2A-02

仮想計算機間におけるライブラリ重複に注目したホスト計算機資源の縮小化の検討

中尾 司ピエール† 坂下 善彦‡ 大谷 真‡

†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

共有ライブラリは位置独立コードで構成され実行時には複数のプロセスに共有される. これによりライブラリによる冗長なメモリ消費が抑えられている. 仮想化環境において複数の仮想計算機が稼働している場合に, 共有ライブラリはホスト OS が管理する仮想メモリ空間に個々に配置される. これら個別に配置されている共有ライブラリを仮想計算機間で共通化することにより, 重複を抑えることが可能であると考えている. 我々は, 特定の条件での共有ライブラリの数を調査することとした.

2 先行研究

メモリ重複排除技術は今までに様々な研究がなされており主に [1][2][3][4] が存在する. そして,Linux KVM を主眼とした重複排除手法として,Kernel SamePage Merging が存在する (KSM)[5]. 類似研究として井上らにより KSM をベースに高速化をし重複排除の効果を上げた手法が提案されている [6].

3 共有ライブラリ数の調査

KSM では重複排除効果が確率的に得られるものであることがわかっている. KSM のようなアプローチによる重複排除行った場合, 仮想計算機内で走行するアプリケーションによるメモリの利用方法に依存するためシステムの安定運用においての不安材料となる. 我々は調査結果より重複排除の効果は確率的に得られておりアプリケーションの挙動に依存すると判断した [7]. 以上を踏まえ抜本的なホスト計算機資源の縮小化を目指しゲスト OS 間のライブラリ共通化の提案を行った [8][9]. 提案手法の実現に向け本研究では上に述べた通り, 特定の条件での共有ライブラリの数を調査する. 共有ライブラリの数は使用条件に依存するが, かなりの数であると考えられるので, これらの VM 間での重複がなくなることによる効果は大きいと考える. どの程度共有ライブラリがゲスト OS 稼働時にロードされているかを調査した. 条件として選定した Linux ディストリビューションを最小構成でインストールをし, その時点でロードされている共有ライブラリの種類を対象とした. ディストリビューションは広く利用されている CentOS7 と Debian Wheezy とした. 集計方法は走行中の全てのタスクの/proc/PID/maps 内からロードされている共有ライブラリの一覧を集計し重複しているライブラリを除去している. 集計結果を表 1 に示す.

表 1: 計測結果

OS	タスク数	ユニークなライブラリ数
Debian	65	49
CentOS	98	157

Debian と CentOS で最小構成にも関わらず差が出たが, ディストリビュータ側でのソフトウェアの最小構成の差異が原因と思われる. しかし, これは単一で稼働している際の数であり, 実際はこれらのゲスト OS が複数動く形が仮想化を行うメリットである. 複数動かす結果としてライブラリの重複が発生してしまう. 用途によっては大量の共有ライブラリの重複が仮想計算機間で見受けられることとなる.

4 想定している環境

本研究で想定している仮想化環境は Linux Kernel-based Virtual Machine である (KVM). 仮想化環境上で動作する仮想計算機は複数が単一の物理計算機上で動作することとなる. 本研究では, 仮想計算機内で動作させるゲスト OS は Linux ベースと限定する.

5 仮想計算機間におけるライブラリの重複

仮想計算機上のゲスト OS 内で走行するアプリケーション, 付随するライブラリ群とホスト物理メモリの対

Consideration about the computer resource reduction focused on the library duplication between virtual machines Tsukasa Pierre Nakao†, Yoshihiko Sakashita‡, Makoto Oya‡†Shonan Institute of Technology, Graduate school of electric and information science
‡Shonan Institute of Technology, Department of Information Science

応関係は図1のようになる.

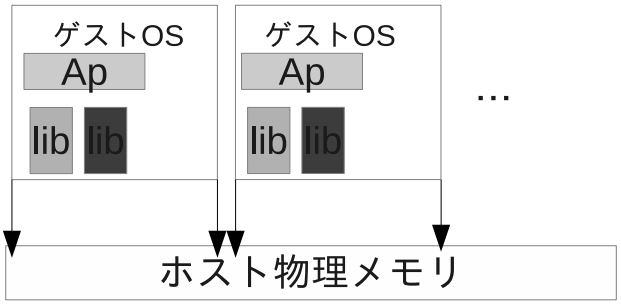

図 1: ホスト物理メモリの対応

すでに述べたがゲスト OS はホスト Linux から見た場合ユーザーランドのプロセスとして動作するためゲスト OS 内で動作するアプリケーションが利用するライブラリも複数のゲスト OS が走行する場合それぞれ個々のホストが管理する物理メモリ空間にマップされる. ゲスト OS 内で同一のアプリケーションが走行する際にアプリケーションが利用するライブラリも個々にホストの管理する物理メモリ空間にマップされるため重複が発生する.

6 検討した手法

調査結果より共有ライブラリの重複を踏まえ検討を行った. 検討している点を図2に示す.

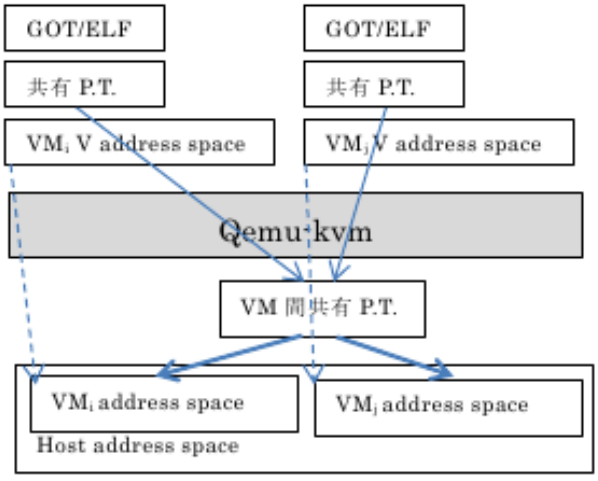

図 2: 検討手法

現在の OS ではシステム内での共有ライブラリの共有制御は行われている. これは仮想化されたゲスト OS でも同様の振る舞いをする. 本研究はこれらの既存の機構に仮想計算機側の共有ライブラリの管理情報を何等かの機構を考案してホスト OS 側に渡す事を目指している. ゲスト OS 側で共有ライブラリのアドレスを獲得する挙動を qemu 側でフェッチする. それに応じホスト OS 側にも仮想計算機が共有ライブラリのアドレスを獲得することであることを通知する. ホスト OS 側では共有ライブラリの管理を行うテーブルを持つ. このテーブルに置いては同一共有ライブラリの有無を刷新し, 同一であれば同じメモリアドレスを返すことを行う. 以上の検討を踏まえたホスト OS とゲスト OS を繋ぐツールの作成が必要となる. ツールは qemu 側から VM 側をプローブするツール,qemu 側からホスト OS 側へリクエストを出すことを可能とするツールとなる. 現在, ツール作成については解析フレームワークである LibVMI[10] を参考にしている. ホスト OS 側からゲスト OS の改変をせずにプロセス, カーネルモジュールの一覧の取得等に成功している. ここから更に上に述べたゲスト OS 内の共有ライブラリアドレスの獲得等を行う.

7 まとめ

以上の事を踏まえ更に詳細を明らかにし検討した手法の実装を目指す.

参考文献

[1] EDOUARD BUGNION, SCOTT DEVINE, KINSHUK GOVIL, and MENDEL ROSENBLUM:Disco: Running Commodity Operating Systems on Scalable Multiprocessors OSDI97

[2] Grzegorz Mios, Derek G. Murray, Steven Hand and Michael A.Fetterman:Satori: Enlightened page sharing (USENIX09)

[3] CarlA.Waldspurger:MemoryResource Management in VMwareESXServer (OSDI '02)

[4] Diwaker Gupta, Sangmin Lee, Michael Vrable,Stefan Savage, Alex C. Snoeren, George Varghese, Geoffrey M. Voelker, and Amin Vahdat:Difference Engine:Harnessing Memory Redundancy in Virtual Machines (OSDI08)

[5] Andrea Areangeli, Izik Eidus, and Chris Wright: Increasing memory density by using KSM, Proceedings of the Linux Symposium 19-28, 2009.

[6] 井上宗士 山田浩史 河野健二 :仮想化環境におけるメモリ情報に着目したページシェアリング効率化手法 SIGOS 2014-OS-128,no2, pp.1-7,2014-02-27

[7] 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, vol.2012, no.32, pp.1-5, 2012-09-06

[8] 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)

[9] 中尾 司ピエール, 坂下 善彦:仮想化環境におけるゲスト OS 内共有ライブラリのホスト OS による共通化 第 76 回 情報処理学会 全国大会

[10] Bryan D. Payne:Simplifying Virtual Machine Introspection Using LibVMI SANDIA REPORT SAND2012-7818