

## 仮想計算機モニタを用いた OS 動作状況観測手法

金城 聖<sup>†</sup>鈴木 和久<sup>††</sup>毛利 公一<sup>†††</sup><sup>†</sup>立命館大学理工学部 <sup>††</sup>立命館大学大学院理工学研究科 <sup>†††</sup>立命館大学情報理工学部

## 1 はじめに

近年、メモリの大容量化や CPU のマルチコア化などの計算機資源の増大によって、効率的な計算機資源の管理が求められるようになった。また、情報家電や、ブロードバンド通信を利用したコンテンツの配信サービスなどの普及によって、計算機に高い処理能力を要求するようになった。計算機の資源管理や処理能力は、OS への依存度が高いため、その性能が OS によって左右されることが多い。汎用的な OS では、多くの計算機資源やアプリケーションに適応できる方式が採用されているため、特定の計算機資源の利用やアプリケーションには適していない。特定の計算機資源やアプリケーションに適した処理を実現するためには、専用 OS の開発が必要となる。専用 OS の開発には、OS の動作状況を把握、分析する必要がある。また、OS の動作状況を観測することで、OS の開発以外に、システムの性能向上、不具合の検出、OS の性能評価などを行うことができる。

既存の動作状況観測手法には OS カーネル内部に観測機構を実装する手法 [1] と、IEEE1394 のメモリ参照機能を用いる手法 [2] がある。前者の手法は、アプリケーションが動作している際の OS の動作状況を観測することが目的とされている。しかし、カーネル内に観測機構を実装しているために、継続して観測する上ではオーバーヘッドが大きい。後者の手法は、OS の動作に対して影響を与えることなく観測することを目的としているが、詳細な状況観測を行うことが困難である。

以上の問題を解決するために、本論文では、仮想計算機モニタ (VMM) を利用した OS 動作状況観測手法を提案する。本手法を利用することで、OS に改良を加えることなく、詳細な動作状況観測が可能となる。

以下、本論文では、2 章で関連研究とその目的について述べ、3 章で本研究で提案する手法の概要と応用について述べ、4 章で提案手法の構成について述べる。

## 2 関連研究

既存の動作状況観測手法には、カーネル内部に観測機構を実装する手法と、IEEE1394 のホストコントローラが持つメモリ参照機能を用いた手法がある。

前者の手法は、OS が 1 つの命令を実行する毎に、割り込みを発生させる。割り込み発生後、制御を観測機構に移行することで実行された命令を保存し、OS の処理状況を観測する。この手法は、特定のアプリケーションの性能評価やシステムの不具合調査を目的とする場合が多い。そのため、継続的に動的な動作状況を観測する目的には適さない。また、カーネルに改良を加えることで実現しているため、本来の OS の動作を観測することが困難である。

後者の手法は、2 台の計算機を使用し、IEEE1394 のホストコントローラが持つ Physical Read という計算機のメインメモリを参照する機能を使用することで OS の状況を観測する。この手法を用いることで、割り込み禁止区間に関わりなく、CPU の処理を介さずに計算機の物理メモリのデータやデータの遷移を観測することが可能である。しかし、この手法では、IEEE1394 の通信速度の制限を受け、OS の詳細な状況変化の観測が見込めない。

## 3 本研究の概要と応用

本研究の目的は OS の動作に与える影響を必要最小限にとどめつつ、詳細な OS の動作状況を観測することである。前章で述べた手法ではその目的を満たすことが不可能である。そこで本論文では、VMM を用いた OS 動作状況観測手法を提案する。

## 3.1 VMM を用いた動作状況観測手法

VMM とは、CPU やメモリなどの計算機資源を仮想化し、OS における計算機資源を必要とする処理を仮想的な計算機資源を使用することで、複数の OS を同時に動作させるためのソフトウェアである。VMM を使用することで、OS は仮想的な計算機資源を実際の計算機資源と同様に利用して処理を行う。VMM は仮想的な計算機資源を管理し、実際の計算機資源を VMM が持つスケジューラによって VMM 上で動作している OS に割り付ける。この仕組みによって、OS は実際の計算機と同様に仮想計算機 (VM) で動作する。

本論文で提案する手法では、VMM を利用することで、OS が利用する CPU やメモリの動作状況を観測することが可能となる。VMM は CPU、MMU、ページテーブル、割り込みコントローラなどのハードウェアの仮想化を行うことで VM を提供する。VM 上の OS が仮想化されたハードウェアに対して、アクセスや処理を要求すると、VMM が持つスケジューラが実際の CPU、MMU、割り込みコントローラを割り付ける。そのため、実際の CPU

OS Monitoring Method By An Virtual Machine Monitor

Akira Kanasiro<sup>†</sup>, Kazuhisa Suzuki<sup>††</sup>, and Koichi Mouri<sup>†††</sup><sup>†</sup>College of Science and Engineering, Ritsumeikan Univ.<sup>††</sup>Graduate School of Science and Engineering, Ritsumeikan Univ.<sup>†††</sup>College of Information Science and Engineering, Ritsumeikan Univ.

の命令やメモリへのアクセスは、実際のハードウェアで行われている。本手法では、この仮想化されたハードウェアの利用状況を監視することで、OSの動作の状況を観測する。また、近年の登場した IA-32 アーキテクチャの CPU に実装されている仮想化支援技術を利用することで、観測対象 OS に改良を加えることなく実際の計算機上で動作するのと同様に仮想ハードウェア上で動作する。これによって観測対象 OS の動作に対する影響を少なくすることが可能となる。

### 3.2 本手法の応用

本手法では、仮想的な CPU やメモリの状況を観測することによって、観測対象 OS の動作状況を観測する。OS の動作状況を観測することで、仮想化されたハードウェアに対する要求の頻度を把握することが可能となる。この観測結果に基づいて、ハードウェアのアクセスが頻繁に発生する VM の優先度を高めることで、OS が要求する資源を適応的に割り付けることが可能となる。これによってシステム性能の向上を計ることが可能となる。また、近年の、マルチコア CPU の登場やメモリの低価格大容量化によって発生した計算機資源の余剰問題も解決することが可能となる。

## 4 システム構成

本稿で提案する手法は、IA-32 アーキテクチャ上で動作する仮想計算機モニタ Xen [3] の完全仮想化方式に改良を加えることで実現する。以下では Xen と本論文で提案するシステムの構成について述べる。

### 4.1 Xen

Xen における完全仮想化は Intel の Virtualization Technology (VT) を使用して実現されている。CPU、ページテーブル、入出力デバイス、ストレージデバイスなどを抽象化した環境を VM として提供する。Xen では VM のことをドメインと呼ぶ。特に Xen の制御や VM の起動や終了を行うためのドメインをドメイン 0 といい、Xen に併せて改良した OS が使用される。

仮想計算機の状態は Virtual Machine Control Structure (VMCS) と呼ばれる構造体で管理される。VMCS は VM 上の仮想 CPU 毎に存在し、VM entry / VM exit と呼ばれる VM 実行モード状態遷移の情報を管理している。実行する処理が特権レベルを必要とするか否かによって状態が遷移する。仮想化されたメモリへのアクセスは、VM 用のゲストページテーブルを Xen が管理する実メモリ上のシャドウページテーブルに置換することで制御される。VM への割り込み処理は、実計算機の割り込みコントローラによって通知された割り込みを仮想化された割り込みコントローラを介して VM の割り込みハンドラに送ることで処理を行う。このような仮想化されたデバイスを使用した処理を監視することで、OS の動作状況観測を実現する。

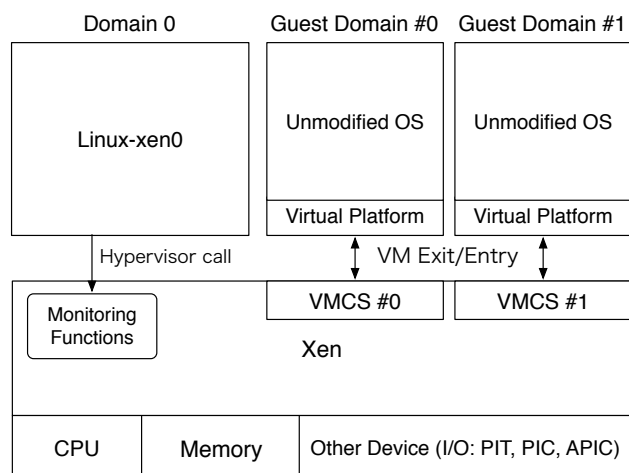


図 1 システム構成

### 4.2 構成

図 1 にシステムの構成を示す。図 1 では、Monitoring Functions で各ドメインでの VM 実行モード状態、仮想化されたメモリへのアクセス、割り込みコントローラを監視する。これによって VM 上の OS の動作を観測する。ドメイン 0 から Hypervisor Call を Xen に発行することで、観測結果を取得する。このため、取得するための Hypervisor Call を新たに追加する。

## 5 おわりに

本論文では、仮想計算機モニタを用いることで、OS の動作に与える影響を最小限にとどめ、動作状況観測を行う手法について述べた。今後の検討すべき事項としては Hypervisor call と Monitoring Functions の仕様の検討がある。

## 参考文献

- [1] 森本 洋行, 小宮山彰一郎, 毛利公一, 吉澤康文: 性能評価のための命令トレサの開発, 電子情報通信学会論文誌 Vol. J84-D-I No.6, pp.584–pp.593 (2001 年 6 月).
- [2] 山之内 暢彦, 多田 好克: IEEE1394 を利用した OS プロファイラの開発, システムソフトウェアとオペレーティング・システム研究会論文集, pp.25–pp.32 (2001 年 6 月).
- [3] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield: Xen and the Art of Virtualization, ACM Symposium on Operating Systems Principles, (2003).