

B-053

ホスト OS とゲスト OS の統合的動作解析による VM メモリ割当量に関する考察 A Study on Virtual Machine Memory Allocation based on Behavior of Guest OS and Host OS

渡邊 有貴[†] 山口 実靖[†]
 Yuki Watanabe Saneyasu Yamaguchi

1. はじめに

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]。

VM (仮想計算機) 上のアプリケーションから発行された I/O 要求はゲスト OS とホスト OS の様々なレイヤを経由してホスト計算機の物理ディスクへ到着する。これに対して我々は各レイヤにおいて個別に I/O 処理の様子を観察し、アプリケーションの要求量より多くの I/O が発生することや、VM で発行された I/O が各レイヤを通過する際に分割されることがあり、性能が劣化してしまうことを確認した[2]。

本稿では、ホスト OS とゲスト OS を統合的に動作解析できる環境を構築し、両 OS の I/O 処理について考察する。

2. 仮想計算機に与えるメモリ量

仮想化環境では、ホスト計算機のメモリの一部が VM に割り与えられる。VM に多くのメモリを与えると VM が多くのキャッシュメモリを使用可能となり VM におけるディスク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。このことから高い I/O 性能を得るには割当メモリ量を適切に調整する必要がある。

3. 基本性能評価

適切な VM へのメモリ割当量、および I/O 処理方法について考察するために、VM のメモリ割当量、および I/O 処理方法(キャッシュ有効/無効)を変更して VM の I/O 性能の測定を行った。

3.1 FFSB

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。測定結果を図 1 の”非 DIRECT”の線にて示す。図内の”DIRECT”の線については次節にて後述する。測定結果より、FFSB の測定においては VM に与えるメモリ量が多い方が良い性能が得られることが確認できた。

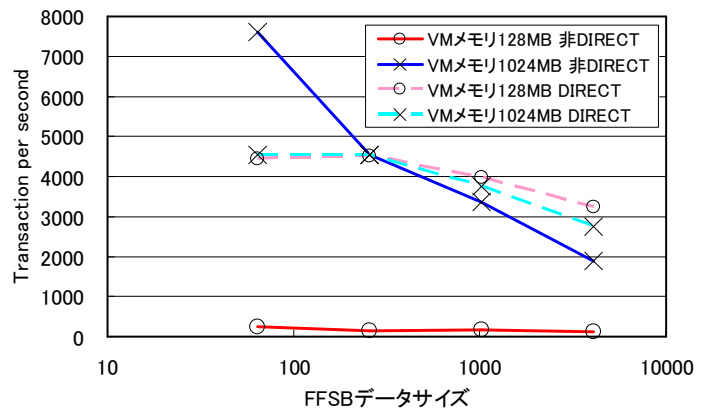


図 1 FFSB の性能

3.2 FFSB (DIRECT I/O)

前節と同様の測定を DIRECT I/O を用いて行った。測定結果を図 1 の”DIRECT”の線にて示す。ファイル総容量が大きくキャッシュヒット率が低い場合は DIRECT I/O を有効にして(キャッシュを無効化して)I/O 処理する方が良い性能となった。

4. 各層における非統合的動作解析

前章の測定結果、VM キャッシュとホスト計算機キャッシュの動作の関連性、DIRECT I/O の影響を調査するために、各層における I/O 量とその内訳を調査した。具体的には、(1)VM 上でアプリケーションが発行した I/O 要求の量、(2)VM の仮想ディスクに到着した I/O 要求の量、(3)ホスト計算機の物理ディスクに到着した I/O 要求の量を比較した。(1)と(2)の間には VM のディスクキャッシュが存在し、(2)と(3)の間にはホスト計算機のディスクキャッシュが存在する。

データサイズが 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 2 に示す。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と VM のキャッシュが効果的に機能し、VM ディスク層における I/O 量が大幅に減少し性能が向上していることが分かる。

データサイズが 4096MB の時に I/O 処理方法を変更して 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 3 に示す。データが VM キャッシュに格納不可能な場合、各レイヤを経由することににより I/O 量の増加が見られるが、DIRECT I/O を有効化しキャッシュを無効化することによってその増加を抑えることが可能となっていることが分かる。

[†] 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

5. 各層における統合的動作解析システムの構築

本章にて、アプリケーション層、ゲスト OS 層、ホスト OS 層のモニタリング結果を統合的に解析する手法を提案する。

アプリケーション層にて発行された I/O 要求の種類 (read/write) とアクセスファイル名とオフセットと時刻を記録し、ゲスト OS の仮想 HDD 層にて処理された I/O の種類とアドレス (仮想 HDD 空間におけるアドレス) と時刻を記録し、ホスト OS の SCSI 層にて処理された I/O 命令の種類とブロックアドレス (実 HDD 空間におけるアドレス) と時刻を記録する。

次に各層にてモニタリングされた結果の関連付け方法を述べる。まず、ゲスト OS のファイルシステム (本実験では ext3) のスーパーブロック内のメタ情報を用いて、「ゲスト OS においてモニタしたブロックアドレス」から「アクセスされた VM 内のファイルの inode 情報とファイル内オフセット」を特定する。これによりアプリケーション層のモニタリング結果とゲスト OS 層のモニタリング結果を関連付けることができる。

そして、ホスト OS のファイルシステム (ext3) のメタデータ情報を用いて、「モニタされた SCSI アドレス」から「アクセスされたホスト OS 上のファイルの inode 情報とファイル内オフセット」を特定する。ここで仮想 HDD イメージのファイル内オフセットがゲスト OS におけるブロックアドレスとなる。これにより、ゲスト OS 層のモニタリング結果とホスト OS 層におけるモニタリング結果を関連付けることができる。

以上より、アプリケーションが発行したファイルアクセス要求と、ゲスト OS にて処理されたブロック I/O とホスト OS にて処理されたブロック I/O の同一性を特定することが可能となり、3 層を統合的に解析できるようになる。

本手法を用いて、図 1 の FFSB データサイズ 4GB、VM メモリ 128MB、DIRECT I/O 有効と無効の実験の write 処理を解析したものを図 5 と図 6 に示す。図内の横軸はワークロード中に処理した順番を表しており、時間とは比例していない。

図より、VM 内のアプリケーションより発行された I/O 要求を実 HDD までトレースすることが可能であり、アプリケーションが発行したのとは異なる順で実 HDD にて処理されていることなどを把握することが可能であることが確認された。

6. まとめ

本稿では、VM 内のアプリケーション、ゲスト OS、ホスト OS の動作を統合的に解析する手法を提案し、その解析例を示した。

今後は本解析手法を用いてワークロードの動作解析を行い、性能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, 浅谷 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, 2008.

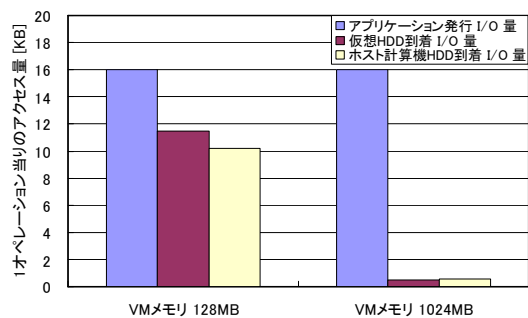


図 2 FFSB データサイズ 64MB 時の各 I/O 要求量

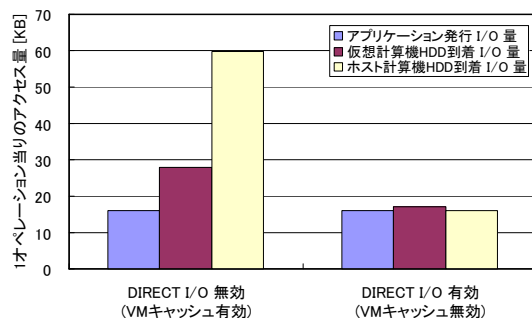


図 3 FFSB データサイズ 4096MB
I/O 処理方法変更時の各 I/O 要求量

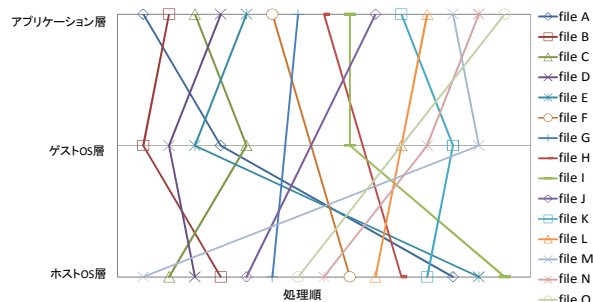


図 4 FFSB データサイズ 4096MB VM メモリ 128MB
非 DIRECT I/O 時の各レイヤ I/O 処理解析

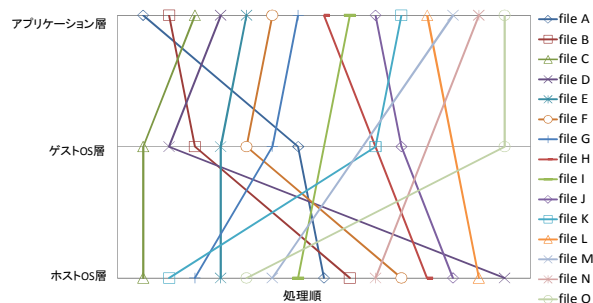


図 5 FFSB データサイズ 4096MB VM メモリ 128MB
DIRECT I/O 時の各レイヤ I/O 処理解析

[2]渡邊 有貴, 山口 実靖, “I/O 処理の動作を考慮した仮想計算機メモリ割当量に関する考察”, 第 9 回情報科学技術フォーラム, 2010.