

ワーキングセットとキャッシュヒット率を考慮した 動的 VM メモリ割り当て

VM Memory Allocation based on Cache Hit Ratio and Estimated Working Set Size

坂本 雅哉†
Masaki Sakamoto

山口 実靖†
Saneyasu Yamaguchi

1. はじめに

データセンタ等におけるサーバの消費電力の増加や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。

仮想化環境では、仮想マシン(VM)を停止させることなく割り当てメモリ量を変更することが可能である。一つの物理マシン上に複数の VM が稼働しており、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。しかし、xenballoon がメモリ割り当てを行うために考慮する情報は、VM 内のプロセスが使用するメモリ量のみであり、ページキャッシュとして利用されるメモリの量を考慮していない。よって xenballoon では適切に I/O 性能向上を行えないと考えられる。

我々は、読み込みのみを行う応用を想定し、動的に VM メモリ割り当て量を制御する手法の提案を行った。そして、読み込みのみを行う FFSB や DBMS (TPC-H) などの応用を用いて評価を行い、キャッシュヒット率 100% を目標とする手法の有効性を確認した[1][2]。

本稿では、読み込みと書き込みの両方を行う応用を想定し、Xen を用いた仮想化環境における、メモリ割り当て量と、キャッシュヒット率と、読み書きの両方を行う DBMS (TPC-C) の性能の関係性について調査を行う。これにより読み込みと書き込みの両方を行う応用においてはキャッシュヒット率 100% を目標とすることが必ずしも適切でないことを示す。そして、キャッシュヒット率を考慮した動的 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量を動的に変更する機能である。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(Committed_AS)を監視してホスト OS への要求メモリ量を調整する[3]。

3. キャッシュヒット率と DBMS 性能の関係

本章で、キャッシュヒット率と DBMS 性能の関係を示す。

キャッシュヒット率と DBMS 性能の関係を調査するため、1 台の VM 上で TPC-C を実行し、DBMS 性能とメモリ量とキャッシュヒット率を測定した。本実験では TPC-

C の SF(scale factor)は 11 とし、データサイズは約 1[GB]にて評価を行った。また本実験の計測時間は 2000 秒とした。実験に用いた物理マシンとホスト OS の仕様は CPU Intel Celeron G530 2.40GHz, OS CentOS 6.3, カーネル Linux 2.6.32.57, 仮想化システム Xen 4.1.2, メモリ 8[GB], HDD 1[TB]である。仮想マシンとゲスト OS の仕様はカーネル Linux 2.6.18.8, メモリサイズは動的に変更し、HDD 100[GB]である。キャッシュヒット率は、Linux カーネルおよび Xen のブロックデバイス実装 blkback.c と sd.c を改変する手法[1]により求めた。

測定結果を図 1 に示す。図より、キャッシュヒット率が 0-80[%]の間はメモリ割り当て量を増やすことによる DBMS 性能の向上の効果が大きいことが分かる。一方キャッシュヒット率が 80-100[%]の間はメモリ割り当てによる DBMS 性能向上の効果が小さいことが分かる。以上よりキャッシュヒット率と DBMS 性能の関係に着目すると、キャッシュヒット率が 0-80[%]の VM のメモリ量を増加させることにより I/O 性能を効果的に向上可能であると考えられる。

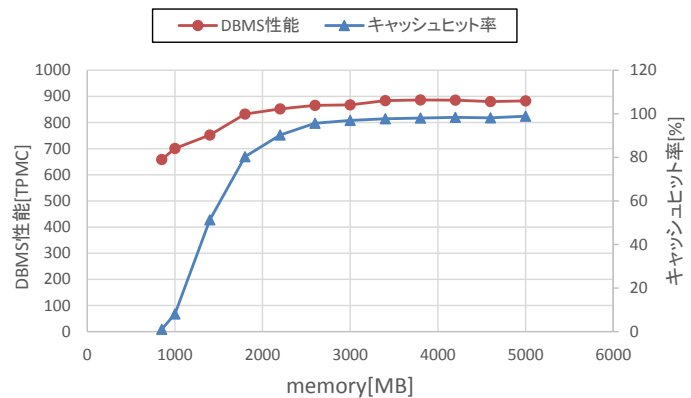


図 1 DBMS 性能とキャッシュヒット率とメモリ量の関係

4. 提案手法

本章で、ゲスト OS のページ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手法を提案する。応用としては、読み込みと書き込みの両方を行うものを想定する。

本手法では、各 VM のキャッシュヒット率の値をホスト OS にて集計し、以下の手順により VM メモリ割当量を決定する。

1. 全 VM のメモリ割り当て量を 10% 減少させ、これを再配分用メモリとする。
2. 上記 1 にて得た再配分用メモリをキャッシュヒット率が 0-9[%]の VM には各 VM のキャッシュヒ

†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

ット率により比例配分し、10-80[%]のVMには一律キャッシュヒット率10[%]とみなして比例配分し、81-100[%]のVMには配分を行わない。

例えば、VMが3台あり、それぞれのキャッシュヒット率が7%、30%、85%であった場合は、再配分用メモリは7:10:0の割合で再配分される。

提案手法は、ゲストOS上で動作するxenballoonに変更を加えることにより実装できる。キャッシュヒット率の取得は3章の手法により実現する。キャッシュヒット率の測定機能、メモリ量とキャッシュヒット率の情報をホストOSに伝える機能、メモリ割り当て量の情報をホストOSから取得する機能を追加で実装する。

上記の動作は転送インターバル i_{trans} 秒ごとに行われる。またホストOS上では、各VMのメモリ割り当て量とキャッシュヒット率を取得、提案手法に従った各VMのメモリ割り当て量の決定、メモリ割り当て量の変更が行われるが、これらは割り当てインターバル i_{alloc} 秒毎に実行される。

5. 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。

5.1 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 (VM1, VM2, VM3) を立ち上げ、全VM上でTPC-Cを並列実行しDBMS性能を測定した。VM1にはSFが33でサイズ約3[GB]、VM2にはSFが22でサイズが約2[GB]、VM3にはSFが11でサイズが約1[GB]にてTPC-Cを実行し、DBMS性能、各VMのメモリ割り当て量とキャッシュヒット率の推移を観察し評価した。また本実験の計測時間は3000秒とした。実験に用いた物理マシンと仮想マシンの仕様は3章と同様である。 i_{trans} は30[秒]、 i_{alloc} は5秒とした。

5.2 DBMS性能の測定

図2に提案手法と静的メモリ割り当て手法とxenballoon手法の全VMのTPC-C性能の平均を示す。静的メモリ割り当て手法とは全VMのメモリを1685[MB](均等割り当て)で固定したものである。図より静的メモリ割り当て手法やxenballoon手法より提案手法の方が性能が高いことがわかる。

図3にVMのメモリ割り当て量の推移を示す。図よりメモリ割り当ての効果が低いVM1のメモリを減らし、メモリ割り当ての効果が高いVM2とVM3に積極的にメモリを割り当てていることが分かる。

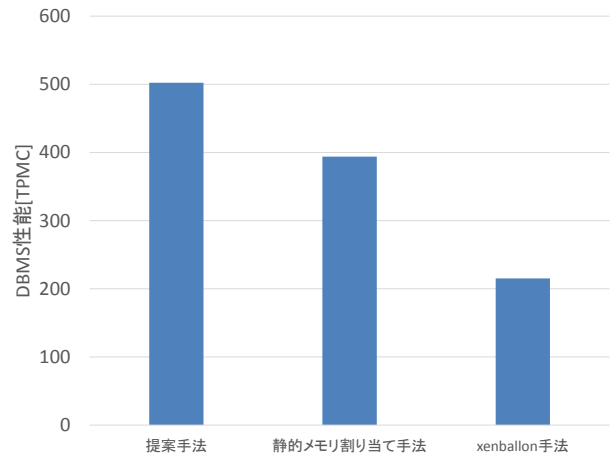


図2 性能評価

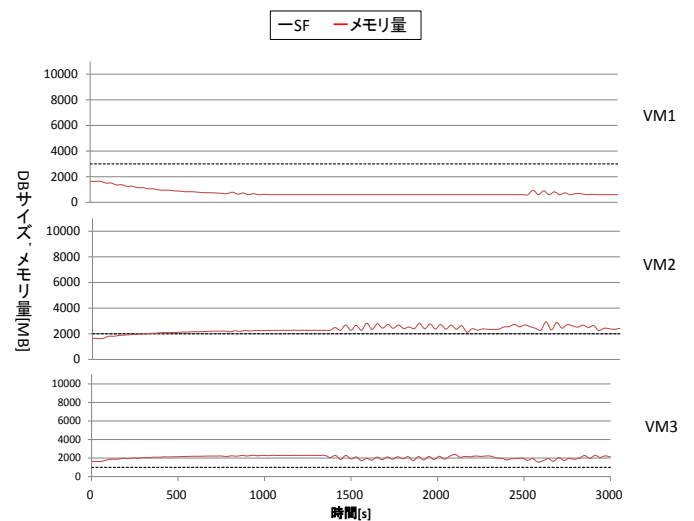


図3 データベースサイズとVMのメモリ量の推移

6 おわりに

本稿では、読み込みと書き込みの両方を行う応用を想定したキャッシュヒット率を考慮したVMメモリ割り当て手法を提案し、TPC-Cによる評価を行った。評価の結果、提案手法の有効性が確認された。

今後は、閾値の自動決定手法の検討、さまざまなDBMSが混在した環境で評価を行う予定である。

謝辞 本研究はJSPS科研費25280022, 26730040, 15H02696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 (FIT2013), RC-009
- [2]坂本雅哉, 山口実靖, "複数アプリケーション動作環境におけるキャッシュヒット率を考慮した動的VMメモリ割り当て", 情報処理学会第77回全国大会, 5U-07
- [3]Stephen Spector, "Memory Overcommit", August 27, 2008, <http://blog.xen.org/index.php/2008/08/27/xen-33-feature-memory-overcommit/>