仮想化環境における CPU 資源管理

追川 修一
筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻

1 はじめに

近年，計算資源を有効利用したり，またよりセキュアなシステムの構築を可能にするために，複数のオペレーティングシステム（OS）環境を 1 台のマシンで提供することのできる Xen [1] や UML [2] などの仮想化環境が再び注目を集めている．一方で，実時間性が要求される組み込みシステムなどでは，プロセスに実時間スケジューリングクラスの優先度を割り当て使用することも多い．その場合，固定の高い優先度でスケジューリングされるプロセスの実行が，通常の優先度で実行されるシステムサービスを提供するプロセスの実行を妨げる可能性が出てくる．安定した実行環境構築のため，プロセスに割り当てられる CPU 時間を保護するための機能に対する要望は依然としてある．

本論文では，組み込みシステム向けのプロセッサでも普及の兆しがあるマルチコアプロセッサを前提として，仮想化環境においてプロセスの CPU 利用時間を制御可能にする CPU 資源管理について述べる．マルチコアプロセッサを前提とすることで，仮想化環境における CPU 資源管理を単純化することができる．

2 CPU 資源管理

本章では，CPU 資源管理について述べる．ここで，実機上の実行環境をホスト環境，仮想マシン上の仮想化実行環境をゲスト環境と呼ぶ．

2.1 ホスト環境における CPU 資源管理

CPU 資源管理を提供する機構として Linux/RK を構成する Portable RK [3] がある．Portable RK（以下単に RK）は，実時間スケジューリングの周期実行モデルに基づき，CPU 時間保護を実現する．プロセスが使用できる CPU 時間は，周期実行モデルに基づき，周期 T のうちの実行時間 C という二つのパラメータのペア $\{C, T\}$ で表現される．この時

$$\frac{C}{T}$$

が使用可能な CPU 時間の割合となる．n 個のプロセスに CPU 時間保護を提供する場合，$1 \leq i \leq n$ の i について $\{C_i, T_i\}$ が定義され

$$\sum_{i=1}^{n} \frac{C_i}{T_i}$$

が保護される CPU 時間の割合の合計となる．この合計がいくつになるまで CPU 時間保護が可能かは，スケジューリングポリシに依存する．

実行時に CPU 時間保護を実現するため，RK は受付制御，スケジューリング，CPU 使用時間の課金，プロセスの強制中断，周期的な CPU 時間の補充といった機能を提供する．

2.2 ゲスト環境における CPU 資源管理

ホスト環境における CPU 資源管理を単純にそのままゲスト環境に適用することはできない．ゲスト環境では，図 1 に示すように，2 つのレベルでスケジューリングが行なわれる．仮想マシンモニタ（VMM）またはホストカーネルによりゲスト環境は作成され，スケジューリングされる．一つのゲスト環境において，ゲスト OS

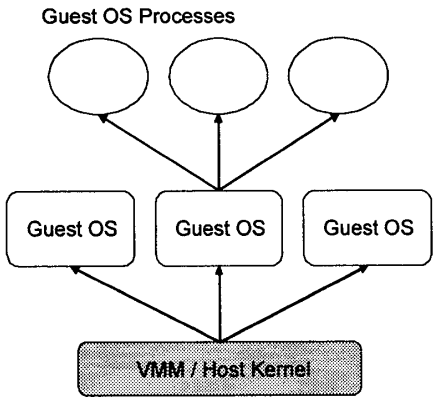

図 1 ゲスト環境における CPU 資源管理

CPU Resource Management in Virtual Execution Environments.
Shuichi Oikawa, Department of Computer Science,
University of Tsukuba

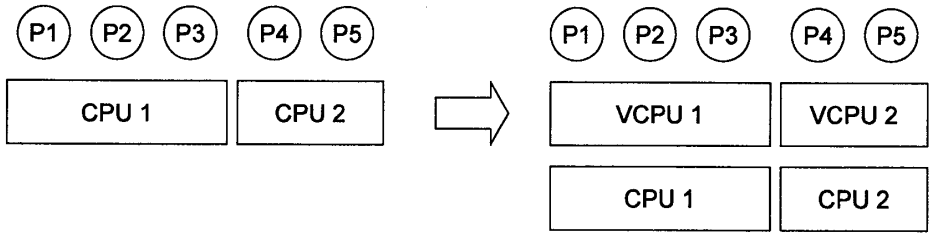

図2 マルチコアプロセッサにおける実CPUと仮想CPUの対応

が実行され，ゲストOSのプロセスがスケジューリングされる．従って，ゲストOSのプロセスに対するCPU資源管理を行なうためには，そのホスト環境においてゲスト環境に対するCPU資源管理を行なう必要があるため，ホスト環境におけるCPU資源管理をそのままゲスト環境に適用することはできない．ホスト環境においてゲスト環境に対するCPU資源管理を行なうにしても，ホスト環境からゲストOSのプロセスを直接スケジューリングすることはできないため，異なる保護ドメインにまたがった階層的なCPU資源管理を行なうのも難しい．

そこで，マルチコアプロセッサを前提とすることで，擬似的にゲスト環境が直接CPU資源を管理できるようにする．このCPU資源管理手法を図2に示す．図2左側はホスト環境が直接CPU資源を管理する場合である．この場合，ホストOSがCPU1，CPU2を管理し，CPU1上でプロセスP1〜3，CPU2上でプロセスP4，5を実行する．OSがCPU上のプロセススケジューリングを直接管理しているため，前節の資源管理機構によりCPU時間保護が可能である．

図2右側が提案手法である．ホスト環境はゲスト環境に仮想CPUとしてVCPU1，VCPU2を提供する．ゲストOSは，VCPU1，VCPU2を管理し，VCPU1上でプロセスP1〜3，VCPU2上でプロセスP4，5を実行する．仮想CPUから実CPUへの対応が時分割で行なわれる場合は，前述のようにゲスト環境でのCPU資源管理は意味をなさない．しかしながら，仮想CPUから実CPUへの対応が固定的であれば，それはゲストOSが直接実CPUを管理しているのと同じであるため，ゲスト環境でのCPU資源管理は有効になる．また，ホスト環境は仮想CPUから実CPUへの対応を管理するだけで済み，仮想化環境におけるCPU資源管理を単純化することができる．

3 実装

Linux/RK [3] をもとに，CPU資源管理のみを行なうように簡素化されたCABI [4] が，Linux 2.6のSMPカーネルに対応しているため，これを用いる．仮想環境としてはXen [1] を用いて，実装を行なっている．

4 まとめ

本論文では，プロセスのCPU時間保護を可能にするCPU資源管理を仮想化環境で実現するにあたって，組み込みシステム向けのプロセッサでも普及の兆しがあるマルチコアプロセッサを前提とすることで，CPU資源管理を単純化することができることを示した．

謝辞

本研究の一部は，文部科学省リーディングプロジェクト「e-Society基盤ソフトウェアの総合開発」の助成による．

参考文献

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. In *Proceedings of the 19th ACM Symposium on Operating System Principles*, pp. 164–177, October 2003.

[2] J. Dike. A User-Mode Port of the Linux Kernel. In *Proceedings of the 2000 Linux Showcase and Conference*, October 2000.

[3] S. Oikawa and R. Rajkumar. Portable RK: A Portable Resource Kernel for Guaranteed and Enforced Timing Behavior. In *Proceedings of IEEE Real Time Technology and Applications Symposium*, pp. 111–120, June 1999.

[4] M. Sugaya, S. Oikawa, and T. Nakajima. Accounting System: A Fine-Grained CPU Resource Protection Mechanism for Embedded System. In *Proceedings of the Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, pp. 72–84, April 2006.