

計算機システムのジョブ処理効率測定モニタ (JPTM) について†

末 永 正^{††} 景 川 耕 宇^{††} 武 富 敬^{††}

近年、計算機システムが複雑化するにつれ、その処理能力を正確に評価・分析する技術はシステムを運用するうえで非常に重要である。性能評価基準の二大要素であるスループットとレスポンスタイムに関して、これまでいろいろな評価システムが開発されてきたが、そのほとんどがサンプリングによる状態調査によって行われており、測定時のシステム負荷の変化といった未解決の問題も多く残されている。とりわけ、スループットの評価・分析に関しては、仮想計算機システムのような複雑なシステム構成に対応できていない。ここでは、スループット評価時の基礎となる CPU の処理効率について、サンプリングによる CPU 利用率の算出ではなく、OS が管理しているジョブの CPU 使用時間を直接計測する方法によって、測定オーバーヘッドがなく、OS 走行を含まないジョブだけの CPU 利用率を評価できる測定システムを実現した。これによって、チューニング効果を正確かつ即時的に把握することができ、しかも、仮想計算機 (VM) 環境と実計算機環境の比較といった従来のソフトウェアモニタでは行えなかった評価も特別なハードウェア測定装置を用いず可能になる。

1. ま え が き

近年、計算機システムが複雑化するにつれ、その処理能力を正確に評価・分析する技術はシステムを運用するうえで非常に重要である。性能評価基準の二大要素であるスループットとレスポンスタイムに関して、これまでいろいろな評価システム^{1)~4)}が開発されてきたが、そのほとんどがサンプリングによる状態調査によって行われており、測定時のシステム負荷の変化といった未解決の問題も多く残されている。とりわけ、スループットの評価・分析に関しては、仮想計算機システムのような複雑なシステム構成に対応できていない。また、システムをチューニングする際は、その効果を正確に調査する必要があるが、従来のサンプリング方式では平均化されたシステム状態しか認識できず、的確な分析が困難である。

ところで、計算機システムの処理効率を評価する場合、一般的に用いられる尺度に CPU 利用率がある。しかし、計算機の内部では、問題解決のためのジョブ (Problem Program Job: 以後、単にジョブと呼ぶ) のほかに、オペレーティング・システム (OS) が動作しており、CPU 利用率が 100% といっても、OS 動作が大部分を占めていたのでは、システムが有効に機能しているとはいえない。したがって、OS とジョブの CPU 利用割合を明確に区別する必要がある。

さて、システム効率を調査するには、これまでシステム内部の計算機資源 (リソース) の利用状況を監視する測定モニタが用いられていた。それらは、各種のリソース状態を一定時間ごとのサンプリングによって収集し、統計処理によって測定データを得るものである。このような測定モニタには、ハードウェアやソフトウェアによるものがあるが、ハードウェア・モニタは測定装置にかなりの費用が必要なこと、データ収集時の操作性等から、システム・チューニングの際の大部分がソフトウェア・モニタを用いて行われている。

しかし、このソフトウェア・モニタでは、上記の OS とジョブの利用比率が区別しにくいこと、および正確を期してサンプリング間隔を小さくすることによってシステム負荷 (オーバーヘッド) が増加するといった問題がある。さらに、仮想計算機 (VM) 下では、システム全体を見渡せないことによる使用制約もある。

このような処理効率測定上の問題を克服するため、筆者らは、CPU 利用効率の統一的評価法を考案した⁵⁾。ここでは、本来処理すべきジョブの CPU 使用時間をベースに、計算機システムのジョブ処理効率を直観的に把握できる形式で定義し、VM のような特殊環境に対しても適用可能なことを示す。

2. 測定システム

2.1 CPU の処理効率

CPU 利用率は、通常、一定間隔で CPU の動作状態をサンプリングし、その集計によって算出されている。ここで得られるものには、

- i) スーパーバイザ・モード、システム・キー

† A Performance Measurement Monitor (JPTM) of a Computer System by TADASHI SUENAGA, KOU KAGEKAWA and HIROSHI TAKETOMI (Computer Center, Kyushu University).

†† 九州大学大型計算機センター

表 1 CPU 利用率の測定例
Table 1 An example of CPU utilization.

Master scheduler	System space	User spaces	Total busy	Wait
2.72%	10.2%	87.05%	99.99%	0.01%
Supervisor	Mode	Problem	Program mode	
system key	user key	system key	user key	
46.29%	0.83%	0.50%	50.38%	

- ii) スーパーバイザ・モード, ユーザ・キー
- iii) プロブレム・モード, システム・キー
- iv) プロブレム・モード, ユーザ・キー

の4種類の動作比率, および

- a) マスタスケジューラ
- b) システム空間
- c) ユーザ空間

の使用比率等がある (表1).

しかし, これらの値では, OS 走行時間 (オーバーヘッド) 等を正確に区別することができないため, 実際に有効なジョブの CPU 利用率を求めることができない.

ここでは, ジョブの実行だけに使用される CPU の利用比率をジョブ処理効率と呼び, 以下のように定義する.

時刻 t において実行状態にあるジョブの集合を J_t とし, ジョブ $x (x \in J_t)$ がそれまでに使用した CPU タイムを $U_{x,t}$ とする. また, 時刻 t から $t + \Delta t$ の間に終了したジョブの集合を $S_{t,\Delta t}$ とし, 終了したジョブ $y (y \in S_{t,\Delta t})$ の CPU タイム総量を $U_{y,\Delta t}$ で表すと, t から $t + \Delta t$ 間に全ジョブが使用した CPU タイム $T_{t,\Delta t}$ は

$$T_{t,\Delta t} = \sum_{x \in J_{t+\Delta t}} U_{x,t+\Delta t} - \sum_{x \in J_t} U_{x,t} + \sum_{y \in S_{t,\Delta t}} U_{y,\Delta t}$$

で求められる. これを CPU 台数 (a), および計測間隔時間 (Δt) で正規化して, 時刻 t におけるジョブ処理効率 θ_t を

$$\theta_t = T_{t,\Delta t} / (a \times \Delta t) \tag{1}$$

と定義する.

ここで求められる θ_t は以下の点で従来のソフトウェアモニタの測定データと異なっている.

- (i) 測定値がジョブの CPU 使用時間と実時間との比であること.
- (ii) リソース管理情報から直接算出しているため測定精度が計測間隔 (Δt) に依存しないこと.

この方式では, θ_t が正確に得られることによって, いままで定量的な評価が困難であった次のようなことが

可能になる.

- システムオーバーヘッドに関する OS の機能別内訳の分析
- システム環境による効率変化の分析

ただし, CPU が低負荷状態において, システム環境によるオーバーヘッド比較を行うには, CPU ウェイト時間も θ_t に反映させなければならない. そのためには, 最低実行優先権をもつ無限ループジョブによって, 見かけ上 CPU を過負荷状態にしておく必要がある.

なお, (1)式で得られる θ_t は, システムが固定のサービス量 (ここでは CPU タイム1秒) を要する特殊なトランザクション処理系であると考えれば, CPU 1台あたりのスループットとみなすこともできる.

2.2 測定方法

一般に, ジョブは, TSS ジョブ, バッチジョブ, オンラインジョブ等からなり, CPU タイムが, OS の走行時間と区別して記録されている. ここでは, このような OS の CPU タイム収集機能を前提に, FACOM OSIV/F 4 の下で, θ_t を求めることのできる測定モニタ (Job Processing Time Monitor: JPTM) を開発した. 以下に, 測定環境のソフトウェア概念図 (図1) と JPTM の処理概要 (θ_t の算出法) を示す.

- i) 単位時間 (ここでは1分) ごとに, OS のタスク管理制御テーブルから CPU タイムの累積値 $U_{i,t}$ (時刻 t_k における累積値) を全ジョブについて求め, 時刻 t_{k-1} の $U_{i,t_{k-1}}$ との差を計算する.
- ii) 時刻 t_k と t_{k-1} との間に終了したジョブの CPU タイムを i) の値に加算し, 計測間隔時間 ($t_k - t_{k-1}$) をプロセッサ台数 (a) 倍した値で割り, θ_t ((1)式) を求める.
- iii) θ_t , プロセッサ台数, ジョブ多重度 (アクティブ数), 前回測定時刻からの CPU タイム合計および当時刻での各ジョブの CPU タイム累積値を下記 (図2) の形式でファイル出力

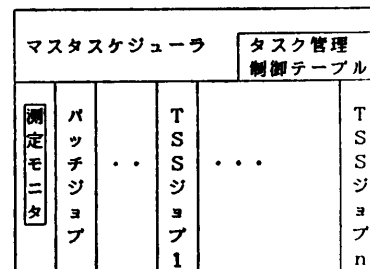


図 1 ソフトウェア概念図
Fig. 1 Software configuration.

測定時刻	θ _i	プロセッサ数	バッチ多重度	バッチのCPUタイム合計
TSS多重度		TSSのCPUタイム合計		ジョブ番号1
CPUタイム累積値1		ジョブ番号2	CPUタイム累積値2 . . .	

図2 出力データフォーマット
Fig. 2 The output data format.

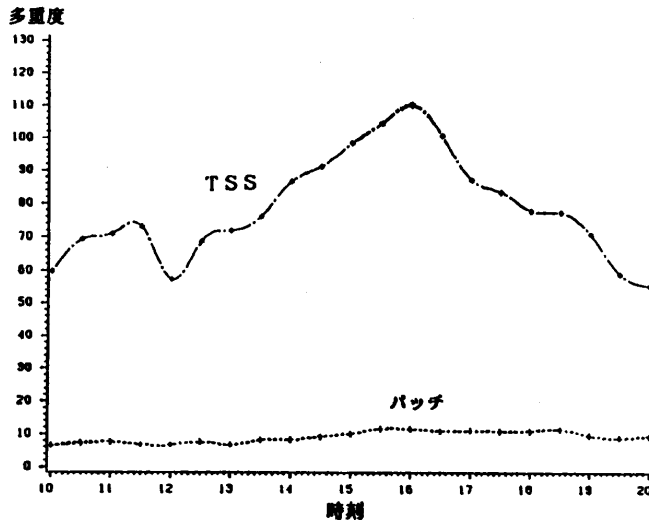


図3 ジョブ多重度 (TSS, バッチ) の時刻による変化
Fig. 3 The number of active TSS terminals and BATCH jobs.

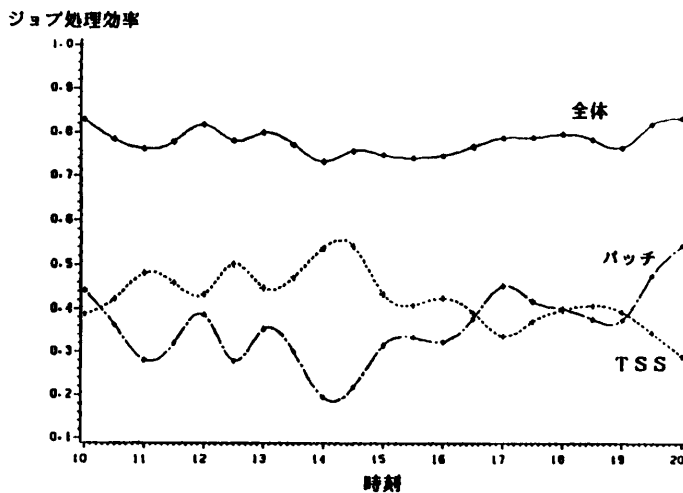


図4 ジョブ処理効率
Fig. 4 Job processing performance.

する。

2.3 測定データの分析

JPTM によって得られるデータは、SAS⁵⁾その他の統計パッケージによって次のような分析を行うことができる。

- i) ジョブ多重度の時刻による変化 (図3)
- ii) ジョブ処理効率の時刻による変化 (図4)
- iii) ジョブ処理効率と TSS アクティブ端末数との相関 (図5)

- iv) CPU タイム割当ての度数分布 (図6)

これらの分析は、前述した出力データ (図2) をいろいろな観点から統計処理することによって得られるものであり、時々刻々変化するシステムの処理状況を的確に把握することができる。ここでは、昭和61年2月18日に行った測定データの分析例を示している。図3および図4は、システム状況 (TSS アクティブ端末数、バッチ多重度、ジョブ種類別ジョブ処理効率) の時系列変化を表したものである。図5は、TSS アクティブ端末数に対するジョブ処理効率の平均をプロットしたもので、TSS アクティブ端末数を x 、ジョブ処理効率を y としたときの回帰直線は以下の式で示される。

$$\text{全体: } y = 0.900 - 0.00145x$$

(相関係数 -0.931)

$$\text{TSS: } y = 0.346 + 0.00085x$$

(相関係数 0.453)

$$\text{バッチ: } y = 0.554 - 0.00230x$$

(相関係数 -0.747)

図6は、ある1分間において、各ジョブが割り当てられた CPU タイムの度数分布を表したもので、TSS の全アクティブ端末のうち30台は何も実行しておらず、大半が1秒以下の CPU タイム割り当てであることが分かる。また、バッチジョブには2~3.5秒の CPU タイムが割り当てられている。

2.4 特徴

JPTM の測定法は極めて単純であるが、これから得られるデータは直観的に分かりやすく、興味深い。しかも、OS のジョブに関する CPU タイム収集方法が変更されない限り、システム環境変化の際の共通の効率評価基準として有効である。ところで、VM 等の特殊環境においては、ハードウェアが同じであっても実計算機との CPU 速度差が懸念される。これは、VM モニタの特権命令シミュレーションや入出力 (I/O) 介入のた

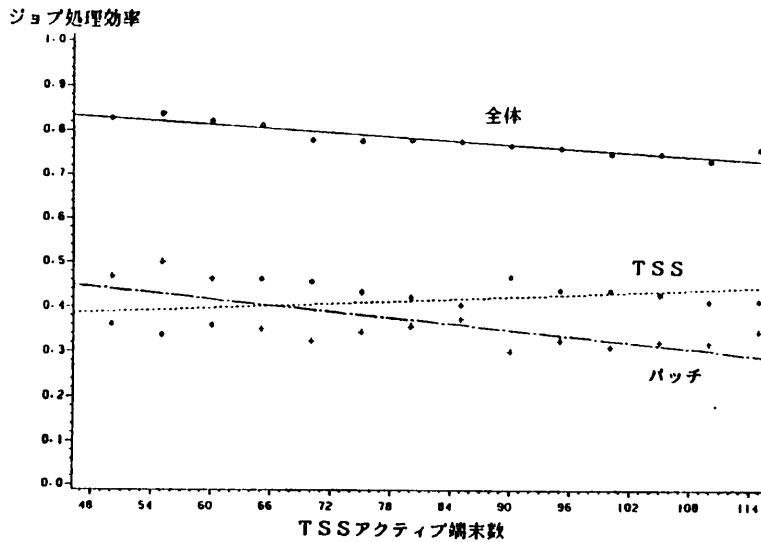


図5 TSS アクティブ数に対するジョブ処理効率

Fig. 5 Relation between active TSS terminals and job processing performance.

表2 CPU 速度比較の例

Table 2 An example of CPU time.

	件数	平均(秒)	最大(秒)	最小(秒)	標準偏差
実計算機	12	19.65	19.91	19.47	0.15
VM	11	19.68	19.90	19.49	0.10

めに、一部の命令実行時間（ファームウェア実行ステップ数）が変化するためであるが、CPU タイムとしてカウントされる時間は、ハードウェアの改良⁶⁾もしくはタイマ補正⁷⁾によって実計算機のとおり値が得られるように工夫されている。表2に、SAS 統計パッケージを用いて 17,000 回の I/O 処理を行うジョブを FACOM M380S システムの両環境で実行した場合の CPU タイム比較を示す。VM および実計算機における CPU タイムの差は、同一環境の再現誤差と比較して十分小さく、差異は認められない。

また、この計測法では、測定のためのオーバーヘッド増加がほとんど生じない (FACOM M380 で 1 回の計測所要時間は 1 msec 以下、60 秒間隔で計測した場合 0.001% 程度)。したがって、システムに常時組み込んで効率上問題なく、利用状況のオンライン観測に役立てることができる。

3. 適用例

ここでは、九州大学大型計算機センターの 3 機種システムのジョブ処理効率の測定結果を示す。測定時の主なハードウェア構成は

- スカラプロセッサ FACOM M382 (主記憶 48 MB), ディスク 24 台 (40 GB), MSS (100 GB), MT 装置 8 台, TSS 端末 231 台 (周辺型 73, 回線型 158), リモート・ジョブ・エントリ端末 15 台
- スカラプロセッサ FACOM M380S-AP (主記憶 64 MB), ディスク 24 台 (40 GB), MSS (100 GB), MT 装置 8 台, TSS 端末 319 台 (周辺型 73, 回線型 246), リモート・ジョブ・エントリ端末 15 台
- ベクトルプロセッサ FACOM VP-100 (主記憶 32 MB), ディスク 24 台 (40 GB)

である。スカラプロセッサのソフトウェア環境は、OS が FACOM OSIV/F4 で、ジョブとして科学技術計算 (Fortran), 文書処理, 図形処理, 統計処理等が多く使用されている。一方、ベクトルプロセッサでは、OS として FACOM VSP を使用し、ジョブは Fortran のバッチ処理だけある。JPTM の測定は、CPU の処理能力限界 (CPU ビジー率 100%) に近い状況で行った。このような CPU 負荷は、スカラプロセッサでは、バッチジョブが 4 多重以上、TSS アクティブ数が 50 以上のとき、ベクトルプロセッサでは、バッチジョブが 3~4 多重の場合に常時観測される。図7に M382 および M380S-AP システムにおいて、TSS アクティブ端末数が変化したときのジョブ処理効率の変動を示す。ここでは、各システム 3 日間の測定データより、TSS アクティブ数に対する平均的なジョブ処理効率を求めて比較したものである。M382 システムに比べて、M380S-AP ではジョブ処理効率が低下していることが分かる。それぞれの回帰直線は以下の式で示される。

$$M382: y = 0.946 - 0.000876x$$

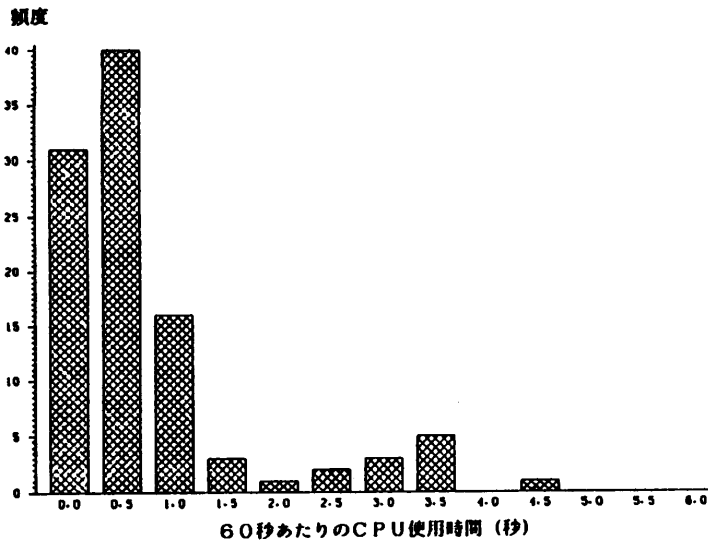
$$(\text{相関係数 } -0.918)$$

$$M380S-AP: y = 0.923 - 0.00166x$$

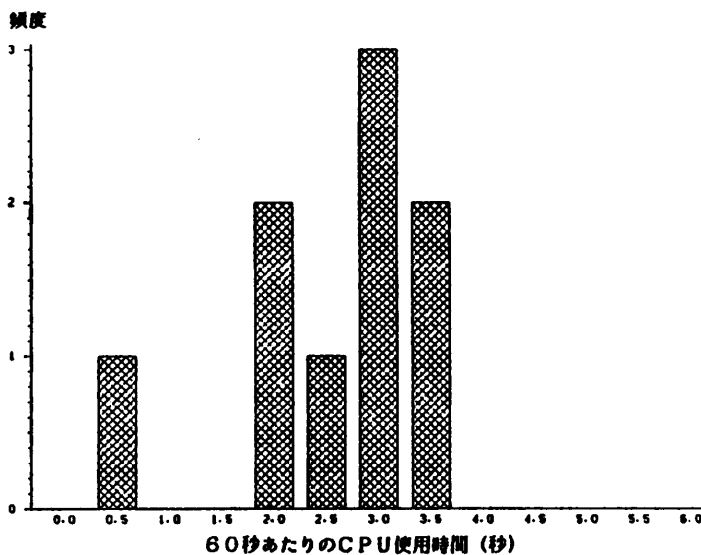
$$(\text{相関係数 } -0.971)$$

$$(y = \text{ジョブ処理効率}, x = \text{アクティブ端末数})$$

両者の違いは、測定時の OS の機能レベルが異なるため、そのままマシン性能差とみなすことはできない



(a) TSS ジョブ



(b) バッチジョブ

図 6 CPU の割当て分布

Fig. 6 A chart of CPU assignment.

が、ハードウェアの CPU 速度が低下するほど軽い OS が必要という常識が定量的に証明できている。

図 8 に VP-100 のジョブ処理効率時間推移例を示す。この例では、昭和 61 年 2 月 18 日の午前 10 時から翌日午前 0 時までの測定結果を示している。スカラプロセッサに比べて、OS のオーバーヘッドが非常に少なく、CPU のほとんどがジョブの実行に使用されているのが分かる。

4. 他の測定モニタとの比較

JPTM では、エンドユーザおよび計算機運用管理

者が知りうるジョブの CPU タイムからシステムの処理効率を測定するものであるが、従来のサンプリング方式（サンプリングモニタ）による測定値との比較を以下に述べる。2.1 節で述べたように、サンプリングモニタでは 4 種類の CPU モードの動作比率、および三つの空間タイプの CPU 使用比率を測定する。これらの CPU 使用比率の中でジョブ処理効率と最も関連のあるものは、CPU モードの中のプロブレムモード・ユーザキー動作の比率 (PU) およびユーザ空間の CPU 使用比率 (US) である。図 9 に、M380S-AP において、TSS アクティブ数に対する PU, US, および JPTM から得たジョブ処理効率 (JP) の比較例を示す。この例は、昭和 61 年 2 月 17 日の 1 日分の測定データを比較したものであるが、PU および US は 1 時間ごとに 5 分間のサンプリングによって得られたものをプロットしている。この図からも分かるように、PU および US は測定時のばらつきが大きく、また、ジョブ処理効率とはかなり異なった値を示している。

システムの処理状況を分析するには、ジョブの CPU タイム変化を的確に把握する必要があるが、従来の測定モニタでは、サンプリング間隔が比較的長く、状況変化の即時対応性に欠ける。

5. あとがき

以上、システム効率の評価・分析のための測定ツール (JPTM) について述べた。このツールは主に次のような特徴を持つ。

- (1) 測定ツール自身によるオーバーヘッドを伴わない。
- (2) 評価データが計算機使用者にとって分かりやすい。
- (3) システム状態のオンライン観測によって、通常非常に検出しにくい OS の障害 (enable loop) も容易に発見できる。
- (4) VM 等の特殊な動作環境においても統一的に使用でき、OS やハードウェア変更時の

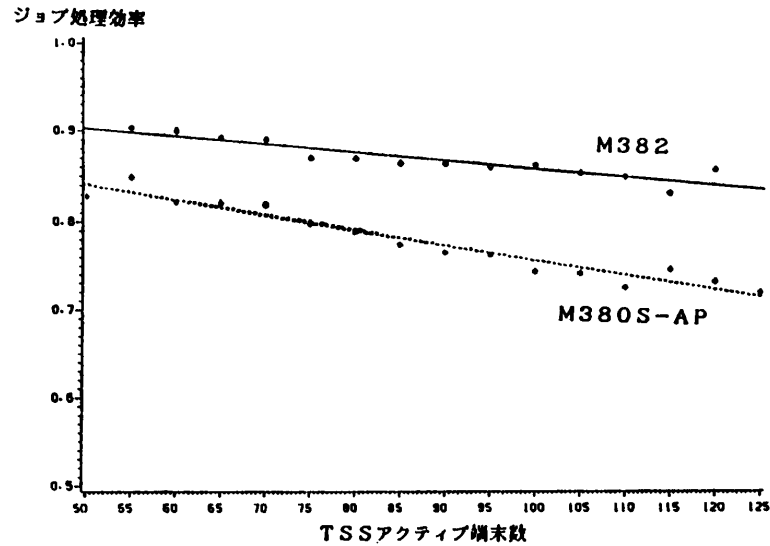


図7 M380 と M380 S-AP のジョブ処理効率
Fig. 7 Job processing performance of M382 and M380 S-AP systems.

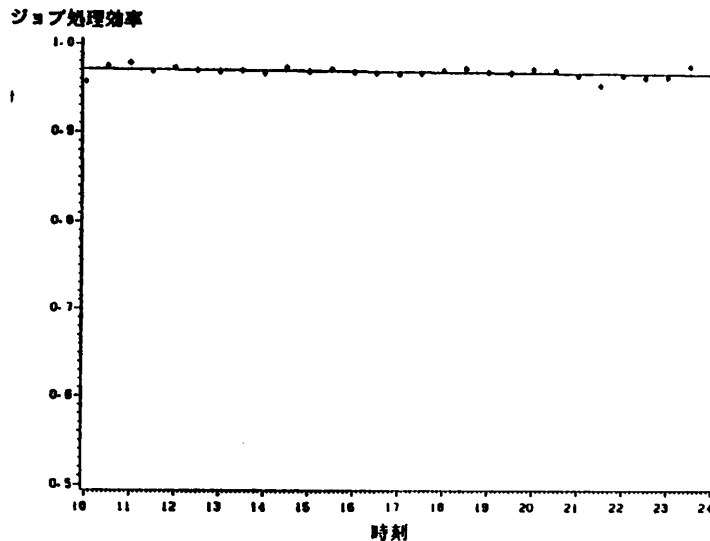


図8 VP-100 のジョブ処理効率
Fig. 8 Job processing performance of VP-100 system.

オーバーヘッド比較等に利用できる。

一方、同じハードウェアのもとで、同一ジョブを実行したときの CPU タイムが異なって収集されるような異種 OS の効率比較には利用できない。しかしながら、このような効率比較においても、その CPU タイム差を調査することによってジョブ処理効率を補正し、概要を把握することは可能である。

最後に、JPTM では本論の適用例のほかに、VM モニタの効率について興味深い測定データを得ているが、これらについての詳細は別稿で報告したい。

謝辞 日頃ご指導と有益なご助言を頂いている九州大学工学部吉田将教授、牛島和夫教授、総合理工学研究科宇津宮孝一助教授、並びに情報処理教育センター藤村直美助教授に謝意を表します。

参 考 文 献

- 1) 田畑, 藤井, 黒嶋, 清水: 運用環境における TSS 計算機システムの性能測定, 情報処理学会論文誌, Vol. 22, No. 6, pp. 600-604 (1981).
- 2) 藤村, 牛島: TSS 計算機システムの性能解析・

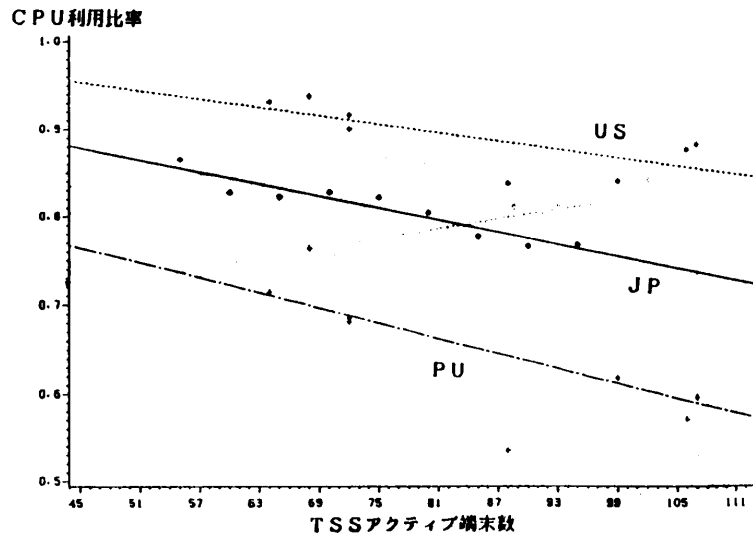


図9 JPTM と他モニタの測定値比較

Fig. 9 The comparison between JPTM and other measurement monitor.

評価システムとその適用例, 情報処理学会論文誌, Vol. 26, No. 6, pp. 365-373 (1985).

- 3) IBM: OS/VS2 MVS RMF General Information, マニュアル GC 28-1115.
- 4) 富士通: FACOM OS IV/F 4 MSPDDL/PDA 使用手引書, マニュアル 78 SP-2190.
- 5) SAS: SAS User's Guide/Basics, 1982 Edition.
- 6) 小野: 仮想計算機システム高性能化の一手法, 情報処理学会計算機システムの制御と評価研究会資料, 18 (1983).
- 7) 梅野, 大町, 井村: 仮想計算機システムにおける時間誤差の分析, 第21回情報処理学会全国大会論文集, pp. 417-418 (1980).
- 8) 末永, 景川, 吉田: スループット直接計測法とその適用例, 電気関係学会九州支部連合大会(第37回連合大会)講演論文集, 121 (1984).

(昭和61年4月7日受付)

(昭和61年7月16日採録)



末永 正 (正会員)

昭和24年生. 昭和47年九州大学工学部電子工学科卒業. 現在, 九州大学大型計算機センター助手. オペレーティング・システムのマンマシンインタフェース, および効率評価

の研究に従事.



景川 耕宇 (正会員)

昭和15年生. 昭和37年東京大学工学部応用物理学科(数理工学専攻)卒業. 昭和39年同大学院修士課程修了. (株)三永通信, (株)新日本パイプを経て昭和42年より九州大学勤務. 現在九州大学大型計算機センター講師. 主な研究分野: 計算機システムの性能評価.



武富 敬 (正会員)

1947年生. 1970年九州大学理学部物理学科卒業. 1975年同大学院理学部研究科博士課程単位修得退学. 同年九州大学大型計算機センター助手, 現在に至る. センター運用・管理のための情報システムの構築, プログラムライブラリ管理システム, 計算機システムの評価等に興味を持つ. 生物物理学会会員.