

プロセススケジューラが仮想マシンへ与える影響に関する考察

東 賢一朗†

†株式会社 日立製作所 プラットフォームソリューション事業部

1 はじめに

物理マシンの集約や、既存システムの延命などの要求に応える技術要素として、仮想化技術が存在する。特に、プロセッサレベルでの仮想化支援機能の登場により、仮想化技術の適用範囲が拡大している。例えば、プロセッサの仮想化支援機能を利用し、Linux カーネルに仮想化機能を付加する Kernel-based Virtual Machine(KVM)[1] は、Linux カーネルにおける仮想化技術のひとつとして注目を集めている。

一方、Linux2.6 系のプロセススケジューラは、Linux2.6.23 を契機に、システムに存在する最も高いプロセス優先度を選択する O(1) スケジューラから、各プロセスの重みに対して公平に CPU 資源を割り当てる Completely Fair Scheduler(CFS) へ変更となった。

そこで本研究では、Linux カーネルに仮想化技術を付加する KVM において、Linux カーネルのプロセススケジューラの違いが仮想マシンに与える影響を実験し、Linux カーネルに内在する仮想化技術の問題点やその解決指針の考察を目的とする。

2 Kernel-based Virtual Machine

Kernel-based Virtual Machine(KVM) とは、仮想マシンモニタの機能を Linux カーネルに付加するカーネルモジュールのことである。KVM の概要を図 1 に示す。

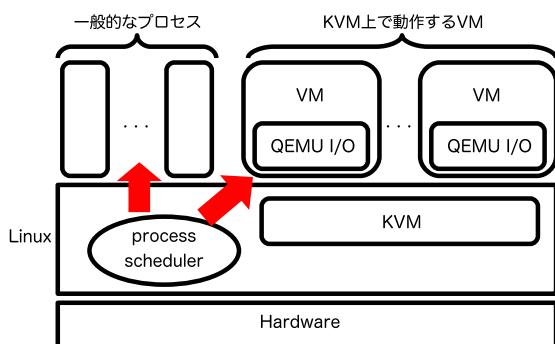


図 1: KVM の概要

KVM が生成する仮想マシンは、プロセッサの仮想化支援機能を使用し、ゲスト OS の修正が不要な完全

仮想化を実現している。仮想マシンの I/O エミュレーションは、各仮想マシンに存在する QEMU と、KVM が提供するデバイスファイル/dev/kvmを経由し、カーネルモードとして I/O 処理を実行するよう Linux カーネルに処理を発行する。また、Linux カーネルのプロセススケジューラは、KVM が生成する仮想マシンも通常のプロセスと同等に扱う。

3 Linux のプロセススケジューラ

3.1 O(1) スケジューラ

O(1) スケジューラとは、システム中にプロセス優先度が最も高い実行可能なプロセスを選択するプロセススケジューラのことである。プロセス優先度は、システムコール nice() やシステムコール setpriority() で指定した nice 値を基に指定可能な静的優先度と、プロセスの対話性をヒューリスティックに計算した動的優先度とで決定する。

O(1) スケジューラは、線形探索によりシステム上で最も高いプロセス優先度を決定していた Linux2.4 系以前のプロセススケジューラと比較し、スケジューリングコストを低減するメリットがあった。しかし、複雑でヒューリスティックな優先度計算や、優先度が異なるプロセス間の公平性確保という点で課題を抱えていた。

3.2 Completely Fairness Scheduler

O(1) スケジューラの問題点を解決するため、Linux2.6.23 から Completely Fair Scheduler(CFS) が登場した。CFS では、プロセス優先度を基に重みを各プロセスに与え、ランキューに存在する全プロセスの重みの総和と、個々のプロセスの重みとの割合により、各プロセスの CPU 利用率を決定する。

O(1) スケジューラと比較して、複雑なヒューリスティックの排除や、各プロセスに対して公平に CPU 資源を配分できるというメリットがある。しかし、ひとつのプログラムが複数のスレッドを生成するマルチスレッド環境では、各スレッドに対して均等にタイムスライスを割り当てるため、結果としてひとつのプログラムが多くのタイムスライスを取得し、公平性に欠ける状況 [2] が発生する場合もある。

4 実験方法

O(1) スケジューラと CFS との特性を考慮すると、マシン上に存在するプロセス数が、仮想マシンに影響

A study of difference Linux process schedulers impact on VM environments

† Kenichiro Higashi

Hitachi, Ltd., Platform Solutions Division (†)

を与えると予想できる。Linuxのプロセススケジューラは、KVMの仮想マシンも通常のプロセスとして扱うため、KVMの仮想マシン数を考慮した実験を実施することにした。

まず、ホストOSとしてO(1)スケジューラとCFSとを実装している2種類のLinuxを個別に準備する。各Linux上において、性能測定を実施する仮想マシン(Target VM)と、マシン上で動作するプロセス数を増やす目的で生成する仮想マシン(Dummy VM)とを準備し、Dummy VM増加に伴うTarget VMの性能変化の傾向を確認する。表1に実験環境を示す。

表 1: 実験環境

CPU	Core 2 Duo E6600 2.40GHz	
メモリ	4GB	
Linux カーネル	2.6.21.3-85	2.6.33.3-85
スケジューラ	O(1)	CFS
Target VM	Ubuntu10.04	
Dummy VM	Puppy Linux 4.3.1	

ホストOSとして、O(1)スケジューラを実装したLinux2.6.21.3と、CFSを実装したLinux2.6.33.3とを使用した。また、性能測定を実施するTarget VMとして、1GBのメモリを割り当てたUbuntu10.04を準備し、Dummy VMとして128MBのメモリを割り当てたPuppy Linux4.3.1を準備した。

Dummy VMは、0個から16個まで増加させ、Dummy VMの個数ごとに、Target VM上でUNIX Benchを3回づつ実行し、その平均を実験結果とした。UNIX Benchでは、CPU演算、I/O処理、プロセス生成、システムコールなどの負荷をかけ、ベンチマーク結果をINDEX値として表現し、高いINDEX値は性能が良いことを表す。性能測定終了後は物理マシンを再起動し、不要なプロセスを生成していない環境で実験を実施した。また、Dummy VMで動作するプロセスはGUI環境の起動のみに抑え、プロセス数増加に伴うTarget VMの性能測定に焦点を当てた。

5 結果と考察

図2に、Target VMの性能測定を実施した結果を示す。

O(1)スケジューラ上で動作するTarget VMは、Dummy VM数が増加してもINDEX値の変化は少なく、著しい性能劣化は起きなかった。

一方、CFS上で動作するTarget VMは、Dummy VM増加に伴いINDEX値が減少し、Dummy VM数が16個の環境では、O(1)スケジューラと比較して2割程度まで性能が低下した。また、プロセス優先度を設定するniceコマンドに-5を指定し、プロセス優先

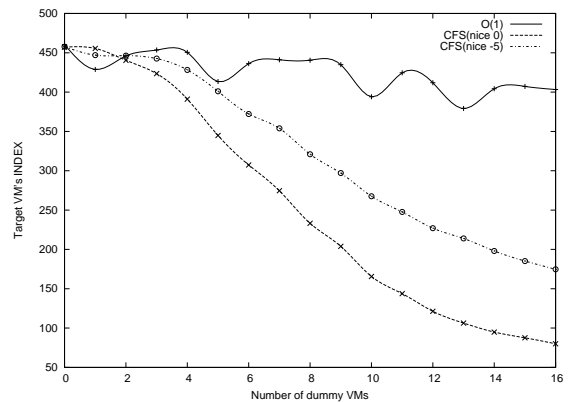


図 2: 実験結果

度を上げた場合でも、CFS上で動作するDummy VM数16の環境では、O(1)スケジューラと比較して4割程度まで性能が劣化した。

KVMを使用した仮想マシン集約を実施する場合、ハードウェアスペック、仮想マシン数、仮想マシンのプロセス優先度だけではなく、プロセススケジューラの特長も考慮する必要がある。

6 今後の課題

本研究の結果から、プロセススケジューラの違いが、仮想マシンの性能に影響を与えることが明確になった。今後は、プロファイラを用いた精度の高い検証を実施し、性能差が出る具体的な原因を調査する予定である。

プロセススケジューラの違いで性能差が生じる問題の抜本的な解決策として、仮想化環境に適したプロセススケジューラの改良も検討できる。しかし、一般的なプロセスと、仮想マシンもプロセスとして混在するKVMのような環境では、プロセススケジューラの改良だけで各プロセスに最適なりソース配分を実施することは困難である。

そこで今後は、プロファイラを用いた検証結果を基に、Linuxカーネルのプロセススケジューラとは独立した、仮想マシンの負荷監視とリソース割り当て最適化を提供する機能を提案する予定である。

参考文献

[1] Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin, and Anthony Liguori. kvm: the Linux Virtual Machine Monitor. In *OLS '07: the 2007 Ottawa Linux Symposium*, volume 1, pages 225–230, Ottawa, Ontario, Canada, June. 2007.

[2] Chee Siang Wong, Ian Tan, Rosalind Deena Kumari, and Fun Wey. Towards Achieving Fairness in the Linux Scheduler. *SIGOPS Oper. Syst. Rev.*, 42(5):34–43, 2008.