



エンドユーザ・システムにおける性能予測システム*

金子朝男** 西原義之** 阪田史郎**

Abstract

In this paper, the design concept of COMPAS, a performance prediction system on end-user computer systems, is proposed. COMPAS is designed mainly for application system designers, although it can be used by hardware designers or OS designers as a conventional performance evaluation tool. COMPAS users are classified on the basis of their objectives or levels, and are provided specialized COMPAS which is dedicated to each user class. The parametric description of the target system and the generation method which produces the specialized COMPAS enables users to simulate the end-user system closely, easily and speedily on their own host computer systems.

Finally, COMPAS/I which was developed to predict the performance of multi-terminal systems and some of its evaluation results are briefly described.

1. ま え が き

コンピュータ応用領域の拡大にともなって、コンピュータ利用者の階層化および利用業務の分業化が進行し、現在では計算機センタの汎用高性能コンピュータ・システムと同様にエンドユーザ・システム（データ発生場所で活動する自立したコンピュータ・システム）の重要性が認識されるに至っている。そしてエンドユーザ・システムは近年、コンピュータ・ネットワークを利用した複雑なシステムも構成するようになり、アプリケーション・システムの設計時において応答時間やスループットの予測、システムバランスの解析等が不可欠になろうとしている。しかしながら現在のところ、エンドユーザ・システムの性能を体系的に評価するシステムは皆無に等しい。本稿ではシミュレーションによりエンドユーザ・システムの性能を予測するシステム COMPAS (COMputer system Performance Analyzing System) の概念を提案し、その一実現システムである COMPAS/I を紹介し、測定結果例を示す。

エンドユーザ・システムは幅広い分野で使用され、

多彩な利用形態を有している。従って一システムに関する性能評価をもとに、他のシステムの性能を類推することがむずかしく、個々のシステムの利用形態に即した性能予測が必要となる。更に、エンドユーザ・システムの導入、稼働数は汎用の中・大型システムと比較するとはるかに大きく、性能予測を専門家のみに任せるとは困難である。故にエンドユーザ・システムの性能予測システムは一般のアプリケーション設計者が自由に使用でき、システムの利用形態に即して予測可能なことが必須条件となる。ところで従来の性能評価システムは OS、ハードウェアの設計者などコンピュータ・システムの専門家が、みずから利用するために作成したものが多く、使用する際に要求される知識、実行コンピュータ・システム、操作性、性能値の表現、測定時間などに関して容易に使えるシステムとはいえない。

本稿で提案する予測システムは、従来の評価システムには見られないジェネレーション方式、パラメータ・セレクション方式を採用し、個々の利用者にとって使い易いシステムを実現する。本システムの主要利用者はアプリケーション・システム設計者、主目的はアプリケーション・システムの設計であるが、利用者階層の分離と予測システムの各階層に対する適合化により、ハードウェア設計者からエンドユーザまで幅広い

* Performance Prediction System on End-user Computer Systems by Asao KANEKO, Yoshiyuki NISHIHARA and Shiroh SAKATA (Central Research Laboratories)

** 日本電気(株)中央研究所

利用者を対象とする。このため、デバイスの性能設定、既存システムの性能改善、システムの内部動作解析等の従来型測定に利用することも可能である。

2. 性能予測システム COMPAS の概念

2.1 システムに対する要請

最初に測定対象であるエンドユーザ・システムを考察する。利用形態は、個々のシステムは定型処理が主体であり固定的であるが、総体的に見ると様々な用途に使われているため多彩である。一方ハードウェア、OS 等の基本システムは、多重処理、ネットワーク処理などの機能を有するものが数多く出現しているが、アプリケーションに比較すると多様性は少ない。OS は主記憶容量が少ないために単純な構造をとり、非常駐部分が大きい。また利用者とのインターフェースが密であり、操作性、特に応答時間が重要である。エンドユーザ・システムの性能予測システムはこれらの特徴を考慮したものでなければならない。

予測システムを利用面から考察するとき、エンドユーザ・システムは導入、稼働数が多いこと、複雑な構成をとるシステムではアプリケーションの性能に与える影響が大きく、システムごとの性能予測が必要であることから、アプリケーション設計者が直接利用できることが必須となる。このことから性能予測システムに要求される条件として以下の項目が導かれる。

- (i) 事務処理部門における販売・在庫管理、技術者の数値解析、図書館の情報検索など多彩なアプリケーション・システムを対象とする。
- (ii) 測定対象となるハードウェア、OS の内部構造を詳細に知らなくても対象をモデル化できる。
- (iii) 予測のために特別な道具、装置を必要とせず、使用法が簡単である。
- (iv) 性能を具体的に予測し、個々の利用者にとって必要な情報を、理解しやすい形で出力する。
- (v) 短時間で結果が得られる。

2.2 実現方式

従来からシステムの性能測定方式としてシミュレーション^{1),2)}、数学的解析³⁾、ハードウェア・モニタリング⁴⁾、ソフトウェア・モニタリング、テストプログラムおよびそれらの複合方式⁵⁾があり、それぞれ利害得失がある⁶⁾。COMPAS は主目的が性能予測であり、シミュレーション方式を採用する。シミュレーション方式は測定時間の点で問題があるが、アプリケーション

を任意の詳細度で記述することができ、エンドユーザ・システムの測定に最適な方式と考えられる。

性能予測システムの利用者はアプリケーション設計者、OS、ハードウェア設計者などであり、おのおの評価項目、評価方法が異なる。OS、ハードウェア設計者は、OS の構造、個々の周辺端末装置の性能がシステム全体に与える影響などを評価する。アプリケーション設計者の評価はより具体的であり、基本システム上に、あるアプリケーションを構築するとき、単位時間あたりの処理量、一回の応答時間などを評価する。ハードウェア、OS など基本システムの性能、機能は、すべての範囲をとるものでなく、あらかじめ定められた中から選択される。したがってアプリケーション設計者が対象システムの性能を予測する際、アプリケーション部分は詳細に、基本システム部分は簡単な記述のみで対象をモデル化できることが望ましい。COMPAS は基本システムをパラメータによる記述形式でモデル化し、利用者階層によってパラメータの範囲、種類を変える。これを実現するには予測システムの提供方式が重要となるが、COMPAS では利用者と提供者を分離し、提供者側にジェネレーション・フェーズを設ける。提供者は個々の利用者の要請をもとにパラメータを選択し、このパラメータ、及びパラメータに対応する基本システムを持つ COMPAS を生成し、利用者へ提供する。

ジェネレーション方式は幅広い測定対象、豊富な機能を有する COMPAS から利用者の要請する範囲内での多様性を持つ専用システムを生成する方式であり、本システムではこれをコンピュータを使用して行う。ジェネレーション方式は従来型シミュレーションには見られない方式であり以下に示す特徴を有する。

ジェネレーション方式の利点

- (i) パラメータ・セレクションと利用者階層への適合化を可能にする。
COMPAS の有する豊富なパラメータを利用者層に応じて捨捨選択、既定値の決定を行い、使いやすい予測システムを幅広い利用者層へ提供する。
- (ii) エンドユーザ・システム上での実行を可能とする。利用者へ提供する COMPAS をコンパクト化し、利用者が使い慣れているエンドユーザ・システム上でシミュレーションを実行することを可能とする。
- (iii) シミュレーション時間の短縮

統計項目に合わせて中間データ（統計のためにシミュレータが出力するデータ）を選択し、シミュレーション時間の短縮をはかる。

2.3 実行フェーズおよび利用手順

COMPAS 構成フェーズを提供者側でジェネレータ、コンパイラの2フェーズ、利用者側でトランスレータ、シミュレータ、エディタの3フェーズとする。

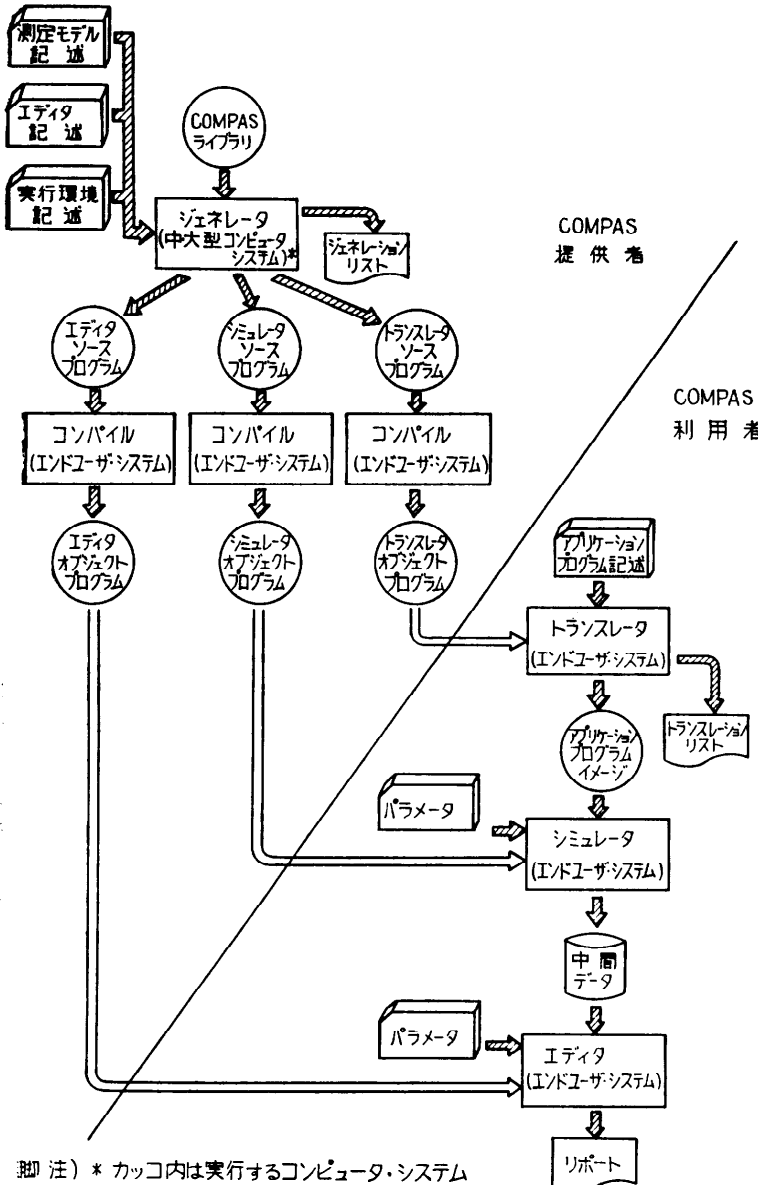
Fig. 1 に COMPAS 利用手順を示す。

利用者は予測対象、測定項目、シミュレーション実行システムなどを定め、提供者に COMPAS 生成を依頼する。この時、利用者はシステム構成、OS 実現方式などで、部分的に変化させて測定する必要がある項目を、シミュレーション実行時のパラメータとして指定する。

提供者は利用者の要請にもとづき COMPAS を専用化する。ジェネレータは予測対象、測定項目、実行

環境などの記述を解釈し、COMPAS ライブラリより対応するモジュールを取り出しシミュレーション実行システムのソースプログラムを生成する。例えば OS 部分はライブラリにエンドユーザ・システムで利用されている種々の記憶管理方式、データ管理方式を持ち、測定モデル記述の内容に従って、これらの管理方式を組み合わせ、対象システムの OS をモデル化する。記憶管理方式をパラメータとする利用者に対しては、複数個の方式をシミュレートするプログラムを組み込み、シミュレーション時に選択可能とする。本フェーズは中・大型コンピュータ・システム上で効率良く実行する。コンパイラはシミュレーションを行うシステムと同等なシステムの上でソースプログラムをオブジェクトプログラムに変換する。

COMPAS 利用者はまずアプリケーション記述言語を使用して、対象システムのアプリケーション・プログラムを記述する。トランスレータはこのアプリケーション・プログラム記述をシミュレータが解釈できる形式に変換する。トランスレータはシミュレーション実行時のメモリ上の制限、解釈効率上の制約から生じるアプリケーション・プログラム記述方式の制約を



脚注) * カッコ内は実行するコンピュータ・システム

Fig. 1 COMPAS Utilization Flow

解消し、利用者の記述を容易にする。シミュレータは、パラメータ、COMPAS 内部に組込まれた情報、アプリケーション・プログラム記述をもとに、測定対象たるエンドユーザをシミュレートする。本フェーズでのパラメータを変化させることにより、システムの一部を変更するときの性能変化を簡単に予測することができる。エディタは中間データを解析し、指定された測定項目に関する性能値を利用者に理解しやすい形で出力する。

2.1 で述べたようにエンドユーザ・システムにおいては、各システム（機種）の基本システム部分の選択要素は少ない。従ってアプリケーション設計者が使用する場合、その選択要素全体をパラメータとすればシミュレーション実行時にアプリケーション・システムのモデル化が可能となり、アプリケーション設計者は一度 COMPAS を入手した後は、エディタ以降の 3 フェーズのみで種々のアプリケーション・システムの予測が可能となる。またアプリケーションを設計しているエンドユーザ・コンピュータ上でシミュレーションを実行することができるため、予測のために他のコンピュータ・システムの利用法を覚えるなどの余分な労力が不要であり、簡便な予測システムといえる。

2.4 測定モデル記述

予測システムが必要とする情報は測定対象となるシステムに関する情報、測定項目およびその表現形式に関する情報、シミュレーションを実行するコンピュータ・システムに関する情報の三種類がある、Fig. 2 に示すように COMPAS ではこれらの情報を三つの部分に分けて独立に記述する。記述形式はアプリケーション・プログラムは実行の流れを示すプロシージャ形式とし、他の情報はパラメータ形式とする。情報を入力するフェーズは利用目的に合わせて可変とする。例えばアプリケーション設計用の COMPAS では基本システムに関する情報はジェネレーション時に与えて固定化し、アプリケーション・システムに関する情報をシミュレーション時の入力とし、利用者が自由に指定できるものとする（パラメータ・セレクション）。また OS 設計用の COMPAS では種々の OS 実現方式をシミュレーション時のパラメータとし、利用者が様々な方式を組み合わせた OS を測定することを可能とする。

次に各記述部を説明する。エディタ記述部はレポートの出力項目、出力形式を記述し、COMPAS 利用者の欲する評価項目が利用者に理解しやすい形で表現さ

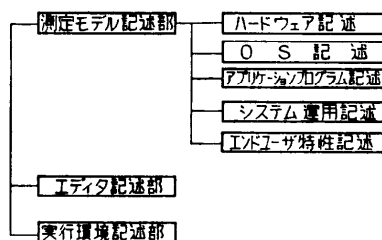


Fig. 2 COMPAS Parameter Structure

れるよう指定する。実行環境記述部はシミュレーションを実行するエンドユーザ・システムの主記憶サイズ、周辺機器構成、被シミュレート時間を指定する。両記述部のパラメータ例を Table 4 に示す。

測定モデル記述部は 5 個の記述に分かれる。ハードウェア記述は測定対象となるシステムに関するハードウェアの性能を記述する。OS 記述は OS で採用する種々の管理方式、オーバヘッドを指定する。アプリケーション・プログラム記述は、プログラムの大きさ、リエントラントか否かななどのプログラム特性とプログラムの実行の流れを記述する。実行の流れは、特別の言語を使用し、プログラムごとにスタティックな流れを記述する。システム運用記述はアプリケーションを実行する際のシステム運用状態を記述する。エンドユーザ特性記述は操作速度、思考時間などを指定する。エンドユーザ・システムでは利用者とのインタフェースが密であり、熟練度の幅が大きいため、これらの特性のシステム性能に与える影響が大きく、本記述が重要になる。測定モデル・パラメータの具体例を Table 2 に示す。

2.5 精度

性能の予測値と真の値との誤差の許容限界をいかに設定するかは測定システムの目的により異なる。新システム設計時の性能予測は未決定事項が多く予測は粗くなる。アプリケーション・システムの設計では基本システムは定まっておらず比較的高い精度が得られる⁷⁾。精度はまた測定方式により大きく異なるが、シミュレーション方式の場合は比較的高い。

シミュレーション方式における誤差の原因は、シミュレータが計上する時間の誤り、イベント発生順序のずれ、対象システムに対する忠実性の粗さなどである。特に人間の操作速度は一様でなく、オペレータが重要な役割をはたすアプリケーションでは誤差が大きくなりやすい。またオンライン処理ではデータ転送に起因するエラー率の相違から予測値がずれることが多い。COMPAS では操作速度、データ転送時間などを

シミュレーション実行時のパラメータとし、この値を実際の値に近づけることにより誤差を減少させる。

2.6 COMPAS と他の性能予測システムとの比較

Table 1 に COMPAS と他の方法による性能予測の比較を示す。この中で PACSS¹⁾ は汎用の OS 性能評価システムである。表が示すように COMPAS は測定対象をエンドユーザ・システムに絞ることにより使い易さを追求したシステムであるといえる。

3. 多端末システムの性能予測システム
COMPAS/I

COMPAS/I は COMPAS のサブセットとして開発され、エンドユーザ・システムの中で多端末システムを主要対象とする性能予測システムである。

〔測定モデル記述〕

Table 2 に測定モデル記述のパラメータを示す。

COMPAS/I 利用者はこの中から目的に合わせて選択し (Table 2 の P に相当)、専用化されたものを使用

する (パラメータ・セレクション)。アプリケーション・プログラムは端末、バッチ用のものをそれぞれ作成し、タスクに対応づける。アプリケーション・プログラムの記述は Table 3 に示すコマンドを使用して表現する。この表からわかるように通常のプログラムと同様に記述するが、一命令の機能が大きく、CPU 処

Table 1 Comparison of Performance Prediction System

	COMPASS	PACSS*	数学的解析
利用者の範囲	◎	△	△
利用の容易性	◎	△	○
アプリケーション記述の 詳細度	◎	○	△
精 度	◎	◎	△
測定項目の多様性	○	◎	△
測定時間	○	△	◎
測定対象の範囲	△(◎*)	○	◎

◎は秀れている ○は普通 △は劣る

* Program and Computer System Simulator¹⁾

** 対象をエンドユーザ・システムに限定する場合

Table 2 Model Description Parameters of COMPAS/I

記 述 部	パ ラ メ ー タ	パラメータ・セレクションの例*	
		ジェネレーション 時の指定	シミュレーション 時の指定
ハードウェア	<ul style="list-style-type: none"> ・CPU 性能 ・補助記憶装置性能 ・周辺端末装置性能 ・回 線 速 度 	NEAC システム 100 の値を指定	—
OS	記 憶 管 理 <ul style="list-style-type: none"> ・管理方式 ・パーティションサイズ ・パーティション数 ・ページサイズ ・OS 非常駐領域単位 ・OS 非常駐領域サイズ ・OS 常駐領域サイズ 	パーティション方式	—
		P**	12 kB
		P	1
		—	—
		3 kB	—
OS 非常駐領域単位	6 kB	—	
OS 非常駐領域サイズ	6 kB	—	
OS 常駐領域サイズ	6 kB	—	
タ ス ク 管 理	<ul style="list-style-type: none"> ・タスク変換方式 ・カンタムタイム ・タスク切替プライオリティ ・最大タスク数 	低速 I/O 切替時	—
		—	—
		ラウンドロビン	—
		25	—
デ ー タ 管 理	<ul style="list-style-type: none"> ・データ処理オーバーヘッド ・ファイル保護 	システム 100 の値	—
		ディスクファイル	—
デバイス管理	・デバイスの共有	CMT, Disk	—
プログラム管理	・連続実行	—	—
アプリケーションプログラム	<ul style="list-style-type: none"> ・プログラムサイズ ・プログラム優先度 ・リエントラントプログラム有無 	P	9 kB
		P	プログラムごとに指定
		P	無 (例 1), 有 (例 2)
システム運用	<ul style="list-style-type: none"> ・主記憶サイズ ・ディスク台数 ・稼動端末台数 	P	24 kB
		P	1 ユニット
		P	1~25 台
エンドユーザ, 特性	<ul style="list-style-type: none"> ・ユーザ思考時間 ・キーボード打鍵速度 	P	プログラム内で指定
		P	5 文字/秒

注) * 4. の測定例で例用する COMPAS/I の値。

** シミュレーション・フェーズのパラメータとする。以下の P も同様。

```

BILLING PROG
DEF RANDOM#RN1,RN2
%STATION=1
LBL90 STRT 0
      CPU 100STP
      CALC #1#*1
      HBR LBL10,R#1,GT,95 NO. OF ISSUED BILL
      LIO STATION,GET,60CHAR,1SEC CUSTOMER CODE
      BR LBL20
LBL10 CALC #2#*2*1 SET SHEET ROLL
      LIO STATION,GET,60CHAR,10SEC CUSTOMER CODE
      LBL20 LIO DISK,INDEXGET,100CHAR CUSTOMER FILE
      CPU 200STP
      LIO STATION,PUT,20CHAR
      BR LBL25
LBL25 CALC #2#*2*1
      LIO STATION,PUT,37CHAR
      LIO STATION,PUT,30CHAR
      CALC LINE#RN2
      HBR LBL40,LINE,GT,70
      HBR LBL50,LINE,GT,30
LBL30 CALC #3#3
      BR LBL40
LBL40 CALC #3#5
      BR LBL60
LBL50 CALC #3#7
      CPU 100STP
      LIO STATION,GET,10CHAR,1SEC PRODUCTS CODE
      LIO DISK,INDEXGET,80CHAR PRODUCTS FILE
      CALC #4#*1
      HBR LBL80,#4,GT,8
      LBL70 LIO CMT,PUT,0CHAR
      BR LBL90
LBL80 LIO CMT,PUT,80CHAR,1SEC DEFER CMT JOURNAL
    
```

Fig. 3 An Example of Application Program Description

理, 入出力処理とマクロに表現する点が異なる。また乱数やパラメータを定義し, 確率的に実行の流れを変えたり, シミュレーション実行時にループ回数を指定することが可能である。アプリケーション側からの統計機能として, 統計コマンド, カウンタを有し, あるまとまった処理 (例えば 1 枚の伝票発行) の実行回数や実行時間の分布などを調べることができる。Fig. 4 にアプリケーション・プログラムのフローチャート, Fig. 3 にコマンドの記述例を示す。

エディタ記述部, 実行環境記述部のパラメータを Table 4 (次頁参照) に示す。エディタの出力するリポ

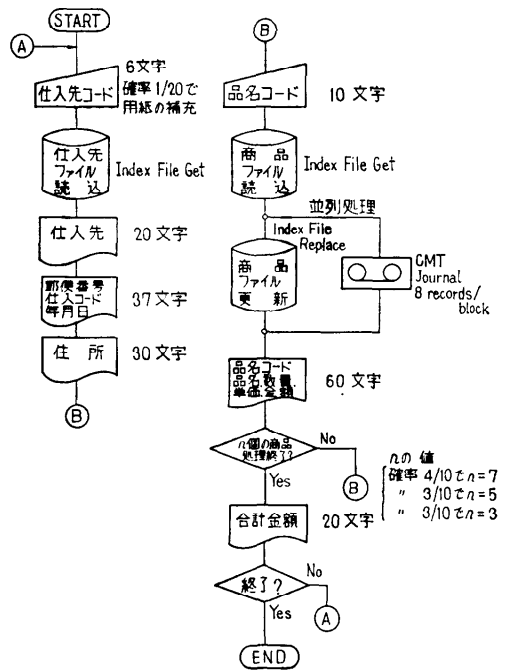


Fig. 4 An Example of Application Program Flow

ートは CPU 使用率, 全端末の平均応答時間などは基本項目とし毎回出力する。それ以外に端末別の処理量, アプリケーション・プログラム記述で指定した統計データが必要とき, パラメータを指定する。

(シミュレータ)

シミュレータの概要図を Fig. 5(次頁参照)に示す。

Table 3 COMPAS/I Application Description Commands

コマンド種類	表現形式	指定項目	機能
宣言コマンド	PROG	プログラム名	プログラム宣言する
	DEF RANDOM	乱数名	乱数を宣言する
	DEF PARAMETER	パラメータ名	パラメータを宣言する
ステップ記述コマンド	CPU	ステップ数	CPU の実行
	LIO	デバイス名, アクセス形式, 入出力文字数, 端末思考時間, I/O完了待ち合わせ方法	ロジカル I/O
	WAIT	デバイス名, アクセス形式	I/O完了待ち合わせのためのマクロ
	BR	レーベル名	無条件分岐
	HBR	レーベル名, 文字列 1, GT, 文字列 2	文字列 1 が文字列 2 よりも値として大きい時分岐
統計用コマンド	STRT	数 字	STRT, END は対として働き, スタート・ポイントからエンド・ポイントまでの実行回数, 通過時間の分布などを調べる。
	END	数 字	
演算用コマンド	CALC	文字列 3# 文字列 1・演算記号・文字列 2	文字列 1 と文字列 2 の値に演算をほどこし, 文字列 3 の値とする。
連続処理コマンド	NEXT	プログラム名	バッチ, 端末で次に実行するプログラム名を指定する。

注) HBR, CALC の文字列は { 数 字
利用者定義語
@数字(カウンタ) } のいずれかである。

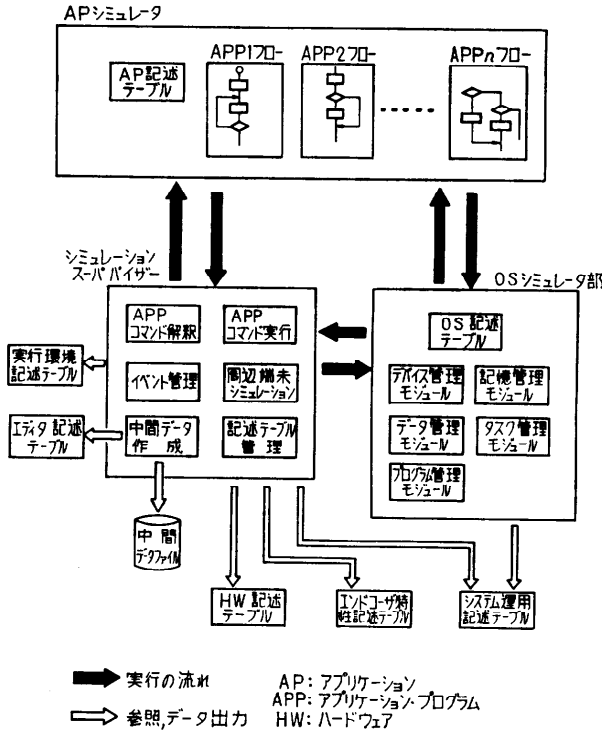


Fig. 5 COMPAS/I Simulator Diagram

Table 4 Editor and Environment Description of COMPAS/I

記述部	内 容	
エディタ記述部	応答時間	・応答時間
	スループット	・アプリケーション・プログラムのポイント通過回数, ポイント間通過時間
	使用率	・CPU, 周辺装置使用率, 使用回数
	基本システム情報	・CPU 待行列 ・CPU アイドル時間 ・非常駐モジュールのスワップ回数
出力形式	・平均	
	・分散	
実行環境	実行システム	・システム構成 ・主記憶サイズ ・リポート出力デバイス ・中間データファイル
	被シミュレート時間	・被シミュレート時間

まずパラメータを入力し、各記述テーブルにセットする。シミュレーションは時刻0から開始し、OSの各コンポーネント、アプリケーション・プログラムのCPUコマンドを通過するごとに指定された時間を加え時刻を進行させる。OS、アプリケーション・プロ

ラムの流れを進める過程において、タスク変換、OS非常駐領域のスワップ等のイベントが発生すると、イベント種類、発生時刻よりなる中間データを出力する。コマンド解釈モジュールで監視されるシミュレーション時刻が指定時間以上になると実行を終了する。

〔測定時間および精度〕

シミュレータ、エディタの実行に要する時間の合計を測定時間と呼ぶとき、測定時間と被シミュレート時間の比率は、主として端末台数とともに増加する。それは端末台数が多いとイベント発生頻度が高くなり中間データの出入力時間が増加するからである。4.の例1では端末3台で1:2, 7台で1:1, 15台で2:1である。これはエンドユーザ・システム上での実行にもかかわらず従来型シミュレーションと比較して良い値である。この理由として、エンドユーザ・システムではOSの制御方式が簡単のため、イベントの発生回数が少ない。入出力速度が遅い、ジェネレーション方式によりプログラムの主記憶と二次記憶とのスワップが少ない、などが挙げられる。

また4.の測定例ではシミュレーションが定常状態に達するまでの初期時間が2~3分、被シミュレート時間20分で実測との誤差が10%以下という結果を得ている。

4. COMPAS/I による測定例

本章ではCOMPAS/Iを使用した性能予測の例を示す。

〔設定モデル〕

多端末システムは小型コンピュータをベースとしたものが多く、その適用分野は多種多様である。本測定では多端末システムの中でも特に代表的なアプリケーション・プログラムとしてFig.4に示す伝票発行業務を取り上げる。その他対象としたモデルの測定条件をまとめてTable2に示す。

〔測定結果〕

(1) 端末台数と仕事量⁹⁾

多端末システムを利用するエンドユーザがシステムの導入あるいは業務の拡大に当って最も必要とする情報の1つは、何台の端末を使用すれば、どの程度の仕事量が可能であるかである。Table5に3台、5台、7台の端末を使用した場合の性能比較結果を示す。こ

Table 5 Performance of Multiterminal Systems

測定項目	3台	5台	7台
伝票発行枚数	169(枚/時間)	267	374
平均応答時間	1.06(秒)	1.12	1.38
90%の時間で保証される応答時間	1.9(秒)	1.9	2.1
バッチ業務の処理量*	50.7(%)	27.3	14.0

* バッチ業務は製表作業とし、その処理量はプリンタへの印字行数であり、バッチ業務のみを実行した場合との比で表わした。

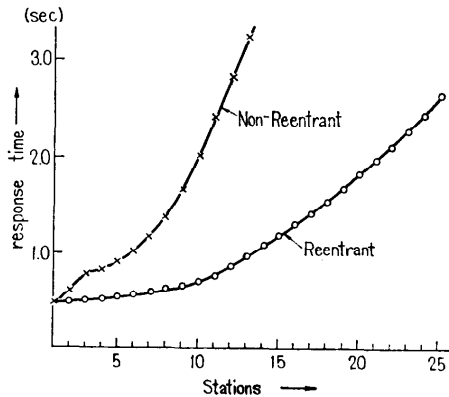


Fig. 6 Average Response time

れによると、5台(実働4時間/日)の端末を使用すれば、1日あたり約1,000枚の伝票が発行できる。

(例2) リエントラント化による効果⁹⁾

マルチプログラミングによる多端末制御を行う場合、アプリケーション・プログラムのリエントラント化を可能にするか否かはシステムの設計上重要な問題である。リエントラント化により端末間でのプログラム選択の自由度は減少するが、システムの性能は向上する。Fig. 6に各端末台数におけるリエントラント化の影響度を示す。Fig. 6によれば、端末台数が10台以上の時リエントラント化による効果が顕著になる。

5. むすび

本稿ではエンドユーザ・システムを対象とした性能予測シミュレータCOMPASを提案し、その第一ステップとして開発されたCOMPAS/Iの概要と、それを用いた測定結果について述べた。

従来の性能評価システムは、システム設計者用に使われており、一般のエンドユーザが直接利用することは不可能に近い。また測定対象のモデル化についてもプロシージャ型の記述が多く、汎用性には富むが指定方法が複雑であり多大なシミュレーション時間を必要としていた。

COMPASでは、アプリケーションがエンドユーザにも簡単に理解できる処理の流れとして表現され、またジェネレーション方式の採用により測定に必要なメモリサイズや時間が節約でき、また高度にレベル化されたパラメータ方式を採用したことにより利用者のレベルに応じた簡単な利用法を可能とした。すなわち、COMPASは一般的な利用者が、日常使用しているコンピュータを利用して、簡単に性能予測を行えるシステムといえる。

今後、COMPASの概念に基づいた多くの性能予測システムが出現し、コンピュータシステムの開発時、ユーザへの導入時、およびユーザシステムの拡張時に利用されるようになるであろう。

最後に本研究の機会を与えて頂いた日本電気藤野部長、祢津課長、有益な御助言を頂いた村山芳朗氏、服部光宏氏、ならびに本システムの開発に当って多大な御援助を頂いたNEACシステム100関係者各位に深謝いたします。

参考文献

- 1) 三上他: 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)
- 2) 川合英俊: シミュレーションによるシステム評価法, 情報処理, Vol. 13, No. 11, pp. 752~758 (1972)
- 3) 村岡洋一: 階層構造を持つシステム性能評価モデル, 情報処理, Vol. 16, No. 5, pp. 426~433 (1975)
- 4) 箱崎, 小野: ハードウェア・モニタの技法について, 情報処理, システム性能評価研究会 74-1 (1974)
- 5) S. R. Kimbleton: A heuristic approach to computer systems performance improvement, I-A fast performance prediction tool, NCC, pp. 839~846 (1975)
- 6) H. C. Lucas: Performance Evaluation and Monitoring, Computing Surveys, Vol. 3, No. 3, pp. 79~91 (Sept. 1971)
- 7) T. E. Bell: Objectives and problems in simulating computers, FJCC, pp. 287~296 (1972)
- 8) 金子他: 事務用小型計算機における実時間業務の性能について, 49年情処大会, pp. 197~198 (1974)
- 9) 金子他: 小型コンピュータにおける主記憶サイズについて, 50年情処大会, pp. 63~64 (1975)
- 10) 阪田他: マルチターミナルシステムにおける最適印字速度について, 50年情処大会, pp. 43~44 (1975)

(昭和51年3月18日受付)
(昭和51年9月20日再受付)