

平成 26 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	11237025	学生氏名	木津 巴都希
論文題目	大きな仮想マシンの複数ホストへのマイグレーション手法		

1 はじめに

近年、クラウドコンピューティングの普及が進んでいる。そのサービス形態の一つである IaaS 型クラウドでは、仮想マシン (VM) をサービスとしてユーザに提供し、ユーザが必要な時に必要なだけ利用することができる。IaaS 型クラウドの発展に伴い、一台のサーバで多くの VM を稼働させるだけでなく、巨大な VM も提供されるようになってきた。例えば、ビッグデータの解析には巨大なメモリを持つ VM が必要とされている。その一方で、巨大なメモリを持つ VM はマイグレーションを行うのが困難になるという問題がある。マイグレーションはホストをメンテナンスする際に VM を停止させることなく他のホストへ移動させる技術である。VM のマイグレーションを行うためには移送先のホストに十分な空きメモリ容量が必要となる。しかし、マイグレーションのために大容量のメモリを備えたホストを確保しておくのはコストの面から難しいことが多い。VM のマイグレーションが行えない場合は、ホストのメンテナンス時に VM を停止させなければならない。

本研究では、巨大なメモリを持つ VM を複数のホストにマイグレーションすることを可能とする S-memV を提案する。

2 巨大なメモリを持つ VM のマイグレーション

VM マイグレーションは稼働している VM を停止させることなく別のホストへ移動させる技術である。マイグレーションを活用することで、VM が提供しているサービスを停止させることなくホストのメンテナンスを行うことができる。マイグレーションを行う際には、移送先のホストに新しい VM を作成し、移送元のホストで動いている VM のメモリの内容をネットワーク経由で移送先の VM のメモリにコピーする。この間、移送元のホスト上で VM のメモリの内容は更新され続けているため、再度、更新されたメモリを転送する。これを繰り返して転送するメモリ量が十分小さくなったら移送元の VM を停止し、更新されたメモリを転送してマイグレーションを完了する。

近年、巨大なメモリを持つ VM が使われるようになってきた。例えば、Amazon EC2 では 244 GiB のメモリを持つ VM が提供されている。このような VM はビッグデータの解析などのように、大量のデータを扱う場合に用いられている。また、メモリ上にデータを保持するデータベースを使う際にも有用である。しかし、このような巨大なメモリを持つ VM は、マイグレーション時に移送先を見つけるのが難しくなるという問題がある。移送先として大きな空きメモリ容量を持ったホストを確保し続けておくことは、コスト面から難しいことが多い。また、十分なメモリ容量を持つホストが多数の小さな

VM を動かすために使われていた場合、まず、それらをマイグレーションする必要がある。

従来、メモリ容量の小さなホストで大きなメモリを持つ VM を動かすには、OS の仮想メモリが用いられてきた。仮想メモリは物理メモリに入りきらないデータをディスク上のスワップ領域に待避することで、物理メモリよりも大きなメモリを扱うための技術である。しかし、メモリに比べてディスクは非常に遅いため、スワップを繰り返すと性能が著しく低下する。特に、マイグレーション中は更新されたメモリが何度も転送されるため、転送のたびにスワップが発生する可能性がある。スワップによってマイグレーションの時間が長くなると、その分だけ更新されるメモリも増え、さらにマイグレーション時間が長くなってしまう。

3 S-memV

本研究では、巨大なメモリを持つ VM を複数のホストにマイグレーションすることを可能とする S-memV を提案する。S-memV では図 1 のように VM の移送先のホストは 1 つのホストとは限らず、1 つのメインホストと複数のサブホストからなる。S-memV は CPU やデバイスの状態のような VM の核となる情報をメインホストに送る。また、VM が頻繁にアクセスするメモリはできるだけメインホストに送る。一方、メインホストに入りきらなかった VM のメモリはサブホストに送る。マイグレーション後はメインホストで VM を動作させ、VM がサブホストにあるメモリを必要とした時には、メインホストとサブホストの間でメモリをスワップする。

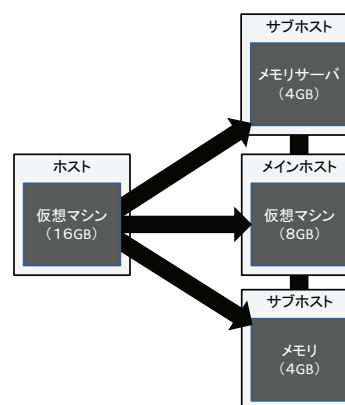


図 1 S-memV におけるマイグレーション

S-memV では、複数のホストにマイグレーションされた VM を再び 1 つのホストで動作するようにマイグレーションすることもできる。その場合、メインホストで動いている VM

のメモリとサブホストにある VM のメモリを移送先ホストに送り、元の完全な VM のメモリを復元する。さらには、複数のホストにマイグレーションされた VM を別の複数のホストにマイグレーションすることも可能である。その場合、移送先のメインホストのメモリ容量に応じて、移送元のメインホストとサブホストからメモリを送り、残りのメモリは移送先のサブホストに送る。

3.1 memserver

memserver は VM のメモリの一部を管理するサーバであり、サブホスト上で動作する。memserver は VM のメモリを 4KB のページ単位で管理する。この管理にはメモリの利用効率がよいデータ構造である基数木を用いる。基数木は Linux カーネルにおいてページキャッシュの管理にも用いられている。memserver は基数木を用いて、VM のメモリのアドレスをキーとして、対応する VM のメモリページを管理する。

memserver が VM のメモリアドレスと VM のメモリページからなる登録要求を受信した時には、4KB のメモリを確保して VM のメモリページの内容をコピーし、基数木に登録する。一方、VM のメモリアドレスからなる探索要求を受信した時には、基数木を探索し、VM のメモリページが見つければそのデータを要求元に送信する。

3.2 マイグレーション処理の拡張

複数のホストを利用して VM のマイグレーションを行えるようにするために、仮想化ソフトウェア KVM のマイグレーション機構への拡張を行った。KVM では QEMU-KVM と呼ばれる VM を管理するためのソフトウェアがマイグレーションを行う。S-memV では、マイグレーションを開始した時に、図 2 のように、移送先のメインホストで動作している QEMU-KVM だけでなく、サブホストで動作している memserver にもネットワーク接続する。適切な数のサブホストを見つけて、それらすべてに接続を行う必要があるが、現在の実装ではサブホストは 1 つで固定としている。

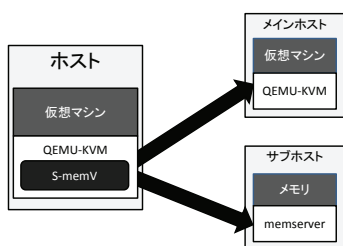


図 2 メインホストとサブホストへのメモリ転送

QEMU-KVM はすべてのメモリページをメインホストと各サブホストのいずれかに転送する。マイグレーション中に更新されたメモリは再送されるため、再送時にも同じホストに送られるように、送信先の一覧を管理しておく。現在の実装では、簡単のために、メモリアドレスに閾値を設定して、閾値より小さければメインホストへ、大きければサブホストへメモリページを送っている。

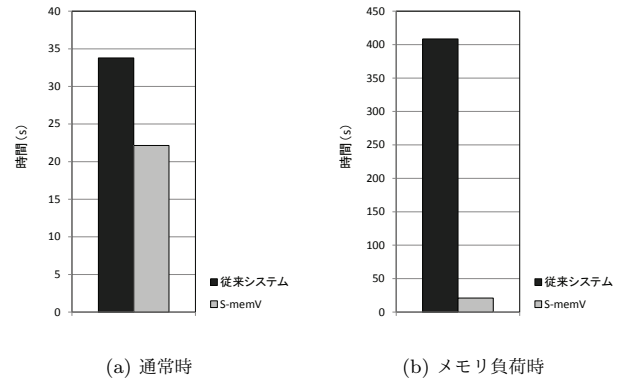


図 3 マイグレーション時間

4 実験

S-memV を用いて、移送先のホストに入りきらない大きさのメモリを持つ VM をマイグレーションに要する時間を測定した。比較として、移送先でディスク上のスワップ領域を用いる従来システムについてもマイグレーション時間を測定した。VM にはメモリを 2GB 割り当てた。移送先のホストのメモリは 2 GB であったが、その内約 1 GB は使用中であった。また、移送先のスワップ領域のサイズは 4GB であった。

まず、VM 内でアプリケーションを動作させずにマイグレーションを行った。その結果を図 3(a) に示す。従来のシステムでもスワップ領域を用いることでマイグレーションに成功していた。しかし、S-memV と比べると、ディスクへのアクセスを行うため、マイグレーションに約 1.5 倍の時間がかかった。

次に、VM 内で memcached を動作させ、ベンチマークを用いてメモリに負荷をかけた。memcached はメモリ上にデータを保持するデータベースである。この場合のマイグレーション時間は図 3(b) のようになった。スワップ領域を用いる従来システムではディスクアクセスが頻発し、マイグレーションに非常に時間がかかることが分かった。一方、S-memV ではメモリに負荷をかけても通常時と同じ時間でマイグレーションが行えることがわかった。

5 まとめ

本研究では、巨大なメモリを持つ VM を複数のホストへマイグレーションすることを可能とする S-memV を提案した。S-memV は移送先のメインホストに入りきらない VM のメモリをサブホストに転送する。マイグレーション後、VM はメインホストで動作し、必要に応じてメインホストとサブホスト間でメモリをスワップする。S-memV を KVM に実装し、サブホストで memserver を動作させて、メインホストとサブホストを用いたマイグレーションを実現した。

今後の課題は、VM をマイグレーションした後、メインホストとサブホストの間でメモリをスワップできるようにすることである。また、VM が頻繁にアクセスするメモリページを検出して、それらのページをなるべくメインホストに送ることができるようにする必要がある。さらに、複数のホストで動作する VM をマイグレーションできるようにすることも今後の課題である。