

L4VM：マイクロカーネル上で動作する仮想マシンの構築

—マルチアーキテクチャ対応仮想マシン—

1. 背景

昨今、仮想化技術が再び注目を集めている。1990年代に登場した VMware などの仮想化技術の製品は、デスクトップ PC で異なるオペレーティングシステムを手軽に利用できるため、システムレベルソフトウェアのデバッグ、サーバの管理の手段として用いられてきた。現在、仮想マシンは計算機システムのセキュリティや可用性を向上することが期待されている。Xen などの仮想マシンモニタ(VMM: Virtual Machine Monitor)も注目を集めており、Intel や AMD などのプロセッサメーカーは仮想化をハードウェアでサポートする技術を既に製品化して市場に投入している。しかし、既存の仮想マシンにはいくつかの問題がある。例えば、多くの仮想マシンはホスト・ゲストともにサポートする CPU アーキテクチャが限られている。VMware はホスト・ゲストとして x86、x86_64 アーキテクチャのみをサポートしている。Intel や AMD のプロセッサを用いたハードウェアによる仮想化も x86 アーキテクチャに限定されている。VMware や Xen のように、ホストとゲストのアーキテクチャが一致していなくてはならないケースも多い。

2. 目的

多くの仮想マシンがサポートする CPU アーキテクチャが限定的であるのに対し、QEMU はホスト、ゲストに様々な CPU アーキテクチャをサポートしている。QEMU はエミュレータのなかでも最も多くのホストアーキテクチャ、ゲストアーキテクチャ、そしてそれらの様々な組み合わせをサポートしつつ、高速な実行を可能としているエミュレータである。しかし QEMU はホスト環境として Windows、FreeBSD、Linux、Mac OS X などの高機能なオペレーティングシステムを必要としている。これらのオペレーティングシステムは多量のメモリを消費し、多くのタスクを動作させ、大規模な仕組みによって管理されているため、エミュレータ単体の動作を見ると、大きなオーバーヘッドがかかっている。またホストオペレーティングシステムが提供する API はそもそもアプリケーションを支援することを主眼に設計されているため、システムレベルのソフトウェアをその上で動作させるエミュレータを支援するには柔軟性に欠けていると考えられる。

このような問題を解決するために本プロジェクトでは、マイクロカーネル上で動作する仮想マシンを開発する。マイクロカーネルとは最小限の機能だけを実装することで、性能、柔軟性、移植性を向上したオペレーティングシステムである。本プロジェクトではマイクロカーネルとして、L4 マイクロカーネルを用い、その上に仮想マシンとして QEMU を移植する。マイクロカーネルを用いることで、オペレーティングシステムが消費するシステムリソースを抑えつつ、マイクロカーネルの提供する単純かつ柔軟性の高い API を用いた、柔軟性、拡張性の高い仮想マシンの構築を目標とする。

3. 開発の内容

(1) システム概要

L4VM システムは単一のハードウェア上で複数のオペレーティングシステムの同時起動を可能とする仮想実行環境である。複数の仮想マシンを生成し、それぞれで異なるオペレーティングシステムを起動することができる。さらに各仮想マシンは異なるアーキテクチャの仮想 CPU、異なる仮想デバイスを提供することができる。

L4VM システムは、仮想マシン L4VM とその動作を支援するマイクロカーネルやサーバなどによって構成されている（図 1）。L4VM は L4VM システムの中核を成す仮想マシンである。L4VM は実在の CPU やハードウェアのインタフェースを持つ、仮想 CPU や仮想デバイスを提供する。そのベースとしてマシンエミュレータ QEMU を用いている。デバイスサーバは複数の L4VM がひとつの実ハードウェアを使うとき、複数の仮想マシンの入出力命令を適切に調停する。L4 マイクロカーネルは各 L4VM やデバイスサーバのスケジューリングや資源管理、サーバ間の通信のためのプロセス間通信などを提供している。

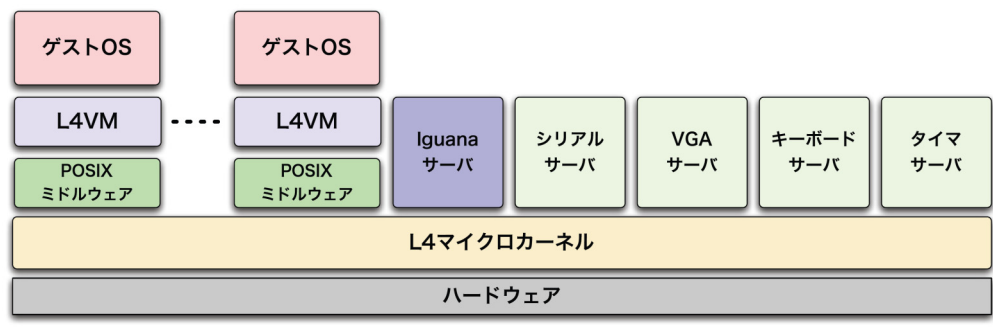


図 1：L4VM システム構成図

(2) L4VM がサポートするデバイス

L4VM がサポートする仮想 CPU と仮想デバイスを表 1 に示す。VGA（テキストモード）サーバとキーボードサーバは複数の L4VM への入出力を自由に切り替えることができる。また、IDE ハードディスクのデータは実ハードウェアでは RAM 上に展開されている。

表 1：L4VM がサポートする仮想 CPU と仮想デバイス

仮想 CPU	x86, PowerPC, SPARC, ARM
仮想デバイス	シリアルポート、キーボード、VGA（テキストモード）、VGA（グラフィックモード 640x480 16 色）、IDE ハードディスク

(3) 動作環境

L4VM システムは PC/AT 互換の計算機で動作する。現在 ThinkPad X23, ThinkPad R50, ThinkPad T43 で動作を確認している。消費メモリは起動する L4VM の個数、各 L4VM のメモリ割当量に依存する。

4. 従来の技術（または機能）との相違

（１）ゲスト・ホストに異なるアーキテクチャをサポート

多くの仮想マシンはホスト、ゲストともにサポートする CPU アーキテクチャが限られている。それに対し、L4VM システムは QEMU のダイナミックトランスレーション機能を利用することで、仮想マシンのアーキテクチャとして、ホストマシンとは異なったアーキテクチャを提供することができる。例えば、x86 アーキテクチャのホストマシン上で SPARC アーキテクチャの仮想マシンを実行することが可能である。

（２）拡張性・移植性

L4VM システムは、ホストオペレーティングシステムとしてマイクロカーネルである L4 を用いることで、拡張性・移植性に優れた構造となっている。

L4 はユーザレベルタスクにアドレス空間の管理、スレッドの制御、I/O ポートを直接読み書きするための機能を提供しており、Linux や Windows などのアプリケーションと比較すると、多様な処理が可能である。そのため、L4VM システムは拡張性の高いシステムとなっている。

また、L4 自体が移植性の高い設計となっている。Xen などの仮想マシンモニタはアーキテクチャを変更すると提供するインタフェースが変わってしまうため、ゲストオペレーティングシステムの修正が必要となる。L4 は異なるアーキテクチャに移植してもアプリケーションに対するインタフェースが変更されないため、L4VM システムを修正することなく異なるプラットフォームへ移植することが可能である。

さらに、L4 の消費メモリが少ない点もメリットとして挙げることができる。L4 は非常にシンプルなオペレーティングシステムであるため、Linux や Windows などの汎用オペレーティングシステムと比較して消費するメモリの量が少ない。そのため、ホストオペレーティングシステムとして Linux を用いた場合と比較して、システム全体の消費メモリを抑えることができる。

5. 期待される効果

これまでの仮想マシンとしては、Linux や Windows や Mac OS などの高機能なオペレーティングシステム上で動作するハードウェアエミュレータは存在していたが、マイクロカーネルという抽象度の低いオペレーティングシステム上で実装した例は見られなかった。本プロジェクトでは、仮想マシンをマイクロカーネル上に構築するという新しいチャレンジに挑戦しており、これをきっかけにマイクロカーネルを基盤とした仮想実行環境の開発が広く行われていき、VMware や Xen に代わる次世代の仮想実行環境が登場することを期待する。

6. 普及（または活用）の見通し

これまで学会発表やワークショップ、Embedded Technology 2005 といった展示会などで発表およびデモンストレーションを行った。既に開発成果の公開用 Web ページは作成済みであり、後々には開発プログラムの配布を行う予定である。今後も、学会発表やワークショップや展示会などでの発表およびデモンストレーションを行い、開発成果のアピールを行っていく予定である。

7. 開発者名（所属）

- ・ 腰前 秀成（早稲田大学 大学院理工学研究科 修士課程）
- ・ 杵渕 雄樹（早稲田大学 大学院理工学研究科 修士課程）