

平成 24 年度 卒業論文概要			
所 属	機械情報工学科	指導教員	光来 健一
学生番号	09237015	学生氏名	大庭 裕貴
論文題目	ネストした仮想マシンを用いたメンテナンスの高速化		

1 はじめに

近年、計算機の高性能化に伴い、一台の計算機の中で複数の仮想マシン（VM）を動作させ、VM 内でサービスを提供するようになってきている。このような仮想化を行う場合、仮想化システム自体のメンテナンスが必要となる。VM を停止させずに仮想化システムのメンテナンスを行うために、メンテナンス時に VM を別のホストに移動させるマイグレーションと呼ばれる手法が用いられている。しかし、マイグレーションはシステムやネットワークへの負荷が大きいことから、その影響を抑えるためにはマイグレーションの速度を調節しなければならなかった。

本研究では、ネストした VM を用いて仮想化システムのメンテナンスを高速化する VMBeam を提案する。

2 仮想化システムのメンテナンス

VM を用いた仮想化を行う場合、仮想化システム自体のメンテナンスが必要になる。仮想化システムはハイパーバイザと呼ばれるソフトウェアを基盤として用いている。ハイパーバイザに不具合が見つかった場合には、それを修正するためにパッチを当て、ハイパーバイザの再起動を行う必要がある。その際には、ハイパーバイザ上で動作しているすべての VM を一旦停止させ、ハイパーバイザの再起動後に再び VM を起動し直すことになる。多くの VM の停止や起動には時間がかかるため、その間、サービスを提供できないダウンタイムが生じる。

メンテナンス時のダウンタイムを削減するために、図 1 のように、VM を別のホストにマイグレーションしてからハイパーバイザを再起動する手法が用いられている。マイグレーションは VM を動作させたまま別のホストに移動する機能である。マイグレーション中に VM が停止するのはわずかな時間であるため、VM 内のサービスのダウンタイムは非常に短い。すべての VM をマイグレーションしてからハイパーバイザを再起動することで、VM に再起動の影響を及ぼさないようにすることができる。

しかし、VM のマイグレーションはシステムやネットワークに大きな負荷をかけるため、マイグレーション中の性能低下が問題となる。VM をマイグレーションするには、VM のメモリの内容をネットワークを介して別のホストに転送する必要がある。すべての VM をマイグレーションするには数 GB～数十 GB のデータを転送しなければならず、動作中の VM に対しても大きな影響を与える。そのため、システム全体への影響を考えて、マイグレーションの速度を抑える必要があり、メンテナンスに時間がかかる原因となっていた。

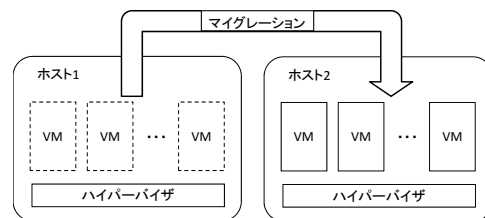


図 1 マイグレーションを用いたメンテナンス

3 VMBeam

本研究では、ネストした VM を用いて仮想化システムのメンテナンスを高速化する VMBeam を提案する。ネストした VM とは、VM の中で別の仮想化システムを動作させることを可能にする技術である。この技術を用いて、同一ホスト上の二つの VM それぞれで別々の仮想化システムを動作させ、一方の仮想化システムをメンテナンスする際には、もう一方の仮想化システム上に VM をマイグレーションする。この際に、VM のメモリの内容をネットワーク経由で転送する代わりに、同一ホスト上にあることを利用して VM 間でメモリをスワップする。この手法により、マイグレーションの高速化および負荷の軽減を図る。

3.1 ネストした VM

ネストした VM を用いたシステム構成を図 2 に示す。区別のために、通常の仮想化システムにおけるハイパーバイザ、VM をそれぞれホスト・ハイパーバイザ、ホスト VM と呼び、ホスト VM 内で動作するものをそれぞれゲスト・ハイパーバイザ、ゲスト VM と呼ぶ。本研究でメンテナンスの高速化の対象とするのはゲスト・ハイパーバイザであり、ホスト・ハイパーバイザは対象としない。ホスト・ハイパーバイザは最低限の機能だけを提供することを想定しているため、メンテナンスの頻度は低いと考えられる。

本研究では、ホスト・ハイパーバイザに Xen を用い、ゲスト・ハイパーバイザに Xen-Blanket[1] を用いた。Xen-Blanket は Xen を拡張してホスト VM 内で動作させられるようにしたも

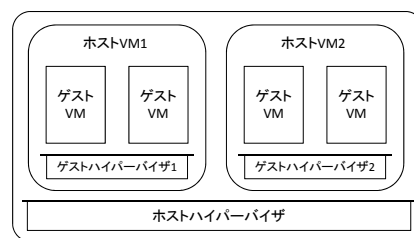


図 2 ネストした VM のシステム構成

のであり、ネストした VM による性能低下も改善している。ただし、現在の実装では、ゲスト VM 内では Xen 用に改造された Linux だけを動作させることができる。

3.2 VM 間メモリスワップ

VMBeam では、ホスト・ハイパーバイザがホスト VM 間でメモリを高速に入れ替える機能を提供する。この VM 間メモリスワップを実現するために、ホスト・ハイパーバイザは二つのホスト VM へのメモリ割り当てを変更する。Xen において、ホスト・ハイパーバイザはマシンメモリと呼ばれるホスト全体で管理している物理メモリを扱う。また、ホスト・ハイパーバイザはマシンメモリの一部をホスト VM に割り当て、疑似的な物理メモリとして扱わせる。この疑似物理メモリからマシンメモリへの対応は、P2M テーブルと呼ばれる対応表で管理されている。そこで、ホスト・ハイパーバイザが P2M テーブルを書き換え、図 3 のようなメモリ割り当てに変更することで、二つのホスト VM 間でメモリを入れ替える。

ホスト・ハイパーバイザ内ではメモリ管理のために、マシンメモリから疑似物理メモリへの対応表 (M2P テーブル) も管理されているため、この対応表も同時に書き換える。また、ホスト VM に割り当てられているメモリの一覧もリストで管理しているため、入れ替えたメモリに応じてこのリストも更新する。

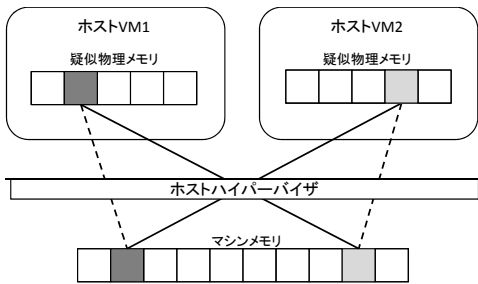


図 3 ホスト VM 間でのメモリスワップ

3.3 メモリスワップを用いたマイグレーション

VMBeam では、ホスト VM 間のメモリスワップを用いてマイグレーションの負荷を軽減する。このマイグレーションの流れを図 4 に示す。

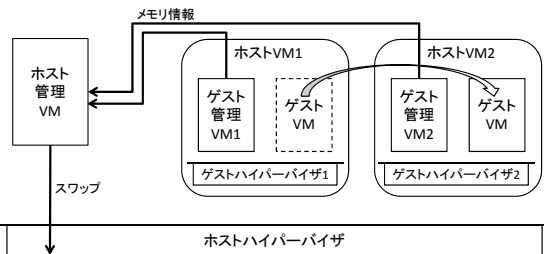


図 4 メモリスワップを用いたマイグレーション

ゲスト VM のマイグレーションはゲスト管理 VM と呼ばれる VM 間で行われる。従来のマイグレーションでは、マイグレーション元のゲスト管理 VM がゲスト VM のメモリの内容をマイグレーション先に送っていた。VMBeam ではその代わりに、ゲスト VM に割り当てられている疑似物理メモリの情報だけをホスト管理 VM に送る。同時に、マイグレーション

先のゲスト管理 VM は新たに作成したゲスト VM に割り当てた疑似物理メモリの情報をホスト管理 VM に送る。これらのメモリ情報を受け取ったホスト管理 VM は、ホスト・ハイパーバイザを呼び出すことによってメモリのスワップを行う。

4 実験

ゲスト VM に割り当てるメモリサイズを 256MB, 512MB, 1024MB と増やした場合のメモリスワップにかかる時間を計測した。実験には Intel Xeon E5-2665 2.40GHz (16 コア) の CPU, 32GB のメモリを搭載したマシンを使用した。ホストハイパーバイザとして Xen 4.1.3, ゲストハイパーバイザとして Xen-Blanket 4.1.1 を動作させ、ホスト管理 VM では Linux 3.2.0, ゲスト管理 VM では Xen-Blanket 用に拡張された Linux 3.1.2 を用いた。ホスト管理 VM には 2 CPU と 8GB のメモリ, ゲスト管理 VM には 4 CPU と 4GB のメモリを割り当てた。

メモリスワップ時間を 10 回ずつ計測した平均値と、VM 間の仮想ネットワークで転送した場合にかかる時間を図 5 に示す。この結果より、メモリスワップにかかる時間はゲスト VM のメモリサイズに比例することが分かる。割り当てるメモリサイズが 1024MB の場合でも、ゲスト VM 間のメモリスワップにかかる時間は 0.5 秒となった。仮想ネットワークを用いた転送と比較しても、メモリスワップの方が 28% 高速に行えることが分かった。

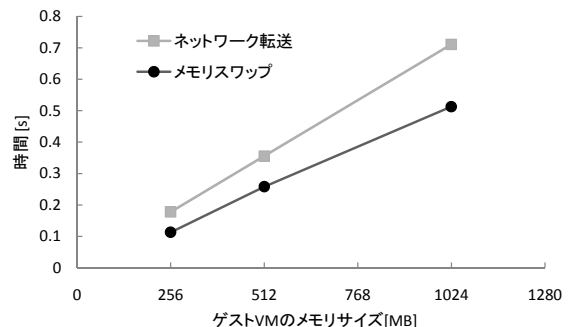


図 5 メモリスワップと仮想ネットワーク転送の比較

5 おわりに

本研究では、ネストした VM を用いて仮想化システムのメンテナンスを高速化する VMBeam を提案した。VMBeam では、同一ホスト上で二つの仮想化システムを動作させ、仮想化システムのメンテナンス時には VM をもう一方の仮想化システムにマイグレーションする。このマイグレーションは VM 間メモリスワップを用いて高速化する。今後の課題は、VM 間メモリスワップをマイグレーションに組み込む実装を完成させることである。

参考文献

[1] Williams et al, The Xen-Blanket: Virtualize Once, Run Everywhere. Proc. EuroSys, 2012.