

# サーバ仮想化における Linux I/O スケジューラの活用

太田 毅<sup>†</sup>, 岡本 秀輔<sup>‡</sup>

<sup>†</sup> 成蹊大学大学院 理工学研究科

## 1 はじめに

コンピュータリソースの柔軟な利用や、コンピュータの電力消費削減を目的として、サーバ仮想化の利用が最近広まっている。管理者は、複数の VM(仮想マシン) がホストマシンのリソースを共有することによる負荷を考慮して、VM の資源割当の管理を行う必要がある。我々は、特に負荷が大きい VM 上の I/O 処理を I/O スケジューラを使用して管理できるかどうかをいくつかの実験を通して調査した。また、LinuxOS に標準搭載されている CFQ スケジューラで使用できる I/O 優先度に注目し、VM 上の I/O 処理に優先度を与えて管理する手法の有効性を調査した。

## 2 背景・目的

仮想化環境では、管理者は複数の VM がホストマシンのリソースを共有することによる負荷を考慮しなければならない。特に、複数の VM 上の I/O 処理によるホストマシンの I/O 資源の競合は、仮想化システム全体のボトルネックとなる。我々は、この問題のために、Linux I/O スケジューラを用いる方法に着目した。

この論文内で使用する Linux カーネル 2.6.32 は、4 種類のスケジューラを搭載しており、それぞれ異なる特徴を持っている。CFQ(Completely Fair Queuing) は、各プロセスごとに I/O 命令のキューを持っており、すべてのプロセスの I/O 処理が公平に実行されるようにスケジューリングする。また、CFQ では高い順に Realtime、BestEffort、Idle の 3 つの I/O 優先度を持っており、プロセスごとにこれらの優先度を割り当てることもできる。プロセスは、優先度を与えられたタイムスライス分だけ I/O を実行する。同じ優先度を与えられた複数のプロセスが I/O 要求を同時に行った場合は、要求が到着した順番に I/O を実行する。デフォルトでは、OS にとって重要なプロセスのみ優先度を高く、その他のプロ

セスは同じ優先度に設定される。Deadline スケジューラは、キューの先頭の I/O を実行してからある一定の期限がすぎると、その I/O を後回しにして次の I/O を実行する。Anticipatory スケジューラは、I/O 要求をすぐ処理せず、ある程度 I/O 要求が来るまで待機し、ディスクシーク時間を最小にするように、I/O 要求を並び替えてから処理を行う。Noop スケジューラは I/O 処理の並び替えは行わないスケジューラである。

文献 [1][2] では、VM を複数動作させるホストマシンと VM の両方でこれらの I/O スケジューラを使用した際、スケジューラの選択が VM 上の I/O 処理に影響を与えることを示した。しかしこれらの結果は SSD を考慮に入れていない。SSD は従来の磁気ディスクとは異なり、シーク時間を考慮に入れる必要がない。我々は、文献 [3] において、SSD を前提とした仮想化環境における I/O スケジューラの特性の調査を行った。その結果、VM 上でファイル書き込み処理を行う際、ホストマシンと VM の I/O スケジューラの組み合わせによって、実行時間が変動することを確認した。また VM 上で、シーク時間を考慮に入れる anticipatory スケジューラを使用した時、実行時間の遅延を引き起こした。本論文では、この結果を考慮にいれ、ホストマシン上の CFQ による VM への I/O 優先度の割り当てが、より効率的なスケジューリングとなるかどうかを調査する。

## 3 仮想化環境における I/O 優先度の利用

VM をホストマシン上の 1 つのプロセスとして扱い、CFQ で I/O 優先度を割り当てた時、VM 上の I/O 処理が受ける影響を調査する。実験で使用するホストマシンは、クアッドコア CPU × 2、12GB メモリ、64GB SSD (読み込み速度:116MB/sec、書き込み速度:43MB/sec) を搭載している。ホストマシンの OS には UbuntuOS を使用し、仮想化は KVM を用いて行った。VM には、1 台につき CPU を 1 つ、1GB メモリ、10GB の仮想ディスクを割り当て、OS は UbuntuOS とした。尚、VM 上のスケジューラは Deadline とする。

実験にあたり、VM 上で動作させるワークロードを 2 つ用意した。1 つ目は、Sqlite3 を用いて、約 17MB

### Using Linux I/O schedulers in virtualization environment

Tsuyoshi Ota<sup>†</sup>, Shusuke Okamoto<sup>‡</sup>

<sup>‡</sup>Department of Information Science Seikei University  
Tokyo, 180-8633 Japan  
{dm106206@cc.seikei.ac.jp, okam@st.seikei.ac.jp }

のデータを書き込むプログラムである。このプログラムでは、データベースを作成する際にトランザクション処理を行わずにI/Oを実行する。2つ目は、同様に、約1GBのデータを書き込むプログラムであるが、トランザクション処理を行い、複数のI/O処理を1つのI/O処理として実行する。我々は、実験で使用するVMの内、半分のVM上でこれらのワークロードを動作させ、残りのVMでは全体に負荷をかけるI/O処理として、I/Oベンチマークとして知られるBonnie++を動作させた。

用意したワークロードを動作させるVM群の優先度を最も高くした時、最も低くした時、そして優先度なしの場合の実行時間を図1、図2に示す。図1は、トランザクション処理を行わないため、書き込み処理ごとに仮想ディスクにアクセスしており、オーバーヘッドが大きく、非常に実行に時間がかかっている。一方で、図2は、トランザクション処理によって、ある程度一括で仮想ディスクへの書き込みを行うため、実行時間は小さい。どちらの図においても、ワークロードの種類に関わらず、マシンの台数が増加するほど、優先度による影響が大きくなっていることがわかる。これはホストマシンのスケジューラが扱うプロセス数が多くなることから、要因として考えられる。しかし、トランザクション処理を行っていない図1では、VM台数が増加しても、優先度と実行時間に関連性が見られず、VM群に高い優先度を割り当てても、優先度の高い処理とはなっていない。これは、ホストマシンのスケジューラ上で管理できるI/O処理の量が少ないためである。トランザクション処理を行った図2では、プロセス数が多く、かつ同時にスケジューラできるI/O処理の多い4VM、8VMの時、高い優先度を割り当てたVM群の実行時間が最も小さくなっている。8VMでは、優先度が高い時の実行時間は、優先度が低い時の実行時間の約89%まで実行時間を縮小させている。

#### 4 考察

今回の実験で、ホストマシンのCFQを用いてVMごとに優先度を与えた結果、優先度の高いVMは、低いVMの約89%の時間でI/O処理を実行することができた。しかし、最も高い優先度を割り当てたVMは2台以上存在しているため、そこでも競合が発生していることが予測される。したがって、最も高い優先度を1つのVMのみに与えることによって、さらに他のVMのI/O処理と差異を付けられると考えられる。またCFQのタイムスライス値は自由に設定することが

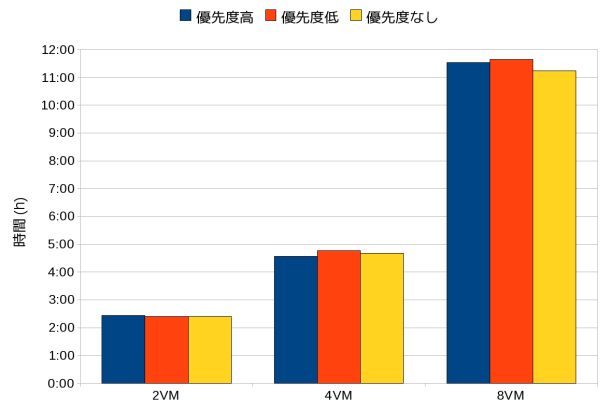


図1: 優先度を与えた時のVMのI/O処理の実行時間 (トランザクション処理なし)

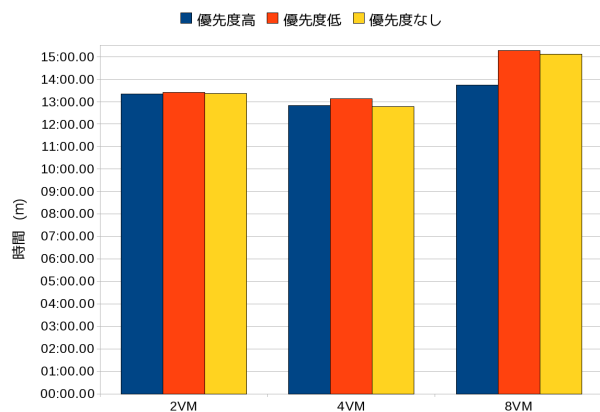


図2: 優先度を与えた時のVMのI/O処理の実行時間 (トランザクション処理あり)

できるため、さらに多くの時間を優先度の高いVMに与えることが可能である。これらの結果から、この手法がVM上のI/O処理をホストマシン上で管理できる可能性を持っていると言える。

#### 参考文献

- [1] M. Kesavan, A. Gavrilovska, K. Schwan, "On Disk I/O Scheduling in Virtual Machines", the Second Workshop on I/O Virtualization (WIOV'10), pp 6-6, 2010.3
- [2] D.Boutcher, A.Chandra : "Does Virtualization Make Disk Scheduling Passe?", ACM SIGOPS Operating Systems Review, Volume 44 Issue 1, January 2010
- [3] 太田 毅, 岡本秀輔 : "プライベートクラウドにおける電力消費量", IEICE SIG Reports, CyberWorld 10, pp.1-6, 2010.3