ケーススタディー

サーバー、ネットワーク
SDN/NFV ソリューション
FPGA による通信事業者向けクラウド・アクセラレーター

低遅延 GRE 処理アクセラレーターの検証

個別アプリケーション、ネットワーク運用に特化した FPGA アクセラレーション

会社概要

株式会社ミクシィ

ミクシィグループは、"ユーザーサプライズファースト"の企業理念のもと、ユーザーの皆さまの想像や期待を超える価値提供に取り組んでいます。当社グループは 1997 年の創業以来、SNS「mixi」やスマホアプリ「モンスターストライク」など、友人や家族といった親しい人と一緒に楽しむコミュニケーションサービスを提供してきました。これからも、"フォー・コミュニケーション"と定めたミッション（私たちのやるべきこと）を遂行するため、人々の生活がより豊かになる未来を思い描き、IT の側面からコミュニケーションの活性化を促す事業・サービスを推進し、より良いコミュニケーションの創造に取り組んでいきます。

mixi GROUP

はじめに

株式会社ミクシィ 開発本部 インフラ開発グループ（以下、ミクシィ）では、データプレーン開発キット（DPDK）などを用いて Evolve Packet Core（EPC）ソフトウェアやスマートフォン・アプリケーションが動くオンプレミス・ネットワーク環境のためのパケット処理を実装し、サービスを運用してきました。

その一例として、マルチクラウドを実現するために採用しているレイヤー 3 トンネリング・プロトコルである Generic Routing Encapsulation（GRE）のデカプセル化を FPGA でアクセラレートする仕組みについて、このケーススタディーで紹介します。

ミクシィでは、クラウド上のサーバーとオンプレミスのサーバーを Point-to-Point の GRE プロトコルを用いて接続することで、クラウド側のネットワーク制約を吸収してシームレスなマルチクラウド環境を構築しています。

ここで Point-to-Point の GRE トンネルを設定するためには、双方の終端 IP アドレスを決めてネットワークを構築する必要があり、そのため、クラウド環境上のサーバーリソース変更に連携した IP アドレスの構成変更が必要となります。

今回、ミクシィのネットワーク環境ではオンプレミス側で GRE ヘッダーを削除する機能のみが必要であり、商用のネットワーク機器が提供するフル機能の GRE ヘッダーの処理機能は過剰なため、特定条件を満たす GRE ヘッダーを削除するだけの処理を必要十分かつ低レイテンシーに実装したいと考えました。

この資料では、マルチクラウド接続の制約を解決する GRE ヘッダー削除を題材に、コンテンツ・プロバイダーの特定ケースでのネットワーク設計・運用における効率的かつ実用的な FPGA アクセラレーションに関して、DPDK などを用いたソフトウェア処理と比較しながら説明します。

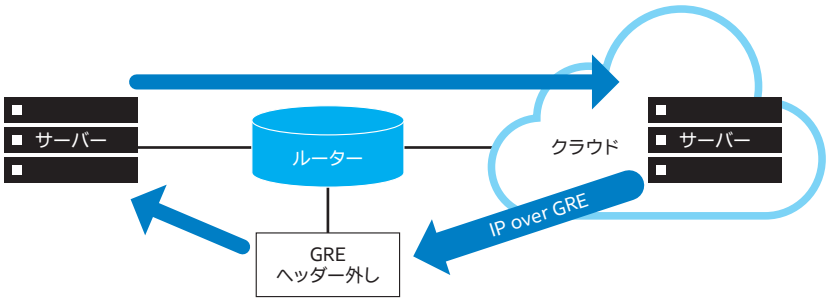

図 1. GRE を使う通信の例

インテル® アクセラレーション・スタック

インテル® アクセラレーション・スタック（インテル® Xeon® CPU & FPGA 対応）は、インテルが開発、提供するソフトウェア、ファームウェア、ツールを集めた堅牢なフレームワークです。

これにより、インテル® FPGA が搭載されたアクセラレーション・ボードの導入開発を容易にしてデータセンターのワークロードを最適化できます。インテル® アクセラレーション・スタックには、最適化および簡素化されたハードウェア・インターフェイスとソフトウェア・アプリケーション・

プログラミング・インターフェイス (API) が備わっています。これにより、FPGA アクセラレーション・カードに必要なインフラ環境の開発時間が大幅に節約されるため、開発者はより付加価値の高いソリューションの構築に集中できます。

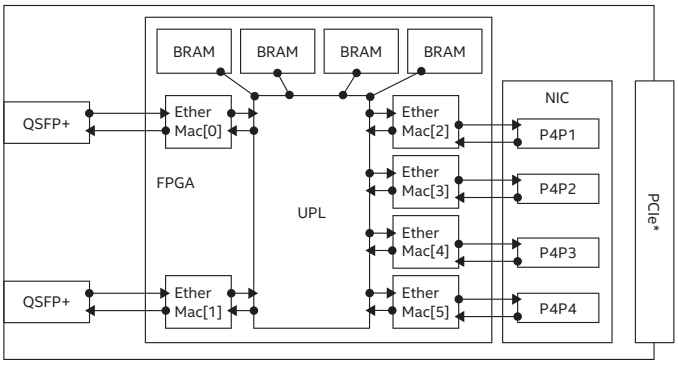

図 2. インテル® FPGA PAC N3000 評価ボード

今回使用したインテル® FPGA PAC N3000 評価ボード (インテル® FPGA プログラマブル・アクセラレーション・カード N3000) では、QSFP28/QSFP+ に接続する Ethernet MAC 機能と NIC Device であるインテル® イーサネット・コントローラー XL710 に接続する Ethernet MAC 機能の間の User Programable Logic (UPL) ブロックにユーザー独自のパケット処理機能を実装することが可能であり、ユーザーはこのコアロジックの開発に集中することが可能です。

タイミングチャート – デカプセル化

今回評価対象としたFPGAの内部インターフェイスは、1クロックサイクルで、256bit (32Byte) のパケットデータを受信できます。1列目は入力パケットデータを示します。GRE パケットは L3 トンネルなので 2 番目のデータまで受信すれば、デカプセル化処理を開始することができます。

Logic Analyzer を追加して内部波形を確認したところ、GRE パケットの最小サイズ (Ethernet Frame (64B) + 外側 IPv4 ヘッダー (20B) + GRE ヘッダー (4B) = 88 Byte) のケースで 5 ～ 7 クロックサイクルで 1 パケットの全データを受信できます。またデカプセル化処理で 2 クロックサイクル分の受信パケットデータをバッファリングして 3 クロックサイクル遅延でパイプライン処理を実装設計することができました。

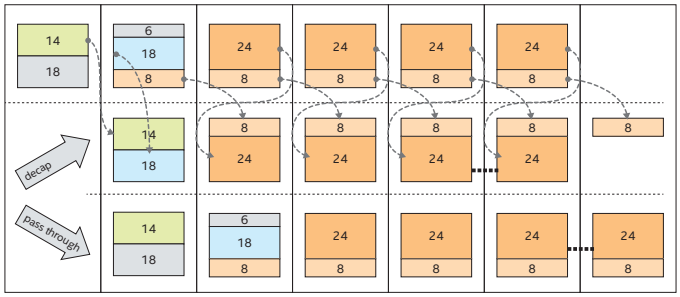

図 3. タイミングチャート

ステートマシン

前記のデカプセル化処理を行うために、今回は以下のようなステートマシンにて実装しています。初期状態 (INIT) から、受信したパケットデータが処理対象である GRE パケットであるか否かを判断して状態遷移し、valid 信号の有無で待機状態 (WAIT) とペイロード部処理状態 (PYLD) を遷移します。最後に、パケットの終わりを示す EOP 状態 (EOP) へ遷移し、初期状態 (INIT) へ戻ります。

```
XGRED_STATUS_INIT:begin
                    if (in_eop==OFF&& in_sop==ON && in_val==ON) begin
                      if (__in_dat[159:144] == 16'h0800 && __in_dat[71:64] == 8'd47) begin
                        __internal_status_next = XGRED_STATUS_SOP;
                      end else begin
                        __internal_status_next = XGRED_STATUS_PASS;
                      end
                      __in_dat_ss = __in_dat;
                      __counter = 8'd0;
                    end else begin
                      __internal_status_next = XGRED_STATUS_INIT;
                      __counter = 8'd0;
                    end
end
```

図 4. ステートマシン実装の一部

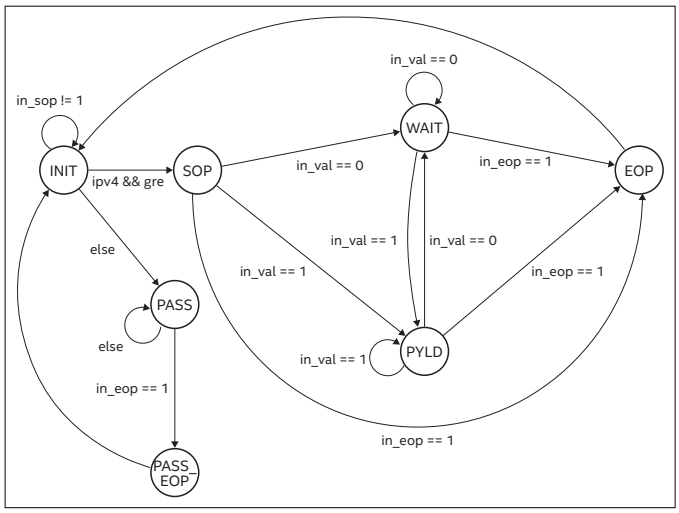

図 5. 状態遷移図 (一部省略あり)

プラットフォーム比較

GRE デカプセル化処理のパフォーマンス (レイテンシー、スループット) を、FPGA アクセラレート、DPDK を用いたソフトウェア処理、およびネットワーク機器の直接接続で比較します。評価トポロジーを次図に示します。

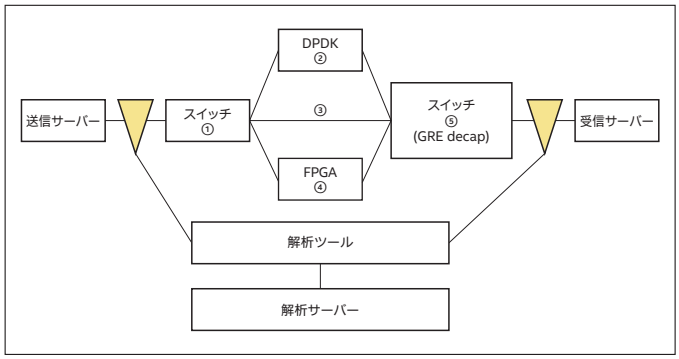

図 6. プラットフォーム比較

プラットフォームのレイテンシー比較結果

それぞれ 1 パケットのレイテンシーの測定値は次表のようになりました。DPDK はユーザーランド・ソフトウェア処理であり、CPU ポールモード+バッチ処理によって高速パケット・プロセシングを実現するプラットフォームのため、1パケットのレイテンシーは予想どおり、FPGA によるハード処理より大きくなっています。

また電力消費においても、DPDK では CPU をパケット取得処理で占有してしまうのに対して、FPGA でアクセラレートすることにより削減できることが期待されます。

No	プラットフォーム	経路	レイテンシー
1	DPDK GRE	1 + 2 + 5	9.52 us
2	DPDK TCP		9.45 us
2	FPGA GRE	1 + 4 + 5	3.18 us
3	FPGA TCP		3.01 us
4	Switch GRE	1 + 3 + 5	2.29 us
5	Switch TCP		2.09 us

測定方法 : TCP セグメントは、Ethernet FCS 込みの Ethernet Length 100Byte を使用。GRE は、Ethernet FCS 込みの Ethernet Length 100Byte のデータを送信し、IP/GRE ヘッダー(24B) を削除した 76Byte の Ethernet フレームを受信サーバーで受信。1 パケットの送信を 5 回行い、その平均値を測定。

表 1. プラットフォームのレイテンシー比較結果

プラットフォームのスループット比較

スループットの測定値は次表のようになりました。十分な性能が達成できていることが分かります。

length 92byte(GRE TCP 1:1)				
	DPDK+QFX	QFX(Decap)	FPGA	理論値
tx byte	55168165240	21304829160	12308110704	
rx byte	47972317600	18525938400	10702704960	
rx/tx byte(%)	0.8695652174	0.8695652174	0.8695652174	0.8695652174
tx pps	11.17 Mpps	11.17 Mpps	11.19 Mpps	
rx pps	11.17 Mpps	11.17 Mpps	11.19 Mpps	
loss packet	0	0	0	0

測定方法 : TCP セグメントは、Ethernet FCS 込みの Ethernet Length 92Byte を使用。GRE は、Ethernet FCS 込みの Ethernet Length92Byte のデータを送信し、IP/GRE ヘッダー(24B) を削除し 68Byte の Ethernet フレームを受信サーバーで受信。TCP と GRE を 1:1 の割合で送信し、受信サーバーで受信。

表 2. プラットフォームのスループット比較

プラットフォーム比較考察

ソリューションには、常にメリットとデメリットの両面があり、自社チームの状況に応じたソリューションを選択することがもっとも重要だと言えます。

商用のネットワーク機器

- メリット : 安定性に優れ、人的リソースを配置しやすい
- デメリット : コンフィグレーション管理が煩雑(今回のケースの場合)

FPGA NIC でのハードウェア処理

- メリット : 低レイテンシー、消費電力、スループット効率
- デメリット : 人的リソースを配置しにくい、開発環境など

DPDK ベースのソフトウェア処理

- メリット : 人的リソースを配置しやすい
- デメリット : 開発環境の追随、消費電力、レイテンシー

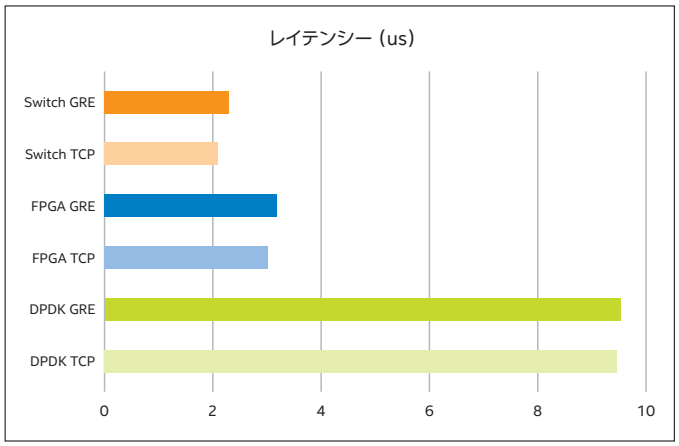

図 7. レイテンシーの比較

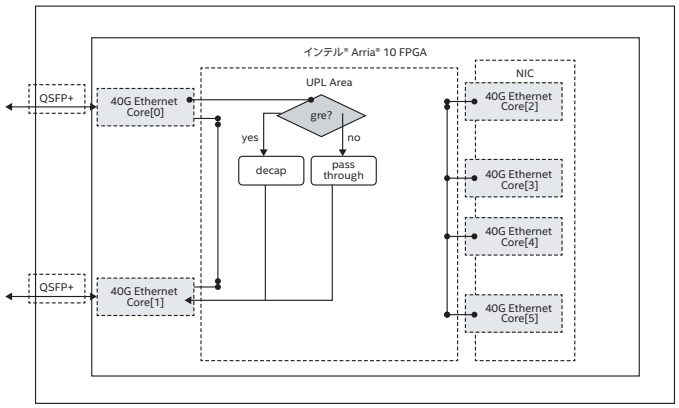

図 8. FPGA モジュール論理構成

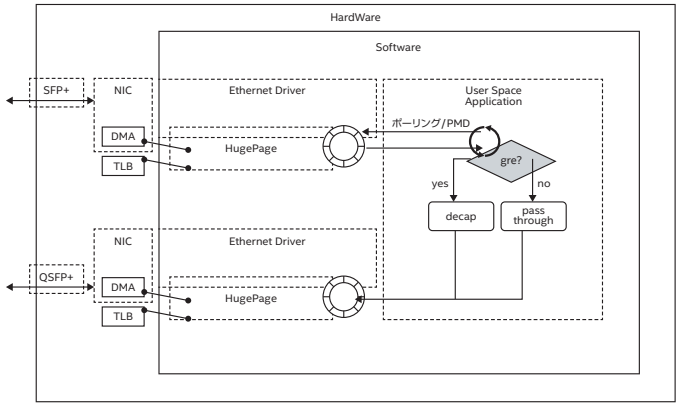

図 9. DPDK モジュール論理構成

まとめ

昨今、サービスのクラウド化、集中化にネットワーク機器の拡張は追いついていないと言われており、そこにはサーバーによるソフトウェア処理によって従来の装置を置き換えるSDNやNFVのソリューションが提案されています。

しかし、このケーススタディーで、商用のネットワーク機器、FPGA NICによるハードウェア処理、DPDKベースのソフトウェア処理を比較検証したとおり、狭義のソフトウェア処理によるSDNやNFVは最適なソリューションとならないケースがあることを示しました。

個別に専用のネットワーク機器が必要だった多様なネットワーク機能の一部は、FPGAによるハードウェア処理やDPDKベースのソフトウェアによってアクセラレートすることができ、このケーススタディーのように自社ネットワーク要件によっては、独自にFPGAでネットワーク機能の一部を代替する方が効率的なケースもあります。このようなケースでハードウェア・プログラマブルなFPGAが選択肢になるのは、広義のSDNとNFVだとも言えるでしょう。

参考文献

- DPDKのウェブページ
  https://www.dpdk.org/（英語）
- インテル® FPGAアクセラレーション・ハブのウェブページ
  https://www.intel.co.jp/fpgaacceleration/

338801-001JA