

仮想マシン草創期

Beginning of Virtual Machine

大町一彦
東京農工大学

はじめに

20世紀半ばのメインフレーム創世記から見ると、現在の計算機の性能は100万倍以上向上し、処理コスト・計算コストは飛躍的に低下しました。それにより1台の計算機の使用効率向上という意識は薄れました。この意識の低下は、多くの企業システムにおいて、物理サーバが乱立する状況を生みだし、設置スペースの拡大、管理・保守コストの増大、セキュリティの低下、各物理サーバの利用率低下を招くに至りました。そのため、シンクライアントとブレードサーバ^{☆1}による新たな集中システムが生み出され、世の中のシステムはその方向に動きつつあります。ハイエンドのブレードサーバにより、従来の分散された物理サーバ群が統合されつつあり、そのための有効な手段として、再び、仮想マシンシステムが注目されています。すなわち、仮想マシンシステム技術は安全なシステム移行や統合的なシステム運用管理などに有効な手段として、メインフレームのみならずオープンシステムのサーバにおいて、今後とも活用される状況となっています。仮想マシンシステム技術は40年余りにCambridge Scientific CenterのR. Adairらにより創始された技術ですが、その応用性は永続的であり、計算機アーキテクチャの発展とともに、仮想マシンシステム技術もまた発展していくものです。

仮想マシンシステム技術は工学の問題であり、実システムにおける性能面や機能面から見た、実用性、応用性において常に新しい問題を提供してくれると考えます。計算機アーキテクチャやOSの技術がメインフレームからミニコンピュータ、パーソナルコンピュータへと移転していったように、仮想マシンシステム技術もまた同様に移転しつつあります。しかし、パーソナルコンピュータのアーキテクチャやOSにはメインフレームにない新技術や新応用・新展開があるように、パーソナルコンピュータの仮想マシンシステム技術においてもまた新技術・新展開があるといえます。たとえば仮想マシンシ

ステムにおけるネットワークやクラスタなどの中にそれらを見ることができます。それらが今後、盛んに、研究・開発されていくと思われます。

現在、SOA (Service Oriented Architecture) 技術、SaaS (Software as a Service) モデル等の進展に伴い、機能のセキュリティ境界を明解にしつつ、処理能力やストレージなどのリソースを効率よく利用するための広い意味の仮想化技術が注目を浴びています。その中で、仮想マシンシステム技術は計算機システム本体の仮想化であり、広義仮想化技術の主要な構成要素であります。

本連載では、最も基本的なプラットフォームであるマシン自体の仮想化を行う技術のルーツ、および最新動向についてご紹介します。これが、仮想化技術の循環的な発展に、すこしでもお役に立てれば幸いです。

メインフレームの黎明期

メインフレーム (汎用大型計算機) は、周知のように、1945年の秋、John William MauchlyとJohn Adam Presper Eckert Jr. と、その仲間十余名により発明されました。有名なENIACがそれです。その大きさは三寝室付きのアパート位の大きさであり、真空管を約18,000本使用したということです。その運用には大型の放送局が消費するに匹敵する電力を必要としたということです。しかし、当時のどのような機械式計算機よりも1,000倍速く、毎秒5,000回の加算ができたそうです。ENIACの発明は最高軍事機密であり、弾道ミサイルの設計や、核兵器の設計に使用されたということです。このあたりの開発秘話はパーソナルメディア社の「エニアク 世界最初のコンピュータ開発秘話 スコット・マッカートニー著/日暮雅通訳」に詳しく載っています。

当初は0,1のパターンの機械語でプログラムを手で組んで動かしていましたが、1955年~1960年くらいにかけて、アセンブリ言語やFORTRAN等のコンパイラ言語が使用されるようになりプログラムローダや常駐モニタ (小型の初期OS) 等が考案され、複数のプログラムを一括して処理するバッチ処理が行われるようになりました。当時は、計算機の価格 (MIPS (Million Instructions Per Second) 単価) が今より数桁高かった

^{☆1} 1つの計算機システムが1枚の基板 (ブレード: blade と呼ばれる) に納められ、その基板が数枚~数十枚あり、それが1台の筐体に統合された形のサーバ。

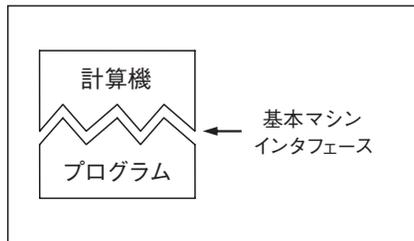


図-1 計算機上に1つのプログラム

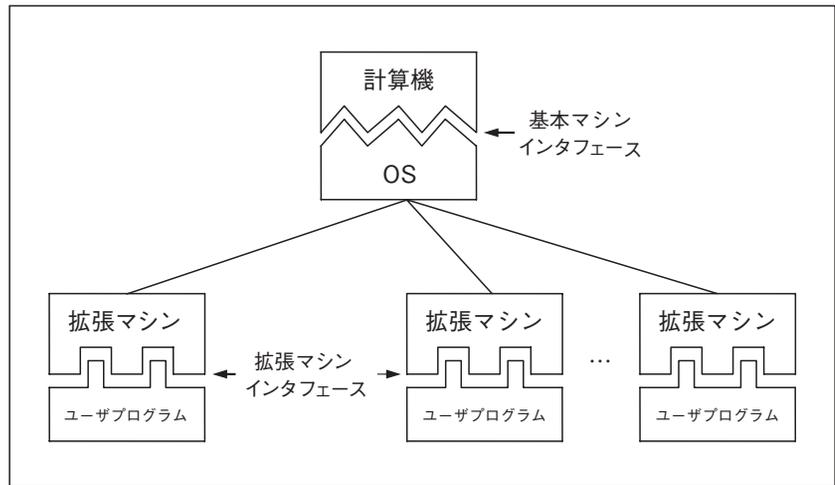


図-2 OSによるマルチプログラミング

こともあり、計算機の使用効率の向上が至上命題でした。計算機上に1つのプログラムが実行される場合（図-1）、中央処理装置に比べて極端に処理速度の遅い入出力装置をプログラムが使用すると、その入出力動作の終了まで中央処理装置の処理は待たされ、計算機の使用効率は大幅に低いものとなっていました。

【マルチプログラミング・オペレーティングシステムの登場】

複数のプログラムの同時走行を可能とする、マルチプログラミング機能を有するオペレーティングシステム（Operating System, OSと略記）と呼ばれるシステムプログラムが、この問題を大幅に改善しました^{1), 2)}。このマルチプログラミングシステムでは、各プロセス^{☆2}が独立にメモリを要求するため必然的にセグメンテーション/ページングによる仮想記憶の方式をもたらしました³⁾。

このようなマルチプログラミング方式、仮想記憶方式を支援する計算機ハードウェアは複数の保護レベルを持っており、OSは最高位の保護レベルで動作し、一般ユーザは、より低い保護レベルで動作していました。その保護レベルに応じて使用可能な命令セットも異なっていました。さらに、同じ命令でも、そのレベルによって機能が異なることもありました。記憶装置の特定領域や入出力装置などの計算機資源の直接使用を要求する命令（これを特権命令と呼ぶことにします）セットはシステム全体の安全性・信頼性に影響を及ぼすためOSしか使用できません。OSはこれらのすべてのレベルの命令セット（基本マシンインタフェース）を使用してマルチプログラミング機能をユーザに提供します。ユーザプログラ

ムは主に四則演算・比較・分岐等を行う命令（これを非特権命令と呼ぶことにします）セットしか使用できませんが、ユーザプログラムが記憶装置の特定領域や入出力装置などの計算機資源を使用するときはOSにシステムコール命令を発行し、そのサービスを要求します。すなわちOSは、非特権命令とシステムコール命令とを合わせたインタフェースである拡張マシンインタフェースをユーザプログラムに提供することにより、複数ユーザプログラムの同時走行を可能にしました（図-2）。

このような、多重の拡張マシンインタフェースは、ユーザから見た計算機の仮想化であり、これによりさまざまなユーザプログラムが同時に動作し、また、そのための種々の開発支援ツールも用意されて、その結果、ユーザプログラムの生産性や計算機の利用効率は飛躍的に向上しました。

しかし、特権命令を含む基本マシンインタフェースを用いるOSのようなプログラムについては、計算機上に1つしか走らせることはできません。このため、OSの開発・保守などで計算機を使用する際には、低い使用効率にならざるを得ませんでした。そこで、複数個のOSを走行させることにより、OSの開発・保守などにおいても、高い計算機使用効率を達成することのできる仕掛けの実現が望まれました。

【仮想マシン(Virtual Machine, VMと略記)の出現】

このような背景から Cambridge Scientific Center の R. Adair らにより、複数個の基本マシンインタフェースを実現する仮想マシンシステム（Virtual Machine System, VMSと略記）が考案されました⁴⁾。ハーバード大学 Applied Physics の大学院の同級生であった Robert P. Goldberg と Jeffrey P. Buzen は、このVM技術を研究し、そのアーキテクチャ上の原理や実装技

☆2 プログラムの実行単位を含み、計算機資源の割り当て単位をプロセスと呼びます。

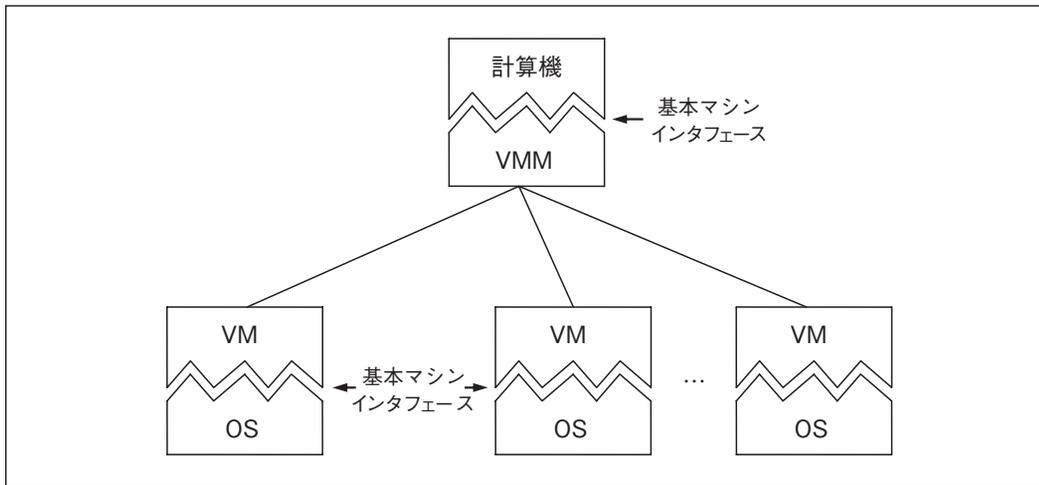


図-3 仮想マシンシステム

術やハードウェアパーチャライザによる性能向上方式を研究・開発しました(図-3)^{5), 6)}。特に文献6)はRobert P. GoldbergがVMの原理, 定義, 実装例, 性能向上方式, 応用例を幅広く纏めた解説論文であり, VM研究や応用にかかわる研究者や技術者は, 一度は読まれることをお勧めします。筆者は, このVM技術の研究・開発者であるRobert P. Goldbergの研究室に留学してVM技術を学び, 帰国後に, その研究を継続して, 実際の企業におけるVM技術の工業化・実用化に貢献することができました。

VMSでは, 仮想マシンモニタ(Virtual Machine Monitor, VMMと略記)が複数個の基本マシンインタフェース(VM)を実現し, それぞれのVM上にOSを走行させることにより, 1台の実計算機上で複数のOSを同時に走行させることが可能となりました。

VM上のOSから発行される非特権命令についてはそのまま計算機で直に実行されます。特権命令についてはその実行により, 同時に走行している他のVMに影響を与えるため, そのまま実行させるわけにはいきません。そこで, VMMが他のVMに影響を与えないようにその命令のシミュレート処理を行います。その特権命令が入出力装置の起動であった場合, マルチプログラミングOSが他のユーザプログラムに制御を移すのと同じような意味で, VMMは他のVMに制御を移し, 中央処理装置の有効利用を図ります。また, 通常, 特権命令のシミュレート処理には数百ステップの処理が行われます。この処理の高速化を図るため, VM Assist⁷⁾と呼ばれるファームウェアも開発されました。

おわりに

次回は, 仮想マシンがどのようにして実用段階に入っていたか, 仮想マシンの性能がどのようにして向上さ

れていたか, 仮想マシンがどのように使われるようになっていったかについて, メインフレームの仮想マシン技術, パソコンの仮想マシン技術を織り交ぜて解説したいと思います。

参考文献

- 1) Dijkstra, E. W. : The Structure of the 'THE'-multiprogramming System, Commun. ACM, 11, 5 : pp.341-346 (1968).
- 2) Mealy, G. H. : The Functional Structure of OS/360, IBM Systems Journal, Vol.5, No.1, pp.3-11 (1966).
- 3) Green, P. : Multics Virtual Memory - Tutorial and Reflections. <http://www.multicians.org/exec-env.html>
- 4) Adair, R., Bayles, R. U., Comeau, L. W. and Creasy, R. J. : A Virtual Machine System for the 360/40, Cambridge Scientific Center Report No. G320-2007 (May 1966).
- 5) Goldberg, R. P. : Architectural Principles for Virtual Computer Systems, Ph. D. Thesis, Division of Engineering and Applied Physics Harvard University, Cambridge, MA (1972).
- 6) Goldberg, R. P. : Survey of Virtual Machine Research COMPUTER, IEEE, pp.34-45 (June 1974).
- 7) MacKinnon, R. A. : The Changing Virtual Machine Environment : Interfaces to Real Hardware, Virtual Hardware, and Other Virtual Machines, IBM System Journal, Vol.18, No.1, pp.18-46 (1979).

(平成19年6月18日受付)

大町一彦(正会員)
k.ohmachi@ieee.org

東京農工大学大学院技術経営研究科技術リスクマネジメント専攻教授, 独立行政法人海洋研究開発機構地球シミュレータセンター特任上席研究員兼務, 電子情報通信学会, IEEE各会員。