラムでは測定値がばらつく。Hは語処理向きの計算機システムであるからA，Bと同様字処理には何もないのに加えてマルチジョブストリームによるオーバーヘッドがある。Iは字処理向きの計算機システムでGと同様にマルチジョブストリームにオーバーヘッドの影響を受けている。

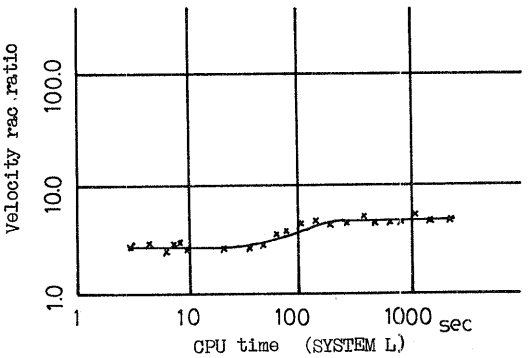

Fig.10 SYSTEM E

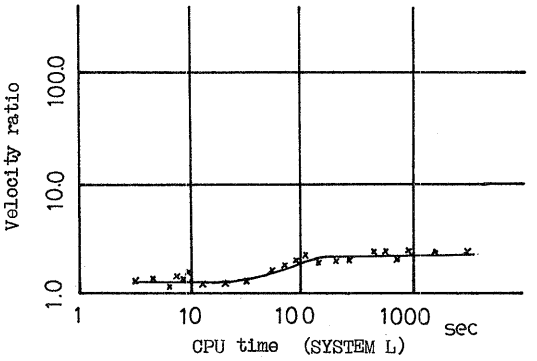

Fig.11 SYSTEM F

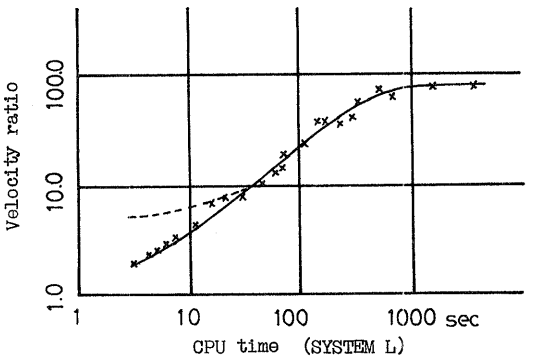

Fig.12 SYSTEM G

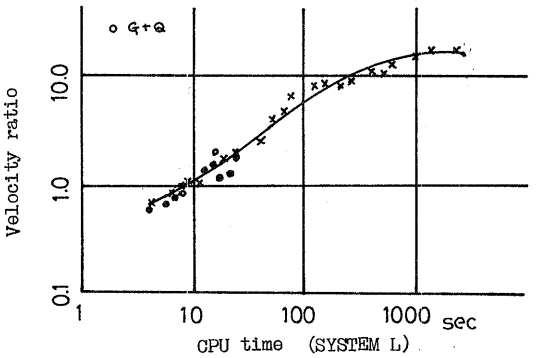

Fig.13 SYSTEM H

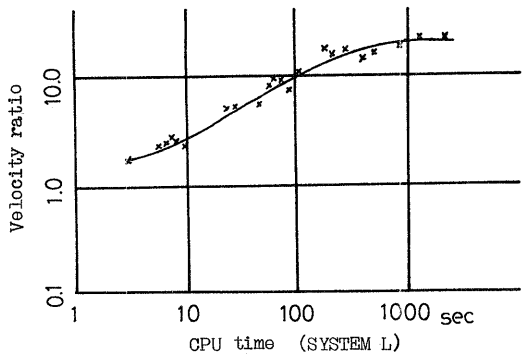

Fig.14 SYSTEM I

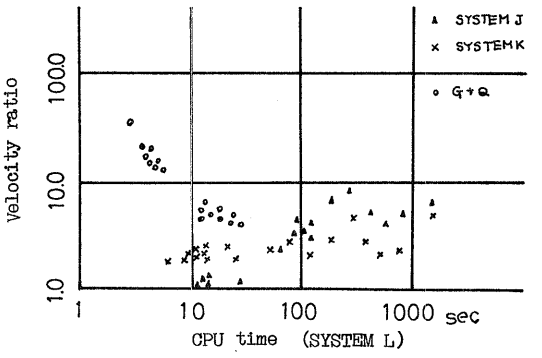

Fig.15 SYSTEM J,K

(4) システムJ，Kの検討

J，Kは同一のOSで測定したがこのシステムの時間計測の方法はFig.15にしめすように $V_{r(t)} < V_{r(t)}$ となっていることからFig.2のモデルのG-stateを計測していないことがわかる。

6 評価パラメータの検討

各計算機システムの測定データから性能評価式のパラメータを算出し、Table.3にしめす。ただし、$I_v = V_1$，$I_s = V_2$（Fig.1参照）、Gmはメーカー公称のGibsom mix値である。システムGの上段は実測値、下段は推定値による値である。

(1) α・βの検討

αはオペレーティングシステムのジョブミックスに対する反応時間の程度（Fig.1のt_2）をあらわすパラメータである。したがってαが大きいほど性能が良いといえる。

βは計算機システムの実行時の性能が翻訳・結合によって生じるオーバーヘッドを上まわるために必要な処理時間（Fig.1のt_1）を与えるパラメータで、プロセス2，3が比較的遅い計算機システムA，Bでは $G_{v(t)} \simeq Q_{v(t)}$ となり、βの値は小さく、$V_{r(t)}$ に対するオーバーヘッドの影響が大きい。マルチジョブストリームによるオーバーヘッドの影響は資源の競合がおきやすいプロセス2，3に集中する。したがって実質的にコンパイラ・リンケージエディタの性能低下となり、計算機システムG，H，Iではβは小さい。

SYSTEM.ID	K_0	K_p	α	β	I_v	I_s	G_m
A	10.15	1.10	0.0016	0.0015	9.60	0.55	2.00
B	8.06	-0.01	0.0017	-0.37×10^{-10}	8.30	0.23	2.54
C	24.30	4.80	0.0084	0.0610	31.50	12.00	0.18*
D	7.65	1.85	0.0072	0.0382	13.00	7.20	0.68*
E	3.21	1.66	0.0140	0.0394	4.65	3.10	3.16
F	2.16	1.16	0.0180	0.0385	2.20	1.20	6.31
G	73.00 / 71.60	1.06 / 3.60	0.0037 / 0.0038	0.0038 / 0.0832	74.00	2.08 / 6.00	—
H	13.30	0.09	0.0029	0.0030	14.00	0.80	1.1 ~ 1.4
I	21.40	0.40	0.0048	-0.26×10^{-6}	23.00	2.00	0.80
L	—	—	—	—	—	—	16.00

Table.3 The value of evalution parameter　　*all in buffer

(2) I_v, I_sの検討

I_vは実算的な計算機システムのCPU速度の上限を，I_sは下限をあらわす。したがって I_s / I_v の値が大きい計算機システムがバランスのとれたシステムといえる。

(3) K_o, K_p の検討

K_oはオペレーティングシステムの導入によるオーバーヘッドが原因となった性能低下の定数項をあらわし、K_pはコンパイラ・リンケージエディタによる性能向上の定数項をあらわす。K_oはなるべく小さく，K_pは大きい方が望ましい。

(4) I_vとG_mの検討

3で述べたようにI_vは Instruction mix の比近い値となるがG_mと比較してみるとかなり差があるシステムもある。その原因としては

① テストプログラムの性質
② キャッシュメモリーの効果
③ レジスタの効果的利用

が考えられる。

以上の結果、各計算機システムの評価パラメータの値が測定結果の概念的検討と一致することが認められた。

7 おわりに

計算機システムのCPU評価に対して Instruction mix は絶対評価量を与える。本稿ではジョブ処理過程をモデル化し性能評価式を求めた。この性能評価式の各パラメータはCPU評価に関する相対評価量を与える。

現時点では評価パラメータと計算機システムのどの要素あるいは事象と関連しているかは不明であるが計算機システムをさまざまな形態で測定することにより明らかになると思われる。

謝辞

本研究をおこなうには、多くの計算機システムと時間を必要とした。電通大有山教授、東大大型センター石田助教授はじめ多くの方々から御理解と有益な助言をいただいたことに深く感謝します。また日本学術振興会のユニコンを利用したことを付記します。

（参　考　文　献）

1) 情報処理学会 プログラミング・シンポジューム委員会：
“システム性能評価シンポジューム報告集”，昭和47年7月．
2) 石田晴久：“ギブソンミックスの起源について”，情報処理，Vol.13, No.5, 1972.
3) 田中・高橋：“エンプティ・ジョブの処理件数”，数理科学 1971年11月号 PP.50-54.
4) 石田晴久：“センターシステムの性能評価について(1)”，東大大型センターニュース
Vol.6, No.4, 1974.