Stratosphere

SDNソフトウェアスイッチLagopus

2014年11月26日

沖　勝
m-oki@stratosphere.co.jp

目次

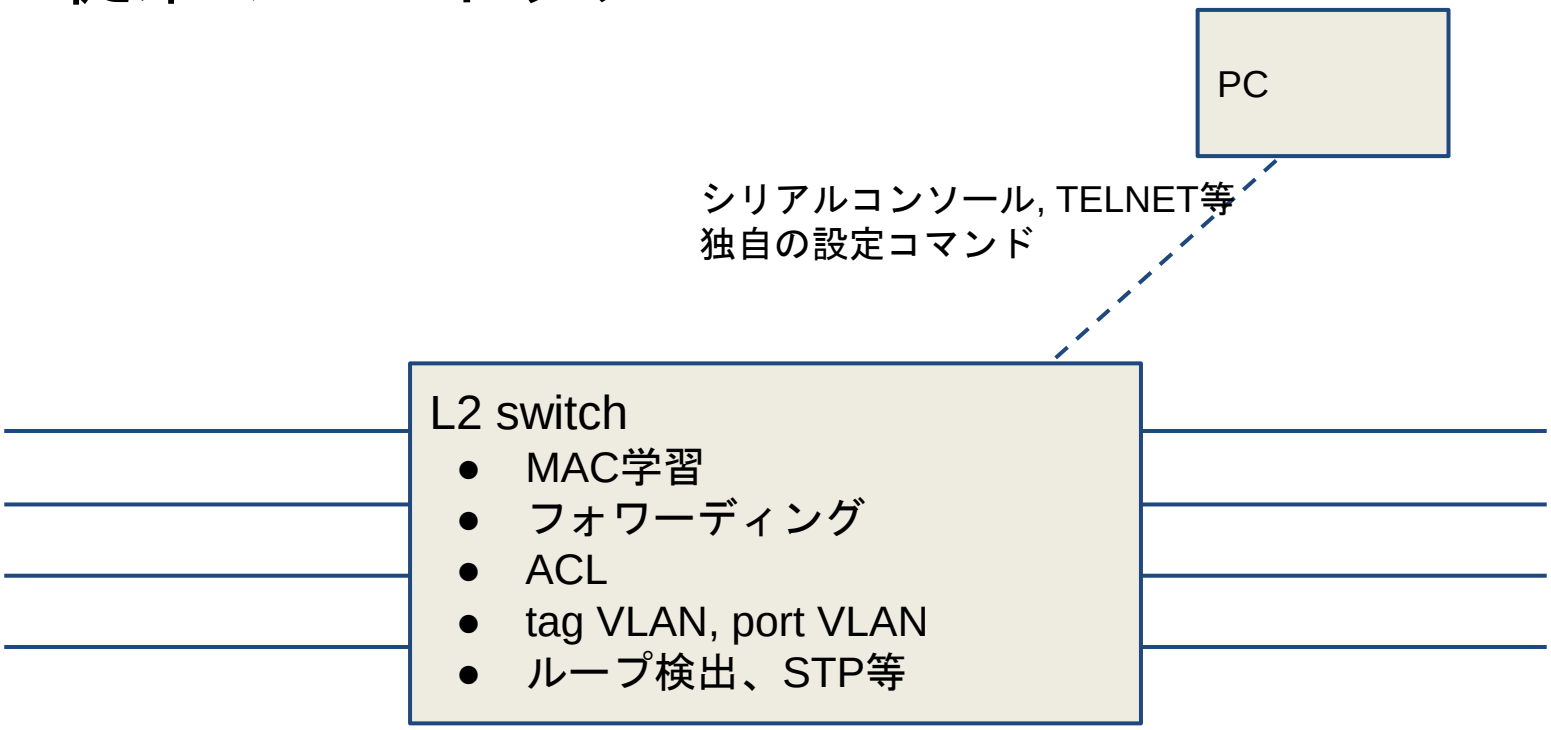
Stratosphere

- OpenFlowスイッチとは
- ソフトウェアスイッチLagopus(ラゴパス)
- I/O性能を高める工夫
- OpenFlow処理性能を高める工夫
- まとめ

OpenFlowスイッチとは

Stratosphere

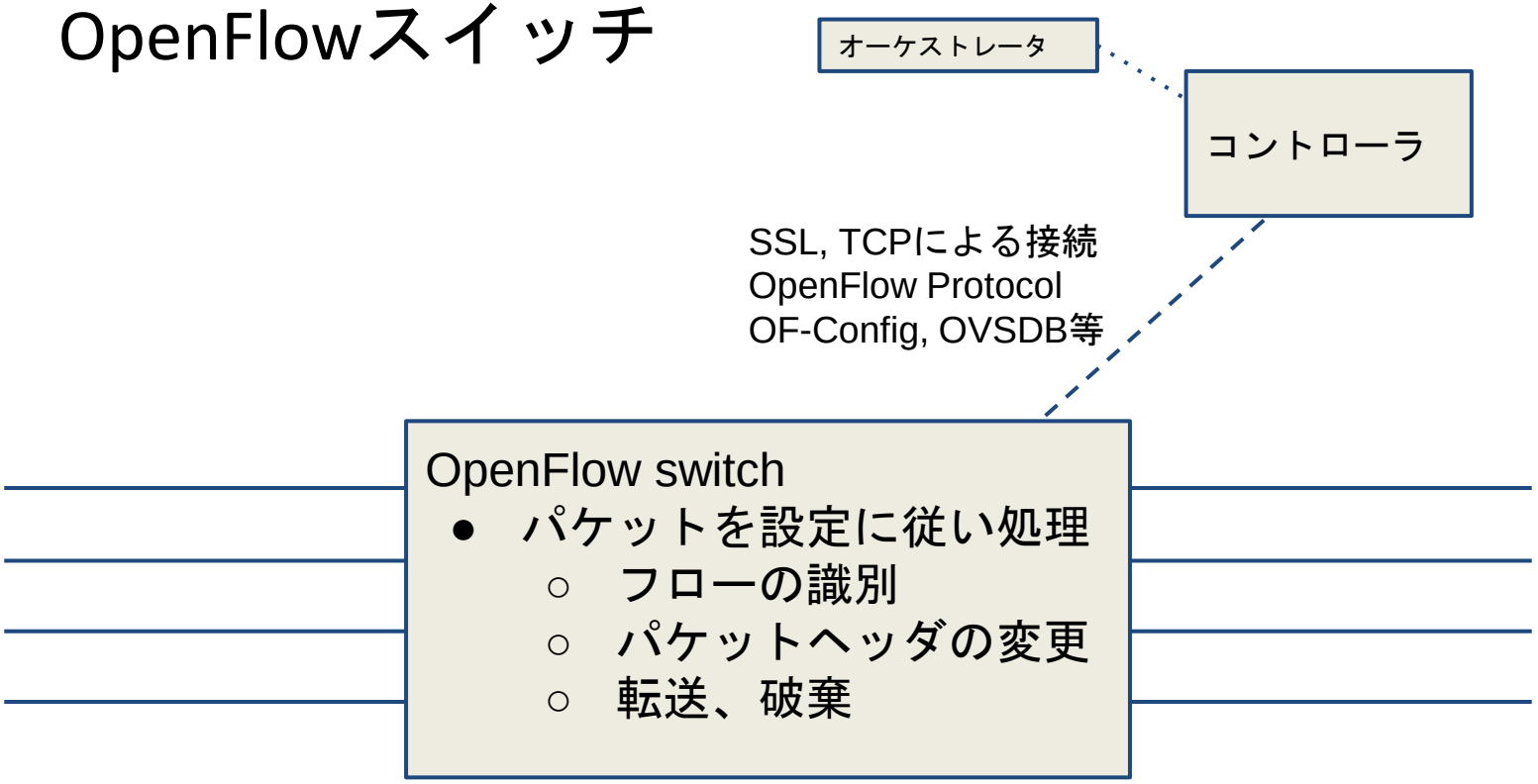
従来のL2スイッチ



OpenFlowスイッチとは

Stratosphere

OpenFlowスイッチ



OpenFlowスイッチとは

Stratosphere

- オープン標準なプロトコルによって制御される
  - ポート設定等はOF-Config or OVSDB (or 独自)
  - パケット処理の設定、参照はOpenFlowプロトコル
- ネットワークスタックを使わないパケット処理
  - 受信パケットをどのフローに属するか識別して
  - フローごとにパケットヘッダ変更や転送を実行する
  - より複雑な処理はコントローラに任せる
- 従来のL2スイッチとのハイブリッドもある

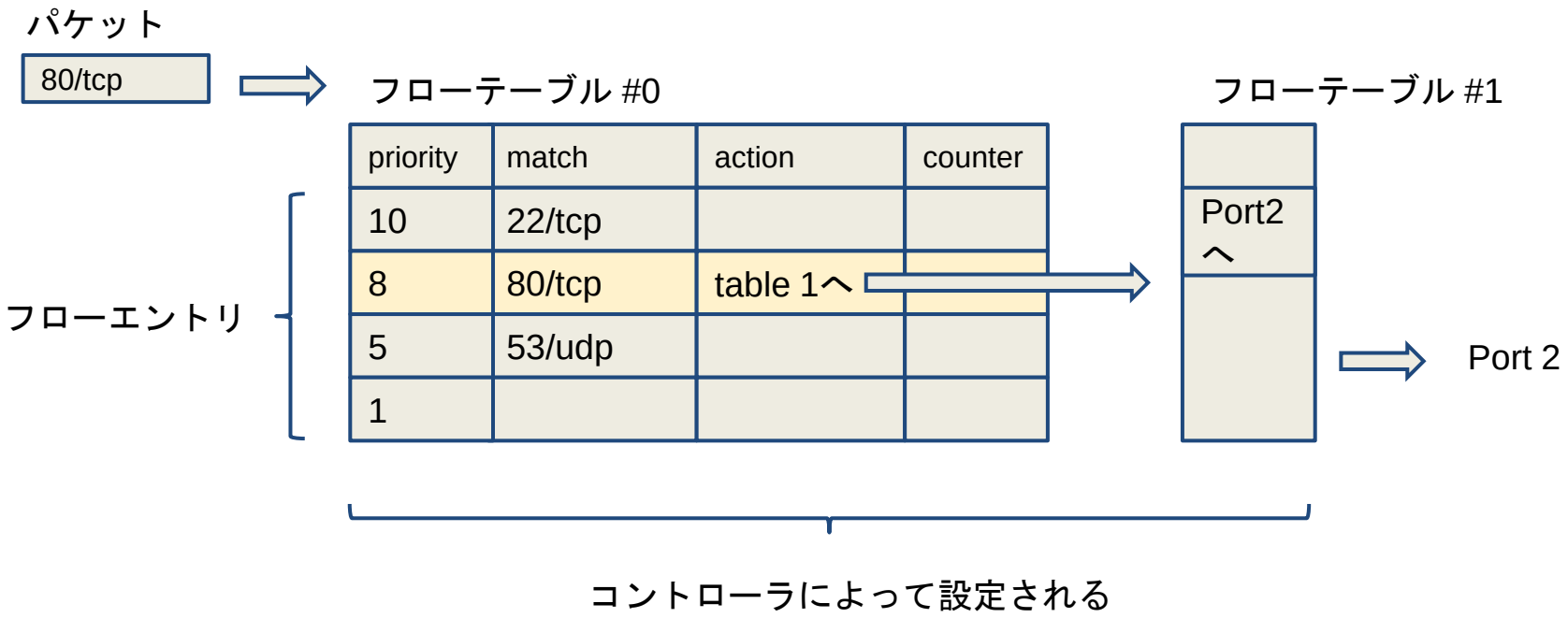
OpenFlowスイッチとは

Stratosphere

- フロー
  - パケットヘッダのフィールドの任意の組み合わせ
  - 受信したポート番号等の情報を含めて40種類
  - 例: Port1受信のパケット、80/tcp宛のIPパケット
- フローエントリ
  - フローの情報と実行処理(action)、カウンタの組合せ
  - 例: VLAN tag挿入しPort2へ出力、全ポートへ出力
- フローテーブル
  - フローエントリの集合
  - パケットは一つのフローエントリのみにマッチする

OpenFlowスイッチとは

OpenFlow処理の例



ソフトウェアスイッチLagopus(ラゴパス)

Stratosphere

- Linux上で動作する、OpenFlowスイッチ
  - 従来のL2スイッチの機能はなし
  - OpenFlow 1.3.4仕様に準拠
- Xeon, 10GbE NICにて高性能処理を目標に開発
  - 10万フローエントリを処理可能に
  - 10GbE line rate(14.88Mfps)のスループット
- オープンソースソフトウェア(Apache License)
  - http://lagopus.github.io

ソフトウェアスイッチLagopus(ラゴパス)

Stratosphere

特徴

- Intel DPDKを用いた高速なI/O
- マルチコアを活用した同時並行処理
- 10万フローエントリで10GbE line rateの性能
- OpenFlow 1.3.4仕様のほとんどをカバー
  - MPLSヘッダ、VLAN tagの多段push, popに対応
  - group, meterに対応
  - Ryu certificationにて OK (982) / ERROR (9) ※開発版
  - queueは未実装

I/O性能を高める工夫

Stratosphere

I/Oが遅くなる要因

- メモリコピー
- コンテキストスイッチ
- ページフォルト
- バスを跨いだI/O,メモリアクセス

I/O性能を高める工夫

Stratosphere

メモリコピー

- 通常のI/O処理
  - 受信時にカーネル→ユーザのコピーが発生
  - 送信時にユーザ→カーネルのコピーが発生

- 解決策
  - カーネルに閉じた処理とする (kernel network stack)
  - カーネルメモリをshareする (netmap等)
  - カーネルを介さず送受信処理する (DPDK)

I/O性能を高める工夫

Stratosphere

- コンテキストスイッチ
  - 割り込み処理やプロセス切り替え
- 割り込みを減らす、割り込みしない
  - 負荷が上がった時はポーリング (Linux NAPI)
  - ポーリングモードドライバ (DPDK)
- プロセス切り替えをさせない
  - core affinity指定によってコアを占有(DPDK)

I/O性能を高める工夫

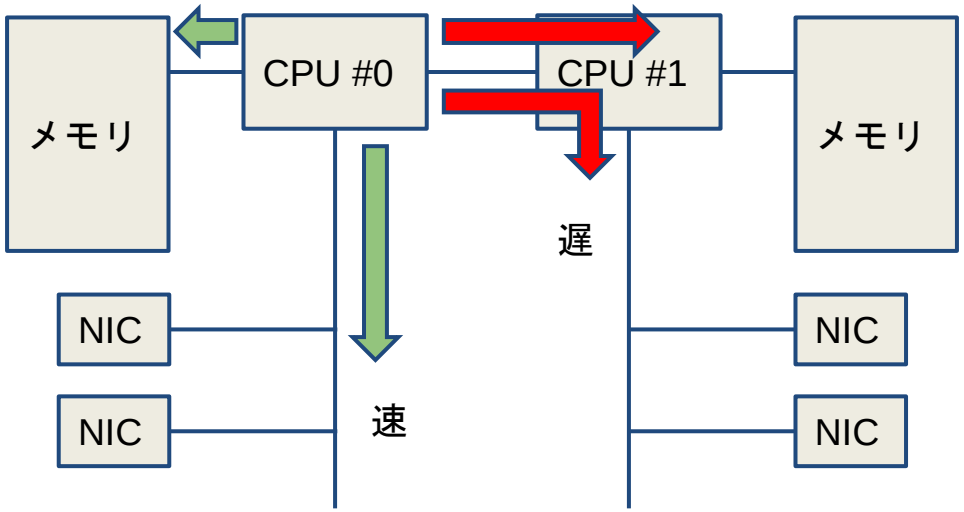
Stratosphere

- ページフォルト
  - 4KB/page単位のメモリ管理
  - TLBで物理ページ-論理ページのマッピング管理
  - TLBにないページへのアクセスはTLB miss例外
  - 例外処理にてTLBの内容を入れ替えてアクセス
- ページフォルトを減らす
  - hugepage (2MB/page or 1GB/page)を利用
  - CPUが提供している機能、OSのサポートが必要
  - 特殊ファイルシステム(hugetlbfs)をmountして使う

I/O性能を高める工夫

Stratosphere

- バスを跨いだI/O,メモリアクセス
  - 2 Socket以上のサーバーボードの構成
  - どちらのコアが処理するかで性能が変わる
  - Socketを指定したメモリプール確保などで最適化



OpenFlow性能を高める工夫

Stratosphere

- フロー探索アルゴリズム
- リソースの排他アクセス
- マルチコアを活用した分散処理
- フローキャッシュ

OpenFlow性能を高める工夫

Stratosphere

フロー探索アルゴリズム

- 10万フローエントリを順にスキャン→遅すぎる
  - 試作での測定結果: Xeon 2.2GHzにて 20Kbps

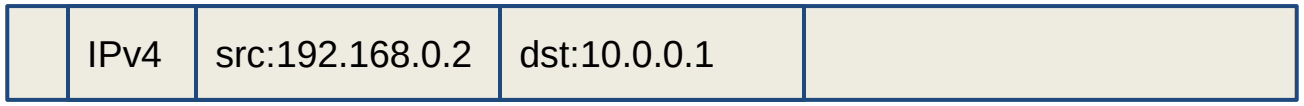
- OpenFlow仕様による制約
  - フローのマッチ対象のフィールドは40種類
  - フローのマッチ条件にワイルドカード有り
  - フローエントリの優先順位に従う必要がある

OpenFlow性能を高める工夫

Stratosphere

例

パケット



フローテーブル

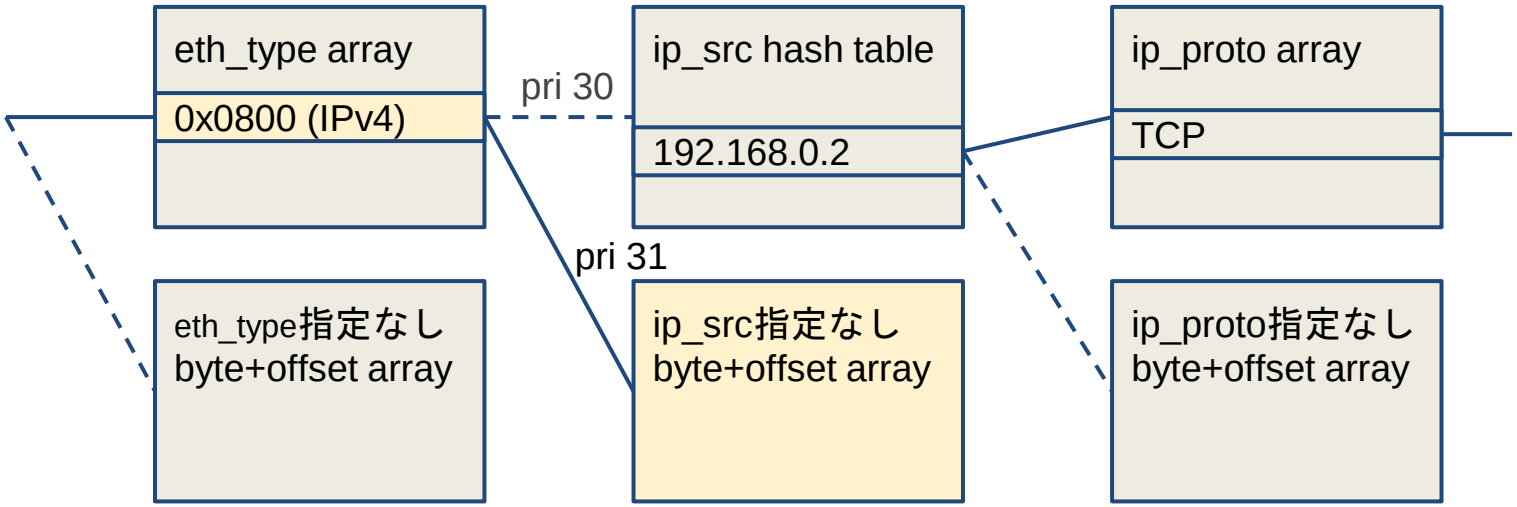
フロー#	priority	ip_src	ip_dst
1	32	192.168.0.1	-
2	31	-	10.0.0.1
3	30	192.168.0.2	-
4	29	192.168.0.3	-

priority順に探索する

ip_srcで探索すると
フロー3となるが誤り

OpenFlow性能を高める工夫

- exact matchとその他のテーブルを用意
- その他はオフセットと比較バイト値に翻訳
- 双方を探索し、priorityにて適切なフローを選択



OpenFlow性能を高める工夫

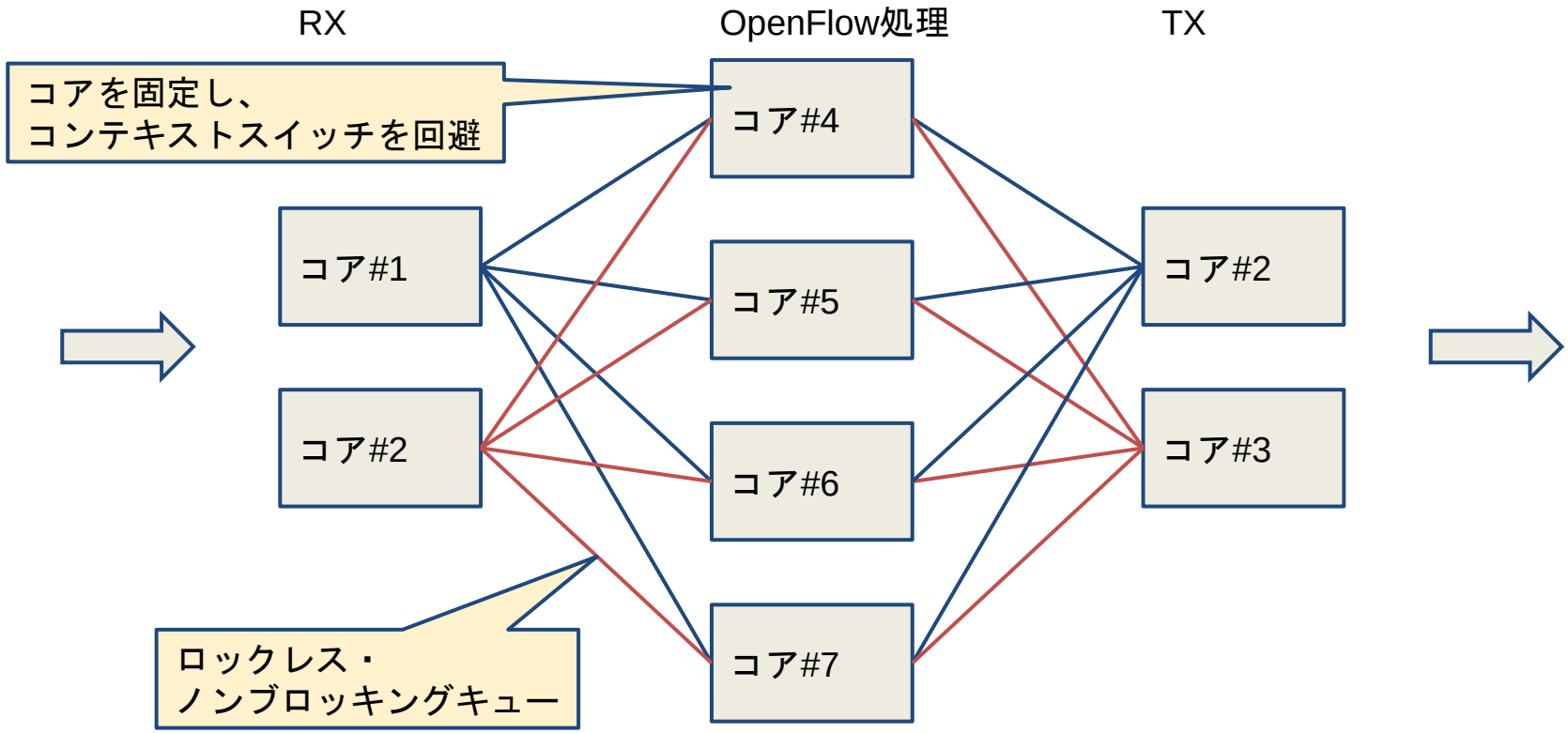
Stratosphere

- リソースの排他アクセス
  - フローテーブルは圧倒的にread lockが多い
  - フロー追加・変更はwrite lock
  - パケットが来るたびテーブルをlock, unlockしない
  - 単一スレッドでリソース保持できる場合lockしない

- 10GbE line rate → 2Mfps

OpenFlow性能を高める工夫

- マルチコアを活用した分散処理
- 10GbE line rate → 8Mfps (8core使用)



OpenFlow性能を高める工夫

Stratosphere

- それでもOpenFlowのマッチ処理は重い
- 対策: フローキャッシュ
  - マッチしたパケットヘッダの情報をキャッシュ
  - キャッシュを引いてヒットすればフローが確定
  - ミスした場合、従来の処理を実行しキャッシュ登録
  - フローの2つ目以降のパケットを高速処理できる

- 10GbE line rate → 14.88Mfps
  - 10万フローエントリ、MPLS100フロー、VLAN変換
  - Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz

Lagopusの課題

Stratosphere

- 取り組むべき課題
  - 論理ポート(トンネル)の対応
  - コンフィグレーションシステムの改良
  - OF-Config, OVSDB対応
  - L2スイッチとのハイブリッド化
  - さらなる処理性能の向上(多ポート処理等)
  - ハードウェアアクセラレーションの活用

まとめ

- LagopusはOpenFlowソフトウェアスイッチ
- Intel DPDKを活用し、I/O性能を向上
- OpenFlow処理の実装を工夫し性能を向上
- マルチコア10GbEにて14.88Mfpsの性能を実現
- 今後も改良・高速化を進めていきます!
- http://lagopus.github.io