

複数の仮想マシンが動作する環境における効率的なディスクキャッシュ管理手法の提案

岡本 崇典[†] 芝 公仁[†]

[†] 龍谷大学理工学部

1 はじめに

近年、仮想化技術を利用するクラウド環境では一台の物理サーバ上で複数の仮想マシン(以下 VM)が動作することが増えてきている。このような環境で動作する VM が他の VM と共通するデータを読み出す場合、ホスト OS の管理するキャッシュを利用することでディスクアクセスを減らし、データ読み出し速度を向上させることができる。

ただし、各 VM が読み込むデータには複数の VM で共通のものと、各 VM 固有のものがある。したがって、各 VM が読み込んだデータを全てキャッシュしていくとホスト OS のキャッシュ領域には、複数の VM 間で共有できるデータとそうでないデータの両方が存在することとなる。

また、仮想化環境においては、ゲスト OS のキャッシュにないデータへのアクセスは、ホスト OS のキャッシュ領域を介して HDD へアクセスされる。このような環境において、ゲスト OS のキャッシュ置き換え方式に LRU が使用されている場合、ホスト OS 側のキャッシュは、一度アクセスされたデータが近い将来再度アクセスされる可能性が低くなり、通常とは逆向きの負の参照の局所性が存在することが確認されている [1, 2]。

VM から読み出したデータがその VM 固有のものである場合、ホスト OS がそれをキャッシュしても、そのデータを読み出した VM から他の VM から利用されない。そのため、VM 固有のデータはホスト OS 側でキャッシュしても効果的に機能しない。

以上のことから、複数の仮想マシン間で共有可能なデータのみをホスト OS 側のキャッシュ領域に置くことで、より多くのデータをキャッシュでき、キャッシュのヒット率が向上し、データの読み出し速度を向上させることができると考えられる。

2 提案するキャッシュ制御手法

既存のキャッシュ管理手法では、共有できるデータと VM 固有のデータの両方をキャッシュしていく。このとき、ホスト OS 側のキャッシュ領域には VM 固有の共有することができないデータが存在するため、各 VM

で共有できるデータを利用する場合、キャッシュヒット率が低下し、データの読み込み速度が低下する。

本章では、上記の問題を考慮したキャッシュの制御手法について述べる。本手法では、共有できるデータのみをホスト OS 側にキャッシュすることで上記の問題に対処し、既存のキャッシュ方式よりデータの読み込み速度の向上を図る。本稿では、QEMU を用いた提案手法の実装について述べ、その有効性を示す。

提案手法の構成を図 1 に示す。実装したシステムでは、ゲスト OS からの I/O 要求はすべて QEMU 内の I/O 制御部が受けとる。I/O 制御部で、ゲスト OS からの I/O 要求を受け取り、読み出しを行うデータが各 VM 間で共有可能かその VM 固有であるかを判別し、共有可能であればホスト OS がキャッシュする読み出しを行い、VM 固有であればキャッシュしない読み出しを行う。

読み出しを行うデータが共有可能か VM 固有であるかは読み出しの対象となるファイルで判別する。具体的には、各 VM 間で共有可能なデータと VM 固有のデータを分けて格納した二種類の仮想 HDD イメージファイルを利用し VM を動作させることで、読み出すデータが共有可能か VM 固有かを判別する。

実装したシステムでの I/O 処理の流れは以下のようになる。

- (1) ゲスト OS が仮想 HDD からデータの読み出しを行う
- (2) I/O 制御部で読み出すデータが VM 間で共有可能か VM 固有かを判別し、共有可能であればキャッシュする読み出し、VM 固有であればキャッシュしない読み出しを行う
- (3) ホスト OS が物理 HDD からデータを読み出す
- (4) I/O 制御部へ読み出し完了を通知する
- (5) ゲスト OS へ読み出し完了を通知する

このシステムにより、共有できるデータのみホスト OS 側にキャッシュし、読み出しを行うことが可能となる。

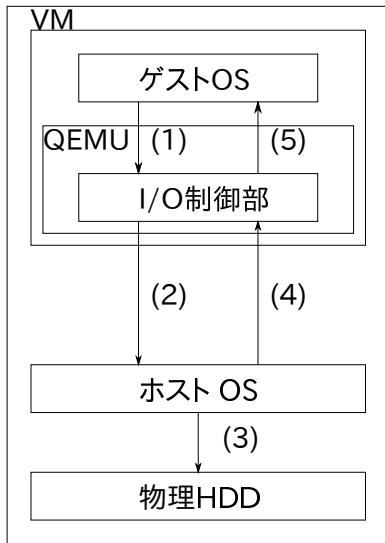
3 評価

本手法の効果を確認するために、各 VM から共有可能なデータと各 VM 固有のデータそれぞれに対しランダムリードを行い、読み出し速度を測定する実験を行った。それぞれのデータサイズは 5GB から 50GB まで変

An Efficient disk cache management method for multiple virtual machines

Takanori Okamoto[†], Masahito Shiba[†]

[†] Faculty of Science and Technology, Ryukoku University



物理マシン
図1 実装構成

表1 実計算機の仕様

| Name | Description |
|-----------------------|----------------------------|
| OS | Debian 8.6 (64bit) |
| Kernel | 3.16-2-amd64 |
| CPU | Intel (R) Core(TM) i5-4460 |
| HDD | 2TB |
| Memory | 16GB |
| Virtualization system | KVM |
| File system | ext4 |

表2 仮想計算機の仕様

| Name | Description |
|-------------|--------------------|
| OS | debian 8.6 (64bit) |
| Kernel | 3.16.0-4-amd64 |
| HDD | 250GB |
| Memory | 2GB |
| File System | ext4 |

化させ、データサイズ分だけ読み出しを行う。ランダムリードのファイルアクセスの分布は正規分布に従う。なお、同時に読み出すVM固有のデータと共有可能なデータの量は同じものとする。

本実験では5台のVMを使用した。5台のVMの読み出し速度の平均を図2、図3に示す。縦軸が5台のVMの読み出し速度の平均、横軸が読み出したデータサイズである。また、実験環境を表1、表2に示す。

図2より、共有可能なデータの読み出しにおいて、データサイズがいずれの場合でも読み出し速度が向上していることが確認できる。

本実験より、ホストでキャッシュした場合はしなかった場合より、速度が向上していることが確認できる。このことから、VM間で共通したデータの読み出しを行う場合はホストのキャッシュを利用することで読み出しの

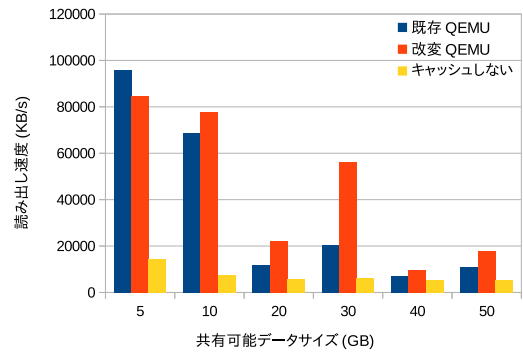


図2 共有可能なデータの読み出し速度

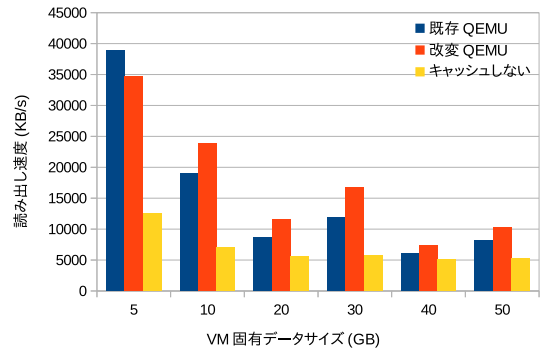


図3 VM固有のデータの読み出し速度

性能が向上することがわかる。また、変更したQEMUの共有可能なデータの読み出し速度は、既存のQEMUと比べ向上していることが確認できる。これは、変更後のQEMUではVM固有のデータをキャッシュしないことで変更前の環境より共有可能なデータを多くホスト側にキャッシュし、キャッシュのヒット率が向上したためであると考えられる。

4 おわりに

本稿では、複数のVMが動作する環境において、共有できるデータのみをホストOS側にキャッシュし、読み出す手法の実装と評価を行った。評価の結果、提案手法により、読み出し速度を向上させることができた。

参考文献

- [1] 宮野新平, 山口実靖, 浅谷耕一: 多段キャッシュ型ネットワークストレージへのアクセスの時間的局所性を考慮したメモリキャッシュ制御, 情報処理学会研究報告. マルチメディア通信と分散処理研究会報告 2009, No. 20.
- [2] 杉本洋輝, 山口実靖: 二重キャッシュ環境における負の参照の時間的局所性を考慮したキャッシュ管理手法, マルチメディア, 分散, 協調とモバイル (DICOMO2014) シンポジウム