

## 仮想計算機モニタ Xen における RTOS 向け割り込み通知機構

渡邊 和樹†

瀧本 栄二††

檜山 武浩†††

毛利 公一††

†立命館大学大学院理工学研究科

††立命館大学情報理工学部

†††立命館大学グローバル・イノベーション研究機構

### 1 はじめに

近年、携帯端末や車載システムといった組込みシステムでは、製品の多機能化に伴い、リアルタイム性に加えて、高い機能性の両立が求められている。組込みシステムでは、VxWorks などのリアルタイム OS(RTOS) が利用される。RTOS は、機器の制御を主目的とした最小限の機能提供により、処理時間の予測可能性を高め、リアルタイム性を保証する。一方、高度な情報処理を行うシステムでは、Linux などの高機能 OS が利用される。高機能 OS は、高い機能性を実現するために複雑なシステム構成をしており、処理時間の予測可能性は低い。このようにリアルタイム性と高機能性は相反する性質であり、その両立は容易ではない。

リアルタイム性と高機能性を両立する手段として、2通りの既存の手法がある。一つは、両方の特性を併せ持ったハイブリッド OS を用いる手法である。しかし、相反する性質を持つため、その設計は容易ではなく、ソフトウェア開発コストの増加を招く。もう一つは、システムに複数の計算機を搭載し、RTOS と高機能 OS を個別に動作させる手法である。この手法は、物理的な計算機の増加に伴い、ハードウェアのコスト増加を招く。また一方で、組込み用のマルチコア CPU も発売されている。以上の背景から、既存の仮想計算機モニタ(VMM)である Xen[1] を拡張し、仮想計算機(VM)を用いて、RTOS と高機能 OS を同時動作可能とする VMM「Natsume」を開発している。

VM 上で RTOS を動作させる場合の課題として、リアルタイム性を意識した資源管理があげられる。VM 上では、計算機資源が仮想化・共有され、性能が他の VM や VMM の負荷に影響される。これは、RTOS の処理時間の予測可能性を低下させる。そこで、計算機資源を RTOS に占有させ、この解決を試みる。本論文では、RTOS 向けの資源管理機構、特に割り込みの占有を実現する割り込み通知機構について述べる。

### 2 仮想計算機モニタ Natsume

Natsume は、VM を用いて RTOS と高機能 OS を同時動作させる。Natsume は、管理用インターフェース

Management of interrupt for RTOS in Xen Hypervisor  
Kazuki WATANABE†, Eiji TAKIMOTO††, Takehiro KASHIYAMA†††,  
and Koichi MOURI††

†Graduate School of Science and Engineering, Ritsumeikan University  
††College of Information Science and Engineering, Ritsumeikan University

†††Ritsumeikan Global Innovation Reserch Organization, Ritsumeikan University

持つ特権 Domain, RTOS を動作させる RT-Domain, 高機能 OS を動作させる Domain-U, 計算機資源を提供する RT-Hypervisor で構成される。

Natsume は、リアルタイム性が重要な処理を RTOS で実行し、高度な情報処理を高機能 OS で実行することで、リアルタイム性と高機能性を両立する。また、単一の計算機上で複数の OS を動作させることで、ハードウェア開発コストを削減する。加えて、既存の VMM である Xen を基にすることで、ソフトウェア資産の有効活用による、ソフトウェア開発コスト削減を実現する。

### 3 Natsume における資源管理

#### 3.1 概要

Natsume では、VM 上で RTOS を動作可能とするために、3つの資源を占有させる。CPU と主記憶の占有は、Xen が提供する XenTools を用いて実現する。入出力の占有は、Xen の PCI Pass-through と我々が提案する RTOS 向け割り込み通知モデルを用いて実現する。

#### 3.2 PCI Pass-through と課題点

PCI Pass-through は、Xen が提供する資源管理機構であり、VM に PCI デバイスを排他的に割り当てる。対象デバイスは、他の VM から隠蔽され、ゲスト OS による直接アクセスが可能になる。

PCI Pass-through は、I/O 処理の占有を実現する。しかし、発生する割り込みは、他のデバイスと同様の機構を介して通知される。Xen の割り込み通知機構であるイベントチャネルは、割り込みを平等に扱い、割り込み通知を直列化する。結果、割り込み処理の横取りができない。

すなわち、PCI Pass-through を利用した場合でも、割り込み通知において他のデバイスと平等に扱われ、他の VM や VMM の影響を受ける。これは、割り込みの遅延や損失の原因となり、リアルタイム性の保証を困難にする。これを解決するために、MSI(Message Signaled Interrupts)[2] と専用割り込みハンドラを用いた「RTOS 向け割り込み通知モデル」を提案する。

### 4 RTOS 向け割り込み通知モデル

#### 4.1 概要

提案モデルは、RTOS でリアルタイム性を保証する目的で、I/O と割り込み双方の占有を実現するために、デバイス毎に独立した割り込み通知機構を利用可能にするものである。提案モデルと PCI Pass-through により、完全なデバイスの占有可能とする。その概要を図 1 に示す。

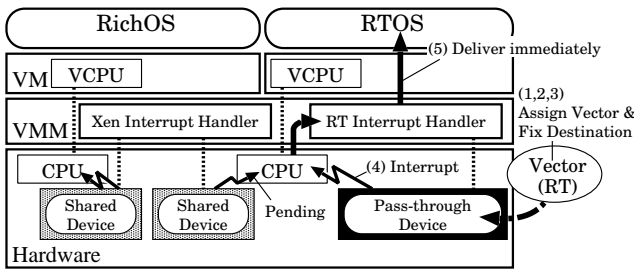


図 1: RTOS 向け割り込みモデルの概要

1. システム起動時に、占有デバイスのバス・デバイス・機能番号を取得し、デバイスを識別する
2. MSIを用いて、占有デバイスに専用の Vector を割り当て、他デバイスと割り込みレベルで区別する
3. 同様に、MSIを用いて、占有デバイスの割り込み通知先を RTOS が占有する CPU に固定する
4. 専用の Vector を持つ割り込み (占有デバイスからの割り込み) を、専用の割り込みハンドラで処理する
5. イベントチャネルの割り込み処理を横取りし、ゲスト OS の割り込みハンドラに即座に制御を移す

#### 4.2 現在の実装状況

提案モデルの実現に向けて、以下の機構を実装した。

**デバイス識別機能** 起動オプションから、PCI Pass-through の対象デバイスの情報を取得する機構を実装した。

**MSIによるデバイス設定機能** 対象デバイスで MSI を有効にし、割り込み Vector の割当てと、割り込み通知先を特定の VM が占有する CPU に固定する機構を実装した。

**軽量化した割り込みハンドラ** MSI 以外の割り込みをハンドルしないこと、通知先のゲスト OS が常に 1 つであることを意識して軽量化したハンドラを実装した。

なお、既存機構から割り込みを横取りする機能は未実装である。また、現在は、物理 CPU を占有する設定であっても、VM がアイドル状態になった際、IdleDomain という仮想 VM に物理 CPU が割り当てられる。これを回避する目的で、VM 上で低優先度で動作する CPU バウンドなユーザプログラムを用いている。

### 5 評価

先述した機構を実際に適用し、性能評価を行った。評価環境を表 1 に示し、具体的な手順を以下に示す。

1. 負荷用 OS に共有デバイスとして負荷用デバイスを、評価用 OS に PCI Pass-through を用いて評価用デバイスを割当てて
2. 負荷用デバイスに対して、外部から大量の UDP パケットを送信し、割り込み負荷をかける
3. 計測用デバイスに対して、外部から 100msec に 1 回の間隔で、1024 個の UDP パケットを送信する
4. VMM とゲスト OS の割り込みハンドラで TSC の値を記録し、差分から所要時間を算出する

表 1: 評価環境

CPU/Mainboard	Core i7 920/DX58SO
評価用デバイス	82567LM-2 Gigabit Network Connection
負荷用デバイス	82541GI Gigabit Ethernet Controller
評価用ゲスト OS	linux-2.6.18-xen(CPU1 個/RAM2GB)
負荷用ゲスト OS	linux-2.6.18-xen(CPU3 個/RAM4GB)

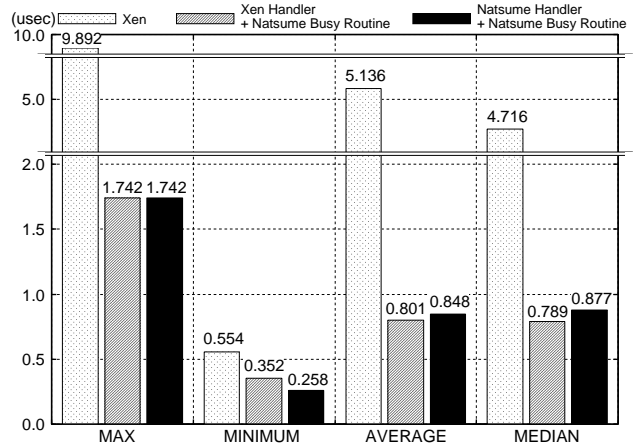


図 2: 性能評価結果

計測結果を図 2 に示す。オリジナルの Xen と比較して、最大値で約  $7\mu\text{sec}$ 、平均・中央値において約  $4\mu\text{sec}$  の高速化となり、55%から 82%の改善を確認できた。また、提案手法の内、アイドル状態を回避するためのユーザプログラムをオリジナルの Xen で実行した場合と、Natsume で実行した場合を比較すると、最小値が約  $100\text{nsec}$  の高速化となり、25%良い結果になった。平均・中央値では、約  $40\text{nsec}$  から  $80\text{nsec}$  低速である結果になったが、これは誤差の範囲であると考えられる。

割り込みハンドラへ、割り込み処理を横取りする機能を実装することで、更に高速・安定化が可能であると考えられる。

### 6 おわりに

本論文では、組込みシステムにおける高機能化の要求に対して、単一計算機で RTOS と高機能 OS の同時動作を実現する、VMM「Natsume」を提案した。また、実現に向けた RTOS 向け割り込み通知機構の設計と、実装・評価について述べた。結果、基である Xen と比較して最大 82%の改善を確認した。今後、未実装の機能を実装し、RTOS を用いた評価を行う予定である。

#### 参考文献

- [1] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. *ACM Symposium on Operating Systems Principles*, pp. 164–177, 2003.
- [2] PCI-SIG. *PCI Local Bus Specification Revision 3.0*. PCI-SIG, aug 2002.