NFV の性能に配慮したライブマイグレーション手法の検討

黄　　掣[†]　高野了成[††]　広渕崇宏[††]
市川昊平[†]　渡場康弘[†]

1. はじめに

従来専用ハードウェアで実装されてきたネットワーク機能を，汎用サーバで動作するソフトウェアとして実現する NFV(Network Functions Virtualization) が注目されている．さらに NFV を仮想マシン上で実行することで，ライブマイグレーション機能等の仮想化技術を活かした柔軟なネットワーク運用管理が可能となる．しかし，ライブマイグレーション時に生じるダウンタイムに起因するパケットロスの影響は，大量のパケットを処理する NFV では特に問題となる．本研究では，NFV のライブマイグレーション時に発生するパケットロスを削減するために，OpenFlow 技術を用いて，NFV が処理中のフローをマイグレーション先にも予めミラーリングする手法を提案する．また，評価実験によって，ライブマイグレーション中のパケット処理性能の低下を抑制でき，通信途絶時間をわずか 1 秒に抑えられたことが示せた．

2. 提案手法

VM のマイグレーションによるパケットロスを軽減するために，VM の通信に使用されていたフローをマイグレーション完了後の VM の位置合わせて素早く切り替える手法 (以降，従来手法と称する) に関する研究は，現在までに何点か報告されている．本研究は，NFV のマイグレーションにおいてさらにパケットロスを減らすため，送受信側において，VM のマイグレーション完了前に通信フローをマイグレーション元および先の両方にミラーリングする手法を提案する．具体的には，NFV のマイグレーションの開始と連動して，OpenFlow Switch のフローエントリを書き変え，マイグレーション先の新しい通信経路と元の通信経路の両方にパケットを送信させる．マイグレーションが完了したら，元の通信経路へ転送するフローエントリを削除する．

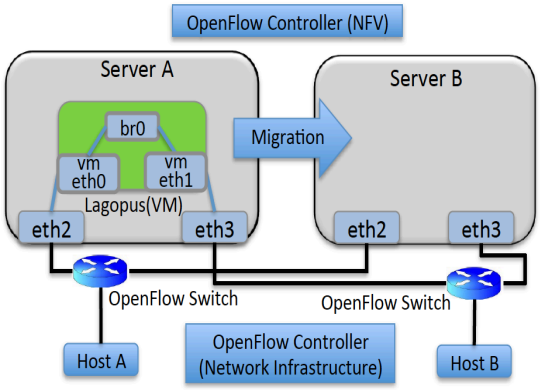

図 1　実験環境の概要図

提案手法を実現するために，本研究ではまず，OpenFlow Controller フレームワークの Ryu を利用し，REST API でフローエントリの書き換えを受け付ける簡易なコントローラを実装した．さらに，NFV のマイグレーションのタイミングに合わせ，パケットのミラーリング処理や元経路の削除を実施するプログラムを開発することで提案手法を実装した．

3. 実験環境

図 1 に実験環境の概要を示す．Server A と Server B は，NFV をホストするハイパバイザとして機能し，それぞれ eth2,3 (Intel 82599ES 10GbE) の NIC を通じて，2 つの OpenFlow Switch(Pica8 P-3780，48-Port 10GbE) に接続している．Host A と Host B は 1 つの NIC(Intel 82599ES 10GbE) を通じて，それぞれ OpenFlow Switch の 1 つと接続されている．

今回マイグレーションする NFV は Lagopus を利用した．Lagopus は intel DPDK を活用した高性能なソフトウェア OpenFlow Switch であり，NFV の実装に広く用いられると期待されている．実験では，

† 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
†† 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology

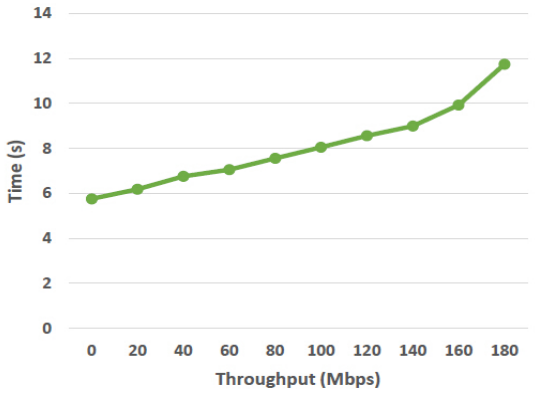

図 2 NFV の使用帯域とマイグレーションの完了時間

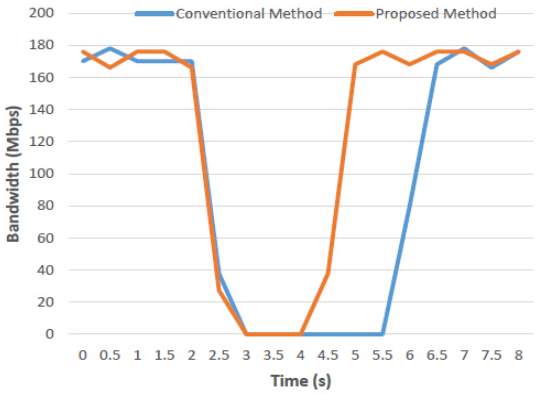

図 3 マイグレーション時における通信帯域の推移

Server A 上で動作する Lagopus をインストールした VM (6 CPU cores, 1GB Memory) を Server B にマイグレーションする際の評価を行う．マイグレーション自体の通信は Server A と B のオンボードの 1GbE NIC を使用した独立したネットワーク上で行う．図 1 に記している経路は，NFV を介して Host A および B が通信するデータプレーンのみを示している．本研究では，DPDK および Lagopus にそれぞれバージョン 1.7.1 と 0.1.2 のものを用いた．また，Lagopus の起動プションは以下の設定を用いた．

```
# lagopus -- -cf -n2 -- --rx '(0,0,1),(1,
0,1)' --tx '(0,2),(1,2)' --w 3 --bsz "(32
,32),(64,64),(32,32)" --fifoness none
```

また，本研究では 2 つの OpenFlow Controller を用いた．Networking Infrastructure OpenFlow Controller はデータプレーンを制御するコントローラで，2 つの OpenFlow Switch を制御し，本研究の提案手法を実装している．NFV OpenFlow Controller は NFV の実装を行っているコントローラで，VM 上の Lagopus Switch を制御する．今回は NFV の機能としては単純なスイッチングハブ機能を実装している．

4. 評価結果

今回の実験環境では，Server A 上で Lagopus の VM が稼動している状態で，iperf で Host A と B の間の帯域を測定したところ，平均 1Gbps ぐらいの結果が得られた．ただし，このように負荷が掛かった状態ではマイグレーションが完了することはなかった．これは NFV 内でのパケット処理によるメモリ更新頻度高すぎることが要因として考えられる．そこで，送信側に帯域制限をかけ，使用帯域を調整してみたところ，マイグレーションが完了できる使用帯域は最大およそ 180Mbps であった．使用帯域を 0Mbps から 180Mbps まで 20Mbps ずつ増やしていって，マイグレーションの完了時間を計った結果を図 2 に示す．図に示される通り，使用帯域が大きくなると，マイグレーションの完了時間が増加することが分かる．

本研究では提案手法による，パケットロスの改善とそれによる帯域の改善を評価するため，ping と iperf を用い，マイグレーション直後に経路を切り替える従来手法と，予めミラーリングを実施する提案手法を比較した．ping を用いた評価では，1 秒間隔で ping パケットを Host A から Host B に送信し続けている状況で，マイグレーションを実施し，ping 応答がロスした回数を評価した．その結果，従来手法では 3 つの応答パケットがロスしたのに対し，提案手法では重複した応答パケットが返ってきたことは確認されたが，パケットロスは確認されなかった．

また，iperf を用いた評価では，180Mbps の帯域でデータを送信し続けている状況で，マイグレーションを実施し，その間の 0.5 秒ごとの実際の使用帯域を評価した．実験結果は図 3 に示すように，従来手法では通信途絶時間がおよそ 2.5 秒であるのに対し，提案手法では通信途絶時間がわずか 1 秒程度であり，従来手法に対して，ダウンタイムを抑えた結果が得られた．

5. 今後の課題

今後は，提案手法によって具体的にどれだけのパケットロスが減っているかなど詳細な検証をしていく予定である．また，ミラーリング手法では，ネットワーク帯域を無駄に使用することや，パケットが重複して届くなどの問題があるが，これらの影響を調査する必要もある．