

仮想環境下におけるゲスト OS 内共有ライブラリの ホスト OS による共通化

中尾 司ピエール[†] 坂下 善彦[‡]

[†] 湘南工科大学大学院 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

Linux 仮想化環境における仮想化ソフトウェアとして Kernel-based Virtual Machine(KVM) が存在する。KVM を利用した仮想化環境において有効とされているメモリ節約手法に Kernel Samepage Merging(KSM) が存在する [1]。KSM は、複数のゲスト OS が走行する環境においてゲスト OS が管理するメモリをホスト OS 側からページ単位で横断する。横断結果から同じデータがページ上に存在する場合はホスト OS 上に一つのページとしてまとめ Copy on Write(CoW) の状態で管理する [2]。KSM は同一のデータがページ上に存在する場合において有効である。同一ホスト OS 上で稼働する複数のゲスト OS 内で類似したプログラムが稼働するパターン、ホスト OS の視点からはゲスト OS もアプリケーションとして見えるため同一のゲスト OS が複数稼働する場合においても確率的に有効である。KSM によるメモリ圧縮効果に関して我々は調査を行いその有効性を確認している [3]。我々は KVM による仮想化環境においてゲスト OS 内のアプリケーションが保持する共通のライブラリーを、KSM の機構とは別にホスト OS が管理する共通の領域に置くことによって管理する手法を検討している [4]。提案手法は確率的に圧縮効果を得られる KSM と違いゲスト OS 間で共通しているライブラリ等を明確にした上でゲスト OS 間で共通化をはかりメモリ圧縮効果を狙うものである。本稿は提案手法を検討したものである。

2 仮想化環境における資源問題

仮想化環境においてゲスト OS は CPU, メモリ, ハードディスク, その他の入出力などの資源を再現することによって実現している [5]。これらの資源はゲスト OS

を実行するホスト OS 側によって分配され管理している。この内のメモリ資源に着目する。KVM により構築される仮想化環境においてゲスト OS には必要なメモリが静的に割り当てられる。複数のゲスト OS が要求したメモリ資源を全て使用している状況に陥った場合、ホスト OS 側では高確率でメモリスワップが発生すると考えられる。このため仮想化環境全体の性能劣化が引き起こされる可能性ある。この資源問題はゲスト OS 内で稼働するアプリケーションの実行時メモリ使用量を見通し良く見積もることが困難であることに起因する。ホスト OS 側の管理者側はユーザーの利用を妨げることなく性能劣化を起こす要因を避けたいものである。この問題は、クラウドコンピュータシステムにおける資源見積りの課題と同質のものである。

3 KSM による圧縮効果

KVM によって構築された仮想化環境においてホスト OS によって提供される KSM を利用することでホスト OS のメモリが節約できることが分かっている [1][3]。KVM による仮想化環境は KVM カーネルモジュールとユーザーモードで動作する qemu-kvm から構成されている。ゲスト OS はこのユーザーモードで動作する qemu-kvm プロセスが確保したメモリ上で動作し、必要に応じて KVM カーネルモジュールとやりとりをすることによって実現している。KSM は、ホスト OS 側カーネルにて動作する ksmd カーネルスレッドを有効にすることで動作する。ユーザープロセス (qemu-kvm プロセス) の匿名ページを指定間隔でスキャンし同一内容のページが存在する場合は 1 ページへ CoW の状態でマージし、管理される。マージ済みの領域に対して書き込みが発生した場合はページの複製を作成しマージを解消する。マージ結果として使用されるメモリの容量が圧縮されることになりホスト計算機の資源に余裕が存在するようになる。我々の実験では [3] 意図的にメモリ資源をゲスト OS 内で確保しそこに同一データを書き込んだ場合、ランダムな値のデータを書き込んだ場合におけるマージ状況を外観し確認した。しかし、現実

The common holding of the guest OS shared library
by the host OS in a virtual machine environment

Tsukasapierre Nakao[†], Yoshihiko Sakashita[‡]

[†]Shonan Institute of Technology, Graduate School

[‡]Shonan Institute of Technology, Department of Information
Science

的な利用状況におけるアプリケーションのメモリ利用形態は予測することが難しいと考えられる。これはメモリ書き換え等が頻繁に発生するアプリケーションの場合においては圧縮効果は得られにくく実際は要求はしたが使用されていないメモリ領域などがマージ対象となるだけである。複数のゲスト OS が同一の OS で構成されている場合はプログラム、データのメモリへの配置は同一になる確率が高いためその部分においてのメモリ圧縮の効果が得られることも実験で確認している。これはゲスト OS が KSM を動作させるホスト OS から見た場合アプリケーションとして見えるからである。KSM は各々のゲスト OS 内で使用している共有ライブラリーが共通のものである場合、同一のものとして共通化される。これは複数のゲスト OS が独立して稼働はしているが使用しているライブラリーが同じものの場合に起こることである。

4 ゲスト OS 間でのライブラリー共有

我々の研究目標は、既に述べたように KSM の提案をさらに進め指定したゲスト OS 内のライブラリーを複数のゲスト OS 間でも共有できる機構を目指している。以下に研究目標を実現する上での検討項目を述べる。

4.1 ELF ロード関連処理に注目する方法

共有ライブラリーの呼び出しに利用される ELF ロードに注目している。ゲスト OS 内でアプリケーションが起動する際にはかならず ELF ロードを通して起動される。ロードが機能する際に互いのゲスト OS 間で qemu-kvm を通して、共通するライブラリーを連携するアルゴリズムの考案の必要がある。本研究では現在の ELF ロードを目的にあわせて改修もしくは新規に作成することを考えている。ELF ロードと関連モジュールがゲスト OS 間で連携機能する方法と各ゲスト OS とホスト OS 間で連携機能する方法を検討している。本研究では、後者の連携方式を対象とする。しかし、ELF ロードは個別のゲスト OS の環境依存するものとなるため、環境を超えた制御が必要となる。

4.2 ゲスト OS とホスト OS 間にて連携する方法

ゲスト OS 側で ELF ロードが起動した際、あるいは仮想メモリ空間が生成されるときにそれらの実態をホスト OS の管理領域に生成し、各ゲスト OS 側から参照・変更をできるようにする方法である。この概念を図 1 に示す。各ゲスト OS 内で共有ライブラリーを使用することを検出しその実行する機構と同じ形態の機構をホスト OS 側に設け仮想メモリ空間に関わる管理データを生成管理する。ホスト OS 側で把握したライブラリーをどのゲスト OS が必要としているのかを管理しておく。これらの情報は qemu-kvm を通して検索収集す

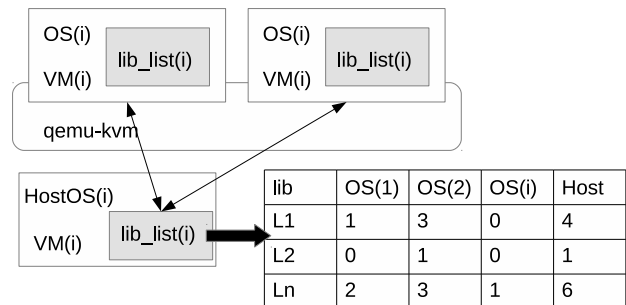


図 1: 提案手法イメージ図

る。ゲスト OS 内で ELF ロードが起動したタイミングで一部の処理機能はホスト OS 側に移管し実行された形態とする。共有する対象となるライブラリー情報はゲスト OS 側に存在し実際の処理は qemu-kvm の処理機構の中で実施されるためその段階の前後で対象ライブラリーに関する情報を取得できると考えている。

5 まとめ

我々は KSM を用いた場合に仮想計算機環境におけるメモリ圧縮効果を予測するための指針を得るための評価実験を行った経験を基に本提案に至った。共有の可能性のあるライブラリー、オブジェクトを意識した上でゲスト OS 間で構造的に共有し利用する機構を検討した。現在は提案、調査段階である。今後は検討をさらに進め実装を目指す。提案手法によりメモリの消費量を削減できる可能性がある。

参考文献

- [1] Andrea Areangeli, Izik Eidus, and Chris Wright: Increasing memory density by using KSM, Proceedings of the Linux Symposium(2009)
- [2] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating System Concepts, John Wiley Sons, Inc. (2005)
- [3] 中尾司ピエール, 坂下善彦:Linux 仮想環境におけるメモリコミットの分析, 情報処理学会 DPS 研究会報告, (2012.9.14)
- [4] 坂下善彦, 中尾司ピエール:仮想環境下におけるゲスト OS 間共有ライブラリーの構築手法 DICOMO (2013.7.12)
- [5] Daniel Bartholomew: QEMU: a multihost, multi-target emulator, Linux Journal, Volume 2006 Issue (145, May 2006)