NVIDIA TESLA V100 GPU アーキテクチャ

世界最先端のデータセンター GPU

WP-08608-001_v1.1 | August 2017

目次

図一覧

表一覧

NVIDIA TESLA V100 GPU アーキテクチャ概論

10 年以上前に先駆的な CUDA GPU コンピューティング プラットフォームが登場して以来、NVIDIA® GPU は、世代を重ねるたびに、アプリケーション性能の向上、電力効率の向上、主要なコンピューティング新機能の追加、GPU プログラミングの簡素化を実現してきました。現在、NVIDIA GPU は、数千に及ぶ高性能コンピューティング (HPC) アプリケーション、データセンター アプリケーション、機械学習アプリケーションを高速化しています。NVIDIA GPU は、人工知能 (AI) 革命を支える最先端のコンピューティング エンジンとなりました。

NVIDIA GPU は、膨大な数のディープラーニング システムとアプリケーションの高速化を実現しています。自動運転プラットフォーム、高精度音声/画像/テキスト認識システム、創薬、医療診断、天気予報、ビッグ データ分析、金融モデリング、ロボット工学、工場自動化、リアルタイム翻訳、オンライン検索の最適化、パーソナライズ機能など、さまざまな分野で活用されています。

新しい NVIDIA® Tesla® V100 アクセラレータ (図 1) には、新しい強力な Volta™ GV100 GPU が組み込まれています。GV100 は、前世代の Pascal™ GP100 GPU が遂げた進化を基盤に、パフォーマンスとスケーラビリティが大幅に強化され、プログラミングを向上させるさまざまな新機能が追加されています。これにより、HPC、データセンター、スーパーコンピューター、ディープラーニング システムとアプリケーションはさらに強力に進化します。

このホワイト ペーパーでは、Tesla V100 アクセラレータと Volta GV100 GPU のアーキテクチャについて説明します。

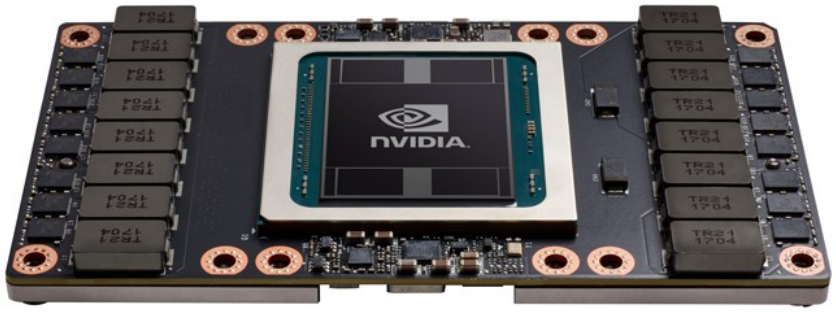


図 1. Volta GV100 GPU 搭載 NVIDIA Tesla V100 SXM2 モジュール

TESLA V100: AI コンピューティングと HPC の主戦力

NVIDIA Tesla V100 アクセラレータは、膨大な計算量の HPC、AI、グラフィックスなどのワークロードを強力にサポートするために設計された、世界最大のパフォーマンスを誇る並列プロセッサです。

GV100 GPU は、815 mm2 のダイ サイズに 211 億個のトランジスタが組み込まれています。製造には、NVIDIA 専用にカスタマイズされた新しい TSMC 12 nm FFN (FinFET NVIDIA) 高性能製造プロセスが用いられています。GV100 は、これまでの Pascal GPU と比較して計算性能が大幅に向上し、多くの新機能が追加されています。GPU プログラミングとアプリケーション移植のさらなる簡略化により、GPU リソース使用率も向上しています。これはきわめて電力効率の高いプロセッサであり、優れたワットあたりのパフォーマンスを発揮します。

主な機能

Tesla V100 の主なコンピューティング機能は次のとおりです。

- **ディープラーニングに最適化された新しいストリーミング マルチプロセッサ (SM) アーキテクチャ**
  Volta では、GPU の中核となる SM プロセッサ アーキテクチャが大幅に刷新されています。新しい Volta SM は、前世代の Pascal よりもエネルギー効率が 50%

も高く、同じパワー エンベロープ内の FP32 と FP64 のパフォーマンスが大幅に向上しています。ディープラーニング向けに特別に設計された新しい Tensor コアは、トレーニング時で最大 12 倍、推論時で最大 6 倍のピーク TFLOPS を実現します。整数と浮動小数点に並列の独立データ パスを使用する Volta SM は、コンピューティングとアドレス指定計算が混在するワークロードにおいても、非常に効率的です。新しい独立型スレッド スケジューリング機能は、並列スレッド間のより細かい同期と協調を可能にします。さらに、新しい内蔵 L1 データ キャッシュと共有メモリ ユニットにより、パフォーマンスが大幅に向上すると共にプログラミング処理が簡素化します。

- **第 2 世代の NVIDIA NVLink™**
  NVIDIA の第 2 世代 NVLink 高速インターコネクトは、マルチ GPU およびマルチ GPU/CPU システム構成向けに高い帯域幅、さらなるリンク、高いスケーラビリティを提供します。NVLink リンクが 4 つ、合計帯域幅が 160 GB/秒の GP100 に対し、Volta GV100 は最大 6 つの NVLink リンクと合計帯域幅 300 GB/秒をサポートしています。NVLink は、IBM POWER9 CPU ベースのサーバーで CPU マスタリング機能とキャッシュ コヒーレンス機能をサポートします。V100 AI スーパーコンピューター搭載の新しい NVIDIA DGX-1 は、NVLink を使用して、超高速ディープラーニング トレーニングのスケーラビリティを向上させます。
- **HBM2 メモリ: 高速、高効率**
  高度に調整された Volta 32 GB HBM2 メモリ サブシステムは、900 GB/秒のピーク メモリ帯域幅を実現します。Samsung の新世代 HBM2 メモリと Volta の新世代メモリ コントローラーの組み合わせにより、メモリ帯域幅は Pascal GP100 の 1.5 倍となり、メモリ帯域幅使用率を最大 95% 向上させて多数のワークロードを実行できます。
- **Volta マルチプロセス サービス**
  Volta マルチプロセス サービス (MPS) は Volta GV100 アーキテクチャの新機能です。CUDA MPS サーバーの重要なコンポーネントにハードウェア アクセラレーションを提供することで、GPU を共有する複数のコンピューティング アプリケーションのパフォーマンス、分離性、サービス品質 (QoS) が向上します。Pascal の MPS クライアント最大数が 16 個であるのに対し Volta はその 3 倍の 48 個となります。

- **拡張統合メモリおよびアドレス変換サービス**
  GV100 統合メモリ テクノロジには新しいアクセス カウンターが組み込まれています。メモリ ページを頻繁にアクセスするプロセッサに正確に移動できるため、プロセッサ間で共有されるメモリ範囲の効率も向上します。IBM Power プラットフォームでは、新しいアドレス変換サービス (ATS) により、GPU が CPU のページ テーブルに直接アクセスできます。
- **最大パフォーマンス モードと最大効率モード**
  最大パフォーマンス モードでは、Tesla V100 アクセラレータが最大 300 W レベルの TDP (熱設計電力) で動作し、計算速度とデータ スループットを必要とするアプリケーションを高速化します。最大効率モードでは、データセンター管理者が、最適なワットあたりのパフォーマンスになるように電力量を調整できます。ラック内のすべての GPU に電力の上限を設定することで、優れたラック性能を維持しつつ、消費電力を劇的に削減できます。
- **Cooperative Groups と新しい Cooperative Launch API**
  Cooperative Groups は、通信スレッドをグループ管理するために CUDA 9 で導入された新しいプログラミング モデルです。開発者は、Cooperative Groups を使用してスレッドの通信粒度を表現し、より多機能で効率的な並列分割を実現できます。Cooperative Groups の基本機能は、Kepler 以降のすべての NVIDIA GPU でサポートされています。Pascal と Volta は、CUDA スレッド ブロック間の同期をサポートする新しい Cooperative Launch API に対応しています。Volta では新しい同期パターンがサポートされています。
- **Volta 最適化ソフトウェア**
  Caffe2、MXNet、TensorFlow などの最新バージョンのディープラーニング フレームワークは、Volta を利用して、トレーニング時間を劇的に短縮し、マルチノード トレーニングのパフォーマンスをさらに向上させています。GPU アクセラレーション ライブラリの中でも cuDNN、cuBLAS、TensorRT などの Volta に最適化されたバージョンは、Volta GV100 アーキテクチャの新機能を活用して、ディープラーニング推論と高性能コンピューティング (HPC) アプリケーションの両方に高いパフォーマンスを発揮します。NVIDIA CUDA Toolkit バージョン 9.0 に追加された新しい API と Volta 機能のサポートにより、プログラミングはさらに容易になっています。

図 2 は、Tesla V100 に組み込まれた新しいテクノロジです。

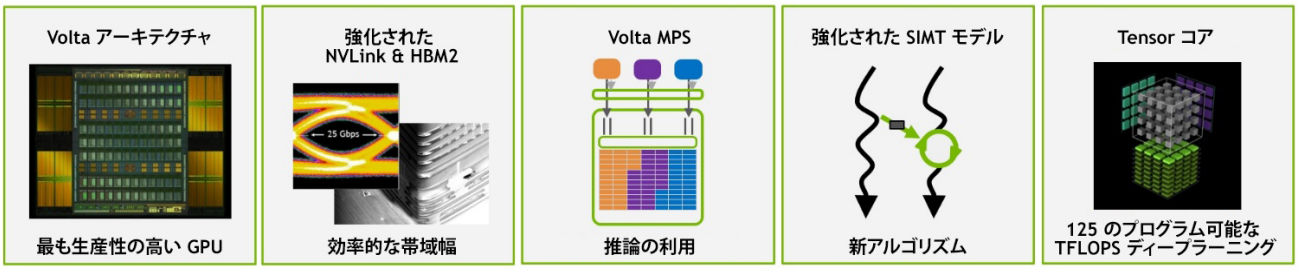


図 2. Tesla V100 の新しいテクノロジ

AI および HPC 向けの究極のパフォーマンス

Tesla V100 は、浮動小数点演算と整数演算で業界最大のパフォーマンスを実現します。以下はピーク時の計算速度です。図 3 は、新しい Tensor コアを使用した Tesla V100 のディープラーニング性能を示しています。

- 7.8 TFLOPS[1] の倍精度浮動小数点 (FP64) 演算能力
- 15.7 TFLOPS[1] の単精度 (FP32) 演算能力
- 125 Tensor TFLOPS[1]

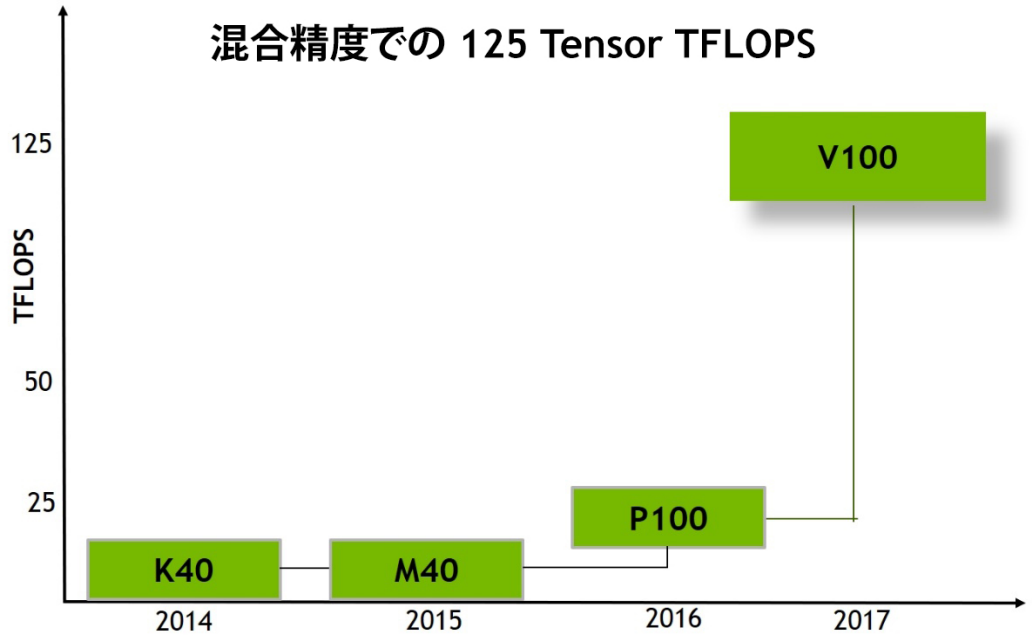


図 3. 新しい Tensor コアによって飛躍的に向上した Tesla V100 のディープラーニング性能

[1] GPU Boost クロック基準

NVIDIA GPU - 最高の柔軟性を備えた最速のディープラーニング プラットフォーム

ディープラーニング トレーニングや推論演算において、GPU アクセラレーションはシングル GPU とマルチ GPU のどちらのシステムにも大きなメリットとなります。NVIDIA Pascal GPU は、この 1 年でディープラーニング システムの高速化に幅広く使用されており、トレーニングおよび推論で CPU のスピードを驚異的に超えています。ディープラーニング向けの新しいアーキテクチャに加えて、NVIDIA Tesla V100 GPU の計算性能が強化されたことで、ニューラル ネットワークのトレーニングと推論のパフォーマンスがさらに向上しました。さらに、マルチ GPU システムと NVLink の組み合わせにより、パフォーマンス スケーラビリティも大きく進化しています。

柔軟な GPU プログラミング性により、新しいアルゴリズムを迅速に開発して展開できます。NVIDIA GPU は、高いパフォーマンス、スケーラビリティ、プログラミング性により、AI、ディープラーニング システム、トレーニング/推論アルゴリズムの継続的なニーズに応えます。

ディープラーニングの背景

人間の知性をモデル化するために、人工知能の分野では長年さまざまなアプローチが採用されてきました。判断や結果予測ができるようにシステムをトレーニングする機械学習も、主要な AI 手法です。ディープラーニングは、人間の脳の神経学習プロセスに着想を得て開発された機械学習法です。ディープラーニングは、相互に接続された多数

の人工ニューロン (パーセプトロンとも呼ばれる) が何層にも積み重なったディープ ニューラル ネットワーク (DNN) を使用します。DNN を膨大な量の入力データでトレーニングすることで、複雑な問題を高精度で迅速に解決できるようになります。トレーニングされたニューラル ネットワークを推論と呼ばれるプロセスで使用して、オブジェクトの識別やパターンの分類を行います。ニューラル ネットワークの動作について、詳しくはこのホワイト ペーパーの付録 C をご覧ください。

ほとんどのニューラル ネットワークは、相互に接続された複数のニューロン層で構成されます。各ニューロンや層でトレーニングされたネットワークのタスクを実行します。たとえば、2012 ImageNet コンテストで優勝した畳み込みニューラル ネットワーク (CNN) の AlexNet は、8 つの層、65 万個の相互接続ニューロン、約 6,000 万個のパラメーターで構成されています。現在のニューラル ネットワークは著しく複雑化しており、深層残差ネットワーク (例: ResNet-152) などでは 150 以上の層、数百万個以上の接続ニューロンとパラメーターで構成されます。

GPU アクセラレーション ディープラーニング

従来の CPU ベースのプラットフォームよりも高速でエネルギー効率が良い NVIDIA GPU は、ディープ ニューラル ネットワークのトレーニング向け最先端エンジンに最適であると、学界や産業界で広く認知されています。多数の同一ニューロンから成るニューラル ネットワークは、高度に並列化されているという特性があります。これが GPU に自然にマッピングされることで、単独の CPU よりも高速なトレーニングが実現します。

ニューラル ネットワークは行列数値演算に大きく依存し、複雑な多層ネットワークは、効率と速度の両面で膨大な量の浮動小数点演算能力と帯域幅を必要とします。GPU は、行列数値演算用に最適化された数千個のプロセッシング コアを備えており、数十から数百 TFLOPS のパフォーマンスを発揮します。そのため、ディープ ニューラル ネットワークに基づく人工知能や機械学習アプリケーションに最適のコンピューティング プラットフォームと言えます。

Volta のアーキテクチャは、ディープラーニング ワークロードの実行に特化されており、前世代のアーキテクチャと変わらない電力量でパフォーマンスの大幅な向上を実現します。この技術的なしくみは、次のアーキテクチャのセクションで説明しています。.

GV100 GPU ハードウェア アーキテクチャの詳細

Volta GV100 GPU を搭載した NVIDIA Tesla V100 アクセラレータは、現在、世界最大のパフォーマンスを誇る並列コンピューティング プロセッサです。GV100 は、HPC システムおよびアプリケーションで強力なコンピューティング能力を発揮するだけでなく、ディープラーニング アルゴリズムおよびフレームワークを大幅に高速化する重要な革新的ハードウェアを備えています。

Pascal GP100 GPU と同様に、GV100 GPU は、複数の GPU 処理クラスター (GPC)、テクスチャ処理クラスター (TPC)、ストリーミング マルチプロセッサ (SM)、メモリ コントローラーで構成されています。GV100 GPU のフル構成は次のとおりです。

- GPC x 6
  各 GPC の構成:
  - TPC x 7 (各 TPC に 2 個の SM)
  - SM x 14
- Volta SM x 84
  各 SM の構成:
  - FP32 コア x 64
  - INT32 コア x 64
  - FP64 コア x 32
  - Tensor コア x 8
  - テクスチャ ユニット x 4

- 512 ビット メモリ コントローラー x 8 (合計 4,096 ビット)

フル GV100 GPU は 84 個の SM を搭載し、合計 5,376 個の FP32 コア、5,376 個の INT32 コア、2,688 個の FP64 コア、672 個の Tensor コア、336 個のテクスチャ ユニットを備えています。各 HBM2 DRAM スタックは、1 組のメモリ コントローラーによって制御されます。フル GV100 GPU は、合計 6,144 KB の L2 キャッシュを搭載しています。**図 4** は、84 個の SM を搭載したフル GV100 GPU を示しています (GV100 の構成は製品によって異なります)。Tesla V100 アクセラレータは 80 個の SM を使用しています。**表 1** は、過去 5 年間の NVIDIA Tesla GPU の比較です。

図 4. 個の SM ユニットを搭載した Volta GV100 フル GPU

表 1. NVIDIA Tesla GPU の比較

Tesla 製品	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SM 数	15	24	56	80
TPC 数	15	24	28	40
FP32 コア数/SM	192	128	64	64
FP32 コア数/GPU	2,880	3,072	3,584	5,120
FP64 コア数/SM	64	4	32	32
FP64 コア数/GPU	960	96	1792	2560
Tensor コア数/SM	なし	なし	なし	8
Tensor コア数/GPU	なし	なし	なし	640
GPU Boost クロック	810/875 MHz	1,114 MHz	1,480 MHz	1,530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	なし	なし	なし	125
テクスチャ ユニット数	240	192	224	320
メモリ インターフェイス	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
メモリ サイズ	最大12 GB	最大24 GB	16 GB	32 GB
L2 キャッシュ サイズ	1,536 KB	3,072 KB	4,096 KB	6,144 KB
共有メモリ サイズ/SM	16 KB/32 KB/48 KB	96 KB	64 KB	最大 96 KB まで構成可能
レジスタ ファイル サイズ/SM	256 KB	256 KB	256 KB	256KB
レジスタ ファイル サイズ/GPU	3,840 KB	6,144 KB	14,336 KB	20,480 KB
TDP	235 ワット	250 ワット	300 ワット	300 ワット
トランジスタ数	71 億	80 億	153 億	211 億
GPU ダイ サイズ	551 mm^2	601 mm^2	610 mm^2	815 mm^2
製造プロセス	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] ピーク TFLOPS レートは GPU Boost クロック基準

究極のパフォーマンスと効率

NVIDIA の GPU は、世代を重ねるたびにパフォーマンスが大幅に向上し、エネルギー効率も改善しています。Tesla V100 は、最大限のパフォーマンスまたはエネルギー効率が最も良いパフォーマンスのどちらにも構成可能で、データセンター設計者に新次元の柔軟性を提供します。この 2 つのモードを最大パフォーマンス モード、最大効率モードと呼びます。

最大パフォーマンス モードでは、Tesla V100 アクセラレータが最大 300 W の TDP レベルで動作して、最高の計算速度とデータ スループットを必要とするアプリケーションを高速化します。

最大効率モードでは、データセンター管理者が最適なワットあたりのパフォーマンスで Tesla V100 アクセラレータを実行できます。V100 は、最大のパフォーマンスと最大の電力効率を実現する電力/パフォーマンス曲線に沿って設定できます。たとえば、曲線上の TDP の最大効率が 50 ～ 60% であるときに、GPU は最大 75 ～ 85% のパフォーマンスを発揮できます。データセンター管理者は、ラック内のすべての GPU に電力上限を設定して、優れたラック性能を維持しながら消費電力を大幅に削減できます。この機能により、データセンター設計者は、ラックの電力範囲でパフォーマンスを最大限に引き出すことができます。この最適化は、サーバー ノードをラックに追加するのと同等の効果がある場合もあります。

電力制限は、NVIDIA-SMI (データセンター管理者用コマンドライン ユーティリティ) または NVML (Tesla OEM パートナーが自社ツールセットに統合可能な電力制限コントロールを提供する C ベースの API ライブラリ) で設定できます。最大効率モードは、通常の動作でピーク クロックやメモリ クロックを低下させるのではなく、電力制限範囲内の最大クロック速度で GPU が動作するようにします。ほとんどのワークロードは 300 W TDP をすべて消費することはないため、電力を大幅に制限できる場合もあります。ただし、データセンター設計者は、ラックの電力上限を超えないように、予想される最大のワークロードに基づいて GPU の電力レベルを設定する必要があります。

VOLTA ストリーミング マルチプロセッサ

Volta は、パフォーマンス、エネルギー効率、プログラミング性が大幅に向上した新しいストリーミング マルチプロセッサ (SM) アーキテクチャを採用しています。

主な特長は次のとおりです。

- ディープラーニング行列演算専用の新しい混合精度 Tensor コアにより、GP100 の 12 倍の TFLOPS を実現 (同じパワー エンベロープでのトレーニング時)
- 一般的なコンピューティング ワークロードのエネルギー効率を 50% 向上
- 強化された高性能 L1 データ キャッシュ
- 以前の SIMT/SIMD プロセッサ設計の限界を超えた新しい SIMT スレッド モデル

Pascal GP100 と同様に、GV100 SM では、各 SM に 64 個の FP32 コアと 32 個の FP64 コアが組み込まれています。ただし、GV100 SM は、新しいパーティショニング方法を使用して SM 使用率と全体的なパフォーマンスを向上させています。GP100 SM は 2 つの処理ブロックにパーティション分割され、それぞれに FP32 コアが 32 個、FP64 コアが 16 個、命令バッファーが 1 つ、Warp スケジューラが 1 つ、ディスパッチ ユニットが 2 つ、128 KB レジスタ ファイルが 1 つあります。一方、GV100 SM は 4 つの処理ブロックにパーティション分割され、それぞれに FP32 コアが 16 個、FP64 コアが 8 個、INT32 コアが 16 個、新しいディープラーニング行列演算用の混合精度 Tensor コアが 2 個、新しい L0 命令キャッシュが 1 つ、Warp スケジューラが 1 つ、ディスパッチ ユニットが 1 つ、64 KB レジスタ ファイルが 1 つあります。新しい L0 命令キャッシュが各パーティションで使用されるようになり、従来の NVIDIA GPU の命令バッファーより高い効率で動作します (**図 5** の Volta SM を参照)。

GV100 の SM には Pascal GP100 の SM と同じ数のレジスタがありますが、GV100 GPU の SM 数がはるかに多いため、合計レジスタ数も増加します。総合的に見ると、GV100 は、従来の世代の GPU より多くのスレッド、Warp、スレッド ブロックをサポートしています。

共有メモリと L1 リソースを統合することで、GP100 の 64 KB の共有メモリ容量に対して Volta SM では 96 KB に増やすことができます。

図 5. Volta GV100 ストリーミング マルチプロセッサ (SM)

Tensor コア

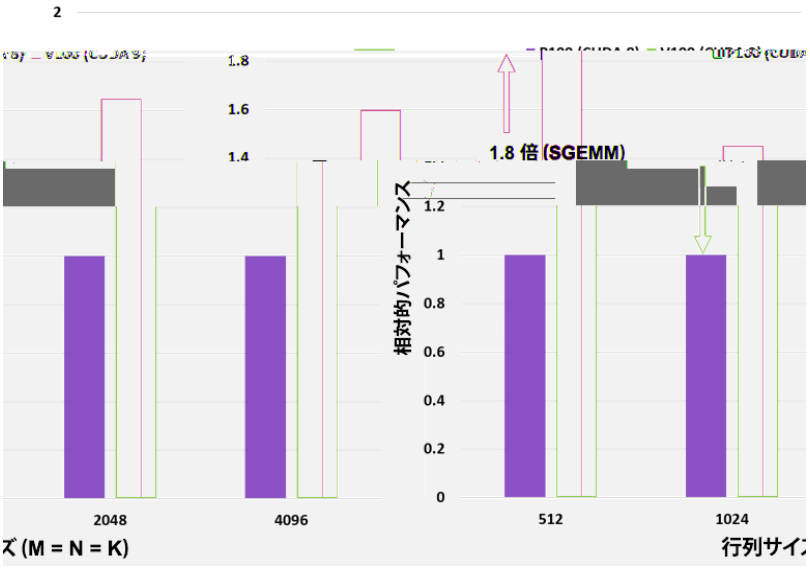
Tesla P100 は、ニューラル ネットワークのトレーニングにおいて、前世代の NVIDIA Maxwell や Kepler アーキテクチャよりも飛躍的に高いパフォーマンスを実現しましたが、同時にニューラル ネットワークの複雑性とサイズも増しています。数百万個のニューロンが数千層に重なる新しいネットワークには、さらに高いパフォーマンスと高速なトレーニングが求められます。

Tensor コアの新機能は、Volta GV100 GPU アーキテクチャが大規模なニューラル ネットワークのトレーニングに必要なパフォーマンスを発揮する鍵となります。

Tesla V100 GPU には、SM 内の各処理ブロック (パーティション) に 2 個 (各 SM に 8 個)、合計 640 個の Tensor コアが含まれています。Volta GV100 では、各 Tensor コアがクロックあたり 64 回の浮動小数点 FMA 演算を実行し、1 SM 内の 8 個の Tensor コアがクロックあたり合計 512 回の FMA 演算 (または 1,024 回の個別浮動小数点演算) を実行します。

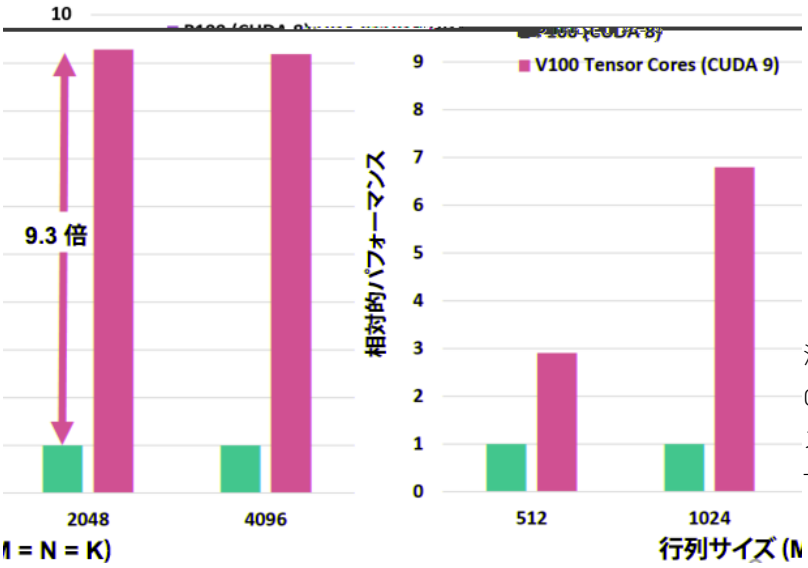
Tesla V100 の Tensor コアは、トレーニングおよび推論アプリケーションにおいて最大 125 Tensor TFLOPS を発揮します。これは、P100 での標準的な FP32 演算と比較して最大 12 倍のピーク TFLOPS となります。ディープラーニング推論の場合、V100 Tensor コアは、P100 での標準的な FP16 演算と比較して最大 6 倍のピーク TFLOPS を発揮します。

行列-行列積 (GEMM) 演算は、ニューラル ネットワークのトレーニングおよび推論の中核となる処理です。何層にもわたって接続されたネットワークで、入力データと重み付けで構成された大規模な行列どうしを乗算します。単精度の行列積を使用するアプリケーションの場合、CUDA 9 搭載 Tesla V100 は、**図 6** のとおり UDA 8 搭載 Tesla P100 の 1.8 倍のパフォーマンスを発揮します。半精度入力の行列積によるトレーニングおよび推論演算の場合、**図 7** の FP16 入力/FP32 和行列演算において、Volta の混合精度 Tensor コアは、P100 の 9 倍以上のパフォーマンスを実現しています。



単精度 (FP32) 行列-行列積では、CUDA 9 搭載 Tesla V100 のスピードが最大で CUDA 8 搭載 Tesla 100 の 1.8 倍

図 6. cuBLAS 単精度 (FP32)



混合精度行列-行列積では、CUDA 9 搭載 Tesla V100 のスピードが CUDA 8 搭載 Tesla P100 の FP32 の 9 倍以上

図 7. cuBLAS 混合精度 (FP16 入力、FP32 コンピューティング)

Tensor コアと関連するデータ パスは、高いエネルギー効率で浮動小数点演算のスループットを劇的に増加できるようにカスタム設計されています。

各 Tensor コアは 4 x 4 行列に対して次の演算を実行します。

D = A×B + C

ここで、A、B、C、D はそれぞれ 4 x 4 行列です (図 8)。行列積の入力 A および B は FP16 行列、行列和の C および D は FP16 行列または FP32 行列です (**図 8** 参照).

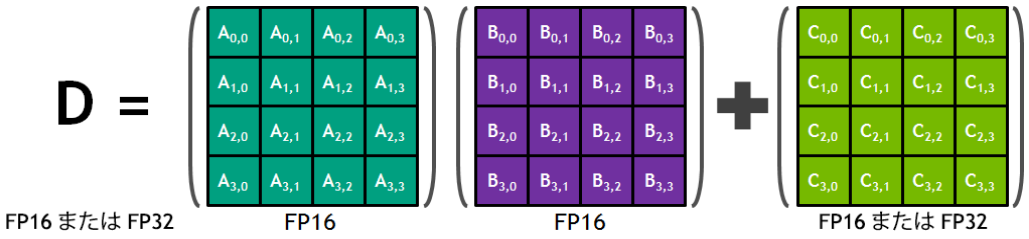


図 8. Tensor コア 4 x 4 行列積和演算

Tensor コアは、FP16 入力データに対して FP32 和演算を行います。FP16 乗算の結果は完全精度の積になり、それに他の中間積結果が FP32 和演算されて、4 x 4 x 4 行列積になります (**図 9**を参照)。実際、Tensor コアがこれらの小さな要素で構成されている大きな 2 次元以上の行列演算を実行します。

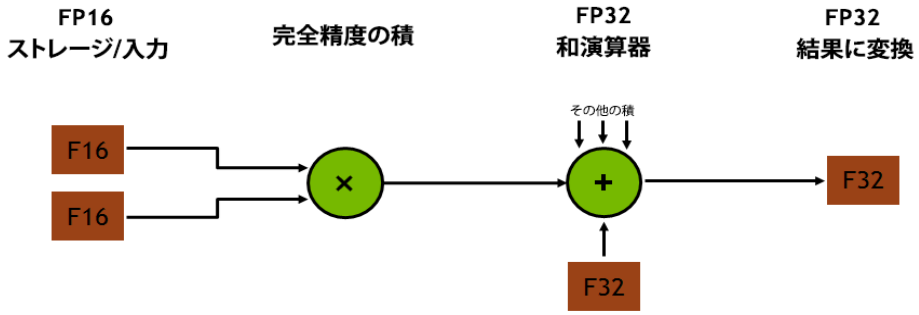


図 9. Tensor コアでの混合精度積和演算

図 10 は、4 x 4 行列積 (キューブの外にある 2 つの 4 x 4 入力行列) によって 4 x 4 出力行列 (キューブの下に表示) を生成するために 64 回の演算 (キューブ) を必要とするようすを示しています。Tensor コア搭載 Volta ベース V100 アクセラレータは、このような計算を Pascal ベース Tesla P100 の 12 倍のスピードで行うことができます。

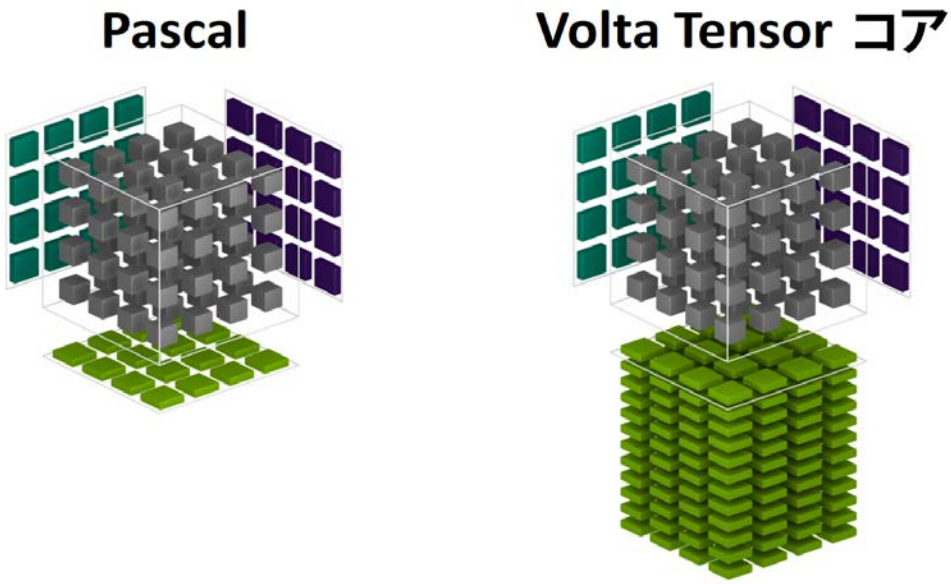


図 10. Pascal および Volta による 4 x 4 行列積

Volta Tensor コアは、Warp レベル行列演算として CUDA 9 C++ API で公開されてアクセス可能です。この API は、CUDA-C++ プログラムから Tensor コアを効率的に使用するために、専用の行列ロード演算、行列積和演算、行列ストア演算を公開しています。CUDA レベルでは、Warp レベル インターフェイスは、Warp 内の 32 スレッドすべてにまたがる 16 x 16 サイズの行列を前提としています。

Tensor コアを直接プログラムする CUDA-C++ インターフェイスに加えて、cuBLAS ライブラリと cuDNN ライブラリが更新されています。これらは、ディープラーニング アプリケーション/フレームワーク用に Tensor コアを使用するための新しいライブラリ インターフェイスを提供します。NVIDIA は、Volta GPU ベースのシステムでディープラーニング研究に Tensor コアを使用できるように、Caffe2、MXNet などの多くの一般的なディープラーニング フレームワークと協力してきました。NVIDIA は、他のフレームワークでも Tensor コアがサポートされるように取り組んでいます。

拡張 L1 データ キャッシュと共有メモリ

Volta SM の内蔵 L1 データ キャッシュと共有メモリ サブシステムは、パフォーマンスを大幅に向上させると共に、プログラミングを簡略化し、最高のアプリケーション パフォーマンスの実現に必要なチューニングを削減します。

データ キャッシュと共有メモリの機能を 1 つのメモリ ブロックで組み合わせることで、両方のタイプのメモリ アクセスが全体として最高の性能を発揮します。両方を合わせた容量は 128 KB/SM で、GP100 データ キャッシュの 7 倍以上になり、共有メモリを使用しないプログラムでは、そのすべてをキャッシュとして使用できます。テクスチャ ユニットもキャッシュを使用します。たとえば、共有メモリが 64 KB に設定されている場合、テクスチャ演算とロード/ストア演算で L1 の残り 64 KB を使用できます。

Volta GV100 は、L1 キャッシュを共有メモリ ブロックと統合することで、これまでの NVIDIA GPU の L1 キャッシュよりはるかに低遅延、高帯域幅になります。Volta の L1 は、データをストリーミングするための高スループットな導管として機能すると同時に、頻繁に再利用されるデータが高帯域幅および低遅延でアクセスできるという特長があります。この組み合わせは Volta 独自のものであり、これまでよりも使いやすくなっています。

GV100 で L1 データ キャッシュと共有メモリを統合した主な理由は、共有メモリのパフォーマンス メリットを L1 キャッシュ操作でも活用するためです。共有メモリは高帯域幅、低遅延、安定性能 (キャッシュ ミスなし) を提供しますが、CUDA プログラマがこのメモリを明示的に管理する必要があります。Volta は、共有メモリを明示的に管理するアプリケーションと、デバイス メモリ内のデータに直接アクセスするアプリケーション間のパフォーマンス差を縮めます。これを実証するために、共有メモリ アレイをデバイス メモリ アレイに置き換えて、アクセスが L1 キャッシュを通過するようにプログラムを変更しました。図 11 に示すように、共有メモリを使用せずにこのコードを実行すると、Volta での 7% のパフォーマンス低下に対して、Pascal では 30% の低下となりました。共有メモリはパフォーマンス向上のための重要な要素ですが、新しく設計された Volta L1 を利用することで、プログラミングに労力をかけずに優れたパフォーマンスを迅速に引き出せるようになります。

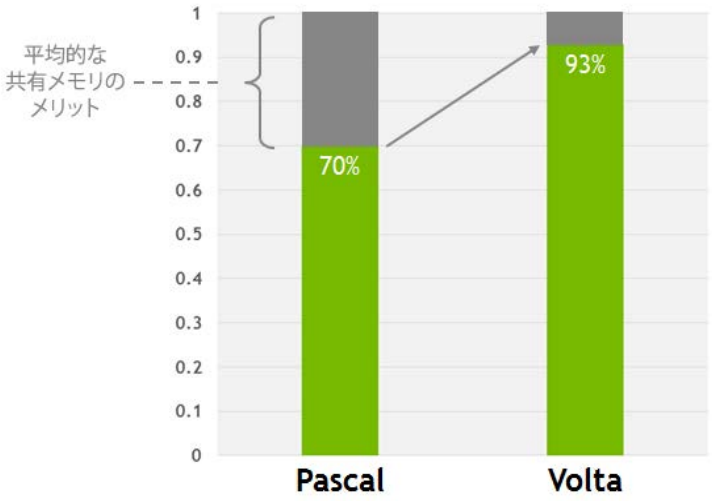


図 11. Pascal と Volta のデータ キャッシュ比較

GV100 L1 キャッシュは、共有メモリの効果が低い、または共有メモリを利用できないといった状況下でのパフォーマンス改善に役立ちます。Volta GV100 は、共有メモリと L1 の統合によってグローバル メモリへの高速パスを提供し、無制限のキャッシュ ミス アクセスも即座にストリーミングできます。従来の NVIDIA GPU はロード キャッシュのみでしたが、GV100 はライト キャッシュ (ストア演算のキャッシュ) を導入してパフォーマンスをさらに向上させました。

FP32 演算と INT32 演算の同時実行

FP32 命令と INT32 命令を同時に実行できない Pascal GPU とは異なり、FP32 コアと INT32 コアが別々に組み込まれている Volta GV100 SM では、FP32 演算と INT32 演算をフル スループットで同時実行でき、命令発行スループットも向上します。コア FMA (融合積和) 演算では、依存した命令発行の遅延も短縮され、6 クロック サイクルが必要だった Pascal に対して Volta では 4 クロック サイクルで済みます。

多くのアプリケーションは、内部ループでポインター演算 (整数メモリ アドレス演算) と浮動小数点計算を組み合わせて実行しているため、FP32 命令と INT32 命令を同時に実行できるのはメリットです。パイプライン ループの反復ごとに、アドレスを更新し (INT32 ポインター演算)、次の反復処理に使用するデータをロードしながら、同時に FP32 で現在の反復処理を行うことができます。

COMPUTE CAPABILITY

GV100 GPU は、新しい Compute Capability 7.0 をサポートしています。表 2 は、さまざまな NVIDIA GPU アーキテクチャにおける Compute Capability のパラメーターの比較です。

表 2. Compute Capability の比較: GK180 vs GM200 vs GP100 vs GV100

GPU	Kepler GK180	Maxwell GM200	Pascal GP100	Volta GV100
Compute Capability	3.5	5.2	6.0	7.0
スレッド数/Warp	32	32	32	32
最大 Warp 数/SM	64	64	64	64
最大スレッド数/SM	2,048	2,048	2,048	2,048
最大スレッド ブロック数/SM	16	32	32	32
最大 32 ビット レジスタ数/SM	65,536	65,536	65,536	65,536
最大レジスタ数/ブロック	65,536	32,768	65,536	65,536
最大レジスタ数/スレッド	255	255	255	255[1]
最大スレッド ブロック サイズ	1,024	1,024	1,024	1,024
FP32 コア数/SM	192	128	64	64
SM レジスタ数と FP32 コア数の比率	341	512	1024	1024
共有メモリ サイズ/SM	16 KB/32 KB/ 48 KB	96 KB	64 KB	最大 96 KB まで構成可能

[1]強化 SIMT モデルに含まれるスレッド単位プログラム カウンター (PC) は、通常、スレッドごとに 2 つのレジスタ スロットを必要とします。

NVLINK: 高帯域幅、リンク数と機能を拡張

NVLink は、Tesla P100 アクセラレータや Pascal GP100 GPU と共に 2016 年に初めて導入された NVIDIA の高速相互接続テクノロジです。NVLink は、GPU 間と GPU-CPU 間の両方のシステム構成において、PCIe 相互接続よりもはるかに優れたパフォーマンスを提供します。NVLink テクノロジの基本情報については、Pascal アーキテクチャ ホワイトペーパー (英語) をご覧ください。Tesla V100 には第 2 世代の NVLink が導入されており、リンク速度がさらに上がり、GPU あたりのリンク数が増加し、CPU マスタリング、キャッシュ コヒーレンス、スケーラビリティも強化されています。

追加のリンクと高速化

開発者が AI コンピューティングなどのアプリケーションで並列処理を活用するようになり、さまざまな業界で複数の GPU と CPU で構成されたシステムが一般化しています。こういったトレンドの中、マルチプロセッサ相互接続のさらなる高速化とスケーラビリティへのニーズが高まっています。同様に、さらに規模が拡大する問題の解決に向けて、数万以上の計算ノードで構成される高性能 GPU アクセラレーション システムが、データセンター、研究施設、スーパーコンピューターに導入されています。P100 や V100 を搭載した NVIDIA 独自の DGX-1 システムには、NVLink テクノロジが導入されています。2016 年には、NVIDIA は IBM と緊密に協力して、NVIDIA Pascal GPU と IBM POWER8+ CPU の両方を使用する高性能サーバーを構築しました。現在は IBM と共に、Tesla V100 アクセラレータと POWER9 CPU を NVLink で接続して使用する、さらに高性能のサーバーを構築しています。

Pascal の NVLink の信号速度は 20 ギガビット/秒でしたが、V100 の NVLink では 25 ギガビット/秒に向上しています。現在、各リンクの速度は各方向に 25 ギガビット/秒になっています。サポート対象リンク数は 4 から 6 に増え、GPU NVLink 帯域幅は 300 GB/秒になりました。これらのリンクは、**図 12**に示される V100 搭載 DGX-1 トポロジで GPU 間通信専用に使用できるほか、**図 13** に示される GPU 間通信と GPU-CPU 間通信の組み合わせにも使用できます。

追加機能

第 2 世代の NVLink は、CPU から各 GPU の HBM2 メモリへ直接ロード、ストア、アトミック アクセスを行うことができます。新しい CPU マスタリング機能と共に、NVLink は、グラフィックス メモリから読み取ったデータを CPU のキャッシュ階層に格納するコヒーレンシ操作をサポートしています。CPU パフォーマンスでは、CPU キャッシュへのアクセスの遅延が少ないことが重要です。P100 は、ピア GPU アトミックをサポートしていますが、NVLink からターゲット CPU に送信される GPU アトミックはサポートしていませんでした。今回、GPU または CPU からのアトミックをサポートしました。また、アドレス変換サービス (ATS) をサポートし、GPU が CPU のページ テーブルに直接アクセスできるようになりました。新しいリンクの低電力モード動作により、使用頻度が低いときに電力を大幅に節約できるようになります (**図 14**を参照)。

第 2 世代の NVLink のリンク数の増加、リンクの高速化、機能強化を Volta の新しい Tensor コアと組み合わせた結果、マルチ GPU Tesla V100 システムのディープラーニング性能は Tesla P100 GPU 搭載システムよりも大幅に向上しました。

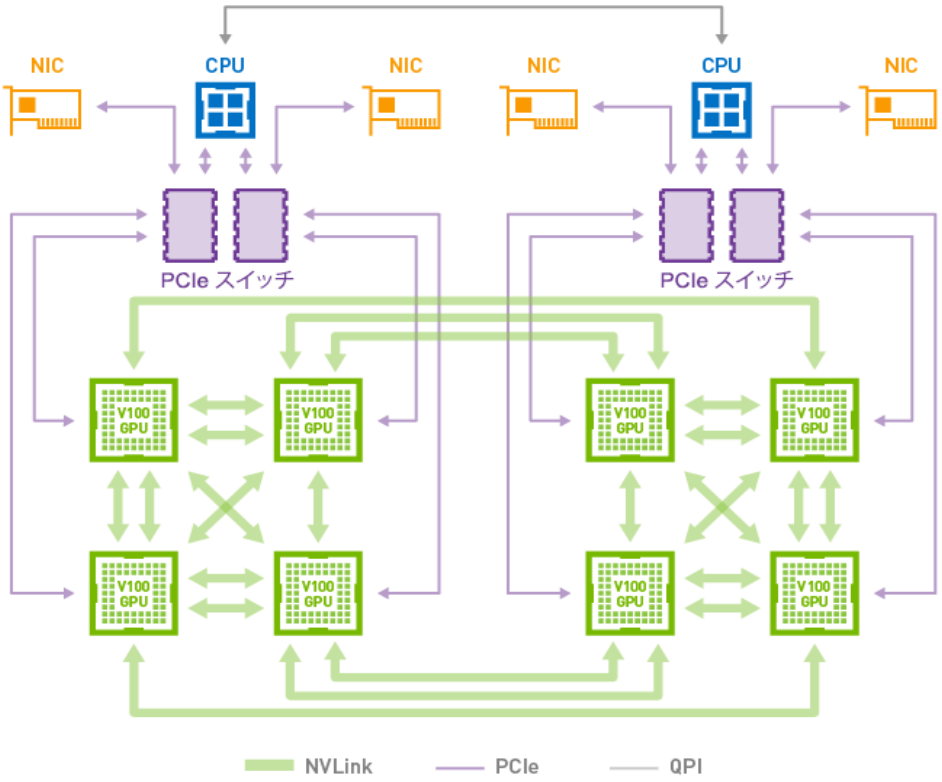


図 12. V100 搭載 DGX-1 で使用されるハイブリッド キューブ メッシュ NVLink トポロジ

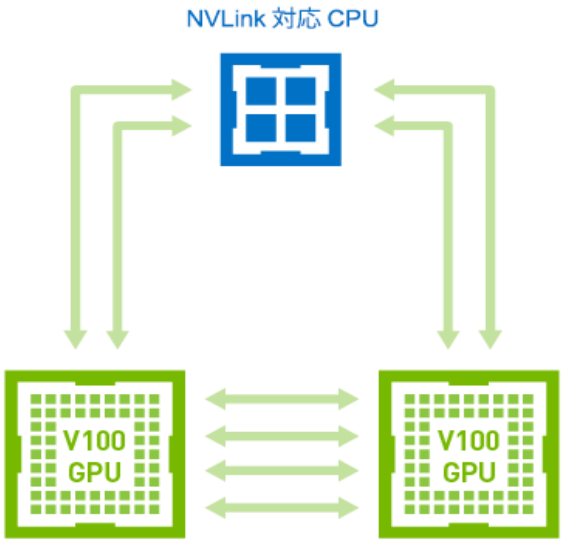


図 13. V100 の GPU 間/GPU-CPU 間 NVLink 接続

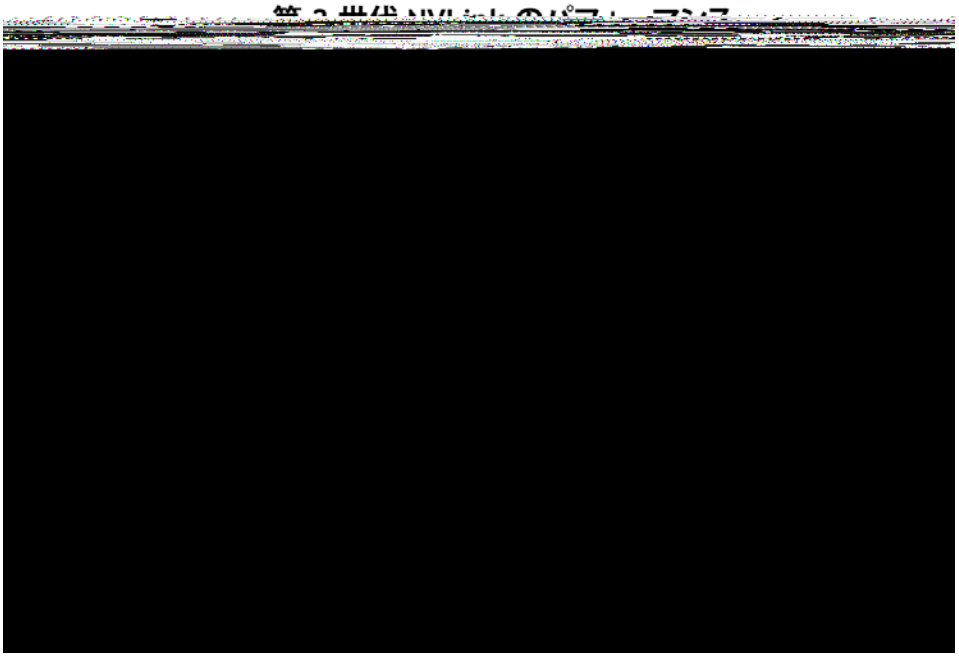


図 14. 第 2 世代 NVLink のパフォーマンス

HBM2 メモリ アーキテクチャ

Tesla P100 は、高帯域幅の HBM2 メモリ テクノロジを世界で初めてサポートした GPU アーキテクチャです。Tesla V100 は、さらに高速かつ高効率の HBM2 を実装しています。HBM2 メモリは、GPU と同じ物理パッケージ内に置かれたメモリ スタックで構成されているため、従来の GDDR5 メモリ設計よりも電力と面積を大幅に削減して、より多くの GPU をサーバーにインストールできます。

Tesla V100 の HBM2 は、HBM2 スタックごとに 4 つのメモリ ダイを使用し、4 スタックで最大 32 GB の GPU メモリを搭載します。HBM2 メモリのピーク メモリ帯域幅は、4 スタック全体で 900 GB/秒になります。これは、Tesla P100 の最大 732 GB/秒に匹敵します。HBM2 テクノロジの詳細は、Pascal アーキテクチャ ホワイト ペーパー。

Tesla V100 は、Tesla P100 よりもピーク DRAM 帯域幅が広いことに加えて、V100 GPU の HBM2 効率も大幅に改善されています。Samsung の新世代 HBM2 メモリと Volta の新世代メモリ コントローラーの組み合わせは、Pascal GP100 と比較してメモリ帯域幅を 1.5 倍にし、多数のワークロードを実行させて、メモリ帯域幅効率 95% 以上を達成しています (図 15 参照)。

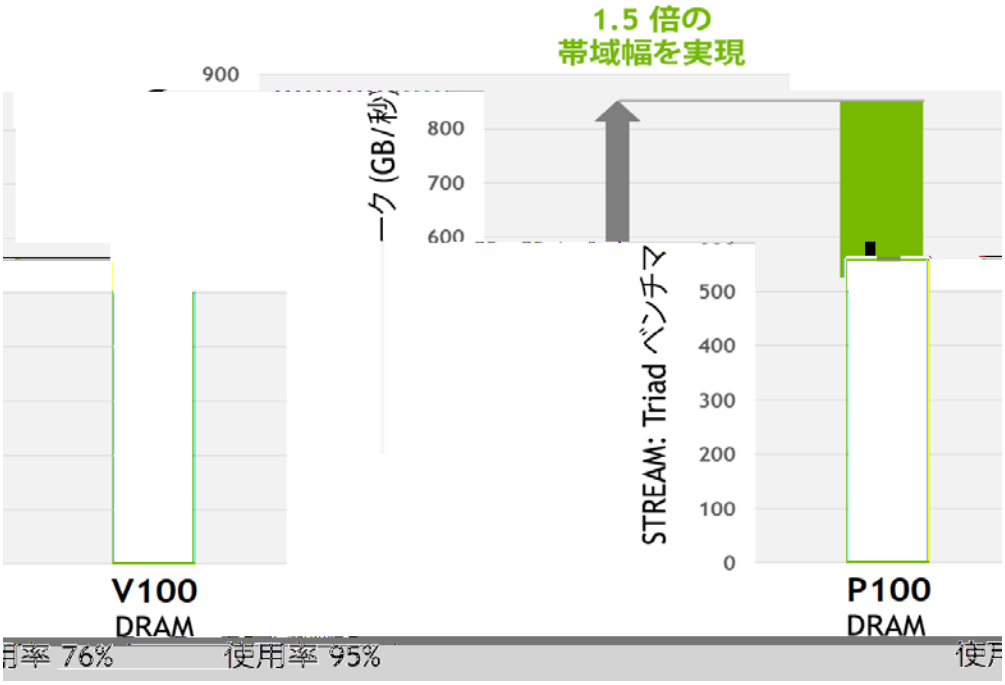


図 15. HBM2 メモリの高速化 - V100 と P100 の比較

ECC メモリ回復性

Tesla V100 HBM2 メモリ サブシステムは、データを保護する Single-Error Correcting Double-Error Detecting (SECDED) のエラー訂正符号 (ECC) をサポートしています。ECC は、データ破損の影響を受けやすいコンピューティング アプリケーションに対して、高い信頼性を提供します。これは、大規模なデータセットの処理やアプリケーションの長時間実行など、大型のクラスター コンピューティング環境に特に効果的です。

HBM2 は、ネイティブまたはサイドバンド ECC をサポートしており、メイン メモリとは別の小さなメモリ領域を ECC ビットに使用します。これは、メイン メモリの一部を ECC ビット用に確保するインライン ECC より有利です。たとえば、Tesla K40 GPU の GDDR5 メモリ サブシステムの場合は、GDDR5 全体の 6.25% が ECC ビット用に予約されます。V100 や P100 を使用すれば、帯域幅や容量を使用することなく ECC を有効にできます。メモリ書き込みの場合は、1 回の書き込みの 32 バイトのデータ全体に対して ECC ビットが計算されます。8 バイトのデータごとに 8 つの ECC ビットが作成されます。メモリ読み取りの場合は、32 バイトの読み取りデータと並行して 32 の ECC ビットが読み取られます。ECC ビットは、シングル ビット エラーの訂正またはダブル ビット エラーのフラグに使用されます。

SM レジスタ ファイル、L1 キャッシュ、L2 キャッシュなど、GV100 の他の重要な構造も SECDED ECC によって保護されます。同じ構造の Pascal GP100 でも同様に、SECDED ECC によって高レベルのエラー検出と訂正、および全体的なメモリ回復性が確保されていました。

コピー エンジン拡張機能

NVIDIA GPU コピー エンジンは、GPU 間または GPU-CPU 間でデータを転送します。従来の GPU では、コピー元またはコピー先メモリ アドレスのどちらかが GPU ページ テーブルにマップされていない場合は、コピー エンジン転送 (DMA 転送と同様) を実行したときに致命的な障害が発生する可能性がありました。また、従来のコピー エンジンでは、コピー元またはコピー先メモリ領域の両方を固定 (ページング不可) する必要がありました。

新しい Volta GV100 GPU コピー エンジンでは、ページ テーブルにマップされていないアドレスに対してページ フォールトを生成できます。これで、メモリ サブシステムがページ フォールトを処理してアドレスをページ テーブルにマッピングした後、転送を実行できます。これは、特に大規模なマルチ GPU またはマルチ CPU システムで効果のある機能強化です。複数のプロセッサ間で複数のコピー エンジンを操作するためにメモリを固定してしまうと、使用できるメモリが大幅に減る可能性があるためです。ハードウェア ページ フォールトを使用することで、アドレスが存在するかどうかを気にすることなくコピー エンジンに渡すことができ、コピー処理が正常に機能します。この機能は現在の ATS システムでも使用されます。

TESLA V100 ボード設計

Tesla V100 の SXM2 ボード フォーム ファクターは Tesla P100 と同じものです。主な違いは、GP100 の代わりに GV100 GPU を使用する点です。SXM2 ボードは NVLink と PCIe 3.0 の接続を提供します。ワークステーション、サーバー、大規模コンピューティング システムで 1 つ以上の V100 アクセラレータを使用できます。V100 アクセラレータは 140 mm x 78 mm で、GPU に必要なさまざまな電圧を供給する高効率電圧レギュレータを内蔵しています。V100 の定格は 300 W TDP (熱設計電力) です。

図 16 は Tesla V100 アクセラレータの表面、**図 17** は裏面です。**図 18** は、NVIDIA Tesla V100 SXM2 モジュールの立体様式図です。

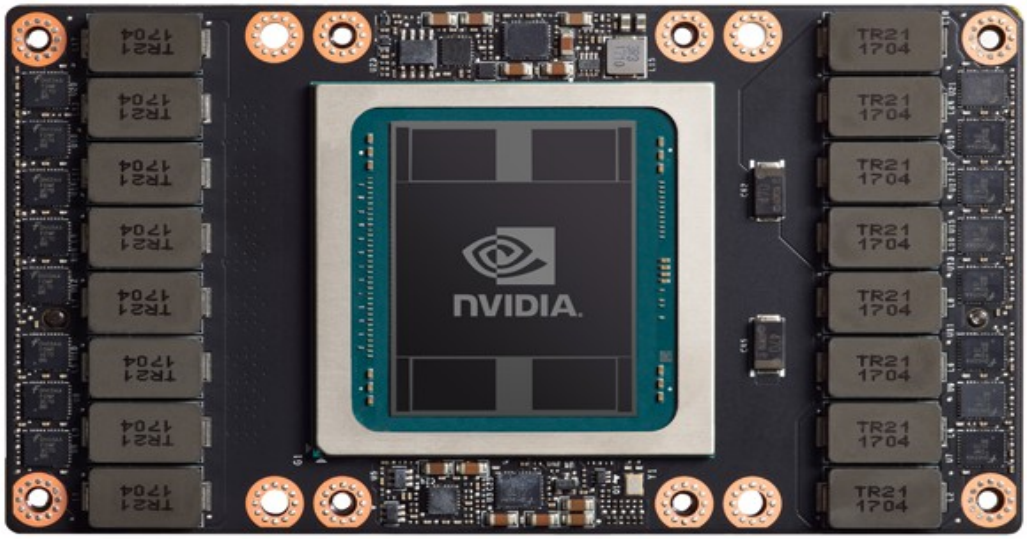


図 16. Tesla V100 アクセラレータ (表面)

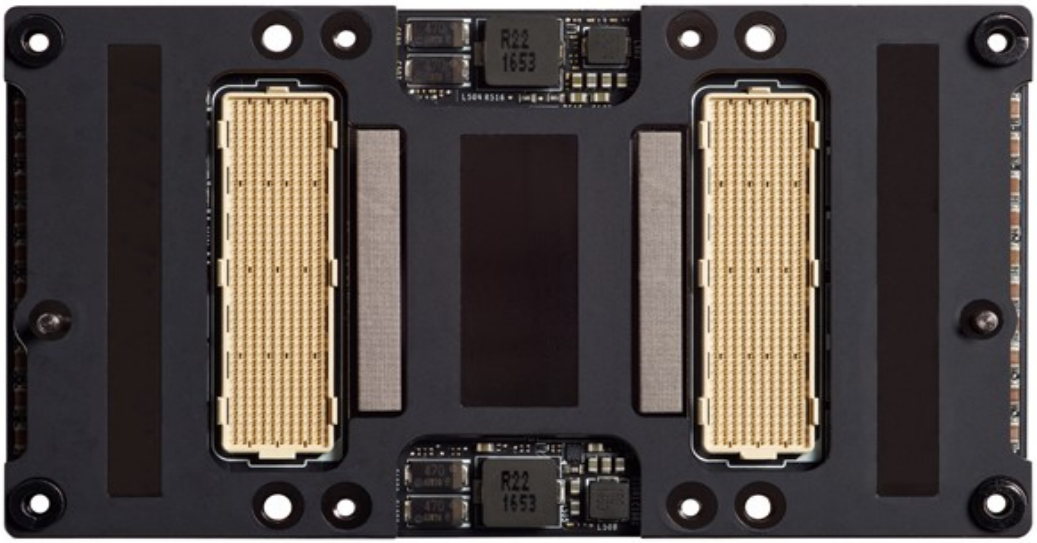


図 17. Tesla V100 アクセラレータ (裏面)

図 18.　NVIDIA Tesla V100 SXM2 モジュール - 立体様式図

GV100 CUDA: ハードウェアとソフトウェア アーキテクチャの進化

NVIDIA® CUDA® は、NVIDIA GPU の大規模な並列処理機能にアクセスするための、アプリケーション開発者向け並列コンピューティング プラットフォームおよびプログラミング モデルです。CUDA は、ディープラーニングから、天文学、分子動力学シミュレーション、金融工学まで、大規模な演算とメモリを必要とする幅広いアプリケーションの GPU アクセラレーションの基盤です。数千の GPU アクセラレーション アプリケーションが CUDA 並列コンピューティング プラットフォームで開発されています。

NVIDIA CUDA ツールキットは、C および C++ プログラミング言語の拡張機能により、大規模な並列アプリケーションを開発する総合的な環境を提供します。柔軟性とプログラミング性に優れた CUDA は、新しいディープラーニングおよび並列コンピューティング アルゴリズムの研究に最適なプラットフォームです。図 19 は、CUDA プラットフォーム上に構築されたディープラーニング イノベーションの歴史です。

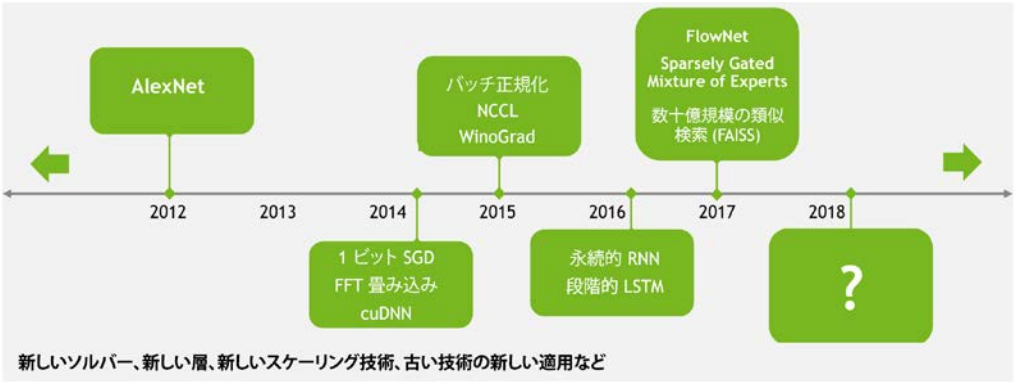


図 19. CUDA を使用して開発されたディープラーニング手法

このセクションで紹介している進化した Volta アーキテクチャにより、CUDA アプリケーション内の並列スレッドの可能性がさらに広がり、CUDA プラットフォームの機能、柔軟性、生産性、移植性が大きく向上します。

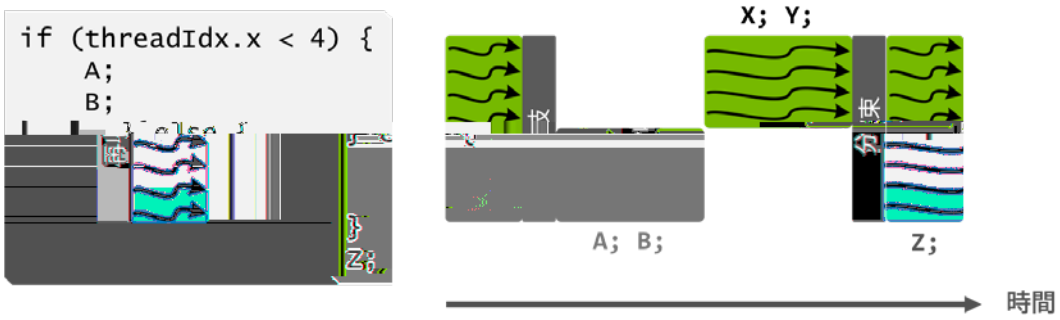
独立型スレッド スケジューリング

Volta アーキテクチャは、以前の GPU よりも簡単にプログラミングができるように設計されているため、ユーザーは、より複雑で多様なアプリケーション開発に生産的に取り組むことができます。Volta GV100 は、独立型スレッド スケジューリングをサポートした初の GPU で、プログラム内の並列スレッド間でより細やかな同期と協調を可能にします。Volta は、GPU 上のプログラム実行に必要な作業を削減し、スレッド協調の柔軟性を高めて細粒度の並列アルゴリズムの効率を向上させることを目的として設計されています。

以前の NVIDIA GPU SIMT モデル

Pascal 以前の NVIDIA GPU は、(Warp と呼ばれる) 32 スレッドのグループを SIMT (Single Instruction, Multiple Thread) 方式で実行します。Pascal Warp は、32 スレッドのすべてに共通の単一のプログラム カウンターと、ある時点で Warp のどのスレッドがアクティブかを指定するアクティブ マスクを組み合わせて使用します。これは、図 20 に示すように、実行パスの分岐によっていくつかのスレッドが非アクティブ

のままになり、Warp のそれぞれの部分の実行がシリアル化されることを意味しています。元のマスクは、Warp が再収束する (通常は分岐セクションの終わり) まで格納され、この時点でマスクが復元されて、スレッドが再度同時に実行されます。



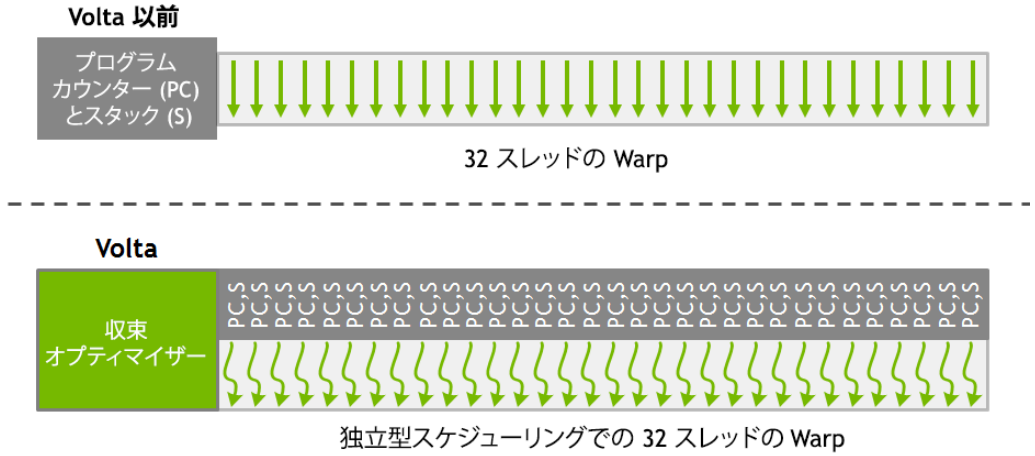
Pascal 以前の NVIDIA GPU の SIMT Warp 実行モデルにおけるスレッド スケジューリング。大文字は、プログラム疑似コード内のステートメントを表しています。Warp 内の分岐がシリアル化され、分岐の片側のステートメントがすべて同時に実行された後、もう片側のステートメントが実行されます。通常は、else ステートメントの後に Warp のスレッドが再収束されます。

図 20. Pascal 以前の GPU による SIMT Warp 実行モデル

Pascal SIMT 実行モデルは、スレッドの状態を追跡するリソースを減らすと共に、積極的にスレッドを再収束させて並列性を高めて効率化します。しかし、Warp 全体のスレッドの状態を集約して追跡すると、実行パスが分岐する際に、異なるブランチのスレッドが再収束するまで並列性を失います。これは、同じ Warp のスレッドが分岐した領域にある場合、または異なる実行状態にある場合には、相互に信号を送ったりデータを交換したりできないことを意味しています。異なる Warp のスレッドは引き続き同時に実行されますが、同じ Warp から分岐したスレッドは再収束するまでシリアルに実行されているため、整合性が取れません。たとえば、ロックやミューテックスによって保護される細粒度のデータを共有するアルゴリズムと、競合するスレッドの Warp とがデッドロックに陥ってしまう可能性があります。したがって、Pascal 以前の GPU では、細粒度の同期を回避するか、ロックを行わないアルゴリズムまたは Warp 対応アルゴリズムを使用するほかにありません。

Volta SIMT モデル

Volta ではこの図式を転換し、Warp に関係なくすべてのスレッドで平等な同時性を実現しました。図 21 のように、プログラム カウンターやコール スタックなどの実行状態をスレッドごとに管理します。



Volta (下) 独立型スレッド スケジューリング アーキテクチャのブロック図と、Pascal とそれ以前のアーキテクチャ (上) の比較。Volta では、プログラム カウンター (PC) やコール スタック (S) などのスケジューリング リソースをスレッドごとに管理しますが、以前のアーキテクチャでは、これらのリソースを Warp ごとに管理します。

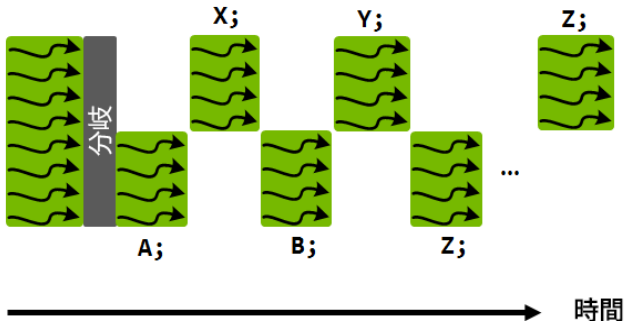
図 21. スレッドごとにプログラム カウンターとコール スタックを持つ Volta Warp

Volta の独立型スレッド スケジューリングを使用すると、実行リソースを調整できるほか、別のスレッドでデータが生成されるまでスレッドを待機させるなど、GPU が任意のスレッドを実行できるようになります。並列効果を最大限に活用するため、組み込みのスケジュール オプティマイザーによって、同じ Warp のアクティブ スレッドを SIMT ユニットにまとめる方法を決定します。これにより、従来と同様の高い SIMT 実行スループットを維持しながら、柔軟性を格段に向上させることができます。スレッドがサブ Warp の粒度で分岐および再収束できるだけでなく、同じコードのスレッドをまとめて並列に実行することで、最大限に効率化します。

図 20 のコード例は、Volta では若干異なる方法で実行されます。図 22 に示すように、プログラム内の if と else で分岐したステートメントを適時にインターリーブできるようになります。実行するのは変わらず SIMT です。CUDA コアはどのクロック サイクル

においても、これまでと同様に Warp 内のすべてのアクティブ スレッドに同じ命令を実行し、アーキテクチャの実行効率を維持します。重要なのは、Volta では Warp 内のスレッドを個別にスケジュールできるため、複雑で細粒度のアルゴリズムやデータ構造をより自然に実装できるという点です。スケジューラは、スレッドの独立した実行をサポートすると共に、非同期コードを最適化して可能な限り収束を維持することで、最大の SIMT 効率を実現します。

```
if (threadIdx.x < 4) {
    A;
    B;
} else {
    X;
    Y;
}
Z;
```



Volta の独立型スレッド スケジューリングにより、分岐ブランチのステートメントの実行をインターリーブできます。これにより、Warp 内のスレッドどうしが同期と通信を行う細粒度の並列アルゴリズムが可能になります。

図 22. Volta の独立型スレッド スケジューリング

興味深いことに、図 22 では、Warp 内のすべてのスレッドがステートメント Z を同時に実行するようには示されていません。これは、他の分岐ブランチの実行に必要なデータが Z によって生成される可能性をスケジューラは想定する必要があるためです。その場合、自動で再収束を行うのは安全ではありません。A、B、X、Y は同期演算で構成されないことが普通ですが、その場合、スケジューラは、以前のアーキテクチャと同様に、Warp が自然に Z に再収束しても安全であると識別できます。

図 23 に示すように、プログラムは、新しい CUDA 9 Warp 同期関数 __syncwarp () を呼び出して、強制的に再収束を実行できます。この場合、Warp の分岐部分は Z を一緒に実行しないかもしれませんが、いずれかのスレッドが __syncwarp () の次のステートメントに到達する前に、Warp 内のスレッドのすべての実行パスが完了します。同様に、Z を実行する前に __syncwarp () の呼び出しを置くと、Z を実行する前に強制的に再収束が行われます。アプリケーションにとって安全であることがわかっている場合は、これで SIMT の効率が向上する可能性があります。

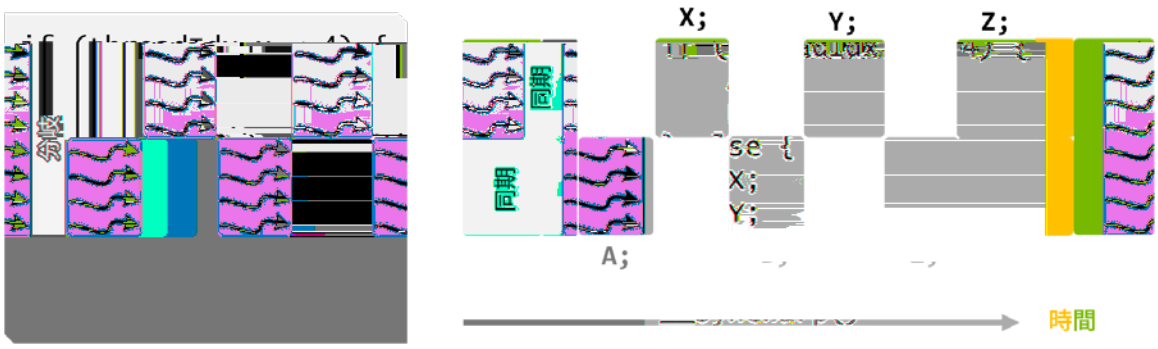


図 23. プログラムが明示的な同期を使用して Warp 内のスレッドを再収束させる

スタベーション フリーのアルゴリズム

スタベーション フリーのアルゴリズムは、独立型スレッド スケジューリングで実現する主要パターンです。これは、すべてのスレッドが競合リソースに適切にアクセス可能であることが保証される限り、正しく実行される並列コンピューティング アルゴリズムです。たとえば、スレッドのミューテックス取得が最終的に成功すると保証されている場合は、スタベーション フリーのアルゴリズムでミューテックス (またはロック) を使用できます。スタベーション フリーをサポートしないシステムの場合は、複数のスレッドがミューテックスの取得と解放を繰り返し、他のスレッドがミューテックスを正しく取得できないことがあります。

マルチスレッド アプリケーションで双方向連結リストにノードを挿入する、Volta 独立型スレッド スケジューリングの簡単な例を挙げます。

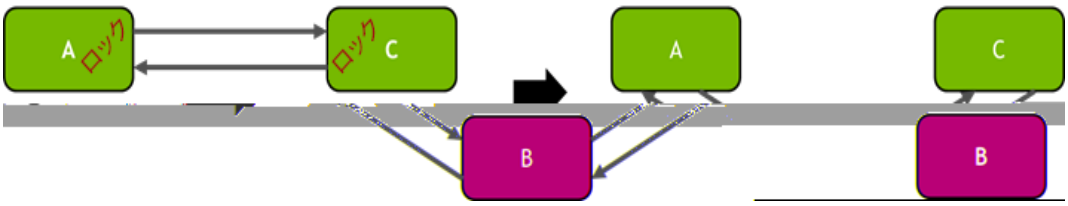
```
__device__ void insert_after(Node *a, Node *b)
{
    Node *c;
    lock(a); lock(a->next);
    c = a->next;

    a->next = b;
    b->prev = a;

    b->next = c;
    c->prev = b;

    unlock(c); unlock(a);
}
```

この例では、双方向連結リストの各要素には、少なくとも「次のポインター」、「前のポインター」、「ロック」の 3 つのコンポーネントがあり、所有者がノードを更新する際に排他的なアクセス権を提供します。図 24 は、ノード A の後にノード B を挿入して、ノード A と C の次のポインターと前のポインターを更新するところを示しています。



ノード単位のロックを取得してから (左)、リストにノード B を挿入する (右)。

図 24. 細粒度ロックによる双方向連結リスト

Volta の独立型スレッド スケジューリングでは、スレッド T0 がノード A をロックしている場合でも、同じ Warp のスレッド T1 が、スレッド T0 の進行を妨げることなく、ロックが使用可能になるまで確実に待機します。ただし、Warp 内のアクティブ スレッドは同時に実行するため、ロックを繰り返し試行するスレッドがあると、そのロックを待つスレッドのパフォーマンスが低下する可能性があります。

GPU のパフォーマンスでは、上の例のようにノード単位のロックがきわめて重要です。従来の双方向連結リストの実装では、ノードを個別に保護するのではなく、構造全体に排他的なアクセスを提供する粒度の粗いロックを使用する場合があります。この手法では、多数のスレッドを持つアプリケーション (Volta の場合は最大 163,840 の並列スレッド) のロック競合が急増してパフォーマンスが低下します。各ノードで粒度の細かいロックを使用することで、非標準的なノード挿入パターンを除いて、大規模リストで発生する一般的なノード間の競合は減少します。

このような細粒度ロックを持つ双方向連結リストの例は、シンプルでありながら、独立型スレッド スケジューリングによって頻繁に使用するアルゴリズムやデータ構造を GPU に自然に実装できることを証明しています。

VOLTA マルチプロセス サービス

Volta マルチプロセス サービス (MPS) は、Volta GV100 アーキテクチャの新機能です。これは、GPU を共有する複数のコンピューティング アプリケーションのパフォーマンスと分離性を強化します。GPU を共有する複数のアプリケーションの実行は、一般にタイムスライスで実装されています。つまり、1 つのアプリケーションが一定の時間排他的アクセス権を取得し、その後に別のアプリケーションがアクセスできるようになります。Volta MPS は、アプリケーションが単体で GPU 実行リソースを利用しきれない場合に、複数のアプリケーションが同時に GPU 実行リソースを共有できるようにして、全体的な GPU 使用率を改善します。

NVIDIA は、Kepler GK110 GPU にソフトウェア ベースのマルチプロセス サービス (MPS) と MPS サーバーを導入しました。これは、複数の CPU プロセス (アプリケーション コンテキスト) を 1 つのアプリケーション コンテキストに結合して GPU 上で実行することで、GPU リソースの使用率を向上させるサービスです。

Volta MPS では、MPS サーバーの重要なコンポーネントにハードウェア アクセラレーションを導入してパフォーマンスと分離性を向上し、MPS クライアントの最大数を Pascal の 16 から 48 に増やしました (**図 25** を参照)。Volta マルチプロセス サービスは、単一ユーザーの複数のアプリケーション間で GPU を共有することを目的としており、マルチユーザーまたはマルチテナントのユース ケースには対応していません。

Pascal の CUDA マルチプロセス サービスは、他の GPU アプリケーションと同時に実行リソースを共有するように要求した GPU アプリケーションの代替となる CPU プロセスです。これが仲介役となり、GPU 内の並行カーネル実行作業キューに作業を送信します。

Volta マルチプロセス サービスのハードウェア アクセラレーションにより、CUDA MPS クライアントが GPU 内の作業キューに作業を直接送信できるようになるため、送信の遅延が大幅に減少し、全体的なスループットが向上します。Volta では、残った CPU MPS 制御プロセスを構成したり MPS へオプトインしたりすることが可能です。

Volta MPS は、サービス品質 (QoS) と独立型アドレス空間という 2 つの重要なメトリックスで MPS クライアント間の分離性を強化します。**図 25** に示すように、Volta では QoS に加えて複数の MPS クライアント A、B、C のアドレスが分離

されます。従来の NVIDIA GPU の CUDA MPS と同様に、クライアント間の致命的な障害の分離はできません。

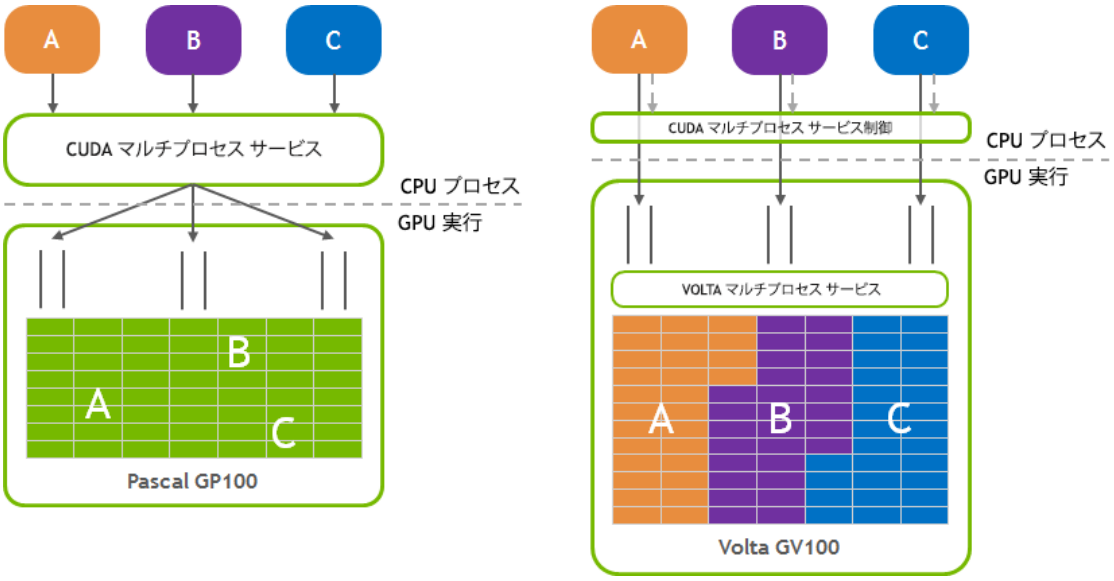


図 25. Pascal のソフトウェア ベース MPS サービスと Volta のハードウェア アクセラレーション MPS サービスの比較

サービス品質とは、作業の送信時に、クライアントでの作業の処理に必要な GPU 実行リソースをどれだけすばやく確保できるかを表しています。Volta MPS は、実行に必要な GPU 部分を指定して MPS クライアントを制御します。これにより、各クライアントの GPU 実行リソースをごく一部に制限し、ヘッドオブライン ブロッキングを削減または解消します。ヘッドオブライン ブロッキングとは、1 つの MPS クライアントの作業が GPU 実行リソースを専有し、作業が完了するまで他のクライアントが進行できなくなることです。QoS を強化することでシステム内の平均遅延/ジッターが減少します。これは、MPI/HPC ユース ケースとディープラーニング推論ユース ケースのどちらにもきわめて重要です。

特に、パフォーマンスを最大化するために複数の画像をまとめて同時に GPU に送信するバッチ処理システムでは、Volta がディープラーニング推論にきわめて高いスループットと低遅延を提供します。バッチ処理システムがない場合、個々の推論ジョブが GPU の実行リソースをフルに活用することはありません。Volta MPS は、多数の個別推論ジョブを同時に GPU に送信して全体の GPU 使用率を高めることで、手軽にスループットを向上させると同時に遅延の要件を満たします。

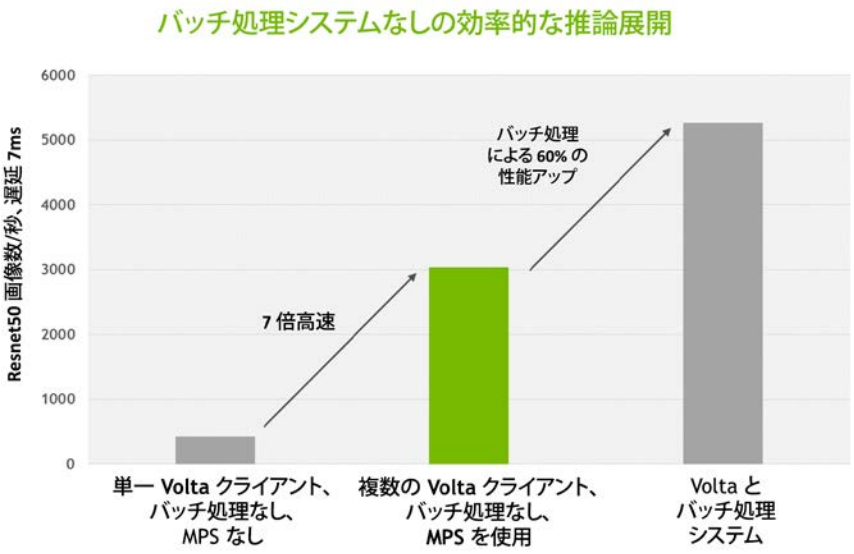


図 26. Volta MPS による推論

Linux 対応統合メモリ機能のロードマップ (GPU からの malloc メモリ アクセスなど) は、Volta MPS の主要機能の 1 つです。従来の NVIDIA GPU アーキテクチャの CUDA MPS クライアントは、GPU 上では単一のアドレス空間で動作しますが、独立した CPU プロセス メモリにアクセスする際の互換性がありません。

統合メモリとアドレス変換サービス

Kepler および Maxwell GPU の CUDA 6 に統合メモリの一部機能を導入し、Pascal GP100 GPU にハードウェア ページ フォールトとさらに大きなアドレス空間を追加しました。統合メモリは、単一の統合仮想アドレス空間を CPU と GPU のメモリとして使用することで、GPU プログラミングや GPU へのアプリケーション移植を大幅に簡略化できます。プログラマが GPU と CPU の仮想メモリ システム間で共有するデータの管理に悩む必要がなくなります。Pascal GP100 の統合メモリでは、GPU と CPU の仮想アドレス空間全体での透過的なデータ移行が可能になります (Pascal 統合メモリ テクノロジの詳細については、Pascal アーキテクチャ ホワイト ペーパー (英語) をご覧ください)

Pascal GP100 の統合メモリにより、さまざまな部分の CUDA プログラミングが強化されましたが、Volta GV100 と組み合わせることでさらに統合メモリの効率とパフォーマンスが向上します。新しいアクセス カウンター機能により、他のプロセッサ上の

メモリに GPU がアクセスする頻度を追跡できます。これを使用すると、ページに最も頻繁にアクセスするプロセッサを割り出して、物理メモリにメモリ ページを移動できます。アクセス カウンター機能は、NVLink 接続または PCIe 接続された GPU-CPU または GPU-GPU アーキテクチャ内で動作し、POWER9 や x86 を含むさまざまなタイプの CPU を使用できます。

Volta は、NVLink を介したアドレス変換サービス (ATS) もサポートしています。GPU は、ATS を使用して CPU のページ テーブルに直接アクセスします。GPU MMU でミスが発生すると、CPU に対してアドレス変換リクエスト (ATR) が行われます。CPU は、ページ テーブルでリクエストされた仮想アドレスと物理アドレスのマッピングを検索して GPU に変換結果を返します。ATS は、malloc などで直接割り当てられた CPU メモリに GPU が完全にアクセスできるようにします。

COOPERATIVE GROUPS

並列アルゴリズムでは、集合的な計算を行うために、スレッドの協調が必要になることがあります。協調型コードを作成するには、協調するスレッドをグループ化して同期する必要があります。CUDA 9 は、スレッド グループを管理する新しいプログラミング モデルとして Cooperative Groups を導入しています。

CUDA プログラミング モデルは、以前より、協調するスレッドを同期するために、1 つのスレッド ブロックのすべてのスレッドを 1 つのバリアで覆うというシンプルな構成概念を __syncthreads() 関数で実装してきました。しかし、高いパフォーマンス、設計の柔軟性、グループ間の集合的な関数インターフェイス形式でのソフトウェア再利用性などを考慮して、スレッド ブロックより細粒度のスレッド グループを定義し、その中で同期したいと考えるプログラマも少なくありません。

Cooperative Groups は、サブブロックおよびマルチブロックの粒度で明示的に定義されたスレッドのグループで、同期などの集合的な処理を実行します。ソフトウェアの境界を越えたクリーンな構成をサポートしており、収束を仮定する必要なく、ライブラリやユーティリティ関数をローカルなコンテキスト内で安全に同期できます。また、プログラマの意図が示された安全でサポート可能な方法で柔軟に同期することで、ハードウェア ファスト パス (GPU Warp サイズなど) を最適化できます。Cooperative Groups プリミティブは、プロデューサーとコンシューマー操作の並列性、

日和見並列性、グリッド全体のグローバル同期など、新しい協調型並列性パターンを CUDA で可能にします。

Cooperative Groups は、将来の GPU 機能へのスケーリングなどを、さまざまな GPU アーキテクチャで安全に動作する柔軟で拡張可能なコードを記述できる抽象的概念を提供しています。スレッド グループのサイズは、少数のスレッド (Warp より小さい) から、スレッド ブロック全体、1 グリッド内のすべてのスレッド ブロック、さらには複数の GPU にまたがる複数のグリッドまで対応しています。

Cooperative Groups はすべての GPU アーキテクチャで動作しますが、一部の機能は GPU 機能が進化すると必然的にアーキテクチャに依存することになります。スレッド ブロックや Warp より小さな粒度のグループの同期など、基本的な機能はすべてのアーキテクチャでサポートしています。一方、グリッド全体やマルチ GPU などの新しい同期グループは、Pascal および Volta GPU でサポートしています。さらに、Volta の独立型スレッド スケジューリングにより、任意のクロス Warp 粒度およびサブ Warp 粒度での柔軟なスレッド グループの選択とパーティショニングが可能です。Volta 同期はすべてスレッド単位のため、Warp 内のスレッドを複数の分岐コード パスから同期できます。

Cooperative Groups プログラミング モデルは、以下の要素で構成されています。

- ディープラーニング行列演算専用の新しい混合精度 FP16/FP32 Tensor コア
- 協調スレッドのグループを表すデータ型
- CUDA 起動 API で定義された既定のグループ (スレッド ブロックおよびグリッドなど)
- 既存のグループを新しいグループにパーティショニングする演算
- グループ内のすべてのスレッドを同期するバリア演算
- グループ プロパティおよびグループ固有の集合的通信を検査する演算

以下の簡単な例で、Cooperative Groups 演算の基本を説明します。

```
__global__ void cooperative_kernel(...)
{

  // obtain default "current thread block" group
  thread_group my_block = this_thread_block();

  // subdivide into 32-thread, tiled subgroups
```

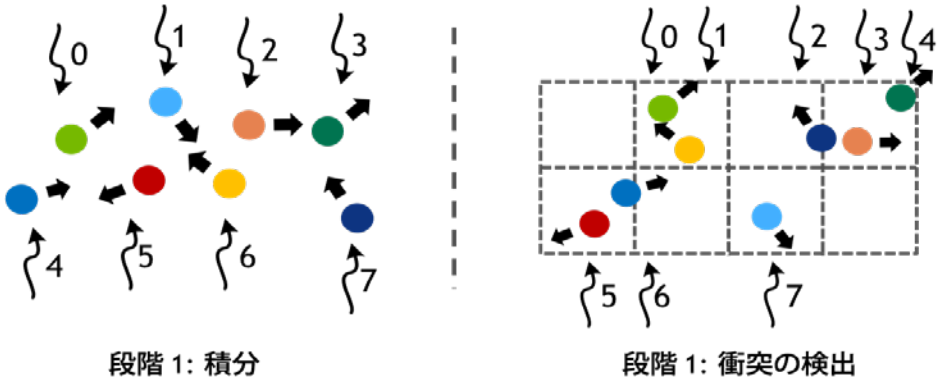
```
    // Tiled subgroups evenly partition a parent group into
    // adjacent sets of threads - in this case each one warp in size
    thread_group my_tile = tiled_partition(my_block, 32);

    // This operation will be performed by only the
    // first 32-thread tile of each block
    if (my_block.thread_rank() < 32) {
        ...
        my_tile.sync();
    }
}
```

Cooperative Groups は、C++ のテンプレートを使用して、グループを表すための型と API オーバーロードを提供します。このグループのサイズは、効率化のために静的に決定されます。言語レベルのインターフェイスは、CUDA C++ 実装の基盤となる PTX アセンブリ拡張機能セットでサポートされており、同様の機能を持つ任意のプログラミング システムで使用できます。また、cuda-memcheck の競合検出ツールや CUDA デバッガーは、Cooperative Groups の柔軟な同期パターンと互換性があるため、RAW (Read After Write) 問題などの細かな並列同期バグを比較的簡単に検出できます。

Cooperative Groups を使用すると、これまで不可能だった同期パターンを表現できるようになります。同期の粒度が自然なアーキテクチャ粒度 (Warp やスレッドブロック) に対応している場合、この柔軟性のオーバーヘッドは無視できるレベルです。Cooperative Groups で記述された集合通信プリミティブのライブラリで高いパフォーマンスを得るには、より簡素なコードが必要です。

シミュレーションのステップごとに 2 段階の計算を行う粒子シミュレーションを考えてみます。最初に、各粒子の位置と速度を時間方向に積分します。次に、正規グリッド空間データ構造を作成して、粒子間の衝突をすばやく検出します。図 27 は、この 2 つの段階を示しています。



2 段階の粒子シミュレーション。番号付きの矢印は並列スレッドと粒子のマッピングを表します。積分と正規グリッド データ構造の構築の後で、メモリ内の粒子の順序とスレッドへのマッピングが変化するため、段階の間で同期する必要があります。

図 27. 段階の粒子シミュレーション

Cooperative Groups 以前のシミュレーション実装では、段階 1 から段階 2 でスレッドのマッピングが変化するため、複数のカーネルを起動する必要がありました。また、正規グリッド アクセラレーション構造を構築するプロセスでメモリ内の粒子の順序が再設定されるため、スレッドから粒子への新しいマッピングが必要になります。このような再マッピングには、スレッド間の同期が必要です。以下の CUDA 疑似コードが示すように、この要件は、連続して起動されるカーネル間で暗黙に同期が行われることで満たされます。

```
// threads update particles in parallel
integrate<<<blocks, threads, 0, s>>>(particles);
// Note: implicit sync between kernel launches
// Collide each particle with others in neighborhood
collide<<<blocks, threads, 0, s>>>(particles);
```

Cooperative Groups は、柔軟でスケーラブルなスレッド グループ タイプを提供し、上の例のような状況では、同期プリミティブが 1 回のカーネル起動で並列性を再マッピングします。以下の CUDA カーネルは、粒子系が 1 つのカーネルでどのように更新されるかを示しています。this_grid() を使用して、このカーネル起動のすべてのスレッドを含むスレッド グループを定義し、次にそれを 2 つの段階の間で同期します。

```
__global__ void particleSim(Particle *p, int N) {

 grid_group g = this_grid();
 // phase 1
```

```
  for (i = g.thread_rank(); i < N; i += g.size())
    integrate(p[i]);
  g.sync() // Sync whole grid
  // phase 2
  for (i = g.thread_rank(); i < N; i += g.size())
    collide(p[i], p, N);
}
```

このカーネルの記述を見ると、このシミュレーションの複数 GPU への拡張がきわめて容易なことがわかります。Cooperative Groups 関数 this_multi_grid() は、複数の GPU にまたがるカーネル起動のすべてのスレッドを含むスレッド グループを返します。このグループに対して sync() を呼び出すと、複数の GPU でこのカーネルを実行しているすべてのスレッドを同期します。どちらの場合も、thread_rank() メソッドがスレッド グループ内のスレッドの線形インデックスを提供します。カーネルは、スレッド数より粒子の数が多い場合に、このインデックスを使用して粒子を並列に繰り返し処理します。

```
__global__ void particleSim(Particle *p, int N) {

  multi_grid_group g = this_multi_grid();
  // phase 1
  for (i = g.thread_rank(); i < N; i += g.size())
    integrate(p[i]);
  g.sync() // Sync whole grid
  // phase 2
  for (i = g.thread_rank(); i < N; i += g.size())
    collide(p[i], p, N);
}
```

複数のスレッド ブロックまたは複数の GPU にまたがるグループを使用するには、アプリケーションで cudaLaunchCooperativeKernel() または cudaLaunchCooperativeKernelMultiDevice() API を個々に使用する必要があります。同期するには、すべてのスレッド ブロックが同時に存在している必要があるため、アプリケーションは、起動されたスレッド ブロックのリソース使用量 (レジスタと共有メモリ) が GPU の総リソース量を超えないようにする必要があります。

まとめ

新しい Volta GV100 GPU ベースの NVIDIA Tesla V100 アクセラレータは、世界で最も進化したデータセンター GPU です。AI、HPC、グラフィックスを高速化する V100 により、データ サイエンティスト、研究者、技術者は、かつて不可能だと考えられていた課題に取り組めるようになりました。

Volta はこれまでにない強力な GPU アーキテクチャであり、GV100 はディープラーニングのパフォーマンスにおいて 100 TFLOPS の壁を突破した初のプロセッサです。CUDA コアと Tensor コアを組み合わせた GV100 は、1 基の GPU で AI スーパーコンピューターのパフォーマンスを発揮します。第 2 世代の NVIDIA NVLink は、複数の V100 GPU を最大 300 GB/秒で接続し、世界で最も強力なコンピューティング サーバーを構築します。Tesla V100 アクセラレーション システムを使用すれば、数週間分のコンピューティング リソースを消費する AI モデルを、数日でトレーニングできるようになります。このトレーニング時間の劇的な短縮により、新次元の問題も、NVIDIA Tesla V100 アクセラレータを活用した AI で解決できます。

付録 A. TESLA V100 搭載 NVIDIA DGX-1

データ サイエンティストや人工知能の研究者が求めるのは、正確性、シンプルさ、スピードを兼ね備えたディープラーニング システムです。トレーニングと反復が高速なほど、イノベーションや市場への投入時期も早くなります。図 28 に示す NVIDIA DGX-1 は、ハードウェアとソフトウェアを完全に統合し、すばやく簡単に展開可能な世界初のディープラーニング専用サーバーです。

図 28.　NVIDIA DGX-1 サーバー

NVIDIA は、2016 年に第 1 世代の DGX-1 を発表しました。これは、ハイブリッド キューブ メッシュ ネットワーク内で NVIDIA の高性能 NVLink で相互接続された 8 基の NVIDIA Tesla P100 GPU を搭載しており、さらにデュアル ソケット Intel Xeon CPU および 4 個の 100 Gb InfiniBand ネットワーク インターフェイス カードを組み合わせることで、ディープラーニング トレーニングで並外れたパフォーマンスを発揮します。最大 170 FP16 TFLOPS でトレーニング時間を大幅に短縮可能な NVIDIA DGX-1 は、世界初のオールインワン AI スーパーコンピューターです。Tesla P100 ベースの DGX-1 システム アーキテクチャの詳細については、このホワイト ペーパー (英語) をご覧ください。

DGX-1 システムは、パフォーマンスと信頼性の高いコンポーネントがラックマウント可能な 3U シャーシに組み込まれているため、スタンドアロンで使用することも、クラスターに統合することも可能です。

NVIDIA Tesla V100 のリリースに伴い、NVIDIA は DGX-1 プラットフォームを新しい SKU に更新しました。Tesla V100 ベースの DGX-1 プラットフォームは、NVLink で相互接続された 8 基の NVIDIA Tesla V100 GPU を備え、驚異的な 1 peta FLOPS のパフォーマンスをディープラーニング アプリケーションで実現します (図 29 を参照)。

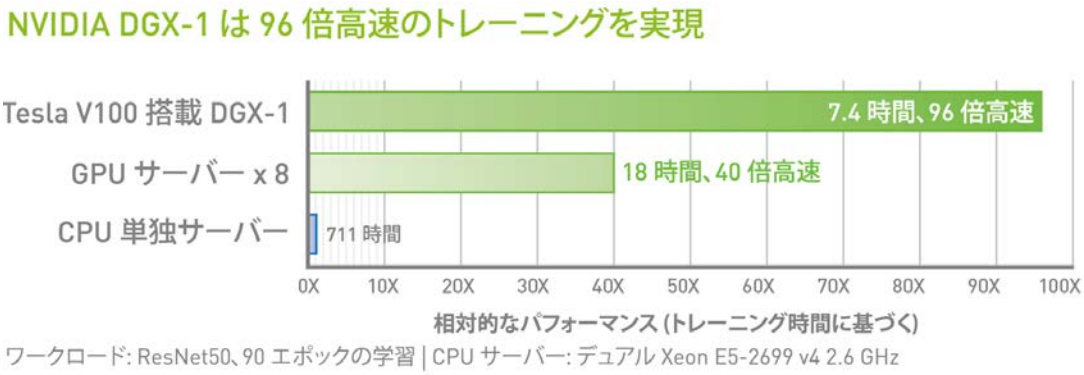


図 29. DGX-1 は GP100 ベースの 8 way サーバーの 3 倍のトレーニング スピードを達成

NVIDIA DGX-1 システム仕様

NVIDIA DGX-1 は、ハードウェアとソフトウェアが完全に統合された、すばやく簡単に展開できる世界初のディープラーニング専用サーバーです。トレーニング時間を大幅に短縮できる革命的なパフォーマンスを誇る、初のオールインワン AI スーパーコンピューターとも言えます。表 3 に、NVIDIA DGX-1 システムの仕様を示します。

表 3. NVIDIA DGX-1 システムの仕様

仕様	DGX-1 (Tesla P100)	DGX-1 (Tesla V100)
GPU 数	Tesla P100 GPU x 8	Tesla V100 GPU x 8
TFLOPS	170 (GPU FP16) + 3 (CPU FP32)	1 (GPU Tensor PFLOP) + 3 (CPU FP32)
GPU Memory	GPU ごとに 16 GB/DGX-1 ノードごとに 128 GB	GPU ごとに 32 GB/DGX-1 ノードごとに 256 GB

CPU	デュアル 20 コア Intel® Xeon® E5-2698 v4 2.2 GHz	デュアル 20 コア Intel® Xeon® E5-2698 v4 2.2 GHz
FP32 CUDA コア	28,672	40,960
Tensor コア数	--	5120
システム メモリ	最大 512 MB 2,133 MHz DDR4 LRDIMM	最大 512 MB 2,133 MHz DDR4 LRDIMM
ストレージ	1.92 TB SSD RAID 0 x 4	1.92 TB SSD RAID 0 x 4
ネットワーク	デュアル 10 GbE, 4 IB EDR	デュアル 10 GbE, 4 IB EDR
システム重量	134 ポンド (約 60 kg)	134 ポンド (約 60 kg)
システム寸法	866 D x 444 W x 131 H (mm)	866 D x 444 W x 131 H (mm)
パッケージ寸法	1180 D x 730 W x 284 H (mm)	1180 D x 730 W x 284 H (mm)
消費電力	3200 W (最大)。1,600 W 負荷分散電源装置 x 4 (3 + 1 冗長)、AC 200 ～ 240 V、10 A	3200 W (最大)。1,600 W 負荷分散電源装置 x 4 (3 + 1 冗長)、AC 200 ～ 240 V、10 A
運用温度範囲	10 - 35°C	10 -35°C

DGX-1 ソフトウェア

強力な DGX-1 ハードウェアには、開発ツールとライブラリの包括的な統合ソフトウェア スタックが含まれており、大規模ディープラーニング向けに最適化されています。これにより、トレーニング実施者はディープラーニング フレームワークとアプリケーションを、設定の手間なく DGX-1 上に展開できます。

プラットフォーム ソフトウェアは、サーバーへの OS とドライバーのインストールが最小限で済むように設計されています。すべてのアプリケーションと SDK ソフトウェアは、NVIDIA が管理する DGX コンテナー レジストリ[2] を通して、NVIDIA Docker と呼ばれるコンテナーでプロビジョニングされます。

DGX-1 用のコンテナーには、最適化された複数のディープラーニング フレームワーク、NVIDIA DIGITS ディープラーニング トレーニング アプリケーション、サードパーティのアクセラレーション ソリューション、NVIDIA CUDA ツールキットなどがあります。

[2] NVIDIA が提供する Docker レジストリ サービス。http://docs.nvidia.com/dgx/dgx-registry-guide/ (英語) をご覧ください

このソフトウェア アーキテクチャのメリットは以下のとおりです。

- 各ディープラーニング フレームワークは個別のコンテナーに格納されるため、libc、cuDNN などのさまざまなバージョンのライブラリを互いに干渉することなく使用できます。
- ディープラーニング フレームワークのパフォーマンス改善やバグ修正がリリースされると、コンテナーの新しいバージョンが DGX コンテナー レジストリで利用可能になります。
- システムの維持が容易で、アプリケーションを OS に直接インストールしないため、OS イメージをクリーンに保つことができます。
- セキュリティ更新、ドライバー更新、OS のパッチがシームレスに提供されます。
- ディープラーニング フレームワークと CUDA ツールキットには、DGX-1 上での高いマルチ GPU パフォーマンスのためにカスタマイズされたライブラリが含まれます。図 30 は DGX-1 ディープラーニング スタックの内容です。

図 30. 生産性を瞬時に向上できる完全統合型の NVIDIA DGX-1 ソフトウェア スタック

ディープラーニング用に調整されたソフトウェアと強力なハードウェアと組み合わせた NVIDIA DGX-1 は、高性能の GPU アクセラレーション ディープラーニング アプリケーションの開発、テスト、ネットワーク トレーニングに即座に使用できる開発者および研究者向けソリューションを提供します。.

付録 B. NVIDIA DGX STATION - ディープラーニング用パーソナル AI スーパーコンピューター

NVIDIA DGX Station™ は、画期的なディープラーニングおよび分析用のスーパーコンピューターです。オフィスのデスクの下に収まる軽量なワークステーションでありながら、CPU 400 個分の驚異的なコンピューティング性能を発揮します (図 31 を参照)。DGX Station は NVIDIA Volta を活用した 4 個の Tesla V100 GPU を搭載する静音の水冷式ワークステーションで、最大 500 Tensor TFLOPS のディープラーニングアプリケーション能力を実現します。

DGX Station は、現時点で最速の GPU ワークステーションと比較して、ディープラーニング トレーニングで約 3 倍、推論でも 3 倍のパフォーマンスを発揮します。DGX Station に搭載されている 4 個の Tesla V100 GPU は、NVIDIA の第 2 世代 NVLink 相互接続テクノロジで接続され、PCIe ベースの GPU ワークステーションの約 5 倍の I/O 帯域幅を実現します。

図 31. Tesla V100 搭載 DGX ステーション

図 32 は、4 way Tesla V100 を搭載する DGX Station のパフォーマンスを示します。Tesla V100 は、CPU ベースのサーバー[3] の 47 倍も高速です。表 4 は DGX ステーションの仕様です。

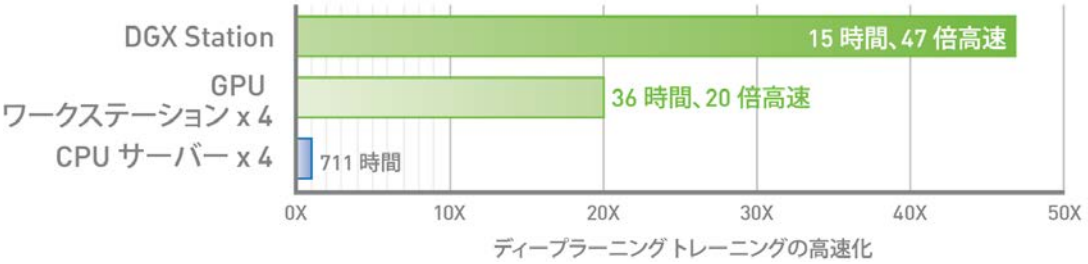


図 32. NVIDIA DGX ステーションでトレーニングのスピードが 47 倍に

表 4. DGX Station の仕様

仕様	DGX Station
GPU 数	NVIDIA Tesla V100 x 4 を NVLink で相互接続
TFLOPS	500 Tensor TFLOPS、15.7 FP32 TFLOPS
Tensor コア数	2,560
CPU	Intel Xeon E5-2698 v4 2.2 GHz (20 コア)

[3] ワークロード: ResNet50、エポック数 90 | CPU サーバー: デュアル Xeon E5-2699 v4 2.6 GHz

システム メモリ	256 GB LRDIMM DDR4
ストレージ	データ: 1.92 TB SSD RAID 0 x 3 OS: 1.92 TB SSD x 1
ネットワーク	デュアル 10 Gb LAN
ディスプレイ	DisplayPort x 3
稼働音	< 35 dB
システム重量	88 ポンド (40 kg)
システム寸法	518 mm (D) x 256 mm (W) x 639 mm (H)
最大電力	1500 Watts
動作温度	10° C - 30° C
オペレーティング システム	Ubuntu Desktop Linux

最新のディープラーニング ソフトウェアをプリロード

NVIDIA DGX Station は、すべての DGX ソリューションに同じソフトウェア スタックがインストールされています。この革新的な統合ソフトウェア スタックでは、一般的なディープラーニング フレームワークを利用でき、NVIDIA のディープラーニング専門家による最適化と毎月の更新を受けられます。さらに、NVIDIA DIGITS ディープラーニング トレーニング アプリケーション、サードパーティのアクセラレーション ソリューション、cuDNN、cuBLAS などの NVIDIA ディープラーニング SDK、CUDA ツールキット、NCCL (高速マルチ GPU 集合通信ライブラリ)、NVIDIA ドライバーなどが同梱されています。

この包括的なディープラーニング ソフトウェア スタックは、すべての DGX プラットフォームで共通した NVIDIA Docker コンテナーおよび NVIDIA コンテナー レジストリ サービスによって継続的に調整、最適化、配信されます。これによりワークフローが簡素化され、データ サイエンティストは、作業を簡単にスケーリングし、データセンターや NVIDIA ディープラーニング クラウド上の DGX-1 サーバーに DGX Station で開発したソリューションを展開できるようになります。

さらに重要な点として、NVIDIA がソフトウェア スタックの管理と提供を行うため、データ サイエンティストは、ソフトウェア コンポーネントの調整や更新に時間を割くことなく、ディープラーニング ソリューションのトレーニングと展開に集中できます。生産性の向上と希少なディープラーニング専門知識の有効利用によって数千ドルのコストを削減できる可能性があり、ハードウェアへの初期投資を抑えることができます。

AI イニシアティブの開始

NVIDIA DGX Station は、合理化されたプラグインと強化されたエクスペリエンスにより、個々の研究者や組織が AI イニシアティブを開始できるように設計されているため、わずか 1 日でニューラル ネットワークのトレーニングを実施できます。

DGX Station は、優れたコンピューティング能力に加えて、以下のような安心できる統合ソリューションを提供します。

- エンタープライズ レベルのサポート
- NVIDIA のディープラーニング専門知識へのアクセス
- ディープラーニング用に最適化されたツールのライブラリとソフトウェア
- タイムリーなソフトウェア アップグレード
- 重大な問題を優先的に解決

DGX Station と NVIDIA のツールおよび専門知識を組み合わせることで、データ サイエンティストの作業を最大限にサポートします。

NVIDIA DGX Station の詳細については、www.nvidia.com/dgx-station をご覧ください。.

付録 C. GPU によるディープラーニングと人工知能の高速化

GPU で開発されたディープ ニューラル ネットワーク (DNN) は、この 5 年間にアルゴリズムの分野に急速に普及しています。自動運転車、迅速な医薬品開発、オンライン映像データベースの自動イメージ キャプション、ビデオ チャット アプリケーションのスマート リアルタイム言語変換など、用途は無限に広がっています。ディープラーニングは、コンピューターが人間とかかわるあらゆる場面で驚くような効果をもたらします。このセクションでは、ディープラーニングの概要と、GPU を次世代のディープラーニングに活用している NDIVIA ユーザーの事例をご紹介します。

ディープラーニングの概要

ディープラーニングは、人間の脳の神経学習プロセスをモデル化した手法です。絶えず学習し知識を増やしていくことで、時間の経過と共により正確で迅速な判断ができるようになります。子供は、最初は大人からさまざまな形を正しく識別して分類することを学び、最終的には自身で識別できるようになります。同様に、ディープラーニングや神経学習システムは、基本的なオブジェクトや遮られたオブジェクトなどをより賢く効率的に識別できるように、オブジェクトにコンテキストを対応させながらオブジェクト認識と分類をトレーニングする必要があります。

簡単に言うと、人間の脳のニューロンはさまざま入力情報を得て、それぞれの入力情報に重要性レベルを割り当て、その出力を他のニューロンに渡して処理します。

図 33 に示されたパーセプトロンは、人間の脳のニューロンに似たニューラル ネットワークの最も基本的なモデルです。この図に示されているように、パーセプトロンはいくつかの入力を持っており、さまざまな対象オブジェクトの特徴を識別するためのトレーニングに使用されます。オブジェクトの形状を定義する際の重要性に基づいて、それぞれの特徴に一定の重みが割り当てられます。

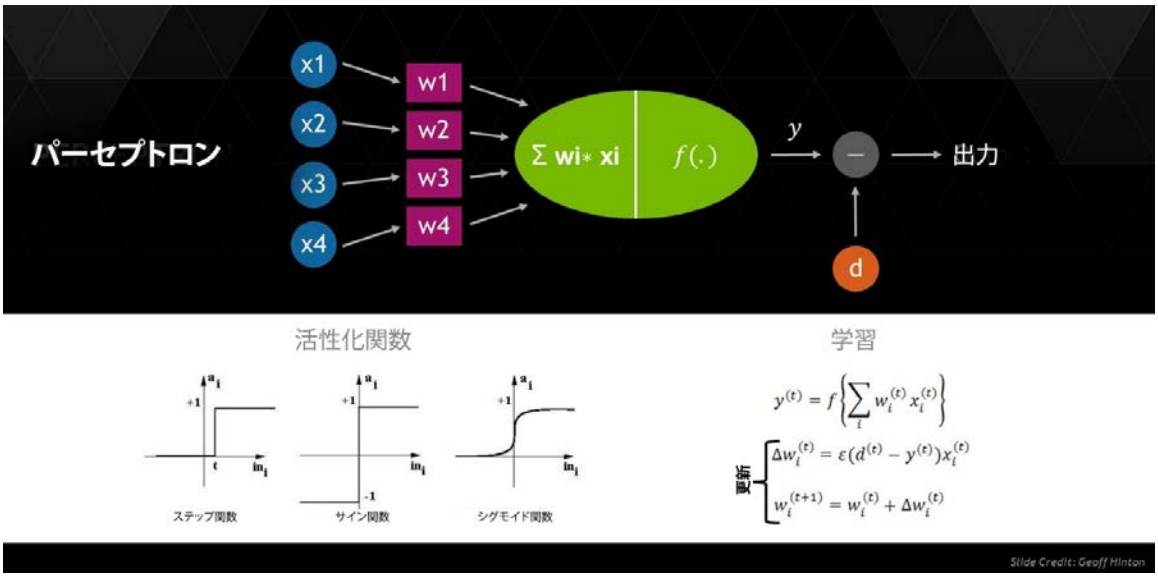


図 33. パーセプトロンは最もシンプルなニューラル ネットワーク モデル

たとえば、手書きの数字のゼロを識別するようにトレーニングされたパーセプトロンがあると仮定します。もちろん、人によってさまざまな書き方があります。パーセプトロンはゼロの画像を受け取り、さまざまなセクションに分解して、それらを特徴 x1 から x4 に割り当てます (ゼロの右上のカーブを x1、下側を x2 など)。正しい判定に必要な重要度に応じて特定の特徴が重み付けられます。図の中央の緑色の楕円では、パーセプトロンが画像のすべての特徴の重み付きの合計を計算しています。次に、その結果に関数が適用され、数字がゼロかどうかを true または false の値で出力します。

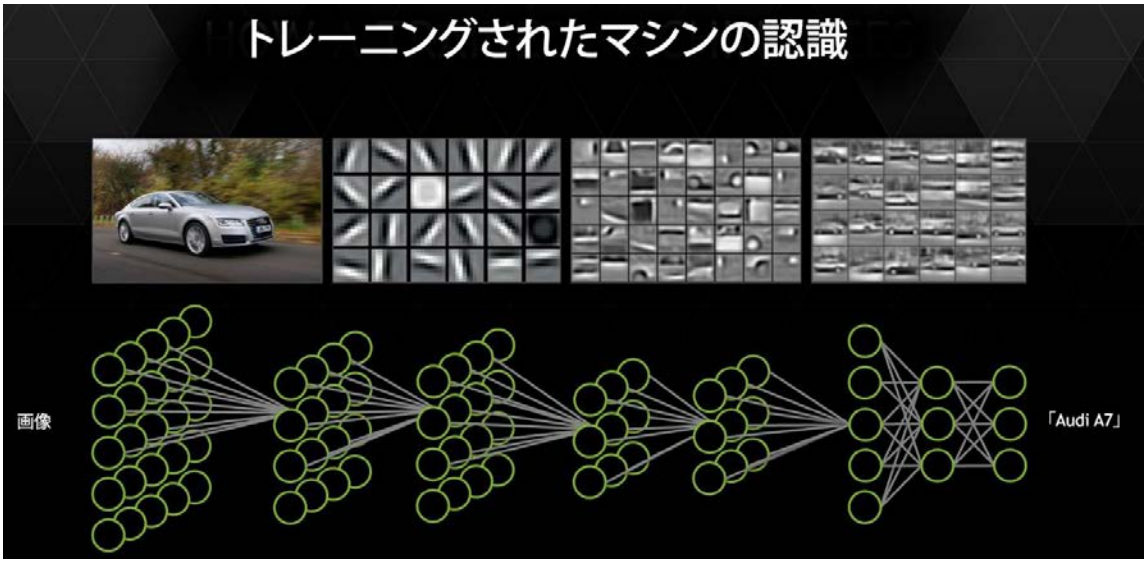
ニューラル ネットワークの主な目的は、ネットワークをトレーニングして予測能力を向上させることです。手書きのゼロを検出するパーセプトロンのモデルは (図 33 を参照)、最初に数字のゼロを構成するそれぞれの特徴に一連の重みを割り当てることによってトレーニングされます。次に、パーセプトロンにゼロを与えて、正しく数字を識別できるかどうかを確認します。結論に到達するまでのこのネットワーク データフローは順伝播フェーズです。ニューラル ネットワークが数字を正しく識別しない場合は、識別エラーの理由と重みを理解し、正しく識別できるようになるまで各特徴の重みを調整する必要があります。さまざまなスタイルで書かれたゼロを正しく識別できるまで、さらに重みの調整を続けます。エラー内容をフィードバックして、各特徴の重みを調整するプロセスは逆伝播と呼ばれます。図の中の複雑に見える数式は、ここで説明したトレーニング プロセスの基本の数学的表現です。

パーセプトロンはきわめてシンプルなニューラル ネットワーク モデルですが、現在、同様の概念に基づく高度な多層ニューラル ネットワークが広く使用されています。オブジェクトを正しく識別するようにネットワークをトレーニングしたら、それを実際の現場に導入して繰り返し推論処理を実行します。推論とは、入力から DNN が有用な情報を抽出するプロセスです。推論の例には、ATM での振り込み伝票の手書き数字の識別、Facebook の写真の顔識別、5000 万人以上の Netflix ユーザーへのお勧め映画の配信などがあります。他にも、車両、歩行者、路上障害物の識別やスピーチのリアルタイム翻訳などが挙げられます。

図 34 は、複数の相互接続を持つ、複雑なパーセプトロンのようなノードで構成された多層ニューラル ネットワーク モデルです。各ノードは、入力されたさまざまな特徴を、相互接続ノードで構成された後続層に出力します。

図 34 のモデルでは、ニューラル モデルの最初の層で自動車の画像をさまざまなセクションに分解し、線や角度などの基本パターンを探します。2 つ目の層で、この線を組み合わせて、ホイール、フロント ガラス、鏡などのより高いレベルのパターンを探します。3 つ目の層で車種を識別し、後続の層では、特定の自動車ブランドのモデル (ここでは Audi A7) を識別します。

ニューラル ネットワークの全結合層に代わる手段として畳み込み層があります。畳み込み層のニューロンは、その下層の小さな領域にあるニューロンにのみ接続されます。通常、この領域はフィルター サイズと呼ばれる 5 x 5 グリッドのニューロンから成ります (7 x 7 または 11 x 11 の場合もあります)。このような畳み込み層は、その入力に畳み込みを実行すると考えることができます。この接続パターンは、一次視覚野細胞や網膜神経節細胞など、脳の知覚領域に見られるパターンを模倣しています。



画像提供: Unsupervised Learning Hierarchical Representations with Convolutional Deep Brief Networks、ICML 2009 & Comm. ACM 2011、Honglak Lee、Roger Grosse、Rajesh Ranganath、Andrew Ng.

図 34. 複雑な多層ニューラル ネットワーク モデルにはさらなるコンピューティング能力が必要

DNN 畳み込み層では、層内の各ニューロンのフィルターの重みは同じです。通常、1 つの畳み込み層は、異なるフィルターを持つ多数のサブ層として実装されます。1 つの畳み込み層に数百のフィルターが使用されることもあります。DNN 畳み込み層は、入力に同時に数百の異なる畳み込みを実行し、結果を次の層に提供します。畳み込み層を持つ DNN は、畳み込みニューラル ネットワーク (CNN) と呼ばれます。

NVIDIA GPU: ディープラーニングのエンジン

最先端の DNN と CNN では、逆伝播を使用して数百万から数十億のパラメーターを調整できます。また、DNN は、精度を高めるために大量のトレーニング データが必要です。つまり、数十万から数百万の入力サンプルを双方向のパスで実行する必要があります。

GPU は、速度とエネルギー効率のどちらにおいても従来の CPU ベースのプラットフォームより優れており、ディープ ニューラル ネットワークのトレーニングにおける GPU 活用が産業界や学術界で広く認められています。多数の同一ニューロンから成るニューラル ネットワークは本質的に高度に並列化されており、これが GPU に自然にマッピングされることで、CPU 単独よりも大幅にトレーニングを加速します。

ニューラル ネットワークは行列数値演算に大きく依存し、複雑な多層ネットワークは、効率と速度を向上するために膨大な量の浮動小数点演算能力と帯域幅を必要とします。数千のプロセッシング コアを搭載した GPU は、行列数値演算に最適化され、数十から数百 TFLOPS のパフォーマンスを発揮します。これは、ディープ ニューラル ネットワークベースの人工知能と機械学習のアプリケーションに最適なコンピューティング プラットフォームです。

ディープ ニューラル ネットワークのトレーニング

最先端のニューラル ネットワークには、逆伝播で調整する数百万から数十億のパラメーターがあります。さらに、収束の精度を高めるために大量のトレーニング データが必要です。つまり、数十万から数百万の入力サンプルを双方向のパスで実行する必要があります (**図 35** を参照)。

複雑なニューラル ネットワークのトレーニングは、基本レベルで数兆回の浮動小数点の乗算や加算などの演算を含むため、膨大な並列コンピューティング能力が必要になります。初期の GPU ニューラル ネットワークのトレーニングでは、NVIDIA Fermi と Kepler の GPU アーキテクチャで利用可能な数千のコアで単精度浮動小数点演算 (FP32) を使用してこのような演算を並列実行していました。このアーキテクチャのコアは、単精度 FP32 データ型と倍精度 FP64 データ型をサポートし、高速で高精度の浮動小数点演算が可能な FMA 命令を使用して、主に HPC 向けに最適化されていました。

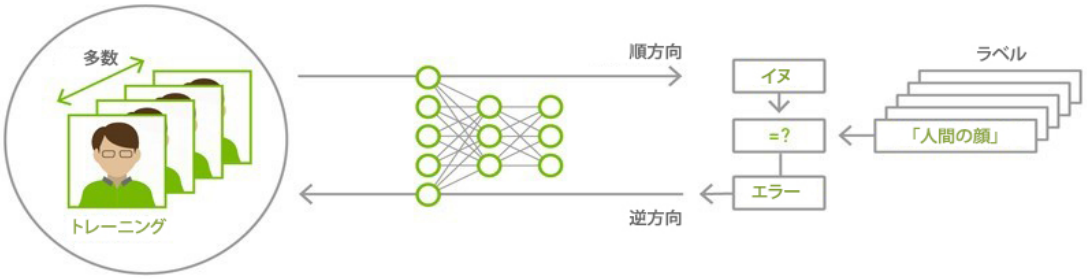


図 35. ニューラル ネットワークのトレーニング

ディープラーニングの現場でのさらなる研究開発により、多くの場合、ニューラル ネットワークは半精度 FP16 データ型を使用して FP32 データと同じレベルのトレーニング精度を実現できることが判明しました。FP16 データのみの収束に対応していないネットワークのトレーニングもありますが、ネットワークの畳み込み層の大部分に低精度のデータ型、結果の蓄積に高精度のデータ型4を使用することで解決できるとの調査結果が出ています[4]

FP16 データを使用すると、より高精度の FP32 や FP64 よりも、ニューラル ネットワークのメモリ使用量と帯域幅の要件を軽減して、大幅に高速化できます。たとえば、NVIDIA Pascal GPU アーキテクチャでの FP16 演算パフォーマンスは FP32 演算の 2 倍、FP16 データ転送は FP32 データ転送よりも速く、使用するメモリ帯域幅は半分になります。

トレーニング済みニューラル ネットワークを使用した推論

ニューラル ネットワークのトレーニングは、大量の入力データ、エラー検出のための順方向パス、ネットワークの各層の数百万のニューロンの重みを調整するための逆方向パスなどを必要とする、高度な処理プロセスです。推論のプロセスは、それほど高い処理能力を必要としませんが、トレーニング済みのネットワークを、画像識別やスピーチ翻訳などの処理を実行したことがない新しい入力に適用して、新しい情報を推論するため、遅延の影響を受けやすくなります (図 36 を参照)。

[4] https://arxiv.org/abs/1412.7024 (英語)

半精度 FP16 データを使用する推論は、FP32[5]. と同じ精度で分類できるという調査結果が出ています。FP16 データ型[6] を使用する場合、Pascal GPU と Tegra X1 SoC のアーキテクチャでの推論のスループットは、最大で FP32 データ型の 2 倍になります。確度の低下を最小限に抑えながら推論のスループットを格段に高速化する 8 ビット整数 (INT8) の精度を使用した推論も可能です。

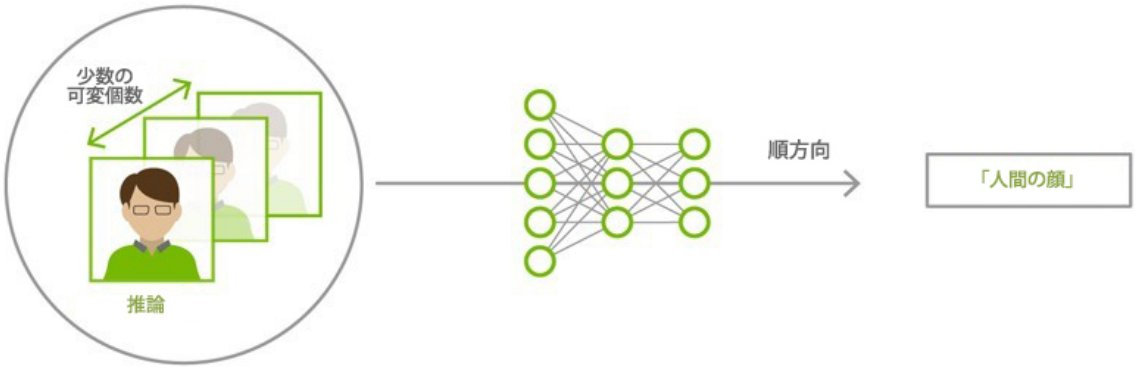


図 36. ニューラル ネットワークでの推論

このようなメリットを活かすために、以前の Pascal GP100 アーキテクチャは FP16 データ形式をネイティブでサポートし、さらなる推論パフォーマンスの向上に向けて、NVIDIA Tesla P40、NVIDIA Tesla P4 などの Pascal ベースの GPU は INT8 をサポートしました。

Pascal GP100 ベースの Tesla P100 カードは、FP16 で 21.2 TFLOPS のパフォーマンスを実現します。INT8 演算をサポートする NVIDIA Tesla P40 などの GPU は、約 48 INT8 TOPS のパフォーマンスを実現し、データセンターのサーバーの推論パフォーマンスをさらに向上します。前述しているとおり、Volta の Tensor コアは、推論とトレーニングの両方に対して、最大 125 TFLOPS というまったく新しいレベルのパフォーマンスを発揮します。

[5] https://arxiv.org/pdf/1502.02551.pdf (英語)

[6] https://www.nvidia.com/content/tegra/embedded-systems/pdf/jetson_tx1_whitepaper.pdf (英語)

包括的なディープラーニング ソフトウェア開発キット

AI イノベーションの勢いは驚異的です。プログラミングの容易さと開発者の生産性は最高レベルに到達しています。NVIDIA の CUDA プラットフォームの豊富なプログラミング機能は、研究者の技術革新をさらに加速します。NVIDIA は、クラウド、データセンター、ワークステーション、組み込みプラットフォームで革新的な GPU アクセラレーション対応の機械学習アプリケーションをサポートするため、NVIDIA DIGITS、cuDNN、cuBLAS、などの高性能ツールとライブラリを備えたディープラーニングのソフトウェア開発キット (SDK) を提供します。

開発者は、あらゆる場所でアプリケーションを作成して展開することを望んでいます。NVIDIA GPU は、世界中のあらゆる PC OEM を通じて入手でき、デスクトップ、ノートブック、サーバー、スーパーコンピューター、さらには Amazon、Google、IBM、Facebook、Baidu、Microsoft などの主要なクラウドで使用できます。インターネット企業をはじめ、研究開発やスタートアップなどの主要な AI 開発フレームワークは、すべて NVIDIA GPU アクセラレーションに対応しています。どの AI 開発システムを利用しても、GPU アクセラレーションによる高速化が実現します。

あらゆる種類のインテリジェント マシンで DNN を活用できるように、NVIDIA はほぼすべてのコンピューティング フォームファクターに対応した GPU を開発しました。PC 用の GeForce、クラウドとスーパーコンピューター用の Tesla、ロボットとドローン用の Jetson、自動車用の DRIVE PX 2 などです。これらはすべて同じアーキテクチャを採用しており、ディープラーニングを高速化します (図 37 を参照)。

図 37. すべてのフレームワークを高速化

Baidu、Google、Facebook、Microsoft は、ディープラーニングと AI 処理用の NVIDIA GPU をいち早く導入した企業です。そして実際に AI テクノロジを、会話への応対、スピーチやテキストの他言語翻訳、画像認識と自動タグ付け、さらにはニュースフィード、エンターテイメント、製品などのレコメンデーションなどに活用しています。

スタートアップや大手企業は、競って AI を使用した新製品やサービスを提供し、事業を改善しています。この 2 年間だけで、NVIDIA とディープラーニング分野で協力する企業の数は約 13 倍も増加し、19,000 社を超えました (図 38 を参照)。

医療、生命科学、エネルギー、金融サービス、自動車、製造、エンターテイメントなどの分野では、膨大なデータからインサイトを推論することによって、多くの利益を得ることができます。Facebook、Google、Microsoft などの企業が、だれでも使用できるディープラーニング プラットフォームを築くことで、AI を活用するアプリケーションは迅速に普及するでしょう。

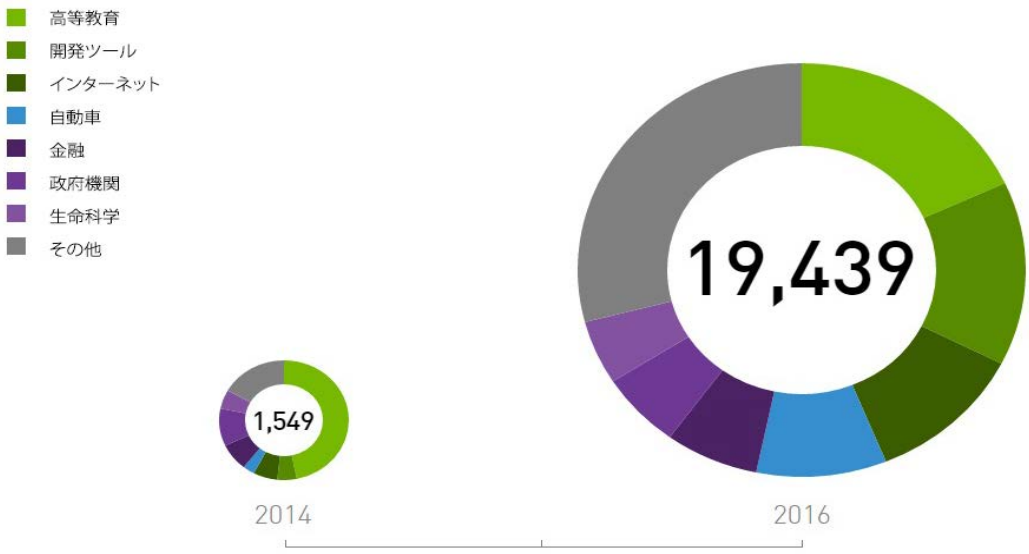


図 38. ディープラーニング活用で NVIDIA と協力している組織

自動運転車

間よりも優れた助手の運転サポート、個人向けの移動サービス改革、肥大化する都市部の駐車場ニーズの削減など、自動運転車は社会に驚くべきメリットをもたらす可能性を秘めています。運転は複雑です。冷たい雨が凍結して道路が滑りやすくなったり、目的地へ向かう道路が閉鎖されていたり、子供が飛び出してきたりと、想定外のことが次々に起こります。自動運転車が遭遇するかもしれないシナリオをすべて予測するソフトウェアを作ることはできませんが、ディープラーニングの真価は、学習し、適合し、向上できるという点です。NVIDIA は、NVIDIA DRIVE PX 2、NVIDIA DriveWorks、および NVIDIA DriveNet (図 39 を参照) を使用して、トレーニング システムや車内搭載の AI コンピューターなど、自動運転車用のエンドツーエンドのディープラーニング プラットフォーム ソリューションを構築しています。今後、次々に優れたサービスが生まれることでしょう。人間の能力をはるかに超えたロボット ナビゲーションや無人シャトルが登場する未来は、もはや SF の世界の話ではありません。

Daimler は、NVIDIA DriveNet によって、従来のコンピューターの能力を上回り、より人間のレベルに近い自動車の環境認知能力を開発しています。

パートナーである Audi のデータセットを使用して、NVIDIA DriveNet で過酷な積雪環境でも自動車を検出できるように、NVIDIA の技術者が短期間のトレーニングを実施しました。

図 39.　NVIDIA DriveNet

ロボット

大手製造ロボット メーカーである FANUC は、不規則に置かれた対象物を容器から取り出すことを学習する組立ライン ロボットを実証しました。このロボットは、GPU を活用して試行錯誤手法で学習します。このディープラーニング テクノロジは、ウォールストリート ジャーナルの「日本が目指す人工知能による技術大国の復活」という記事で紹介された企業、プリファード・ネットワークスが開発したものです。

NVIDIA は、2017 年 5 月の GTC 見本市で Isaac という真に革新的な新しい AI ベースの仮想ロボット トレーニング シミュレーション システムを発表しました。Isaac システムは、高再現度ロボット シミュレーションと高度なリアルタイム レンダリングを実現する開発ツール スイートを備えています。これにより、開発者は、複数の仮想ロボットに複製可能な詳細かつ現実的なテスト シナリオでトレーニングを実施できます。これまで数か月かかっていたシミュレーションは、わずか数分で実行できます。また、システムが完全に仮想であるため、故障や損傷のリスクもありません。シミュレーションが完了すれば、トレーニング済みの AI を即座に実世界のロボットに移行できます。開発者は、仮想と現実の 2 つの環境間で学習成果を交換しながら、テスト手法を反復して調整します。Isaac は、Epic Games の Unreal Engine 4 の拡張バージョンの上に構築されており、NVIDIA の高度なシミュレーション、レンダリング、およびディープラーニングのテクノロジを使用します。

医療と生命科学

Deep Genomics では、GPU ベースのディープラーニングを活用して、遺伝的変異が病気につながるしくみを研究しています。Arterys は、GPU を活用したディープラーニングにより、医療用の画像分析を高速化しています。このテクノロジは GE Healthcare の MRI 装置に導入されており、心臓病の診断に利用されています。Enlitic は、ディープラーニングを使用して医療用の画像を分析し、肉眼では見えない小さな腫瘍などの病状を識別しています。

これらは、GPU と DNN がさまざまな分野における人工知能と機械学習をどのように変革しているかに関する例のほんの一部です。その他に何千もの用途に応用されています。

ディープラーニングの躍進により、さまざまなレベルにおける AI 能力が高速化され、GPU アクセラレーション対応のディープラーニング、AI システム、およびアルゴリズムを活用するさまざまな現場が飛躍的に進化しています。

お読みください

本書に記載される情報は、提供時点において正確かつ信頼できると考えられているものです。ただし、NVIDIA Corporation（以下「NVIDIA」という）は、これらの情報の正確性と完全性について、明示的か黙示的かを問わず、一切の表明も保証も行うものではありません。これらの情報の使用の結果として、もしくはこれらの情報の使用に起因して第三者の特許権またはその他の権利の侵害が発生しても、NVIDIA は一切責任を負わないものとします。本書は、過去に提供された可能性のある本製品に関する他のすべての仕様に優先し、それに代わるものです。

NVIDIA は、この仕様に対する訂正、修正、拡充、改善、その他の変更を随時行える権利と、任意の製品またはサービスを通知なしに終了する権利を留保します。お客様は、注文を行う前に最新の関連仕様を入手し、それらの情報が最新かつ完全であることを確認する必要があります。

NVIDIA とお客様のそれぞれの承認を得た担当者によって署名された個別の販売契約に別段の定めがない限り、NVIDIA 製品は、注文確認時点で提供される NVIDIA の標準的な販売条件に従って販売されます。NVIDIA は、この仕様で参照される NVIDIA 製品の購入に関連した一切の顧客向け一般条件を適用することに明示的に反対します。

NVIDIA 製品は、医療、軍事、航空、宇宙、生命維持の各装置で使用したり、NVIDIA 製品の故障または誤動作の結果、負傷、死亡、物的損害、環境劣化などが起こることを合理的に予想できるような用途で使用したりするよう設計または許可されておらず、また、そのような用途への適合性も保証されていません。NVIDIA は、そのような装置や用途に NVIDIA 製品を含めたり使用したりすることに対して一切の法的責任を負いません。そのため、そのような使用はお客様自身の責任において行っていただきます。

NVIDIA は、これらの仕様に基づく製品が追加的なテストや修正を行わずに特定の用途に適合することを表明するものでも、保証するものでもありません。各製品の全パラメーターのテストが NVIDIA によって実行されるとは限りません。お客様によって計画された用途への製品の適合性を確認し、用途または製品の不履行を避けるために必要なテストを実施することは、お客様側の責任です。お客様の製品設計に含まれる欠点は、NVIDIA 製品の品質および信頼性に影響する可能性があり、その結果、この仕様には含まれていない追加的あるいは異なる条件や要件が生じる可能性があります。NVIDIA は、次に基づく、またはそれに起因する一切の不履行、損害、コスト、あるいは問題に対しても責任を負いません。(i) この仕様に違反する方法で NVIDIA 製品を使用すること。(ii) お客様の製品設計。

この仕様の下では、明示か黙示かを問わず、NVIDIA の特許権、著作権、その他の知的財産権が適用されるいかなるライセンスも供与されません。サードパーティ製品またはサービスに関して NVIDIA によって公開される情報は、それらの製品またはサービスを使用するための NVIDIA からのライセンスを構成するものでも、それらの製品またはサービスを保証もしくは是認するものでもありません。これらの情報を使用するには、サードパーティの特許またはその他の知的財産権の下でサードパーティから提供されるライセンスが必要になるか、NVIDIA の特許またはその他の知的財産権の下で NVIDIA から提供されるライセンスが必要になる場合があります。この仕様に含まれる情報を複製することは、複製が NVIDIA によって書面で承認されており、改変なしで複製されており、かつ、関連するあらゆる条件、制限、および通知を伴っている場合に限り許可されます。

NVIDIA デザイン仕様書、リファレンス ボード、ファイル、図、診断、リスト、およびその他のドキュメント（以下、併せておよびそれぞれ「資料」という）はすべて、「現状有姿」とします。NVIDIA は資料について、明示または黙示、あるいは法定または非法定にかかわらず保証しません。さらに、特定の目的に対する黙示的保証、非抵触行為、商品性、および適正すべてに対する責任を明示的に否認します。お客様が何らかの理由で被るいかなる損害にかかわらず、NVIDIA がここに記載される製品に関してお客様に対して負う累積責任は、本製品の販売に関する NVIDIA の契約条件に従って制限されるものとします。

VESA DisplayPort

DisplayPort および DisplayPort コンプライアンスのロゴ、デュアルモード ソースの DisplayPort コンプライアンスのロゴ、アクティブ ケーブルの DisplayPort コンプライアンスのロゴは、米国およびその他の国における Video Electronics Standards Association の商標です。