

仮想化環境におけるシステム全体の I/O 性能の向上を目的とした メモリスケジューラの提案

氏名[†] 田島 幸恵

所属[†] (株)日立製作所 横浜研究所

氏名^{††} 野中 裕介

所属^{††} 同 横浜研究所

氏名[‡] 竹内 理

所属[‡] 同 横浜研究所

氏名^{‡‡} 三留 浩幸

所属^{‡‡} 同 ITプラットフォーム事業本部

1. はじめに

物理マシン上に複数の仮想マシン (VM) が稼動する時、各 VM の性能向上のためにメモリ等のリソースの効率的な割当てが求められる。

従来の仮想化機構 [1] は、スワップ等の発生を契機にメモリ割当て調整を行うなど、VM の状況に応じた自動リソース管理を行っている。しかし、互換性を重視して VM 上の OS は改変しないため、メモリ割当て量の判断に利用可能な情報は限られている。例えば VM 毎の OS のディスクキャッシュ (以下、キャッシュ) の利用状況を把握し、キャッシュヒット率が高い VM により多くのメモリを割り当てるような調整はできない。また、ユーザが手動でリソース管理を行う方法も提示されている [2]。しかし、変化する VM の状況に追従して手動で対応するのは困難である。

本稿では、VM へのメモリ割当て量を調整するキャッシュウェアメモリスケジューラ (以下、スケジューラ) の方式を提案し、その有効性検証結果を報告する。提案方式では、OS のキャッシュヒット状況の通知機構を用いて、キャッシュヒット率の高い VM に優先的にメモリを割り当て I/O 性能の向上を図る。

2. 提案方式

システム全体の I/O 性能向上方法として、キャッシュヒット率 (ヒット I/O 数/実施 I/O 数) の向上が挙げられる。単位時間当たりにヒットするキャッシュページの数 (以下、ヒットページ数) が多い程キャッシュヒット率は向上する傾向があるため、ヒットページ数の増加を図る。

VM への割当てメモリ量を増やすと VM のキャッシュページ数が増加する。そのため、よりキャッシュヒットが期待できる VM によりメモリを割り当てることで、システム全体のヒットページ数の増

加が可能と考えられる。VM のキャッシュヒットの期待度は、過去にキャッシュがヒットページを含有した度合いを示すページ利用効率から推定する。VM のページ利用効率を以下の式で定義する：

VM のページ利用効率

$$= \text{ヒットページ数} / \text{キャッシュページ数}$$

システム全体のヒットページ数を増加させるために、本推定に基づき、利用効率の低い VM から回収したメモリを利用効率の高い VM に割り当てる。提案方式では、OS のカーネルが VM のページ利用効率を監視してスケジューラに提供し、スケジューラは利用効率の高い VM に優先的にメモリを割り当てるメモリ割当て量の調整を繰り返し行う。

本方式を実現するソフトウェア構成図を図 1 に示す。

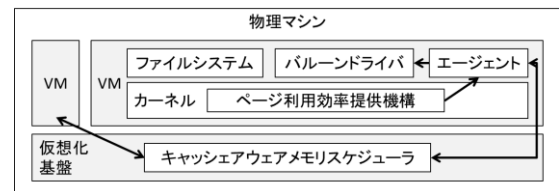


図 1 ソフトウェア構成図

構成要素を以下に示す。

- (1)-(a) キャッシュウェアメモリスケジューラ
 - ◇ 提案方式に基づき VM のメモリ割当て量を決定
- (2)-(a) ページ利用効率提供機構
 - ◇ VM のページ利用効率を提供
 - kernel-3.10.9-200.fc19 を 90 行改変
- (2)-(b) エージェント
 - ◇ 利用効率情報をスケジューラに通知
 - ◇ スケジューラの指示に従いバルーンドライバにメモリ割当て量を通知
- (2)-(c) バルーンドライバ
 - ◇ エージェントの指示に従いメモリ割当て量を

Proposal of memory scheduler which improve the I/O performance of a system in virtualization environment

[†]Sachie Tajima, Yokohama Research Laboratory, Hitachi, Ltd.

[‡]Tadashi Takeuchi, Yokohama Research Laboratory, Hitachi, Ltd.

^{††} Yusuke Nonaka, Yokohama Research Laboratory, Hitachi, Ltd.

^{‡‡} Hiroyuki Mitome, IT Platform Division Group, Hitachi, Ltd.

変更

3. 有効性検証実験

実運用を想定したユースケースに対する提案方式の I/O 性能向上効果を検証する。検証は前節のソフトウェア構成図に以下の変更を加えたシミュレート構成で行なう。シミュレートは検証結果には影響しないと考えている。

- スケジューラをリモート PC 上に実装
- バルーンドライバをユーザプロセスとして実装
- ディスク I/O の実行を省略したテスト用ファイルシステムを実装

以下の検証目的を持つ 2 ユースケースを用いる。

- ケース 1: スケジューラ適用により I/O 性能向上が期待できるケースの性能向上率検証 (割当メモリ量の増加によりキャッシュヒット率向上が期待できる VM (表 1 ケース 1VM2) を含む)
- ケース 2: スケジューラ適用による I/O 性能向上が期待できず低下も懸念されるケースの性能低下率検証 (メモリ均等割当でもキャッシュヒット率が十分高い VM (表 1 ケース 2VM2) を含む)

各ケースでは表 1 に示す VM1, VM2 を稼働させる。スケジューラ適用時とスケジューラ非適用時(メモリ均等割当)のシステム全体のキャッシュヒット率を比較する。キャッシュヒット率から, IOPS で示される I/O 性能が導出できる。

表 1 実験対象のユースケース

	ケース 1	ケース 2
VM 1	Web サーバ/通常アクセス型 ● 100 ユーザが 6 回/分のアクセスを実施 ● I/O 空間局所性:低 (メモリ均等/調整割当時も, 大部分のアクセス範囲はキャッシュに収まらないケース)	
VM 2	Web サーバ/突発アクセス型 ● 突発的に一部のコンテンツに対して通常アクセス型の 100 倍のアクセスが発生 ● I/O 空間局所性:中 (メモリ均等割当時にはアクセス範囲がキャッシュに収まらないケース)	VM イメージが格納された NFS サーバ ● 格納している VM イメージを起動 ● I/O 空間局所性:高 (アクセス範囲が狭く, メモリ均等割当時にアクセス範囲がキャッシュに収まるケース)

4. 有効性検証結果

図 2 はケース 1 の実験結果を示す。スケジューラ適用により VM1 から回収されたメモリが VM2 に割当たつことを確認した。VM1 のヒット率は低下

せず VM2 のヒット率は向上し, 全体のヒット率が最大で約 1.5 倍向上することを確認した。

図 3 はケース 2 の実験結果を示す。スケジューラ適用により VM1 から回収されたメモリが VM2 に割当たつことを確認した。VM1, VM2 ともにヒット率に変化はなく, 全体のヒット率は向上も低下もないことを確認した。

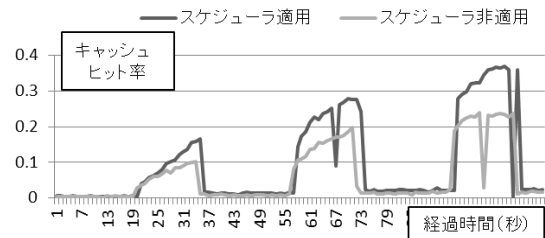


図 2 アルゴリズム適用結果 (ケース 1)

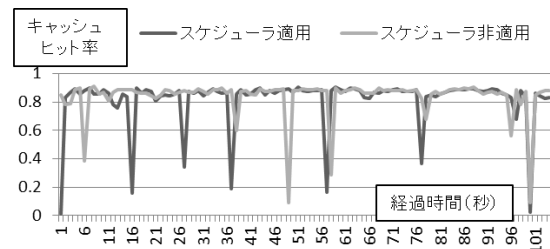


図 3 アルゴリズム適用結果 (ケース 2)

以上の検証結果から, 本スケジューラはケース 1 の VM2 のような, 割当メモリ量によってヒット率が変化する I/O の空間局所性:中の VM を含むケースに対して効果があることを確認できた。

5. おわりに

VM のキャッシュヒット状況に応じた自動リソース管理を実現するキャッシュウェアメモリスケジューラの方式提案および有効性検証を行った。

実運用を想定したユースケースで検証を行なった結果, 割当メモリ量によりヒット率が向上する VM を含むケースでキャッシュヒット率が向上することを確認し, I/O 性能向上効果があることを示した。また, 空間局所性が十分高い VM に対して性能向上効果はないが, 性能低下も見られないことを確認した。組み合わせる VM の局所性の特性によっては性能低下の可能性も考えられるため, 更なる検証と方式改善を継続する。

6. 参考文献

[1] Automatic Ballooning, <http://www.linux-kvm.org/page/Projects/auto-ballooning> (参照 2013/12/19)

[2] QEMU Emulator User Documentation, <http://qemu.weilnetz.de/qemu-doc.html> (参照 2013/12/19)