

組込みシステムを想定した Xen の軽量化手法

小野 利直[†] 金城 聖^{††} 永島 力^{††} 毛利 公一[†][†]立命館大学情報理工学部 ^{††}立命館大学大学院理工学研究科

1 はじめに

近年、携帯電話や車載システムなどに利用されている組込みシステムにおいて、従来の要求である応答性や信頼性に加えて、高い機能性が求められている。

従来の組込みシステムでは、リアルタイム OS (以下、RT-OS と記す) を利用し、機器の制御を主要機能とした最小限の機能提供により、高い応答性と信頼性を保障している [1]。そのため、機能性の確保は不要であった。一方、Windows や Linux などの高機能 OS は、動画の再生やアプリケーションの制御などの複雑な処理を主要機能として提供しており、高い機能性が確保されている。しかし、複雑な情報処理を目的とする機能は、構造が複雑になり、処理の予測性が低下するため、RT-OS のような高い応答性と信頼性は保証されていない。

以上から、単一の OS で高い応答性、信頼性、機能性を同時に保障することは困難である。この問題を解決するために、我々は、仮想計算機モニタ (以下、VMM と記す) を利用し、RT-OS と高機能 OS を同時に運用可能とする、リアルタイム仮想化ソフトウェア基盤 (以下、RT-VMM と記す) の研究・開発を行っている。RT-OS と高機能 OS を 1 台の計算機上で同時に動作させることにより、RT-OS で高い応答性、信頼性の確保を、高機能 OS で高い機能性の確保を行うことが可能となる。

組込み機器は、大量生産のコストや製品サイズの観点から、CPU やメモリといったハードウェア資源が限られており、システムにはより高速な処理、より小さいメモリフットプリントが要求される。そのため、組込みシステムにおいては、サーバシステムの VMM をそのまま組込み機器上で使用することは適切ではなく、仮想化によるハードウェア資源の消費を抑えた組込みシステム用の VMM が必要となる。VMM がわずかな CPU 資源の消費とわずかなメモリ使用量で、ゲスト OS の動作をネイティブと同様に動作できる環境を仮想的に実現することができれば、組込み機器上でも実用的な VMM となる。

以下、本稿では、2 章でリアルタイム仮想化ソフトウェア基盤について述べ、3 章で仮想計算機モニタ Xen の概要について述べ、4 章で Xen の軽量化手法について述べ、5 章でツールの軽量化について述べる。

2 リアルタイム仮想化ソフトウェア基盤

RT-VMM は、VMM を基盤に、RT-OS と高機能 OS の共存を可能にするシステムソフトウェアである。現在、構成しようとしている要素は、各ドメインの管理用インターフェースを提供する特権ドメイン、RT-OS が動作する RT ドメイン、高機能 OS が動作するゲストドメインと、これらのドメインに対してハードウェア資源を提供する RT ハイパーバイザからなる。

現在、既存の VMM である Xen に改良を加えることで、この RT-VMM を実現しようとしている。既存の VMM を利用することで、既存の資産を容易に利用することが可能となる。

3 仮想計算機モニタ Xen

3.1 概要

Xen は既存の VMM の一つである [2]。特権ドメインと呼ばれる Domain-0 と、ゲストドメインと呼ばれる Domain-U という 2 種類のドメインが存在し、それぞれがハイパーバイザによって仮想化され、資源の提供を受けて動作している。

- Domain-0
Xen の起動を行うほか、ハードウェア資源の管理や、ハイパーバイザ上で動作するゲスト OS の起動、管理を行う。Linux が使用される。
- Domain-U
ゲスト OS が動作するドメインで、Domain-0 によって起動される。

また、Xen はさまざまな機能を提供するためのツールを持っている。これらのツールを用いることにより、Xen は仮想的なネットワークを構築することや、Domain-0 からコマンドによる Xen の制御を行うことが可能となる。

3.2 Xen のインタフェース階層構造

Xen のインタフェースの階層は図 1 に示すようになっている。Xen のツールがコマンドを発行すると、C 言語ならば libxen、Python ならば pyxen などの言語バインディングによって、XML-RPC 要求に変換される [3]。次に、バインディングライブラリは、ソケットを通じて、xend のインスタンスに XML を送信する。xend は、要求を独自に処理するか、カーネルのハイパーバイザインタフェースに受け渡す。カーネルに受け渡した場合、要求は最終的にハイパーバイザに受け渡される。

A construction of light-weight Xen for embedded systems
Toshinao Ono[†], Akira Kanasiro^{††}, Chikara Nagashima^{††}, and
Koichi Mouri[†]
[†]College of Information Science and Engineering, Ritsumeikan
University
^{††}Graduate School of Science and Engineering, Ritsumeikan
University

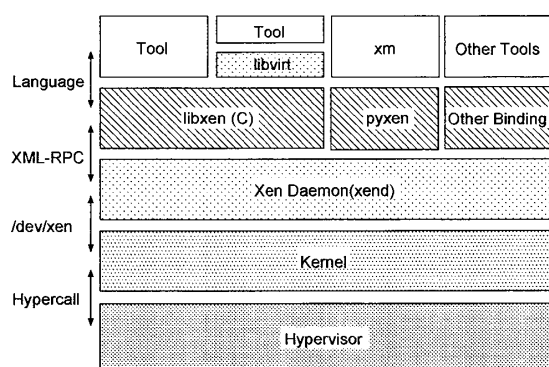


図 1 Xen のインタフェース階層構造

4 Xen の軽量化手法

Xen には、デバッガやテスト用のデータ、また開発者用のプログラムなどが含まれており、そのまま組み込みシステムに導入するには適していない。そのため、資源が限られる組み込み機器上に導入するには、Xen の軽量化が必要となってくる。本研究では、Xen の軽量化を Domain-0 の軽量化、ツールの軽量化、ハイパーバイザの軽量化の 3 つのアプローチで行う。

- Domain-0 の軽量化
Xen 導入時の Domain-0 には、Xen を管理するにあたり不要な機能が含まれている。Domain-0 を最小構成にすることで、小さいメモリフットプリントで、かつセキュアという組み込みシステムに適した環境を実現することが可能である。
- ハイパーバイザの軽量化
組み込み機器上で動作させるにおいて、ハイパーバイザにも必要のない機能が含まれている可能性がある。フットプリントと応答性を考慮した設計をすることで、ハイパーバイザの軽量化が可能である。
- ツールの軽量化
Xen には、Xen を動作させる上でさまざまな機能を担うツールが含まれている。これらツールの中には組み込み機器上で運用する上で、不要なものが含まれている。これらの不要な機能を削減することにより、生成されるイメージサイズの削減が可能である。

本稿では、特に、ツールの軽量化について述べる。

5 ツールの軽量化

ツールの軽量化を、ツールに含まれる不要な機能の削減と、ツールが仕様している domain-0 の Python ライブラリの削減という 2 つの方法で行う。

5.1 ツールに含まれる不要な機能の削減

上位レイヤから順番に、不要な機能の選別と削減を行う。最上位レイヤのツール群から、組み込み機器に適したツールの選別を行い、下位レイヤの言語バインディング

からそれらに対応する機能の選別、その後、RPC、xend の順に各々対応する機能を選別し、削除を行う。これを行う事により、make 時に生成される Xen のイメージサイズの削減が可能である。

携帯電話ならば、xen3.2.0/tool/以下の vnet に含まれるネットワークインタフェースや、blktap に含まれるディスク I/O、security に含まれるアクセスコントロールなどのツールが必要となり、pygrub に含まれる grub 仕様のブートローダなどのツールが不要となる。これらのツールの選別を行った後、それぞれの下位レイヤの対応するツールの削減を行う事で、携帯電話に適した軽量の Xen を実現する事が可能である。

また、他の組み込み機器、例えば車載システムなどでも同様に、必要な機能の選別を行い、各機能、各レイヤごとに削除を行うことで、それらに適した Xen を実現する事が可能となる。

5.2 Python ライブラリの削減

ツールの一部が使用している Python のライブラリは、システムの中でも 26MB とサイズが大きい。Python ライブラリは、組み込みシステムを想定して作られていないため、ウィンドウの作成や電子メールの解析など、組み込みシステムでは不要な機能や、Xen では使用されていないモジュールを多分に含んでいる。Xen で使用されていないモジュールの選別、削減を行う事で Domain-0 の軽量化が可能となる。

今回実際に、Xen で使用されないモジュールの選別、削減を行った。/usr/lib/python2.4/以下の画像に関する演算に使用する imageop モジュールや、オーディオデバイスへのアクセスに使用する ossaudiodev モジュールなど、計 1,130 ファイルの削除を行い、その結果、26MB あった Python ライブラリを 8.4MB まで削減することができた。

6 おわりに

本稿では、組み込みシステムを想定した Xen の軽量化手法について提案した。また、Xen のツールの軽量化の必要性と手法について述べた。ツールに不要な機能の削減と Python ライブラリの削減を行う事で、軽量の Xen を実現する事が可能である。

参考文献

- [1] 永井 正武監修, 澤田 勉, 権藤 正樹, 永井 正武共著: “実用組み込み OS 構築技法,” 共立出版, pp. 6-15 (2001).
- [2] 宮本 久仁男, 大島 孝子, 平 初, 長谷川 猛共著: “Xen 徹底入門,” 翔泳社 (2007).
- [3] David Chisnall 著, 渡邊 了介訳: “仮想化技術 Xen 概念と内部構造,” 毎日コミュニケーションズ, pp. 275-299 (2008).