Vol.2011-DBS-153 No.28
2011/11/3

仮想化環境における VM メモリ割当量とキャッシュ利用に関する一考察

渡邉有貴† 山口実靖†

クラウドコンピューティング環境をはじめ多くの環境にて仮想化技術が用いられている．仮想化環境ではゲスト OS とホスト OS の二つの OS が動作しており，それぞれがメモリキャッシュ機能を提供している．よって，I/O 性能の向上を目指すには，両 OS のキャッシュを統合的に最適化することが好ましい．しかし，ユーザにはキャッシュの動作を確認することができず，両キャッシュのどちらが性能向上に寄与しているかを判断することは容易でない．これにより，性能に関する考察や性能向上の実現が困難となっている．
本稿では，ゲスト OS キャッシュとホスト OS キャッシュを統合的に解析する手法を提案する．そして，提案手法を実装し，それをベンチマークの実行に対して適用しその有効性を示す．

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等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 を一台の物理計算機に集約する手法がある[1]．この手法はクラウドコンピューティングなどで採用されており，仮想化環境は非常に重要なプラットフォームとなっている．しかし，仮想化環境における I/O 処理の動作の把握は容易ではなく，結果として I/O 性能の向上が困難となっている．I/O 処理の動作の把握が困難である理由の一つに，仮想化環境ではホスト OS とゲスト OS の二つの OS が動作しており，それぞれがメモリキャッシュを保持していることが上げられる．すなわち，ユーザからはシステム内部で動作するキャッシュ群のそれぞれの動作を把握することができず，どのキャッシュが I/O 性能向上に寄与しているのかを理解することが困難となっている．さらに，仮想計算機とホスト計算機は物理的な計算資源を共有しており，仮想計算機に多くの資源を割り当てると物理計算機が使用可能である計算資源が減少することとなる．よって，両キャッシュの効果を統合的に考察することが必要となる．本稿では，ホスト OS とゲスト OS における I/O 要求の処理を統合的に観察できる手法を提案し，それによる解析結果をについて考察する．

† 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

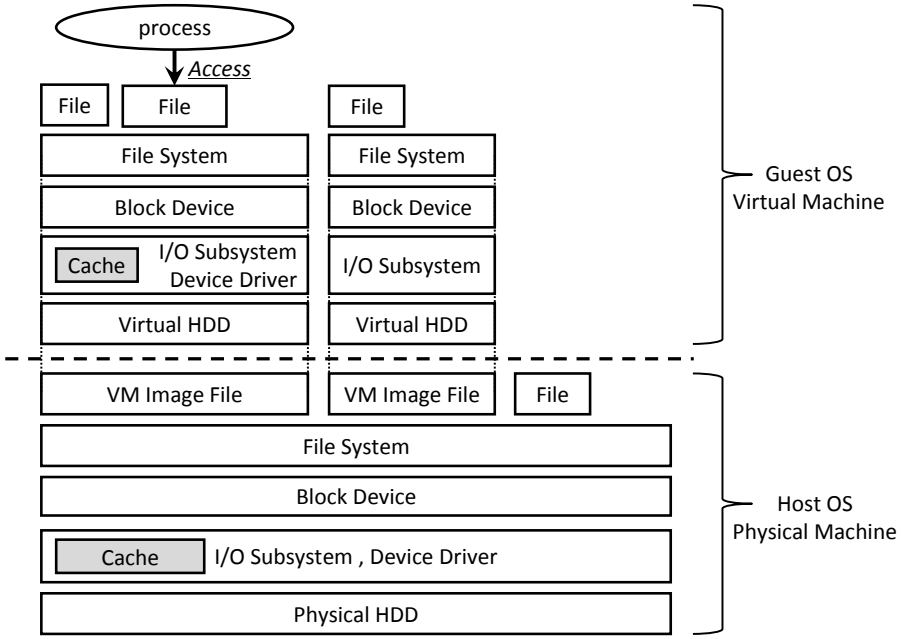

図 1 仮想化環境の構成図

2. 仮想計算機における I/O 処理

図 1 にイメージファイル手法を用いた仮想計算機環境の構成と，I/O 要求処理の手順を示す．図のように仮想化環境は仮想計算機-ゲスト OS の層と，物理計算機-ホスト OS の層の二種類の層に分けられ，同時に二種類の OS が動作している．仮想計算機内のプロセスから発行された I/O 要求は，次に述べるように非常に多くの層を経由して処理される．まず，仮想計算機内のプロセスからファイルアクセス要求が発行されると，I/O 要求がゲスト OS のファイルシステムに到着する．そして，ゲスト OS ファイルシステムがファイルレベルアクセスをブロックレベルアクセスに変換し，ブロックレベル要求をゲスト OS のブロックデバイスに対して発行する．ブロックデバイスは要求を I/O サブシステムおよびデバイスドライバに転送し，この要求はさらに仮想ハードディスクに転送される．仮想ハードディスクはホスト OS のイメージファイルで

あるため，ゲスト OS における仮想ハードディスクへのアクセスが，ホスト OS における仮想計算機プロセスによるイメージファイルへのアクセスを発生させ, ホスト OS ファイルシステムにファイルレベルアクセス要求が送られる．そして，ホスト OS ファイルシステムがファイルレベルアクセス要求をブロックレベルアクセス要求に変換し，ホスト OS ブロックデバイスにブロックレベル要求が発行する．ブロックデバイスは要求をホスト OS の I/O サブシステムおよびデバイスドライバに転送し，要求が物理ハードディスクに到着する．これらの全ての層が性能に影響を与える可能性があるため, I/O 性能について考察するにはこれらを広範囲に解析することが重要となる．

加えて，仮想化環境ではゲスト OS のブロックデバイスとホスト OS のブロックデバイスの両方にメモリキャッシュが存在しており，通常はどちらのキャッシュが I/O 性能の向上に寄与しているかを確認することができない．このこともシステム動作の把握を困難としている．また，仮想計算機に割り当てられるメモリは，ホスト計算機のメモリの一部を割いて用意される．よって，仮想計算機に多くのメモリを与えると仮想計算機が多くのキャッシュメモリを使用可能となり，仮想計算機におけるキャッシュヒット率は向上するが，ホスト計算機におけるキャッシュヒット率は低下すると考えられる．一方，仮想計算機に少ないメモリを与えるとホスト計算機が多くのキャッシュメモリを使用可能となり，ホスト計算機におけるキャッシュヒット率は向上するが，仮想計算機におけるキャッシュヒット率は低下すると考えられる．これらのことから，高い I/O 性能を得るには両キャッシュの効果を正確に把握し割当メモリ量を適切に調節する必要があると考えられる．

3. 仮想計算機割当メモリ量と I/O 性能の関係の調査

適切な仮想計算機（VM）へのメモリ割当量について考察するために，VM メモリ割当量および I/O 処理方法（bufferd I/O, direct I/O）を変更して VM の I/O 性能の測定を行った．実験に用いた物理計算機の仕様を表 1 に，仮想計算機の仕様を表 2 に示す．

性能評価はアプリケーションベンチマークとマイクロベンチマークを使用して行った．アプリケーションベンチマークでは Postmark と FFSB（Flexible File System Benchmark）[2]を使用し，マイクロベンチマークでは Linux コマンドの dd を用いてシーケンシャルリードを行い性能評価を行った. Postmark ベンチマークの設定は表 3 に，FFSB ベンチマークの設定は表 4 に示す. Linux カーネルは, Xen が対応している Linux 2.6.18.8 に統一して実験を行った.

3.1 ランダムアクセスベンチマーク（Postmark）

I/O ベンチマークソフト Postmark を用いて，VM の I/O 性能の測定を行った．1 台の物理計算機上に 6 台の VM を起動し，全ての VM で Postmark を実行した．VM へのメモリ割当量は 128MB から 1GB まで変更し, ベンチマークのデータサイズは 128MB か

表 1　物理計算機の仕様

Host OS	CentOS 5.4 x86_64
Host Kernel	Linux 2.6.18.8
Xen Version	3.3.1
CPU	Athlon 64 X2 2.7[GHz]
CPU Core	2
Memory	8[GB]
HDD	500[GB]，7200[rpm]

表 2　仮想計算機の仕様

Guest OS	CentOS 5.4 x86_64
Guest Kernel	Linux 2.6.18.8
Virtual CPU Core	2
Virtual Memory	測定によって変動
Virtual HDD	20[GB]

表 3　ベンチマークソフト(Postmark)の設定

試行回数	5000[回]
ファイル数	測定によって変動
ファイルサイズ	1[MB]
1 オペレーションあたりの読込量	1[MB]
1 オペレーションあたりの書込量	1[MB]
読込/書込比率	5:5

表 4　ベンチマークソフト(FFSB)の設定

測定時間	1800[秒]
ファイル数	測定によって変動
ファイルサイズ	64[KB]
1 オペレーションあたりの読込量	16[KB]
1 オペレーションあたりの書込量	16[KB]
読込/書込比率	5:5
スレッド数	1

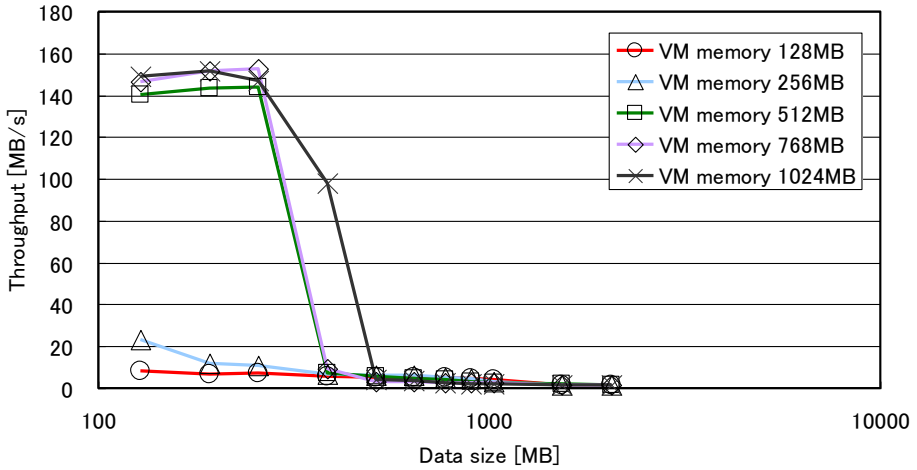

図 2　Postmark 測定結果 (全体図)

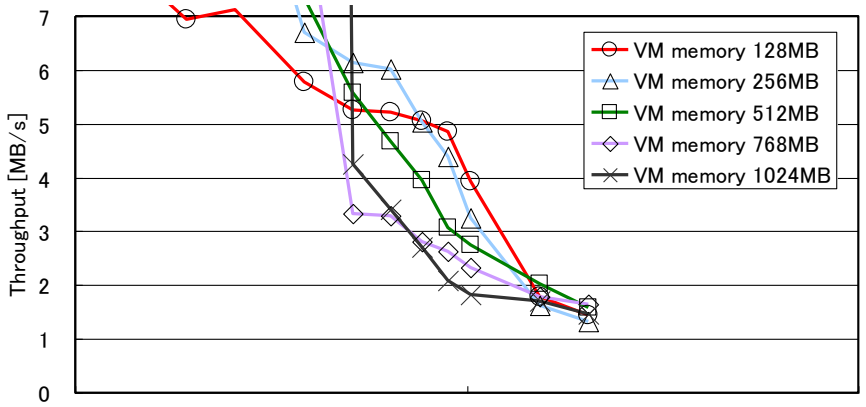

図 3　Postmark 測定結果 (拡大図)

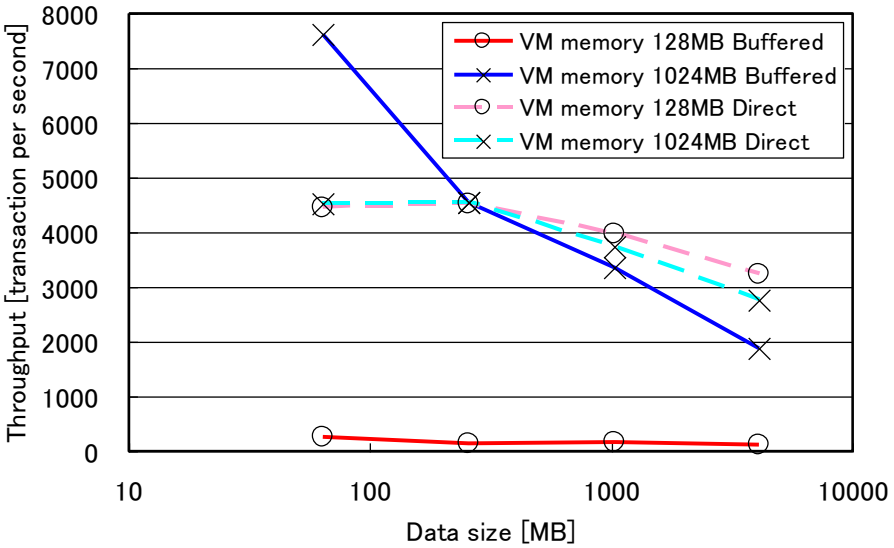

図 4　FFSB 測定結果

ら 2GB まで変更して測定を行った．測定結果を図 2，図 3 に示す．

図 2 より，ファイル総容量が小さく VM キャッシュヒット率が高い場合は VM に与えるメモリ量は多い方が良い性能が得られることが確認できた．また図 3 より，ファイル総容量が大きく VM キャッシュヒット率が低い場合は VM に与えるメモリ量は少ない方が良い性能が得られることが確認できた．

3.2　ランダムアクセスベンチマーク（FFSB）

前節と同じ計算機環境で I/O ベンチマークソフト FFSB を用いて，VM の I/O 性能の測定を行った．1 台の物理計算機上に 6 台の VM を起動し，全ての VM で FFSB を実行した．VM へのメモリ割当量は 128MB と 1GB に変更し，ベンチマークのデータサイズは 128MB から 4GB まで変更して測定を行った．また同様の測定を DIRECT I/O を有効にし，VM キャッシュを無効化して行った．測定結果を図 4 に示す．

図 4 より，ファイル総容量が小さく VM キャッシュヒット率が高い場合は，Postmark と同じく VM に与えるメモリ量は多い方が良い性能が得られることが確認できた．また，ファイル総容量が大きく VM キャッシュヒット率が低い場合は，VM に与えるメモリ量を少なくし，さらに DIRECT I/O を用いて VM キャッシュを無効化した方が良い性能が得られることが確認できた．

3.3 シーケンシャルアクセスベンチマーク

前節までと同じ計算機環境で Linux コマンドの dd を用いてシーケンシャルリードを実行し，VM の I/O 性能の測定を行った．1 台の物理計算機上に 6 台の VM を起動し，全ての VM で dd を実行した．VM へのメモリ割当量は 128MB から 1GB まで変更し，dd によるシーケンシャルアクセスのデータサイズは 128MB から 1GB まで変更して測定を行った．測定結果を図 5 に示す．

図 5 より，前節と同じくファイル総容量が小さく VM キャッシュヒット率が高い場合は，Postmark と同じく VM に与えるメモリ量は多い方が良い性能が得られることが確認できた．また，ファイル総容量が大きく VM キャッシュヒット率が低い場合は，VM に与えるメモリ量を少なくし，さらに DIRECT I/O を用いて VM キャッシュを無効化した方が良い性能が得られることが確認できた．

4. ホスト OS とゲスト OS の統合解析

本章において，仮想計算機に与えたメモリ量と I/O 性能の関係について，仮想計算機のメモリキャッシュと物理計算機のメモリキャッシュの効果とともに考察する．

4.1 解析手法

I/O 要求をアプリケーション層，仮想計算機仮想ハードディスク層，物理計算機物理計算機層で観察し，各層間で I/O バイト量や I/O 命令数を比較することにより仮想化環境における I/O 処理の観察と解析を行う．アプリケーション層と仮想ハードディスク層の間にはゲスト OS のメモリキャッシュが存在しており，両層の I/O 量を比較することによりゲスト OS のメモリキャッシュの効果を確認することが可能となる．同様に，仮想ハードディスク層と物理ハードディスク層の間にはホスト OS のメモリキャッシュが存在しており，両層を比較することによりホスト OS のキャッシュの効果を確認することができる．

実装は前章の表 1, 表 2 に示した環境を用いて行い, オープンソースである Linux と Xen のソースコードにモニタリング機能を追加することにより行った．モニタリング機能は，まずカーネル空間内に I/O 履歴保持用のメモリを確保する．そして，各層にて I/O 要求の処理が行われるたびにその時刻，I/O の種類(read/write)，ブロックアドレスあるいはファイル名とオフセット，アクセスサイズを確保メモリの中に記録する．

仮想計算機の仮想ハードディスク層における I/O 要求処理の観察のためには，Xen の仮想ブロックデバイスドライバ(xvd)における I/O 要求をデキューし実行する実装部にモニタリング機能を追加した．物理ハードディスクへのアクセス要求を観察するためには，ホスト OS における SCSI サブシステムの SCSI 命令の発行部にモニタリング機能を追加した．また本稿の実験では，アプリケーションによる発行 I/O を観察するためにアプリケーションのファイルアクセス要求部にモニタリング機能を追加した

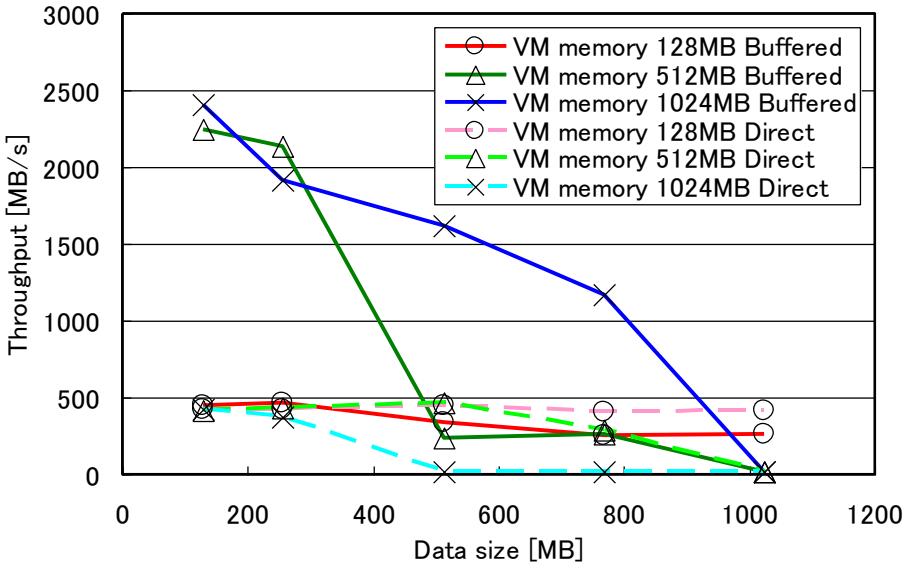

図 5 シーケンシャルアクセス測定結果

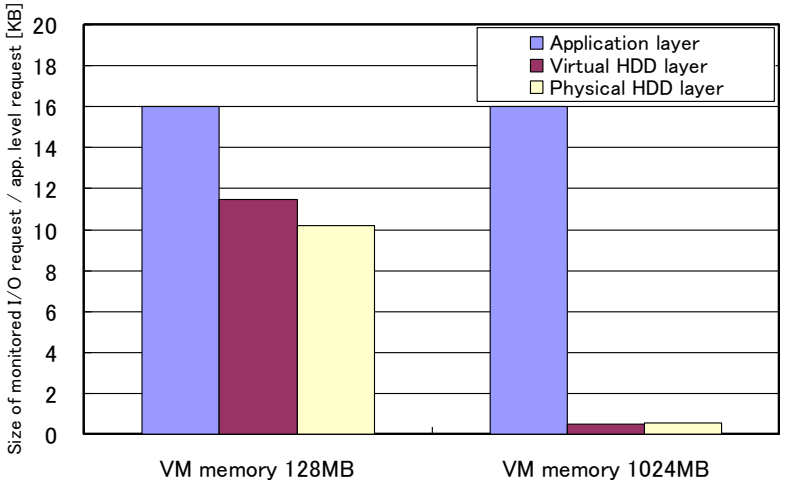

図 6 FFSB データサイズ 64MB 時の各層の I/O 要求量

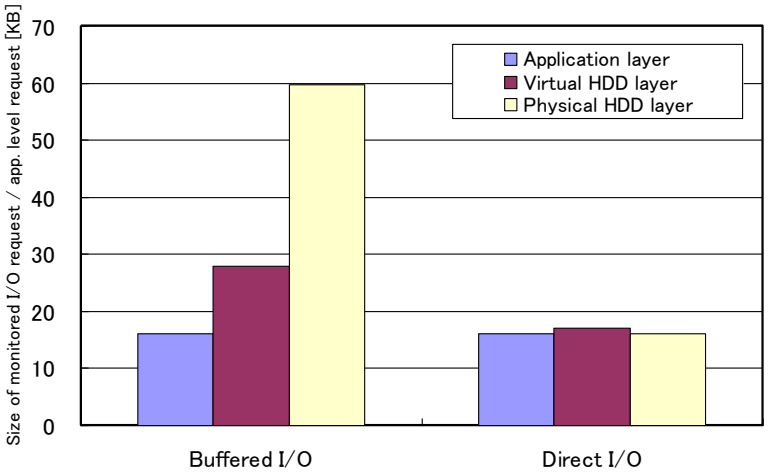

図 7　FFSB データサイズ 4096MB
キャッシュ処理方法変更時の各層の I/O 要求量

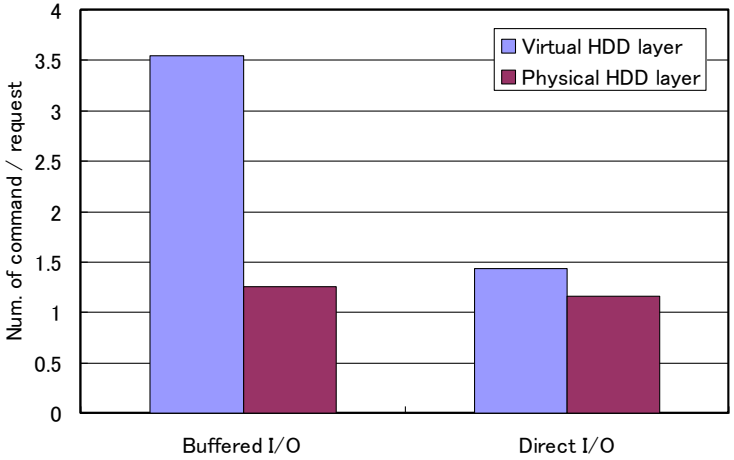

図 8　ベンチマークの 1 オペレーション当たりの
各 HDD への平均アクセス回数

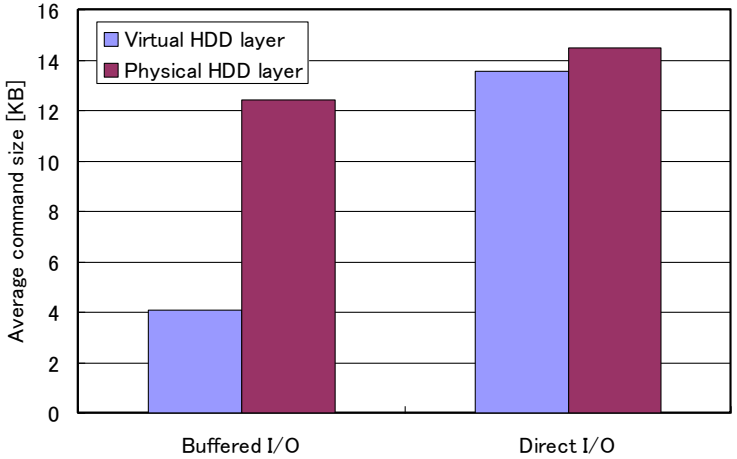

図 9　各 HDD への 1 入出力命令当たりの平均 I/O 処理サイズ

アプリケーション層における I/O 処理を正確に観察するためにはアプリケーション層において観察することが最も優れているが，アプリケーションに対する改変が行えない場合も多い．そのような場合はゲスト OS のファイルシステム層のファイルアクセスの処理部にて観察することにより，アプリケーションの発行 I/O 要求をほぼ正確に観察することが可能となる．

4.2　解析結果

前節にて解説した統合解析システムを用いて仮想化環境における I/O の統合的動作解析を行った．まず，データサイズ 64MB で FFSB を行った際の各層における 1 オペレーション当りの I/O 量を図 6 に示す．この解析結果から，データが VM キャッシュに格納可能な場合，VM に多くのメモリを与えることで VM のキャッシュが効果的に機能し，仮想 HDD 層における I/O 量が大幅に減少して性能が向上していることが分かる．

次に，データサイズ 4096MB で VM 搭載メモリ 128MB の時に，キャッシュ処理方法を変更して FFSB を行った際の，各層における 1 アプリケーションレベルオペレーション当りの I/O 量を図 7 に示す．この解析結果から，データが VM キャッシュに格納不可能な場合，キャッシュを有効にしてアクセスを行うと各層を経由することによる I/O 量が増加することが確認できる．また，DIRECT I/O を用いて VM キャッシュを無効化することによって I/O 量の増加が抑えられていることが確認できる．

さらにデータサイズ 4096MB で VM 搭載メモリ 128MB の時にキャッシュ処理方法を変更して FFSB を行った際の，1 アプリケーションレベルオペレーション当りの仮想 HDD および物理 HDD への発行命令数(HDD レベル命令数)と，HDD レベル命令の平均サイズについて解析する．図 8 に 1 アプリケーションレベルオペレーション当りの各 HDD への発行命令数を，図 9 に HDD レベル命令の平均 I/O サイズを示す．これらの結果から, VM キャッシュが有効になっている場合はベンチマークが発行した I/O 要求が各層で小さなサイズに分割されて両 HDD 層に複数個到着していることが確認できる．また，DIRECT I/O を用いて VM キャッシュを無効化すると，両 HDD に到着する I/O のサイズが大きくなり，少数の大きな I/O により処理が行なわれていることが確認できる．これらの結果が DIRECT I/O を用いた際の性能向上に繋がっているのではないかと考えられる．

5. 関連研究

仮想化環境における I/O 性能の向上に関する研究として次のものがあげられる．Boutcher らは仮想化環境における I/O スケジューラの性能の評価を行い，ゲスト OS とホスト OS に適した I/O スケジューラの選択手法について示している[3]．Xu らは仮想化環境に適した新しい I/O スケジューラを提案し, その性能の評価を行っている[4]．Kesavan らは，仮想ハードディスクの振る舞いと物理ハードディスクの振る舞いには大きな違いがあり，仮想ハードディスクの応答性能は同一物理計算機内の他の仮想計算機の振る舞いに大きな影響を受けることを指摘している[5]．これらは，仮想化環境における I/O 性能の向上手法として有益なものであるが，二重キャッシュ環境のキャッシュの性能，キャッシュの動作について考察したものではなく，本稿とは貢献の内容が異なっている．

また，仮想化環境を想定した研究ではないが，文献[6]において iSCSI 環境を想定したサーバコンピュータとストレージ機器の統合的解析手法が提案されている．統合的な解析を目指す点において類似性はあるが，本研究とは研究目標が大きく異なっており，貢献の内容も大きく異なっている．

6. おわりに

本稿では，二重キャッシュ環境である仮想化環境に着目し，二重キャッシュの解析手法を提案した．そして，提案手法を Linux および Xen を用いて実装し，その有効性の検証を行った．検証の結果，提案手法および提案実装により両キャッシュの効果を観察と考察することが可能となることが確認され，有効性が確認された．

今後は，提案手法を用いて仮想化環境における I/O 性能の向上を実現していく予定である．

謝辞 本研究は科研費（22700039）の助成を受けたものである．

参考文献

1) 越智 俊介，山口 実靖，淺谷 耕一，“仮想計算機 KVM によるサーバ統合におけるサーバ性能の向上”，電子情報通信学会データ工学ワークショップ論文誌 DEWS2008 D5-4
2) FFSB，http://sourceforge.net/projects/ffsb/
3) D. Boutcher and A. Chandra,”Does Virtualization Make Disk Scheduling Passe?”，SOSP Workshop on Hot Topics in Storage and File System(Host Storage '09)
4) Y. Xu and S. Jiang,”A Scheduling Framework that Makes any Disk Schedulers Non-work-conserving solely based on Request Characteristics”，FAST2011
5) M. Kesavan，A. Gavilovska and K. Schwa,”On Disk I/O Scheduling in Virtual Machines”，WIOV'10，May 2010
6) Saneyasu Yamaguchi，Masato Oguchi，Masaru Kitsuregawa,”Trace System of iSCSI Storage Access”，SAINT 2005