Open Flow を用いた遠隔ライブマイグレーションの提案

2009SE037 原 起知　2009SE224 岡田 幸大
指導教員：青山 幹雄

1　はじめに

近年，クラウドコンピューティングの発達により，ネットワークを利用して様々なサービスを得ることができる．仮想化した OS のことを VM (Virtual Machine) と呼び，クラウドサービスを提供するサーバは，VM 上で稼働できる．また VM は別のホスト上に移動できる．これをマイグレーションと呼ぶ．本研究ではマイグレーションを効率良く行うことを目的とする．

2　研究の背景

OS を仮想化することによって，1台のホスト上で複数の VM を稼働できる．ホストの稼働台数を抑えることができ，初期コストや運用コストを削減できる．また，VM のリソース不足時に簡単にリソースの割当を変更できる．しかし，VM が稼働しているホストのリソース不足時や，障害発生時においてマイグレーションを行う必要がある．

3　研究課題

仮想化ソフトウェアは Xen を使用する[2]．Xen を用いて手動でマイグレーションを実行する場合，xm コマンドを用い，VM 名，移動先 IP アドレスを指定する．マイグレーションは通常 LAN 内で行われることを想定している．LAN 内で行われるマイグレーションを図 1 に示す．

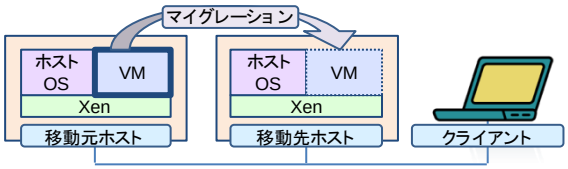

図1 LAN 内のマイグレーション

LAN 間でマイグレーションを行おうとすると，VM に接続しているクライアントがアクセスできなくなる．IP アドレスやサブネットマスクなどの設定も引き継ぐため，変更を行う必要がある．

本研究では，オープンソースのソフトウェアを用いて LAN 間でマイグレーションを行う仕組みを提案する．

4　関連技術

4.1　ライブマイグレーション

マイグレーションはあるホスト上で稼働する VM を，別のホスト上へ移し替える技術である．特に，クライアントと VM との接続が途切れることなくマイグレーションを行うことをライブマイグレーションと呼ぶ[1]．ライブマイグレーションの様子を図 2 に示す．

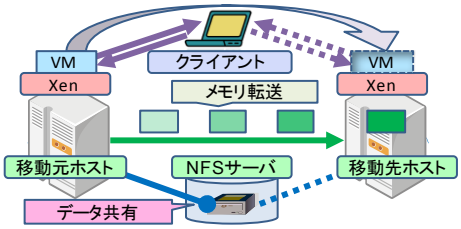

図 2 ライブマイグレーション

4.1.1 マイグレーション時のデータ引き継ぎ

VM は通常の OS と同じく，主記憶装置と補助記憶装置から成り立つ．したがって，マイグレーションを行うにあたり，この 2 つのデータに注目する．

主記憶装置内のデータは，ネットワークを経由して転送する．VM の主記憶装置の容量分を，移動先ホストの Xen に転送する．

補助記憶装置はメモリに対して容量が大きいため，そのまま転送を行うと多大な時間を要する．一般的に，補助記憶装置のデータをそのまま転送する方法はとられていない．例として，ネットワーク上のデータにアクセスできる NFS (Network File System) サーバを用いて，データを転送せずにディスクデータを引き継ぐ．

4.1.2 ライブマイグレーションの動作

ライブマイグレーションは次の 6 ステップから成る．

(1) 機能要件のチェック
　ライブマイグレーションが可能であるかどうか，プロセッサの互換性，移動先ホスト上に必要な空きメモリがあるかをチェックする．

(2) 移動先ホストの VM 構成
　移動先の仮想マシンのメモリ領域を確保し，メモリ転送の受け入れ準備を行う．

(3) メモリの転送
　移動元ホストから移動先ホストへ，メモリデータを転送する．一度転送が終了してもその間にメモリの変更が発生するため，差分ブロックを再転送する．移動元と移動先が同期するまでブロック単位で転送する．

(4) 移動元ホストの VM 停止
　メモリ同期完了後，VM を停止する．

(5) 移動先ホストの VM 再開
　移動先ホストは，メモリを受け取っており，ディスクデータにもアクセス可能なため，そのまま VM を再開する．同時にネットワークスイッチに対して，MAC アドレステーブルを更新する．

(6) 移動元ホストの VM 構成の削除

VM 構成を削除してマイグレーションが終了する.

4.2 OpenFlow

OpenFlow とは, ネットワークパケット転送の流れをソフトウェアを用いて制御する技術である[4].

4.2.1 OpenFlow の構成

OpenFlow は OpenFlowSwitch (以下スイッチ) と OpenFlowController (以下コントローラ)から構成される. スイッチとコントローラは OpenFlow Protocol を用いて通信する.

4.2.2 フロー制御

OpenFlow スイッチは, フローエントリを用いてパケットの転送を制御する. フローエントリは, ヘッダーフィールド, アクション, 統計情報から成り立つ.

(1) ヘッダーフィールド

ヘッダーフィールドでは, スイッチが受信したパケットのレイヤ 1 からレイヤ 4 での情報の組み合わせを元に, パケットの識別をする.

(2) アクション

アクションでは, ヘッダーフィールドに一致するパケットを受信した場合に行う動作を定義する.

(3) 統計情報

統計情報では, ヘッダーフィールドに一致した通信の量を管理する. この管理されている情報は, コントローラからアクセスできる.

4.2.3 OpenFlowProtocol

スイッチがパケットを受信すると, スイッチ内に記録されているフローエントリのヘッダーフィールドを元にアクションを決定する. ヘッダーフィールドの条件に一致しないパケットを受信した場合は, スイッチからコントローラに問い合わせを行い, スイッチに新しいフローエントリを書き込む. スイッチとコントローラの通信に用いる.

4.2.4 Open vSwitch

Open vSwitch とは OpenSwitch をソフトウェアで実現したオープンソースの仮想ソフトウェアスイッチである. 仮想化ソフトウェアとともに利用されることが多い. Open vSwitch を用いると, PIF (Physical Interface)と Open vSwitch が接続する. また, Open vSwitch は VIF (Virtual Interface) を介して VM と接続する. Open vSwitch を用いることで, VM と PIF 間のスイッチングを行うことが可能になる.

また Open vSwitch のトネリング機能を用いて, 仮想ネットワークを作成することができる. これにより, 既存のネットワーク構成を変更することなく, 遠隔地へのアクセスをあたかもローカルにアクセスしているかのように設定できる. Open vSwitch を用いたトネリングの構成を図 4 に示す.

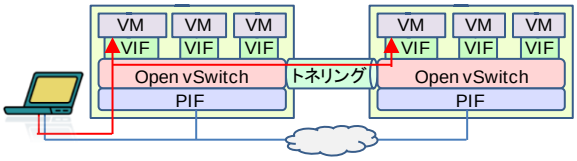

図 4Open vSwitch を用いたトネリング

4.2.5 Trema

Trema とは, コントローラを開発するためのプラットフォームを提供するオープンソースのソフトウェアである. Trema には, コントローラを実行するライブラリが含まれている. ユーザは, C 言語もしくは Ruby を用いてコントローラとなるアプリケーションを開発する. コントローラとスイッチが接続された状態で, Trema コマンドを用いコントローラを実行することで, 仮想ネットワークを構築できる.

4.3 OpenStack

OpenStack[3]は, クラウドを構成する仮想マシンや物理サーバの運用管理を実行し, それを効率的に行うためのオープンソースソフトウェアである. OpenStack の利用者(クラウド利用者)は, KVM や Xen で構成される Hypervisor 上で動作する仮想マシンに外部ネットワークからアクセスし, CPU, メモリ, HDD, IP アドレス等の計算資源を利用する.

OpenStack は複数のコンポーネントから構成され, これらのコンポーネントが連携することで IaaS のサービスを提供するアーキテクチャである. 各コンポーネントの機能は次のとおりである.

(1) Compute(Nova):「計算資源管理」「計算資源割り当て」「メッセージ通信」を行う.

(a) 計算資源管理

OpenStack が管理する物理サーバの CPU, メモリサイズ等を管理する.

(b) 計算資源割り当て

管理されている物理マシンからクラウド利用者が利用する計算資源を決定する.

(c) メッセージ通信

仮想マシンの起動, 停止等, クラウド利用者によるさまざまな制御メッセージを送受信する.

(2) Object Storage(Swift):利用可能な仮想マシンの VM イメージを保管する.

(3) Image Registry(Glance):Compute が決定した物理マシンや VM イメージの内容に基づき, VM イメージを Object Storage から読み出し物理マシンに送信する.

(4) Identity Service(Keystone):クラウド利用者やクラウド管理者の ID, パスワードを管理する. また, 個人認証

および各コンポーネントの処理内容確認を行う.

(5) Dashboard(Horizon):クラウド利用者に WebGUI を提供する.

OpenStack の構成を図 5 に示す.

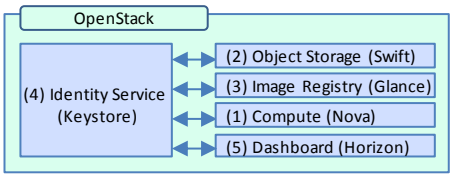

図 5 OpenStack のアーキテクチャ

5 アプローチ

本稿では OpenStack, OpenFlow を用いて, ソフトウェアによる仮想ネットワークを構築することで, 簡易なオペレーションのマイグレーションを実現するアーキテクチャを提案する.

OpenFlow を用いることで, L2 ネットワークを意識せずにマイグレーションを実現することができる. Open vSwitch 同士のトンネリングを用いた仮想ネットワークを構築し, 同一 LAN 内と同様にマイグレーションを実行することができる.

また, Quantum サーバが Open vSwitch と仮想マシンのネットワーク接続を行うことができる. そのため, 容易にマイグレーションを行うネットワークが構築できる.

6 提案アーキテクチャ

6.1 システム構成

システム構成を図 6 に示す.

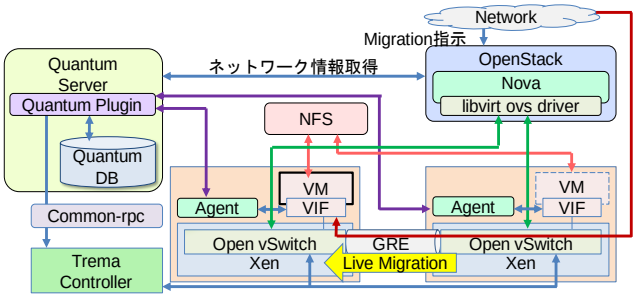

図 6 システム構成

コントローラは Quantum サーバと REST を用いて連携し, Open vSwitch を制御できる. Quantum サーバは OpenStack と連携し, 起動された仮想マシンの VIF 情報を取得, 仮想ネットワークの構成, 機器の接続を行う.

コントローラは, Quantum サーバから VIF 情報を取得する. そして, コントローラから Open vSwitch へ指示される移動先仮想マシンの VIF 情報は, OpenFlow Protocol によって定義される.

Open vSwitch はオープンソースの仮想ソフトウェアスイッチであり, トンネリングプロトコルの GRE(Graduate Record Examination)が組み込まれている. GRE トンネリングを用いて, Open vSwitch 同士でライブマイグレーションを実行する[5].

6.2 システムの挙動

システムの実行のシーケンスを図 7 に示す. OpenStack のコンポーネントである Compute がマイグレーション要求を受け取る. 次に, 移動先の物理マシンを決定し, 移動先の仮想マシンを起動する.

Quantum サーバは, 起動された Xen の VIF 情報を取得し, OpenStack が Open vSwitch のポートと VIF を接続する. コントローラは, Quantum サーバから VIF 情報を取得し, 移動先と移動元の Open vSwitch 間で GRE トンネリングを用いて, ライブマイグレーションを実行する.

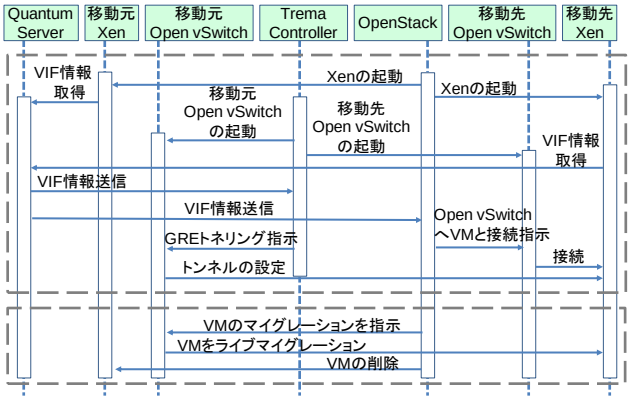

図 7 マイグレーションのシーケンス図

7 プロトタイプ

7.1 プロトタイプの構成

Xen を起動した移動元, 移動先ホストにおいて, Open vSwitch を利用する. 移動元ホストにおいてはゲストを起動する. また, Xen は NFS サーバを用いてゲスト OS のディスクデータを共有する. コントローラとなるホスト上で Trema を利用し, Open vSwitch と接続する. クライアントはゲストに接続する. プロトタイプの構成を図 8 に示す.

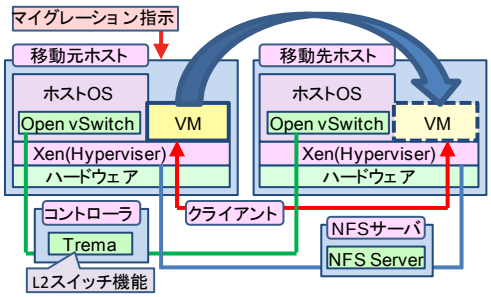

図 8 プロトタイプの構成

7.2 コントローラの構成

Trema を用い, L2 スイッチと通信量測定機能を持つコントローラを開発した. コントローラの構成を図 9 に示す.

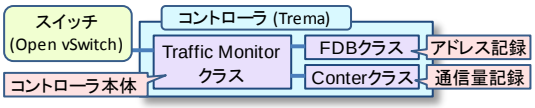

図 9 コントローラの構成

7.3 コントローラの振る舞い

このコントローラを用いた振る舞いを図 10 に示す.

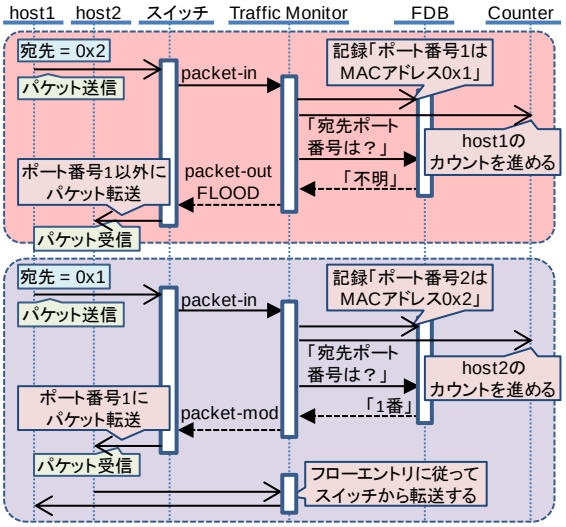

図 10 コントローラの振る舞い

7.4 プロトタイプの実行結果

プロトタイプを用いて, Trema を用いた仮想ネットワーク環境の下, VM のライブマイグレーションが行えることを確認した.

7.5 プロトタイプの実行時間

ライブマイグレーションとブロックマイグレーションの処理時間を比較した. メモリ量は 512MB として測定し, HDD はデータ量に応じて 4 通り測定した. 図 11 にその結果を示す.

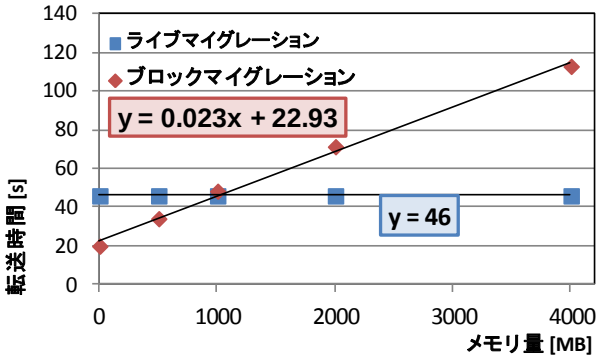

図 11 マイグレーション完了時間の比較

8 評価

ライブマイグレーション時間はメモリデータ同期のため再転送を行うが, マイグレーション中のクライアントは VM にアクセスが可能である. ブロックマイグレーションはメモリデータを一斉に転送するが, ディスクデータも転送するため時間がかかる. また, マイグレーション中, クライアントは VM にアクセスすることはできない点がある.

9 考察

本研究では OSS のみで実現する, 遠隔ライブマイグレーションアーキテクチャを提案した. そのため, ライセンス費用などをかけずに, 低コストで遠隔ライブマイグレーションを行うネットワークを構築できる. また, 従来の遠隔ライブマイグレーションアーキテクチャよりも, 短い処理時間でマイグレーションを完了することができる. さらに, ネットワークを自動構成できるため, システムの運用が容易になる.

10 今後の課題

(1) 遠隔マイグレーションの実現

Open vSwitch の機能である GRE トネリングを用いて仮想ネットワークを構築し, 遠隔ライブマイグレーションを実現する.

(2) 通信継続性を確認

遠隔マイグレーションにおけるの通信継続性を確認する. 遠隔ライブマイグレーションでは共有サーバが LAN 外に存在するため, LAN 内のみの通信よりも遅延が発生する. マイグレーション中やマイグレーション後の. 通信継続性を確認する.

11 まとめ

本研究では OSS のみを用いたアーキテクチャを提案した. Open Flow を用いて, ソフトウェアによる仮想ネットワークを構築し, Open Stack を用いて簡易なオペレーションでマイグレーションを実現する. また Quantum が OpenStack と連携することで, 遠隔マイグレーションにおけるネットワーク構成を行い, マイグレーション後でも自動的にネットワークを構築できる.

また, Open vSwitch 同士がトネリングを行いライブマイグレーションが可能になり, マイグレーション後も通信経路を確保する.

参考文献

[1] C. Clark, et al., Live Migration of Virtual Machines, Proc. NSDI '05, May 2005, pp. 273-286.

[2] 宮本 久仁男, ほか, Xen 徹底入門 第 2 版, 翔泳社, 2009.

[3] 日本 OpenStack ユーザ会, http://openstack.jp/.

[4] OpenFlow, http://www.openflow.org/.

[5] P. S. Pisa, et al., OpenFlow and Xen-Based Virtual Network Migration, Proc. IFIP Advances Information and Communication Technology, Sep. 2010, pp. 170-181.