仮想計算機システムの制御効率を向上させるための方式について
――インライン・バーチャル マシン方式（In-line VM Features）――

田口 敏夫　堀越 彌　栗原 潤一
（（株）日立製作所　中央研究所）

1. まえがき

大規模計算センタでは，システム開発や新システムのテストなどが常に必要であり，その都度，センタ業務を停止させねばならないのが実状である。これに対して，仮想計算機システム（VMS：Virtual Machine System）[2,3]のもとでは，

(1) 複数個のOSが同時に走行でき，
(2) システム開発のための豊富なテスト機能が使えるので，

これを用いることにより，システム開発のためにセンタ業務が中断するという問題は一応解決できる。

しかし，通常の用い方では，VMSの制御プログラム（VMM：Virtual Machine Monitor）が消費するCPU時間，すなわちCPUオーバヘッドが300～500%（システムの性能が1/4～1/6に低下する）と大きく，計算センタ業務の処理能力に重大な影響を及ぼすことになる。　このオーバヘッドは，最近一般化しているVMSの高速化機能[4]を使用しても100%程度までしか低減できず，計算センタで一般的に用いるにはいぜんとして壁がある。

さらに，断続的に行なわれるシステム開発作業のために，実計算機システム（BMS：Bare Machine System）からVMSへ切替える必要が生じ，このために30分以上のロス・タイムが生じることがあった。　VMSからBMSへ移る逆の場合にも同一のロス・タイムが生じ，これらの時間が無視できないため，いわゆるオープン使用でシステム開発を行なうのが従来の姿であった。

以上をまとめると，VMSを利用することによって，システム開発作業を計算センタ業務と共存して行なう上での問題点は次のようになる。

(1) VMMのCPUオーバヘッドが100%前後と大きいこと。
(2) 動作モードの切替えに30分以上のロス・タイムが生じること。

そこで，筆者らはこの点を改善するホイステップとして，

(1) センタ業務を実行する仮想計算機（ホスト・マシン，Host VM）に対するCPUオーバヘッドを30%以下に抑える方式[5]と，
(2) センタ業務を停止することなく動作モードを30秒以内で切替える方式[6,7]，

について検討し，その実験システムを開発した。　筆者らは，上記2つの方式によって，VMSをセンタ業務と共存させること（インラインに使用すること）が可能であると考えており，この方式をインライン・バーチャルマシン方式（In-line Virtual Machine Features）とよんでいる。

本報告では，現在までに開発したインライン・バーチャルマシン方式の実験システムと得られた結果について述べる。なお，本稿では第2章にて従来の問題点を述べ，第3章，4章にてインライン・バーチャルマシン方式とその効果について述べることにする。

2. 従来の問題点

2.1 CPUオーバヘッドの要因

VMSの制御プログラムであるVMMは，主メモリ，CPU，入出力装置などのハードウェア資源をすべて管理しており，特権モードで動作する。　一方，VMS下のOSはすべて非特権モードで動作する構造になっており，OSから発行される特権命令はVMMによってシミュレートされる。　さらに，VMMは各VMのページングやスプーリングの処理も行なうので，各VM下で走るOSのこれら

の処理と重複することになる。

したがって，VMSを実現するためにVMMが費やすCPUオーバヘッドの要因は，図1の左側のように分類して考えることができる。これらがVM下のOSに対してVMM固有のCPUオーバヘッドとして付け加わり，このCPUオーバヘッドが，各OS下で動作するユーザ・プログラムの経過時間に影響を及ぼして性能の低下を招くことになる。

2.2 CPUオーバヘッドの低減方法

2.1節で述べたVMMのCPUオーバヘッドの各要因に対して，図1の右側に示すような低減方法がVMMの高速化機能[2,4]として一般的に考えられている。

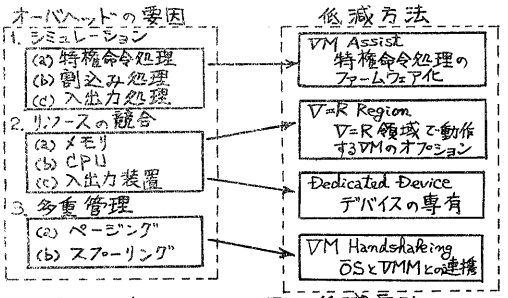

図1. オーバヘッド要因と低減方法

これらの高速化機能の詳細は参考文献3),4),5)に述べられているが，通常，VMS下においてセンタ業務を実行するマシンはこれらの高速化機能を用いることにする。したがって，本報告ではセンタ業務を実行する仮想計算機をホスト・マシン，あるいはHost VMということにする。一方，システム開発作業を行なう仮想計算機をテスト・マシン，あるいはTest VMという。

図1に示した高速化機能をホスト・マシンに適用すると，VMMのCPUオーバヘッドは，バッチ・システムの場合従来の500%から100%程度まで低減するが，通常の計算センタで用いるにはまだ多いと考えられる。

2.3 動作モードを切替えるときの問題点

2.1，2.2節ではVMMのCPUオーバヘッドに関して，そのオーバヘッド要因とその低減方法を述べ，従来のCPUオーバヘッドを把握した。

次に実計算機システム(BMS：Bare Machine System)のもとでセンタ業務を行なっている環境から，VMS下での動作モードへ切替えるための手順について考察する。

BMSの動作モードからVMSの動作モードへ切替えるには，通常，

(1) バッチ・ジョブのイニシェータを終了させ，
(2) TSS，ネットワーク・ジョブなどの業務を終了させた後，
(3) OSを終結(シャットダウン)

することになり，次に，

(4) VMSのVMMをIPL(Initial Program Loading)して，VMSの環境を創り出し，

再び，

(5) OSをIPLして，
(6) TSS，ネットワーク・ジョブのサービスを開始し，
(7) バッチ・ジョブのイニシェータを再起動

させなければならない。また，逆にVMSからBMSの動作モードへ戻るには，再び逆の手順が必要であり，これら一連の処理のために約30分程度のロス・タイムが生じる。なお，長時間のバッチ・ジョブが走行している場合には，動作モードの切替えに30分以上要することもある。

これに対して，筆者らが検討したインライン・バーチャルマシン方式では，上記の手順を踏むことなく計算センタ業務を続行させながら，動作モードの切替えを30秒以内で実現しようとするものである。

3. インライン・バーチャルマシン方式

3.1 概要

第2章ではVMSをセンタ業務の中でインラインに使用するうえでの問題点を述べた。それらは，

(1) VMMのCPUオーバヘッドが100%前後と大きいこと，

(2) 動作モードの切替えに30分以上のロス・タイムが生じること，

である。

上記の問題点に対して，筆者らはテスト・マシンのために最小限1MBの主メモリを増設し，従来から存在する1つのホスト・マシンにBMSでの動作時と同一容量の主メモリを与えることによって，VMMのCPUオーバヘッドを大幅に低減させ，さらに，動作モードの切替えも瞬間的に行なえる方式を検討した。この方式をインライン・バーチャルマシン方式（In-line VM Features）とよんでおり，以下の2つの機能で成っている。

(1) ホスト・テストVM機能[5]

VMSの動作モードにおいてホスト・マシンに対するVMMのCPUオーバヘッドを大幅に低減(30%以下)させる機能。[6],[7]

(2) クリープ・イン/アウトVM機能

システム開発作業が必要なときに，センタ業務を停止することなくVMMが忍び込み(クリープ・イン動作)，動的（30秒以内）にVMSの環境を創り出し，システム開発作業が終了後には逆にVMMが抜け出し(クリープ・アウト動作) BMSの動作モードへ戻る機能。

このインライン・バーチャルマシン方式は，ホスト・OS（ホスト・マシン下で動作するOS）として当社のVOS3（Virtual-storage Operating System 3）[8]を対象にして実現している。

3.2 システム構成

図2は，インライン・バーチャルマシン方式のブロック図を表わしたもので

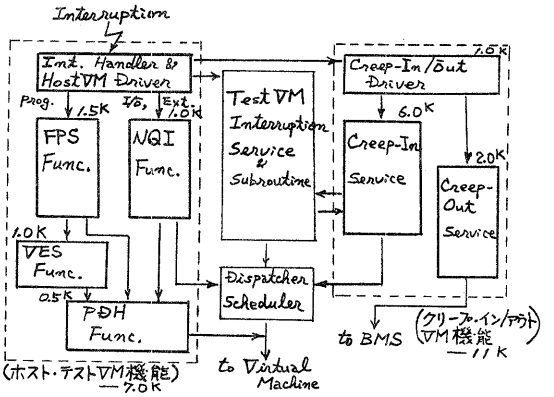

図2. インライン・バーチャルマシン方式のブロック図

あり，各機能は従来のVMMとは独立したモジュール構成となっており，それぞれVMMに接続している。図2の点線内がインライン・バーチャルマシン方式のためのモジュール群であり，開発ステップ数はアセンブラ言語で18Kステップである。この内訳は本体が13K，初期化処理およびVOS3側の制御プログラム（クリープ・イン/アウトVM機能）が5Kステップである。

なお，ホスト・テストVM機能とクリープ・イン/アウトVM機能は組合せて利用できるものであるが，一方の機能のみでの動作も可能である。

3.3 主メモリの割当て方法

図3はVMSにおける主メモリの割当て方法を表わしたものである。図3において，(a)はホスト・マシンが2.2節で述べたVMSの高速化機能の1つであ

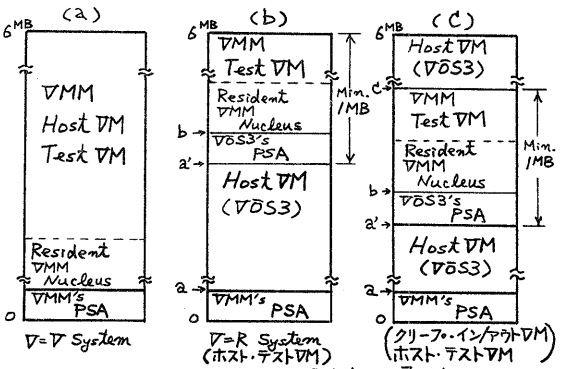

図3. 主メモリの割当て方法

るV=R Regionのオプション[9,10]を使用しない場合であり、(b)はV=R Regionのオプションを使用した場合，あるいはホスト・テストVM機能を単独に使用した場合であり，ホスト・マシン(Host VM)は一定量の主メモリを専有(a~b)し，増設された高位の領域内にテスト・マシン(Test VM)とVMMが存在する。
(c)はホスト・テストVM機能とクリープ・イン／アウトVM機能を組合せた場合であり，ホスト・マシンは主メモリ領域全体を使用しているのが特徴である。
したがって，(c)のa~cはホスト・マシンとVMM，テスト・マシンとが共有することになる。　なお，VMS[9,10]ではPSA(Prefixed Storage Area)をVMMが使用する。

3.4　ホスト・テストVM機能[5]

この機能はホスト・マシンに対するVMMのCPUオーバヘッドを大幅に低減させるものであり，2.2節で述べたオーバヘッドの低減方法に加えて，以下の機能を付加するものである。

(1) VES機能(Virtual Equal Shadow Feature)
(2) FPS機能(Fast Path Selection Feature)
(3) NQI機能(Non-Queued Interruption Feature)
(4) PDH機能(Preferred Dispatching to Host VM Feature)

これら4つの機能は，図2に示したように各々モジュール化されており，それぞれVMMに接続している。
したがって，ホスト・マシンはVMMに対して，
"自分はホスト・マシンである。"
と宣言することによって，上記4つの機能のサービスを受けることができる。

3.4.1　VES機能

ホスト・マシン下のOSが当社のVOS2，VOS3のようにバーチャル・ストーレジをサポートしている場合には，VMSにおけるストーレジのアドレッシング方法は，図4に示すような3段階となる。

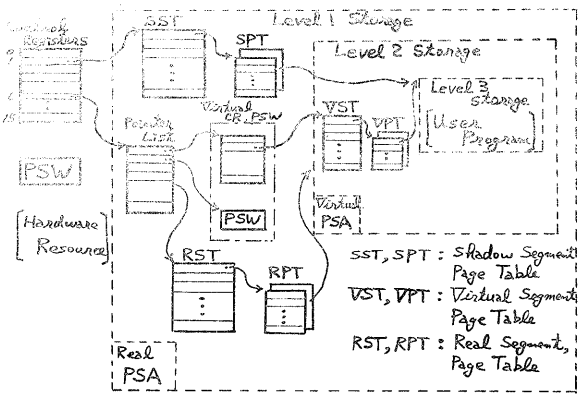

図4　VMSにおけるアドレッシング方法

すなわち，VMSの主メモリを第1レベル・ストーレジ(Level 1 Storage)とすると，ホスト・マシンが実ストーレジとみなすストーレジは第2レベル・ストーレジ(Level 2 Storage)となり，RST(Real Segment Table)，RPT(Real Page Table)によって管理される。　したがって，ホスト・マシンのOSはこの第2レベル・ストーレジ上にアドレス変換テーブルVST(Virtual Segment Table)，VPT(Virtual Page Table)を作り，ユーザ・プログラムを第3レベル・ストーレジ上で動作させる。　ユーザ・プログラムは2回のアドレス変換を受けることになるが，現在のハードウェア[9,10]では1回の変換しか行なえない。
そこで，VMMは2つのアドレス変換テーブル(VST，VPTとRST，RPT)をマージしてシャドウ・テーブル(SST: Shadow Segment Table，SPT: Shadow Page Table)を作り，アドレス変換が1回で済むようにしている。このシャドウ・テーブルの管理をVMMが行なっており，その処理がVMMのオーバヘッドとなる。
VES機能は，図4のシャドウ・テーブルを削除し，ホスト・マシンのOS(筆者らの例ではVOS3)が作るVST，VPTをシャドウ・テーブルとみなして動作させる機能である。　これは図3に示したように，ホスト・マシンはVMSのV=R Region[3]内に存在するためPSAを除いて

第1レベル・ストーレジと第2レベル・ストーレジは1対1の対応がとれているためである。　したがって,ホスト・マシンのPSAに関してはVPTの第0エントリを書替える(図3のa'の値)ことで正しいマッピング関係が成立する。

このVES機能によって,従来シャドウ・テーブルの管理のために要していたCPUオーバヘッドが削減できる。

3.4.2　FPS機能

この機能は,ホスト・マシンに関する特権命令のシミュレーション処理[9),10)]と入出力割込み処理[9),14)]を高速化するものである。VMMはシミュレーション処理が必要になったとき,要求元がホスト・マシンであるか否かを調べる。　ホスト・マシンであるならば,VMMは図2に示したホスト・マシン専用の処理モジュールへ制御を渡す。　そのモジュールでは,①命令コードの解釈,②シミュレーション処理を高速に行なった後,③ホスト・マシン専用のディスパッチャを介してホスト・マシンに制御を返す。

このFPS機能の対象となる特権命令[9),10)]および処理は,

(1)入出力命令(SIO, TCH),
(2)状態制御命令(LPSW, PTLB, LCTL),
(3)タイマ命令(SCKC, STCKC, SPTなど),
(4)入出力割込み処理

であり,VMMの処理ステップ数は,従来に比べて1/2～2/3まで減少した。

3.4.3　NQI機能

この機能の特徴は,

> ホスト・マシンに関する割込みを最優先に処理して,必らずホスト・マシンをディスパッチする.

ことである。　したがって,割込み発生時にテスト・マシンが走行していても,制御はホスト・マシンへ移ることになる。

この機能の対象となる割込みは,

(1)外部割込み(External Interruption),

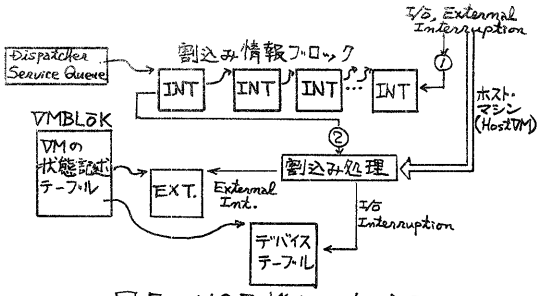

図5　NQI機能の概念図

(2)入出力割込み(I/O Interruption),

であり,図5はNQI機能の基本概念を示したものである。　図5より,ホスト・マシンに関する割込みは,ディスパッチャの待ち行列に登録されることなく直ちに処理されることが分かる。

この機能の効果は,

(1)ディスパッチャの処理ステップ数が削減できること,
(2)割込み処理が高速化できること,
(3)割込み発生後,直ちにホスト・マシンがディスパッチされること,

である。

3.4.4　PDH機能

この機能は,

> ホスト・マシンがウェイトしない限り,VMMはホスト・マシンをディスパッチする.

ことを原則としている。　したがって,PDH機能ではVMMのスケジューリング方法を変更している。

従来,VMMは各VMに対してCPU資源を均等に割当てるためにタイム・スライス制御を行なっていたが,このPDH機能では,ホスト・マシンに対して,

(1)テスト・マシンよりも長いタイム・スライス値を割当て,かつ,
(2)ホスト・マシンがウェイトしない限り,ホスト・マシンをディスパッチする,

ようになっている。

なお,このタイム・スライス値はVMMのコマンド[5)]によって自由に変更できるようにしてある。

3.5 クリープ・イン/アウトVM機能[6),7)]

3.5.1 基本概念

計算センタ業務を停止することなく，BMSの動作モードから図6に示すようなVMSの動作モードへ切替えるためには，CPU，主メモリ，入出力装置などのハードウェア資源の制御権を，瞬間的にVOS3からVMMへ移す必要がある。逆に，VMSからBMSへ戻る場合も同一のことが言える。これらの操作をハードウェア資源の切替え制御という。

したがって，ハードウェア資源の切替え制御を円滑に行なうためには，VOS3のサービスを瞬間的に凍結させ，その凍結時間内にVMMが忍び込み(クリープ・イン：Creep-In)，図6に示したVMSの環境を創り出し，VMMがハードウェア資源を管理すれば良い。逆に，VMSからBMSへ戻るときも，VOS3のサービスが瞬間的に凍結している間にVMMが抜け出す(クリープ・アウト：Creep-Out)ことになる。

この機能を実現することは，VOS3が動作するCPU利用率が図7に示すように変化することになる。すなわち，クリープ・イン動作，クリープ・アウト動作が開始されるとVOS3のサービスは除々に低下し，ある時点(t_4, t_{14})ではVOS3のサービスが完全に凍結することになり，この凍結時間(Freezing Time：T_{in}^{f}, T_{out}^{f})内に動作モードを切替えるための処理がなされる。

以上をまとめると，図7より次の3つの指標が定義できる。

(1) クリープ・イン・タイム (Creep-In Time : T_{in})

(2) クリープ・アウト・タイム (Creep-Out Time: T_{out})

(3) 凍結時間 (Freezing Time : T_{in}^{f}, T_{out}^{f})

これらの指標のうち，(1) (2)はVOS3が他のサービスを並行して行なえる区間も存在するので，T_{in}, T_{out}が30秒前後，T_{in}^{f}, T_{out}^{f}は15秒前後を目標としてシステムを開発すれば，端末ユーザへの影響も少ないと考えた。

以下に本機能の実現方式，およびその処理方式について述べる。

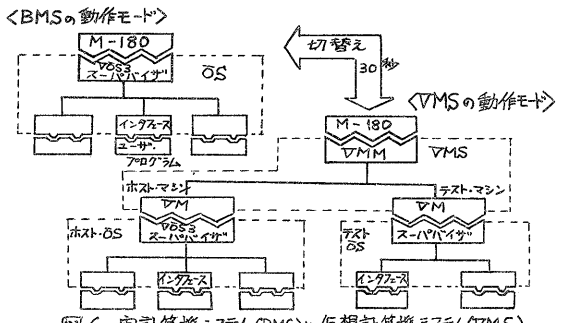

図6. 実計算機システム(BMS)と仮想計算機システム(VMS)

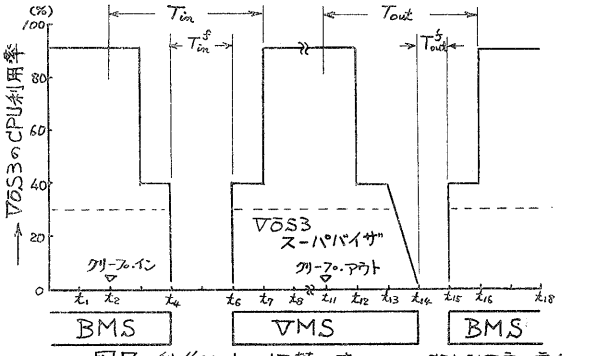

図7. 動作モードの切替え時におけるCPU利用率の変化

3.5.2 検討事項と解決方法

VOS3のサービスを瞬間的に凍結させる方式とは，①CPUサービスと，②入出力サービスの停止・再開方法を検討することになる。さらに，VMMが動作するために，③主メモリの確保方法，④VMMのローディング方法も検討する必要がある。①②はクリープ・イン動作，クリープ・アウト動作に必要なものであり，③④はクリープ・イン動作時に必要である。

表1は，検討すべき事項と解決方法を対比して示したものである。表1において，○印は採用したことを表わし，×印はVOS3の拡張が必要となるため不採用としたものである。表1より，VOS3の制御下で走行する制御プログラムを設ければ良いことが分かる。この制御プログラムをCIOP(Creep-In/Out Program)ということにする。

このCIOPは，

(1) VOS3のV=R領域[8)]で動作し，

(2) 特権モードで走行する

制御プログラムとなり，図6に示したVMS

表1　検討事項と解決方法

	検討事項	説明	解決方法	コメント	評価
1	主メモリの確保と解放	VMSのための主メモリ領域	(a) スーパバイザ・マクロを新設する。	OS依存	×
			(b) VOS3のもとで V=R ジョブを走行させる。		○
2	VMMモジュールのローディング	VMMモジュールはファイル形式で格納してない	(a) 凍結時間内にロード (SIO命令)	凍結時間の増加	×
			(b) VMMをファイルとして扱い前処理でロード		○
3	CPUサービスの停止・再開	凍結時間を作り出す。	(a) スーパバイザ・マクロを新設する。	OS依存	×
			(b) VOS3のもとで動作するプログラムが制御	特権モード	○
4	入出力サービスの停止再開	同上	(a) スーパバイザ・マクロを新設する。	OS依存	×
			(b) 制御プログラムが制御する。	クリープ・イン	○
			(c) 制御プログラムとVMM間で交信する。	クリープ・アウト	○

を創り出すための主メモリを確保し，VMMをロードした後，クリープ・イン/アウト動作を制御することになる。

なお，本機能を用いたときの主メモリ構成は図3の(C)のようになる。すなわち，図3のa'～cは，VOS3がCIOPに割当てた領域であり，この領域内でVMMが動作する。

3.5.3　処理方式

図8はクリープ・イン/アウトVM機能の処理方式を示したものである。図8より，CIOPがクリープ・イン/アウト動作を制御していることが分かる。

(1)　クリープ・イン動作[6]

クリープ・イン動作の起動は，BMSのもとでCIOPを走行させることによってなされる。CIOPはVMMモジュールを主メモリ内にロードし，CPUサービス，入出力サービスを強制的に停止（図8の2.～4.）させた後，VMMに制御を渡す。VOS3サービスの凍結は，CIOPが特権モードで動作し，かつ，入出力割込み情報を順次スタックすることで実現している。なお，CIOPはVMMに制御を渡す前に，VOS3のPSAを図3で示したように移している。

VMMは図8の6.～10.の処理を行なうことによって，図6で示したVMSの環境を創り出すことになる。

CIOPに制御が戻った時点（図8の11.）では，VOS3およびCIOPはVMS下で動作している。CIOPは，後にクリープ・アウト要求がなされるまでロング・ウェイト状態となる。これはVMSのための主メモリを確保し続けるためである。

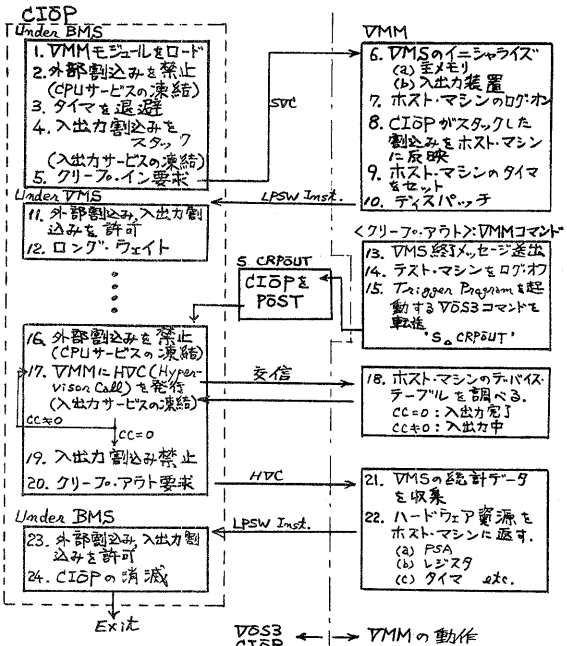

図8.　クリープ・イン/アウトVM機能の処理方式

以上によって，動作モードをBMSからVMSへ動的に切替えることができる。なお，図8の2.～10.の処理は図7の T_{in}^{f}（t_4～t_6）内でなされ，この時間が15秒以下ということになる。

(2)　クリープ・アウト動作[7]

クリープ・アウト動作の起動は，VMMのコマンドによって行なわれる。VMMはこのコマンドを受けると，ホスト・マシン以外のすべてのテスト・マシンに対して，

"VMSの環境が終了した。"

のメッセージを送出する。コマンドで指定された時間が経過すると，テスト・マシンを強制的にログ・オフさせる。

その後，ホスト・マシンのもとでウェイト状態となっているCIOPを再起動するために，ホスト・マシンのコンソールに対して，

"S. CRPOUT"

のデータ・ストリームを生成する。

このデータ・ストリームは，CRPOUTプログラムを起動するためのOSコマンドであり，これによってCRPOUTプログラムがVOS3のもとで実行される。CRPOUTプログラムはウェイト状態となっているCIOPを再起動するために，仮想空間連絡機能（Cross Memory POST）を使用しており，トリガ・プログラム（Trigger Program）ともよんでいる。

再起動されたCIOPは外部割込みを禁止状態にすることで，VOS3が他のジョブのサービスを行なわないようにした後，入出力サービスの凍結処理を行なう（図8の16.～18.）。

入出力サービスの凍結処理とは，

(a)新たな入出力サービスの抑止と，

(b)既に発行されている入出力サービスの完了確認，

を行なうことであり，(a)は上記のように外部割込みを禁止にすることで保証できる。一方，(b)はクリープ・イン動作の場合と異なり，表1に示したようにCIOPとVMMとが交信しながら進める。この交信のためにVMMのHVC（Hypervisor Call）機能[4]を拡張してある。

CIOPからHVCが発行されると，VMMはホスト・マシンのデバイス・テーブルを調べ，入出力サービスが完了しているか否かを確認する。この結果はCC（Condition Code）に反映される。したがって，CIOPはVMMからCC=0が返されると，次にクリープ・アウト要求を発行する。このような方式を採用することによって，VOS3のCPU利用率は図7に示したように除々に低下してゆき，$t_{1/4}$で完全に凍結することになり，クリープ・イン動作との違いが生じる。

VMMはクリープ・アウト要求を受けとると，今まで使用していたハードウェア資源をホスト・マシンに返却することによって，BMSの動作モードへ切替ることができ，この処理は図7のT_{out}^{+}内に行なわれる。

4. 実験結果

筆者らは，第3章で述べたインライン・バーチャルマシン方式（In-line Virtual Machine Features）の効果を把握するために，実験システムを開発した。この章では，バッチ・システムを例として，この実験システムの性能測定結果を種々の角度から検討する。

4.1 ベンチマーク・ジョブ

バッチ・システム用の負荷として，30題のジョブよりなるベンチマーク・ジョブを設定した。これらのジョブは，表2に示す特性を有しており，

(1) 技術計算，

(2) ファイル処理，

(3) メーリプロッタ処理

などの各ジョブを含む平均420 KBのプログラム群からなり，M-180を使った計算センタの標準像の1つと考えられるものである。

このベンチマーク・ジョブをBMSのもとで実行すると，表3に示す時間で終了するものである。なお，これらの測定にはハードウェア・モニタを用いている。

表2 ベンチマーク・ジョブの特性

	項目		値
1	ジョブ数		30題
2	課金の対象となるCPU時間	最大	162秒
		最小	2
		平均	39
3	ジョブ当りの入出力回数	最大	8,776回
		最小	57
4	ジョブ当りのメモリ使用量	最大	1,220 KB
		最小	200
		平均	420
5	使用する磁気テープ台数		6台
6	磁気ディスク台数	システム・ファイル用	3台
		ユーザ・ファイル用	3台

表3 実計算機システムでの性能

	項目		値	
1	中央処理装置		HITAC M-180	
2	オペレーティング・システム	システム名	VOS3 02-00	
		起動したイニシエータ数	6	
3	主メモリの容量		2 MB	5 MB
4	性能データ	CPU時間	19.5分	16.3分
		経過時間	38.0分	19.1分
		CPU利用率	51 %	85 %
		特権モードの比率	53 %	45 %

4.2 ホスト・テストVM機能の効果

4.2.1 CPUオーバヘッドの低減効果

4.1節のベンチマーク・ジョブをVMSのもとで実行したときのCPUオーバヘッドについて検討する。

VMSにおける主メモリの割当て方法は,図3で示したようにホスト・マシンに対して2MBの主メモリを専有させ,残りの1MBをVMM,およびテスト・マシンが使用し,合計3MBとなる。

したがって,ベンチマーク・ジョブを実行するホスト・マシンの主メモリは,BMSとVMSとで同一容量となる。 なお,BMSで5MBの場合にも,VMSでは5MB+1MBの合計6MBとなる。

図9の上段は,ホスト・マシンの主メモリ容量を2MBとして,かつ,ホスト・テストVM機能を用いないときのVMMのCPUオーバヘッドを表わしている。 なお,2.2節で述べた一般的に考えられている高速化機能は使用しており,それらは,

(1) VM Assist,
(2) V=R領域の機能,
(3) 入出力装置の専有機能,
(4) OSとの連携機能

である。 図9より,VMMのCPUオーバヘッドは110%であり,特権命令のシミュレーションの処理に75%を要していることが分かる。 また,非特権モードで実行されるVM AssistのCPUオーバヘッドは25%であり,そのうちシャドウ・テーブルの保守処理に18%を要している。

一方,ホスト・テストVM機能を適用したときのCPUオーバヘッドは,図9の下段に示すとおりである。

3.2節で述べたホスト・テストVM機能を付加すると,VMMのCPUオーバヘッドは30%まで減少し,従来に比べて3.7倍の性能向上となっている。 このように,VMMのCPUオーバヘッドが大幅に低減したのは,以下の効果が大きいものと考えられる。

(1) VES機能によって,PTLB命令[9),10)],LCTL命令[9),10)]のシミュレーション処理が,従来に比べて約6倍に高速化できたこと。

(2) VES機能によってシャドウ・テーブルが削除されたために,VM Assistによるシャドウ・テーブルの保守処理が削減できたこと。

(3) FPS機能によって,入出力命令などのシミュレーション処理が約2倍に高速化できたこと。

(4) NQI機能,PDH機能によって,割込み処理が約1.7倍に高速化できたこと。

さて,上記で述べたCPUオーバヘッドは,ホスト・マシンの主メモリ容量が2MBのときであるが,5MBの容量の場合について考察してみる。 ホスト・マシンの主メモリ容量が5MBのときのCPUオーバヘッドは,

(1) 従来,65%のものが,
(2) ホスト・テストVM機能によって,19%まで低減できており,約3.5倍の性能向上となっている。

これは,表3からも明らかなようにBMSの環境においても主メモリを増加させると,CPU利用率が51%から85%へと向上し,それにともないユーザ・プログラムが走行する時間も増加(逆に,VOS3のスーパバイ

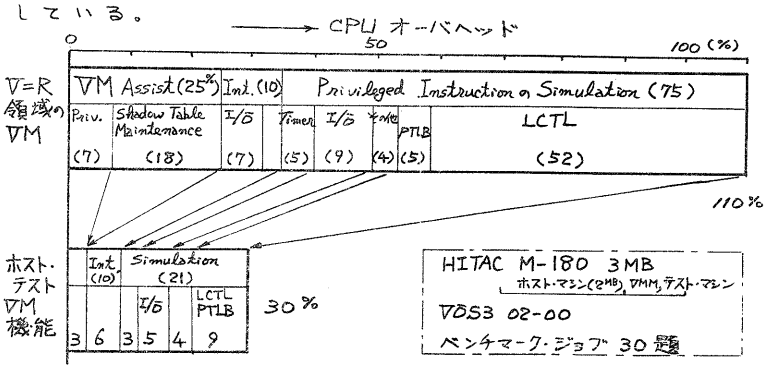

図9. CPUオーバヘッドの低減効果

ザ時間が減少）することによって，VMMが介入する頻度が減少したために，VMMのCPUオーバヘッドも２０％以下になったものと思われる。

4.2.2 特権モードの比率との関係

したがって，BMSにおけるスーパバイザ時間の比率，すなわち特権モードの比率と，VMSにおけるVMMのCPUオーバヘッドとの関係について興味が持たれる。そこで，筆者らは，FORTRANプログラムによるコンパイル（Compile），リンク（Linkage-edit），ゴー（Go）で成り立つ技術計算専用のベンチマーク・ジョブを設定してみた。

このベンチマーク・ジョブは２０題で成っており，４MBのBMSのもとで以下の特性を有するものである。

(1) CPU時間 ： 21.5分
(2) 経過時間 ： 22.4分
(3) CPU利用率 ： 96 %
(4) 特権モードの比率 ： 6 %

このベンチマーク・ジョブでは，VMMのCPUオーバヘッドが従来１８％であったものが，６％まで減少している。

図１０は，BMSでの特権モード比率を種々変化させたときに対するVMMのCPUオーバヘッドを示したものである。

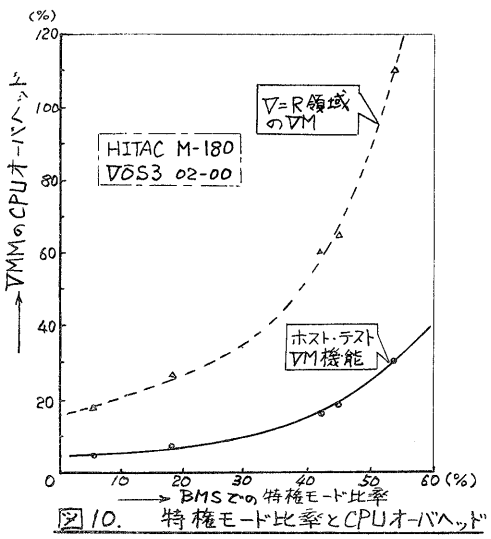

図10. 特権モード比率とCPUオーバヘッド

図１０より，以下のことが分かる。

(1) ホスト・テストVM機能による性能向上の効果は，特権モード比率が３０％以下では３倍程度であり，比率が高くなるに従い，3.5，4倍へと効果が大きくなる。

(2) これは，特にVES機能とFPS機能の効果によるものと思われる。

(3) TSSを含む一般の計算センタでの特権モード比率は30～50％と考えられ，ホスト・テストVM機能の効果が十分に期待される。

4.2.3 経過時間の低減効果

次に経過時間について検討する。この経過時間に関しては，次式に示すようなRBT（Relative Batch Throughput）とよばれる評価指標を導入する。

$$RBT = \frac{T_N}{T_V} \qquad \text{-----(1)}$$

T_N：BMSでの経過時間
T_V：VMSでの経過時間

このRBTはベンチマーク・ジョブがすべて終了するまでの経過時間の比率であるが，一方，一般ユーザにとっては，

"自分のジョブの経過時間がどの程度延びるのか？"

ということが興味深い問題であろう。

そこで，RBTに加えてEF（Elongation Factor）とよばれる評価指標を導入する。このEFは次式のように定義する。

$$EF = \frac{1}{n}\sum_{j}^{n}\frac{T_j^V}{T_j^B} \qquad \text{------(2)}$$

T_j^B：j番目のジョブに関するBMSでの経過時間
T_j^V：j番目のジョブに関するVMSでの経過時間

表４は，筆者らが設定したベンチマーク・ジョブに関してRBTとEFの結果をまとめたものである。以下，RBTとEFについて考察する。

(1) RBTについて

従来のRBTは0.70程度であるが，ホスト・テストVM機能によって0.96程度まで向上し，BMS下で実行した場合とほとんど

表4. ベンチマーク・ジョブとRBT, EFの関係

ベンチマーク・ジョブ			条件	CPUオーバヘッド %	RBT	EF
A	2MB	CPU: 51% SV: 53%	V=R領域のVM	110	0.69	1.36
			ホスト・テストVM機能	30	0.97	1.11
	5MB	CPU: 85% SV: 45%	V=R領域のVM	65	0.70	1.58
			ホスト・テストVM機能	19	0.96	1.21
B	4MB	CPU: 96% SV: 6%	V=R領域のVM	18	0.86	1.27
			ホスト・テストVM機能	6	0.93	1.24

差がなくなる。 なお,技術計算専用のベンチマーク・ジョブの場合(B)には0.93である。これはBMSでのCPU利用率が96%と高く,VMMのCPUオーバヘッドがRBTに影響しているためである。

(2) EFについて

従来のEFは1.36~1.58程度であったものが,ホスト・テストVM機能によって1.11~1.24まで減少している。しかし,ユーザに対してはジョブの応答時間が20%程度延びており,これはVMMのCPUオーバヘッドのために,各ジョブのサービスのされかたがBMSの場合と異なってくるために,RBTよりも悪いものと思われる。
したがって,今後,さらにCPUオーバヘッドを減少させる方式を検討する必要がある。

4.3 クリープ・イン/アウトVM機能の効果

図11は,クリープ・イン/アウトVM機能を動作させたときのコンソール・シートを抜萃したものである。
このときのVOS3の負荷状況は次のとおりである。

(1) バッチ・ジョブのイニシェータ数 ... 6本
(2) アクティブなTSS端末数 ... 20台

図11より,以下のことが分かる。

(1) CIOPが処理を開始してからクリープ・イン動作が完了するまでの時間,すなわちクリープ・イン・タイム(T_{in})は17秒であり,目標値の30秒以下を達成している。

(2) クリープ・イン動作におけるVOS3の凍結時間(T_{in}^{s})は,目標値の15秒に対して14秒であり一応達成しているが,このうちCIOPが割込みを吸収する時間に5秒間要している。 今後,この時間を1~2秒間とすることによって,凍結時間をさらに短縮させることも検討している。

(3) クリープ・アウト動作に関しては,
(a) クリープ・アウト・タイム(T_{out})が2秒,
(b) VOS3の凍結時間(T_{out}^{s})が0.2秒,
となっており,目標値を十分に満たしている。

5. むすび

以上,仮想計算機システム(VMS: Virtual Machine System)を計算センタ業務と共存させる(インラインに使用する)方式,すなわちインライン・バーチャルマシン方式(In-line Virtual Machine Features)の処理方式と実験システムについて述べた。

このインライン・バーチャルマシン方式は,

(1) センタ業務を実行するホスト・マシンに対するVMMのCPUオーバヘッドを30%以下に抑えるホスト・テストVM機能[5]と,
(2) システム開発作業が必要になったときに,センタ業務を停止することなく動作モードを動的に切替えるクリープ・イン/アウトVM機能[6),7)],

(a) クリープ・イン動作

```
4000 19.20.51 STC   36  ** IN-LINE VMF **
 781 CREEP-IN VMF STARTED          ... 19:20:33.960
 781 INTERRUPTION STACK STARTED    ... 19:20:36.614
 781 INTERRUPTION STACK ENDED      ... 19:20:41.859
 781 VM/CP INITIALIZE STARTED      ... 19:20:42.477
 781 VIRTUAL MACHINE READY         ... 19:20:50.765
 781 USED CPU TIME (M-SEC)         ... 00:00:00.065
```

(b) クリープ・アウト動作

```
4000 19.21.19 STC   36  ** IN-LINE VMF **
 798 CREEP-OUT VMF STARTED         ... 19:21:17.985
 798 VMS SHUTDOWN STARTED          ... 19:21:19.751
 798 BARE MACHINE READY            ... 19:21:19.974
```

図11. クリープ・イン/アウトVM機能の動作例

で成り立っている。

実験システムを開発した結果，

(1) VMMのCPUオーバヘッドが従来の110%から30%まで低減でき，

(2) BMSからVMS，あるいはその逆への切替えが30秒以内，

で可能となっている。

この実験システムは，従来のシステムに最小限1MBの主メモリを増設し，その増設された主メモリ内でVMMとテストマシンを走行させ，従来から有する主メモリ部で走行する1個の実マシンに対するVMMのCPUオーバヘッドを低減させたものであるが，今後

(1) ホストマシンが複数個存在するときにもCPUオーバヘッドを低減し，また，

(2) テストマシンに対するCPUオーバヘッドをさらに低減させるための，

第1段階として意味があるものと考える。

この実験システムは，現在，当社中央研究所の計算センタにおいて試験的な使用を開始しており，本方式によって，最近の24時間運転サービスとシステム開発作業のための時間確保という2つの相反する要求を満たすことが期待される。

最後に，本研究の遂行にあたって開発の機会を与えていただいた(株)日立製作所中央研究所堤副所長，ならびに御協力いただいた(株)日立製作所ソフトウェア工場山本，片岡，野口主任技師，同神奈川工場小高主任技師，および本実験システムの開発に際し貴重な御討論，御指導をしていただいた中央研究所の久保，長島，吉住の各研究員に謝意を表するとともに，実際の運用面で御協力いただいた桑原計算センタ長，本林研究員，並木，下位技師，有本氏その他の諸氏に厚く感謝の意を表します。

参考文献

1) R.P. Goldberg: Survey of Virtual Machine Research, COMPUTER, P.P34-45 (June 1974)

2) 日立製作所：VMS 概説，日立マニュアル，8080-3-001

3) IBM社：VM/370 Introduction，IBM社マニュアル，GC20-1800

4) IBM社：VM/370 System Programmer's Guide，IBM社マニュアル，GC20-1807

5) 日口，堀越，桑原：仮想計算機システムの制御効率を向上するための方式と実験結果，情報処理学会論文誌，Vol.20 No.4 P.P281-287 (1979-7)

6) 日口，桑原，堀越：実計算機モードと仮想計算機モード間の動的切替え制御方式について ――実計算機モードから仮想計算機モードへ移る方式，情報処理学会 第20回全国大会，予稿集，P.P125-126 (1979-7)

7) 桑原，日口，堀越：実計算機モードと仮想計算機モード間の動的切替え制御方式について ――実計算機モードへ戻る方式，情報処理学会，第20回全国大会 予稿集 P.P127-128 (1979-7)

8) 日立製作所：VOS3 概説，日立マニュアル 8090-3-001

9) 日立製作所：M170/M180処理装置解説，日立マニュアル，8090-3-001

10) IBM社：IBM System/370 Principles of Operation，IBM社マニュアル，GA22-7000