

Linux 仮想化環境における VM システムメモリ使用量推測と評価

中尾 司ピエール[†] 坂下 善彦[‡][†] 湘南工科大学 工学研究科 電気情報工学専攻[‡] 湘南工科大学 工学部 情報工学科

1 はじめに

1.1 背景

計算機資源をインターネット経由でサービスとして提供する形態の一つとして Infrastructure as a Service (IaaS) がある。IaaS によるサービス提供形態の一つとして仮想マシンを貸し出す方式が存在する。仮想マシンとはコンピュータの動作やソフトウェアの動作をエミュレートすることによって実現される計算機の一つである。仮想マシンを実現するためにはハイパーバイザと呼ばれる機能が必要となる。ハイパーバイザは CPU 資源、メモリ資源、ハードディスク資源、その他入出力装置などを再現することによりその上で OS を動かすことを可能とするものである。結果 IaaS において計算機資源として提供される。以上を踏まえ我々は計算機資源を仮想マシン単位で提供する側の視点に注目することとした。計算機資源を仮想マシン単位で提供する上で重要となる点としてメモリ資源の工夫であると考えられる。このメモリ資源を節約できればより多くの仮想マシンを有限な物理資源上に確保することが可能となる。

1.2 目的

Linux Kernel-based Virtual Machine (Linux KVM) によって構築される VM 環境において、ホスト側 Linux にて Kernel SamePage Merging (KSM) を利用した場合、KSM の効果としてホスト側のシステムメモリ使用量が圧縮することが分かっている [1]。以上のメカニズムを用いて VM のプロセスによるメモリ使用量が低下する量を求める。また、同一 Linux ホスト上で更に VM を追加稼働させる際にホスト側 Linux のシステムメモリ使用量を推定する。本研究は Linux KVM 環境における KSM 効果に基づく資源利用の限界見込み予想とその実態及び性能変化の関係を述べるために、追

VM system memory usage guess and evaluation in Linux virtual environment

Tsukasapierre Nakao[†], Yoshihiko Sakashita[‡]

[†] Graduate school of electric and information science, Shonan institute of technology

[‡] Dept. of information science, Shonan institute of technology

11t2013@sit.shonan-it.ac.jp

加調査をしたものである。

2 Linux Kernel-based Virtual Machine

Linux KVM は Linux カーネル 2.6.20 から標準搭載されている仮想化基盤である。Linux KVM は Linux カーネルモジュールとして実装されている。Intel VT もしくは AMD-V を使ったネイティブ仮想化をサポートしている。ネイティブ仮想化とは OS を修正することなく隔離させた状態で動作させることである。Linux KVM は周辺装置のエミュレーションは行わずそれらはデバイスエミュレータである qemu-kvm によって行われる。Linux KVM の特徴は Linux カーネルの一部となって動作するため Linux カーネルの機能を利用することができる。特にメモリ管理、プロセススケジューリングの機能をそのまま利用しているため実装はコンパクトである。KVM 上で動作する仮想機械はホスト側 OS からは一つのプロセスとして見える。図 1 に Linux KVM による仮想化の構成を示す。

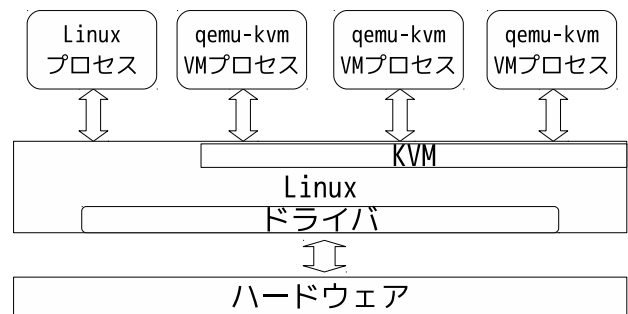


図 1: Linux KVM による仮想化

3 Kernel SamePage Merging

Kernel SamePage Merging (KSM) は Andrea Arcangeli によって提案、実装 [2] されたメカニズムである。Linux カーネル 2.6.32 でマージされた。KSM はユーザプロセスの匿名ページを領域をスキャンし、同一内容のページが存在した場合一つのページへマージするこ

とでメモリ使用量の低減を実現する。匿名ページ領域のスキャンには ksmd というカーネルスレッドが指定感覚に合わせて指定ページ数をスキャンする。スキャンしたページを同定するためには memcmp が用いられている。マージされたページは CoW(Copy on Write) 状態にされる。CoW の動作は Linux カーネルにある機能をそのまま利用している。マージされているページは現在の実装ではスワップアウトされない。マージされているページに更新が発生した場合、マージされていたページはカーネルによって自動でメモリページの複製を作成する。ページの更新確認にはハッシュ値が用いられている。ハッシュ値は jhash2 によって求められている。KSM のプログラマ向けインタフェースとして madvise(2) が存在する。スキャン対象となるメモリ領域は madvise に対して MADV_MERGEABLE を引数として与えることによってスキャン対象となる。KSM は /sys/kernel/mm/ksm にあるファイルで設定とデータの収集が可能となっている。ページの探索、挿入、削除のアルゴリズムに赤黒木を用いている。赤黒木は $O(\log n)$ 以下のオーダーで動作する (n は KSM によってスキャンされたページの合計)。このため短時間に探索、挿入、削除が実現できている。KSM を有効に使用する推奨条件として、複数の同一タイプの VM が存在している状況があげられる。

4 追加調査

Andrea ら [2] の提案ではメモリの外観による検証は行われていない。また中尾 [1] らの検証においては搭載物理メモリに余裕のある状況下での調査であったためそれらについて追加調査を行うこととした。検証環境を表 1、表 2 に示す。

表 1: ホスト側検証環境

CPU	Intel Core2 Quad CPU Q8400 2.66GHz
メモリ	8GB
OS	Debian GNU/Linux 6.0
カーネル	2.6.32-5-amd64
KSM	カーネル 2.6.32-5 時点のもの

表 2: ゲスト側検証環境

CPU	QEMU Virtual CPU version 0.12.5 2.66GHz
メモリ	512MB 割り当て 15VM 構築
OS	Debian GNU/Linux 6.0
カーネル	2.6.32-5-686

ゲスト VM のメモリ割り当て量を 512MB とし、ホ

スト搭載物理メモリを使い切る形で 15 台構築した。3 点の焦点を設けて調査を行った。焦点 1:VM を立ち上げているだけの状態で KSM の無効時、有効時でのメモリ量の概観。焦点 2:VM に割り当てた 512MB を食いつぶす形でメモリを確保し、その上で KSM の無効時、有効時でのメモリ量の概観。焦点 3:VM に割り当てた 512MB のメモリを食いつぶす形でメモリを確保し、それらに乱数を書き込んだ場合における KSM 無効時、有効時でのメモリ量を概観するとした。

5 検証結果

焦点 1~3 に対応した検証結果は以下の図 2 のようになった。単位は MB である。焦点 1 の圧縮量は

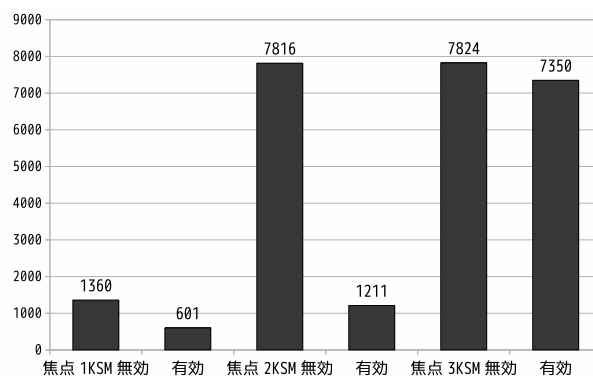


図 2: 焦点 1-3 の検証結果

50.6MB/VM、焦点 2 は 440.3MB/VM、焦点 3 においては 31.6MB/VM となった。以上の結果から分かったことはアプリケーションの特性によってメモリの圧縮量が大幅に変わることである。特に乱数を書き込んだ場合はほぼメモリは圧縮されることがなく、KSM による効果があまり得られないことが判明した。

6 まとめ

アプリケーションによって特性があることが分かったので今後は更にアプリケーションによるメモリ圧縮効果のパターンの調査などを行うことによって、VM システムメモリの使用量の推測が可能になると考えられる。

参考文献

- [1] 中尾 司ビエール, 坂下 善彦. Linux 仮想化環境におけるメモリコミットの分析 2012-DPS-152
- [2] Andrea Arcangeli, Izik Eidus, and Chris Wright.: Increasing memory density by using KSM, Proceedings of the Linux Symposium (2009).