

環境に対して自動最適化する高性能通信基盤の開発 — 高性能で自動チューニングされる仮想ネットワーク —

1. 背景

汎用PCの高性能化に伴い、従来では専用のハードウェアを用いて実現されていたルータやFirewall等のネットワーク機能 (Network Function) は汎用のIAサーバで実現可能になった。このように、ネットワーク機能をソフトウェアや仮想化技術等を用いて実現することをNetwork Functions Virtualization (NFV) と呼ぶ。ソフトウェアを用いて実現されたネットワーク機能をVirtual Network Function (VNF) と呼び、ソフトウェアの利点により、開発コスト低下、運用コスト低下等のいくつかの利点が存在する。

NFVを10/40GbE以上で実現する場合、DataPlaneDevelopmentKit (DPDK) 等の独自の packets IO フレームワークを用いて、ネットワーク機能を実現することが定石であるが、これらを用いた開発のコストは高く、NUMAやキャッシュを意識するなど、コンピュータ理論に精通する必要がある。また、DPDKを用いたルータ等のネットワーク機能は「N個のコアを使い、合計40Gbpsを実現する」といったように、コアの割り当てを開発時に指定し、動作させるハードウェアに対してFineTuneした状態で開発されることが多い。しかし、インターネットのトラフィックは1日の間に一定ではなく、時間や様々な要因によって、流動的に変化するものである。

既存のNFVでは10/40GbE以上の高性能なVNFを実現しようとする段階であるが、これはサーバのリソース (CPUコア) を多く消費し、トラフィックに対して最適にリソースを消費しているとは言い難いという問題がある。トラフィックが流れていないルータの対して与えているリソースを一時的にトラフィックが飽和しているルータに貸し出すことで、インターネットのトラフィック処理の効率化を行うことが可能である。

2. 目的

本プロジェクトでは10/40GbE以上の高性能なVNFの開発を支援し、それらを自動で最適化する仮想ネットワークの実現を目的とする。従来の様に、ネットワークを構成するルータやFirewallの性能バランスを検討していく達人のネットワーク構築ではなく、性能のチューニングや高性能化はコンピュータが自動で行い、スイッチ一つで高性能で高機能な仮想ネットワークの実現が可能になるためのシステムを新規に開発する。

本プロジェクトにより、VNF開発は高性能化と、機能実装が分離され、性能調節はネットワークオペレータやアルゴリズムで制御を10/40GbE以上の高い性能で可能とする、次世代のネットワーク形態の実現を目標としている。

3. 開発の内容

本プロジェクトの目的を達成するために、ルータ等のVNFの開発を支援するフレームワークを開発した。そして、それらを用いて実装されたVNFを適宜監視し、チューニングをアルゴリズムを用いて動的に行うことにより、インターネットの環境に対して自動で最適化する通信基盤を開発した。パケット処理のバックエンドにはDPDKを用い、開発の難易度が上がる部分をフレームワークによりできる限り隠蔽した。それにより、10/40GbE以上のVNFの開発サイクルを高めるだけでなく、VNFの性能チューニングをリモートから行えるようなシステムを開発した。本プロジェクトではこれらのシステム全体をsusanow（スサノヲ）と呼び、susanowを構成するサブシステムコンポーネントとしてD2を開発した。D2はD2-Lib, D2-Infra, D2-Agentの3コンポーネントで実現されている。図1にsusanowの動作イメージを示す。

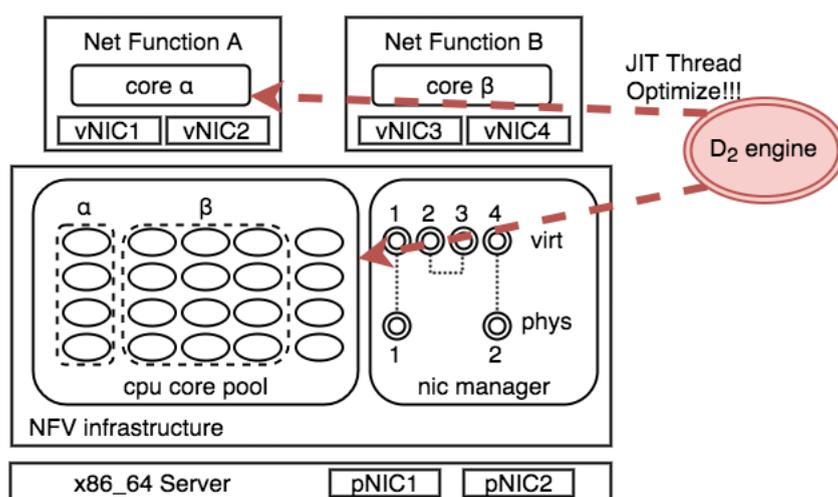


図1. susanowのシステムアーキテクチャ

- D2-Lib
D2-Libはあらかじめ、NICのMultiQueueを一定数確保しておき、それを複数のスレッドで分担してpollingを行う。スレッドの多重度を変更するタイミングで、MultiQueueの分担構成を更新することで、動的にスレッドの多重率を変更することを可能にしている。この時、変数等を別のスレッドで共有しないで実装するため、スレッドローカルストレージを活用することにより、スレッド間でデータ共有を行わないShared-Nothingで実装をすることが可能になった。
本プロジェクトでは、VNFのスレッド多重率の増加（ex. 4多重→8多重）のような動作を「D2out」と呼び、逆に多重率の減少（ex. 8多重→4多重）の動作を「D2in」と呼ぶ。これらのD2out, D2inを総じて「D2命令」とまとめて呼ぶ。

図2に、VNFのD2outの流れを図で示す。

- D2-Infra
D2-Infraはユーザから、物理サーバのリソース管理,取得や、物理サーバ上で動作するVNFの情報の取得や、VNFに対する操作のエンドポイントとなる。物理/仮想ポートに対するトラフィック状況の取得や、VNFに対するInbound Trafficの状況、処理性能等を確認可能である。D2-Infraは1つの物理サーバ上

で1プロセス動作する。NFVの専門用語ではNFV Infrastructureに相当する。

- D2-Agent

D2-Agentはsusanow上で動作しているVNFを定期的に監視し、あらかじめ指定したアルゴリズムに応じて、遠隔からD2out等のD2命令を発行する。

NFVの専門用語で表現すると、VNF Managerに相当する。

D2-Agentは1台以上のD2-Infraを管理可能であり、必要に応じてVNFをマイグレーションする機能もPoCレベルでは実装している。

susanowでは各VNFに対して、以下のような規則でD2命令を発行している

- Aコア消費中にInbound TrafficがB[pps]を下回ると、D2inを発行
- VNFがInbound TrafficをC%処理しているとき、CがDを下回った場合にD2outを発行
- A,Cは実行時情報、B,Dは閾値となるパラメータ（一部省略）

D2-AgentのアルゴリズムのパラメータB,DはVNFごとに指定可能であり、これらのパラメータをチューニングすることにより、ネットワークの自動最適化の内容を調節可能である。

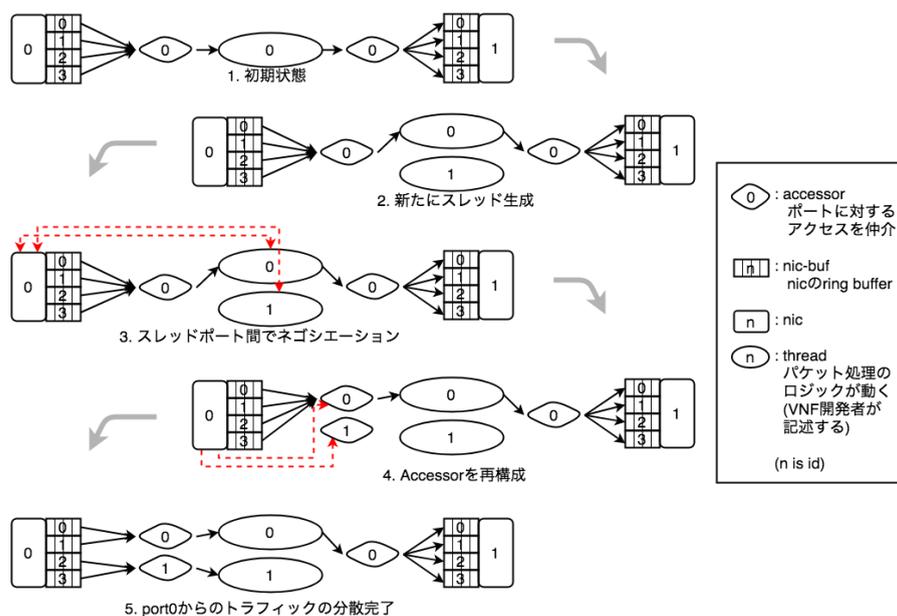


図2. D2outの流れ

4. 従来の技術（または機能）との相違

susanowには以下に示す3つの特徴がある。

1. 高性能

susanowではDPDKを用いることにより、10/40GbE以上の高性能な通信グレードをサポートしている。また、DPDKを用いた場合の高度な開発技術を要する部分（NUMAを意識したコアの割り当てや、NICのハードウェア支援機構の活用）をD2-Libのフレームワークが隠蔽し、VNFの実行時に自動で行うた

め、VNFの開発コストが劇的に低下する。SDNにおけるControl plane, Data planeの分離の様に、susanowではVNFの高性能化部分と、パケット処理ロジックの分離に成功した。

2. 動的スケール

D2-Libを用いてVNFを開発することで、VNFは秒単位でリソース量（具体的には使用するコアの数）を調節することが可能になった。これまでのDPDKを用いた高性能なVNFでは開発時や、デプロイ時に使用するリソース量を決定し、それを変更することがないため、VNFにトラフィックがほとんど来ていない場合等でリソースの無駄使いが生じていたが、D2を活用することにより、トラフィックに合わせてリソース量を調節することが可能になったため、より最適にサーバリソースを使用することが可能になった。

3. オーケストレーション

susanowは、高性能にパケット処理を行うシステムにはC/C++を用いて開発し、susanowの機能を制御するシステムはPythonで開発している。VNFのチェックや、デプロイ、生成、破棄等の基本的な制御はssnpyというsusanow制御用Pythonモジュールを使用することで制御可能である。また、ssnctlというssnpyをラップしたコマンドを利用することでssnpyを用いてプログラムを実装しなくても簡単にsusanowの制御が可能である。

ssnpyはsusanow上に実装されたREST APIをラップしたものであり、Ruby等の別のユーザフレンドリーな言語に移植可能である。

周辺ツールのsusanow/hackにはgrafana可視化等の他システムとの連携を実現するためのプログラムなどがある。

susanowは従来のNFV実装では不可能であった、「ネットワーク性能の自動スケール」を可能にしている。前節でも述べた通り、サーバが受け入れるトラフィックのパターンに合わせて、各VNFが協調的にリソース使用量を増加/減少させることで、少ないリソース量でネットワーク処理の効率を最大化することが可能である。

5. 期待される効果

図3は既存のDPDKのみを用いた固定リソース割り当てで高性能NFVを構築した場合とsusanowを用いてリソース割り当てをアルゴリズムで自動制御した場合のNFVの性能を比較している。従来のNFVではDPDKを用いることでそれぞれのVNFを高性能に実現することは可能であるが、各VNFの使用リソース量が固定であるため、トラフィックのバランスによって、限界が生じていた。susanowではD2を用いることでトラフィック量に合わせて、VNFの使用リソース量を動的に自動制御するため、チューニングなしで、従来と比べ約15%の性能向上を実現した。

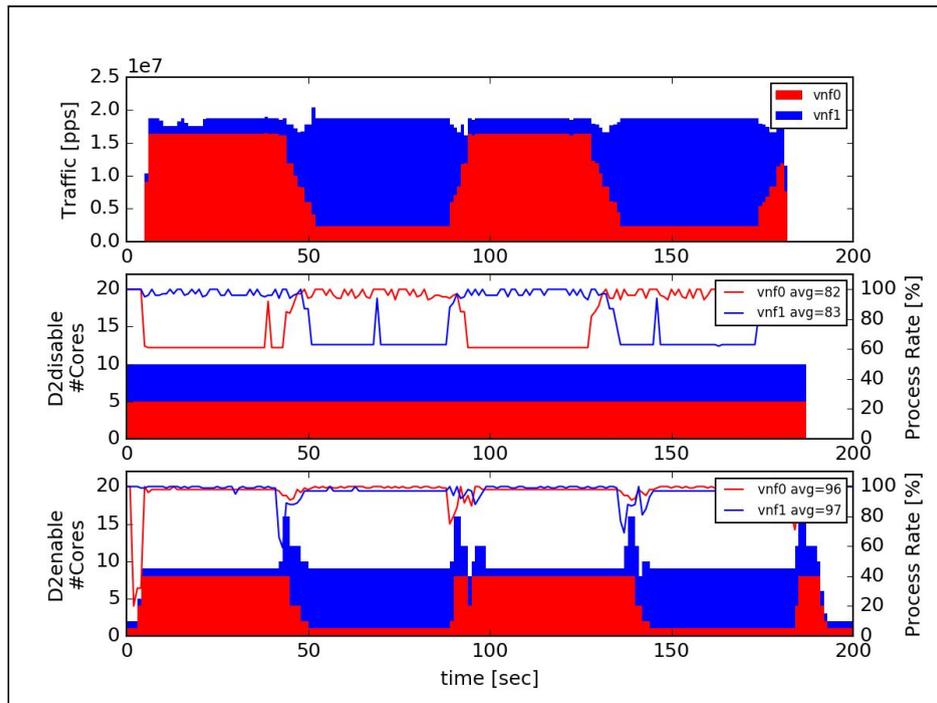


図3. 従来のNFVとsusanowを用いたNFVの性能

また、susanowではネットワーク性能のチューニングをアルゴリズムで制御することにより、ネットワーク性能の最適化は人の手を使うことなく、自動で行なっている。今後ネットワーク規模がより巨大化した場合において、インターネットを構成する各ネットワークの制御を人力で行うのではなく、統計情報等から自動で行える可能性を示すことができた。

6. 普及（または活用）の見通し

本プロジェクトはOSSとして公開されており、現在もバグ修正と機能追加を行なっている。また、susanow上で動作するVNFを積極的に開発していき、開発コストの低下具合等を改善していく。

7. クリエータ名（所属）

城倉 弘樹（法政大学 理工学部 金井敦研究室）

（参考）関連URL

- susanow <https://github.com/susanow/susanow>
- ssnpy <https://github.com/susanow/ssnpy>
- D2-Agent <https://github.com/susanow/d2agent>
- その他ツールやベンチマーク等 <https://github.com/susanow/hack>