v.02 White Papers

rebellions_

リベリオンの ソフトウェアスタック

8月 05, 2024

序論

人工知能（AI）の急速な普及とあらゆる分野での活用により、短い処理時間と高いエネルギー効率を両立したハードウェアへの需要が増えています。AI加速器（AIチップ）はAIアルゴリズムの実行速度を高め、電力消費を減らします。既存のCPUやGPUより複雑な演算をより効率的に処理できるように設計されたハードウェアです。

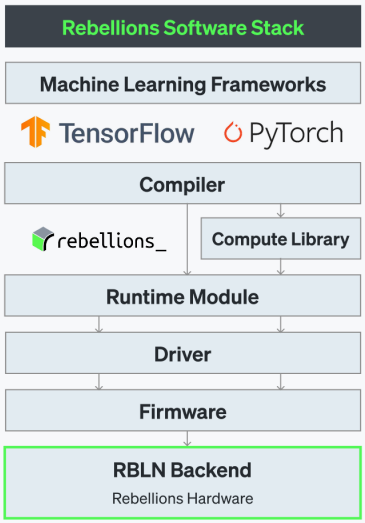
リベリオンのATOM™は、ディープラーニングのモデル処理を最適に行えるように設計されたSoC（System-on-Chip）です。8台の強力なニューラルエンジンが、高性能と処理時間の短縮向けに設計されたメモリアーキテクチャで演算を行います。このようなハードウェアの潜在力を最大限に発揮するには、ソフトウェアの最適化が欠かせません。

ソフトウェアは、ハードウェアが最大限の性能を発揮できるよう支える中核的な存在です。リソース管理をはじめ、データフローを最適化したり、アルゴリズムを効率よく実行できるように担います。また、GPUからリベリオンのチップへ移行するユーザーのために互換性を最大化し、統合しやすくします。リベリオンのソフトウェアスタックは、特に利便性と信頼性を優先しています。さらに、開発者とエンジニアがアクセスしやすくするために、包括的なユーザー向けドキュメントおよびSDKを提供しています。

本文書では、リベリオンのソフトウェアスタックの主なコンポーネントと中核的な機能を紹介します。また、**ATOM™**が実現した**圧倒的な演算性能と電力消費の大幅な削減**について説明します。

さらに、**YOLOv6物体検知（Object Detection）**モデルを対象にした**NVIDIA RTX A5000との性能を比較した結果**を提示し、**高性能かつ高効率のフルスタック（Full-Stack）AIソ**リューションを提供するリベリオンの持続的な革新を示します。

リベリオンのソフトウェアスタック



[Figure 1．リベリオンのソフトウェアスタック]

ATOM™が複雑なAIワークロードを処理するための強力な演算性能を提供するのに対し、リベリオンのソフトウェアスタックはこうしたハードウェアアーキテクチャの強みを最大化できるようにモデルの実行を最適化します。これにより、ATOM™の潜在力を効率よく最大限に発揮できるようにサポートします。

RBLN SDKはリベリオン独自のコンパイラ、演算ライブラリー（Compute Library）、ランタイム、ドライバー、ファームウェアで構成されています。また、TensorFlow、PyTorch、Hugging Faceなどさまざまなフレームワークで事前に学習した（pre-trained）モデルをリベリオンのチップ基盤のサービング環境で円滑に統合できるように設計されています。このソフトウェアスタックの全てのコンポーネントはレイテンシ（latency）を最小化するために高度に連動しており、高性能なAI推論環境向けの完成型統合プラットフォームを提供しています。

フレームワークへの対応

TensorFlow、PyTorchなど主要なフレームワークやHugging Faceモデルを含む200以上のリファレンスモデルおよびオペレーションに対応するRBLN SDKは、開発者がリベリオンのチップ環境へ円滑にマイグレーションできるようにサポートします。これにより、大型言語モデル（LLM）とディフュージョンモデルはもちろん、ビジョンおよび音声基盤のさまざまな人気モデルも円滑に駆動できます。

RBLNコンパイラ

RBLNコンパイラは、モデルをATOM™で実行できるコマンドに変換します。コンパイラはフロントエンド（Frontend Compiler）とバックエンド（Backend Compiler）の二つの主要なコンポーネントで構成されています。フロントエンドコンパイラは、ディープラーニングモデルを中間表現（Intermediate Representation、IR）に変換・最適化してから、これをバックエンドコンパイラに送ります。バックエンドコンパイラは、このIRをより最適化し、ハードウェアで実行できるコマンドストリーム（Command Stream）、プログラムバイナリ、さらに直列化した重み（Serialized Weights）を生成します。

1. フロントエンドコンパイラ

フロントエンドコンパイラは、モデルを統合したグラフ型のIRに変換します。変換したIRのノードは、バックエンドのコンパイラに送られる前に最適化過程を経ます。この過程には、不要な関数やノードの除去、ノードと重みのバインディング、データおよびカーネルロード・ストアオーバーヘッドを減らすためのノード結合、転送可能なノードへの注釈の追加、さらに効率的な並列化のためのグラフ分割が含まれます。

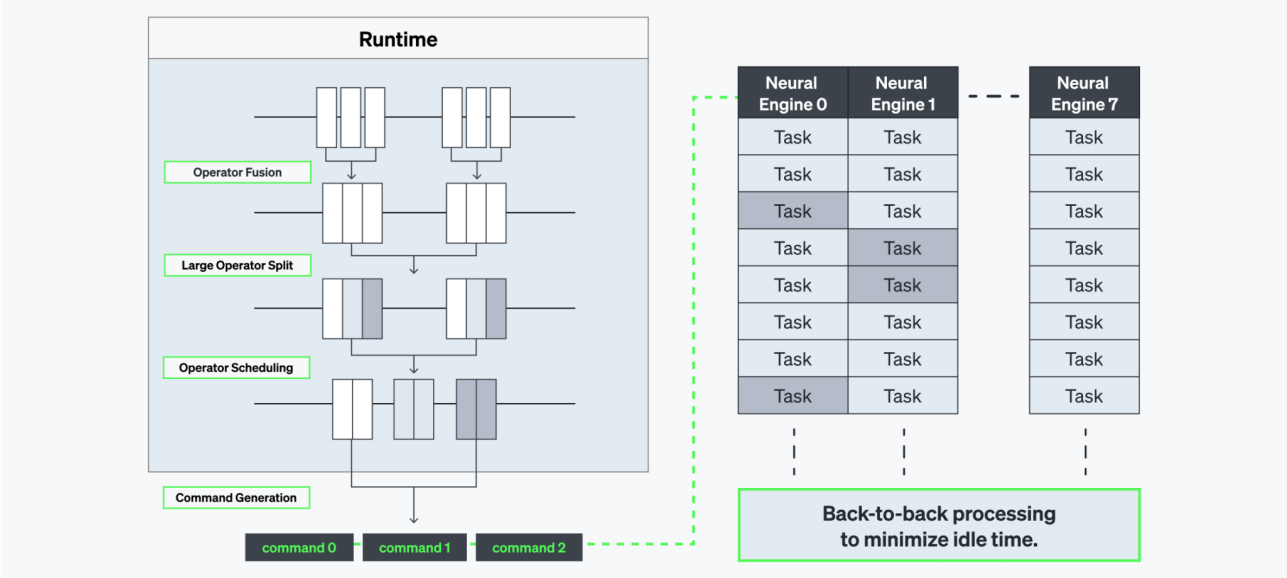
2. バックエンドコンパイラ

バックエンドコンパイラは、上位レベルのコードがハードウェアで効率的に実行されるようにサポートします。IRの演算を基に、演算とメモリ間のリソースを最適化し、稼働率を最大化するためにさまざまな技術を取り入れています。主な技術は以下の通りです。

- **分割（Partitioning）**：複数のカード構成を利用する場合は、モデルをより小さなコンポーネントに分け、さまざまなデバイスに分散して並列処理を効率よく行います。
- **融合（Fusion）**：演算子（operator）を併合して不要な中間活性化データの転送を減らし、Neural Engineで計算を最適化します。
- **分離（Splitting）**：デバイス内で演算を分けパイプライニングとスケジューリングを最適化し、性能を最大限に引き上げます。
- **タイリング（Tiling）**：分離した作業をNeural Engineの全体に分散し、最も効率的な性能を実現しています。

バックエンドコンパイラが生成する結果は以下の通りです。

1. **コマンドストリーム**：チップの複数のレイヤーからワークロードの実行を制御するコマンドプロセッサ（Command Processor）向けのコマンドセット
2. **プログラムバイナリ（Program Binary）**：演算ライブラリーのプログラムストリームを基にしたニューラルエンジン向けに高度に最適化したコマンド
3. **カーネルのシリアライズ（Kernel Serialization）**： FP32の重みをFP16に変換してニューラルエンジンに最適化した形式に直列化



[Figure 2. Model Compilation]

演算ライブラリー（Compute Library）

演算ライブラリーは、モデルの推論に必要な高度に最適化した低レベル演算（low-level operations）の集合で構成されています。これらの演算はニューラルエンジン内で算術論理演算器（ALU）のプログラマブルを構成する要素であり、コンパイラのコマンドに従いプログラムバイナリを用意します。

RBLN SDKは伝統的な畳み込みニューラルネットワーク（CNN）から、最新の生成AI（GenAI）モデルまで全てに対応できます。これには、数百個のGEMM（General Matrix Multiply）、正規化（normalization）、非線形活性化関数が含まれています。ニューラルエンジンの柔軟性が高いため、対応可能な低レベル演算の種類は今後も拡充され続け、さまざまなAIアプリケーションの高速化を実現しています。

ランタイムモジュール

ランタイムモジュールはコンパイルされたモデルとハードウェアの間の中間層で、実際のプログラム実行を管理します。コンパイラが生成した実行可能なコマンドを用意し、メモリとニューラルエンジン間のデータ転送を制御します。さらに、実行過程をモニタリングして性能を最適化します。

ドライバー

ドライバーはカーネルモードドライバー（KMD）とユーザーモードドライバー（UMD）で構成されており、ハードウェアへのアクセスを安全かつ効率的に提供します。KMDはOSがハードウェアを認識できるようにし、UMDが使えるAPIを表示します。また、コンパイラスタックで生成されたコマンドストリームをデバイスに送ります。UMDはユーザー空間で実行され、アプリケーションとハードウェア間の相互作用を管理する仲介役を担います。

ファームウェア

ファームウェアはATOM™の最も下位レベルのソフトウェアコンポーネントで、ソフトウェアとハードウェアを直接つなげる最終的なインタフェースの役割をします。SoC上にあるコマンドプロセッサの作業を制御しながら、メモリ層の全体において実際のAIワークロード（コマンドストリーム）を調整します。また、ハードウェアの状態をリアルタイムでモニタリングします。

ソフトウェアで最大化した柔軟なアーキテクチャ

ATOM™はさまざまなワークロードとアプリケーションに対応できるように設計された柔軟なアーキテクチャを備えています。

Future-Proofing

新しいモデルとアルゴリズムの登場により、新たな低レベル演算が必要になる可能性があります。 ATOM™はリベリオンによる継続的なソフトウェアアップデートにより、別途の専用ハードウェアがなくても、さまざまな演算に柔軟に対応できます。そのため、効率よく実行することができます。

性能及び効率性の最適化

モデルは段階別に違う演算（例えば、畳み込み、行列乗算、プーリング、活性化関数）が必要です。RBLNコンパイラは各段階の特性に合わせて、リソースを動的に割り当て性能を最適化します。

マルチデバイスの拡張性

大規模なGenAIワークロードでも安定的な性能を確保するために、上位ドライバやファームウェアがデバイス間通信を力強くサポートします。また、デバイス間通信のオーバーヘッドを最小化する戦略的なパーティショニングを通じて、性能を最適化します。

動的な構成（Dynamic Configuration）

応用プログラムによって速度やエネルギー効率などさまざまな優先順位があると思います。リベリオンのDynamic Voltage Frequency Scaling（DVFS）は、電力消費の最小化を目指し、特定のアプリケーションニーズに応じて柔軟に構成できます。

YOLOv6-Largeの性能比較

ATOM™はエネルギー効率よく設計されており、さまざまなモデルに柔軟に対応できます。これを立証するために、代表的なAIモデルであるYOLOv6を用いて推論を行いました。YOLOv6は物体を同時に分類し、検知する最新の畳み込みニューラルネットワーク（CNN）基盤の物体検知モデルです。従来のYOLOシリーズを改良・最適化した結果、正確かつ高速なリアルタイム検出を実現できる設計です。

テストはリベリオンのAI加速器であるATOM™を搭載したPCIeカードRBLN-CA12で行い、NVIDIA RTX A5000 GPUと性能を比較しました。比較した指標は以下の通りです。

- Watts：電力消費量。大規模なデプロイ環境では運営コストと効率性に直接影響を与えます。
- Joules per frame：全体のエネルギー効率を示す指標。フレーム当たりのエネルギー消費量を表します。

	ATOM™	A5000
Power Consumption (W)	avg. 40	avg. 110
Energy Consumption (J/Frame)	avg. 0.64	avg. 2.82

テーブル１の結果は、YOLOv6-Largeモデルを実行するにあたり、ATOM™の優れた性能を明確に表しています。ATOM™はRTX A5000対比最大2.1倍の性能と4.5倍のエネルギー効率を実現しました。

効率性と性能：徐々に増えるメリット

ハードウェアとソフトウェアレベルで、リソースの活用やワークロードのバランスを効率的に管理することで、チップの特定のエリアで過度に電力を消費したり、熱が集中的に発生する現象を防ぎます。

RBLNのソフトウェアスタックは同時性（concurrency）を効果的に制御し、チップのニューラルエンジンが提供する並列処理能力を最大限に引き上げます。また、データフローと実行パイプラインが最適化されており、常に処理するユニットにデータが供給されるように設計されています。さらに、低レベル演算の効率的な実現により、演算負荷とレイテンシ（latency）を減らしました。

こうした効率性と性能の最適化はビジネスの観点でも重要な意味を持っています。1,500W規模のサーバーはRBLN-CA12を16枚搭載できますが、GPUは4枚しか搭載できません。このように、個別のカード単位で確保した効率性は、多数のサーバーを運営する環境ではより大きな累積効果をもたらします。

結論

産業全体においてAIの活用事例が急激に増えているだけに、ハードウェアの適切な設計が欠かせません。しかし、時々見過ごされがちなもう一つの重要な要素は最適化したソフトウェアです。リベリオンのソフトウェアスタックは、ハードウェアの柔軟性と性能を最大化するために隠れた動力として働きます。

YOLOv6モデルを実行したRBLN-CA12とNVIDIA RTX A5000の性能を比較した結果は、最適化したソフトウェアと高性能のハードウェアの結合がもたらす性能優位性を明確に示しています。これは、リベリオンが最先端AIソリューションを提供するとの確固たる姿勢を示しています。