

# VM/4の制御プログラムの構成と性能について

大野直哉<sup>\*</sup>, 磯根秋義<sup>\*\*</sup>, 大嶋隆<sup>\*\*</sup>, 山岸裕治<sup>\*\*</sup>, 難波信治<sup>\*</sup>, 久保秀士<sup>\*</sup>  
 (日本電気<sup>\*</sup> CQCシステム研究所, <sup>\*\*</sup>基本ソフト開発本部)

## 1. まえがき

VM/4<sup>1)</sup>は、ACOS-4系マシン上で同じACOS-4系マシンを仮想計算機(VM)として複数台設定し、そのうえでのオペレーティングシステムACOS-4<sup>2)</sup>の運転を可能とする仮想計算機機能であり、特に高速化に重点をおいて開発を行った。即ち、VM/4においては、バヤマシン(BM)上のOSのもとで直接一般のユーザプログラムをVMのオーバヘッドなしに高速に実行することを可能にし、さらにVM/4により実現されるVM上においても、主記憶の常驻化機能、専用デバイスに対する処理の高速化のためのVMモードE/O機能、仮想プロセッサディスパッチャのファームウェアによる実現等、強力なハード/ファームサポートにより高速化を達成している。

本稿では、まず仮想計算機に対して直接にインタフェースをもち、種々の機能を提供するVM/4制御プログラムに関して、外部機能、内部制御方式について述べ、次にVM/4の性能に関して、そのオーバヘッドの要因、性能の実測結果について述べる。なお、VM/4は、現在ACOSシステム450上にインプリメントされ、社内でOS開発用に使われている。

## 2. VM/4制御プログラム

### 2.1. VM/4の構造

VM/4は、オ1図に示すように、統合型仮想計算機<sup>3)</sup>の形態をとっている。即ち、VM/4制御プログラムは、実計算機(BM)上で動作するACOS4/MVPのもとで、一般のユーザプログラムと同じく、一個のジョブとして動作する。このVM/4制御プログラムにより、複数個のVMが実現され、各VM上では、ACOS4/MVPまたは、ACOS4を動かすことができる。また、一般のジョブはVMのオーバヘッドなしに動作可能である。

VM/4制御プログラムは、オ2図に示すように、VM全体に関連する処理を行うための1個の主タスクおよび、各VM個々の処理を行うためにVM対応に用意されたサブタスクより構成されている。主タスクおよびサブタスクには、各々セマフォが対応アけられており、このセマフォを介して、主タスクとサブタスクの通信、および、マシンから各タスクへの各種事象の通知が行われる。各タスクの主な起動の契機をオ2図に示す。なお、VM全体に関する制御テーブルは制御ジョブ内の全タスクからアクセスできる共有アドレス空間におかれ、VM個々の制御テーブルはVM対応のサブタスク固有のアドレス空間におかれる。

### 2.2. VM/4の外部機能

VMの實現: VM/4においては、BM上で動くOS(以下OS/BMと記す)の配下で動く制御ジョブ以外の一般のジョブをまとめて、VMとし、これをシステムにおける他のVMと対等の立場で制御することができる。

仮想資源とその實現: VM/4においては、VM上の種々の資源を次のように實現している。

主記憶: VM起動時に、VM対応の子タスクのアドレス空間上の1個のセグ

メントを割当て、実主記憶の上位における連続常駐エリアとしてこのVMに確保する。VM上の仮想アドレスから実マシンEの実アドレスへの二重アドレス変換は、ハード/ファームにより行われる。これにより、常駐型のVMを複数個実現でき、かつ、主記憶へのアクセスに関するVMのためのオーバヘッドをなくすることが可能となる。なお、現時点においては、常駐型のVMのみをサポートしている。

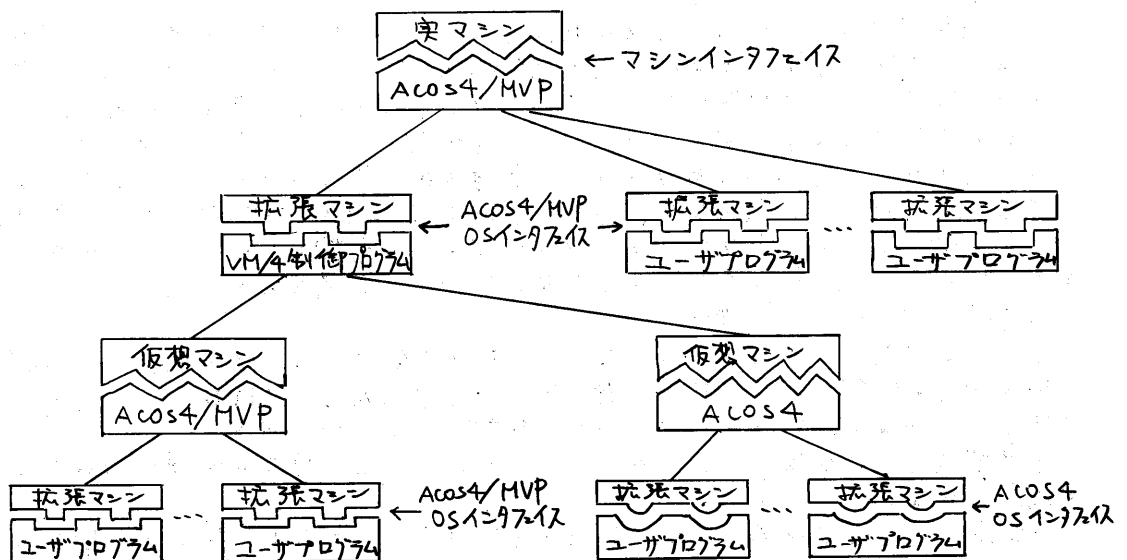
CPU: VMのもつプロセッサに対応して仮想プロセッサ制御ブロックを作成し、これに値を設定することにより、仮想プロセッサの実プロセッサへの割当てに関するプライオリティ、タイムスライスを指定できる。現時点では、プロセッサは1台、タイムスライス機能は働かせずに運用している。

コンソール: 現時点では、VM用にも、実際のコンソールを専用に割当てて使用している。操作パネル上のボタン類(イニシャライズボタン等)はコマンドにより代用している。

入出力機器: 原則として、各VMは、VM上のデバイスと同一タイプの実装置を専有して使用する。専有された装置に対しては、VMからの高速な処理(VMモードE/O)をサポートする。カードリーダー、LP等については、専有使用以外にスプーリング機能も提供する。回線についても、回線を単位にVMに専有させ、VMモードE/Oが適用できる。

VMモードE/Oは、制御プログラムにおいて実チャンネル番号、VM番号、VMにより作られたチャンネルプログラム、および、E/O完了時のメッセージの通知先のセマフォを指定してE/Oを起動することにより、IOCが指定されたVM上のアドレスを実主記憶アドレスに変換しながらE/Oを実行する機能で、これにより、制御プログラムにおいては、VM上のアドレスのBM上アドレスへの変換等のチャンネルプログラムの再構成処理が不要になり、E/O動作のシミュレーションを極めて高速に行うことが可能である。

図1 図 VM/4の基本構造



VM/4に対する操作:

VM/4機能は、BM上のOS(ACOS 4/MVP)上で、VM/4制御プログラムを起動することにより活性化される。このためには、VM/4制御プログラム実行用のジョブを他の一般の直属ジョブと同様の方法で入力すればよい。

活性化されたVM/4に対して、VM利用者はBMのコンソルから、次に示すような一般形式をコマンドを入力することにより、VMの起動、装置の確保等の指示を行う。

オ1表 VM/4サブコマンド一覧

サブコマンド	機能
A	VMを起動する
O	VMを終了する
I	VMをシステムイニシャリスする
R	VMをシステムリセットする
E	システムリスタートする
L	OS/VMをストレージロードする
G	実装置を確保してVMに与える
P	実装置を解放する
D	各種の状態を表示する
END	VM/4を終了(非活性化)する

LV VM識別名 サブコマンド パラメータの並び

オ1表にサブコマンドとその機能の概略を示す。

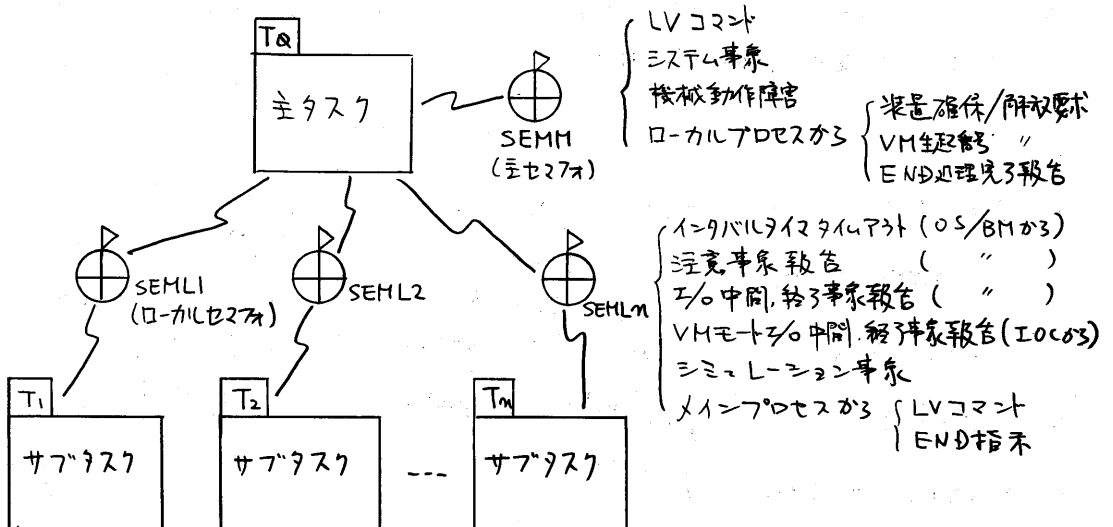
2.3 制御方式

ここではまず、VM/4機能が起動され、VMが起動、実行、終了し、VM/4機能が終了するまでの実行制御の概要について、特にファームウェア機構との関係に着目して述べ、次に、VMモードI/Oの制御の概要について述べる。

① VMの実行制御

- VM/4の起動: VM/4制御プログラムがジョブとして投入され、起動されると制御プログラムは、各種制御テーブルを初期化しVM/4機構を活性化する。
- VMの起動: "A" サブコマンドが投入されると、VMの構成表に基づく制御テーブルの作成、VM実行制御用の制御テーブルの作成、VMのための常駐主記憶領域の確保、VM/4ファームウェアに対するVMの登録を行う。

オ2図 VM/4制御プログラムの構造



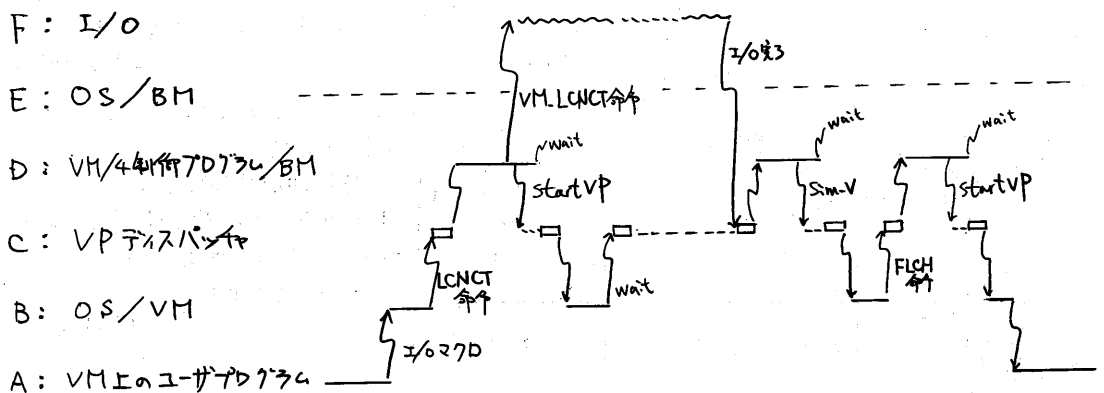
- ISLの起動: "L" サブコマンドが投入されると、VMのISLプログラムをVMの主記憶上に設定し、ファームウェアに対して、VMの実行を指示する。これにより、VM上でのISLプログラムの実行が開始される。
- ソフトウェアシミュレーション事象発生: VM上で、シミュレーション命令が実行されると、仮想プロセッサは停止され、セマフォを介して、VM/4制御プログラムに制御が渡される。制御プログラムでの要求されたシミュレーションが完了すると、仮想プロセッサ (VP) のスタートをファームウェアに指示する。VPデバイスバッファがこのVPに実行プロセッサを割り当てることにより、VPの実行が再開される。この繰り返しの繰り返しにより、VMが実行される。
- システムリセットの要求: "R" サブコマンドが投入されると、VMの停止をファームウェアに通知する。
- VMの終了: "O" サブコマンドが投入されると、VMの主記憶を解放した後、ファームウェアにVMの登録の抹消を指示する。
- VM/4の終了: "END" サブコマンドが投入されると、VM/4機構を非活性化したうえでVM/4ジョブは終了する。

(2) VMモードE/Oの制御

VM上のOS (OS/VM) が入出力指令を実行した場合の制御は次のように行われる。なお、\*3図にその制御の流れを示す。

- OS/VMが入出力命令 (CNCT/LCNCT) を実行すると、VM/4ファームウェアがこれを検出し、VM/4制御プログラムに通知する。
- 制御プログラムは、VMのE/O制御ブロックを参照して、E/Oの妥当性をチェックし、仮想チャンネルの状態を "ACTIVE" に設定する。
- 対応する実装置がVMで専有している装置ならば、VMモードE/Oにより、VM上で作成されたチャンネルプログラムそのものの実行をI/OCに要求する。
- I/OCは、E/Oを完了すると、完了事象をセマフォを介して制御プログラムに通知する。
- 制御プログラムはこれをうけて、BMのE/O制御ブロックから対応するVM制御ブロック内の仮想チャンネルを見出し、この状態を "AVAILABLE" にするとともに、Simulated-V.op指令を出したことにより、VMにE/O終了事象を報告する。

\*3図 代表的な制御の流れ



### 3. VM/4の性能

#### 3.1 VM/4における性能に関する考慮

一般に、仮想計算機システムにおいては、仮想計算機で想定している種々の資源を、仮想的に実現するためのサポートオーバヘッドが必要となる。<sup>4)</sup> VM/4においては、このようなVMのオーバヘッドなしで直接親OS上で動く一般ジョブを実行できるVM&を設けているとともに、VM/4で実現されるVMについてもVM機能のサポートのオーバヘッドを最小限にするために、ハードウェア：H，ファームウェア：F，ソフトウェア：S，および運用：Oも含めた観点から、次のような考慮を施している。

- ・ファームウェアによるVPディスクパックヤの実現(F)
- ・VMのための主記憶セグメントの実主記憶上へのはりつけと、TLB，リロケーション用RAMによる高速二重アドレス変換の実現(H, S, O)
- ・TLBのキーとして、VM識別を付加することによるTLBヒット率向上(H)
- ・VMモードI/Oのサポート(H, F, S, O)
- ・親マシン介入の頻度の削減：必要最小限の処理のみシミュレーション(F)
- ・シミュレーションのサポートのための命令の新設(F)

#### 3.2 VM/4におけるオーバヘッド要因

ここでは、VM/4を実現するための処理時間のオーバヘッドに着目し、その要因を調べる。オ3図に、VM上でI/O要求が発生し、これにより実マシン上の実I/Oが起動され、I/O動作が完了し、これがVMに反映されるまでの代表的な一連の制御の流れを示す。

- A: VM上でのユーザプログラム部での処理
  - B: VM上のユーザプログラムをサポートするためのOS/VMでの処理
  - F: 実I/O処理
- は、実マシン上で動かす場合にも必要な部分であるのに対して、
- C: BMとVMとの切換のためのVPディスクパックヤ部での処理
  - D: VM上での特権命令(I/O命令)をシミュレートするためのVM/4制御プログラムでの処理(これは、親OS上のユーザプログラムでの処理という形であらわれる。
  - E: 実マシン上のOSが、ユーザプログラムとしてVM/4制御プログラムを動かす続けるのに必要な処理(オ3図においてはExplicitには現れない)
- は、VMをサポートするためのオーバヘッド部分であり、これがVMの性能を低下させる要因となっている。

これ以外にも、BMとVMの切換によるTLB，キャッシュメモリの乱れ(これは、平均命令実行時間の遅れとなって現れるはずである)等も、オーバヘッドとして現れるであろう。

機器構成としては、単一プログラム環境として、A: 主記憶4MB，ディスク2台，マルチプログラム環境として、B: 主記憶4MB，ディスク5台の2種類を設定している。即ち、VMの場合には、主記憶は実記憶上に常駐，ディスクはVM専用に割当てられ、VMモードI/Oが可能な状態となっている。

なお、実際には、各マシンには、CR，LPが接続されているが、ここでの測定の期間内には、これらの機器は使用されない。

使用マシンはACOSシステム450で、CPU性能はおおよそ1.1MIPSである。

### 3.3 測定方法

オーバヘッドの見積りを行うために、VM上でいくつかの種類のジョブ(群)を実行し、さらに、このVMと同一の機器構成をもつ実マシン上でも同一のジョブ(群)を実行し、これらの測定結果を比較する。

ジョブ(群)としては、制御プログラムでのシミュレーションの頻度の影響、ジョブスケジューラ等、VM上でのOSの介入の影響、マルチプログラムの影響等を調べるために、次の5種類を設定した。

- a, b, c : 単一ジョブの実行 (I/O発生頻度 高, 中, 低)
- d : ジョブの繰返し実行 (多重度1, 10個, 高I/O発生)
- e : ジョブの繰返し実行 (多重度3, 9個, 高I/O発生)

なお、測定は、各ケースで実行する全ジョブを読み込み、スケジューラに入力し終った時点から最後のジョブの実行が完了するまでに行っている。

オーバヘッド時間等の測定は、VM/4制御プログラムでのシミュレーションオーバヘッドについては内蔵した測定機能を用いてソフトウェアにより、それ以外については、ハードウェアモニターにより測定した。

即ち、各VMの動作時間、BMの動作時間は、VM番号の信号線をプローブすることにより行い、(BMに対しては、VM番号のみを奪っている)、VPディスプレイのオーバヘッドについては、VPディスプレイパックの入口でセットし、出口でリセットするフリップフロップの出力をプローブすることにより行っている。

なお、ハードウェアモニターにより、同時にCPUアイドル時間、TLBヒット率情報、命令実行数等の測定も行っている。

### 3.4 測定結果およびその解析

表2に、各ケースにおける測定結果を示す。ケース1からケース5までは、1台のVM上で種々のジョブ(群)を実行した場合、および、これと同じジョブ(群)を同一の機器構成でBM上で実行した場合であり、ケース6は、3台のVMを同時に動作させたケースである。

経過時間ベースで見ると、オーバヘッド率は測定した各ケースにおいては5%から8%程度となっており、VM上でのI/O発生特性にはあまり依存していない。

表2 測定結果

	ケース1	ケース2	ケース3	ケース4	ケース5	ケース6
実行環境 多重度 I/O発生頻度	シングル 高I/O	シングル 低I/O	シングル 中I/O	シングル繰返し 高I/O	マルチ繰返し 高I/O	VM3台 シングル 高I/O
経過時間(秒) VM/BM	290.0/290.2	356.8/332.8	324.5/300.4	400.4/381.8	536.7/500.5	506.6
オーバヘッド率 (%)	7.3% → (4%)	7.2% → (3%)	8.0%	4.9%	7.3%	—
全CPU時間(秒) VM/BM	92.2/64.7	260.0/221.5	147.9/116.5	134.3/91.1	351.1/237.9	354.3
CPU時間(秒) VM/BM	23.7/65.7	33.7/221.2	27.3/117.3	39.6/89.9	99.5/234.5	101.0/237.2
オーバヘッド率 (%)	42.5% → (29%)	17.4% → (14%)	27.0% → (18%)	47.4% → (32%)	47.6% → (30%)	48.9%
CPUアイドル率 (%) VM/BM	68.2/76.0	27.1/53.5	54.4/61.2	66.5/76.1	34.6/52.5	30.1
VPディスプレイ時間(秒)	1.62	2.48	1.87	2.78	7.33	8.14
シミュレーションオーバーヘッド(秒)	18.2	26.6	21.2	31.5	92.7	84.1
平均I/O時間(ms)	7.8ms	17.9ms	11.7ms	6.0ms	5.9ms	6.1ms
I/Oイベント回数	8715	12667	10025	14885	39568	39135

一方、CPU時間オーバヘッド率は、VMにおけるI/O発生頻度にほぼ比例しており、測定したケースでは17%から49%程度となっている。なおオーバヘッド率はVMにおける所要時間とBMでの時間の差をBMでの時間で割った値である。

経過時間オーバヘッド率がVMでのI/O発生頻度と依存しているのは、I/O発生頻度が高い場合には、CPUのオーバヘッドは大きくなるが、経過時間の中でCPU時間の占める比率が小さいために、CPU時間の延びが経過時間の伸び率に与える影響が少く、I/O発生頻度が低い場合にはこの逆になることによる。

オキ図aにケース1におけるCPUオーバヘッド時間のうちわけを示す。CPUオーバヘッド時間は、BMが動作している時間と、それ以外の時間にわけられるが、BM上での動作時間が9割近くを占めている。BM以外の時間のうち、約4割がVPディスクのオーバヘッドであり、残りは、VMによるTLBヒット率の低下等によるCPU速度自体の低下の影響、その他が含まれている。

BM上での動作時間は、VMのサポートのための制御プログラムの動作時間と親OSの動作時間からなり、前者がその約8割を占めている。なお親OSの動作時間は、本来ならばVMと並行して動作している他のジョブと分担すべきものであるが、ここではVMしか動作していないので、すべてオーバヘッドとしている。

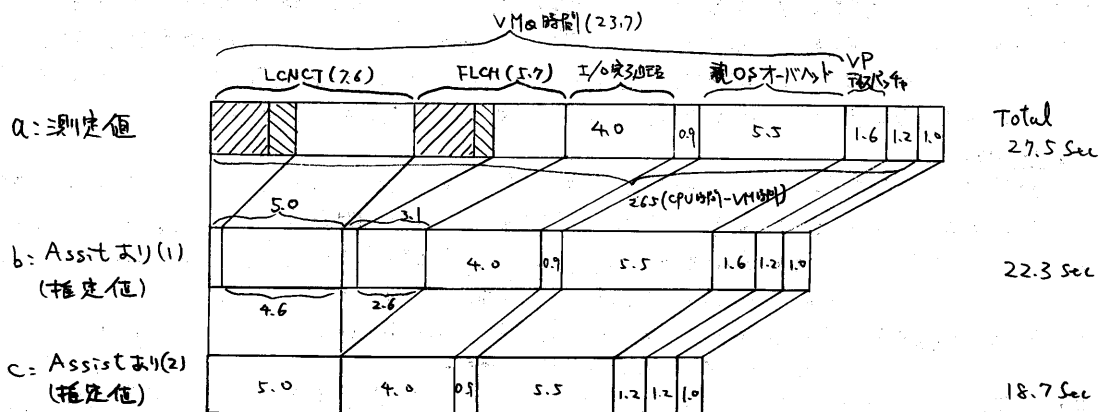
制御ジョブの走行時間の大部分は、論理アドレスモードでのI/Oを起動するためのLCNCT命令、チャンネルの状態を読み込むためのFLCH命令、I/O完了事象のシミュレーションが占めていることかわかる。

またLCNCT命令、FLCH命令のシミュレーションにおいては、VM上の論理アドレスに対するBMからのアクセスの処理(MOVE TO VM / MOVE TO BM)が大きな割合を占めている。(//////で示した部分)、シミュレーション完了時点でプロセッサをスタートさせる処理(Start VP)も比較的大きな部分を占める(///部)

オキ図bに以上の3個の処理をファームウェアで行うとしたときに期待されるオーバヘッド減少の様子を示す。(ここでは、前2者の処理に関しては、ハードウェアのアドレス変換機能が使用できるので処理時間が1/6に、Start VP命令の処理についてはファームウェアのみで実現するので、3割に減少するものとしている。

更にFLCH命令自体をファームウェアで実行する場合についての予測をオキ図cに示す。この場合には、制御プログラムの介入が不要となるため、FLCH命令の処理部がなくなるだけでなく、VPディスクの時間も削減できる。

オキ図 CPUオーバヘッド内訳



これらのVMアシスト機能を加えることにより、CPUオーバヘッドは約3/3、即ち、42.5%であったものが29%程度にまで削減できると推定される。また、VMアシストによるCPU時間の減少は経過時間の減少にほぼダイレクトに寄与すると考えられ、経過時間ベースでのオーバヘッド率も7.3%であったものが約4%にまで削減できると推定される。同様に、各ケースにおいてVMアシストを付加した場合のCPUオーバヘッド率の推定値を表2のCPUオーバヘッド率の項で( )の中に表示す。

VM上の1個のI/O命令により引起される一連の動作に要するオーバヘッド時間は4種類のVMアシストがない場合で約2.2msであり、さらに、VMアシストを加えることにより、1.3ms程度にまで減少させることが可能である。

VM上でマルチプログラムを行った場合(ケース5)についてみても、CPUオーバヘッドはこのケースとI/O発生頻度がほぼ等しいケース(ケース4)とほぼ同じ値を示しており、マルチプログラムで動かすことがVMオーバヘッドに対してほとんど影響を与えていないことがわかる。

また、複数のVMを動作させた場合(ケース6)と一台のVM上で全体として同じジョブ群を実行した場合(ケース5)についてみると、全体のCPU時間はケース6でわずかに(3.2秒)増大している。これは、VPディスプレイの時間が増大したこと、TLBのヒット率が低下しBM、VMでの処理時間が増えたこと等が影響している。しかしながら、CPUオーバヘッドの増大の割合は1%程度であり、ほとんど無視できる範囲に入っていると考えられる。なお、ケース6とケース5においては機器構成が異なり、ディスクへのアクセスが同一ではないために、経過時間ベースでの比較はできない。

#### 4 あとがき

ACOS4/MVPのうえで一般のジョブと並行して複数個のACOS4、ACOS4/MVPを動かすことのできる統合型仮想計算機機能であるVM/4の制御プログラムの概要と、VM/4の性能の測定結果について述べた。

VM/4においては、強力なハードウェア、ファームウェアサポートにより、VMのオーバヘッドの削減が計られており、経過時間ベースでのオーバヘッド率で、5~8%程度が達成されている。現在インプリメント中のいくつかのVMアシスト機能により更に高速化が可能である。また、制御プログラムについても、端末をコンソールとして使える端末コンソール機能等、いくつかの強化を行っている。

最後に、VM/4開発の機会を与えて頂いた基本ソフトウェア開発本部副部長、コンピュータ技術本部坂東課長、C&Cシステム研究所稲崎課長、VM/4制御プログラムの開発に参加された日電ソフト伊藤女に感謝致します。

#### 参考文献

- 1) 藤原他, "VM/4(ACOS-4仮想計算機)のアーキテクチャ" 当研究会資料
- 2) 日本電気, "ACOSシステム450/550システム概説書" DAZ50-2 56.8
- 3) Buzen, J., P.他 "The Evolution of Virtual Machine Architecture" Proc. of NCC 1973, pp291~299
- 4) Goldberg, R. P., "Survey of Virtual Machine Research" Computer, Vol.7, No.6, pp34-45, 1974
- 5) 小野 "簡易型ハードウェアモックの試作" 情報学会第20回全国大会, 547