

ホスト型仮想計算機におけるタイマ精度向上手法の提案

小沢 健史¹ 斉藤 剛² 英樂 英樹¹
新城 靖¹ 佐藤 聡¹ 中井 央³ 板野 肯三¹

1. はじめに

個人利用に適した仮想計算機の形態として、VMware Workstation に代表される Type II の仮想計算機モニタ(以下、VMM)が存在する。Type II の VMM は、一般的に以下のような特性をもつ:

- A) エミュレートしたタイマをゲストに提供している[1].
- B) VMM がホスト OS から見て単一のユーザプロセスにすぎない.

よって、Type II VMM は、Xen に代表される Type I VMM[1]と比較し、タイマ精度が劣るという問題がある。タイマはいわばOSのペースメーカーであり、これがずれるとゲストOS上で行われるタスクのスケジューリングや、TCP のタイムアウト処理などがずれてしまう。この問題は、ゲストOS上で動画や音楽といったマルチメディアファイルを閲覧するなどのリアルタイム性が求められる処理が行われた場合に顕著に表れる。また、タイマのずれが原因で仮想計算機上の現在時刻がずれるため、cron を用いてスクリプトを動かすなどといった特定時刻に決まった処理を行う際に、不都合が生じる。

そこで本研究では、仮想計算機のタイマ精度を向上させるために、2つの方式を提案する。

- A) 本来ゲストOS上で処理するはずの処理をホストOS側のタイマを利用して処理を行うアウトソーシング方式[3].
- B) Type I のVMMで用いられているVMMクロックソース方式[4].

これら2つをType II のVMMに適用し、評価を行う。

2. 問題提起

2.1 ホスト-ゲスト間の断絶による問題

Type II のVMMは、タイマのエミュレーション

を行い、これを元にOSなどの基本ソフトウェアが動作する構成をとっている。だが、この方式には、大きな問題点がある。それは、ゲストOSが「どの程度自分の時計がずれているか」を検知できない点にある。通常、ゲストOSはホストOSの動作を検知不可能であるため、ゲストOSはホストOSの時計を見ることができず、誤差を検知することができない。

2.2 VMMがホストOSにとって単なるユーザプロセスであることに起因する問題

2.1節でも述べたが、Type II のVMMにおいては、ゲストOSはホストOSの状態を知ることができない。さらに、Type II VMMはホストOSから見れば単なるユーザプロセスのうちの1つであるから、ホストOSに大きな負担がかかると、VMMのスケジューリング優先度が相対的に下がり、タイマがずれてしまう。

この問題の対策として、Linuxにおけるniceコマンドなどを用いてVMMプロセスの優先順位を上げることが考えられる。しかし、タイマ処理のみの優先度を上げれば十分であるのに、VMM全体の優先度を上げてしまうという問題がある。

3. 提案手法

本研究では、2章で述べた問題を解決するため、アウトソーシング方式とVMMクロックソース方式の2つのタイマ精度向上方式を提案する。従来方式と提案方式の概要を、図1に示す。

3.1 タイマアウトソーシング

アウトソーシングとは、ゲスト側の高水準モジュールに与えられた要求をホストに投げるという手法である[3].

タイマに関してアウトソーシングを行うための一連の流れは、以下の通りである。まず、ゲストOS

¹ 筑波大学システム情報工学研究科

² 筑波大学第三学群

³ 筑波大学図書館情報メディア研究科

上で、タイマ依存モジュールと呼ばれる。タイマ依存モジュールとは、システムコール、ページャ、TCP プロトスタック、スケジューラなどの、カーネルタイマに依存するモジュールのことを指す。タイマ依存モジュールは Linux の `add_timer` 関数などのタイマモジュールの手続きを呼び出す。タイマモジュール内部では VMRPC が呼ばれ、VMRPC 経由でホスト OS のタイマモジュールが呼ばれる。最後に、ゲスト OS 上で戻り値を解釈する。`add_timer` 関数においては、指定された時間が経過すると、登録された手続きがホスト OS 上で呼び出される。すると、ホスト OS 上のモジュールがゲスト OS に対してイベントを送り、割り込みを生成する。ゲスト OS のモジュールはイベントを受け取り、元の手続きを呼び出す。

3.2 VMM クロックソース

VMM クロックソースは、Xen 上で動作する Linux で用いられている、VMM がハードウェアを参照し、時間管理を行う手法である[4]。Xen 上で動作する Linux 2.6.25 の場合、標準のハードウェアを参照するドライバを Xen 用の VMM を参照するドライバに置き換えることで、この方式を実現している。

VMM クロックソースを行うための一連の流れは、以下の通りである。まず、ゲスト上でタイマモジュールが呼ばれると、専用のクロックソースドライバに処理が移る。クロックソースドライバは、VMM に搭載されているクロックソースを見て、タイマを制御する。ここで、VMM のクロックソースは直接的にハードウェアを参照するか、もしくはホスト OS のカーネルを経由して間接的にハードウェアを参照する。

VMM クロックソースを用いる場合、タイマ割り込みハンドラはゲスト内で完結するため、ゲストのタイマモジュールを改変する必要はない。しかし、最初に述べた通り、ゲスト OS のクロックソースドライバを置き換える必要がある。

4. 関連研究

Network Time Protocol (NTP)では、クロックソースサーバを参照して時刻を合わせるためのプロトコルである[7]。ゲスト OS で NTP のクライアントを実行する方法は、タイマ精度向上の手法として有効であると予想される。しかし、ゲスト OS がネ

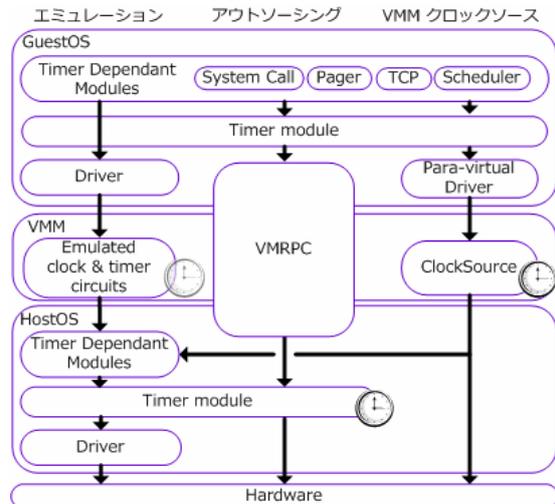


図 1 従来手法であるエミュレーション方式と提案手法であるアウトソーシング方式、VMM クロックソース方式の概要

ットワークに接続されていない状況では、NTP を使用することはできない。本手法は、いずれも処理がホスト OS - ゲスト OS 間で完結しているため、ネットワークが利用不可能な条件下においても適応可能である。

5. おわりに

本稿ではまず、現存する Type II の VMM 上におけるタイマの問題点について述べた。そして、その問題に対する解決策として、タイマのアウトソーシング方式と VMM クロックソース方式を提案した。

今後の課題として、双方の実装と評価が挙げられる。

参考文献

- [1] VMware, "Timekeeping in VMware Virtual Machines", 2005.
- [2] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield, "Xen and the Art of Virtualization", in Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03), pp. 164-177, New York, December 2003.
- [3] Hideki Eiraku, Yasushi Shinjo, Calton Pu, Younggyun Koh, Kazuhiko Kato, "Fast Networking with Socket-Outsourcing in Hosted Virtual Machine Environments", ACM Symposium on Applied Computing(2009) To Appear.
- [4] The Xen Team, "Xen Interface manual", Version 3.0, 2005.
- [5] RFC 1305, "Network Time Protocol Specification", 1992.