ハードウェア・モニタによるシステム測定*

箱崎勝也** 小野隆喜**

Abstract

System measurement is a fundamental technique in designing a computer system. This paper describes concepts of a system measurement, a hardware monitor named SYDAS (System Data Acquisition System) based on these concepts, and some of the results obtained by the use of the SYDAS.

The SYDAS, which processes the system signals detected by the attached high-impedance probes on the wiring of a computer, measures the activities of the working system without affecting both the hardware and the software. The use of an associative memory and associated counters makes it possible to acquire system data in more flexible ways, and to reduce the volume of the data by extracting the data concerned with a specific measurement.

The hardware characteristics of a computer system, the program behavior in forms of the SVC sequence and the working set, and the system profile are shown as an example of system measurement.

1. まえがき

技術の分野では，「技術の進歩は，評価技術の発展に依存する」ということが経験され，認識されている．

こと，コンピュータ・システムの分野でもこのことは言えよう．コンピュータ・システムの評価技法，とくに，性能の評価技法については，今まであまり考慮されていなかった．しかし，最近の複雑・大規模化したコンピュータ・システムでは，この技法なしには設計不可能と言われるまでになった．そこで，システム性能の評価の分野にも多くの労力が注がれ，その技法も各種のレベルのものが開発され，広く利用されている．たとえば，待行列理論を中心とした解析的手法や，コンピュータ・シミュレーションなどがそれである．前者は，多くの仮定のもとに，システムの極めて限定された個所にしか普通は適用できないし，後者は膨大なシミュレーション時間と労力を必要とする割りに結果が，モデルやパラメータの設定に大きく左右される．これらは，有効な性能評価技法を開発するための有意義なデータが，今まで蓄積されていないことに起因する．かような状況から，実際に移動しているシステムを直接測定することが行なわれて来ている[1]．

システム測定には，大別して，特殊なプログラムによりデータを収集するソフトウェア・モニタと[2),3),4)]，特別な測定装置により被測定システムの電気信号を直接取り出して測定するハードウェア・モニタ[5),6),7),8)]の二種がある．これらは，それぞれ一長一短を有している[11]．

一般的に言ってソフトウェア・モニタは，測定の融通性やプログラムに関連した情報を測定しうる利点がある反面，モニタ自身が被測定システムのリソース (CPU 時間，メモリ，I/O など) を使用するために，被測定システムに何らかの影響をおよぼすことは，避けられない．特に，高い頻度で発生するイベントに対するデータ測定が必要な場合には，被測定システムへの影響は，無視できなくなる．

一方，従来のハードウェア・モニタでは，電気的プローブをシステム内に挿入することによる保守者への心理的な影響を除くと，被測定システムへの与える影響はほとんど皆無である．しかし，これは，測定の融通性に乏しく，プログラムと直接関連づけられたデータの収集が（不可能ではないにしても）極めて困難で

* System Measurement Using a Hardware Monitor, by Katsuya Hakozaki and Takaki Ono (Central Research Laboratories, Nippon Electric Co., Ltd.)

** 日本電気株式会社中央研究所

あった．何を測定すべきか，また，測定されたデータをどのように使用すべきかが明確になっていない現状では，多種多様な測定を行なってみる必要がある．この意味からも測定の融通性は，測定装置の持つべき機能の一つである．同様に，プログラムに関連したデータ，たとえば，どのプログラム・モジュールのどの部分が最もよく使用されるかといった類のデータは，プログラム改良のために必要なデータである．このようなデータを直接的に測定することは，性能を評価する時，非常に重要である．

ここに紹介する SYDAS[9,10] (System Data Acquisition System) は，連想記憶装置を用いたことにより，融通性とプログラムに関連したデータの収集機能を備えたよりソフトなハードウェア・モニタであり，システムの測定とともに，その測定技法を研究する目的で開発したものである．

2. システム測定の概念

SYDAS について述べる前に，システム測定の概念について，われわれの見解を簡単に説明する．

われわれはシステムの動作を，システム状態 ($S(t)$) が時間とともに変化することであると定義する．システム状態は，システムを構成する各要素の状態の集合であるとし，微視的に見れば，システムの中のすべての素子の状態の集合を含む．システム状態のサブセットとして，たとえば，CPU の動作状態，I/O 装置の状態等が定義される．有限時間 ($t_a \sim t_b$) 内におけるシステム動作は，

$$S(t),\ t_a \leq t \leq t_b$$

で表わされる．

システム測定は，測定目的に応じた対象とする $S(t)$ のサブセット ($s(t) \subset S(t)$) を，有限時間内で測定することにより，システム動作を部分的に見ることである．$s(t)$ はそれと等価なシステム状態信号 ($s^*(t)$) として検出され，たとえば，CPU が割込みモードであるという状態は，それを表わす論理素子の状態信号で検出される．システム測定においては，選択された $s(t)$ に対応するシステム状態信号を適当な時間でサンプリングした時系列 $\{s^*(t_i)\}$ として測定される．

$s^*(t)$ の変化点をイベントと定義し，イベントが発生する時刻の列を $\{t_i\}$ とすると，サンプリングを $\{t_i\}$ で行なうことによって，$s^*(t)$ の測定を $\{s^*(t_i)\}$, $(i=1,2,\cdots)$ の測定に置き換えることができる．

結局，システム測定はイベント発生時刻でサンプリングされたシステム状態をシステム状態信号から求めることであり，これをイベント時系列と呼ぶ．

具体的には，次のような基本機能によって，システム測定が行われる．

(1) システム状態信号の検出

被測定システムのシステム状態を知るための，システム状態信号の検出であり，能動的または受動的に検出される．

(2) イベントの抽出

検出されたシステム状態信号がそのままイベントとして使用しうる場合，すなわち，被測定システムで検出されたシステム状態信号が注目するシステム状態の変化点に対応する場合は，システム状態信号の検出がイベントの抽出になるが，そうでない場合には，測定側で何らかの方法でイベント抽出が行なわれる．

(3) 測定データの記録

測定には測定データを記録もしくは表示する何らかの機能が必要である．データ量を少なくするために，単にイベント時系列を記録するのではなく，ある程度の統計的な処理が施された形で記録することが普通であり，ハードウェア・モニタでは多くの場合カウンタを用いて，イベント発生頻度や特定のシステム状態にあった時間等の平均値や累積値として記録することが多い．しかし，イベント時系列として記録されたデータは最も汎用性のあるデータを得ることが出来る利点がある．

3. SYDAS

前節で述べた概念にもとづいて，SYDAS ではどのようにしてそれらの機能が実現されているのかを具体的に述べることにする．

3.1 構　成

SYDAS は次の各部により構成されている．

- プローブ
- レベル変換装置
- タイマ (精度 1μs, 36-bit)
- カウンタ (36-bit×2)
- 連想記憶装置 (24-bit×128 tag; 32-bit×128 カウント用メモリ; サイクル・タイム80ns)
- コア・メモリ (8 k-36-bit word; サイクル・タイム) 1.5 μs)
- メイン・コントロール
- コントロール・パネル

◦ 磁気テープ装置
◦ NEAC M 4 ミニコンピュータ

SYDAS の構成図を Fig. 1 に示す.

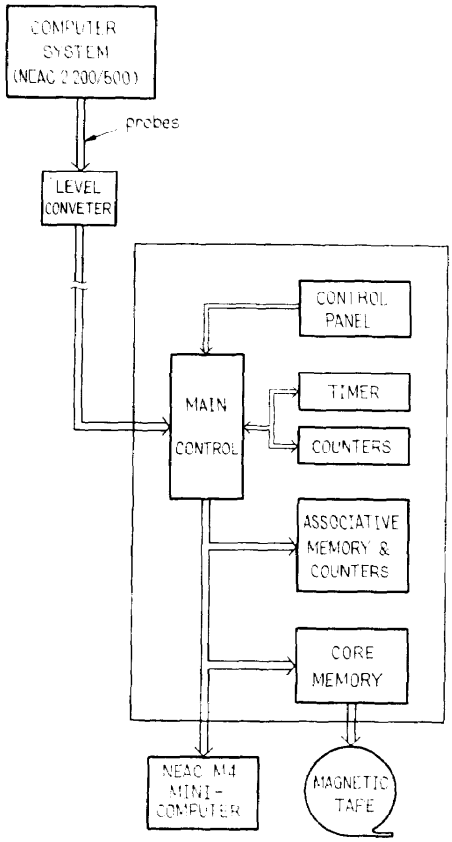

Fig. 1 The system configuration of the SYDAS

3.2 測定機能

(1) システム状態信号の検出

被測定システムの配線面に取り付けられた高い入力インピーダンスを持つプローブにより，システム状態信号が検出される.

現在，システム状態信号として，次のような約60本の信号が選ばれている.

◦ メモリ・アドレスとそのアクセス要因（命令としての読出し，オペランドの読出し／書込み，I/O制御部からの読出し／書込み等）
◦ 実行中の命令コード
◦ 入出力チャネル使用状態（ビジー信号）
◦ CPU 動作モード（割込みモード等）
◦ 各種制御信号（タイミング，その他）

これらの信号は，レベル変換器により，増幅・整形され，SYDAS 本体の信号レベル（TTL）に変換された後，ケーブルを介して本体に送られる.

(2) 測定の制御

SYDAS 本体に送られて来たシステム状態信号は，コントロール・パネルで指定された測定目的に従って，メイン・コントロールのもとに各部へ送られ，測定が行われる．SYDAS には次の4つの測定ルートが用意されている.

ルート(1) イベントの発生頻度またはシステム状態時間を求めるために，コントロール・パネルにより選ばれたシステム状態信号がカウンタに送られ，イベント発生回数または被測定システムが特定のシステム状態にある時間が測定される．測定データは，コントロール・パネルの指示により，1秒または10秒ごとに表示され，またミニコンピュータに送出される.

ルート(2) イベントの発生ごとにコントロール・パネルで指定されたシステム状態および必要な場合にはその時のタイマの値とをコア・メモリを介して磁気テープに記録する.

ルート(3) あらかじめ注目するイベントに対応するシステム状態を連想記憶装置に記憶しておき，測定時に，その時点のシステム状態信号で連想記憶装置の記憶内容を検索する．連想記憶装置ではすべての語について並列的にシステム状態が調べられ，一致したものがあれば一致信号，すなわちイベントを発生する．このようにして抽出されたイベントは，連想記憶装置の各語に対応したカウント用メモリによりイベント発生頻度が計数される．測定データは一定周期(100ms, 1 sec, または10sec) ごとか，または測定終了時に磁気テープに記録される.

ルート(4) ルート(3)の場合と同様にして抽出されたイベントごとに，その時の対象とするシステム状態とタイマの値とをイベント時系列として磁気テープに記録する.

これらの4つのルートを，システム状態信号との結合条件を変えることにより，いろいろなシステム測定が可能である.

4. システム測定方法と測定例

SYDAS は現在，社内の主として科学技術計算に使用されている NEAC 2200 モデル 500 システムにプロ

ーブを設定し，システム測定，解析を行なっているが，ここにその概要と得られた結果の一部を紹介する．

4.1 ハードウェア特性

システムの構成要素であるハードウェアの特性は，システム設計の基本的なデータであるが，CPU の性能をとってみても，アプリケーションのパターンによりかなりの差違が出てくるものである．ここでは，被測定システムのアプリケーションにおける CPU の特性を明確にする目的で，次のような項目について測定を行なって来ている．

（1） 命令使用頻度

割込モード，ノーマル・モードにおける命令使用頻度を，ルート（3）の機能，すなわち連想記憶装置を用いたイベント（各命令の実行）発生頻度の測定機能を用いて測定する．

Fig. 2 にグルー別の命令使用の分布，Fig. 3, 4 にそれぞれユーザプログラムとモニタプログラムの命令使用頻度の累積曲線を示す．第3, 4図から判るように約10個の命令の使用頻度が90％近くに達している．従って，極端な考え方をすれば，これらの10命令を実現するのに90％のコストを注ぎ込んでも良いということになる．

（2） 平均命令実行時間

平均命令実行時間は，ルート（1）のカウンタにシステム状態信号として命令実行に相当する信号を加えることにより，〔測定時間／命令実行数〕として求める．

（3） メモリ・サイクルの利用率

これは，ルート（1）の機能の2つのカウンタを用いてハードウェアの各モジュールからのメモリ・アクセス回数を一方のカウンタにより求め，他のカウンタで使用可能なメモリ・サイクルを計数することから，メモリがどのハードウェア・モジュールからどの程度使用されているかが求められる．

特に I/O チャネルからのメモリ・サイクルの利用率は CPU の設計パラメータとして重要である．

これらのハードウェア特性をアプリケーション・パターンについて測定することは，そのアプリケーションに最適化されたプロセッサの設計を可能にする．将来，マイクロプログラム技術等の進歩によって，柔軟性のある設計が可能になれば，このような測定データがアプリケーションごとの最適化に使用されるであろう．

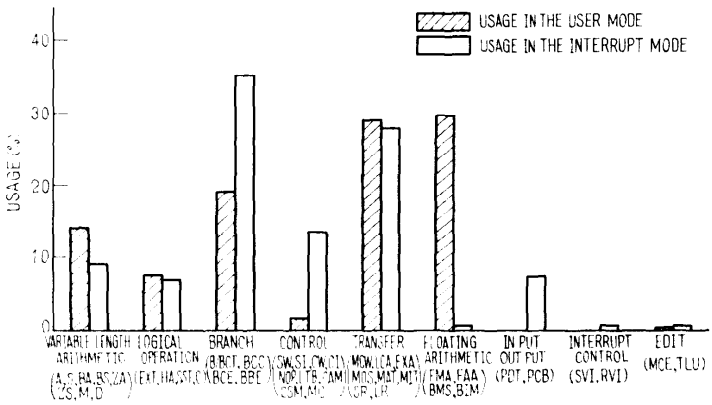

Fig. 2 Distribution of instruction usage

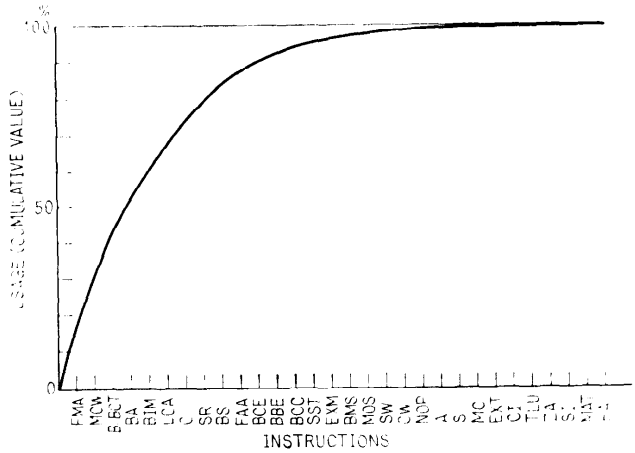

Fig. 3 Instruction usage in the user mode

4.2 プログラム特性

プログラムには，大きく分けてユーザーによって作られるユーザ・プログラムと，システムに用意されるシステム・プログラム（特に OS）とがある．システム側から見たとき，ユーザ・プログラムの内容は，一般に，そのユー

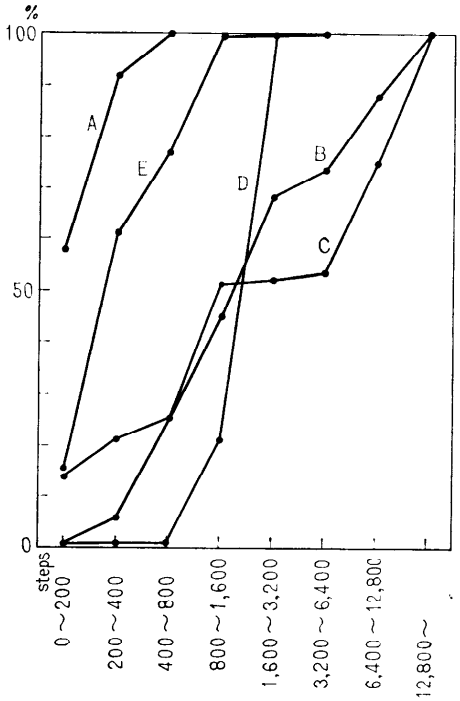

Fig. 4 Cummulative distribution of SVC interval steps

ザにしか判らないのに対し，システム・プログラムの方はその機能内容が明確に判っており，かつ，かなり常駐部分が多いという相違がある．ここではユーザ・プログラムに対しては，そのコントロール・プログラムの常駐部分について，より微視的な特性を求めることとして，次のような項目について測定・解析を行なっている．

（1） SVC 系列

ユーザ・プログラムの動作を表わすモデルとして，プログラムが発するスーパーバイザ・コール (SVC) の時系列でこれをとらえる見方があり，このモデルは CS モデル[12] の概念に含まれるものである．これは，システム・シミュレーションにおいて，プログラムの時間的な動作を表現する一つの有効な方法である．

SVC 系列にはルート（4）のイベント時系列の測定機能を用いている．SVC はユーザ・プログラムからシステム・プログラムに対して発せられるものであるから，システム・プログラム内の各 SVC に対する処理ルーチンを通過したことをもってイベントすると，そのイベント時系列が SVC 系列となる．したがって，システム状態として，実行中のプログラム・アドレスを選び，連想記憶装置に注目する処理ルーチンの入口アドレスをセットしておくことにより，SVC がイベントして抽出され，その時系列が求められる．

得られた SVC 系列データを解析することによって，SVC から次の SVC までの間隔の分布 (Fig. 4). 平均 (Table 2) SVC の種類別の発生頻度等が求められる．シミュレーション入力としては，このようなパラメータで表現されたプログラム特性を取り扱うことで十分な場合が多いと思われる．

	PROG. SIZE	PROGRAM TYPE	SVC INTER-VAL (MEAN)
A	12 K-CHAR.	INPUT READER	210 Steps
B	200K	FORTRAN COMPILER	4,970
C	200K	PROBLEM PROGRAM (FORTRAN)	16,820
D	60K	PROBLEM PROGRAM (FORTRAN)	1,823
E	—	PROBLEM PROGRAM (BUSINESS USE)	500

Table 2 SVC Interval

（2） OS 特性

OS 特性として，OS の各モジュールのアクセス頻度，その処理時間を求め，システム設計の基礎データを得ることと，OS の改良すべき部分を見いだすことを目的とした．

測定は SVC 系列と同様にルート 4 を用いたイベント時系列データとして得られる．この場合，連想記憶装置には注目する OS モジュールの入口，出口のアドレスをあらかじめ設定しておくことで，モジュールの到着・通過ごとにイベントが得られる．

（3） Working Set

プログラムの空間的な動作のモデルとしては，P.J. Denning の Working Set モデル[13] がある．これはプログラムがある時刻 (t) からそれ以前の τ 時間中に使用したメモリの集合を Working Set $W(t,\tau)$ としてとらえるもので，これは主記憶装置の動的配置の問題の評価に有用な考え方である．

Working Set の測定には，ルート（3）の機能が用いられる．すなわち，アクセスされたメモリ・アドレスをシステム状態として選び，連想記憶装置に対象とするアドレス領域をセットしておき，その領域へのアクセスが行なわれるごとにその領域に対応するカウント用メモリでカウントする．このカウント内容を一定期間ごとに磁気テープに記録することにより Working Set を測定することが出来る．記録されたアドレス・パターンを Fig. 5 に示す．

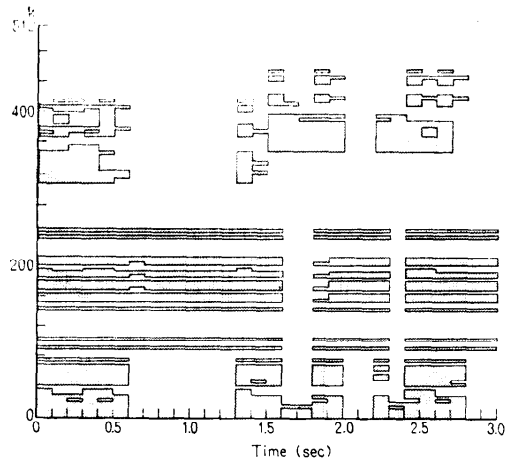

Fig. 5 Address Pattern

メモリ領域の指定はアドレスの下位ビットをマスクすることで代用しているために，2のべき乗を単位とした領域の設定のみが可能であるが，ページング方式を想定するとこの制限は問題ではない．連想記憶装置は128語であるから，4k桁を1つの領域とすると512k桁の領域を測定することが可能である．

4.3 システム特性

前節までに述べたハードウェアとプログラムの特性もシステム特性の一種であるが，ここでいうシステム特性はもう少し巨視的にシステムを見た時のバランス，即ちリソースの使用率を中心にした測定を考える．

(1) システム・プロフィール

リソースの中で，特にCPUとI/Oの動作のバランスを見るために，各々のリソースの使用状況，オーバラップ状態等をシステム・プロフィールと呼ぶ(Fig. 6)．これはシステム構成の改善などに役立つデータであり，従来のハードウェア・モニタを使用したシステム測定では，このシステム・プロフィールを求め，それをもとにシステム改良を行なうものが多い[14]．しかしながら，単にシステム・プロフィールだけからは，システムのアンバランスを知ることが出来ても，その原因をつかむことは困難な場合が多い．SYDASでは，システム・プロフィールをユーザ・プログラムの特性，OS特性と同様にシステム・プログラムの処理要求のイベント時系列を処理することにより，システム・プロフィールを求めているため，システムのアンバランスがある時に，その原因を究明することが比較的容易である．しかし，かなり頻繁に起こるイベントの時系列からこれを求めているために，測定データの量が膨大なものになることが欠点として残る．たとえば，割込みごとに5乃至10個のイベントが測定されるとし，平均割込間隔を10 msとすると，磁気テープ1巻で約30分～1時間程度の測定になる．

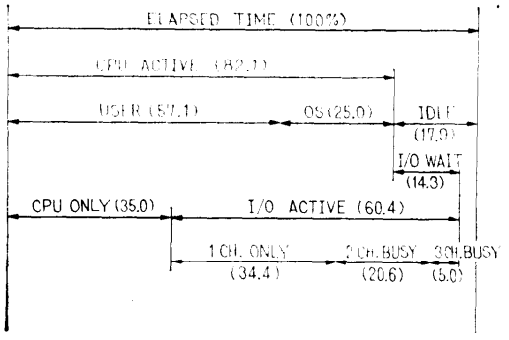

Fig. 6 System profile of a measured system

(2) 入出力チャネル使用率

入出力チャネルの割当てもシステム構成上の大きな問題であるが，各チャネルの使用率，平均I/O時間の測定を，各チャネル使用開始・終了のイベント時系列として行なっている．

現在SYDASを用いて以上に述べた各種の測定が行なわれており，その解析が行なわれてきているが，今後とも，各種の測定を行なって行きたい．

5. むすび

コンピュータ・システム測定のためのハードウェア・モニタ(SYDAS)およびそれを用いたシステム測定の例について述べた．システム測定は，まだ比較的歴史が浅いためもあって，その測定技法が確立されていない．特に，性能評価のための測定において，何をもってシステムの性能とし，その尺度が何であるのかさえ，明確にされていない．このような状況の下にあって，SYDASは何を測定すべきか，そしてそれをどのようにして測定すべきかを実験するための道具としての役割を果しうるように，多分に融通性に対する考慮がはらわれている．連想記憶装置を用いたイベント時系列の測定はその典型的な例であり，IC技術の発達により連想記憶装置が容易に手に入るようになったことが，この測定法を可能にした．

SYDASの開発に当って，困難を感じたのは被測定システムに挿入するプローブの問題であった．被測定システムに与える影響を最小にするために，電気的には高い入力インピーダンスが必要であり，物理的な形状は保守の妨げにならないよう十分小さいことが要求

され，かつ多数のシステム状態信号をとりだしうるために，価格的に安価であることが条件となる．今後のコンピューターの設計においては，あらかじめ測定を考慮して測定のための信号を取りだし易い形で用意しておくような配慮がはらわれるべきであろう．このことは，ハードウェア，ソフトウェアの両方について言えることである．

システム測定を通じて，われわれはいくつかのことを学んだ．一つは，システム測定は単に性能評価のためだけでなく，プログラム・デバッグに対しても，かなりの効果をあげうることである．システムにおけるプログラミング・コストの占める割合は，ますます増加する傾向にあり，プログラミングを容易にするための言語などが，いろいろと論議されているが，デバッグだけは今だに旧態依然のありさまである．現在のままのではプログラム・デバッグの道具として不十分であるが，多少の改良を加えることにより，かなり強力なものになると思われる．より完備したデバッグ用道具を作ることも，システムの一つの大きな課題である．

性能評価については，システムの性能を明確に定義し，そのパラメータを明らかにすることの必要性があらためて痛感された．システムの動作の時間的，空間的な構造を表現し，システム性能とその構造との関係を定めうるようなモデルが確立されれば，システム測定技法の確立は容易であろう．そのためのステップとして，現実のシステムの動作を適当に把握するための努力と，そのモデル化に対する研究が進められる必要がある．

謝 辞

おわりに，日本電気・中央研究所小高部長，鍵山研究マネージャー（現 NEC システム研究所）には日頃から適切な指導をいただいている．システムの基本機能の検討と結果の解析においては，三上研究スペシャリストをはじめ多くの方々から有益なご示唆をいただいた．SYDAS の設計，製作に当られた日本電気エンジニアリング伏見氏，連想記憶装置を担当された石十嵐良氏（現日本 IBM）および大野氏，さらに解析プログラムの作成に当られた平山氏，稲葉氏をはじめ沢山の方に労を費していただいた．システム測定に際しては経営情報システム本部矢板本部長，国分課長，斎藤主任をはじめたくさんの関係各位のご協力をいただいた．また，コンピュータ方式技術本部小林亮開発部長心得，清水氏にはハードウェア上の問題について，いろいろと御指導をうけた．ここに，これらの方方に深く感謝の意を表する次第である．

参 照 文 献

1) 渕一博ほか：ETSS の解析（その1）——データ収集・解析によるシステムの改良——，電気試験所彙報，Vol. 33, No. 12, pp. 53～74.
2) D. J. Campbell and W. J. Heffner: Measurement and Analysis of Large Operating Systems during System Development, Proc. FJCC 1968, pp. 903～914.
3) H. W. Cantrell and A. L. Ellison: Multiprogramming System Performance Measurement and Analysis, Proc. SJCC 1968, pp. 213～221.
4) K. W. Kolence: A Software View of Measurement Tools, Datamation, Vol. 17, No. 1, pp. 32～38, 1971.
5) F. D. Schulman: Hardware Measurement Device for IBM System/360 Time Sharing Evaluation, Proc. ACM Nat. Meeting 1967, pp. 103～109.
6) R. W. Murphy: The System Logic and Usage Recorder, Proc. FJCC 1969, pp. 219～229.
7) 石原孝一郎ほか：システムデータ測定器とその測定結果，昭44年電気四学会連合大会，No. 3173.
8) 富岡進ほか：汎用コンピュータ・パフォーマンス・アナライザ，情報処理学会第12回大会，No. 185, 1971.
9) 小野隆喜，箱崎勝也：システム・データ収集装置—SYDAS，昭46年電子通信学会全国大会，No. 1053.
10) 箱崎勝也・小野隆喜：ハードウェア・モニタ (SYDAS) によるシステム性能評価，情報処理学会第12回大会，No. 186, 1971.
11) G. Carlson: A User's View of Hardware Performance Monitors, Proc. IFIP 1971 pp. TA 5-128～132.
12) 三上徹ほか：コンピュータ・システム性能評価シミュレータ PACSS, 情報処理 Vol. 12, No. 1, pp. 14～25, 1971.
13) P. J. Denning: The Working Set Model for Program Behavior, Comm. ACM, Vol. 11, No. 5, pp. 323～333, 1968.
14) J. S. Cockrum: Interpretting the Result of a Hardware System Monitor, Proc. SJCC 1971, pp. 23～38.

(昭和47年3月25日受付)
(昭和47年4月22日再受付)