

仮想計算機におけるメモリ管理のアウトソーシングの提案

戸 祭 要† 小 沢 健 史†† 新 城 靖††
佐 藤 聡†† 中 井 央††† 板 野 肯 三††

1. はじめに

現在, VMWare Workstation や Linux KVM¹⁾ 等の TypeII の仮想計算機モニタ (Virtual Machine Monitor, VMM) が広く利用されている. TypeII の VMM はホスト OS から確保したメモリをゲスト OS に仮想的なメモリとして提供し, ゲスト OS は提供されたメモリを物理メモリと認識しアクセスする. しかし, VMM が提供したメモリをホスト OS がページアウトしてしまうと, ゲスト OS はそのメモリにアクセスできない. そのため VMM は, しばしば確保したメモリをロックする. ロックされた場合, ホスト OS はアクセス頻度が低いゲスト OS が使用しているメモリをページアウトできない問題がある. また, Linux KVM などの VMM ではゲスト OS に割り当てるメモリ量は, ゲスト OS の起動時に決定する必要があり, 起動後はそのメモリ量を変更できない問題もある.

そこで本研究では, これらの問題を解決する方式としてゲスト OS からホスト OS へメモリ管理機能のアウトソーシング²⁾を提案する. これにより, ゲスト OS が利用しているメモリをホスト OS がページアウト可能にする. また, ゲスト OS の起動中に動的に割り当てるメモリ量を変更することを可能にする.

2. メモリ管理機能のアウトソーシング

Linux KVM をはじめとする TypeII VMM では, ゲスト OS の起動時に VMM がホスト OS から仮想メモリを確保する. ゲスト OS は VMM から提供された仮想メモリを管理する. VMM は, この時ホスト OS にページアウトされないように確保したメモリをロックすることがある. しかし, この手法には1章で述べた問題がある.

これに対して本研究では, ゲスト OS で行っていたメモリ管理をホスト OS に行わせる方式を提案する.

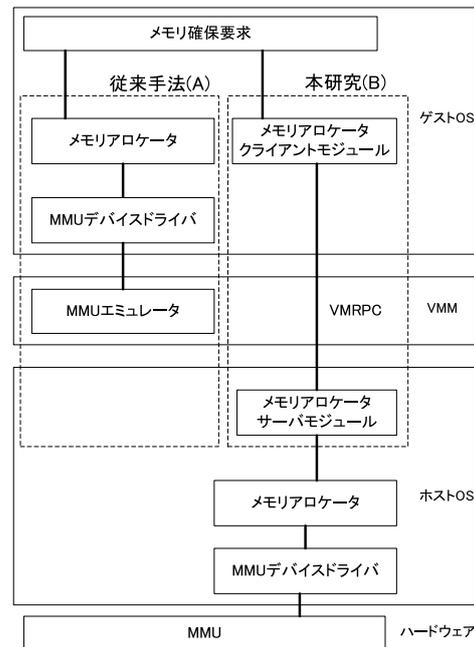


図 1 メモリ確保の流れ

我々はゲスト OS からホスト OS に処理を委譲する手法を, アウトソーシングと呼んでいる. この手法を用いて, ゲスト OS のメモリ管理機能 (メモリアロケータ) をホスト OS に委譲する. これにより, ゲスト OS のメモリをホスト OS がページイン・ページアウトすることを可能にする. また, ゲスト OS 起動後の動的なメモリ量の増減も可能にする.

具体的には, ゲスト OS のカーネルを書き換え, メモリ管理機能をホスト OS に委譲するモジュール (クライアントモジュール) を追加する. クライアントモジュールは, ゲスト OS からのメモリ確保・解放要求などを受け取り, サーバモジュールに要求を発行する. ホスト OS には, クライアントモジュールに対応するモジュール (サーバモジュール) を追加する. サーバモジュールは, クライアントモジュールからの要求を受け取り, メモリ確保・解放などを行う. また, クライアントモジュールを通じて, ゲスト OS のメモリ管

† 筑波大学第三学群情報学類
†† 筑波大学システム情報工学研究科
††† 筑波大学図書館情報メディア研究科

理を行う。両モジュール間で遠隔手続き呼び出しを行うために、VMRPC²⁾を使用する。

OSが行っているメモリ管理のうち、ゲストOSによるメモリ確保について述べる。一般的な仮想計算機では、ゲストOSは、VMMによって提供された仮想メモリからメモリを確保する(図1のA)。発行されたメモリ確保要求は、ゲストOSが管理する仮想メモリから確保される。

一方本研究では、ゲストOSで行っていたメモリ管理をホストOSに行わせる(図1のB)。発行されたメモリ確保要求は、ゲストOS内のメモリアロケータのクライアントモジュールが受け取り、ホストOS内のメモリアロケータのサーバモジュールに対して要求を発行する。

3. 関連研究

Linux KVMにおいて、ホストOSがゲストOSの管理下のメモリをページアウトするための手法として、MMU notifier³⁾がある。この手法では、ホストOSがVMMにゲストOS管理下のメモリをページアウトしたことを通知し、VMMは通知に従って該当するページへの参照を無効化する。これにより、ゲストOSがページアウトされたメモリにアクセスすることが無くなり、ホストOSがゲストOS管理下のメモリをページアウトすることができる。本研究は、ゲストOSのメモリ管理機能をアウトソーシングすることにより、ゲストOSが管理しているメモリをホストOSで一元的に管理するという点で異なる。

ホストOSからゲストOSが管理しているメモリ量を動的に制御する手法として、VMware ESX Serverなど各種VMMに実装されているballooning⁴⁾がある。この手法では、まずゲストOS内のballoon driverがゲストOSからメモリを確保する。balloon driverが確保したメモリをVMMが他のVMへ割り当てることにより、起動中のVMのメモリ量を動的に制御することができる。本研究は、ゲストOSのメモリ管理機能をアウトソーシングすることにより、ホストOSのメモリ管理機能によってVMごとに割り当てるメモリ量を変更する点で異なる。

4. おわりに

本稿では、仮想計算機のゲストOSにおけるメモリ管理機能のアウトソーシングを提案した。これにより、ゲストOSとホストOSで重複したメモリ管理機能を廃止し、ホストOSによる一元的で効率のよいメモリ管理を行うことを可能にする。今後は実装を完了させ、

機能と性能の評価を行う。

参考文献

- 1) Avi Kivity. KVM : the Linux virtual machine monitor. In *OLS '07: The 2007 Ottawa Linux Symposium*, pp. 225–230, July 2007.
- 2) Hideki Eiraku, Yasushi Shinjo, Calton Pu, Younggyun Koh, and Kazuhiko Kato. Fast networking with socket-outsourcing in hosted virtual machine environments. In *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*, pp. 310–317, New York, NY, USA, 2009. ACM.
- 3) Andrea Arcangeli. Integrating kvm with the Linux memory management. In *KVM Forum*, June 2008.
- 4) Carl A. Waldspurger. Memory resource management in VMware ESX server. *SIGOPS Oper. Syst. Rev.*, Vol.36, No.SI, pp. 181–194, 2002.