Vol.2014-DPS-159 No.31
Vol.2014-MBL-71 No.31
2014/5/16

仮想化環境における 読込み書込み比率を考慮した動的 VM メモリ割り当て

坂本　雅哉[†1]　山口　実靖[†1]

近年，サーバの消費電力増加，設置スペース肥大化が問題となっており，その解決策の一つとして，仮想化技術を用いて複数の仮想マシンを一台の物理マシンに集約する手法がある．仮想化環境では，仮想マシンを停止させることなくメモリの割り当て量を変更することが可能である．一つの物理マシンにて複数の仮想マシンを稼働させ，それぞれの仮想マシンの負荷が時間により変化する場合は，静的なメモリ割り当てを行うとメモリを効果的に活用できないことになる．負荷変動に対応するためには，動的に仮想マシンのメモリ割り当て量を変更する必要がある．本稿では，ゲスト OS のアプリケーションの読み書き比率，キャッシュヒット率を考慮したメモリ割り当て量の最適化の手法を評価によりその有効性を示す．

1. はじめに

近年，情報技術が普及し，データセンター等において多数のサーバ計算機が稼動するようになった．これに伴い，サーバの消費電力の増加，設置スペースの肥大化が問題となっている．この問題に対する解決策の一つとして，仮想化技術を用いて複数のサーバ OS を一台の物理マシンに集約する手法がある[1].

仮想化技術では，仮想マシン（VM）のメモリ割り当て量の設定が可能であり，VM を停止させることなくメモリ量を変更することが可能である．一つの物理マシンにて複数の VM を稼働させ，それぞれの VM の負荷が時間により変化する場合は，静的なメモリ割り当てを行うとメモリを効果的に活用できないことになる．負荷変動に対応するためには，動的に VM メモリ割り当て量を変更する必要がある．

仮想化ソフトウェアの Xen には，動的に VM メモリ割り当て量を変更させる機能として xenballoon がある．xenballoon がメモリ割り当てを行うために考慮するパラメータは，VM 内のプロセスが使用するメモリのみであり，ページキャッシュとして利用されるメモリを考慮していない．そのため，I/O 性能の低下を招くと考えられる．

本研究では，Xen によって提供される VM で，メモリ割り当て量，読込み書込み比率，キャッシュヒット率，I/O 性能の関係性について調査を行い，読込み書込み比率，キャッシュヒット率を考慮した VM メモリ割り当て量の最適化の手法の提案を行う．

2. Xenballoon

xenballoon は，Xen が持つ機能の一つで，VM に割り当てるメモリ割り当て量の変更を動的に行う機能である．VM 上でデーモンとして動作し，ゲスト OS 内のプロセスの推定メモリ使用量(Commited_AS)を監視して，ホスト OS への要求メモリ量を調整する[2].

異なるゲスト OS 間でメモリ量調整を行う機能が無く，ホスト OS へ物理メモリ以上のメモリ量を要求してしまう事がある．この場合，メモリ量のオーバーコミットによるエラーなどは発生しないが無いが，先にホスト OS にメモリを要求したゲスト OS がメモリ要求の優先権を得られる．そのため，特定のゲスト OS がメモリを占有してしまい，他のゲスト OS のメモリ量が増えないという事象が発生する．

3. キャッシュヒット率，読み書き比率の解析

3.1 キャッシュヒット率の解析

ページキャッシュヒット率を測定するために，図 1 のようにホスト OS，ゲスト OS のカーネル内でキャッシュ前後の I/O 量を監視した．

キャッシュ前の I/O 量としてページキャッシュへのアクセス量を測定するためにカーネルの改変を行った．ゲスト OS では，カーネル構成ファイルの `mm/filemap.c` 内にある「`find_get_page`」関数の実行回数とブロックサイズの監視，ホスト OS では，`drivers/xen/blkback/blkback.c` 内にある「`dispatch_rw_block_io`」関数の実行回数とブロックサイズの監視をするための改変を行った．

キャッシュ後の I/O 量として VM の仮想ブロックデバイス(XVD)における I/O 量，物理 HDD における I/O 量を測定するためにカーネルの改変を行った．ゲスト OS では，「`do_blkif_request`」関数の実行回数とブロックサイズの監視，ホスト OS では，`drivers/scsi/sd.c` 内にある「`sd_prep_fn`」関数の実行回数とブロックサイズの監視をするための改変を行った．

測定したキャッシュ前後の I/O 量の比よりキャッシュヒット率を求めた．

†1 工学院大学大学院工学研究科電気電子工学専攻
Electrical Engineering and Electronics, Kogakuin University Graduate School

Vol.2014-DPS-159 No.31
Vol.2014-MBL-71 No.31
2014/5/16

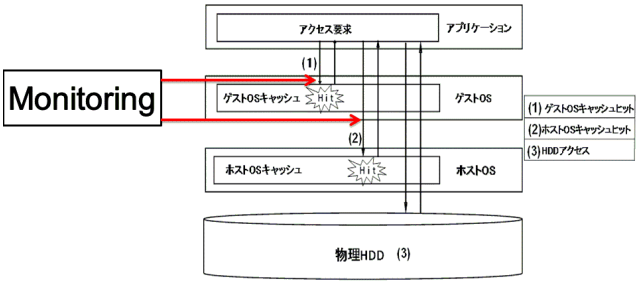

図 1　キャッシュヒット率の測定

3.2 読み書き比率の解析

読書き比率を測定するために，カーネル内での以下の値を測定した．

読込み量の測定は，前節同様に `mm/filemap.c` 内にある「`find_get_page`」関数の実行回数により測定した．

書込み量の測定は，アプリケーションから発行された書込み要求量と仮想 HDD に発行される書込み量が等しいという前提のもと，`drivers/xen/blkfront/blkfront.c` 内にて定義されている I/O 要求を処理する関数「`do_blkif_request(request_queue_t *rq)`」における発行 I/O 要求サイズ量(`req→bio→bi_size`)の測定により測定した．

4. キャッシュヒット率，読込み書込み比率とスループットの関係

本章にて，キャッシュヒット率，読込み書込比率と VM 上の I/O アプリケーションのスループットの関係を示す．

関係を調査するため，VM 上でランダムアクセスベンチマーク(FFSB)を実行した．ベンチマークでは，VM 内ファイルシステム上に 1MB のファイルを 5000 個(5000MB)作成し，これらのファイルに対して読込み書込み混在アクセスを行った．実験に用いた物理マシン，ホスト OS，仮想マシン，ゲスト OS の仕様を表 1 から表 4 に示す．

図 2 から図 9 に測定したキャッシュヒット率とスループットを示す．

図より，読込み比率が高いと，メモリ量を増やすことによる性能向上の程度が大きく，読込み比率が低いとメモリ量を増やすことの効果が小さいことがわかる．

また，読込み比率が高いものの中でも，キャッシュヒット率が高いものの方がメモリ割り当て量を増したことによる性能向上の効果が大きいことが分かる．

表 1　物理マシンの仕様

CPU	AMD Athlon 1640B (2.7GHz)
Memory	8[GB]
HDD	1[TB]

表 2　ホスト OS の仕様

OS	CentOS6.3
Kernel	Linux2.6.32.57
Memory	1024[MB]
仮想化システム	Xen 4.1.2

表 3　仮想マシンの仕様

CPU	AMD Athlon 1640B (2.7GHz)
Memory	動的に変更
HDD	100[GB](イメージファイルモード)

表 4　ゲスト OS の仕様

OS	CentOS 5.5
Kernel	Linux2.6.18.8
Memory	1000[MB]~7000[MB]

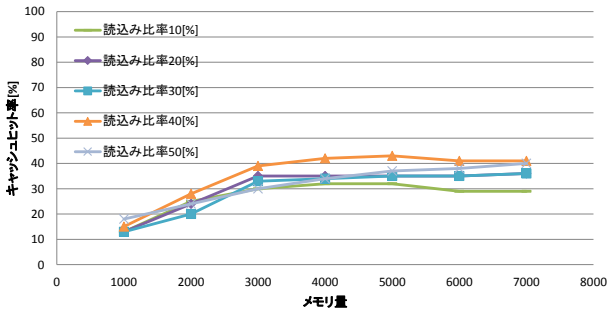

図 2　メモリ割り当て量とキャッシュヒット率の関係
(読込み比率 10－50[%])

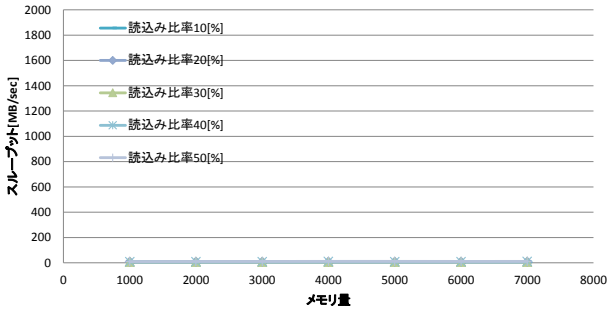

図 3　メモリ割り当て量と FFSB スループットの関係
(読込み比率 10－50[%])

Vol.2014-DPS-159 No.31
Vol.2014-MBL-71 No.31
2014/5/16

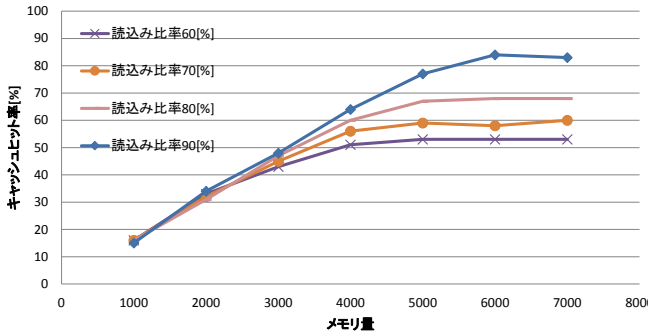

図 4　メモリ割り当て量とキャッシュヒット率の関係
(読込み比率 60－90[%])

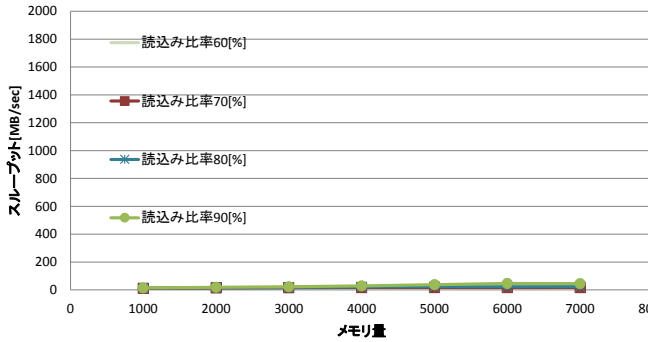

図 5　メモリ割り当て量と FFSB スループットの関係
(読込み比率 60－90[%])

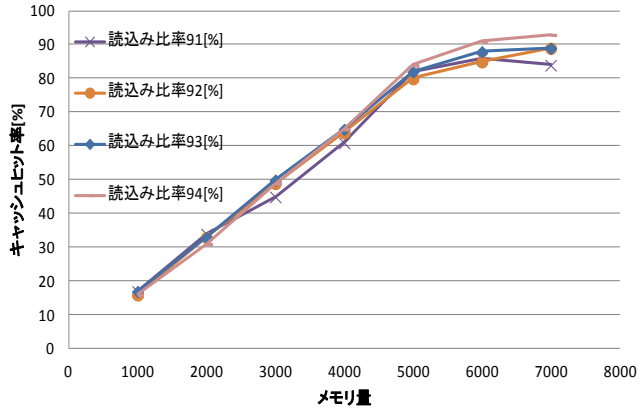

図 6　メモリ割り当て量とキャッシュヒット率の関係(読込み比率 91－94[%])

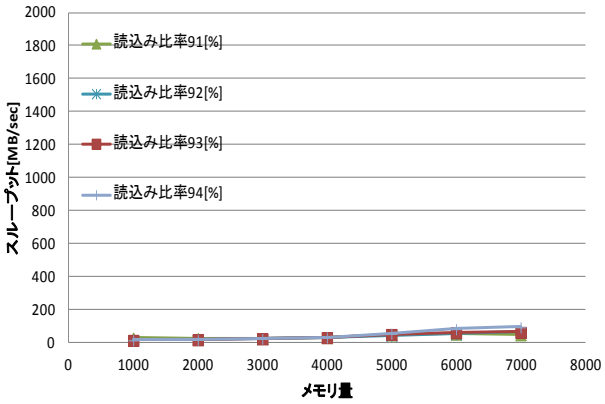

図 7　メモリ割り当て量と FFSB スループットの関係(読込み比率 91－94[%])

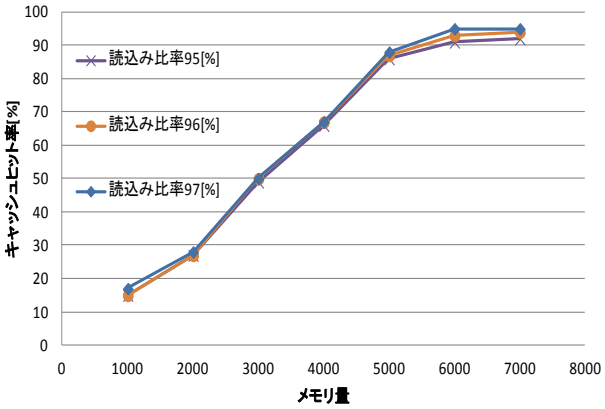

図 8　メモリ割り当て量とキャッシュヒット率の関係(読込み比率 95－97[%])

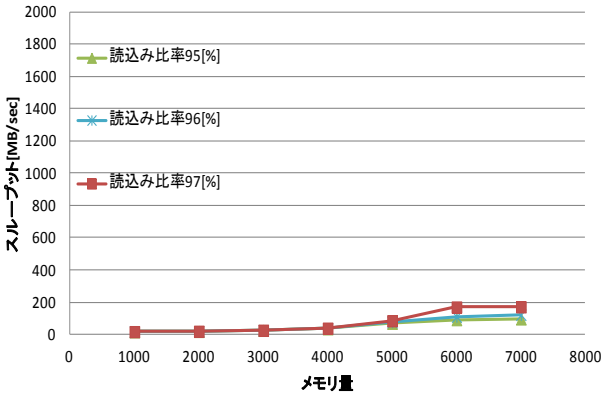

図 9　メモリ割り当て量と FFSB スループットの関係(読込み比率 95－97[%])

Vol.2014-DPS-159 No.31
Vol.2014-MBL-71 No.31
2014/5/16

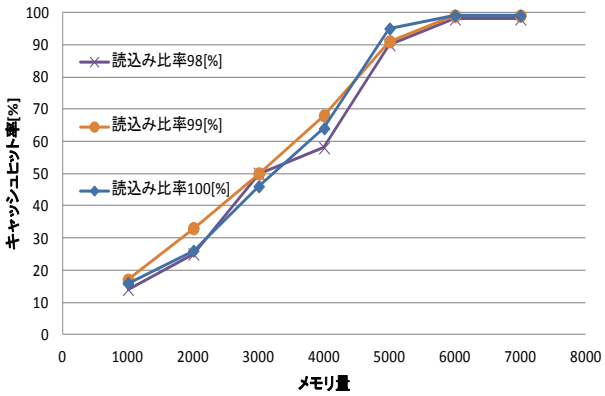

図 10　メモリ割り当て量とキャッシュヒット率の関係(読込み比率 98－100[%])

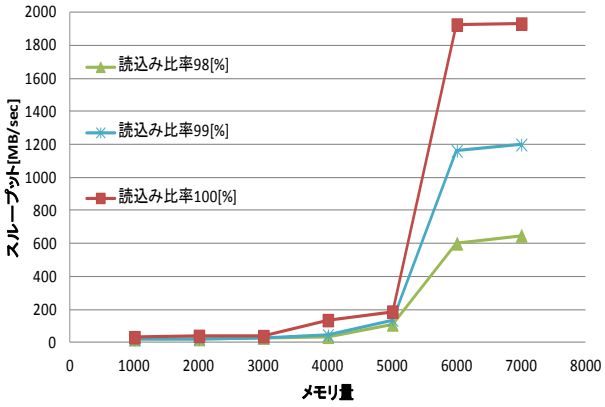

図 11　メモリ割り当て量と FFSB スループットの関係(読込み比率 98－100[%])

5. 既存手法

VM の I/O 性能向上を目的として VM メモリ割り当て量を調整する手法として，キャッシュヒット率に基づく手法[1]がある．

この手法は，各 VM のキャッシュヒット率を監視し，以下のメモリ割り当て方針にしたがってメモリを配分する．

1. キャッシュヒット率が閾値以上の VM のメモリ割り当て量を γ%減少させ，これを再配分用メモリとする．
2. 全 VM のメモリ割り当て量を δ%減少させ，これも再配分用メモリとする．
3. 上記 1, 2 にて得た再配分用メモリをキャッシュヒット率が閾値以上の VM を除く全ての VM に，VM のキャッシュヒット率により比例配分する．

しかし，この手法は読込み処理のみを考慮しており，読み書きが混同する環境では，適切な割り当てが行えないと考えられる．

6. 提案手法

本章では，読込み書込み比率とキャッシュヒット率を監視し，それを元に VM にメモリ割り当てを行う「読込み書込み比率とヒット率に基づくメモリ割当手法」を提案する．

本手法では，各 VM でキャッシュヒット率と読込み書込み比率を測定し，それをホスト OS で集計し，以下のルールに従い各 VM へのメモリ割り当て量を決める．

1. キャッシュヒット率が 99%以上の VM のメモリを 5%減らす．
2. キャッシュヒット率が 99%以上の VM を除くすべての VM のメモリを α%減らす．
3. キャッシュヒット率が 99%以上の VM を除く全ての VM にメモリを(読込み比率)^2×(キャッシュヒット率)の値により比例配分する．

キャッシュヒット率と読込み書込み比率の測定は 3 章で述べた方法で行う．α はチューニングパラメータである．

上記手法により，読込み比率が高い VM ほど多くのメモリを得ることができる．また，キャッシュヒット率が 100%を下回ると，大幅にスループットが悪くなることが 4 章の実験から分かっているため，キャッシュヒット率が高い(99%以上)の VM のメモリ量の変更周期を 35 秒とし，それ以外の VM の周期は 15 秒とした．これにより，高い性能を長い時間維持できると期待される．

提案手法の実装は以下の通りとなっている．ゲスト OS 上で動作する Xenballoon の実行スクリプトを改変し，ゲスト OS からホスト OS にキャッシュヒット率と読込み書込み比率を通知可能とした．ホスト OS 上で，全 VM の上記両比率を集計し，VM のメモリ割当を行うスクリプト(ruby)を実行させた．

7. 性能評価

提案手法の有効性を検証するために性能評価実験を行った．本章にて，性能評価結果について述べる．

7.1 評価方法

実験では，Xen を用いて 1 台の物理マシン上に 3 台の VM を立ち上げ，全 VM 上でベンチマーク(FFSB)を同時実行し，I/O 性能を測定した．FFSB の設定は表 5 の通りである．各 VM で表 5 のベンチマーク(1 回 10 分)を 10 回(合計 100 分)行い，スループットを測定する途中に与えられたメモリ量やキャッシュヒット率の推移を観察した．実験に用いた物理マシンの仕様を表 6 に示す．仮想マシンの仕様は 4 章と同様である．本提案手法は 4 章の基礎調査をもとにしているが，基礎調査と異なる環境に置いても有効である

かを検証するために本章の性能評価ではあえて，4 章とは異なる物理マシンを用いている．

表 5　FFSB の設定

ファイルサイズ	1[MB]
ファイル数	1,1,100,1000,2000,3000,4000, 5000,7500,10000 個
読込み比率	68,84,96,97,97,98,98,99,99,100,100[%]
スレッド数	1 個
実行時間	600 秒

表 6　物理マシンの仕様

CPU	Intel(R) Celeron(R) CPU G530 @ 2.40GHz
Memory	8[GB]
HDD	1[TB]

7.2　I/O 性能の測定（既存手法と提案手法の性能比較）

図 12 に既存手法と提案手法と静的メモリ割当手法の性能を示す．静的メモリ割当手法とは全 VM のメモリを 2333[MB](均等割当)で固定したものである．提案手法の α は 15[%]とし，既存手法の γ は 0[%]，δ は 15[%]とした．既存手法の γ と δ は，既存手法で最も高い性能が得られる値[1]を用いた．図より，提案手法の性能が最も高いことが分かる．

図 13 に実験実行中の各 VM のベンチマークのデータサイズと読込み比率，提案手法による割り当てメモリ量の推移を示す．図より，データサイズが小さい VM に過剰なメモリを与えることを回避しつつ，読込み比率が高く，キャッシュヒット率が高い VM に多くのメモリを与えていることがわかる．

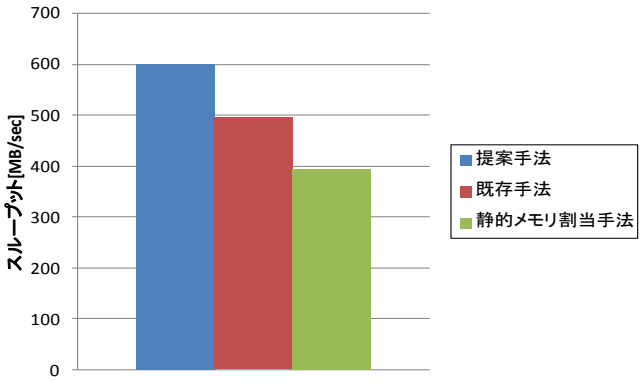

図 12　性能評価

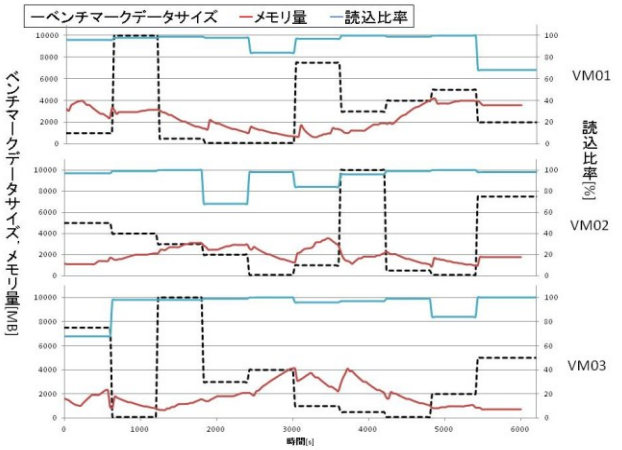

図 13　ベンチマークデータサイズと VM のメモリ量の関係 (α=15)

7.3　I/O 性能の測定（提案手法の α 変更の比較）

図 14 に α を 5，10，15，20，25，30[%]とした時の性能を示す．図より，α を 25[%]に設定した時が一番性能が高くなることが分かる．

図 15 から図 20 に VM のメモリ割り当て量の変化の推移を示す．

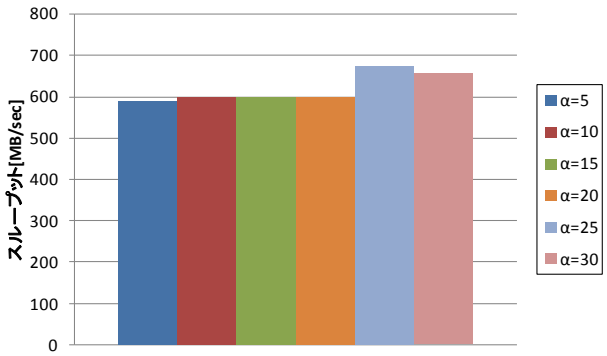

図 14　性能評価

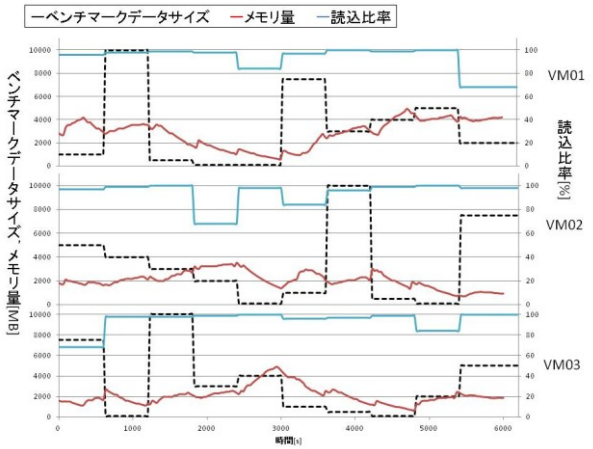

図 15　ベンチマークデータサイズと VM のメモリ量の関係 (α=5)

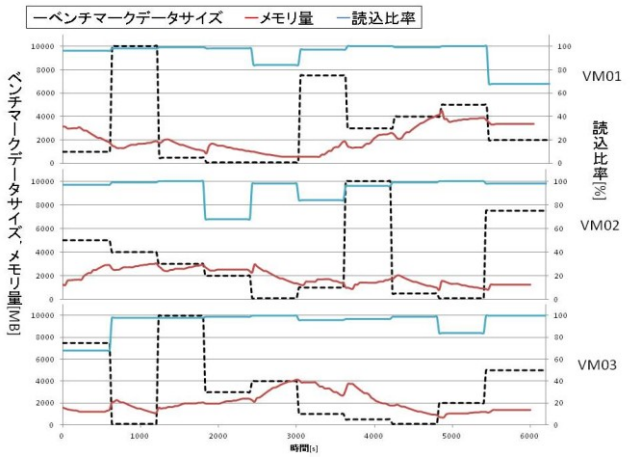

図16　ベンチマークデータサイズとVMのメモリ量の関係
(α=10)

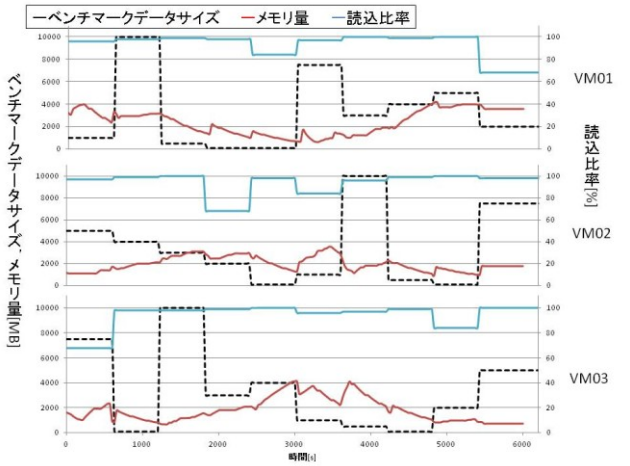

図17　ベンチマークデータサイズとVMのメモリ量の関係
(α=15)

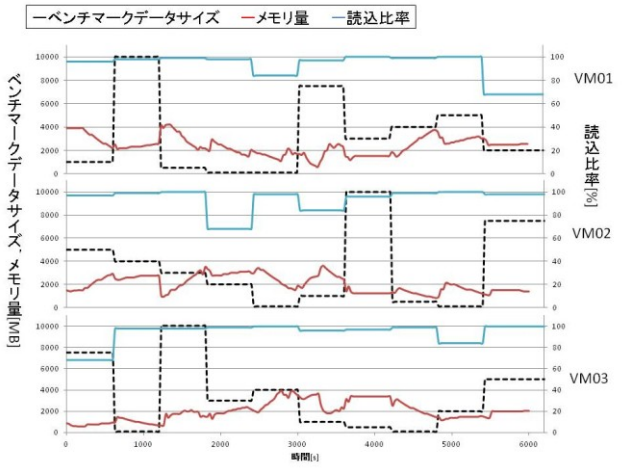

図18　ベンチマークデータサイズとVMのメモリ量の関係
(α=20)

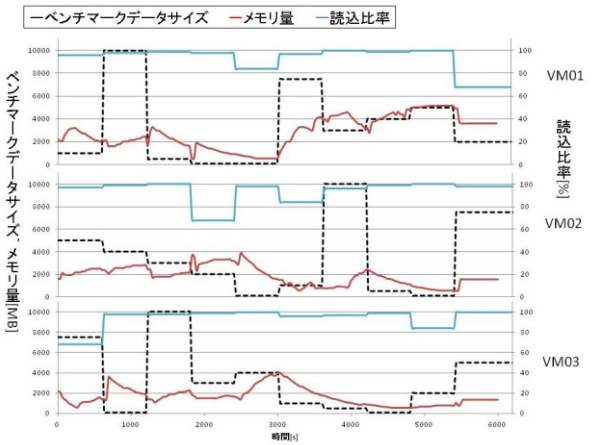

図19　ベンチマークデータサイズとVMのメモリ量の関係
(α=25)

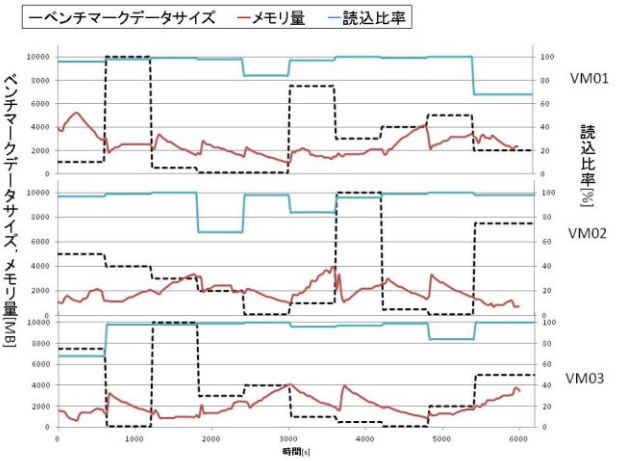

図20　ベンチマークデータサイズとVMのメモリ量の関係
(α=30)

8. 関連研究

VM の割り当てメモリ量の最適化の研究として，以下のものがある．

Zhao らは，LRU ヒストグラムを取得してキャッシュヒット率を予測し ballooning を行う手法を提案している[3]．しかし，仮想記憶や PTE に注目した手法であり，本稿の様な I/O 性能に関して寄与のある手法とはなっていない．

仮想化環境の I/O に着目し，VM のメモリ割り当てについて考察した研究して，Jones らの提案[4]がある．当該手法ではバッファキャッシュへの追加と削除から必要メモリ量の推定を行っている．しかし，キャッシュヒット率についてのみ考察されており，最終的な性能については議論されていない．また，キャッシュヒット率の推定は必ずしも正確ではなく，実際のヒット率を求めて使用している本稿の提案手法と比較し正確さにおいて劣っている部分がある．

キャッシュヒット率を実測しVMのメモリ割り当て量を

調整するとして，文献[1]の手法がある．しかし，当該研究は読込の処理のみを考慮しており，読込み書込み混在処理を考慮していない．よって，読み書きが混在する一般的な環境では十分性能を提供することができない．

9. おわりに

本項では，読込み書込み比率とキャッシュヒット率を考慮した VM へのメモリ割当手法を提案し，評価によりその有効性を示した．

今後は，データベースなど応用を用いての評価を行っていく予定である．

謝辞 本研究は JSPS 科研費 24300034, 25280022, 26730040 の助成を受けたものである。

参考文献

1) 坂本 雅哉,山口 実靖, "仮想化環境におけるキャッシュヒット率を考慮した VM メモリ割り当て", 第 12 回情報科学技術フォーラム FIT 2013 RC-009,2013

2) Stephen Spector, “Memory Overcommit”, August 27, 2008, http://blog.xen.org/index.php/2008/08/27/xen-33-feature-memory-overcommit/

3) Weiming Zhao, Zhenlin Wang, “Dynamic memory balancing for virtual machines,” Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pp, 21-30. 2009.

4) Stephen T, Jones Andrea C, Arpaci-Dusseau Remzi H, Arpaci-Dusseau, “Geiger: monitoring the buffer cache in a virtual machine environment,” Proceedings of the 12th international conference on Architectural support for programming languages and operating systems, pp, 14-24. 2006.