

## マルチコアプロセッサ環境における組込みシステム向けVMMのメモリ管理

前田 剛志<sup>\*1</sup>, 佐藤 未来子<sup>\*1</sup>, 並木 美太郎<sup>\*1</sup>東京農工大学<sup>\*1</sup>

## 1 はじめに

近年、組込みシステム分野において、ハードウェアの高度化に伴い組込みシステムへの要求も高度化している。特に、ITシステムとの連携を行うことによる、高度な組込みシステムを構築することが求められるようになった。

そこで、仮想マシンモニタ (VMM) により、リアルタイム OS、汎用 OS の両者の特徴を生かすことのできる実行基盤が求められている。しかし、組込みシステムに仮想化技術を適用することによって、仮想化処理のオーバーヘッドによるリアルタイム性の阻害、実行時間の予測可能性の損失という課題がある。

これらの課題を満たすため、本研究では、組込み向け仮想マシンモニタ OVM[1] において、CPU、I/O の仮想化と並び重要な要因であるメモリの仮想化について設計を行った。リアルタイム性の要求が異なる OS を動かす環境において、各々のシステムに対して適切なメモリ管理手法を提供することで、上記の課題に対応する。

## 2 仮想マシンモニタ OVM

## 2.1 全体概要

OVM(OptimusVirtualMachine) は準仮想化方式の組込み向けホストハイパーバイザ (Type2) であり、マルチコアプロセッサ、メモリおよび I/O を仮想化する (図 1)。OVM は「システムの必要とするリアルタイム性の保証」、「複数のパラダイムのシステムの並行・並列動作」を目標としている。

本研究では OVM を、組込みシステムで広く利用されており、また FPGA や豊富な I/O が利用できることから、ARM の Cortex-A9 MPCore プロセッサを持つ Zynq-7000 SoC ZC702 に実装する。ARM の TrustZone 技術を用い、VM をノーマルワールドで動かす、OVM をセキュアワールドで動かすという構成をとる。OVM へはハイパーバイザコールによる明示的な遷移が可能であり、ホスト OS からのゲスト OS ロード・ブートなどの基本機能を提供し、ゲスト OS 開発者の OS 開発を支援している。

OVM は、ゲスト OS のリアルタイム性に応じた各ハードウェア資源の仮想化方式を提供するために、ゲスト OS に対し、三段階のリアルタイムレベルを提供する。ゲスト OS のリアルタイムレベルに応じた仮想化方式で OVM が資源を割り当てることで、異なるゲスト OS がそれぞれ要求するリアルタイム性を保証する。

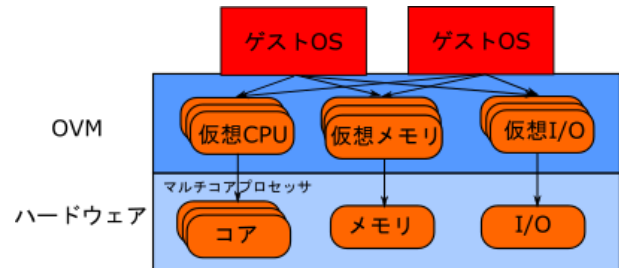


図 1: OVM の概要図

## 2.2 リアルタイムレベル

OVM は以下に示す三段階のリアルタイムレベルをゲスト OS に設定可能にする。リアルタイムレベルによって異なる仮想化方式をゲスト OS に提供することができ、リアルタイム要求の厳しいゲストには資源の占有や優先利用、そうでないゲストには資源の効率的な利用をさせる。これにより、リアルタイム性の異なる OS の共存を実現する。

## (1) HRT(ハードリアルタイムゲスト向け)

他のシステムから独立してリアルタイム性を保証する手法を用いる。

## (2) SRT(ソフトリアルタイムゲスト向け)

優先的に物理資源を獲得するが、余剰資源は共有する。

## (3) NRT(汎用ゲスト向け)

他のシステムと平等に物理資源が分配し、平均応答時間を向上させる手法を用いる。

## 3 組込み向け VMM のメモリ管理の課題

異なるリアルタイム性の OS 共存実現の問題に対して、VM の優先度付きスケジューリング [2] のような仕組みとともに、次に示す課題が考えられるため、我々はメモリ管理に関して研究を行う必要がある。

## 3.1 メモリ仮想化のオーバーヘッドと予測可能性

リアルタイム要求の厳しいシステムでは、メモリ仮想化に伴う仮想化処理がオーバーヘッドとなりえる。また、VM のコンテキストスイッチに伴う突発的な TLB ミスのように、予測可能性を阻害するような事態も起こりえる。このように、求めるリアルタイム性や予測可能性を VMM が確保するという課題がある。

## 3.2 複数 VM が動作する環境でのメモリ保護

重要度の高いシステムがより重要度の低いシステムにより障害を発生しないよう、かつ、リアルタイムシステムのリアルタイム性を阻害しないメモリ保護を VMM が行うという課題がある。

## 4 OVMのメモリ管理設計

OVMのメモリ管理の全体構成は、リアルタイムレベルごとに提供されるメモリ仮想化手法(図2)およびメモリ保護機能からなる。

### 4.1 メモリセパレーション

HRTシステムでは、シャドウページングのような仮想化機構は、オーバヘッドがタスクのリアルタイム性を損ねる問題が考えられる。そこで、特定の物理メモリ領域をメモリ仮想化機構から独立させ、ゲスト物理アドレス空間をストレートマップすることにより上記の問題を解決する。

### 4.2 NRTシステムのためのメモリ仮想化手法

NRTシステムには、ITシステムを処理する汎用OSを想定しており、シャドウページング[3]と呼ばれる手法を用いる。図2のように、ゲスト仮想アドレス空間をOVM上のシャドウページテーブルによって、物理アドレスに変換する仮想化手法である。ページングを用いることにより、動的メモリ割り当て時に、物理メモリが連続である必要が無く、フラグメンテーションを防ぐことができ、メモリ利用効率の向上につながる。

### 4.3 SRTシステムのためのメモリ仮想化手法

SRTシステムには、リアルタイムOSを想定している。リアルタイムOSを採用する利点の一つとして、予測可能性がある。ただしこの特性はVMM上で他のシステムと共存を考えたときに、損なわれてしまう可能性がある。特にメモリに関わる部分では、VMのコンテキストスイッチに伴うTLBフラッシュによって、TLBミスが発生し、ページウォークが行われるという事態が問題である。

そこで、本手法ではゲストOS起動時にOVMにより、TLBにシステムをロックダウンすることで、この問題に対応する。しかし、ロックできるTLBの数は限られている。そこで、ゲストOSのメモリ仮想化手法は前節のNRTレベルと似ているが、ページではなく、より粒度の荒い、ARMのセグメントを利用することにより、エントリの数を減らし、この問題に対処する。

また、一般的にリアルタイムOS自体のシステムの規模はあまり大きいものではなく、また、リアルタイムOSには動的メモリ管理を利用することが考えられていないことが多い。したがって本方式は予測可能性の損失に対し効果的であると考えられる。

### 4.4 メモリ保護機構

本研究で考えているOVMの環境では、異なるリアルタイム性を持ったゲストOSが同じハードウェア上で混在するシステムを構築することとなる。高いリアルタイムレベルを持つゲストOSは、重要なタスクを扱っているため、低いリアルタイムレベルのゲストOSの故障が影響しないようにする必要がある。そのため、

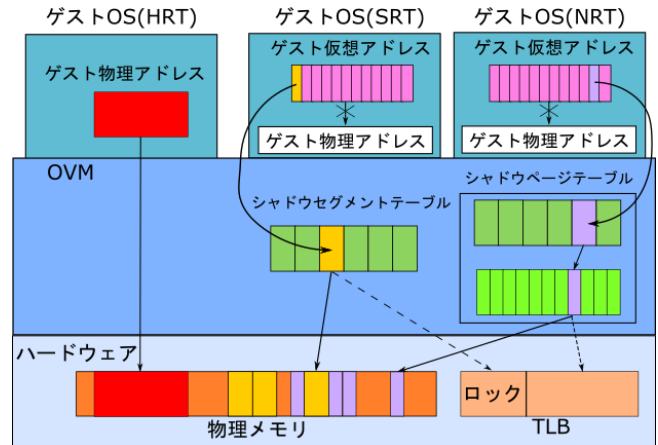


図2: OVMのメモリ仮想化

メモリ保護機構を持たせる必要がある。

メモリ保護機構を実現するために、OVMはメモリ割り付け時に、メモリの割り付け単位ごとに保護キーを持たせる。これにより、ゲストOSは自身に割り振られた保護キーと同じ保護キーを持つ領域にのみアクセスできる。メモリの割り付け単位はすなわちページ単位で、ページテーブルエントリに保護キーの情報を付加することで実現する。

ただし、HRTシステムに対してはメモリ保護機構がオーバヘッドとなる可能性を考慮し、メモリ保護機構を持たせない。つまり、メモリ保護機構を持つのはSRTおよびNRTシステムで、これらのシステムがHRTシステムを含む他のシステムに不正なアクセスを行わないようにする。

## 5 結言

本研究では、組込み向け仮想マシンモニタのためのメモリ管理手法として、三段階のリアルタイムレベルに対するメモリ管理手法を提案した。

これまでに、TrustZoneを用いたセキュアワールド、ノーマルワールドの分離が実現できており、現在設計した手法の実装を行っている。今後の課題として、提案した手法の実装を完了し、それぞれのリアルタイムレベルにおいて、リアルタイム性を保証できるかを検証する必要がある。

## 参考文献

- [1] 小倉佑太 et al., 『マルチコアプロセッサを用いた組込みシステム向けVMMの設計と実装』東京農工大学 / 研究報告システムソフトウェアとオペレーティング・システム(OS) 2013-OS-125, 7, pp.1 - 8, 2013-04-18
- [2] Alejandro Masur et al., 『Designing VM Schedulers for Embedded Real-Time Applications』 Technical University Munich / CODES+ISSS '11, pp.29 - 38, 2011
- [3] 伊藤愛・追川修一『Gandalf VMMにおけるShadow Pagingの実装と評価』筑波大学大学院システム情報工学研究科 / 情報処理学会論文誌コンピューティングシステム(ACS), 49, SIG2(ACS21), pp.98 - 112, 2008-03-15