[ポスター発表] 研究報告

マルチテナント向けコンテナ集積型クラウドサービスにおけるネットワーク分離・制御方式の検討

中田 裕貴[1,a)]　松本 亮介[2,b)]　松原 克弥[1,c)]

1. コンテナ型仮想化とクラウドサービス事業における導入要件

クラウドサービス事業者が，利用者にアプリケーション実行環境（以下，実行環境）を提供する際，単一サーバ上で複数の実行環境を用意するマルチテナント方式がある．マルチテナント方式では，単一サーバ上で高集積に実行環境を提供するために，プロセス単位で実行環境を作成するコンテナ型仮想化の使用が増加している [1]．コンテナ型仮想化を用いたクラウドサービスを実現する際，サーバのリソース管理や分離のために，サーバ上に用意した仮想マシン上でクラウドサービスを提供することがある．しかし著者らは，さらなる高集積化のために，実行環境が単一 OS 上で提供されるクラウドサービスの実現を目指している．

マルチテナント方式では，Linux などの汎用 OS 上で開発された利用者のアプリケーションが，クラウドサービス上の実行環境でも変更することなく動作するように，Linux などの各種 OS 環境を提供することで利用者の利便性を確保している．また，特定のユーザがサーバ上のリソースを過剰に消費することによる，リソースに関わる動作不安定さや性能低下を防ぐために，実行環境間の分離とリソース使用の公平性を担保し，かつ，分離による実行環境の性能低下を最小限に抑えることが求められている [2]．これらの要求から，クラウドサービス事業者は，実行環境の利便性，リソース使用の公平性，実行環境の性能の 3 要件を考慮する必要がある．

コンテナ型仮想化を用いたマルチテナント方式におけるネットワーク環境では，利用者がプロセス単位での分離を意識せず，実行環境間での IP アドレスベースによる通信や，単一サーバ内の複数の実行環境において同一ポート番号の使用が求められる．そのため，利用者のアプリケーションがクラウドサービス上の実行環境でも変更することなく動作できるように，ネットワーク機能の分離が行われる．本研究では，コンテナ型仮想化と組み合わせるネットワーク分離手法について，前述した 3 要件やコンテナ型仮想化の特徴である高集積化に関する比較・検討を行う．

[1] 公立はこだて未来大学
Future University Hakodate
[2] さくらインターネット株式会社 さくらインターネット研究所
SAKURA internet Research Center, SAKURA internet Inc.
[a)] g2120032@fun.ac.jp
[b)] r-matsumoto@sakura.ad.jp
[c)] matsu@fun.ac.jp

2. 既存のネットワーク分離・制御手法と課題

コンテナ型仮想化を用いた実行環境毎にネットワーク機能を分離・制御する既存手法は，以下の 4 つに分類できる．
(1) OS のブリッジとパケットフィルタリング機能の利用
(2) 専用の制御コードを OS カーネル内で実行
(3) 軽量な OS 機能実装
(4) OS カーネルのバイパス

手法（1）は，OS が提供する仮想ネットワークデバイス（veth）を各実行環境に提供し，それらのネットワークデバイスを介する通信をブリッジとパケットフィルタ機能で制御することで公平性を担保する [3]．また，実行環境の利便性を満たすこともできる．しかし，パケット送受信時に OS カーネルのネットワークスタックを経由する回数が増加し，ネットワーク I/O 性能が低下する課題がある．

手法（2）は，手法（1）における仮想ブリッジとパケットフィルタを，専用バイトコードを用いたカーネル内制御プログラムで置き換え，OS カーネル内のネットワークスタック経由回数を削減することで，ネットワーク I/O 性能を向上している [4]．さらに，カーネル内制御プログラムでネットワークトラフィック制御を行い，ネットワークトラフィック制御における公平性も担保する．しかし，本手法では，同一サーバ内の実行環境間通信と外部との通信を区別しないため，複数コンテナを有するテナントに対するネットワークトラフィック制御の公平性担保に制限がある．

手法（3）は，OS を軽量で高速な動作が可能な専用 OS 実装に置き換え，専用 OS 内のパケット送受信機能において，DPDK などの OS カーネルをバイパスする高速パケット処理技術を使用することで，ネットワーク I/O 性能を向上している [5]．また，各実行環境とパケット送受信機能間でのパケットコピー時にトラフィックを制御することで，ネットワークトラフィック制御における公平性を担保している．しかし，本手法は，専用 OS の実行環境で動作する

ために，アプリケーションの移植を利用者に強制する必要があり，実行環境の利便性が低下する．

手法（4）は，手法（1）におけるブリッジ機能の実現にOS カーネルのバイパス技術を用いた高速パケット送受信を用いることで，ネットワーク I/O 性能を向上している [6]．また，ブリッジのネットワークトラフィック制御機能が，ネットワークトラフィック制御における公平性を担保する．しかし，本手法では，OS カーネルのバイパス技術によるパケット送受信のために特定の CPU を専有する必要があるため，各実行環境が使用可能な CPU リソースが減少し，実行環境の集積度が低下する課題がある．

コンテナ型仮想化を用いたマルチテナント方式では，前章で示した 3 要件や高集積化が重要であるが，既存手法はこれら全ての要件を満たすことができていない．

3. ハードウェア仮想化を用いたコンテナ向けネットワーク分離・制御手法の検討

本研究では，コンテナ型仮想化を用いて作成した実行環境に対して，ハードウェア仮想化技術を用いた軽量ハイパーバイザによって仮想ネットワークインタフェースを提供し，軽量ハイパーバイザ内で仮想ネットワークインタフェースに対してネットワークトラフィック制御を行う手法を提案する（図 1）．仮想ネットワークインタフェースを OS 上の機能ではなく，軽量ハイパーバイザにおいて作成し，それらをプロセス単位で分離された実行環境が使用することで，OS 上の機能を用いた分離手法と比べて OS カーネルの経由回数を削減し，ネットワーク I/O 性能を最大限維持することで実行環境の性能要件を満たす．また，仮想ネットワークインタフェースに対してハイパーバイザ内でトラフィックを制御することで，同一サーバ内の実行環境間通信と外部との通信を区別した，ネットワークトラフィック制御に関する公平性を担保できる．さらに，OS 環境への影響が少ないハイパーバイザによるネットワークトラフィック制御の実現により，実行環境の利便性の要件を満たすことができる．

本提案実現のために，軽量ハイパーバイザである BitVisor をベースに，仮想ネットワークインタフェースを複数作成する機能とネットワークトラフィック制御機能の実装を予定している．

4. おわりに

本稿では，コンテナ型仮想化を用いたマルチテナント方式のクラウドサービスで考慮すべきネットワーク性能，トラフィックの公平性，利便性の 3 要件についてまとめ，既存のネットワーク分離手法を整理した．そして，軽量ハイパーバイザによって仮想ネットワークインタフェースを提供し，仮想ネットワークインタフェースに対してハイパーバイザからトラフィックを制御する手法を提案した．

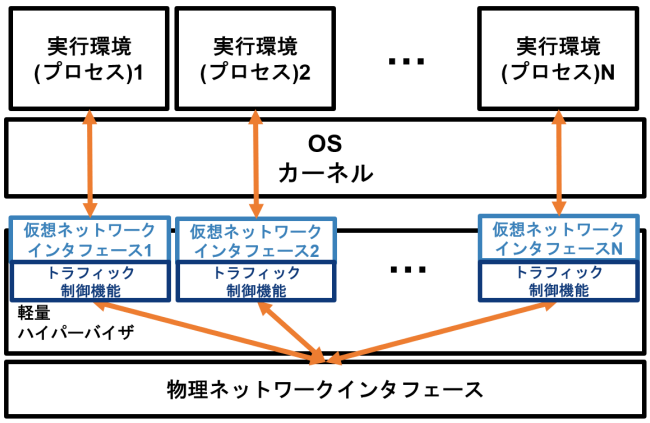

図 1　提案手法のアーキテクチャ

プロセス単位の実行環境を提供するクラウドサービス事業では，複数サーバ上に仮想マシンを作成し，その仮想マシン上においてプロセス単位で実行環境を作成することがある．この運用形態に本提案を導入した場合，二重にハイパーバイザを使用することになる．そのため，本提案手法によって発生するハードウェア仮想化技術を用いたハイパーバイザによるネットワーク性能に対する影響や，単一サーバ上でホストする実行環境の集積度への影響について検討する必要がある．また，本提案を単一サーバ内の実行環境間通信に適用した場合，ハイパーバイザでの処理がボトルネックになることが想定できるため，単一サーバ内通信に対してカーネル内パケットフィルタリングとの併用による最適化を検討したい．

参考文献

[1] Matsumoto, R., Kondo, U. and Kuribayashi, K.: Fast-Container: A Homeostatic System Architecture High-Speed Adapting Execution Environment Changes, *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1, pp. 270–275 (2019).

[2] Jing Zhu, Dan Li, Jianping Wu, Hongnan Liu, Ying Zhang and Jingcheng Zhang: Towards bandwidth guarantee in multi-tenancy cloud computing networks, *2012 20th IEEE International Conference on Network Protocols (ICNP)*, pp. 1–10 (2012).

[3] Zhao, Y., Xia, N., Tian, C., Li, B., Tang, Y., Wang, Y., Zhang, G., Li, R. and Liu, A. X.: Performance of Container Networking Technologies, *Proceedings of the Workshop on Hot Topics in Container Networking and Networked Systems*, HotConNet '17, Association for Computing Machinery, p. 16 (2017).

[4] Cilium: cilium/cilium: eBPF-based Networking, Security, and Observability, `https://github.com/cilium/cilium` ((Accessed on 08/24/2020)).

[5] Ren, Y., Liu, G., Nitu, V., Shao, W., Kennedy, R., Parmer, G., Wood, T. and Tchana, A.: Fine-Grained Isolation for Scalable, Dynamic, Multi-tenant Edge Clouds, *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, USENIX Association, pp. 927–942 (2020).

[6] Yang, M. and Huang, Y.: OVS-DPDK with TSO feature running under docker, *2018 International Conference on Information Networking (ICOIN)*, pp. 270–273 (2018).