平成 30 年度 卒業論文概要			
所　　属	機械情報工学科	指導教員	光來 健一
学生番号	15237038	学生氏名	田内　聡一朗
論文題目	未使用メモリに着目した複数ホストにまたがる仮想マシンの高速化		

1　はじめに

近年、クラウドサービスの一つとして、ユーザに仮想マシン（VM）を提供する IaaS 型クラウドが普及している。それに伴い、大容量のメモリを持つ VM が提供されるようになってきている。例えば、Amazon EC2 では 12TB のメモリを持つ VM が提供されており、ビッグデータの解析などに利用されている。VM はホストのメンテナンス等の際に別のホストへマイグレーションされるが、大容量メモリを持つ VM の転送先として十分な空きメモリを持つホストを常に確保しておくのはコストの面での負担が大きい。そこで、VM のメモリを複数の小さなホストに分割して転送する分割マイグレーション [1] が提案されている。分割マイグレーション後にはリモートページングを行って VM が必要とするメモリをホスト間で転送する。しかし、従来の分割マイグレーションとリモートページングでは、転送しようとするメモリの中に使われているデータがなかったとしても転送を行う必要があった。

本研究では、未使用メモリに関連するオーバヘッドを削減することで複数ホストにまたがる VM の高速化を実現するシステム FCtrans を提案する。

2　複数ホストにまたがる VM

近年、IaaS 型クラウドでは大容量メモリを持つ VM が提供されるようになっている。このような VM が動作しているホストをメンテナンスする際などには、マイグレーションにより VM を別のホストに移動させることで実行を継続する必要がある。マイグレーションを行うには移送先に VM のメモリよりも大きな空きメモリが必要となるが、十分な空きメモリを持つホストを常に確保しておくのはコストがかかる。そこで、図 1 のように複数のホストへ VM を分割してマイグレーションを行う分割マイグレーション [1] が提案されている。分割マイグレーションでは、CPU やデバイスの状態などの VM の核となる情報と VM がアクセスすることが予測されるメモリをメインホストへ転送する。一方、メインホストへ入りきらないメモリはサブホストへ転送する。

マイグレーション後の VM は複数のホストにまたがって動作する。メインホスト上では VM 本体が動作し、サブホストはその VM にメモリを提供する。VM はメインホスト上のメモリに直接アクセスすることができるが、サブホスト上のメモリにはリモートページングを行ってアクセスする。VM がサブホストに存在するメモリを必要とした際には、そのメモリをメインホストへ転送（ページイン）する。同時に、メインホスト上の今後アクセスされる可能性が最も低いと予測されるメモリをサブホストへ転送（ページアウト）する。分割マイグレーションではアクセスされる可能性が高いメモリがメインホストに転送されるため、マイグレーション直後のリモートページングの頻度は低い。

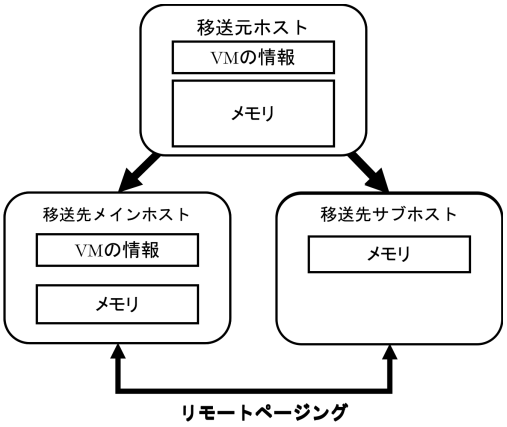

図 1　分割マイグレーション

このような大容量メモリを持つ VM のマイグレーションには時間がかかる。また、VM がメインホストのメモリ容量以上のデータにアクセスすると、リモートページングが発生して性能が低下する。その一方で、VM のメモリの中には使われていない領域が存在することも多い。例えば、VM 内で OS が起動した直後には VM のメモリ領域の多くは未使用である。一度使用したメモリ領域であっても OS が解放すると未使用状態となる。しかし、従来の分割マイグレーションは未使用メモリであっても移送先ホストに転送を行う。また、従来のリモートページングは未使用メモリに対してもサブホストからページインを行い、別の未使用メモリをページアウトする。そのため、メモリの転送のオーバヘッドが大きい。

3　FCtrans

本研究では、VM の未使用メモリに着目し、複数ホストにまたがる VM の高速化を可能とする FCtrans を提案する。FCtrans は VM の起動時から未使用メモリを追跡し、マイグレーション後も追跡を続ける。追跡した未使用メモリの情報を用いて、マイグレーション時には移送先ホストに未使用メモリを転送しないようにする。これにより、ネットワーク転送量を削減し、マイグレーションを高速化することができる。また、分割マイグレーション後には未使用メモリに対してリモートページングを行わないようにする。その結果、VM が未使用メモリを必要とした時により高速にメモリを用意し、即座に VM を再開させることができる。

3.1　VM の未使用メモリの管理

FCtrans はビットマップと呼ばれるデータ構造を用いて、VM のメモリを 4KB のページ単位で使用中であるか未使用で

あるか管理する。このビットマップは使用ビットマップと呼ばれる。使用ビットマップは VM に割り当てられているメモリページ数分のビットからなり、ページが使用中であれば対応するビットが 1、未使用であれば 0 となる。VM 全体のメモリ使用状況を管理するために、使用ビットマップは VM 起動前に作成する。VM の起動時にはすべてのページが未使用であり、VM がメモリにアクセスするとそのページは使用中となる。

VM による未使用メモリへのアクセスを検出するために、FCtrans は Linux の userfaultfd 機構を用いる。この機構により、VM が未使用のメモリページにアクセスするとページフォールトが発生する。その際に、FCtrans はアクセスされたページに物理メモリを割り当て、使用ビットマップの対応するビットを 1 にする。この処理をページ単位で行うとオーバヘッドが大きくなるため、FCtrans はアクセスされたページを含む複数のページに一括で物理メモリを割り当てる。

3.2 分割マイグレーションの高速化

FCtrans は分割マイグレーションにおいて未使用メモリの転送を行わないようにする。移送先ホストにページ単位でメモリを転送する際に使用ビットマップを調べ、対応するビットが 1 の時にだけメモリの転送を行う。これにより、未使用メモリのデータ転送および、移送元ホストにおけるメモリデータの読み出し、移送先ホストにおけるメモリデータの書き込みのオーバヘッドを削減することができる。その結果、分割マイグレーションを高速化することができる。

移送先メインホストでは受信したメモリ情報に基づいて使用ビットマップの再構築を行う。マイグレーション開始時には使用ビットマップのすべてのビットを 0 にしておく。そして、メインホストに格納されるメモリデータとサブホストに格納されるページの情報を受信した際には、使用ビットマップの対応するビットを 1 にする。これにより、移送元ホストで未使用だったページは移送先ホストでも未使用のままとなる。

3.3 リモートページングの最適化

分割マイグレーション後に、VM がメインホストに存在しないメモリにアクセスしてページフォールトが発生すると、FCtrans は使用ビットマップを調べる。VM がアクセスしたページが使用中であれば、従来通りにサブホストからページインを行う。一方、VM がアクセスしたページが未使用であった場合には、メインホストの物理メモリを割り当て、サブホストからのページインは行わない。これにより、ページインに伴うオーバヘッドを削減し、VM の性能を向上させることができる。未使用ページへの物理メモリの割り当てにより、メインホストの空きメモリがなくなった場合には、従来通りにサブホストへのページアウトを行って空きメモリを確保する。空きメモリがある場合にはページアウトも行わないため、さらに VM の性能を向上させることができる。

3.4 OS が解放したメモリの考慮

FCtrans は VM 内の OS が使っていないメモリを未使用メモリとして扱う。OS がメモリを確保して一度でもアクセスすると、そのメモリを解放して使わなくなったとしても VM のメモリは使用中のままである。そこで、FCtrans は定期的に VM 内の OS が管理しているページ情報を取得し、参照カウントが 0 であれば VM のページを未使用状態に戻す。具体的には、ページに割り当てられた物理メモリを解放し、使用ビットマップの対応するビットを 0 にする。VM の外から OS の情報を取得するために LLView[2] を用いた。LLView は OS のソースコードを利用して VM 内の OS の情報を取得することを可能にするフレームワークである。LLView を用いてコンパイルすることにより、記述したプログラムが OS データを取得しようとした時に VM 内のメモリにアクセスするようにプログラム変換を行う。

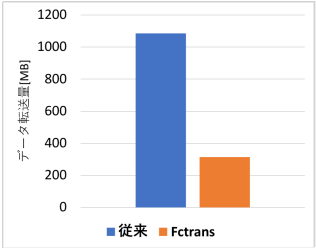

図 2　データ転送量

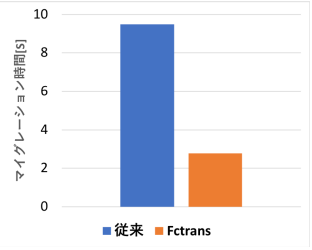

図 3　マイグレーション時間

4 実験

FCtrans による性能向上とオーバヘッドを調べる実験を行った。比較として、従来の分割マイグレーションとリモートページングを行うシステムを用いた。実験には、移送元ホストに Intel Core i7-7700 の CPU、8GB のメモリを搭載したマシン、移送先メインホストに Intel Xeon E3-1225 v5 の CPU、8GB のメモリを搭載したマシンを用いた。移送先サブホストは移送先メインホストと同一とした。仮想化ソフトウェアには QEMU-KVM 2.11.2 を用いた。VM には 1GB のメモリを割り当てて Linux 4.4.0 を動かし、メインホストとサブホストに半分ずつに分割した。

まず、分割マイグレーション中のデータ転送量とマイグレーション時間の測定を行った。図 2 より、FCtrans はデータ転送量を 71 %削減できることが確認できた。それに伴い、マイグレーション時間も 71 %短縮された。次に、VM 内の OS の起動時間を測定したところ、FCtrans では 13 %のオーバヘッドが確認された。

5 まとめ

本研究では、未使用メモリの不要な転送に着目して高速化を実現するシステム FCtrans を提案した。FCtrans は VM のメモリ使用状況を管理して、分割マイグレーション時およびリモートページング時に未使用メモリを転送しないようにする。今後の課題は、VM が解放したメモリを調べる際に VM の実行を一時停止しないように、VM への影響を抑えた未使用メモリの検索方法を検討することである。

参考文献

[1] H.Kizu M. Suetake, T.Kashiwagi and K.Kourai. *S-memV:Split Migration of Large-Memory Virtual Machines in IaaS Clouds*. CLOUD 2018.

[2] 植木あずさ. LLVM の中間表現を用いた IDS オフロードの開発支援. 九州工業大学卒業論文, 2015.