WP504 (v1.0.1) 2018 年 10 月 14 日

ザイリンクス Alveo アクセラレータ カードを使用した DNN の高速化

ザイリンクス Alveo データセンター アクセラレータ カードを使用したザイリンクスの xDNN プロセッシング エンジンは、高性能で電力効率に優れた DNN アクセラレータであり、リアルタイム推論ワークロードの純粋な処理性能と電力効率の点で、今日の一般的な多くの CPU/GPU プラットフォームよりも優れています。xDNN プロセッシング エンジンは、AWS EC2 や Nimbix NX5 などの多くのクラウド環境で利用可能な ML Suite により提供されます。

概要

ザイリンクス Deep Neural Network (xDNN) エンジンは、ザイリンクス Alveo™ データセンター アクセラレータ カードを使用して、高性能かつ低レイテンシで電力効率の高い DNN 高速化を実現します。電力コストを低く保ち、実装に必要な特定のアクセラレータ数を最小限に抑えることで、総保有コスト (TCO) を大幅に削減できます。

Alveo アクセラレータ カードは、高性能で電力効率と柔軟性の高い機械学習 (ML) 推論を実現します。xDNN プロセッシング エンジンは、ResNet50、GoogLeNet v1、Inception v4 などの一般的なたたみ込みニューラル ネットワーク (CNN) だけでなく、カスタム レイヤーが含まれた CNN も実行できるよう開発されています。

このホワイトペーパーでは、xDNN のハードウェア アーキテクチャとソフトウェア スタックについて概説すると共に、クラス最高の電力効率で推論を実現するという主張を裏付けるベンチマーク データを示します。

さらに、Alveo データセンター アクセラレータ カードのベンチマーク結果を再現するためのガイダンスも示します。

データセンター アプリケーションにおける深層学習の適合性

深層学習の方法論はここ数年、さまざまな応用分野で大きな成功を収めています。機械学習 (ML) の応用分野の 1 つに、ビジョン/ビデオ処理があります。インターネット上のビデオ コンテンツもここ数年で急増しており、それに比例して、画像の並べ替え/分類/識別方法に対するニーズも高まっています。

ML ニューラル ネットワークの一種であるたたみ込みニューラル ネットワーク (CNN) は、特にデータセンターの運用において効果的な画像データ処理方法となっています。CNN 画像ネットワークは、クラウド内の画像の分類や解析に利用できます。一般に、ストリーミング ビデオに含まれる違法なコンテンツを見つけるなど、エンド アプリケーションにとって画像処理のレイテンシは極めて重要です。

このホワイトペーパーでは、ザイリンクスの Deep Neural Network (xDNN) エンジンについて説明します。xDNN は、ザイリンクス Alveo アクセラレータ カード上で低レイテンシかつ電力効率の高い推論を実行できる、プログラマブルな推論プロセッサです。xDNN 推論プロセッサは、標準的な CNN ネットワークを幅広くサポートする汎用 CNN エンジンです。xDNN エンジンは、ザイリンクス xfDNN ソフトウェア スタックを通して一般的な ML フレームワーク (Caffe、MxNet、TensorFlow など) に統合されます。Alveo アクセラレータ カード上で動作する xDNN プロセッシング エンジンは、GoogLeNet v1 で 1 秒あたり 4,000 画像以上のスループットを実現します。これは、Batch=1 で 70% 以上の演算処理効率に相当します。

この演算処理効率が示すように、Alveo アクセラレータ カード上で動作する xDNN は、GPU などのアクセラレーション プラットフォームより優れており、低レイテンシの推論を実現します。GPU プラットフォームが多数の画像をまとめてバッチ処理することでパフォーマンスを高めることは広く知られています。ただし、バッチ処理によってパフォーマンスが向上し、必要な GPU メモリ帯域幅が減る反面、レイテンシが大幅に長くなります。

一方、xDNN プロセッシング エンジンはバッチ処理に頼らずに、最大のスループット性能を実現します。各エンジンは独立して動作し、重みメモリは共有されません。各エンジンは Batch=1 で動作し、1 つの Alveo アクセラレータ カード上に複数のエンジンを実装できます。つまり、デバイス内の xDNN エンジン数を増やせば、単純にその総計分バイスの Batch=1 スループットが向上します。

xDNN アーキテクチャの概要

図 1 に、xDNN ハードウェアのアーキテクチャを示します。各 xDNN エンジンは、シストリック アレイ、命令メモリ、実行コントローラー、エレメント単位のプロセッシング ユニットで構成されます。エンジンは命令キューを通して、ホスト プロセッサ上で実行中のコマンド ソフトウェアからテンソル命令を受け取ります。CNN ネットワークに対する命令 (テンソル演算とメモリ演算) が変わるのは、ターゲット ネットワークが変わる場合のみです。同一ネットワークの反復実行では、命令バッファー内にある以前に読み込まれた命令が再び使用されます。

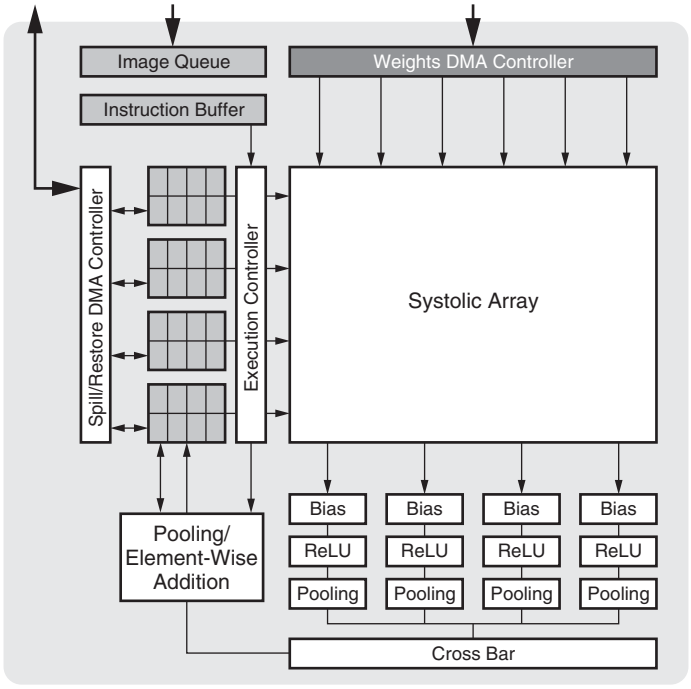


図 1: xDNN ハードウェア アーキテクチャ

xDNN プロセッシング エンジンのアーキテクチャの特長

- デュアル モード: スループット最適化またはレイテンシ最適化
- コマンド レベルの並列実行
- ハードウェア支援型の画像タイル処理
- ヘテロジニアス実行によるカスタム レイヤーのサポート
- シストリック アレイ アーキテクチャ

スループット/レイテンシ最適化モード

xDNN プロセッシング エンジンのアーキテクチャの特長の 1 つは、スループット最適化モードとレイテンシ最適化モードの 2 つの動作モードがあることです。スループット最適化モードでは、最適化されたプロセッシング エンジン (PE) を作成することでデータフローの並列処理を利用し、汎用シストリック アレイに非効率的にマップされた特定のレイヤーを処理します。

たとえば、GoogLeNet v1 の第 1 レイヤーは、全体的なコンピューティング オーバーヘッドの約 10% を占める RGB レイヤーです。ネットワークの残り部分を効率的に処理するシストリック アレイにこのレイヤーをマップすることは、効率的ではありません。このスループット最適化モードでは、3 つの入力チャネル用にカスタマイズされた追加のシストリック アレイが xDNNv3 に組み込まれます。これにより、前の画像のたたみ込みレイヤーと FC レイヤーがそれぞれの処理を実行している間に、次の画像の第 1 レイヤーを処理できるため、全体的な処理効率が上がります。

1 つの画像のレイテンシを最小限にする必要のあるアプリケーションの場合、レイテンシ最適化モードでエンジンを使用できます。このようなアプリケーションでは、xDNN PE のパイプライン処理を調整してレイテンシを削減可能です。

コマンド レベルの並列実行

xDNN プロセッシング エンジンには、コマンドのタイプ (download、conv、pooling、element-wise、upload) ごとに専用の実行パスがあります。これにより、ネットワーク グラフが許容すれば、たたみ込みコマンドをほかのコマンドと並列で実行できるようになります。一部のネットワーク グラフには、命令タイプの異なる並列分岐があり、並列処理が許容される場合があります。たとえば、GoogLeNet v1 開始モジュールの 3x3 最大プーリング レイヤーは、xDNN プロセッシング エンジンを使用してほかの 1x1/3x3/5x5 たたみ込みと並列実行できるレイヤーの典型例です。図 2 に、GoogLeNet v1 ネットワークの開始モジュールを示します。

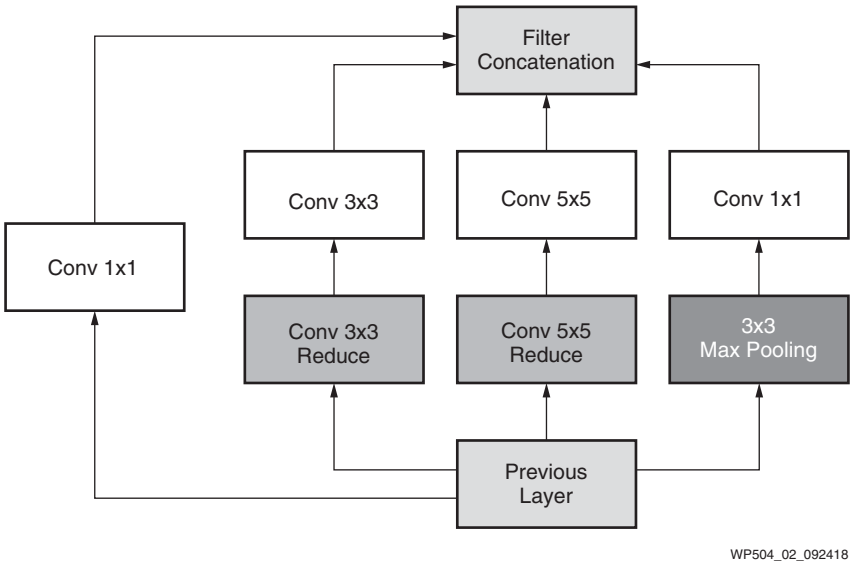


図 2: GoogLeNet v1 の開始レイヤー

図 3 に示すように、このツールは、3x3 最大プーリングを第 2 分岐の 3x3 たたみ込みと並列でスケジューリングできます。

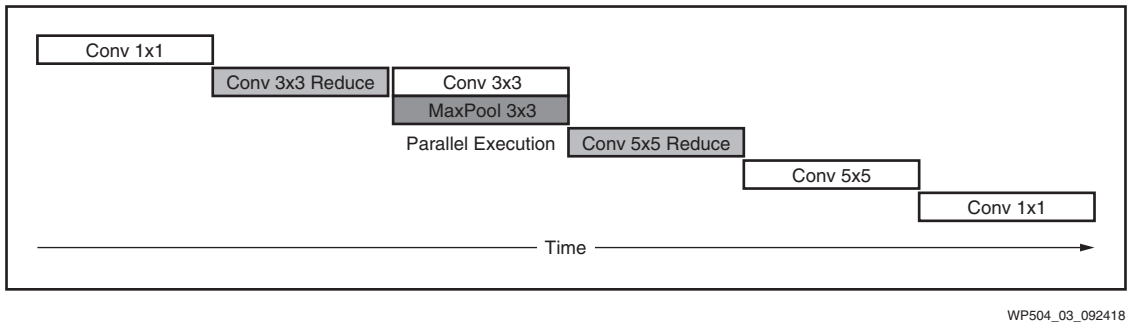


図 3: xDNN による GoogLeNet v1 の開始レイヤーのスケジューリング

ハードウェア支援型の画像タイル処理

xDNN プロセッシング エンジンには、画像/アクティベーション サイズの大きいネットワークをサポートする、ハードウェア支援型の画像タイル処理機能が組み込まれています。xDNN プロセッシング エンジンでは、幅と高さの両方で入力機能マップのタイル処理が可能です。これを図 4 に示します。

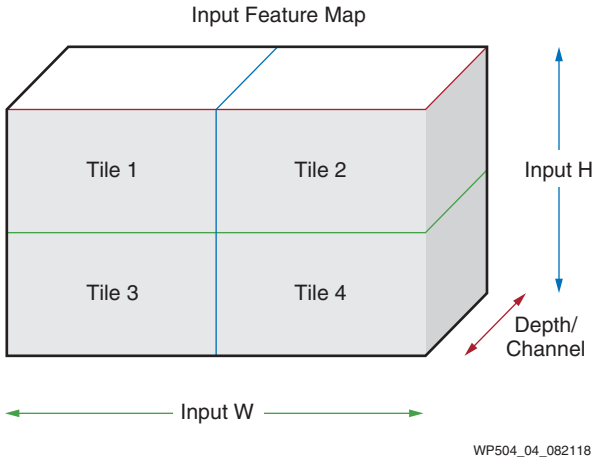


図 4: ハードウェア支援型の画像タイル処理機能

ハードウェア支援型の画像タイル処理は、1 つの非データ移動命令 (Conv、Pool、EW) を受けて、マイクロオペレーション (Download、Operation、Upload) の正しいシーケンスを生成します。マイクロオペレーションは、アクティベーション メモリを 2 つの領域 (ダブル バッファーなど) に論理的に分割することで、ハードウェア内で完全にパイプライン処理されます。

ヘテロジニアス実行によるカスタム ネットワークのサポート

xDNN プロセッシング エンジンは幅広い CNN オペレーションに対応していますが、新しいカスタム ネットワークは絶えず開発されているため、場合によっては、FPGA のエンジンでは対応できないレイヤーや命令が出てくる可能性があります。xDNN プロセッシング エンジンで対応できないネットワーク レイヤーは xfDNN コンパイラによって識別され、CPU 上で実行できます。このような非対応のレイヤーは、ネットワークのどの部分 (開始、中間、終了、または分岐内) でも起こり得ます。

図 5 に、コンパイラによって処理を xDNN プロセッシング エンジン内の各種 PE や CPU に分割する方法を示します。

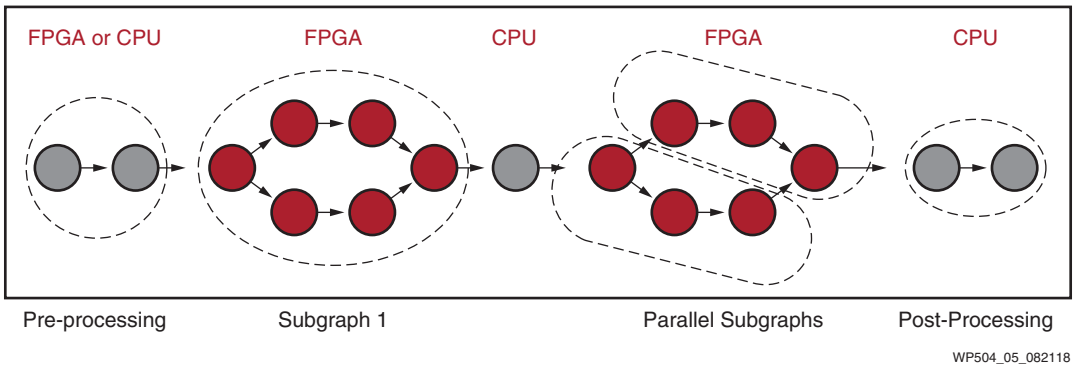


図 5: コンパイラによる処理の分割

シストリック アレイ アーキテクチャ

xDNN プロセッシング エンジンは、SuperTile に関するホワイトペーパー(1) で説明されている技法などを用いて、高い動作周波数を実現します。この SuperTile DSP マクロが提供する相対配置マクロをタイル処理することで、CNN のオペレーションの中でも特に演算負荷の高い、行列乗算やたたみ込みなどの大きな演算アレイを構築できます。

図 6 に、FPGA の DSP48 および CLB-M (LUTRAM) タイルにマップされた論理処理エレメントの例を示します。このマクロ セルは、xDNN シストリック アレイ内の基本的な処理ユニットです。

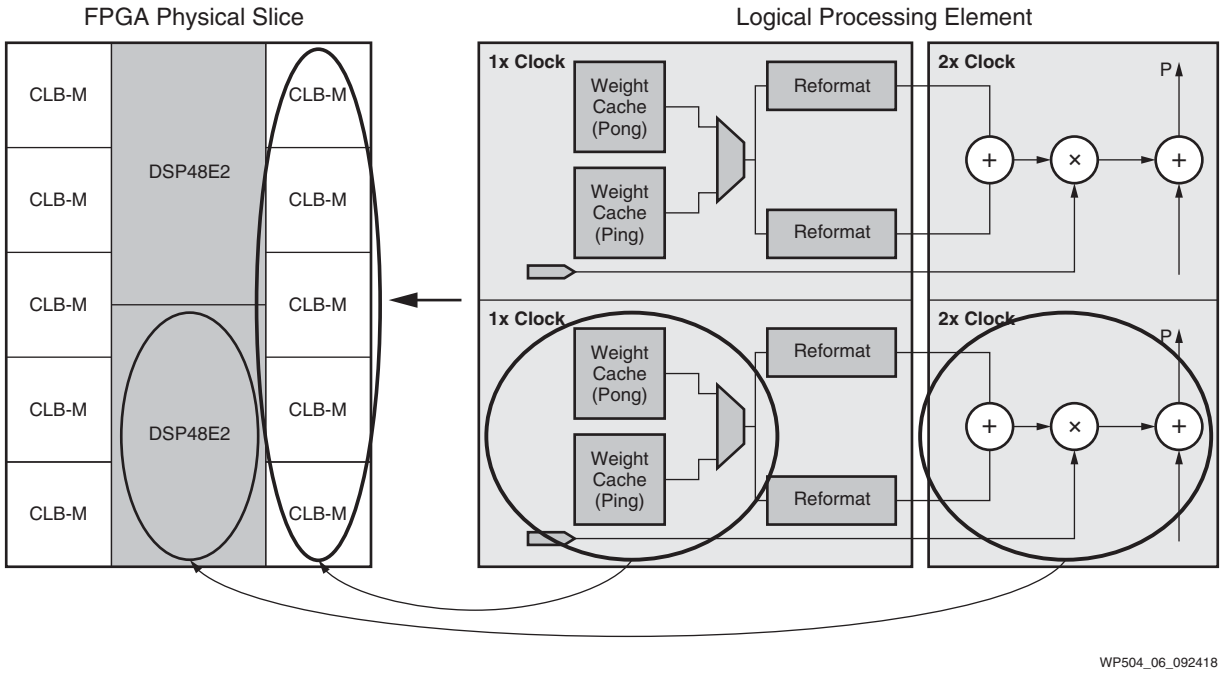


図 6: DSP マクロにおける MAC および重みパッキングの例

1. E. Wu ほか著、Xilinx Inc. IEEEXplore Digital Library, Sept. 2017、『A High-Throughput Reconfigurable Processing Array for Neural Networks』

xfDNN ソフトウェア スタックの概要

xfDNN ソフトウェア スタックは、ソフトウェア ツールと API を組み合わせたもので、一般的な各種 ML フレームワークで xDNN プロセッシング エンジンをシームレスに統合、制御できるようにします。図 7 のフロー図は、Caffe、TensorFlow、または MxNet を使用して xDNN 上で運用するネットワークとモデルの準備方法を示しています。xfDNN コンパイラは、xDNN 対応のレイヤーをサポートすると同時に、非対応のレイヤーを CPU 上で実行します。ネットワーク/モデルのコンパイルおよび量子化が完了したら (この処理の標準的な所要時間は 1 分未満)、設計者は使いやすい Python または C++ の API を使用して、xDNN プロセッシング エンジンとインターフェイスできます。

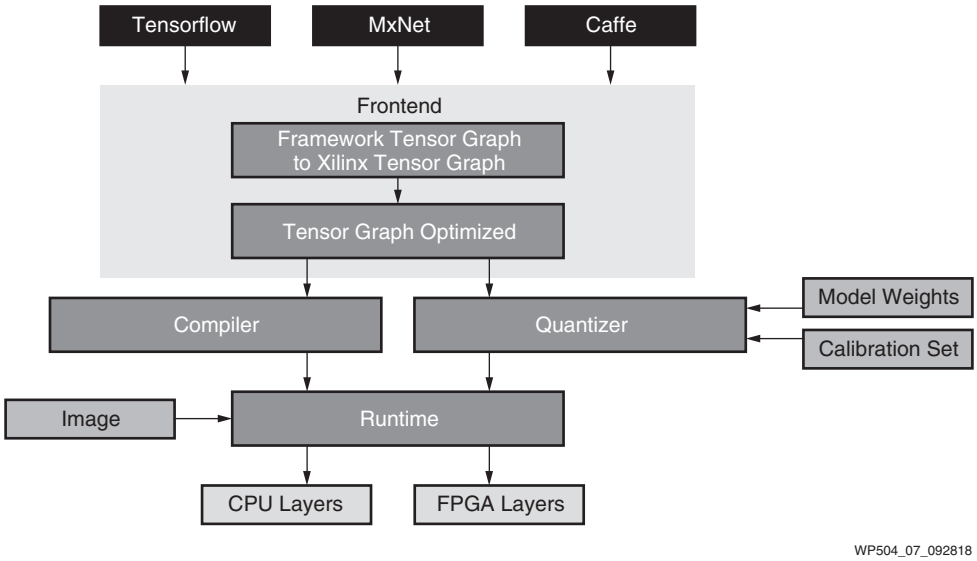


図 7: xfDNN のフロー図

ザイリンクス xfDNN ソフトウェア スタックの構成は次のとおりです。

1. ネットワーク コンパイラおよびオプティマイザー

コンパイラが xDNN エンジン上で実行される命令のシーケンスを生成し、これにより、所定のネットワークを実装するためのテンソル レベルの制御とデータフロー管理が提供されます。

2. モデル クオンタイザー

クオンタイザーがトレーニング済みの CNN ネットワーク モデルからターゲットの量子化 (INT8 または INT16) を生成するため、時間を要する再トレーニングやラベル付きのデータセットは不要です。

3. ランタイムおよびスケジューラ

xfDNN は xDNN プロセッシング エンジンの通信とプログラミングを簡略化し、SDx 準拠のランタイムおよびプラットフォームを利用します。

図 8 に、ザイリンクス FPGA 上で動作する xDNN IP と深層学習フレームワークを接続する xfDNN ライブラリのフロー図を示します。

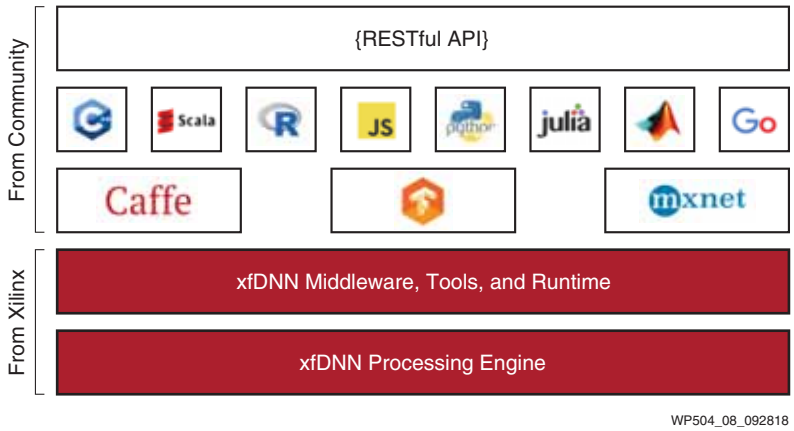


図 8: xfDNN ソフトウェア スタック

xfDNN コンパイラに関する補足情報

現代の CNN は、数百に及ぶ個々のオペレーション (Convolution、Maxpool、Relu、Bias、Batch Norm、Elementwise Add など) からなるグラフです。コンパイラの主な役割は、CNN ネットワークを解析し、xDNN 上で実行する最適な命令セットを生成することにあります。

xfDNN コンパイラには、上位の ML フレームワークに接続するためのシンプルな Python API だけでなく、レイヤーの融合、ネットワーク内のメモリ依存関係の最適化、ネットワーク全体の事前スケジューリングによってネットワークを最適化するためのツールも備わっています。これにより、CPU によるホスト制御のボトルネックが解消されます。図 9 に、この例を示します。

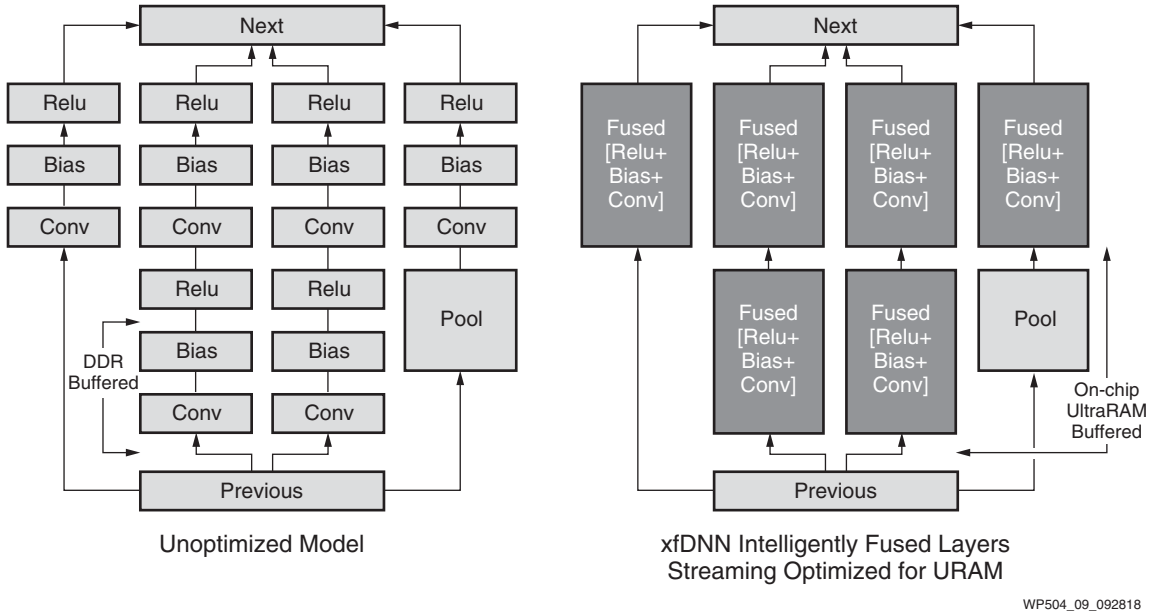
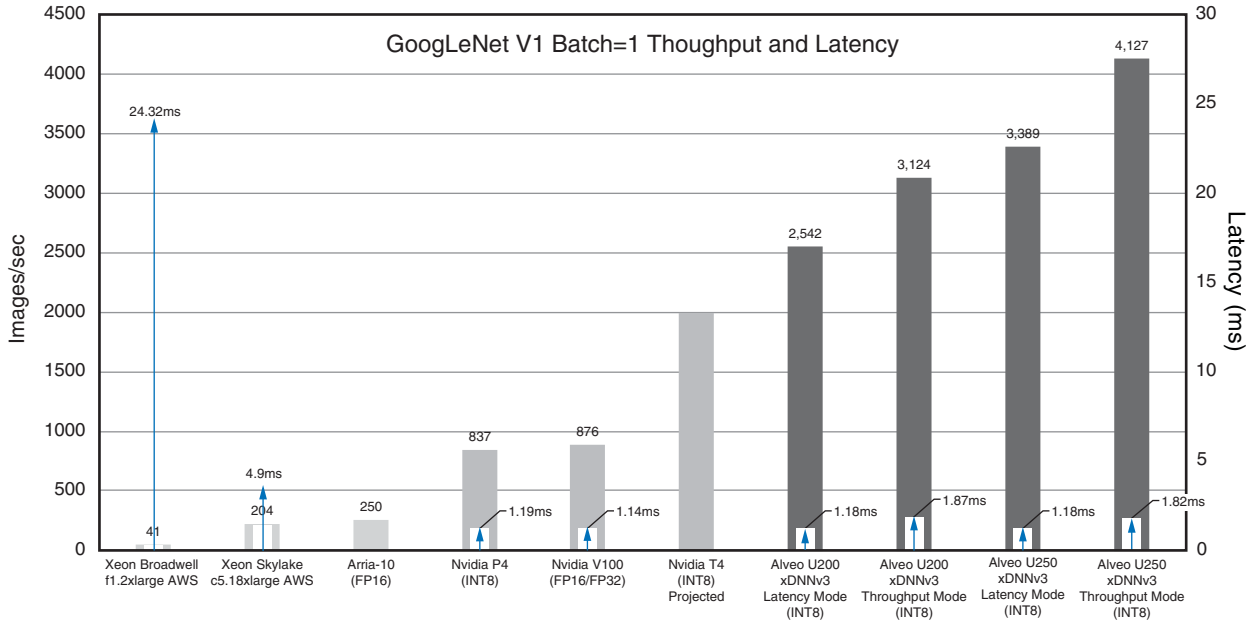


図 9: xfDNN コンパイラによる最適化

パフォーマンス ベンチマークの結果

リアルタイムの AI サービスの数が増えてくると、レイテンシが AI サービスの全体的なパフォーマンスを左右する重要な要素となります。レイテンシとスループットを両立できない GPU とは違い、xDNNv3 DNN エンジンは低レイテンシと高スループットの両方を実現します。さらに、xDNNv3 カーネルはシンプルな Batch=1 インターフェイスを提供するため、入力データの自動バッチ処理を行って最大のスループットを達成するためのキューイング ソフトウェアが不要で、ソフトウェアとインターフェイスする際の複雑性が減ります。

図 10 および図 11 に、Alveo アクセラレータ カードと一般的な GPU および FPGA プラットフォームで測定した、CNN、レイテンシ、スループットのベンチマークを示します。図 10 は、左側の Y 軸に示す 1 秒あたりの画像数で測定した GoogLeNet V1 Batch=1 のスループットを示しています。スループットの上に記載されている数値は、測定/報告されたレイテンシ (ミリ秒単位) です。



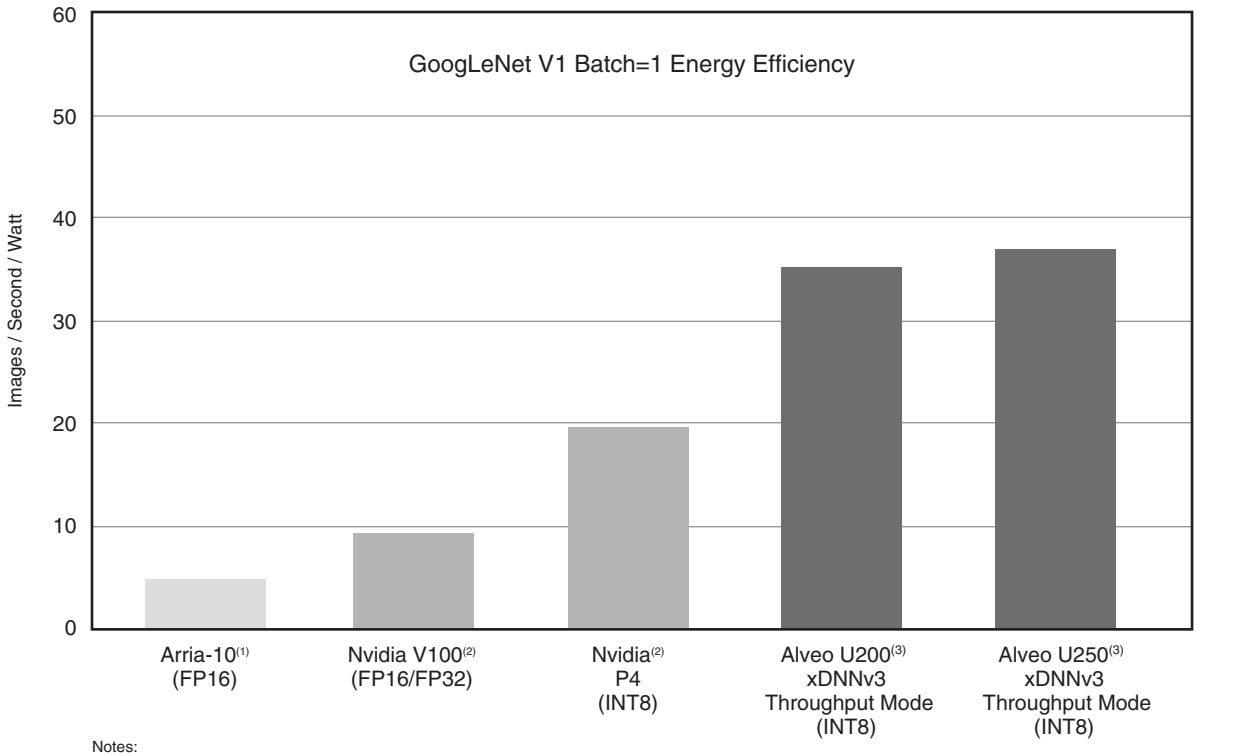
Notes:

1. Xeon E5-2696 v4 f1.2xlarge AWS instance, Ubuntu 16.04LTS, amd64 xenial image built on 2018-08-14, Intel Caffe (https://github.com/intel/caffe), Git Version: a3d5b02, run_benchmark.py w/ Batch=1 modification.
2. Xeon Platinum 8124 Skylake, c5.18xlarge AWS instance, Ubuntu 16.04LTS, amd64 xenial image built on 2018-08-14, Intel Caffe, Git Version: a3d5b02, run_benchmark.py w/ Batch=1 modification.
3. Arria-10 numbers taken Intel White Paper, "Accelerating Deep Learning with the OpenCL™ Platform and Intel Stratix 10 FPGAs." https://builders.intel.com/docs/aibuilders/accelerating-deep-learning-with-the-opencl-platform-and-intel-stratix-10-fpgas.pdf. Arria latency figures have not been published.
4. Nvidia P4 and V100 numbers taken from Nvidia Technical Overview, "Deep Learning Platform, Giant Leaps in Performance and Efficiency for AI Services, from the Data Center to the Network's Edge." https://images.nvidia.com/content/pdf/inference-technical-overview.pdf. Data retrieved on September 3, 2018.
5. Nvidia T4 projection based on current available published benchmark. GoogLeNet Batch=1 performance range between 1700-2000 images/sec based on early power efficiency benchmarks.
6. Alveo U200 numbers measured Intel Xeon CPU E5-2650v4 2.2GHz, 2400MHz DDR4, Ubuntu 16.04.2 LTS Instance running on OpenStack Pike, Centos 7.4, Pre-release Version of MLSuite, streaming_classify.py, synthetic data, MLSuite DSA Thin Shell, FC and SoftMax layers running on Xeon Host and operations not included in compute totals (0.06% of overall compute).
7. Alveo U250 numbers measured Intel Xeon Silver 4110 CPU @ 2.10GHz, CentOS Linux release 7.4.1708, Pre-release version of MLSuite, streaming_classify.py, synthetic data, DSA: ML Thin Shell, FC and SoftMax layers running on Xeon Host and Operations not included in compute totals (0.06% of overall compute).

WP504_10_092418

図 10: GoogLeNet v1 Batch=1 のスループット

図 11 は、Y 軸に示す 1 ワットあたりの 1 秒あたり画像数で測定した GoogLeNet V1 のスループットを示しています。



Notes:

1. Arria-10 numbers taken Intel White Paper, "Accelerating Deep Learning with the OpenCL™ Platform and Intel® Stratix® 10 FPGAs," https://builders.intel.com/docs/aibuilders/accelerating-deep-learning-with-the-opencl-platform-and-intel-stratix-10-fpgas.pdf.
2. Nvidia P4 and Nvidia V100 numbers taken from Nvidia Technical Overview, "Deep Learning Platform, Giant Leaps in Performance and Efficiency for AI Services, from the Data Center to the Network's Edge." https://images.nvidia.com/content/pdf/inference-technical-overview.pdf. Data retrieved on September 3, 2018.
3. Board power figures reported by board managment firmware during benchmark execution.

WP504_11_092418

図 11: GoogLeNet v1 Batch=1 の電力効率

ベンチマークの目的で GoogLeNet v1 のパフォーマンスを記載していますが、xDNN は幅広い CNN ネットワークに対応しています。その他の CNN ネットワークの動作に関する詳細は、ML Suite の資料 (https://github.com/Xilinx/ml-suite) を参照してください。

まとめ

前述のパフォーマンス結果に示すとおり、xDNN プロセッシング エンジンは高性能で電力効率に優れた DNN アクセラレータであり、リアルタイム推論ワークロードに使用される今日の一般的な多くの CPU/GPU プラットフォームよりも優れています。xDNN プロセッシング エンジンは、Amazon AWS/EC2 や Nimbix NX5 などの多くのクラウド環境で ML Suite を通して利用可能です。また、ザイリンクスの新しい Alveo アクセラレータ カードを使用して、オンプレミスの運用環境にシームレスに拡張できます。

ザイリンクスのリコンフィギュレーション可能な FPGA シリコンを使用すれば、xDNN のアップデートを通して、新たな機能改善や新機能を継続的に受け取ることができ、これによって、要件の変更やネットワークの進化にも遅れることなく対応できます。

利用開始にあたっての詳細は、次のページを参照してください。
https://github.com/Xilinx/ml-suite または https://japan.xilinx.com/applications/megatrends/machine-learning.html

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2018 年 10 月 14日	1.0	誤植の修正。
2018 年 10 月 2 日	1.0	初版

免責事項

自動車用のアプリケーションの免責条項