

仮想マシン収容効率向上のためのI/O仮想化技術の利用に関する考察

辻 聡[†] 孫 雷[†] 狩野 秀一[†]

[†] NEC クラウドシステム研究所 〒211-8666 神奈川県川崎市中原区下沼部 1753

E-mail: †{a-tsuji@bq,l-sun@ap,karino@da}.jp.nec.com

あらまし 継続的な CPU コア数の増加は、1 台のサーバに収容可能な仮想マシン数の増加につながっており、将来的には 1 台のサーバに 100 台以上の仮想マシンを収容可能になると予測される。しかしながら、XenServer や KVM などの仮想化環境で仮想マシンとの通信に用いられる仮想スイッチがボトルネックとなり、仮想マシンの収容数が制限されると考えられる。我々がこれまで提案・実装してきた FlexivOffload 技術は、この課題を性能とトラフィック管理の両立を実現しつつ解決可能である。本稿では、仮想化環境のネットワーク I/O 性能を向上させる様々な技術と FlexivOffload 技術の比較を通して、仮想化環境で I/O 仮想化技術を利用する際の課題について述べる。

キーワード 仮想スイッチ, I/O 仮想化技術, データセンター

A Study on Utilizing I/O Virtualization Technology for Improvement of Virtual Machine Density on a Single Server

Akira TSUJI[†], Lei SUN[†], and Shuichi KARINO[†]

[†] Cloud System Research Laboratories, NEC Corporation 1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211-8666 Japan

E-mail: †{a-tsuji@bq,l-sun@ap,karino@da}.jp.nec.com

Abstract The number of virtual machines in a single server is increasing along with continuous increase of the number of CPU cores. In the future, more than 100 virtual machines are expected to be installed into a single server. However, the virtual switch used for packet handling in virtualization environments like XenServer and KVM will be likely the performance bottleneck, which limits the number of virtual machines in a single server. FlexivOffload technology which we have proposed and implemented can resolve this problem with satisfying both performance and traffic management. In this paper, we discuss about the problems which occur when I/O virtualization technology is used in virtualization environments through the comparison between FlexivOffload technology and other technologies which can improve network I/O performance in those environments.

Key words vSwitch, I/O virtualization technology, Data center

1. はじめに

汎用 CPU のコア数の増加に伴って、1 台のサーバに収容可能な仮想マシンの数は増加の一途をたどっている。図 1 は、Intel Xeon について、主だった製品のコア数の変遷を示したものである。Xeon のコア数は年率 2 コア程度で増加しており、現行の Sandy Bridge 世代では最大 8 コア/16 スレッドのプロセッサを複数搭載可能になっている。その結果、仮想マシンに 1 スレッド割り当てる場合に、現在でもサーバあたり数十台の仮想マシンを、2018 年ごろには 100 台以上の仮想マシンを収容可能になることが予想される。つまり、サーバ仮想化技術を用いない場合の 2~3 ラック分のサーバが 1 台のサーバに統合され

ることになる。

1 台のサーバに収容可能な仮想マシン数が増加すると、必要とする機器の数量、機器を設置する建物の規模などを小さくすることができ、CAPEX(Capital Expenditures)の削減につながる。例えば、100 万台の仮想マシンをデータセンタに収容する場合、100 台の仮想マシンを収容可能なサーバを用いると 1 万台、20 台の仮想マシンを収容可能なサーバの場合は 5 万台必要となる。これらのサーバが 1U のサイズであり、1 ラックあたり 40 台のサーバを装着する場合、必要なラック数はそれぞれ 250 台、1250 台であり、サーバの設置面積が 80%改善する。

しかしながら、このような状況においては、サーバ仮想化環

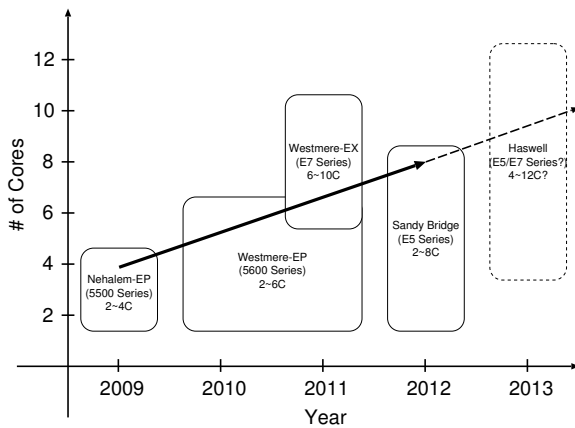


図1 Intel Xeonのコア数の変遷

境の管理領域 (VMM: Virtual Machine Manager)^(注1)の負荷が高くなり、VMMがボトルネックとなって、1台のサーバに多くの仮想マシンを収容できなくなると考えられる。仮想マシンに係るトラフィックを中継するために、VMMには一般的に、ソフトウェアで実現されたスイッチ (仮想スイッチ, vSwitch) が備えられている。この仮想スイッチのパケット処理の負荷が高く、VMMがボトルネックとなる。

そこで、VMMのボトルネックを解消するために、CPUやチップセット、NIC(Network Interface Card)がサポートするI/O仮想化技術を利用することが考えられる。CPUやチップセットがサポートするIOMMU(I/O Memory Management Unit) [1] や、NICがサポートするSR-IOV(Single Root I/O Virtualization) [2] などの技術により、仮想マシンがVMMを経由せずに通信することができ、結果としてVMMがボトルネックとなることを防ぐことができる。

しかし、このようなI/O仮想化技術を利用すると、VMMでトラフィック管理ができないという課題が発生する。サーバの外部に設置された物理スイッチにてモニタリングのための統計情報収集といったC-plane処理を行うことも考えられるが、一般に、物理スイッチでC-plane処理に利用されるCPUは性能が低い[3]、物理スイッチのC-plane処理の負荷がネットワークの運用に影響を与えると考えられる。現在、SDN(Software-Defined Network)の考え方に基づいたネットワーク制御が広まりつつあるが、それを実現する技術の1つであるOpenFlow [4] では、物理スイッチのC-plane処理がネットワークのパフォーマンスに大きな影響を与える [5] [6]。

本稿では仮想マシンの収容効率を向上させるためにI/O仮想化技術を利用することについて、我々がこれまで提案・実装してきたFlexivOffload技術の評価結果をもとに考察する。

以下、2.節では仮想スイッチの概要と、仮想スイッチを利用することの課題について予備評価を交えつつ述べる。3.節で仮想化環境のネットワークI/O性能を向上させる技術について触れる。4.節でFlexivOffload技術の概要について述べ、5.節でその性能評価の結果を示す。そして、6.節でFlexivOffload技術の課題について議論し、7.節でまとめる。

(注1): KVMのホストOS、XenServerのハイパーバイザと管理用仮想マシン(Dom0)など。

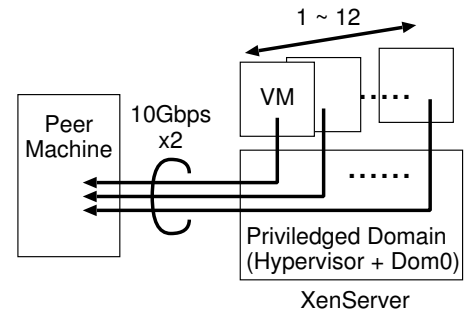


図3 評価環境概観

表1 評価環境

	Peer Machine	Target Machine	DomU
CPU	Intel Xeon X5650 2.67GHz×2	← 1 core	
	HyperThreading=disable		
Memory	24GB	1GB	
NIC	Solarflare Solarstorm SFN5122F [12]		
OS	CentOS 6.0	XenServer 6.0	CentOS 6.0
kernel	2.6.32-71.el6.x86_64	2.6.32.12-0.7.1.xs 6.0.0.529.170661 xen	2.6.32-71.el6. i686 xen_pv_ hvm=enable
vSwitch		Open vSwitch-1.2.2	

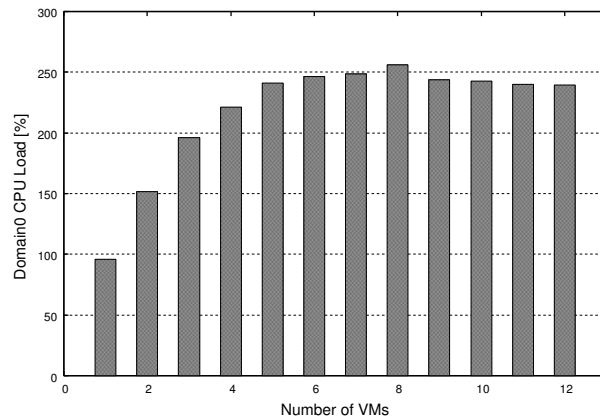
2. 仮想スイッチ

サーバ仮想化環境では、仮想スイッチとしてLinux BridgeやOpen vSwitch [7]のようなOSSのほか、VMware vSphere Distributed Switch [8]のような商用製品 [9] [10] [11] が利用されている。

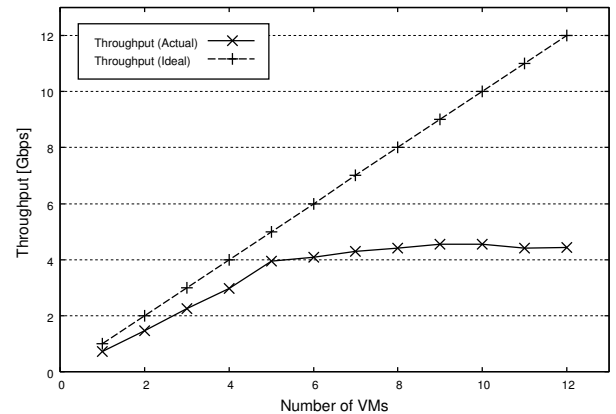
1.節で述べた通り、仮想スイッチにおけるパケット処理が通信のボトルネックとなりうるという課題も抱えているものの、仮想スイッチはソフトウェアで実現されているがゆえに、機能の追加を柔軟に行うことができ、また、新しいプロトコルへの対応は比較的早期に行われるという利点がある。

図2は、仮想マシンに1Gbpsの帯域を割り当て、仮想マシン数を増加させていった場合のVMMのCPU負荷と総スループットを測定した結果を示したものである。図2(b)には、総スループットの理想値 (仮想マシン数×1Gbps)を合わせて示している。評価環境の概観を図3に、評価環境を表1に示す。サーバ仮想化環境にXenServerを用いており、XenServerを導入したサーバ(Target Machine)と対向のマシン(Peer Machine)を、10GbEのリンクを2本用いて、Back-to-Backで接続している。評価用プログラムにiperfを用い、各リンクに流れるトラフィックが均等になるようにして評価を行った。VMMには4つのCPUコア、各仮想マシンには1つのCPUコアを割り当てているが、CPUコアを占有する設定は行っていない。

VMMのCPU負荷は仮想マシンが増加するに従って高くなり、最大で250%程度まで達する。すなわち、複数のCPUコアを用いて処理を行っている。複数のCPUコアを利用しているにもかかわらず、総スループットは最大で4.5Gbps程度に留まっており、各仮想マシンに1Gbpsの帯域を割り当てようとした場合に、高々4台の仮想マシンしか1台のサーバに収容することができないことを示している。つまり、CPUのコア数



(a) Dom0 CPU Load



(b) Total Throughput

図2 vSwitch のスケーラビリティ

がいくら増加したとしても、仮想マシンの収容数を増やすことができず、サーバ仮想化技術の利点を損なってしまうことを示している。今後の CPU コア数の増加に期待し、VMM により多くの CPU コアを割り当てて VMM の CPU リソースに余裕を持たせることも考えられるが、その場合、仮想マシンに割り当てることのできる CPU リソースが減少し、やはり、1 台のサーバに導入可能な仮想マシン数の減少を招く。

この問題点は、最終的にはデータセンタに設置する機器の増加、設置面積の増加につながり、サーバ仮想化による CAPEX の削減幅が小さくなってしまふ。

3. サーバ仮想化環境ネットワーク I/O 性能向上技術

サーバ仮想化環境のネットワーク I/O 性能向上技術は大きく分けて 2 通り存在する。VMM 内でのパケット処理を高速化する方法と、VMM そのものをバイパスしてしまう方法である。

VMM 内のパケット処理を高速化する技術としては、汎用 NIC のアクセラレーション機能を利用した STT(Stateless Transport Tunneling) [13] や、VMQ(Virtual Machine Queue) [14] が存在する。さらに、専用のアクセラレーションカードを用いてネットワーク処理を高速化する手法も存在する [15]。

STT は Nicira Networks(現 VMware, Inc.) が提案するトンネリング技術であり、ユーザの L2 パケットを TCP パケットと同一のパケットフォーマットのヘッダでカプセル化し、L2 over L3 ネットワークを構築する。STT は、パケットフォーマットが TCP パケットと同一であるため、NIC が備える TSO(TCP Segmentation Offload) や LSO(Large Send Offload) などのセグメンテーションオフロードエンジンを利用することができ、これによりネットワーク I/O 性能の向上を実現する。仮想マシンや仮想スイッチが扱うパケットのセグメントサイズを大きくし、NIC のハードウェア機能を利用して複数のセグメントに分割後、ネットワークにパケットを送出する。これにより、仮想マシンや仮想スイッチが扱うパケット数が減り、結果としてパケット処理に要するオーバーヘッドが削減される。STT により、10Gbps クラスのネットワークを用いた仮想化環境で、ほぼラインレートに近いスループットを達成することができる [16]。

VMQ は Microsoft のサーバ仮想化環境である Hyper-V 向

けの技術であり、VMQ に対応した NIC を用いることで利用可能である。VMQ はパケット受信処理を高速化する技術であり、仮想スイッチで実行するパケットのフィルタリングや仮想マシンへのルーティング処理をホスト CPU から NIC へオフロードする。これにより、パケットデータのコピー回数を削減することができ、高いスループットを実現することができる [17]。

文献 [15] ではネットワークプロセッサを搭載したプログラマブルな NIC を用いて、OpenFlow [4] の機能をホストから NIC にオフローディングする手法を提案している。パケットのルーティングや、パケットに対するアクションの実行、フローのモニタリングを NIC 側で実行し、ホスト CPU の負荷を低減する。

しかしながら、これらの技術では VMM 側で多少なりともパケット処理を実行するため、仮想マシン数の増加、あるいはネットワークの高帯域化によって仮想スイッチを通過するパケット数が増加した際に CPU 負荷が高くなる可能性がある [16]。

VMM をバイパスする技術として、VEPA(Virtual Port Ethernet Aggregator) [18] や NIV(Network Interface Virtualization) [19] が存在する。

これらの技術は、NIC に備えられた I/O 仮想化支援機構を利用する。これによって、仮想マシンから NIC に直接アクセス可能にし、パケットの送受信に際して VMM を経由しないようにする。単に I/O 仮想化機能を利用しただけでは、VMM からはトラフィックが見えなくなるため、仮想マシンのトラフィックを管理できなくなる。そこで、VEPA や NIV では同一サーバ内の仮想マシン間の通信を含む、仮想マシンに係る全ての通信を NIC を通じて物理ネットワーク側に出し、サーバに隣接する物理スイッチでトラフィックを制御する。そのため、C-plane と D-plane の両処理は従来通り物理スイッチで行う。

そのため、これらの技術と SDN を組み合わせようとする、1. 節で述べたように、物理スイッチの C-plane 処理がボトルネックとなり、ネットワークの管理・運用に問題が発生する可能性がある。文献 [5] では、OpenFlow ネットワークにおいて新規フローの発生頻度と、それを処理する物理スイッチの CPU 使用率を測定しており、フローの発生頻度が高くなるほど、物理スイッチの CPU 負荷は高くなることを報告している、物理スイッチに接続される仮想マシン数が増加すると、そのスイッチが処理する新規フローの発生頻度が上がるため、物理スイッ

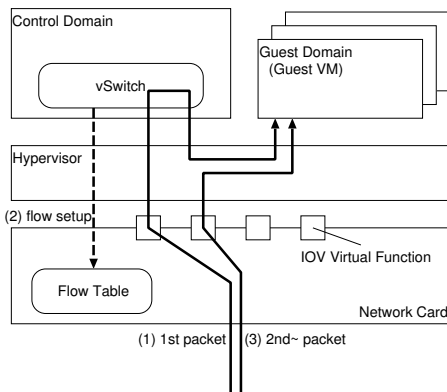


図 4 FlexivOffload 技術 概観

チの制約により、サーバに収容可能な仮想マシン数に制限がかかると思われる。

4. FlexivOffload

我々はこれまで、VMM、及び物理スイッチの負荷の低減を両立する仮想化環境ネットワーク I/O 性能向上技術として、仮想マシンの通信をフロー単位で、NIC の I/O 仮想化機能を利用して行うかどうかを動的に制御する FlexivOffload 技術を提案し、その効果を実証してきた [20] [21].

FlexivOffload 技術の概観を図 4 に示す。図 4 では、ネットワークから仮想マシンに向けてパケットが流れている様子を示している。ここでは、サーバ仮想化環境のアーキテクチャを XenServer に似せて記載しているが、本技術はサーバ仮想化環境の特定のアーキテクチャに限定されない。我々は XenServer を対象としてプロトタイプの実装を行い [21]、ここで得た知見を元に、情報通信研究機構が主導して研究開発を進めている、ネットワーク仮想化基盤 [22] のネットワーク処理計算ノード (プログラマ) のネットワーク I/O 改善技術として、KVM 環境向けに実装を進めている [23].

本技術は、VMM 側に備えられたフロー単位の制御が可能な仮想スイッチと、I/O 仮想化機能とフローテーブルを持つ NIC を利用する。

仮想スイッチ側からフロー単位で NIC のフローテーブルにエントリを登録し、I/O 仮想化機能を利用して仮想マシンに直接パケットを配送 (オフロード) するか、仮想スイッチ経由で配送 (オンロード) するかを制御する。フローテーブルにはあらかじめエントリを追加しておくことも、フロー発生時にエントリを追加することも可能である。後者の場合、フローテーブルにエントリを登録する方式として、パケットヘッダの情報を条件とし、条件に一致するフローを登録する方式と、フローの統計情報 (通過パケット数、通過バイト数、スループットなど) を条件とし、条件を満たしたフローを登録する方式の二通りを採用している。これにより、新規フローが発生した時点でのオフロードの判定と、フローが発生した後のフローの状態に応じたオフロードの判定が可能になっている。例えば、フロー発生時にパケットヘッダの情報からオフロードすると判定されたフローに対して、そのフローで単位時間に流れるパケット数が少なくなった場合に、オンロードに変更することが可能である。

オフロードされたフローを管理するために、VMM はオフ

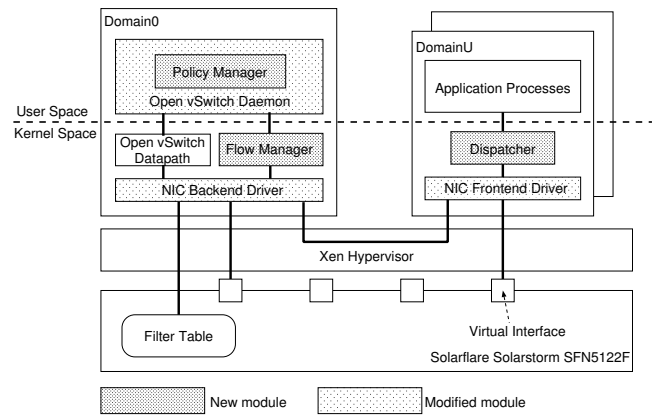


図 5 FlexivOffload 技術の XenServer 向け実装

表 2 FlexivOffload 技術の XenServer 向け実装 諸元

仮想化環境	XenServer 6.0
Open vSwitch	1.2.2
仮想マシン	CentOS 6.0 (i686)
NIC	Solarflare Solarstorm SFN5122F
NIC ドライバ (VMM)	sfc-3.2.1.6099-1
NIC ドライバ (仮想マシン)	sfc-xnap-netfront-v1.0.0.0012

ロードされたフローの統計情報を NIC または VM から取得可能である。

このように、FlexivOffload 技術では、VMM の制御の元で I/O 仮想化技術を利用するかどうかを制御し、オフロードされたフローの統計情報も VMM が管理・収集するため、フロー単位のトラフィック管理を実現しつつ、高い性能を達成可能である。また、サーバに搭載される CPU は物理スイッチのものよりも高性能であり、さらに、VMM はローカルのサーバに収容されている仮想マシンについてのみ管理すればよいため、サーバに隣接する物理スイッチよりも C-plane の処理負荷は低いと考えられる。結果として、サーバに隣接する物理スイッチの C-plane 処理の負荷、及び仮想スイッチの D-plane 処理による VMM の負荷の双方を低減することができる。

5. XenServer 環境向け FlexivOffload の実装、及び評価

本節では、XenServer 環境向けに実装した FlexivOffload 技術の概要と、その性能評価について述べる。

5.1 実装

図 5 に XenServer を対象の仮想化環境とした FlexivOffload 技術の概観を、表 2 に諸元を示す。

Open vSwitch の拡張モジュールとして、オフロード対象のフローを定めたポリシーを管理する Policy Manager と、全フローの状態を管理する Flow Manager を VMM に実装した。

また、仮想マシンがソースとなるトラフィックに対してオフロードの制御をするために、Dispatcher を各仮想マシンに配置している。Dispatcher は内部にフローテーブルを持っており、Policy Manager からの指示に従ってフローエントリの追加/削除を行い、このフローテーブルに従って、フローをオフロードさせるかどうか判定する。この判定に基づいて、NIC ドライバ (NIC Frontend Driver) が、NIC または VMM に対してパケッ

トを送出する。

実装の詳細、及び動作については文献 [21] が詳しい。文献 [21] の実装との大きな差異は、SR-IOV に対応した点である。

5.2 性能・機能評価

本技術が、VMM のトラフィック管理の機能を損なうことなく、VMM の負荷低減と仮想マシンの I/O 性能の向上を実現できることを示すために、2. 節と同一の評価環境、評価手法を用いて評価を行った。性能測定用のトラフィックはフロー発生時にはオフロードされず、ファーストパケットが VMM で判定処理され、後続のパケットがオフロードされるように設定を行った。

図 6 に評価結果を示す。図 6 に示されるオンロードの結果は図 2 と同一である。図 6(a) に示される通り、VMM の負荷はオンロード時に比べ、大幅に低減しており、また、仮想マシン数が増加してもほぼ一定で 10~15% 程度である。さらに、スループットについても、各仮想マシンにほぼ 1Gbps の帯域を提供できており、仮想マシン数が 12 の場合であっても、理想的な場合の約 95% の性能を示している。これらのことから、本技術が仮想マシン数に対してスケラビリティに優れることが分かる。

図 7 は、フローのモニタリング結果を示したものである。Open vSwitch の制御コマンドを拡張し、オフロードされたフローとされていないフローの双方の統計情報を表示している。VMM で管理されているオフロードされたフローの統計情報を利用することにより、オフロードされたフロー、すなわち、I/O 仮想化技術を利用しているフローについてもモニタリングの機能を提供できることを示している。Open vSwitch が備えるスイッチの機能は豊富であり、また、機能の追加が柔軟に行えるため、物理スイッチに比べて機能が不足するということは起こりにくい。そのため、VMM で収集したトラフィックの統計情報を元に、Open vSwitch で様々な C-plane 処理を実現することで、所望のトラフィック管理を実現可能であると考えられる。

これらの結果から、本技術が VMM でトラフィック管理を行いつつ、VMM の負荷の低減と仮想マシンの I/O 性能向上を実現することが可能なことが示された。

6. 考 察

FlexivOffload 技術が性能とトラフィック管理の両立を実現できることは示されたが、汎用性の面で課題を抱えている。

フロー単位でトラフィックを制御するためには、NIC にてフローを識別できる必要がある。いくつかの NIC では、MAC アドレスと VLAN によりパケットを仮想インタフェースに振り分ける機能を持っているが [24]、L3 以上のレイヤの情報で振り分ける機能を持つ NIC は少ない。一方で、STT で利用している TSO や LSO はデータセンタで用いられるような NIC では広く搭載されている。

また、本報告では I/O 仮想化機能の利用を制御することで通信性能が改善することは示したが、本技術を実際に使用する場合には、オフロード時のレート制御も必須である。仮想マシンに仮想インタフェースを割り当てただけでは、そのインタフェースの最大レートまで利用可能になってしまい、特定の仮想マシンのトラフィックにより、他の仮想マシンのトラフィックが影響を受ける可能性があるためである。文献 [25] では Intel 製ネットワークコントローラ [24] [26] の機能を利用して、VMM 側の

ドライバ経由で仮想インタフェースごとの送信レートを制御することで、仮想化環境で SR-IOV を利用しつつ QoS 制御を実現している。このように、技術的には QoS 制御は可能ではあるが、同様の機能を備える NIC であっても設定方法が異なっている場合があり、NIC ごとの制御モジュールを FlexivOffload 技術の制御ソフトウェアに組み込むのは開発コストが大きい。そのため、異なる NIC ベンダ間の NIC で機能の制御を行う共通インタフェースが必要と考えられる。

7. ま と め

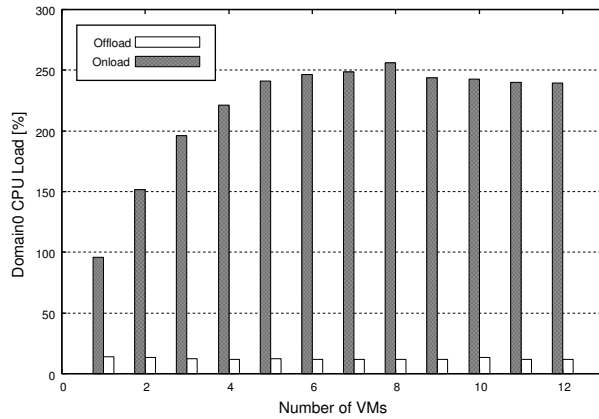
本稿ではサーバに搭載される CPU コアの継続的な増加により、1 台のサーバへの仮想マシンの収容可能数が改善した場合に、サーバ仮想化環境のネットワーク I/O 性能がボトルネックとなり、仮想マシンの収容数に課題が発生しうることを提起した。そして、この課題を我々がこれまで提案してきた FlexivOffload 技術で解決可能なことを示した。

今後の課題として、STT、VMQ などの関連技術の性能評価による、本稿で指摘したこれらの技術が抱える課題の実証、FlexivOffload 技術の汎用化が必要である。

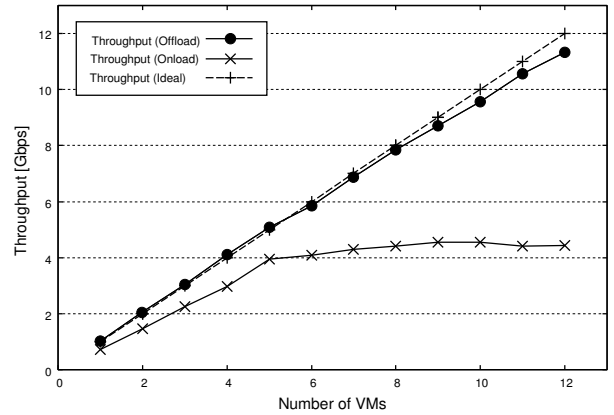
謝辞 本研究の成果の一部は、独立行政法人情報通信研究機構 (NICT) の委託研究「新世代ネットワークを支えるネットワーク仮想化基盤技術の研究開発」により得られたものである。

文 献

- [1] Intel, "Intel® Virtualization Technology for Directed I/O," 2011. Available at [http://download.intel.com/technology/computing/vptech/Intel\(r\).VT_for_Direct_IO.pdf](http://download.intel.com/technology/computing/vptech/Intel(r).VT_for_Direct_IO.pdf).
- [2] PCI-SIG I/O Virtualization, "<http://www.pcisig.com/specifications/iov/>".
- [3] Guohan Lu, Rui Miao, Yongqiang Xiong, and Chuanxiong Guo, "Using CPU as a Traffic Co-processing Unit in Commodity Switches," Proceedings of the first workshop on Hot topics in Software Defined Networks, pp.31-36, HotSDN '12, 2012.
- [4] Open Networking Foundation, "<https://www.opennetworking.org/>".
- [5] Dan Levin, Andreas Wundsam, Anja Feldmann, Srini Seetharaman, Masayoshi Kobayashi, and Guru Parulkar, "A First Look at OpenFlow Control Plane Behavior from a Test Deployment," Technical report, Technische Universität Berlin, Fakultät Elektrotechnik und Informatik, 2011.
- [6] Charalampos Rotsos, Nadi Sarrar, Steve Uhlig, Rob Sherwood, and Andrew W. Moore, "OFLOPS: An Open Framework for OpenFlow Switch Evaluation," Proceedings of the 13th International Conference on Passive and Active Measurement, pp.85-95, PAM'12, 2012.
- [7] Open vSwitch, "<http://openvswitch.org/>".
- [8] VMware, Inc., "VMware vSphere Distributed Switch (VDS)". <http://www.vmware.com/products/datacenter-virtualization/vsphere/distributed-switch.html>.
- [9] Cisco Systems, Inc., "Cisco Nexus 1000V Series Switches". <http://www.cisco.com/en/US/products/ps9902/>.
- [10] Microsoft, "Hyper-V Virtual Switch Overview". <http://technet.microsoft.com/en-us/library/hh831452.aspx>.
- [11] NEC Corporation, "NEC, Microsoft® Windows Server® 2012 Hyper-V® 仮想スイッチを OpenFlow に対応させる拡張ソフトウェアを開発". http://jpn.nec.com/press/201209/20120905_02.html.
- [12] Solarflare Communications, Inc., "Solarflare SFN5122F Dual-Port 10G Ethernet Enterprise Server Adapter". Available at http://www.solarflare.com/products/Solarflare_



(a) Dom0 CPU Load



(b) Total Throughput

図 6 FlexivOffload 技術のスケラビリティ

Every 1.0s: ovs-dpctl dump-flows xenbr2

Mon Oct 22 14:37:17 2012

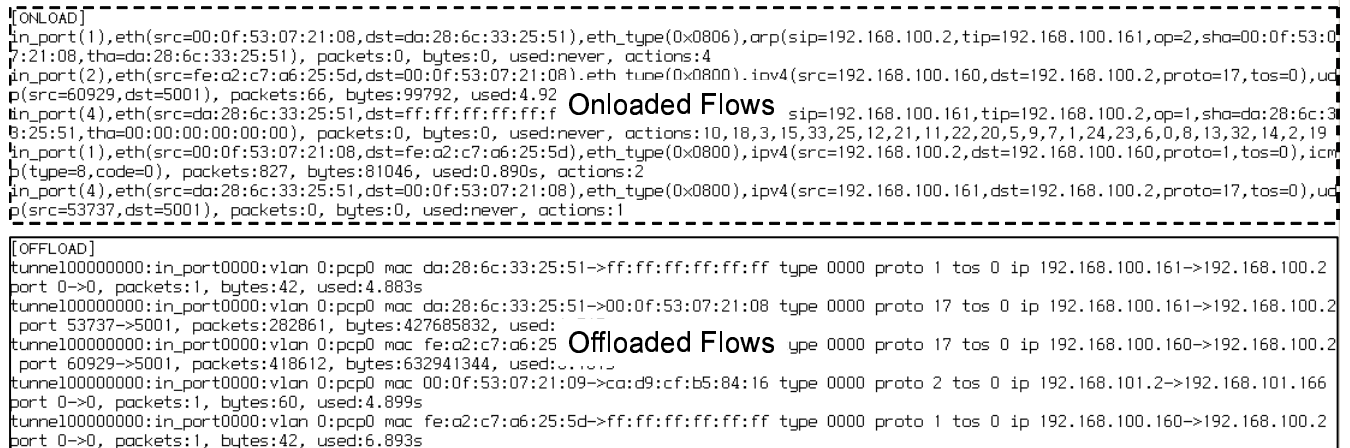


図 7 フローのモニタリング

10GbE_NIC_SFN5122F_v011711.pdf.

[13] A Stateless Transport Tunneling Protocol for Network Virtualization (STT) draft-davie-stt-02, "http://tools.ietf.org/html/draft-davie-stt-02".

[14] Using Virtual Machine Queue, "http://technet.microsoft.com/ja-jp/library/gg162704(v=WS.10).aspx".

[15] Yan Luo, Eric Murray, and Timothy L. Ficara, "Accelerated Virtual Switching with Programmable NICs for Scalable Data Center Networking," Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures (VISA'10), pp.65-72, Sep. 2010.

[16] Network Heresy, "The Overhead of Software Tunneling". <http://networkheresy.com/2012/06/08/the-overhead-of-software-tunneling/>.

[17] Intel Corporation and Microsoft Corporation, "Advanced Virtualization I/O Queuing Technologies - An Intel-Microsoft Perspective," May. 2009. Available at <http://download.intel.com/network/connectivity/virtualization/321968.pdf>.

[18] Blade Network Technologies, Inc. et.al, "Standardizing Data Center Server-Network Edge Virtualization," Oct. 2010.

[19] Cisco Systems, Inc., "Cisco VN-Link:仮想化対応ネットワークング," 2009. Available at http://www.cisco.com/web/JP/solution/places/datacenter/literature/pdf/white_paper_c11-525307.pdf.

[20] 狩野 秀一, "vswitch 処理の動的オフロード方式の提案," 2010 年電子情報通信学会総合大会, vol.ISSN 1349-1377, no.B-6-45, p.45, Mar. 2010.

[21] 辻 聡, 飯屋 貴裕, 狩野 秀一, "vswitch 処理の動的オフロード方式の実装と評価," 電子情報通信学会技術研究報告 NS2011-29, vol.Vol.111, no.No.43, pp.69-74, May. 2011.

[22] 中尾 彰宏, "新世代ネットワーク構想におけるネットワーク仮想化," 電子情報通信学会誌, vol.Vol.94, no.No.5, pp.385-390, May. 2011.

[23] 神谷 聡史, 狩野 秀一, 孫 雷, 百目木 智康, 河邊 岳彦, "進化したネットワーク仮想化基盤を実現するネットワーク処理計算ノードの提案," Mar. 2012. Available at <http://www.ieice.org/~nv/09-nv20120302-kamiya.pdf>.

[24] Intel Corporation, "Intel®82599 10 Gigabit Ethernet Controller Datasheet Revision:2.75," Apr. 2012. Available at <http://www.intel.com/content/dam/doc/datasheet/82599-10-gbe-controller-datasheet.pdf>.

[25] 高野 了成, 工藤 知宏, 児玉 祐悦, "仮想計算機モニタ・バイパス型ネットワークに対する通信制御方式," 情報処理学会計算機システムソフトウェアとオペレーティング・システム研究会, vol.Vol.2011-OS-117, no.No.13, pp.1-6, Apr. 2011.

[26] Intel Corporation, "Intel®82576EB Gigabit Ethernet Controller Datasheet Revision:2.63," Dec. 2011. Available at <http://download.intel.com/design/network/datashts/82576-Datasheet.pdf>.