

高性能仮想計算機システムの開発

梅野英典，大町一彦，日野章（日立製作所・システム開発研究所）
井村淳一（日立製作所・ソフトウェア工場）

1. はじめに

大規模オンラインシステムの移行においては、しばしば、旧オペレーティングシステム（OS）から新OSへの移行が行なわれる。その場合、業務プログラム（MPP）の移行が問題となることが多い。旧MPPが十分な修正で新OS下で使用できる場合や、その機能の陳腐化のために作りなおさなければならぬ場合の問題は少ない。しかし、旧MPPの機能を引き続き新OS下で使用したいにもかかわらず、新OS下では動作できなくなり、その修正に数百年月を要する場合の問題は大きい。この事は、特に実記憶方式OSから仮想記憶方式OSへ移行する場合に多く見出される。従来からの移行方法は、

- (1) 新旧ハードウェアシステムの併設による同時運用、
- (2) 旧MPPのコンバージョン

である。これらは、設備費用を多く取るため得策でなく、また、短期間のコンバージョンは、プログラムの信頼性を低下させやすいという問題点を含んでいる。

この論文では、上記問題点を解決するための仮想計算機システム（VMS）による多重オンラインシステムのサポートについて述べる。

2. 仮想計算機システム

仮想計算機（VM）は、実ホスト計算機と機能的に同等な論理的な計算機である。その大部分の命令は、直接実ホスト計算機により実行される。VMSは1台の実ホスト計算機下で同時走行している複数個のVMからなる。Virtual Machine Control Program（VMCP）は、VMSを実現する制御プログラムである。

VMは実計算機と同様に、2つのプロセッサステート（スーパーバイザステート、プロブレムステート）、完全な命令セット、ストレージ、入出力（I/O）装置を持つ。VMCPは、プロセッサシミュレーション、メモリシミュレーション、I/OシミュレーションによりVMを実現する。

3. オンラインシステムの移行

従来の移行方法とVMSによる移行方法を比較する。

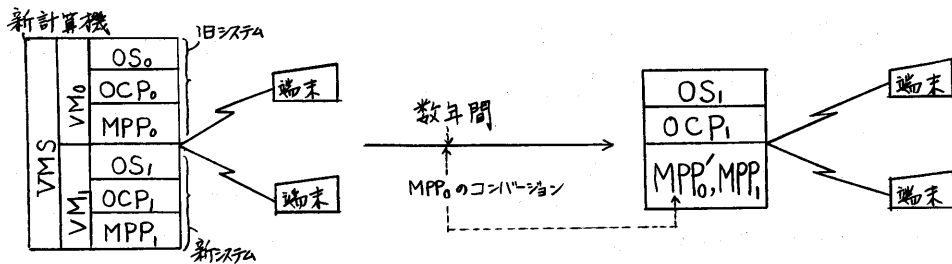
従来の移行方法の問題点は、

- (1) 新旧ハードウェア設置の費用大
 - (2) MPPの短期コンバージョン工数によるコスト集中
 - (3) 信頼性の低下
- である [2]。

これに対してVMSによる移行（図1参照）は、以下の利点を持つ。

- (1) 1台の実ホスト計算機により、旧オンラインシステムの運用と新オンラインシステムの開発及びテストを同時に行なうことができる。
- (2) コンバージョンコストの平準化：新旧オンラインを同時運用することにより旧MPPを徐々に長期間かけてコンバージョンすることができる。これによりコンバージョンコストの短期集中を防ぐことができる。

(3)信頼性の保持：旧MPPをそのまま使用することができる。



新旧システムの共存 (多重オンライン)

移行完了後

図1. VMSにおける移行

これらの利点にもかかわらず、VMSによる多重オンラインをサポートするためには、機能上及び性能上の問題点を解決しなければならない。以下に、これらについて述べる。

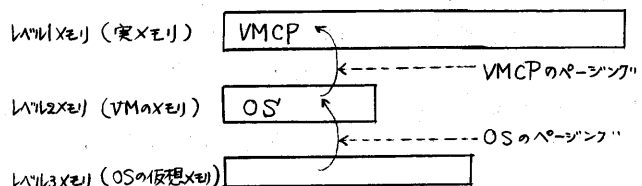
4. VMSの性能上の改良 4.1 CPUオーバヘッドの主要因

VMSの性能はCPUオーバヘッドにより左右される。その主要因は以下のとおりである[1]。

(1)シミュレーションオーバヘッド：VMCPはVM上のOSの発行した特権命令及び割込みのシミュレーションを行なう。これがオーバヘッドとなる。

(2)システムリソースの二重管理オーバヘッド：VM上のOSは、自分のリソース(CPU, メモリ, I/O)を管理し、VMCPは、システム全体の実リソースの管理を行なう。このため両者による二重管理が発生し、オーバヘッドとなる。この典型が二重ページングである。これは、VM上のOSのページングと、VMCP自身のページングの組合せである。図2に示す。

(3)VMCP固有のリソース管理オーバヘッド。



4.2 従来のCPUオーバヘッド削減方法

(1)ファームウェアアシスト：VMの特権命令をシミュレーションするVMCPの処理を支援するファームウェアである。VM/370におけるVMAアシスト(VMA)はこの典型である。

図2 二重ページング

(2)OSとの連携：VM上のOSは通常自分なVM上で走行していることは意識しない。そのために実計算機上では必要だが、VM上では不要となる処理(例えば図2のレベル2メモリのクリア)も同様に行なっている。これを、OSにVM上で走行していることを認識させ、これにより、VM上での不要な処理を省くことにより性能をあげることができる。これをOSとの連携という。

(3)システムリソースの専有化：実リソースは、通常VMCPにより各VMに共有させることとなるが、特定のVMの性能を向上させるために、実メモリの一部分、いくつかのI/O装置等を特定のVMに占有させ、その二重管理を除外することができる。その典型として図3に示すV=RのVMがある。

$V=R$ VMでは、そのXメモリは実メモリ上に連続的に常駐化され、0ページを除いてレベル2メモリアドレス=レベル1メモリアドレスの関係がある。この $V=R$ VMでは、二重ページングや、I/OシミュレーションでのCCW (Channel Command Word: I/Oチャネルが実行するI/O動作を規定するコマンド)変換を省略することができるので性能をあげることができる。CCW変換とは、OSの用意したCCW (レベル2メモリアドレスで構成されている)を、実I/Oチャネルが実行できるように、レベル1メモリアドレスによるCCWに変換することで、VMCPのI/Oシミュレーションの一部分で行なわれている。前者を仮想CCW、後者を実CCWと呼ぶ。 $V=R$ VMは高性能で走行させることができるが、システムに1台しか生成できず、従って、オンラインシステムの移行用として少なくとも2台の高性能VMの要求を満足することができない。このため以下に述べる高速化方法を採用した。

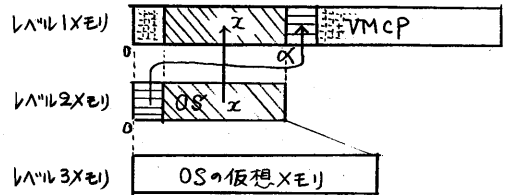


図3 $V=R$ のVM

4.3 複数台の高性能仮想計算機の実現方法
複数台の高性能VMをサポートするために、常駐VMと高速入出カシミュレーションを実現した。これらについて以下に述べる。

4.3.1 常駐VM

$V=R$ VMと異なる高性能VMとして図4に示す常駐VMを導入した。常駐VMのXメモリは連続的にレベル1メモリに常駐されており、そのXメモリアドレス属性は、

レベル1メモリアドレス =

レベル2メモリアドレス + α

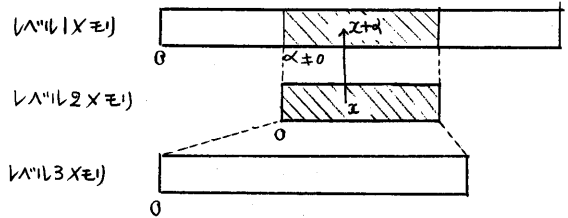


図4 常驻VM

の関係を持つ。変位 α は固定されたページ番号であり、各常驻VMにVMSのシステム生成のときに与えられる。常驻VMでは、そのXメモリの常駐化により二重ページングを除外することができ、さらに、そのXメモリアドレス属性によりCCW変換の高速化、VMAの高速化を図ることができる。常驻VMは、実Xメモリの許す範囲内において複数台定義することができる。

4.3.2 高速入出カシミュレーション (Fast I/O)

VMSのCPUオーバーヘッドの中で、I/Oシミュレーションオーバーヘッドは主要因を占める。表1に示すベンチマークジョブでは、全体の60%~85%がI/Oシミュレーションオーバーヘッドである。これを削減するためにFast I/Oを実現した。Fast I/Oでは、I/O処理の正常処理に注目し以下の方法でCPUオーバーヘッドを削減した。

- (1) 使用頻度の高い専用装置に限定することにより論理を単純化した。
- (2) I/O制御アロックをできる限りシステム開始時に獲得し、そのフィールドの値を確定する。これにより、動的な処理オーバーヘッドを削減した。
- (3) Fast I/O内での固有のXメモリ管理: Fast I/Oで使用するI/O作業領域

は、Fast I/O 内で管理し、VMCPの全体管理から除外した。

(4) I/O 完了時のCSW変換の高速化：I/O完了割り込み時CSW (Channel Status Word) 情報をVMに反映する必要がある。このとき、CSW内のCCWアドレスを実から仮想へ変換しなければならない。このため従来は、CCWをひとつずつ検索していたが、I/O起動時には、最終CCWの仮想-実アドレスを用意しておくことにより高速化した。

(5) 割り込み反映の高速化：I/O割り込みは、従来まず、VMCPの管理する仮想装置に反映され、その後ディスパッチャで優先順位とI/O割り込みマスクを考慮した上でVMに反映される。これをFast I/O内に簡単なディスパッチャ論理を設け、VMがI/O割り込み可の場合は、仮想装置に反映することはやめて、直接VMのオペ-に割り込みを反映することにより高速化を図った。

表1 ベンチマークジョブ

#	測定制件	CPU利用	I/O頻度
1	VOS2 ^{#2}	13.1%	149 ^{#1}
2	V=RVM	13.1	199
3	CCW変換	15.7	122
4	あり	62.8	10
5	Single VM	18.1	117
6	Single VM	43.1	39
7	Job	54.8	15
8		75.0	8
9		81.4	5
10		25.9	98
11	EDOS-MSO ^{#3}	6.2	413
12	常駐VM	5.3	311
13		4.9	477

*1 回/10⁶ 実行ステップ

*2 Single Virtual Storage OS

*3 Real Storage OS

4.3.3 高速CCW変換

VMCPのI/Oシミュレーションオーバーヘッドの中で、CCW変換は主要なオーバーヘッド要因を占める。CCW変換オーバーヘッドの主要因は、

(1) 次のCCWへアクセスするときやCCWデータバッファアドレスを実アドレスに変換するとき、常にページ境界を越えることを考慮しなければならないこと、

(2) CCW内の制御ブランチ (Transfer In Channel) が複雑な場合も考慮してあるため、その処理が複雑なこと、
である。これに対して、図4に示す常駐VMにおいては、

(1) そのメモリが実メモリ上に連続的に常駐化されているため、ページ境界を気にかける必要がないこと、

(2) 仮想CCWを実CCWに変換するのに、単に変位αのみを加算すれば良く、アドレス変換テーブルを検索する必要のないこと、

(3) 単純なCCW内のブランチのみを処理対象とし、複雑な場合は汎用のCCW変換ロジックを通すこと、

等によりCCW変換の高速化を行うことができる。

図5に汎用I/OシミュレーションとFast I/Oシミュレーションとの関係を示す。

4.4 性能向上結果

常駐VMとFast I/OはHITAC Mシリーズ上のVMSで開発され、その性能がM-180上で測定された。性能測定ツールとしては以下のものを用いた。

(1) ソフトウェアモニタ：このモニタは仮想特権命令、割り込み等のイベント発生回数を監視する。

(2) ハードウェアモニタ：これはCPUサービス時間、命令実行回数、not in Buffer

カウンタ等をモニタする。

Fast I/OはI/Oシミュレーションの正常処理のみを対象とするので、その適用率が問題となる。そこで、その適用率を以下の式で定義する。

$$\text{Fast I/Oの適用率} = \frac{D}{C} \times 100 (\%)$$

C = 仮想I/Oの全発行回数
D = Fast I/Oによって完全に処理される仮想I/Oの回数

ベンチマークジョブとして表1に示す13個のジョブを使用した。

4.4.1 Fast I/Oの適用率

適用率は70%~100%だった。適用率減少の要因はI/O装置のユニットチェック(I/Oエラー及びEnd of Cylinder等の境界事象)であった。

4.4.2 スーパーバイサステート時間の減少

図6.にスーパーバイサステート時間の減少結果を示す。適用率90%以上では、50%~60%だけ減少した。VMSのCPUオーバーヘッドは約40%~50%だけ減少した。

4.4.3 VMCPのNot In Buffer Ratioの減少

図7.にVMCPのNot in Buffer Ratioの減少結果を示す。適用率が99%より大きいところでは、Not in Buffer Ratioの減少が認められる。これは、Fast I/Oのプログラムモジュールの局所性の良さを示すものである。

4.4.4 Fast I/Oでのメモリ増加

Fast I/Oは、処理オーバーヘッドを削減するために汎用I/Oに比べ多くの制御ブロックを使用する。そのために、システム内の200台の装置に対して約20%のVMCPのメモリ増が必要である。

以上の性能向上によりVMSのCPUオーバーヘッドを約半分とすることができた。

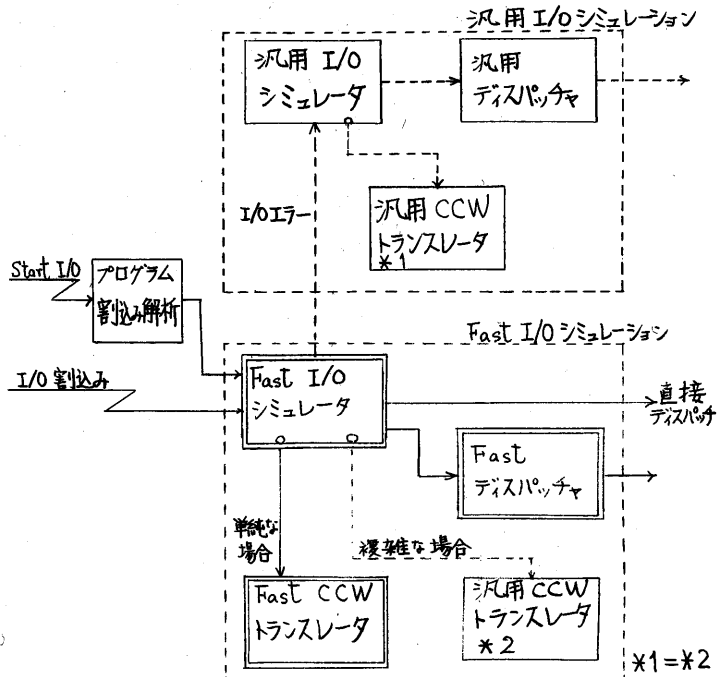


図5 汎用I/OシミュレーションとFast I/Oシミュレーションとの関係

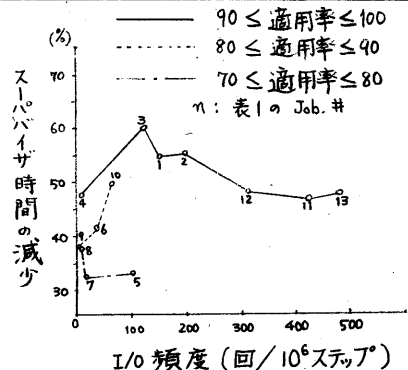


図6 スーパーバイサステート時間の減少

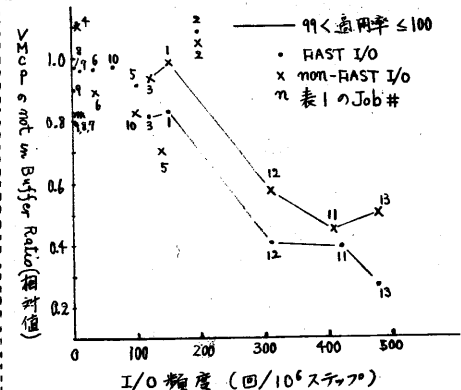


図7 VMCPのNot-in Buffer Ratioの減少

5. VMSの機能上の改良 5.1 従来のVMの機能上の問題点

VMは実計算機と同等の機能を持つが、種々の面で実計算機と異なる。このためオンラインシステム移行用には以下の問題点が発生した。

- (1) タイミングの誤差：VM上のOSは、自分がVM上で走行していることは知らないため、VMCPにより停止させられている時間(他のVMの走行中、特権命令のシミュレーション中等)は、認識することができない。このため、実時間監視や、CPU時間計測に悪影響を及ぼす。
- (2) システムコンソール操作：VMCPのコンソールとOSのコンソールとは通常共有されるが、両者の画面をどのように共存させれば良いか。またコンソール障害のとき両コンソールの交代をどのようにサポートするか。
- (3) システムの開始、再開始：実計算機下でのオンラインシステムでは、システムの開始、再開始について種々の高速化・単純化の機能が備わっているが、これがVMS環境下で使用できるか。また、新たな機能が必要となるか。
- (4) エラ及び保守情報：エラー・保守情報は各OSにより別々に収集されることになるため、管理が不便となる。
- (5) 運用：新旧2つの運用となると、新OS、旧OS、VMCPという3種類のオペレーティングシステムの運用が行なわれるので、オペレータの負担が大きくなる。

これらの問題点に対して以下のように対応した。

5.2 機能改良方法

- (1) タイミング：VMにおけるタイマ(仮想タイマ)は、VMの走行時及び、VMのセルフウェイト時(VMが自分自身の要因でWaitした時)だけ進めることにする。VM上のOSは、この仮想タイマを使用することにより正しいCPU時間を計測することができる。図8.に仮想タイマの進行状況を示す。

VM上のOSが実時間として、この仮想タイマを使用すると、時間に遅れが発生するので、OSとVMCPの連携により、遅れを補正するようにした。

さらにPCI(Program Controlled Interruption)やタイマ割込みの遅れを防ぐため、それを発生させてVMを優先的にスケジュールすることにした。

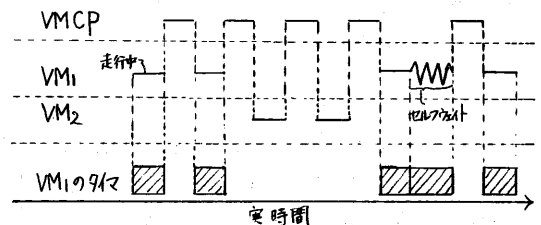


図8. 仮想タイマの進行状況

- (2) システムコンソール操作：VMCPの画面とOSの画面とは切換式とし、プログラムファンクションキーにより切換え可能とした。また、コンソール障害により、OSがコンソール交代を行なった時は、OSとの連携機能により、同時にVMCP側もコンソール交代を行なうことができるようにした。
- (3) システムの開始、再開始：VM下OSの持つ再開始機能は、そのままVM下でも使用可能とし、さらに図9.に示す自動リスタート機能をサポートした。図9の説明：VMCPのIPL後、自動的に複数のVMを起動し、自動的にそのOSをIPLする。これにより、操作性を向上している。また、システムエラー発生時には、VMCPのSYSRESにエラー情報を一括管理・記録し、

自動システムリスタートが動作する。

(4) エラ・保守情報：VM上のOSがエラ情報を記録しようとする場合特定のSVC (Super Visor Call) 命令を発行する。VMCP側は、このSVC命令をトラップし、VMCPのSYSRES上にエラー情報を一括管理記録する。

(5) 運用：操作をできる限り自動化することにより操作誤りを少なくするとともに、VMCPコマンド投入禁止モード等設けることによりオンライン運用中の操作誤りを減らすようにした。

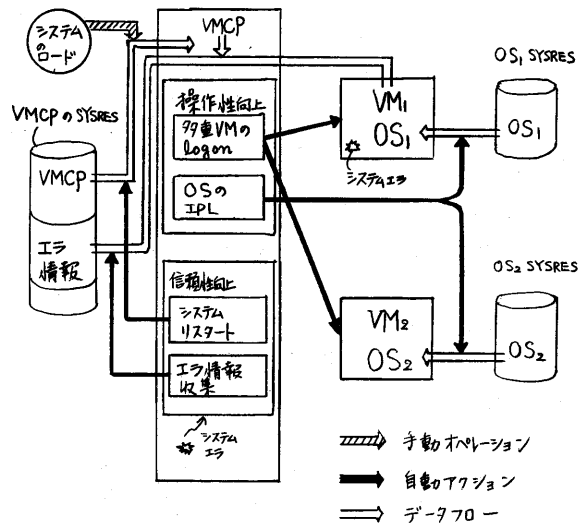


図9. 操作性及び信頼性向上

以上の機能上の改善を行なうことにより、VMSを多重オンラインシステムとして適用できるようになった。しかし、運用面での両システムの統一性や、両システム間のファイル共有、回線端末共有等の残された問題点が存在する。

6. VMSによるオンライン移行例

表2にシステム移行形態を示す。

表2 システムの移行

#	ユーザ	システムの移行 (移行前)	移行方法	VMS 運用 期間	システムの移行 (移行後)	備考
1	A	*1 8700.EDOS-MSO(O) *2 M170.VOS3(B) *3 同等	*1 *5 M280.VMS *6 VOS3(B)	4~5年	*8 M280.VOS3(O,B)	*1 HITAC M CPU *2 Real OS
②	B	8700.EDOS-MSO(O)	*1 M180.VMS *3 EDOS-MSO(O) *4 VOS2(O) *5 VOS3	5年以上	*8 左に同じ	*3 オンライン *4 パッチ *5 Virtual Machine System
③	C	*1 8450.EDOS-MSO(O)	M180.VMS *6 EDOS-MSO(O) *7 VOS2(O)	4~5年	*8 M180.VOS2(O)	*6 Virtual Storage OS
④	D	8450.EDOS-MSO(O)	M170.VMS *7 EDOS-MSO(O) *8 VOS2(OT)	5年以上	*8 左に同じ	*7 On-line Develop- ment /Test
5	E	8700.EDOS-MSO(O)	8700.EDOS-MSO(O) M180.VOS2(OT→O) (ハードウェア共有)	ハードウェア 共有: 半年	M180.VOS2(O)	*8 CPU グレードアップ の手配
6	F	M180.VOS2(B)	M180.VMS *9 VOS2(B) *10 VOS3(BT→B)	1年以下	VOS3(B)	

○ VMS下の多重オンラインシステム

表2.でユーザB, C, DがVMS下の多重オンラインシステムであり, 本報告で述べた性能向上, 機能向上を実現したVMSにより実ストレジOS・EDOS-MSOオンラインと仮想ストレジOS・VOS2(またはVOS3)オンラインの2システムを同時運用している。このうち, ユーザB, Dは, 旧システムのコンバージョンはやめ, 将来に渡って多重オンラインのままで運用して行く予定である。このようになる, VMSによる多重オンラインは, 単なる過渡的な移行ツールではなく, 各種システムを統合統合する定常的なシステムと考えられる。

7. 結 論

VMSによる多重オンラインシステムが実現され, オンラインシステムの移行用に適用できるようになった。これは従来の移行方法と比較して以下の利点を有する。

- (1) 1台の実計算機下で, 旧オンラインシステムを運用しつつ, 新オンラインシステムの開発, テスト, 運用等を行なうことができる。
- (2) 旧業務プログラムの長期間にわたるコンバージョンにより, コンバージョンコストの短期集中を防止することができる。

VMSによる多重オンラインをサポートするために, 以下のVMSの性能上, 機能上の改良を行なった。

性能上の改良

- (1) 高性能VMとして常駐VM方式を導入した。
- (2) VMCPのI/Oシミュレーションオーバーヘッドを削減するために, 高速入出カシミュレーション方式をサポートした。これにより, VMSのCPUオーバーヘッドがV=R VMと常駐VMに対して40% ~ 50% 削減された。

機能上の改良

- (1) タイミングエラーの補正
- (2) VMコンソール障害時の交代コンソールサポート
- (3) 自動システムスタート及びリスタート
- (4) エラー保守情報のVMCPによる集中管理

以上の性能改善, 機能改善によりVMSを多重オンラインシステムに適用できるようになった。

8. 謝 辞

著者等は, 日立製作所ソフトウェア工場山本, 小平両主任技師, システムプログラム部担当者各位, 同システム開発研究所高崎研究員等の, 有益な討論, コメントに感謝いたします。

9. 参考文献

- (1) H. Umeno, K. Ohmachi, et al. (1981). Development of a High Performance Virtual Machine System and Performance Measurements for it, Japan Information Processing Vol.4, No.2 July.
- (2) H. Umeno, K. Ohmachi, et al. (1981). On-line System Change-Over Using a Virtual Machine System, IFAC Real Time Programming Kyoto Japan.