1H-02

マルチテナントシステムの開発を支援するIaaS環境*

松本拓也† 福田浩章‡ 芝浦工業大学 工学部情報工学科 分散ソフトウェアシステム研究室§

1. 研究の背景と目的

マルチテナントでは，サービス提供者はインフラ提供者から計算機やネットワーク等のリソースを必要に応じて獲得し，自身のサービスを利用者に提供する．現在，サービス提供者は自身のサービスに必要なリソースの最大値を見積もり，インフラ提供者のサービスを利用しているが，その決め方は経験に頼るしかなく，個々のサービスに必要十分なリソース量を正しく見積もることは極めて難しい．一方，インフラ提供者はデータセンタに大量の物理計算機とネットワークを保有し，リソースの要求に応じてオンデマンドに仮想マシンを起動して提供している．現在，データセンタの運用効率を高めるため仮想マシンのマイグレーションを利用した資源利用効率向上が注目されたおり，様々なアルゴリズムの研究開発が行われている [1]．これらアルゴリズムでは，物理計算機，仮想マシンそれぞれのリソース消費量 (e.g., CPU 負荷，メモリ使用量) をもとに，物理計算機で実行中の仮想マシンを再配置することでサービス提供者の要求を満たしつつ，データセンタ全体の運用効率向上を目指している．アルゴリズムの検証には実運用に適用することが望ましいが，サービス提供者の SLA を保証する必要があるため実現は難しい．また，大規模なデータセンタを準備することも難しいため，数台の物理計算機で基本性能を測定し，シミュレーションでの検証を行わざるをえない [2]．

そこで本研究では，RaspberryPI(RasPI) を利用し，サービス提供者，インフラ提供者それぞれの要求を容易にテストできる環境を提供する．

1.1 テスト環境の要件

本テスト環境では，サービス提供者，インフラ提供者それぞれの要求を満たす必要がある．すなわち，利用者のアクセスによって仮想マシンを複製し，スケールアウトする機能，および仮想マシン再配置の契機となる各リソース消費量の取得とマイグレーションする機能を提供する必要がある．加えて，仮想マシンのマイグレーションや複製が起こった時，利用者のリクエストを正しく移動後，あるいは複製された仮想マシンに届けるためには，ネットワークの設定も動的に変更する必要がある．

2. アプローチ

1.1 節で述べたように，本研究のテスト環境では RasPI で仮想マシンの起動，マイグレーションを行い，仮想マシンの状態に同期してネットワークを動的に変更する必要がある．そこで本研究では仮想マシンに KVM，ネットワークの変更に OpenFlow を利用する．

本研究では図 1 に示すツリー型ネットワークを想定する．このトポロジは，集中化構造により管理と維持が比較的容易なため、通信システムにおいて広く利用されている [3]．図 1 の Server1 から Server8 は全て物理計算機であり，仮想マシンを VMx(x=1,2,...)，OpenFlow に対応したスイッチ (OpenFlow スイッチ) を OVSx(x=1,2,...) とする．これらの，仮想マシンは外部からのリクエストを受け付ける必要がある．そのためには，グローバルな IP アドレスを割り当てる必要があり，物理計算機で Bridge を用いることで，物理計算機と同じネットワークの IP アドレスを割り当てる．

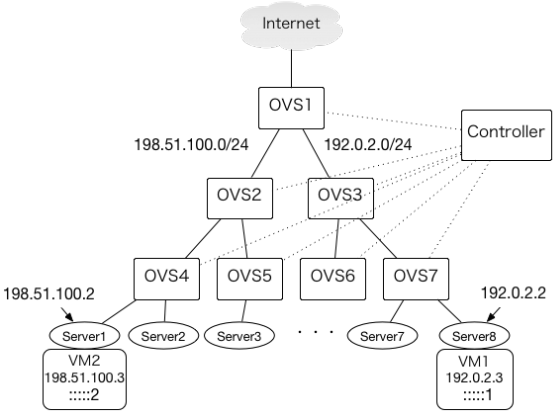

図 1: 使用したトポロジ

次にこれらを実現する時の，問題点とアプローチを述べる．

2.1 仮想マシンの起動

仮想マシンを起動するには，KVM や，Xen といったソフトウェアが必要である．したがって，本研究では KVM を用いる．RasPI 用の OS には一般的な KVM をインストールできないため，カーネルに KVM モジュールを組み込むことで KVM を実行する．

KVM で実行する OS は NFS で共有し使用するためのファイルとローカルに保存されているスナップショットを指定し起動を行う．一般に，KVM を用いた仮想マシンの管理 (起動，マイグレーション) には virt を利用する．しかし，RasPI で virt は使用することができず，代替手段が必要になる．KVM は Intel VT もしくは ARM-V により提供される CPU の仮想化を用いて CPU の割り当てを行い，QEMU と呼ばれるメモリと仮想 I/O を提供するソフトウェアを用いる．本研究では virt の代替として QEMU に付属する機能を用いて実装を行う．

2.2 マイグレーション

本研究でマイグレーションは，QEMU の機能である migration コマンドを転送先で，実行することでマイグレーションを行う．

マイグレーションは，ネットワークによる制約が存在する．通常は仮想マシンのマイグレーションは同一ネットワークに属する物理計算機へのみ行うことが可能であり，仮想マシンをそのまま移動するため，IP/MAC アドレスなどに変化はない．マイグレーションは負荷が高い物理計算機の負荷を下げるために行われるが，ネットワーク内の全ての物理計算機の負荷上がってしまった場合には，マイグレーションを行うことできなくなる．そこで，本研究では，ネットワークの構成にとらわれず，異なるネットワークへのマイグレーションを可能にするため，マイグレーション後に新たに IP アドレスを割り当てる処理を行う．

例えば，図 1 の Server8 で動作する VM1 が Server1 へマイグレーションした場合には正しいネットワークアドレスを保持する IP アドレス 198.51.100.3/24 に割り当てを行う．Bridge では，物理計算機と異なるネットワークの IP アドレスを仮想マシンに割り当てることは不可能

*Instant IaaS Environment to support multitenant system development.

†Takuya Matsumoto

‡Hiroaki Fukuda

§Shibaura Institute of Technology.

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.

であるために，このように割り当てを行う．マイグレーション発生後にルートスイッチとなる OVS(OVS1) に”宛先が VM1 であるパケットである場合，宛先を VM2 に書き換え転送”，”送信元が VM2 である場合，送信元を VM1 に書き換え転送” という 2 つのフローを書き込むことでマイグレーションに同期して構成を変更するネットワークを実現する．

2.3 負荷分散

通常ロードバランスを行う場合，予めロードバランスを行うことを想定し，ロードバランスする仮想マシンを用意し，ロードバランサを設置する．そして，ロードバランサを経由するようなネットワークを構成しなければならない．したがって，予め同じ仮想マシンを複数生成し，待機させておく必要があり無駄なリソースを消費する．

しかし，本研究ではロードバランスの要求が発生した時に，ロードバランサと仮想マシンの調達を行うことで，要求に対して即座に対応する環境を提供する．仮想マシンの調達はクローンを用いて行う．クローンは，ローカルに保存されているスナップショットを転送先に送り，転送先で，転送されたスナップショットを起動することでクローンを実現する．ロードバランサの調達は，特定の OpenFlow スイッチにロードバランサの機能を付与することで行う．

例えば，図 1 中の VM1 宛への通信をロードバランサによって分散する要求が発生した場合に，Server8 に存在する VM1 がクローンにより Server1 へ複製され VM2 として生成する．その時に，OpenFlow コントローラは，クローン元 (VM1) とクローン先 (VM2) の仮想マシンの IP アドレスとクローンを行ったという通知を受信する．その後,OpenFlow コントローラは接続している全ての OVS のフローテーブルの確認を行う．確認の結果，VM1 と VM2 の IP アドレスを比較し，異なるネットワークであれば，VM1 と VM2 間の異なるネットワークを接続している OVS1 にロードバランサの機能を付与する．従って，クローン先のネットワークによってロードバランサとなる OVS の位置が異なる．同じネットワークへのクローンを行った場合はルートスイッチとなる OVS(本環境では OVS1) にロードバランサの機能を付与する．

3. 実装・評価

3.1 実装

本研究では，OpenFlow のフレームワークとして Trema，OpenFlow スイッチとして Open vSwitch を利用し，OpenFlow コントローラとネットワーク環境の実装を行う．また，VM コントローラとして Python 言語の REPL(Read-eval-print loop) を使用し実装を行う．

3.2 評価

実際に環境を構築し，仮想マシンを起動，クローン，マイグレーションを行ったの速度の結果を表 1 に示す．表 1 に示すように，仮想マシンの起動より，クローン，マイグレーションの方が早く処理が完了した．クローンや，マイグレーションはスナップショットを転送し，転送先で中断された仮想マシンを復帰させることにより，起動するため，通常の仮想マシン起動より，処理時間が早く完了する．

また，図 1 のネットワークにおいて，実装した OpenFlow コントローラを実行し，ネットワーク速度を iperf[5] を用いて計測した．最大 TCP セグメントサイズを

表 1: 実行結果

VM 起動	クローン	マイグレーション
47.25[sec]	40.79[sec]	32.28[sec]

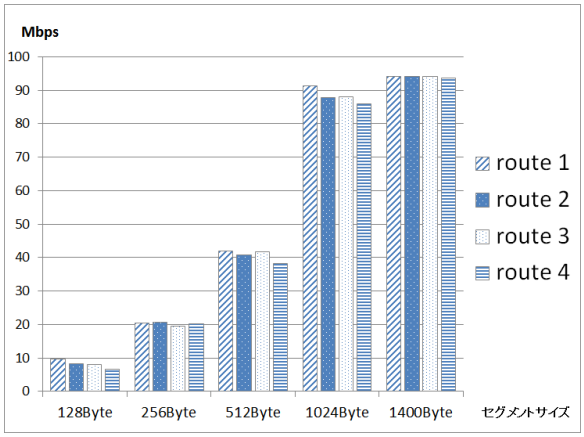

図 2: テスト結果

128Byte / 256Byte / 512Byte / 1024Byte / 1400Byte に設定し，次の表 2 に示す状況においてそれぞれ計測した結果を図 2 に示す．

表 2: route

route	通信間	経由する OVS 数
1	Server1 ⟷ Server2	1
2	Internet ⟷ Server1	3
3	Server1 ⟷ Server3	3
4	Server1 ⟷ Server8	5

route1 と route4 を比較すると分かるように，経由する OVS の数によって僅かに転送速度が低下する．しかし，これは本環境に限らず生じる．そのため，本研究において実装した OpenFlow コントローラによって発生する低下は非常に少ないと考えられ，ボトルネックになることは無いと言える．また，図 2 で示す，セグメントサイズが 1024Byte と 1400Byte にした際のトラフィック速度に大きな変化は見られないが，RasPI のイーサネットは 10/100Mbps ポートであるため 100Mbps 以上の転送速度を得ることはできない．したがって正確な値であると測定でき，仮に 1Gbps の転送速度性能を持つスイッチであれば 10 倍の転送速度を期待できる．

4. まとめ

本研究では，仮想化技術に関するアルゴリズムを容易にテスト可能な開発支援環境の提案，実装を行った．本環境はアルゴリズム開発支援環境として十分利用可能な環境であると考えられる．今後の課題として，ネットワークのトラフィック量やトポロジ，また，RasPI のリソースの状態をわかりやすく表示可能な GUI の設計などが挙げられる．

参考文献

[1] 広渕崇宏, 中田秀基, 伊藤智, 関口智嗣, ”高速マイグレーションを利用した仮想マシン配置最適化システムの検討”, 情報処理学会研究報告 IPSJ SIG Technical Report, Vol.2010-OS-115 No.14, 2010/8/4

[2] H. Kim, J. Kim and Y. B. Ko, ”Developing a cost-effective OpenFlow testbed for small-scale Software Defined Networking,” 16th International Conference on Advanced Communication Technology, Pyeongchang, 2014, pp. 758-761. doi: 10.1109/ICACT.2014.6779064

[3] 日本特許, 公開特許公報, ツリー型トポロジー・ネットワークにおける信頼性強化, 特開 2012-217112. 2012-11-8.

[4] Open Netrowking Foundation website. https://www.opennetworking.org/. 2017-01-11 参照.

[5] iperf website. https://iperf.fr/. 2017-01-11 参照.

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.