



GPU と DPU によるアクセラレーテッド コンピューティング

2022/01/20

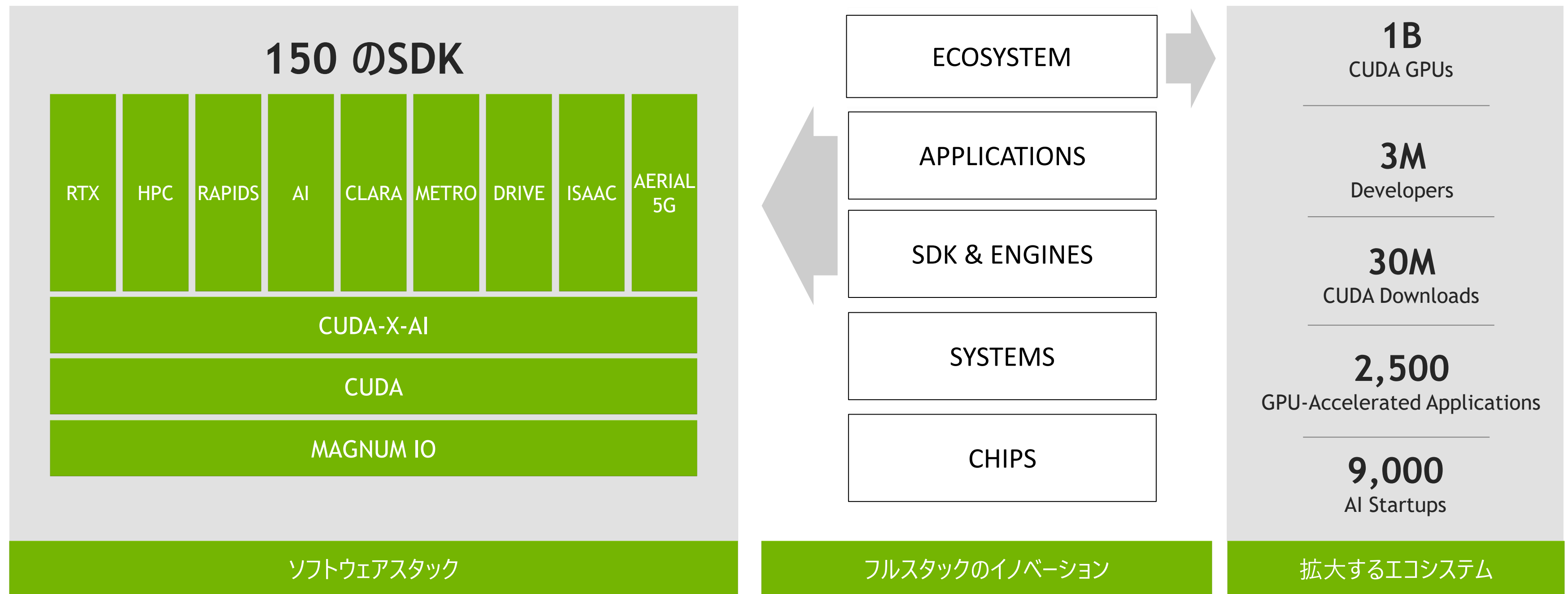
エヌビディア合同会社

NVIDIA コンピューティング プラットフォーム



フルスタックのコンピューティングプラットフォームを提供

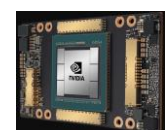
NVIDIA エコシステムの拡大がイノベーションを促進



昨年 11 月の GTC では 65 の更新

CUDA TOOLKIT 11

GPU 対応アプリケーションの開発、最適化、展開



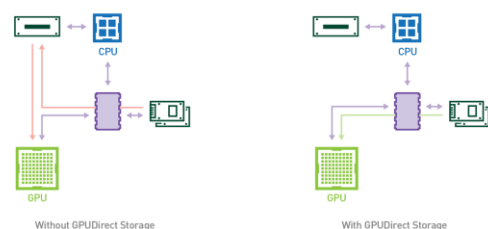
MIG, Tensor
Cores, NVLink



arm

Microsoft
DirectX11 Windows10

GPUDirect
Storage

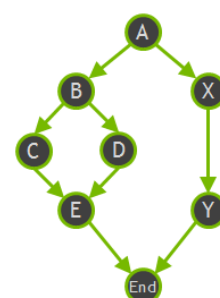
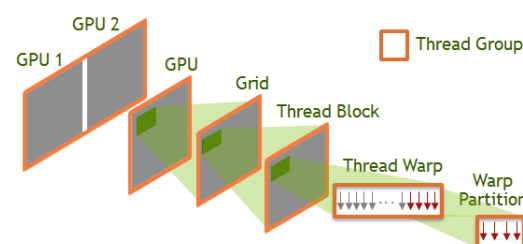


System Memory NVMe PCIe Switch GPUDirect Storage Bounce buffer PCIe

New Platform Capabilities

- A100 Features
- CUDA on Arm Servers

Cooperative
Groups



Fork-Join
Graphs

New Reduce
Op

reduce(tile, value, OP);

Asynchronous
Copy



Programming Model Updates

- Ampere Programming Model
- New APIs for CUDA Graphs
- Flexible Thread Programming
- Memory Management APIs

Nsight Compute



Kernel Profiling with
Rooflining

Nsight Systems



System trace
for Ampere

Developer Tools

- Support for Ampere
- Roofline plots with Nsight
- Next generation correctness tools



libcud++

```
atomicAdd(&h, (half)1.15f);
```

```
half2 hvec(0.94f, -2.13f);  
atomicAdd(&h2, hvec);
```

IEEE-754.2008 FP16 Specification

0	0	1	1	1	0	0	1	0	1	0	0	0
sign					exponent					mantissa		
(5 bits)								(10 bits)				

= 0.707031

$$D = \begin{bmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{10} & A_{11} & A_{12} & A_{13} \\ A_{20} & A_{21} & A_{22} & A_{23} \\ A_{30} & A_{31} & A_{32} & A_{33} \end{bmatrix} + \begin{bmatrix} B_{00} & B_{01} & B_{02} & B_{03} \\ B_{10} & B_{11} & B_{12} & B_{13} \\ B_{20} & B_{21} & B_{22} & B_{23} \\ B_{30} & B_{31} & B_{32} & B_{33} \end{bmatrix}$$

$$D = AB + C$$

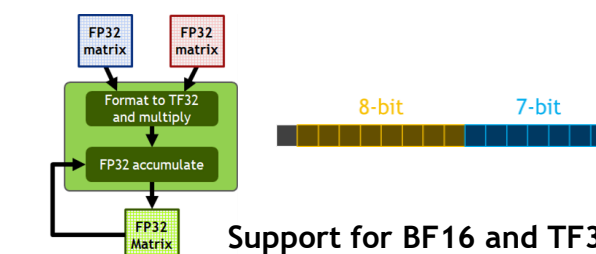


Link Time
Optimization

CUDA C++

- C++ Modernization
- Parallel standard C++ library
- Low precision datatypes and WMMA

Libraries: Hardware decoder
acceleration with nvJPEG



Support for BF16 and TF32
datatypes



Strong and weak
scaling on multi-GPU
systems

Math Libraries

- Low precision datatypes in Ampere
- 3rd Gen Tensor Core support
- Leverage increased memory bandwidth, shared memory and L2 cache

NVIDIA CUQUANTUM

SDK of Optimized Libraries and Tools for Accelerating Quantum Computing Workflows

Enabling Quantum Computing Research

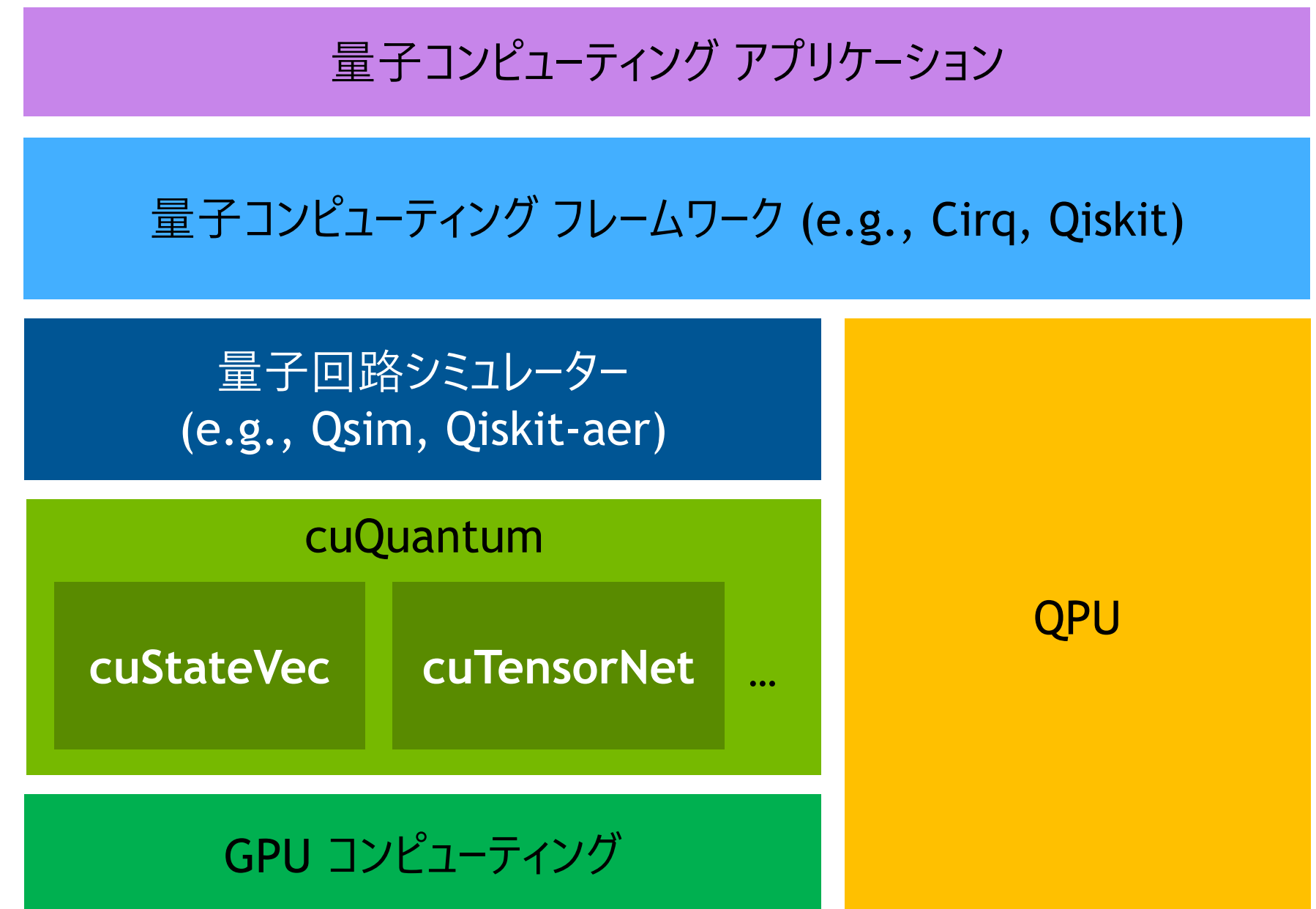
- Accelerate Quantum Circuit Simulators on GPUs
- Enable algorithms research with scale and performance not possible on quantum hardware, or on simulators today

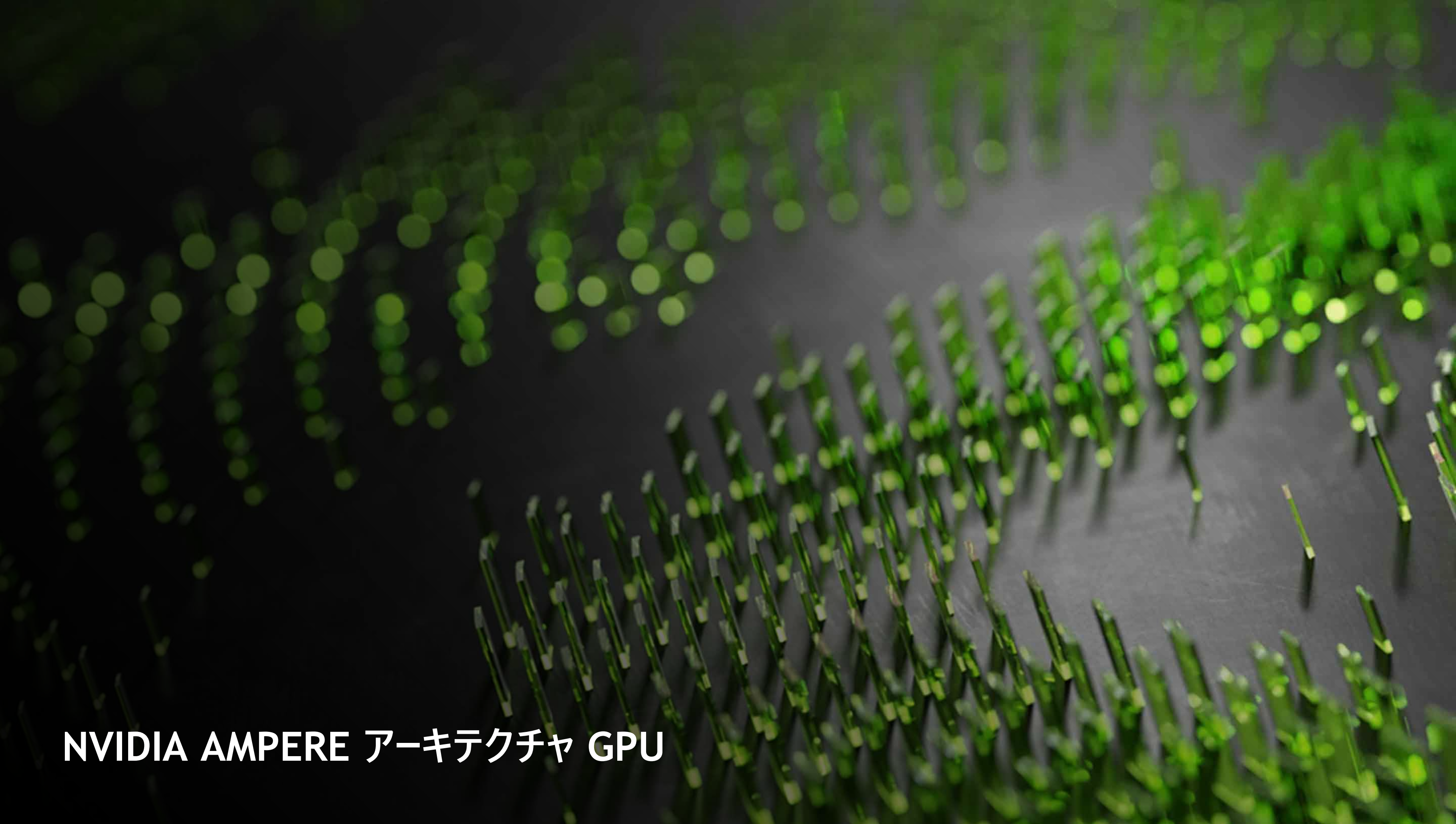
Platform for Algorithm Development

- Leverage Accelerated Circuit Simulators or Quantum Processors
- Enable development of algorithms for scientific computing on hybrid Quantum/Classical systems

Open Beta Available Now

- Integrated in leading quantum computing frameworks Cirq and Qiskit
- Available today at developer.nvidia.com/cuquantum



A close-up, macro photograph of a GPU die. The die is a dark, rectangular silicon chip. It is densely populated with thousands of fine, green, wire-like structures (wire bonds) that connect the die to a substrate. These wires are arranged in a regular, grid-like pattern across the surface of the die. The lighting is dramatic, highlighting the texture of the wires and the smooth surface of the silicon.

NVIDIA AMPERE アーキテクチャ GPU

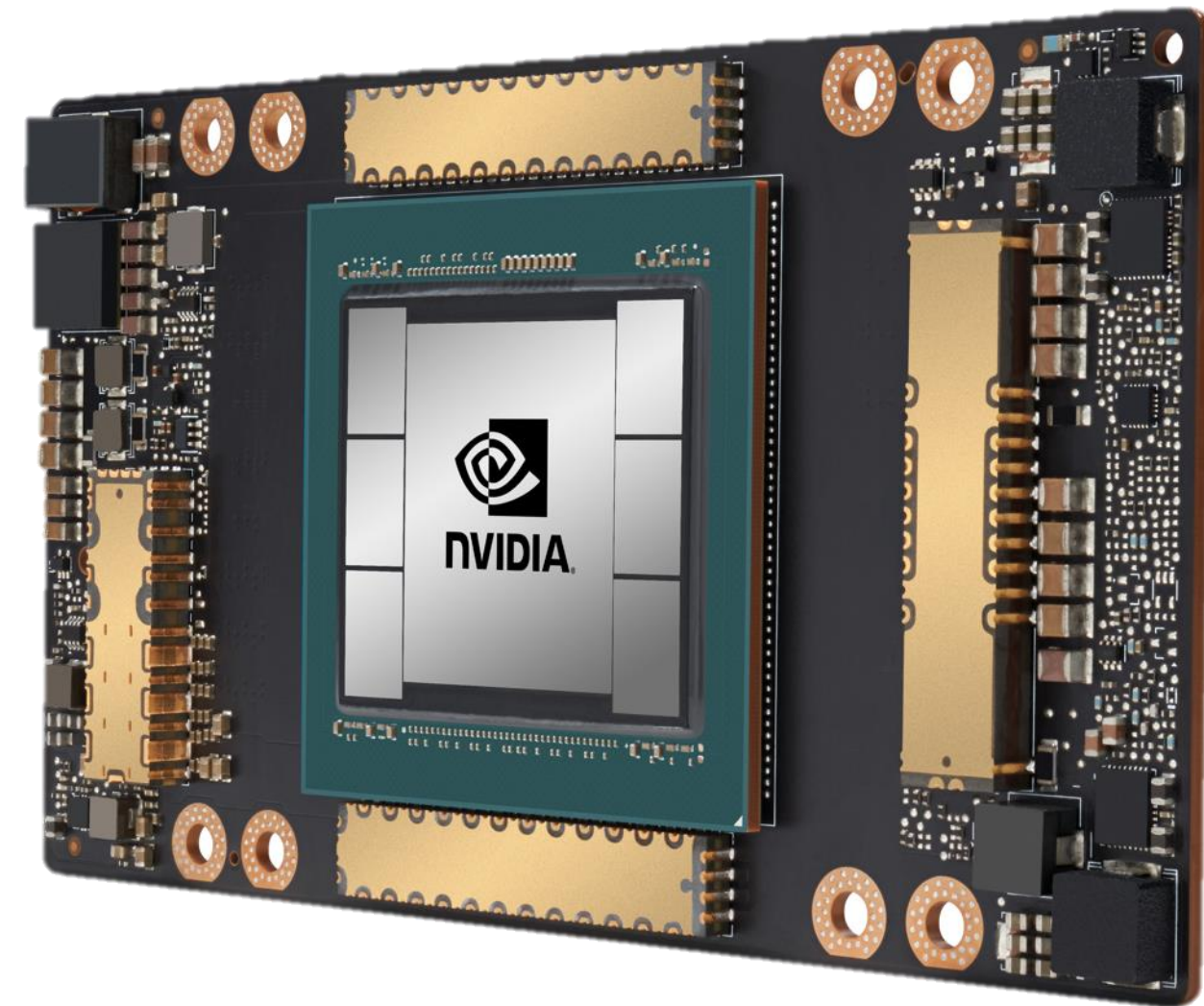
NVIDIA GPU 製品のおおまかな一覧

	Fermi (2010)	Kepler (2012)	Maxwell (2014)	Pascal (2016)		Volta (2017)	Turing (2018)	Ampere (2020)	
データセンター GPU	HPC M2070	K80	M40	P40	P100	V100	T4	A100	A30
	DL 学習			P4					
	DL 推論								
	グラフィック/VDI	K2 K1	M60		P100	V100		A40	A10
Quadro ワークステーション向け	6000	K6000	M6000	P5000	GP100	GV100	RTX 8000	RTX A6000	
GeForce PC 向け	GTX 580	GTX 780	GTX 980	GTX 1080	TITAN XP	TITAN V	RTX 2080	RTX 3090	

NVIDIA A100

かつてない飛躍 - Volta 比最大 20 倍のピーク性能

	ピーク性能	V100 比
FP32 トレーニング	312 TFLOPS	20X
INT8 インファレンス	1,248 TOPS	20X
FP64 HPC	19.5 TFLOPS	2.5X
Multi-instance GPU (MIG)		7X GPUs



54B XTOR | 826mm² | TSMC 7N | 40GB Samsung HBM2 | 600 GB/s NVLink

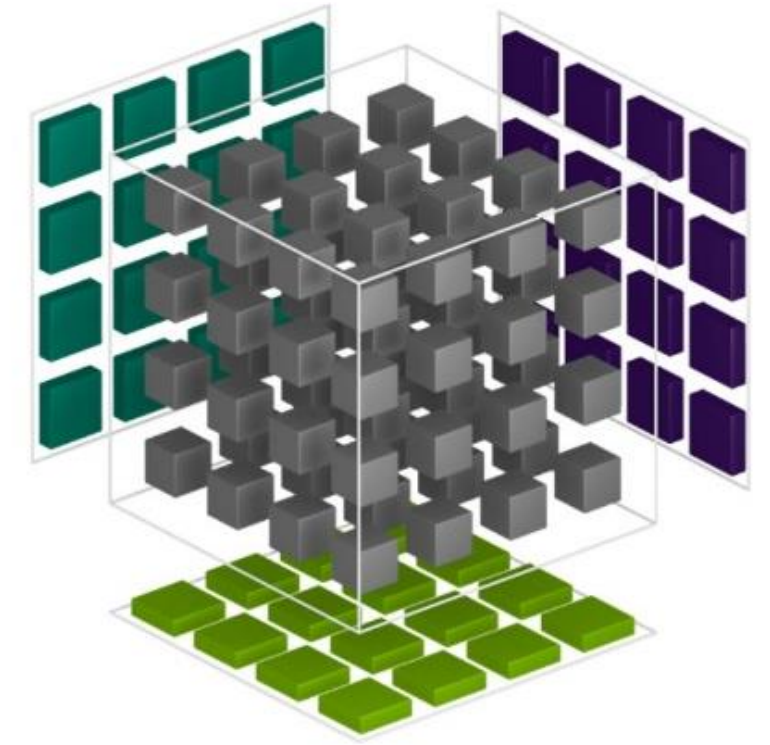
Tensor コア

行列演算ユニット

行列の FMA (Fused Multiply-Add: 融合積和演算)

125 TFLOPS: NVIDIA V100 では FP32 比で 8 倍のピーク性能

NEW! 312 TFLOPS: NVIDIA A100 では FP32 比で 16 倍のピーク性能



$$\begin{matrix} \mathbf{D} & = & \begin{pmatrix} \begin{matrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{matrix} & \begin{pmatrix} \begin{matrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{matrix} & + & \begin{pmatrix} \begin{matrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{matrix} \end{matrix} \\ \text{FP32} & & \text{FP16} & & \text{FP16} & & \text{FP32} \\ (\text{FP16}) & & & & & & (\text{FP16}) \end{matrix}$$

自動混合精度演算 (AMP) の有効化

わずか数行の追加で高速化

TensorFlow

NVIDIA NGC コンテナイメージ 19.07 以降、TF 1.14 以降及び TF 2 以降では、オプティマイザのラッパーが利用可能:

```
opt = tf.train.experimental.enable_mixed_precision_graph_rewrite (opt)
```

Keras mixed precision API in TF 2.1+ for eager execution

https://tensorflow.org/api_docs/python/tf/train/experimental/enable_mixed_precision_graph_rewrite

PyTorch

PyTorch はネイティブに AMP をサポート。詳細は公式ドキュメントを:

<https://pytorch.org/docs/stable/amp.html>

https://pytorch.org/docs/stable/notes/amp_examples.html

MXNet

NVIDIA NGC コンテナイメージ 19.04 以降、MXNet 1.5 以降は、わずかな追加コードで AMP を利用可能:

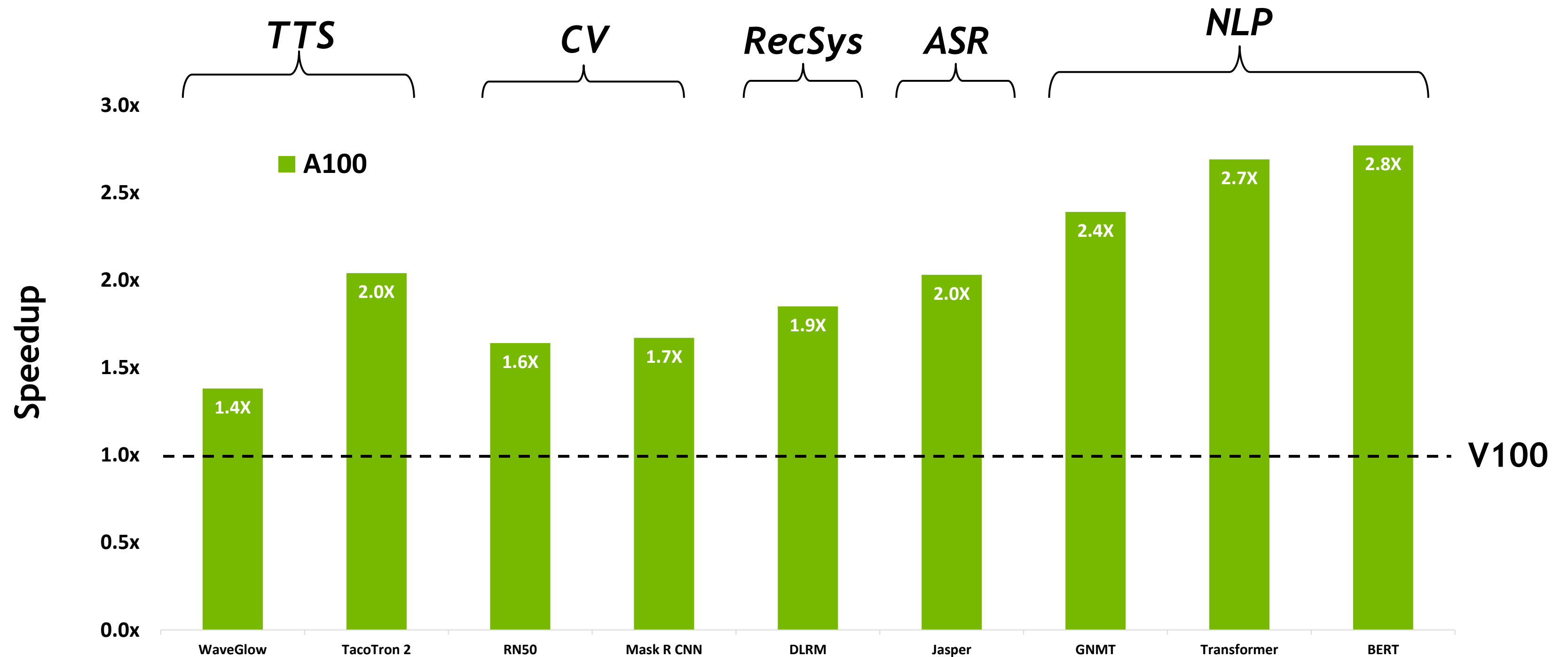
```
amp.init()  
amp.init_trainer(trainer)  
with amp.scale_loss (loss, trainer) as scaled_loss:  
    autograd.backward(scaled_loss)
```

<https://mxnet.apache.org/api/python/docs/tutorials/performance/backend/amp.html>

詳しくはこちら: <https://developer.nvidia.com/automatic-mixed-precision>

AMP による混合精度トレーニングが最大 2.8 倍高速に

V100 (FP16) と A100 (FP16) の比較



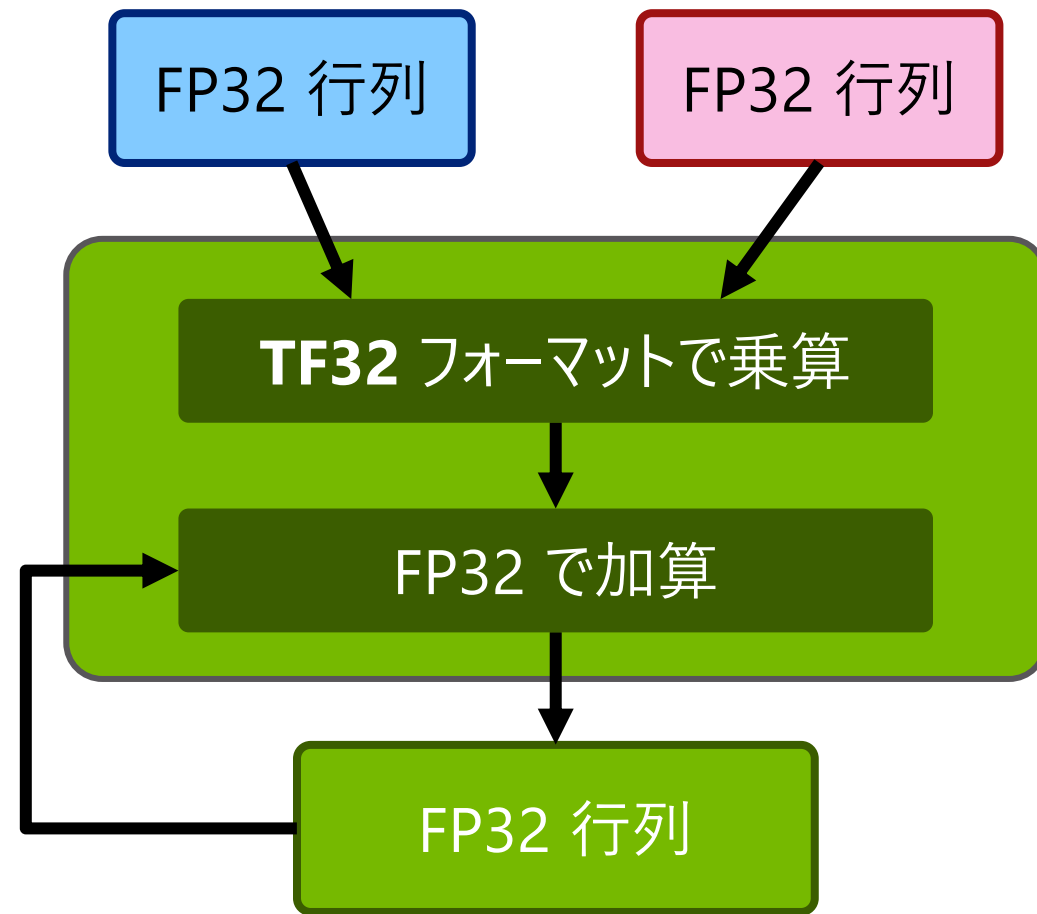
All results are measured

V100 used is DGX-1 (8xV100 16GB). A100 used is s DGX A100 (8xA100 SXM4), except DLRM which uses 1xV100 and 1xA100; all use FP16

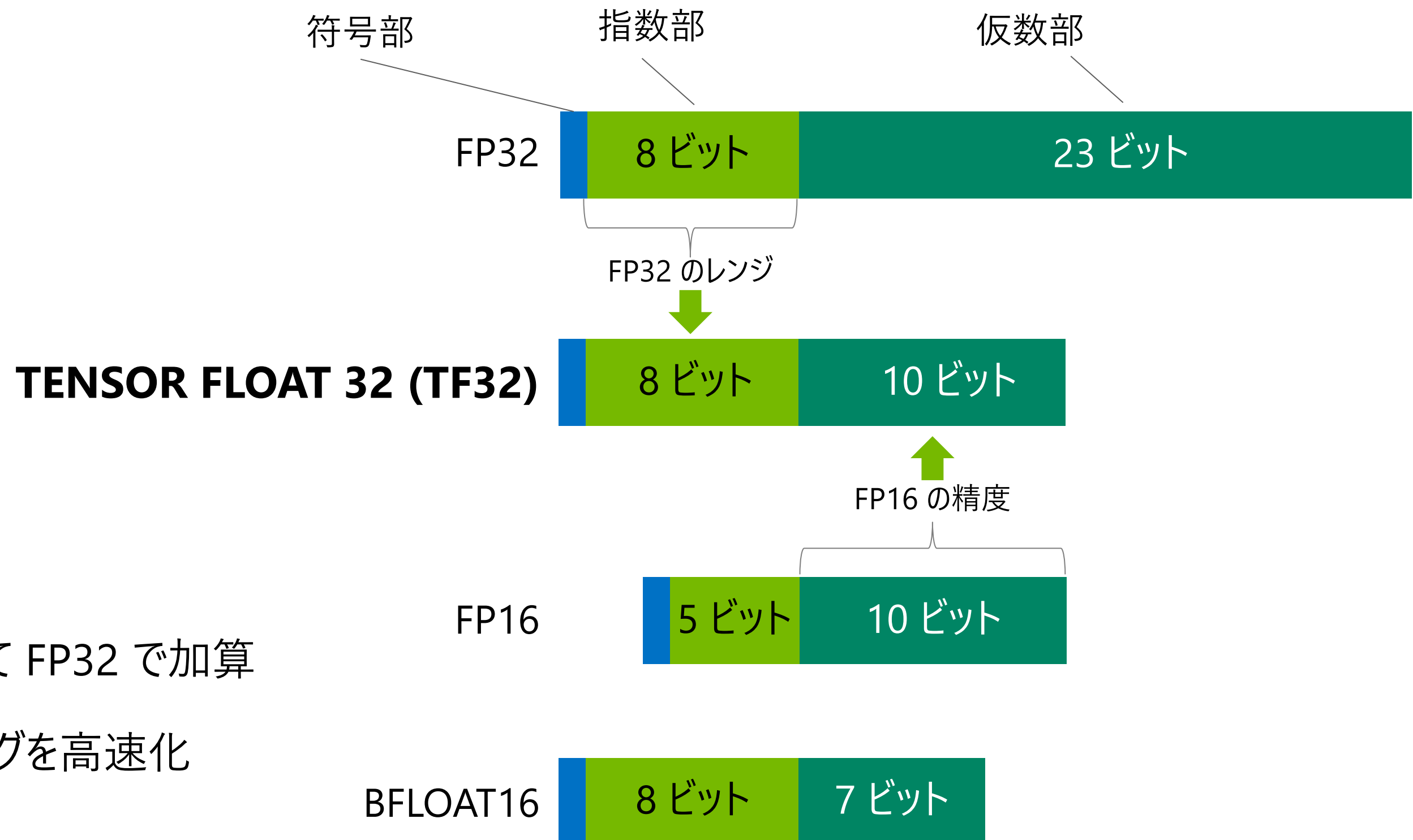
RN50 uses MXNET Batch size = 192, Mask R CNN uses PyTorch BS = 4 (V100) and BS=16 (A100), DLRM uses PyTorch and BS=32768, Jasper uses PyTorch and BS=32 (V100) and 96 (A10), WaveGlow uses PyTorch and BS=10, TacoTron2 uses PyTorch and BS=104 (V100) and 100 (A100), Transformer uses PyTorch and BS=5120 (V100) and 13312 (A100 and GNMT uses PyTorch and BS=128 (V100) and 256 (A100); BERT Pre-Training Throughput using Pytorch including (2/3)Phase 1 and (1/3)Phase 2 | Phase 1 Seq Len = 128, Phase 2 Seq Len = 512

TF32 TENSOR コア

FP32 のレンジと FP16 の精度を合わせ持つ新しい数値データ型

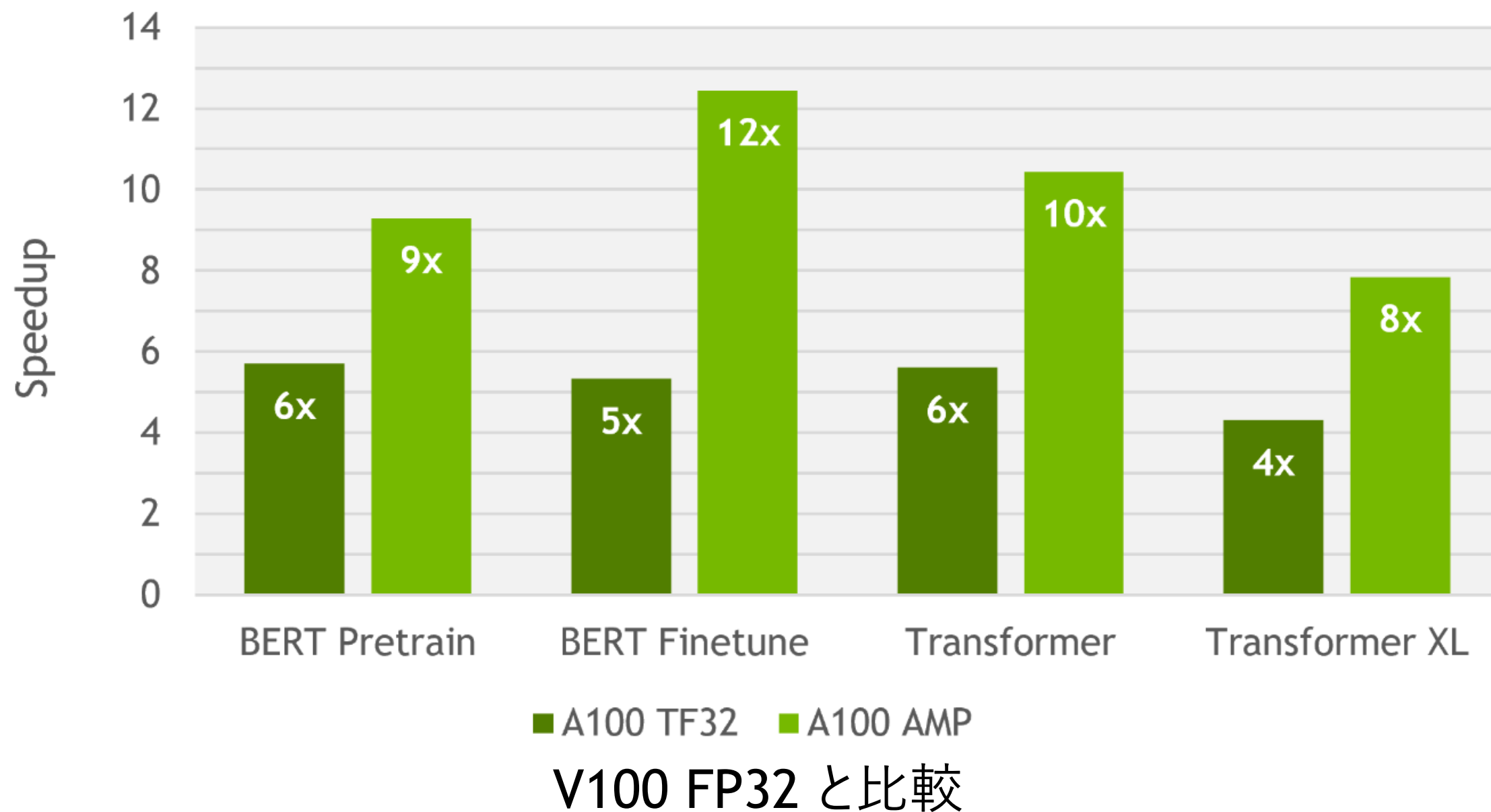


- FP32 の指数部、FP16 の仮数部
- FP32 を受け取り、**TF32** で乗算して FP32 で加算
- コード変更不要でモデルのトレーニングを高速化



TF32 によりコード変更なしで AI トレーニングを高速化

V100 FP32 (CUDA コア) と A100 TF32 (Tensor コア)、A100 FP16 (AMP 利用) の比較



DL フレームワーク (NVIDIA NGC コンテナ):

default: FP32 行列積を **TF32** で加速

cuDNN $\geq$ 8.0:

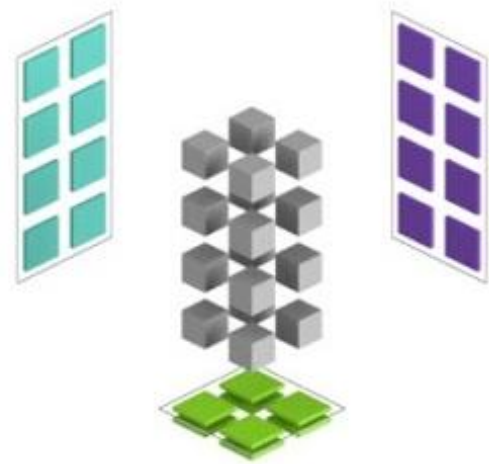
default: FP32 行列積を **TF32** で加速

(*) NVIDIA_TF32_OVERRIDE=0 で TF32 使用を無効化

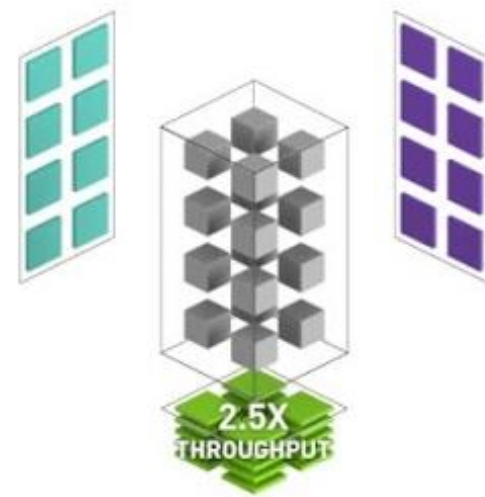
倍精度演算のピーク性能が 2.5 倍に

A100 の Tensor コアは FP64 に対応

NVIDIA V100 FP64

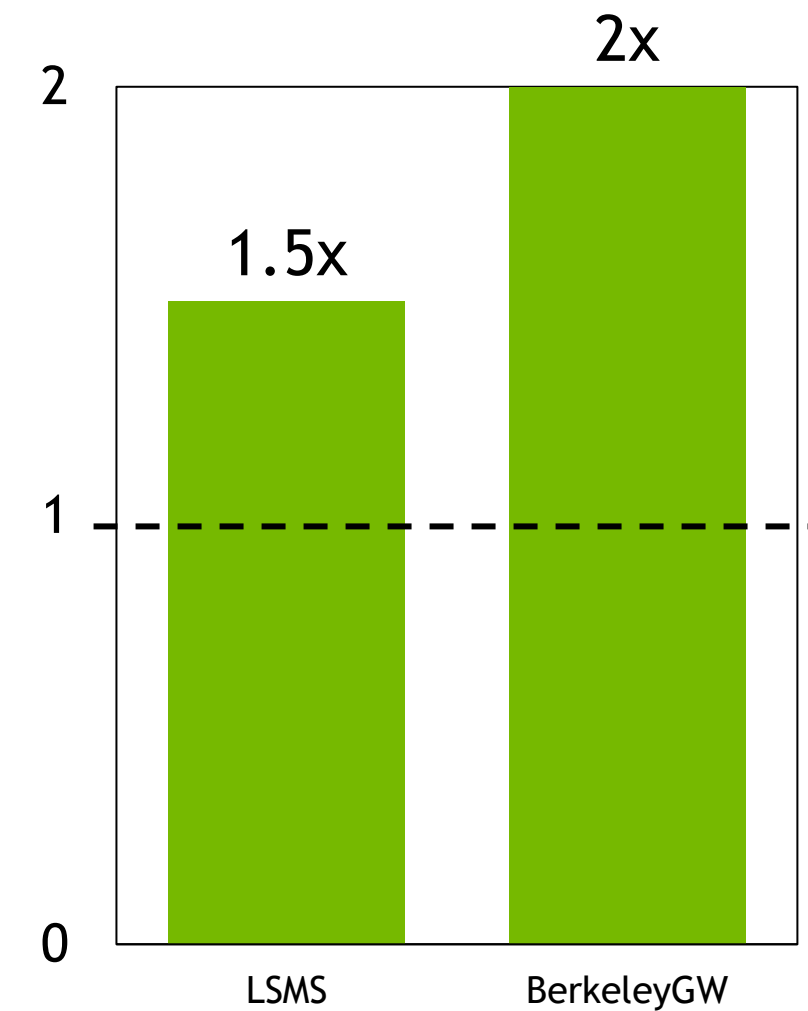


NVIDIA A100 Tensor コア FP64



- IEEE 754 準拠の倍精度浮動小数点数
- cuBLAS, cuTensor, cuSolver 等のライブラリで対応

A100 Speedup vs. V100 (FP64)



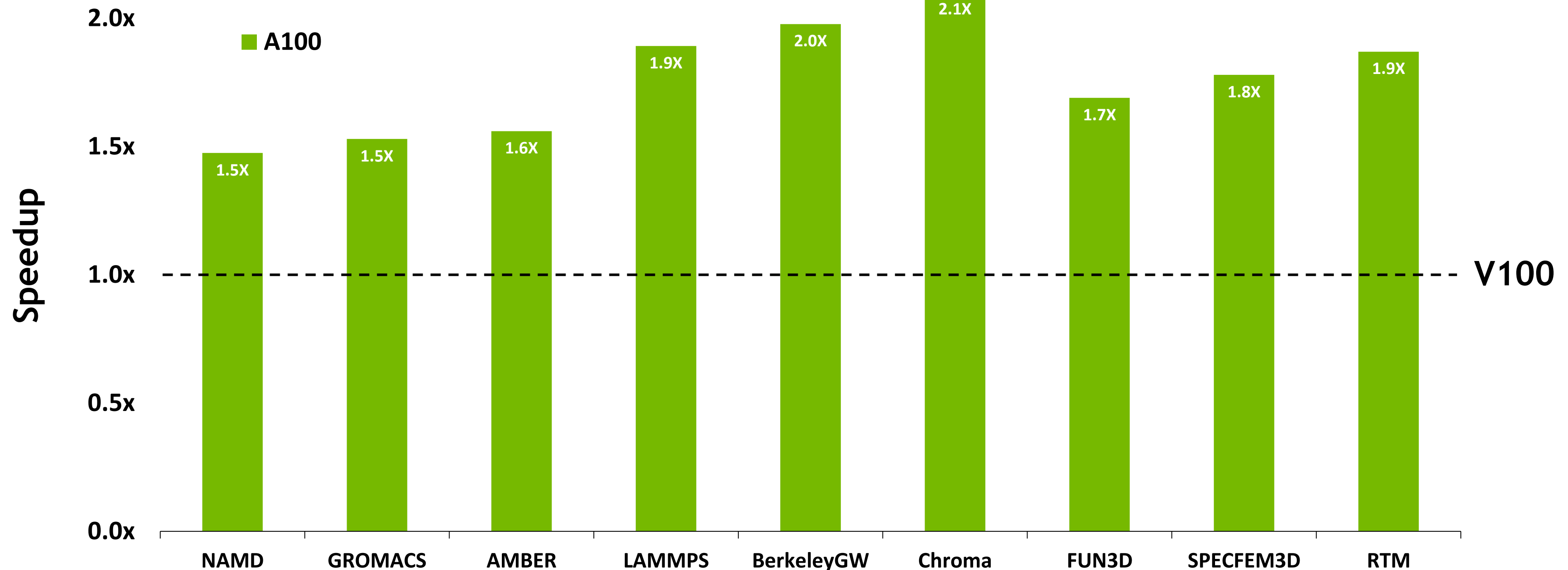
HPC アプリケーションの性能向上

分子動力学

物理

工学

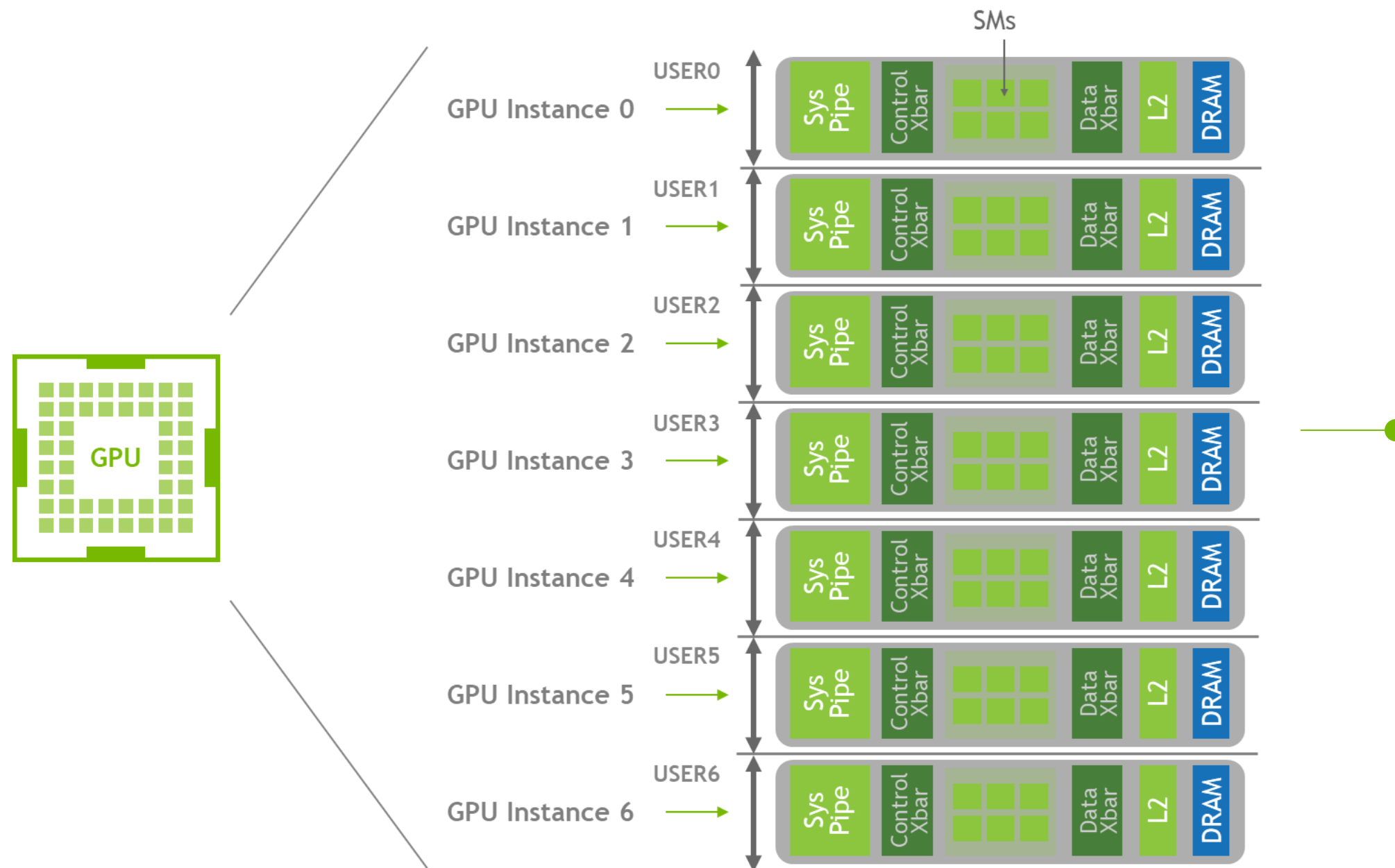
地球科学



All results are measured
Except BerkeleyGW, V100 used is single V100 SXM2. A100 used is single A100 SXM4
More apps detail: AMBER based on PME-Cellulose, GROMACS with STMV (h-bond), LAMMPS with Atomic Fluid LJ-2.5, NAMD with v3.0a1 STMV_NVE
Chroma with szscl21_24_128, FUN3D with dpw, RTM with Isotropic Radius 4 1024^3, SPECFEM3D with Cartesian four material model
BerkeleyGW based on Chi Sum and uses 8xV100 in DGX-1, vs 8xA100 in DGX A100

Multi-Instance GPU (MIG)

GPU をハードウェア分割、複数のユーザーに QoS の確保された GPU アクセスを提供



A100 GPU を最大 7 分割:

パーティション毎に、専用の演算器、メモリ、L2 キャッシュを確保、Noisy Neighbor 問題を回避

QoS を確保しつつ、複数のワークロードを同時に実行:

MIG インスタンスは予測可能なスループットとレイテンシで並列に動作

適切な GPU 割り当て:

ターゲットワークロードに応じて適切なサイズのインスタンスに分割可能

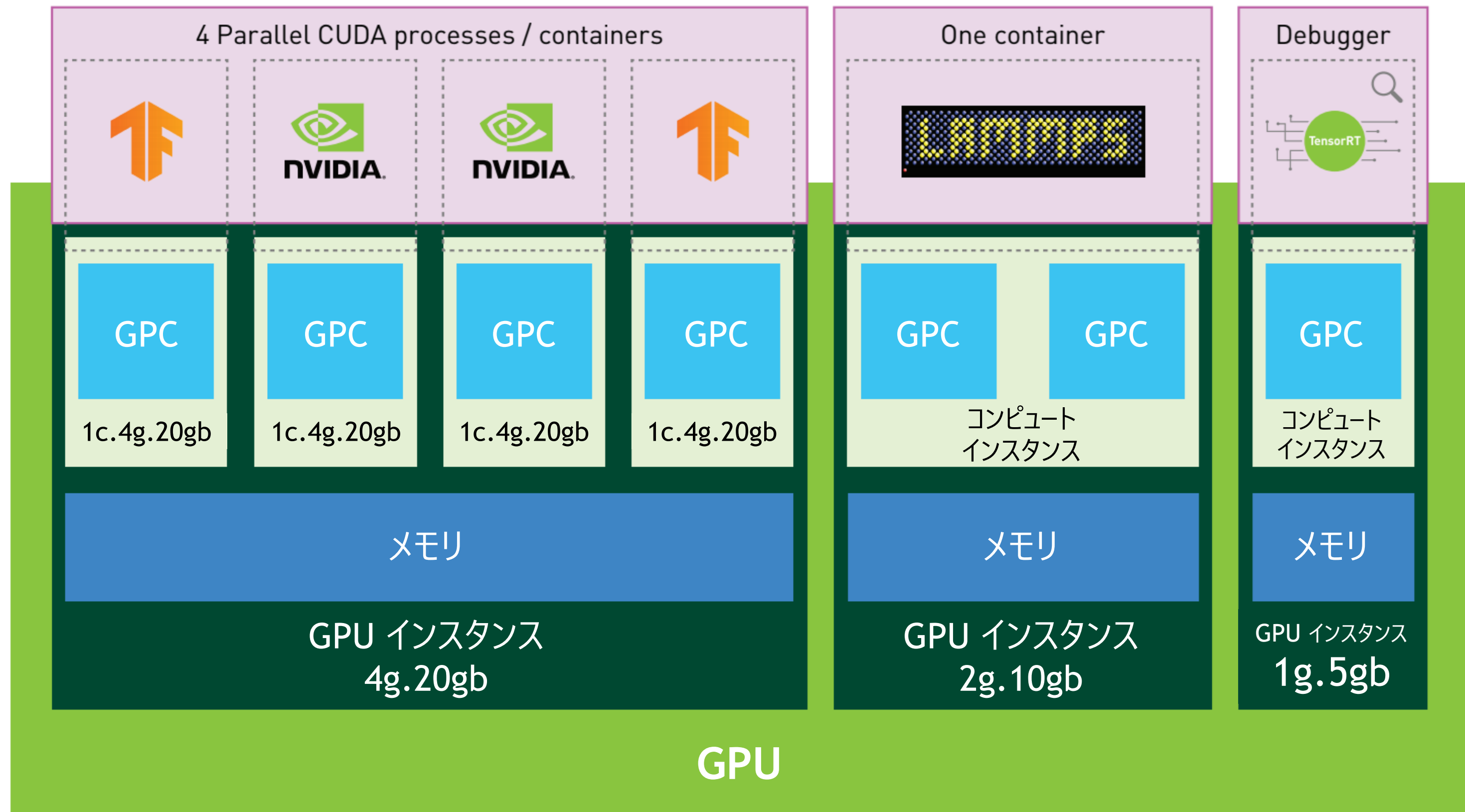
様々な環境で利用可能:

EC2 を含む仮想環境はもちろん、ベアメタル、Docker コンテナ、Kubernetes や Slurm のようなオーケストレーター / ワークロードマネージャの環境をサポート



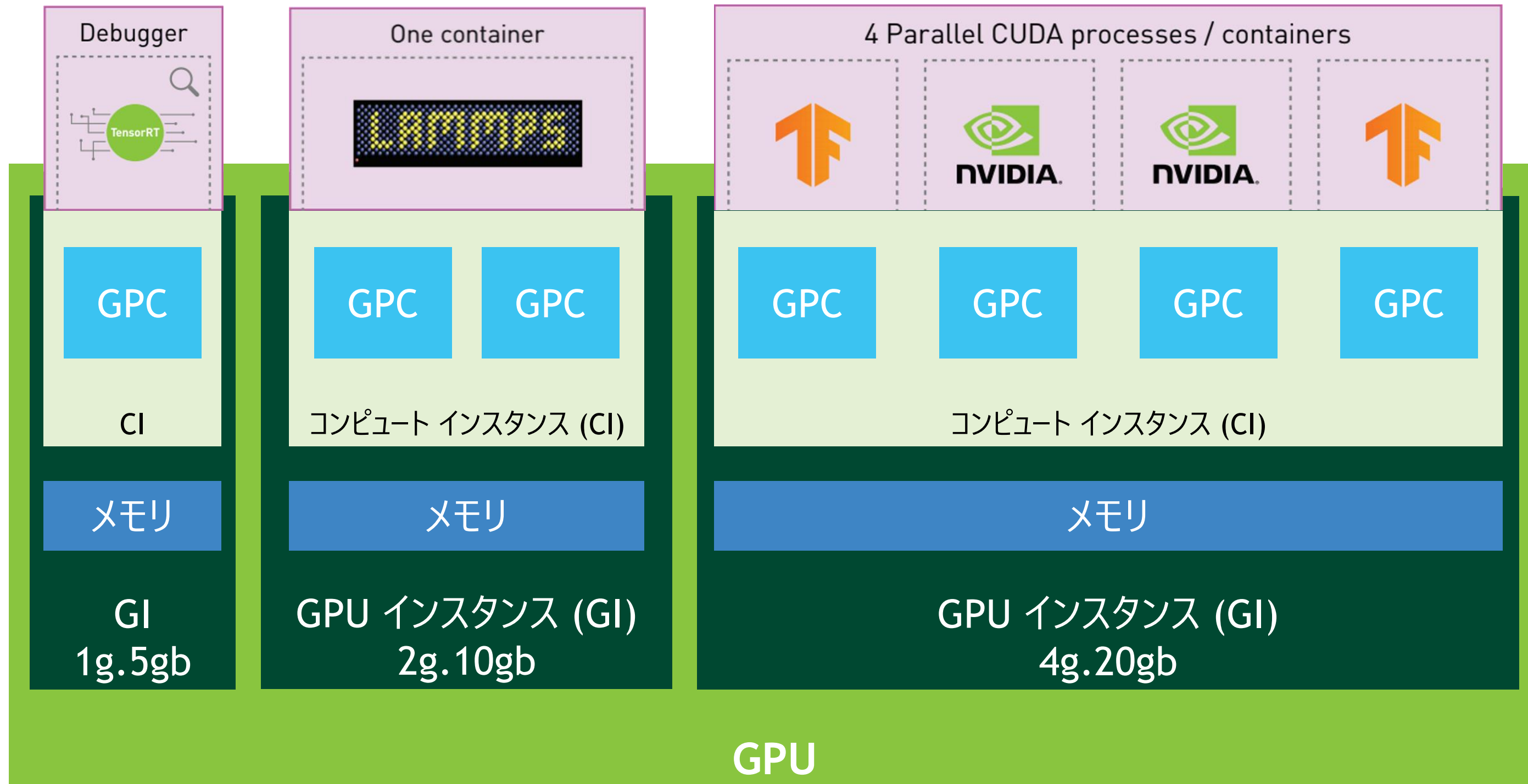
GPU インスタンスとコンピュート インスタンス

GPU をハードウェア分割、複数のユーザーに QoS の確保された GPU アクセスを提供



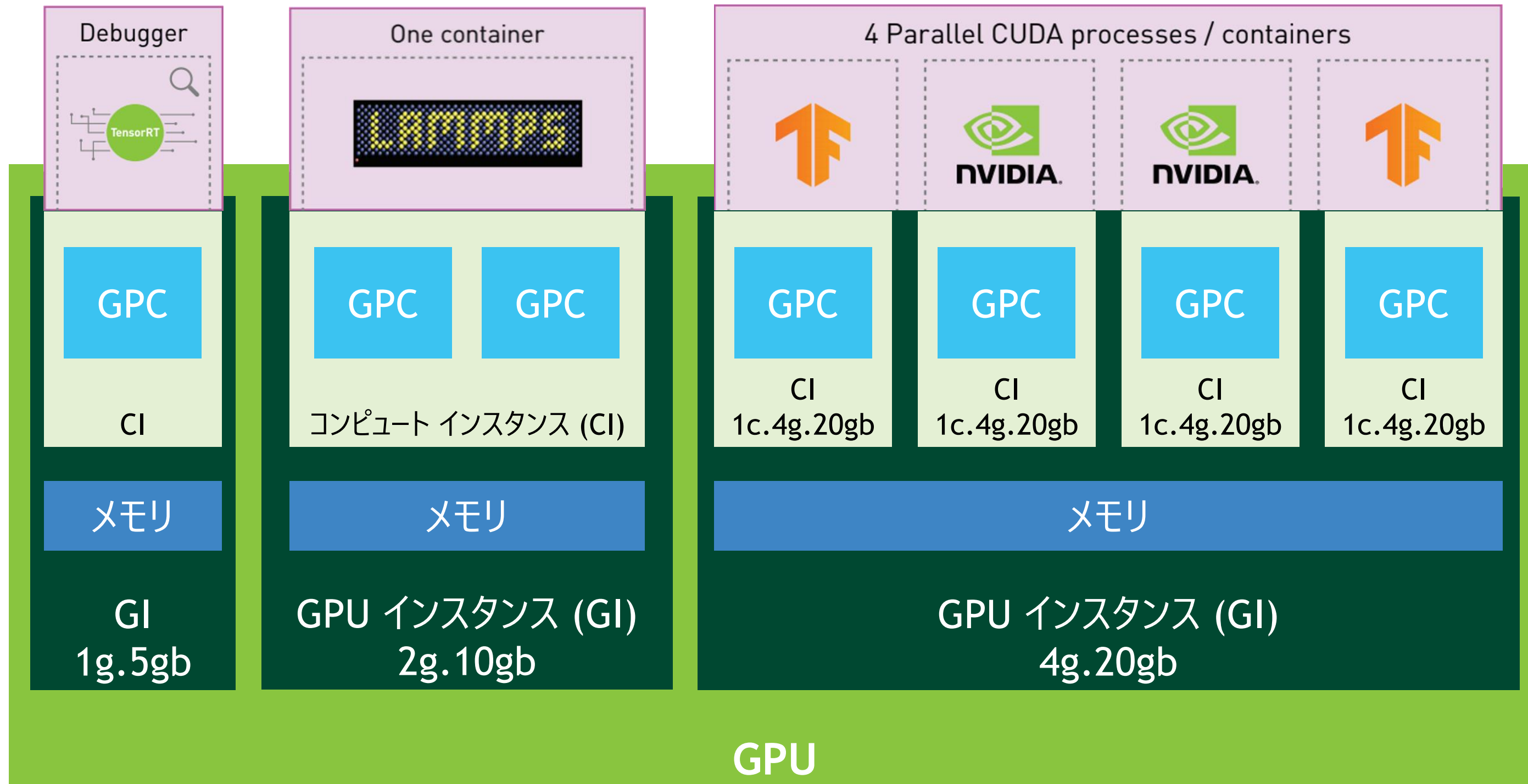
GPU インスタンスとコンピュート インスタンス

1:1 の場合



GPU インスタンスとコンピュート インスタンス

4g.20gb に 4 つの CI



GPU インスタンス プロファイル

For A100-SXM4-40GB

GPU Instance	Number of Instances Available	SMs	Memory	NVDECs	Target use-cases	
					Training	Inference
1g.5gb	7	14	5 GB	0	BERT Fine-tuning (e.g. SQuAD), Multiple chatbots, Jupyter notebooks	Multiple inference (e.g. TRTIS); ResNet-50, BERT, WnD networks
2g.10gb	3	28	10 GB	1		
3g.20gb	2	42	20 GB	2	Training on ResNet-50, BERT, WnD networks	
4g.20gb	1	56	20 GB	2		
7g.40gb	1	98	40 GB	5		

MIG の有効化と無効化

- GPU 毎に有効化・無効化
- MIG 有効と無効の GPU が混在可能
- 使用中のGPUは「保留」状態となり、再起動後に変更が有効に
- MIG の有効化・無効化にはroot権限が必要
- 一度設定した有効・無効の状態は、サーバーの再起動後も有効

```
root@dgxa100-01:~# systemctl stop nvsm
```

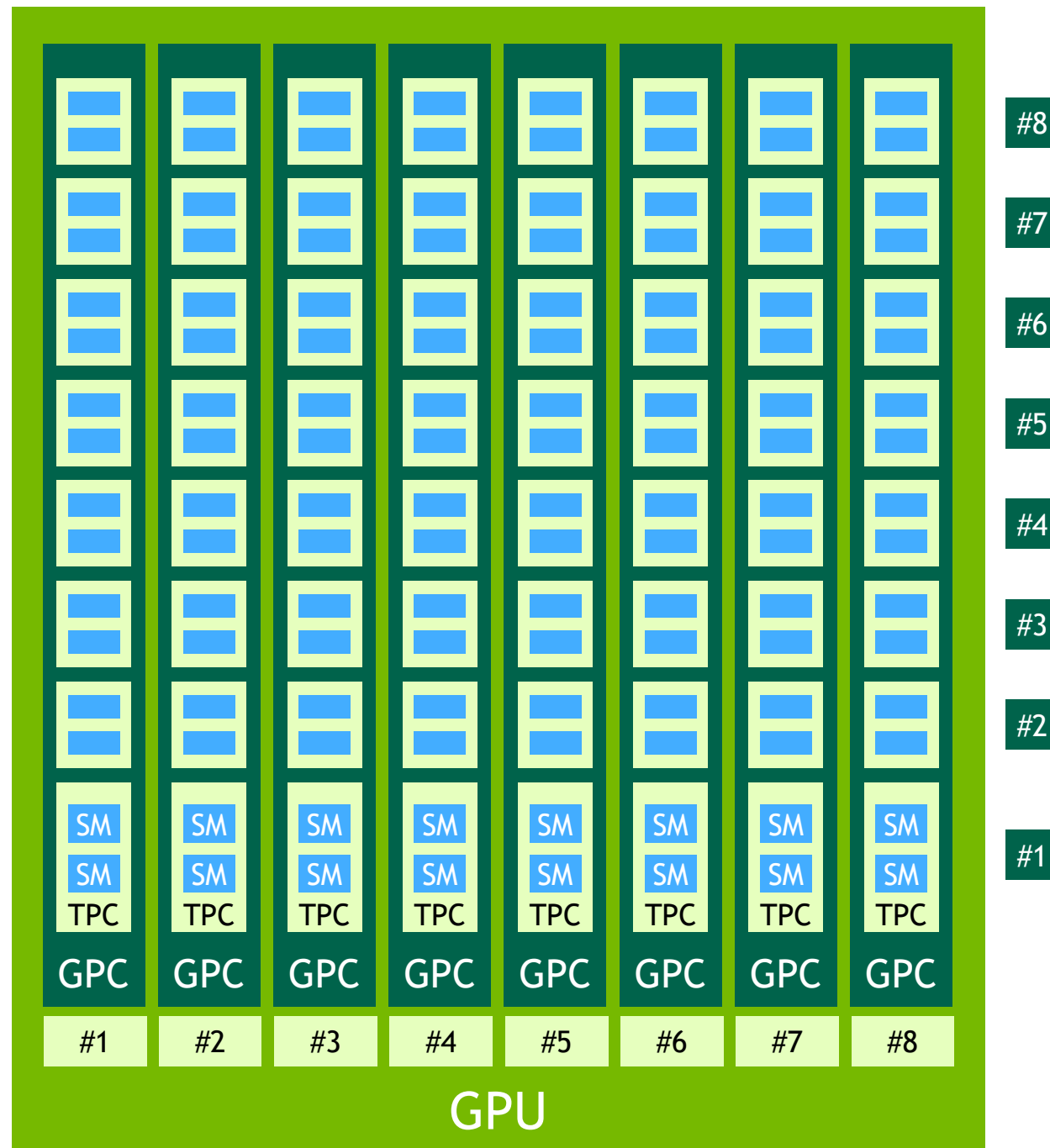
```
root@dgxa100-01:~# systemctl stop dcgm
```

```
root@dgxa100-01:~# nvidia-smi -i 0 -mig 1  
Enabled MIG Mode for GPU 00000000:07:00.0  
All done.
```

GA100 と MIG

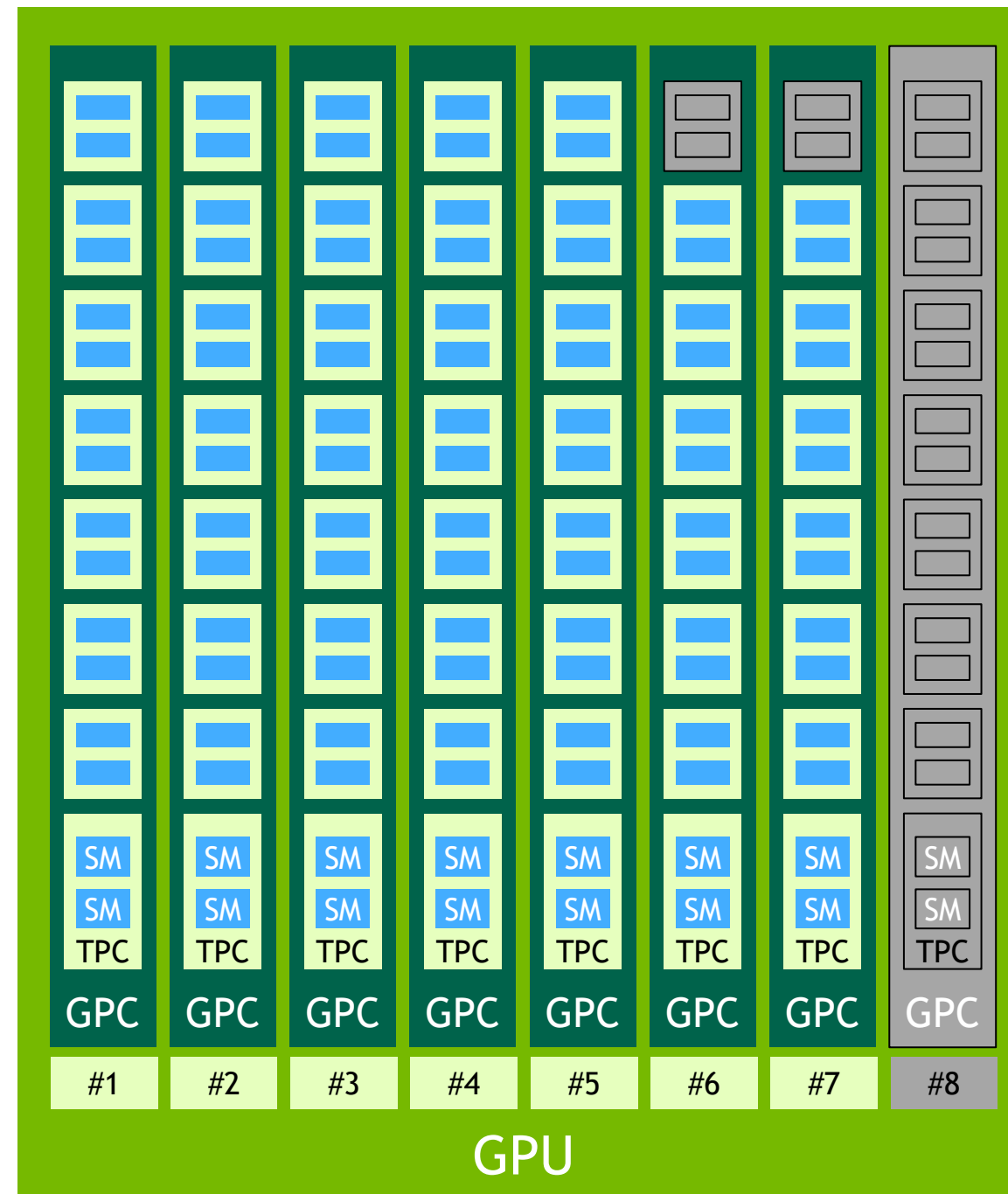
GA100 全体

8 GPC, 8 TPC/GPC, 2 SM/TPC, 128 SM



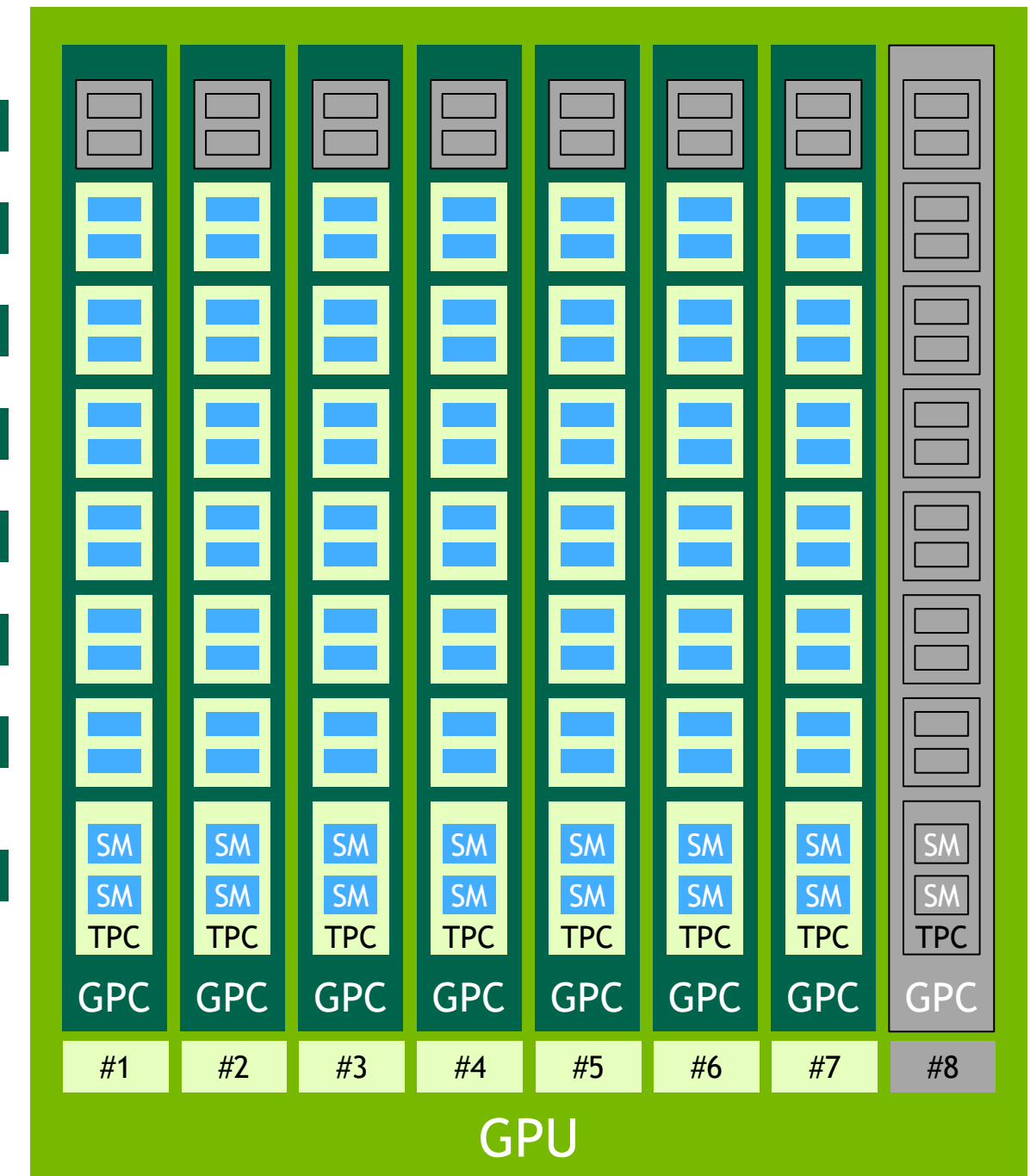
通常の GA100 - MIG 無効

7 GPC, 7 or 8 TPC/GPC, 2 SM/TPC, 108 SM



通常の GA100 - MIG 有効

7 GPC, 7 TPC/GPC, 2 SM/TPC, 98 SM



MIG デバイスの各種 ID

GPU: MIG とは無関係に、A100 GPU 単体を識別する番号。
A100 GPU が 8 基搭載されていれば、この ID は 0 ~ 7

GI ID: GPU インスタンスの ID。GPU 内でユニーク。GI の作成から削除までの間、不変

CI ID: コンピュート インスタンスの ID。GI 内でユニーク。CI の作成から削除までの間、不変

MIG Dev: 「今見えている GPU 内の CI」に連番を振ったもの。GPU 内でユニーク。環境変数 CUDA_VISIBLE_DEVICES 等の設定により、同じ CI に別の番号がつくことがある。

MIG devices:												
GPU	GI ID	CI ID	MIG Dev	Memory-Usage BAR1-Usage	SM	Vol Unc ECC	CE	ENC	DEC	OFA	JPG	
0	2	0	0	14MiB / 20096MiB 0MiB / 32767MiB	56	0	4	0	2	0	0	
0	3	0	1	7MiB / 9984MiB 0MiB / 16383MiB	28	0	2	0	1	0	0	
0	9	0	2	3MiB / 4864MiB 0MiB / 8191MiB	14	0	1	0	0	0	0	

UUID (のようなもの): nvidia-smi -L で確認可能

```
# nvidia-smi -L
GPU 0: A100-SXM4-40GB (UUID: GPU-a4139e7e-4d09-d80b-486e-030e04d3e4b1)
MIG 4g.20gb Device 0: (UUID: MIG-GPU-a4139e7e-4d09-d80b-486e-030e04d3e4b1/2/0)
MIG 2g.10gb Device 1: (UUID: MIG-GPU-a4139e7e-4d09-d80b-486e-030e04d3e4b1/3/0)
MIG 1g.5gb Device 2: (UUID: MIG-GPU-a4139e7e-4d09-d80b-486e-030e04d3e4b1/9/0)
GPU 1: A100-SXM4-40GB (UUID: GPU-731444f0-aa05-0000-a35f-ea9dc09ce349)
GPU 2: A100-SXM4-40GB (UUID: GPU-d55668a0-bd8d-be8e-cf19-14ba757360a5)
GPU 3: A100-SXM4-40GB (UUID: GPU-fea1bc89-8f1d-f399-36ac-2debab0e6a59)
```

MIG デバイスの指定方法

- UUID を指定

```
$ CUDA_VISIBLE_DEVICES=MIG-GPU-391cab19-446b-9b97-5eca-d30416a299de/2/0 BlackScholes  
$ docker run --gpus device=MIG-GPU-a4139e7e-4d09-d80b-486e-030e04d3e4b1/2/0 --rm -it nvcr.io/nvidia/tensorflow:20.12-tf2-py3 nvidia-smi
```

- GPU ID と MIG デバイス番号を指定 (docker コマンド)

```
$ docker run --gpus '"device=0:3,0:4,1:0,1:1"' --rm -it nvcr.io/nvidia/tensorflow:20.12-tf2-py3 nvidia-smi
```

- GPU ID, GI ID, CI ID を指定 (dcgmi コマンド)

```
$ dcgmi dmon -i 0/4/0 -e 100,113,150,155,203,1001,1002,1003,1004,1006,1007,1008
```

#	Entity	SMCLK	SMMAX	TMPTR	POWER	GPCTL	GRCT	SMCT	SMOCC	TENSO	FP64A	FP32A	FP16A
	Id			C	W								
	GPU 0	1410	1410	37	138.545	N/A	0.105	0.103	0.043	0.022	0.000	0.017	0.002
	GPU 0	1410	1410	36	122.252	N/A	0.111	0.107	0.047	0.022	0.000	0.017	0.003

MIG 考慮事項

- GI や CI は揮発性

MIG の GI や CI は稼働中に動的に作成・削除可能

用途に応じてダイナミックに作り替えるものであり、作成した GI や CI は、コンピューターを再起動すると、すべて消滅

- 1 プロセス : 1 インスタンス

現在のところ、MIG 環境の CUDA プログラムで GPU のリストを取得すると、たとえ複数の GPU が存在しても、先頭のデバイスしか取得できない
→ シングルプロセスでマルチ GPU の処理は実行できない。

複数のプロセスを起動し、それぞれに別の MIG デバイスを割り当てることで、複数の MIG デバイスを活用

- GPU 使用率のモニタリング

MIG を有効にした GPU では、nvidia-smi で表示される GPU 使用率が "N/A" となる
DCGM である程度回避可能

```
$ dcgmi dmon -i 0/4/0 -e 100,113,150,155,203,1001,1002,1003,1004,1006,1007,1008
# Entity SMCLK SMMAX TMPTR POWER GPCTL GRACT SMOCC TENS0 FP64A FP32A FP16A
   Id      C      W
GPU 0  1410  1410   37  138.545      N/A  0.105  0.103  0.043  0.022  0.000  0.017  0.002
GPU 0  1410  1410   36  122.252      N/A  0.111  0.107  0.047  0.022  0.000  0.017  0.003
```

MIG とスケジューラー

 Documentation

Search in IBM Spectrum LSF 10.1.0

Q X User

IBM Spectrum LSF

Change version

10.1.0

☐ Show full table of contents

Filter on titles

Administer LSF

Cluster management essentials

Monitoring cluster operations and health

Managing job execution

Configuring and sharing job resources

GPU resources

Enabling GPU features

Automatic GPU configuration

IBM Spectrum LSF / 10.1.0 /

Feedback Product list

Nvidia Multi-Instance GPU (MIG) features

Use the **LSF_MANAGE_MIG** parameter in the `lsf.conf` file to enable dynamic MIG scheduling.

Nvidia Multi-Instance GPU (MIG) features allow a single supported GPU to be securely partitioned into up to seven independent GPU instances, providing multiple users with independent GPU resources.

When dynamic MIG scheduling is enabled, LSF dynamically creates GPU instances (GI) and compute instances (CI) on each host, and LSF controls the MIG of each host. If you enable dynamic MIG scheduling, do not manually create or destroy MIG devices outside of LSF. Set the **LSF_MANAGE_MIG** parameter to Y in the `lsf.conf` file to enable dynamic MIG scheduling.

If **LSF_MANAGE_MIG** is set to N or is undefined, LSF uses static MIG scheduling. LSF allocates the GI and CI based on the configuration of each MIG host, and dispatches jobs to the MIG hosts. LSF does not create or destroy the GI and CI on the MIG hosts. If you use static MIG scheduling and want to change MIG devices, you must wait for the running MIG job to finish, then destroy the existing MIG device, create a new MIG device, and restart the LSF daemons.

If you change the value of this parameter, you must wait until all MIG jobs that are running on a cluster are done, then restart the LSF daemons for your changes take effect.

After changing the value of this parameter, you must restart the LSF daemons for your changes to take effect.

Parent topic:

→ [Enabling GPU features](#)

CUDA 並列実行機構

CUDA ストリーム、MPS、MIG

	並列実行	アドレス空間の 分離	演算性能の分離	メモリ性能の分離	障害の分離
CUDA ストリーム	Yes	No	No	No	No
MPS	Yes	Yes	Yes (by percentage, not partitioning)	No	No
MIG: 同一 GI 内の 複数 CI	Yes	Yes	Yes	No	Yes
MIG: GI (GPU インスタンス)	Yes	Yes	Yes	Yes	Yes

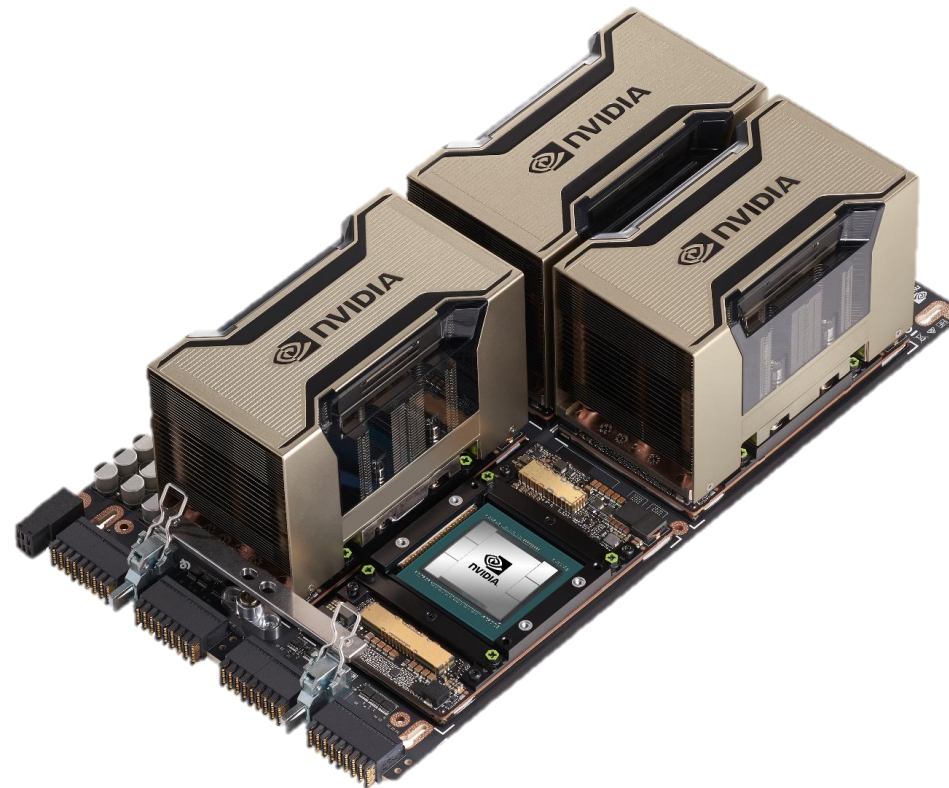
A100 提供形態

NVIDIA A100 PCIe



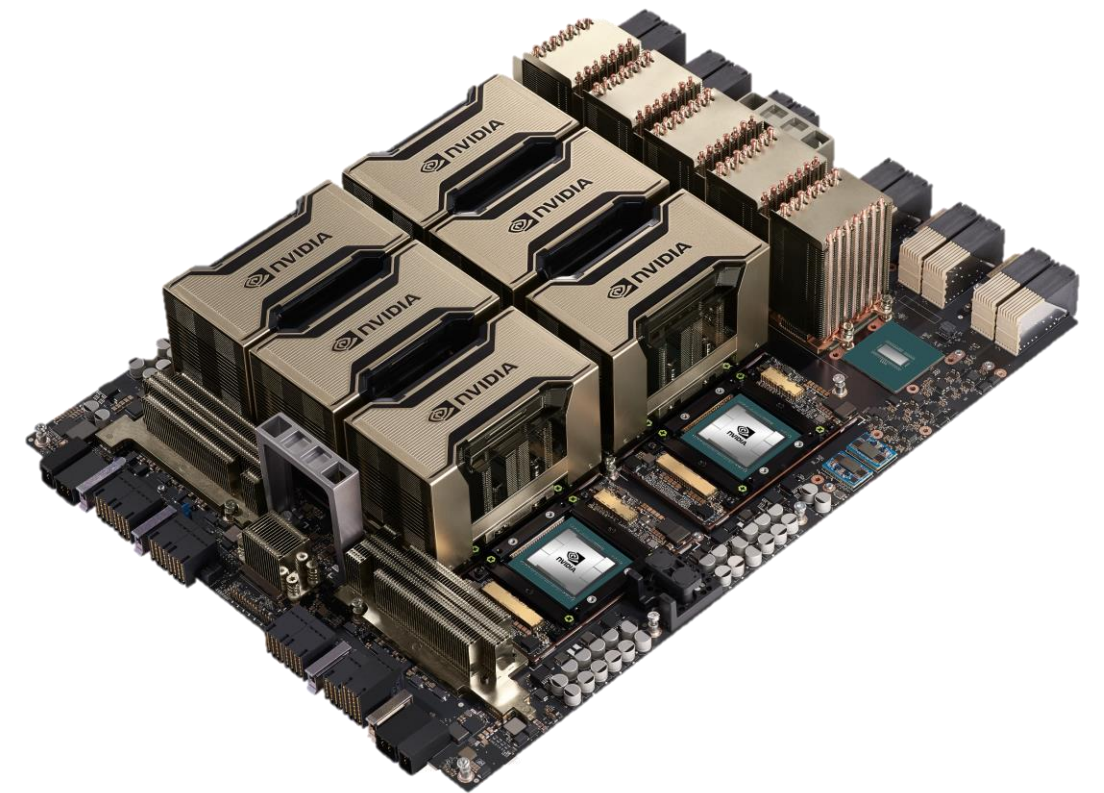
- SXM4 版と同じピーク性能
(実アプリ性能は 10% 程度ダウン)
- TDP: 250W

NVIDIA HGX A100 4-GPU



- 4 基の NVIDIA A100 SXM4
- さまざまなワークロードに対応

NVIDIA HGX A100 8-GPU

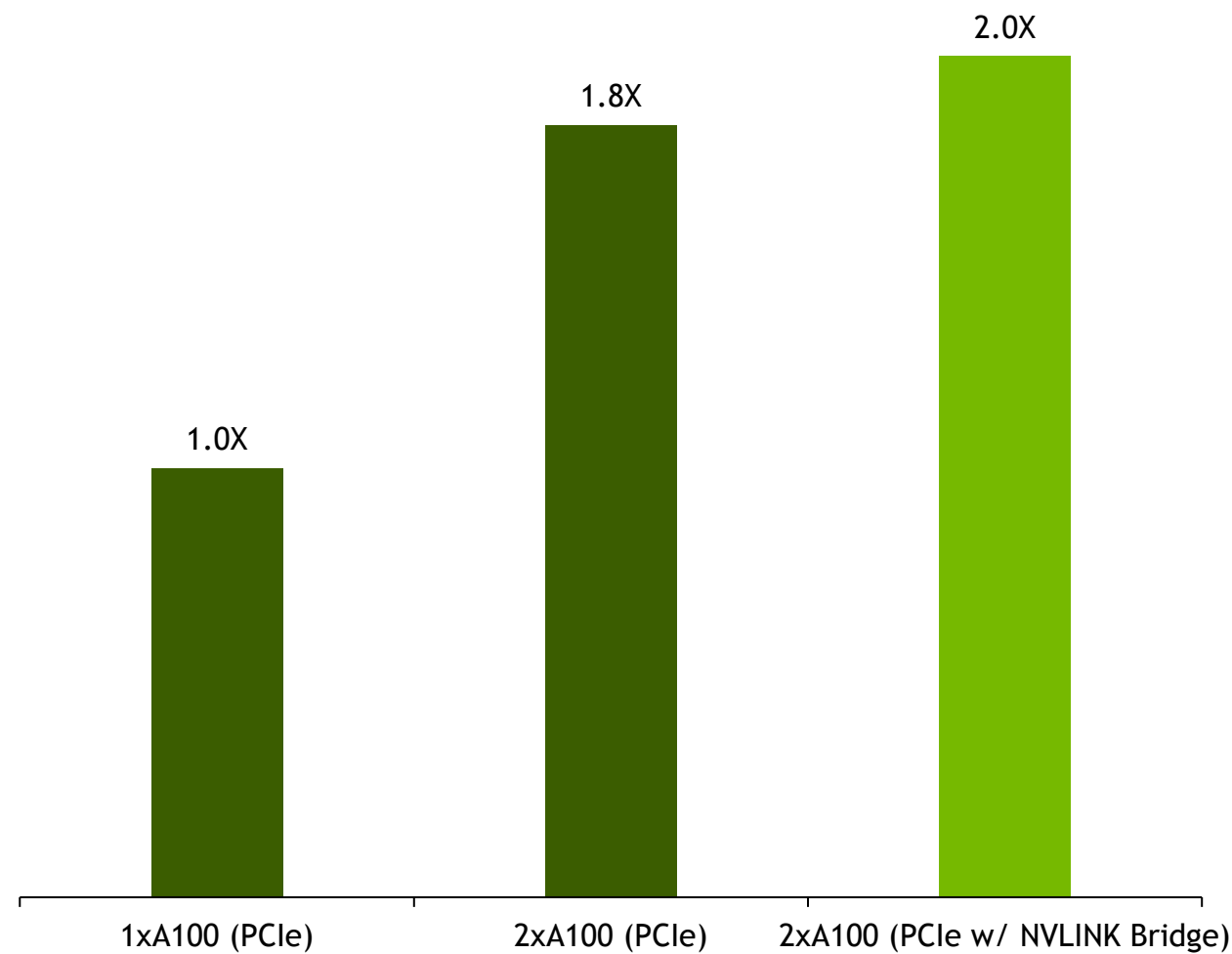


- 8 基の NVIDIA A100 SXM4
- 6 基の NVIDIA NVSwitch
- ハイエンド機向け

HGX のスケールアップ性能

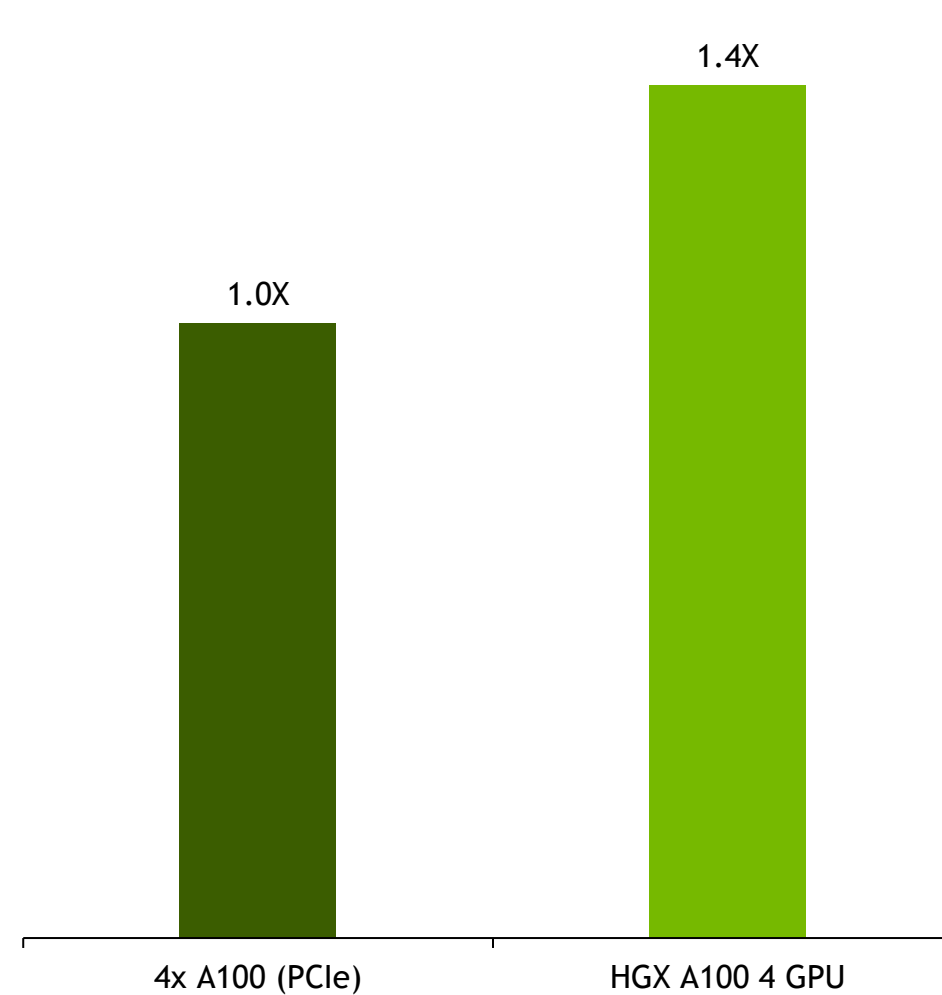
マルチ GPU アプリケーションで最大 1.5 倍の性能アップを実現

BERT-LARGE Pre-Training 2x GPU



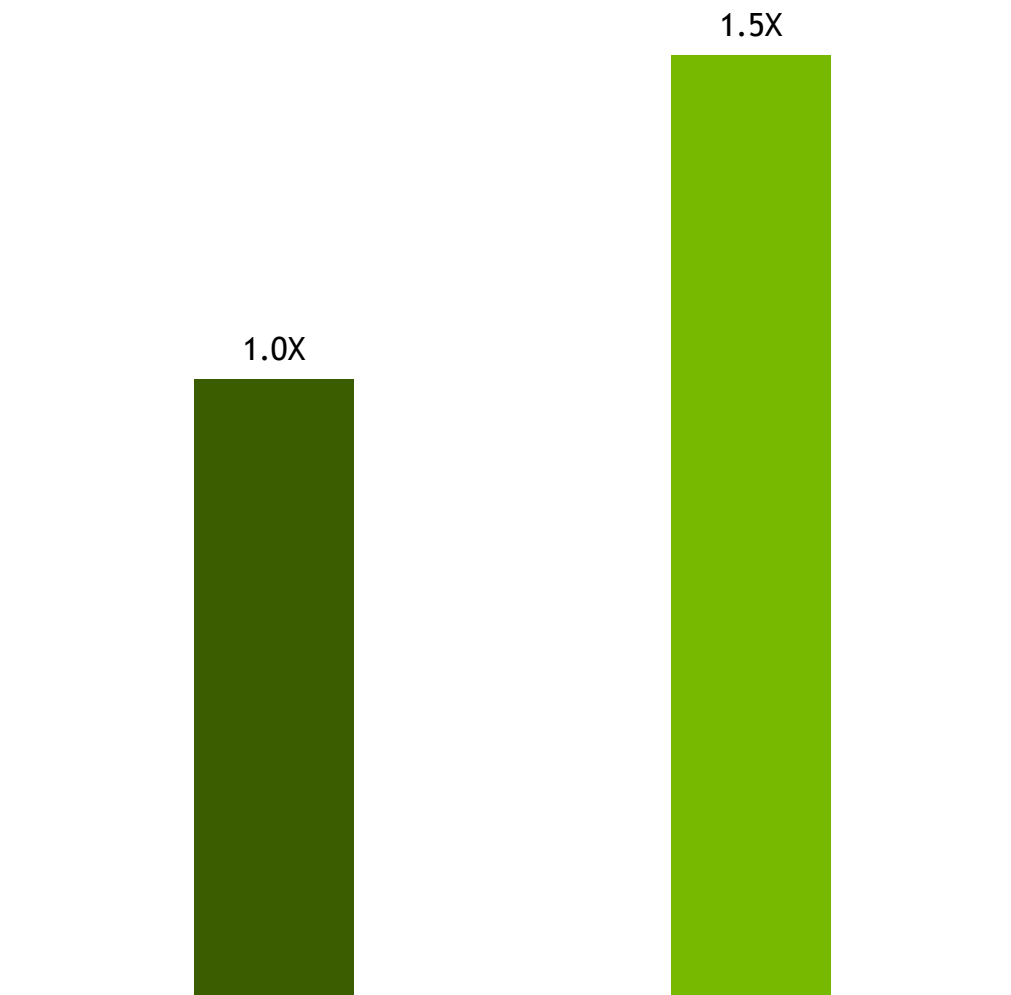
NVLINK Bridge improves 2 GPU scaling by 1.1X

BERT-LARGE Pre-Training 4x GPU

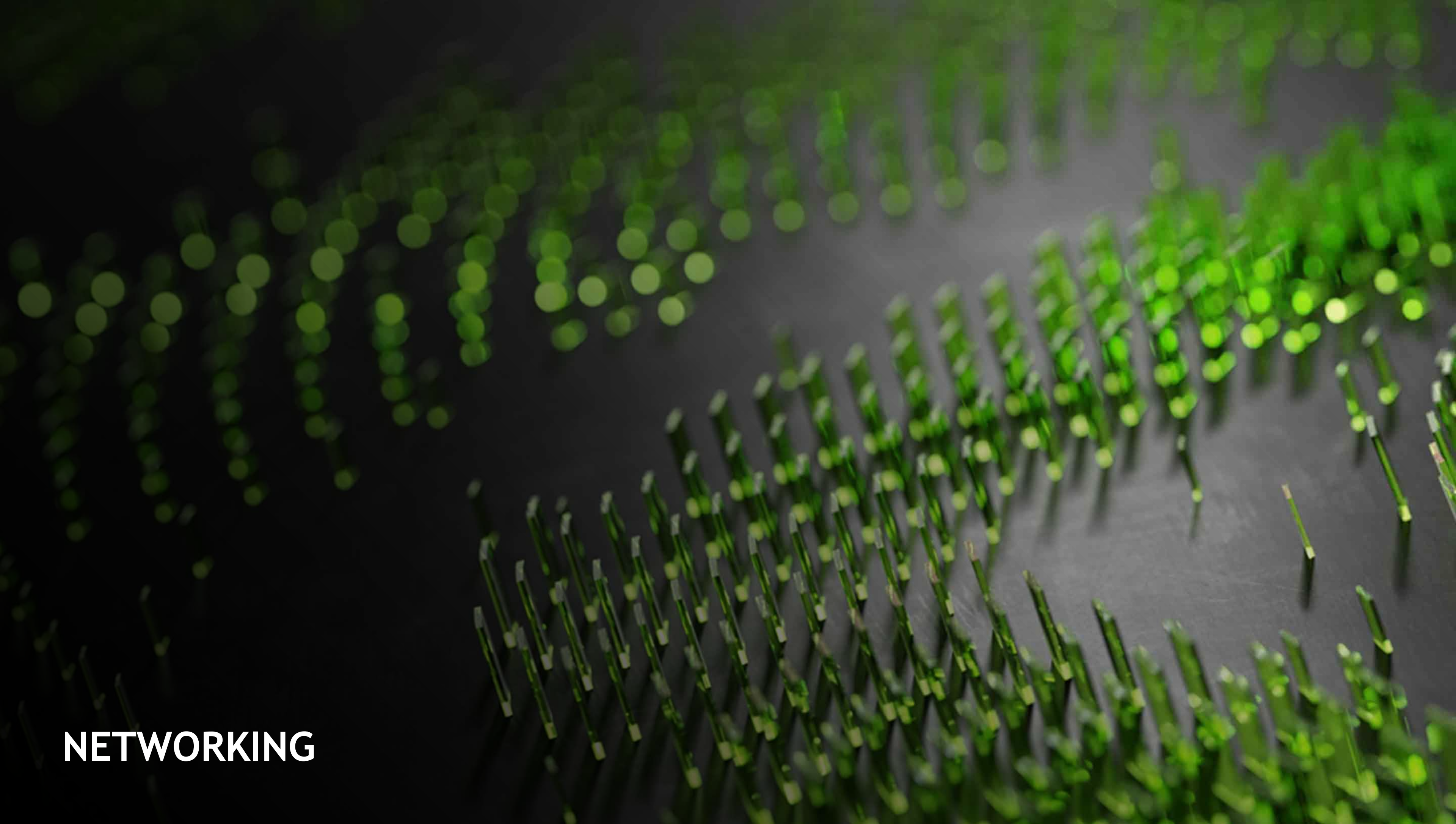


HGX A100 4 GPU improves scaling by 1.4X

BERT-LARGE Pre-Training 8x GPU



HGX A100 8 GPU improves scaling by 1.5X



NETWORKING

IN-NETWORK がスーパーコンピューティングを飛躍的に加速

ソフトウェア デファインド、ハードウェア アクセラレーション、InfiniBand ネットワーク

AI & ML

GPU

DPU

Data Processing Unit

アクセラレーテッド コンピューティング

GPU で加速された AI
& 機械学習
すべてのAIワークロードが
加速

ソフトウェア デファインド、
ハードウェア アクセラレーション

DPUはデータ集約型
タスクを加速
ネットワーク、セキュリティ、ストレージ

CPU

NVIDIA BLUEFIELD-2 DATA PROCESSING UNIT

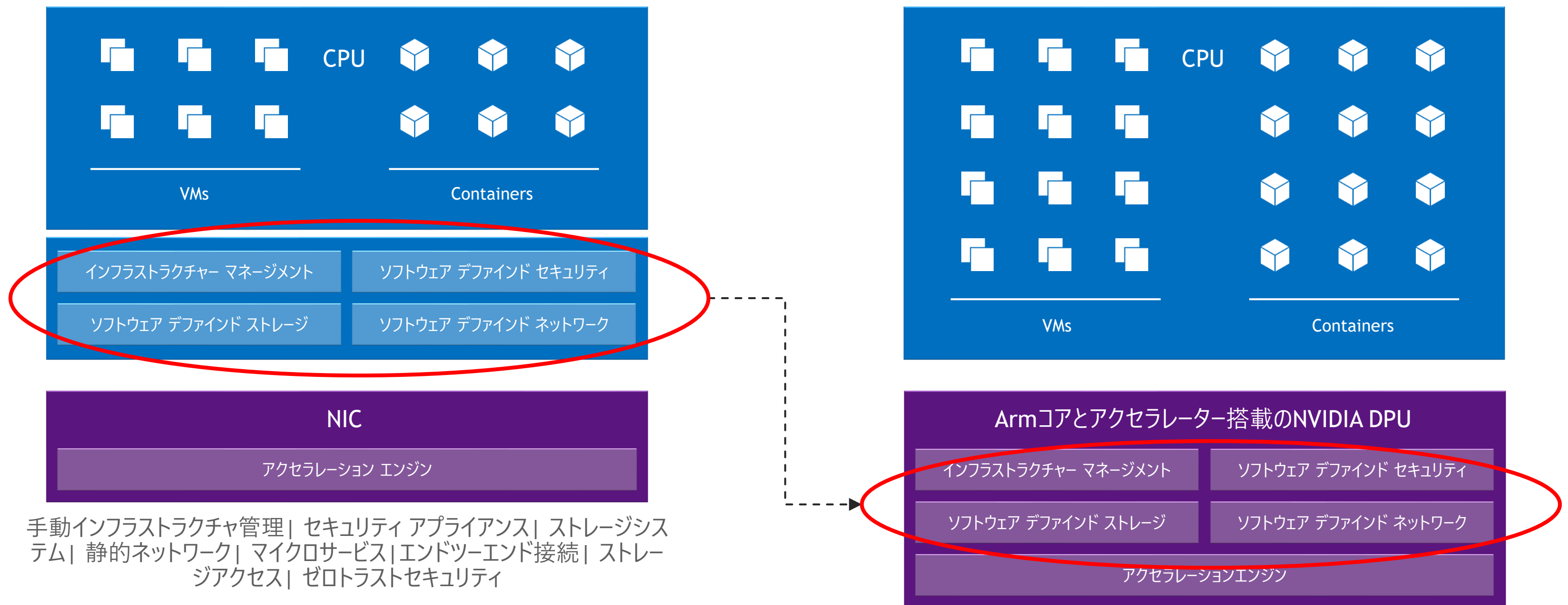
データセンター インフラストラクチャをチップにオフロード

- 最大 200Gb/s Ethernet and InfiniBand, PAM4/NRZ
- ConnectX-6 Dx を内蔵
- 8 Arm A72 CPUs subsystem - up to 2.75GHz
 - 8MB L2 cache, 6MB L3 cache in 4 Tiles
 - 非常に高速で低遅延なインターコネクト
- 外部、内部を統合するPCIe switch, 16x Gen4.0
 - PCIeルートコンプレックスモードまたはエンドポイントモード
- Single DDR4 メモリチャネル
- 1GbE 外部管理ポート
- セキュリティ、ストレージ、ネットワークを加速



DATA PROCESSING UNIT

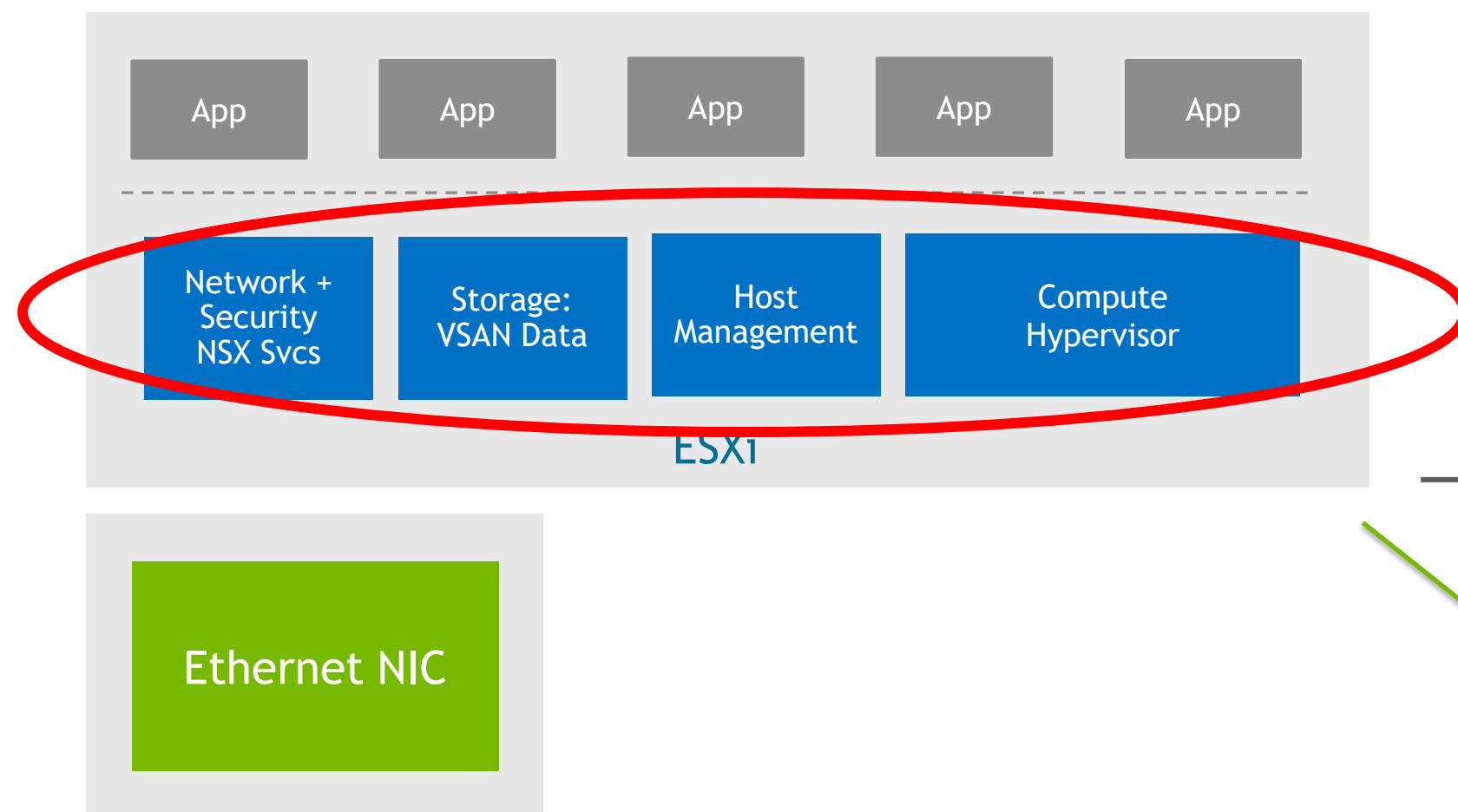
データセンター インフラストラクチャをソフトウェア デファインド、ハードウェア アクセラレーションでチップにオフロード



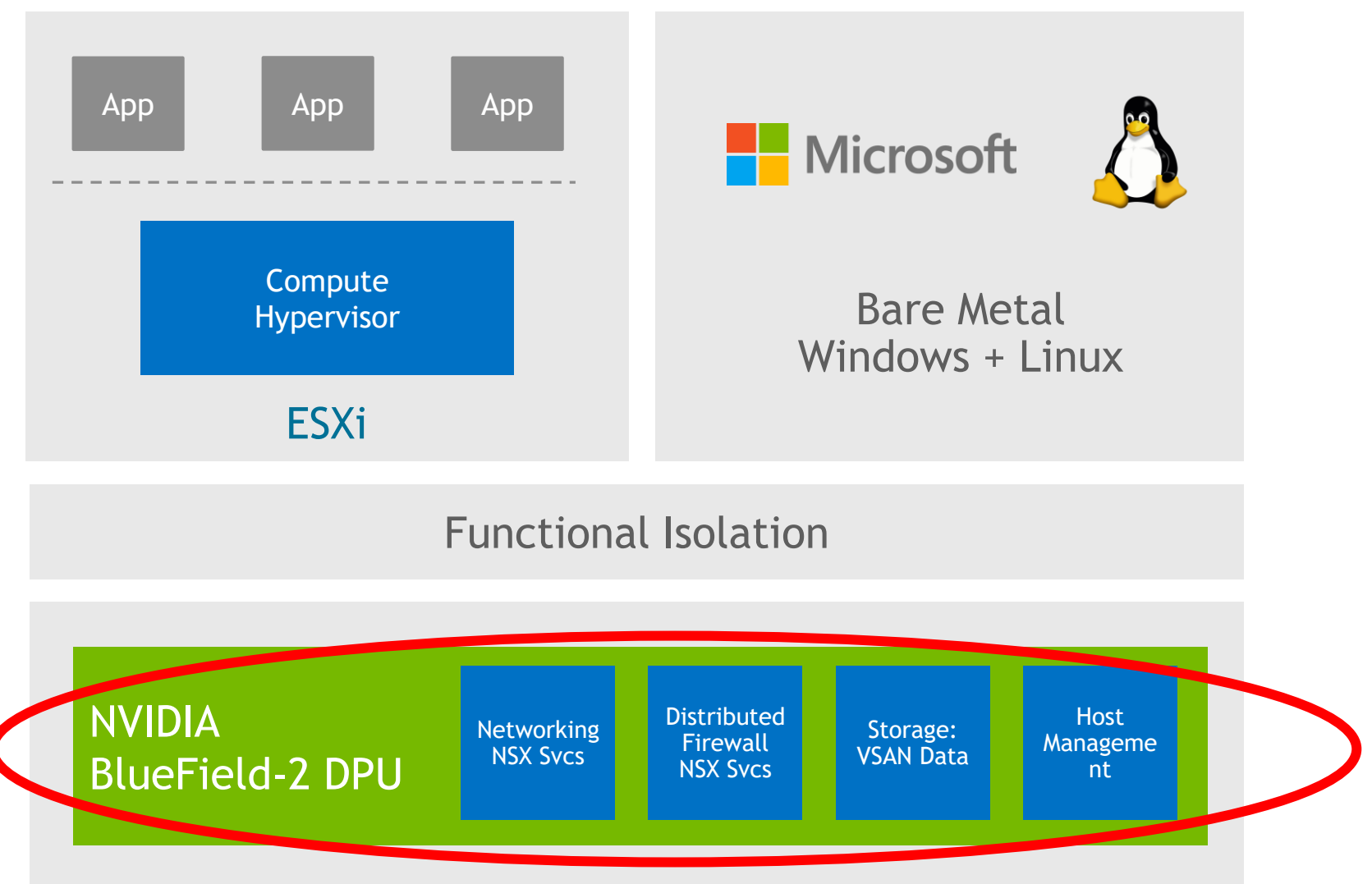
PROJECT MONTEREY

DPU 側に VMware ESXi のインフラストラクチャー処理をオフロードしビジネスのワークロードを効率的に実行

現状の環境

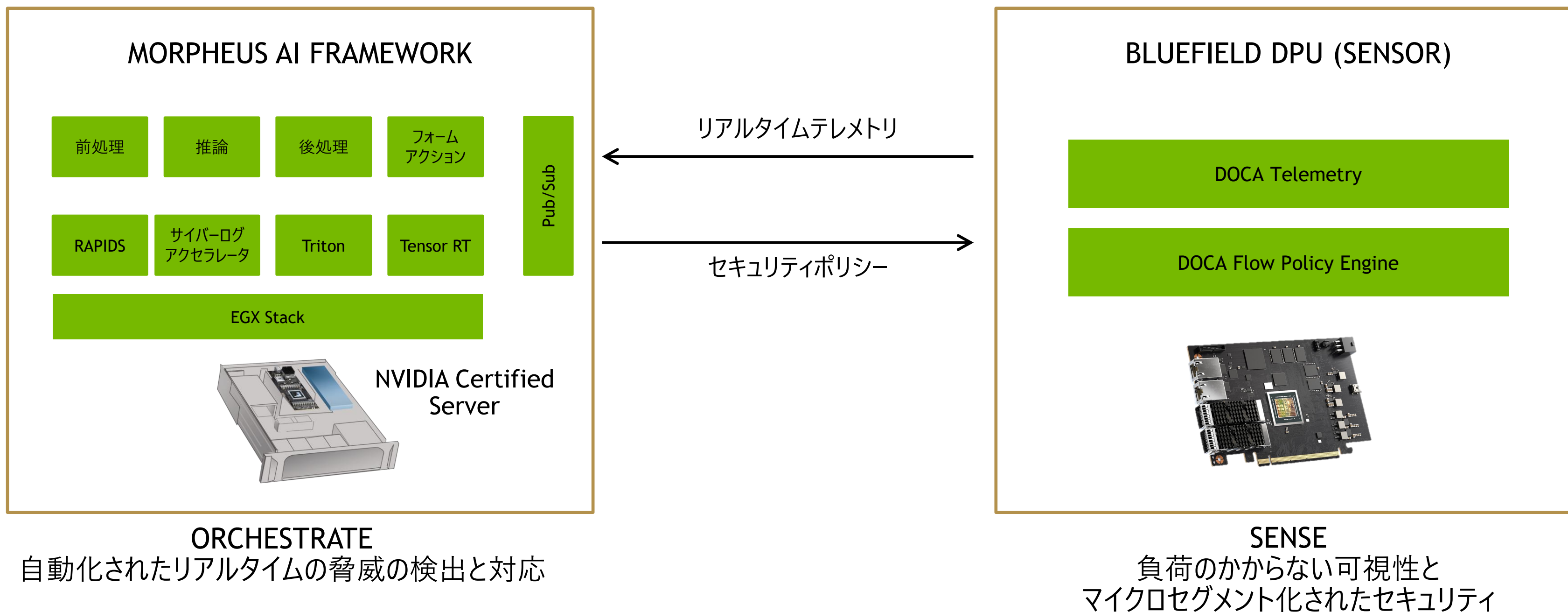


Project Monterey



MORPHEUS AI CYBER SECURITY フレームワーク

サイバーエコシステムを加速するオープンAIコンピューティング



TLS のサポート

200Gb/s での暗号化・複合化

暗号化/複合化/パケットの認証

TLS1.2 と TLS1.3 をサポート

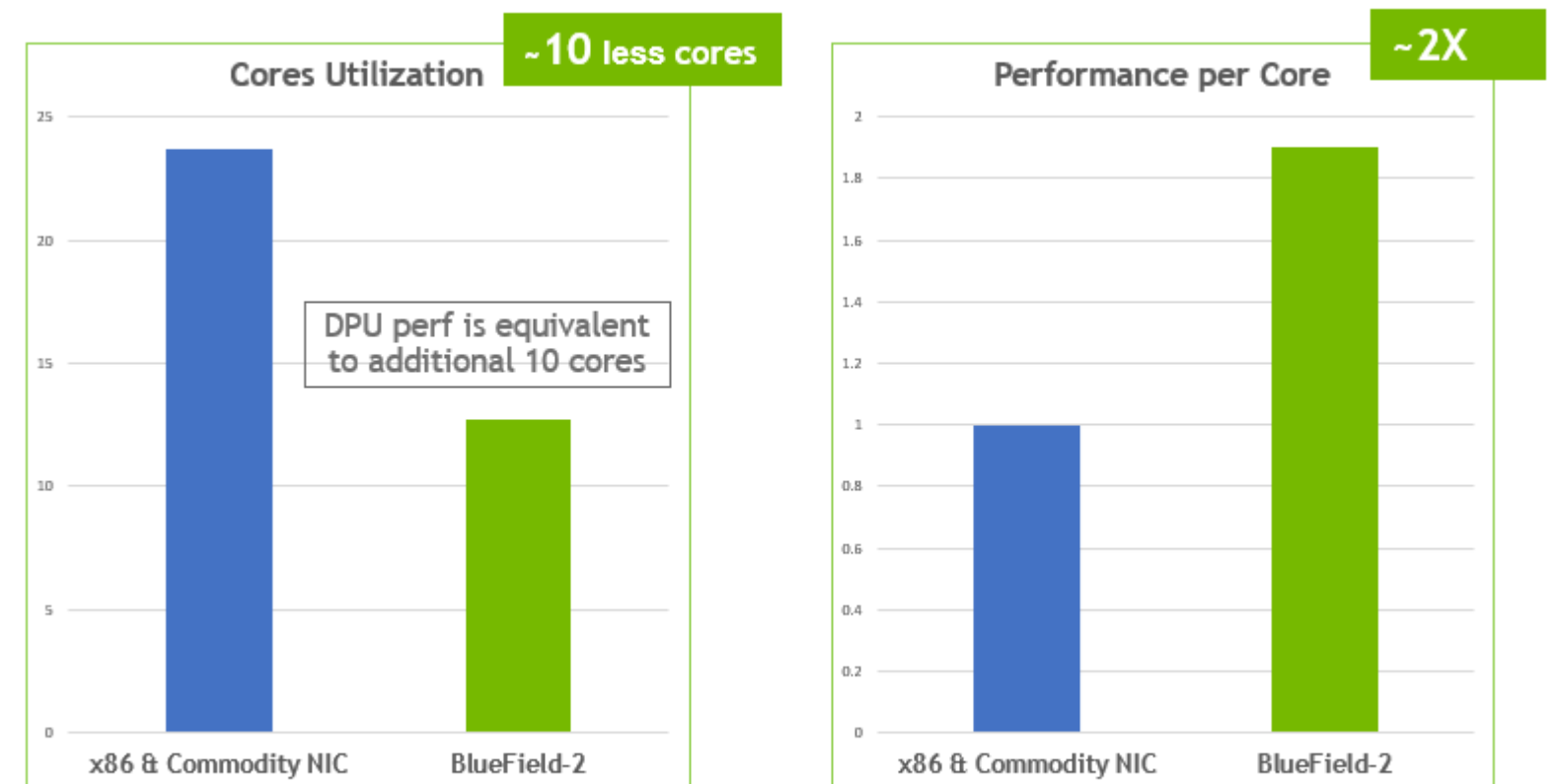
Cipher: AES-GCM 128/256bit keys

ホスト毎に 400万コネクションまでサポート

鍵の管理

鍵は暗号化され DPU のキャッシュに格納

鍵の生成と署名に TRNG (真性乱数発生器) と PKI エンジンを使用



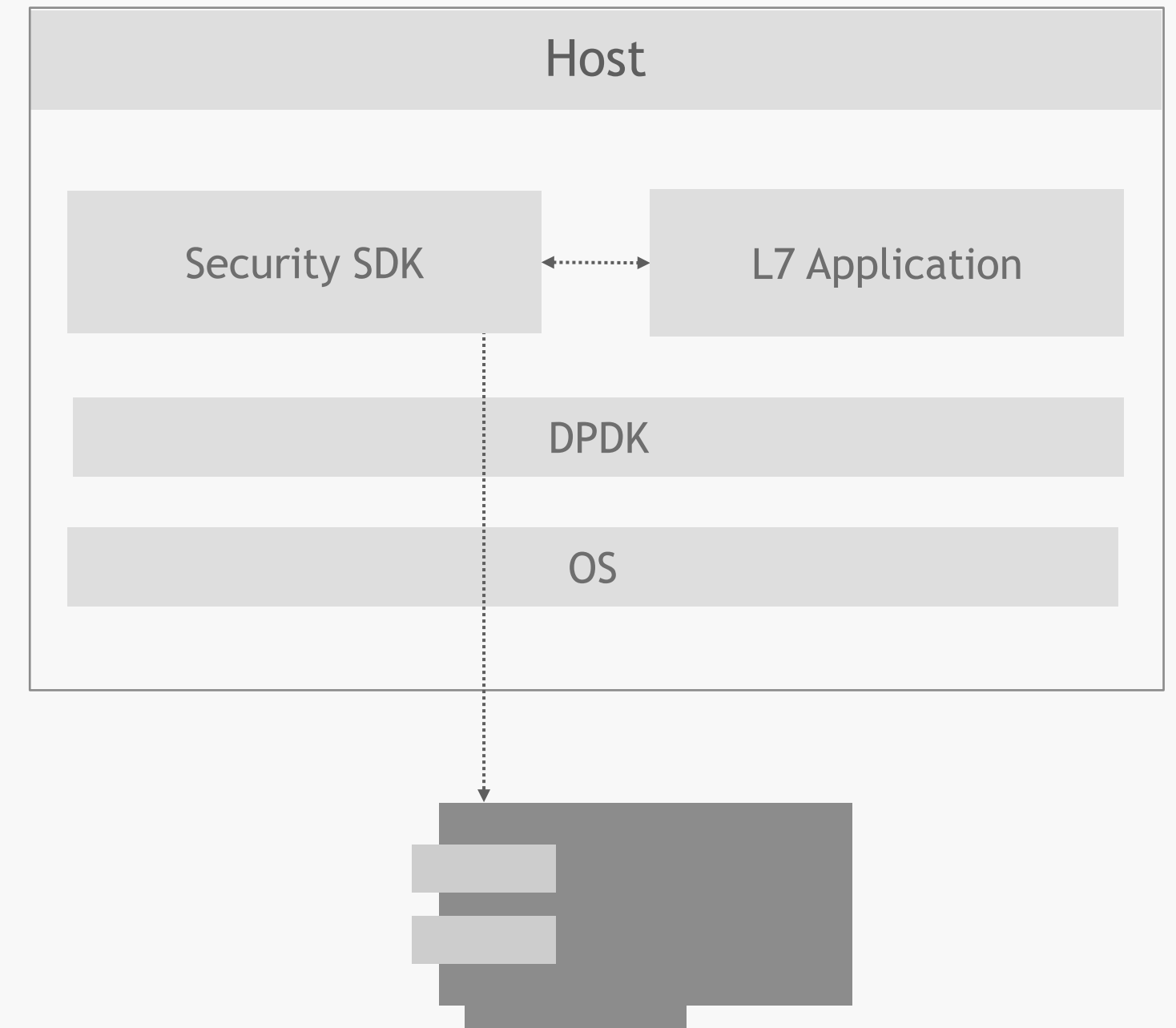
BLUEFIELD-2 での正規表現高速化

トラフィックを分析し悪意のあるコンテンツを検出

- Total inspection bandwidth of 50Gbps
- 100K signatures are supported
- 256 jobs in the air, 32 can be processed in parallel

BlueField-2 の正規表現エンジンが実現すること

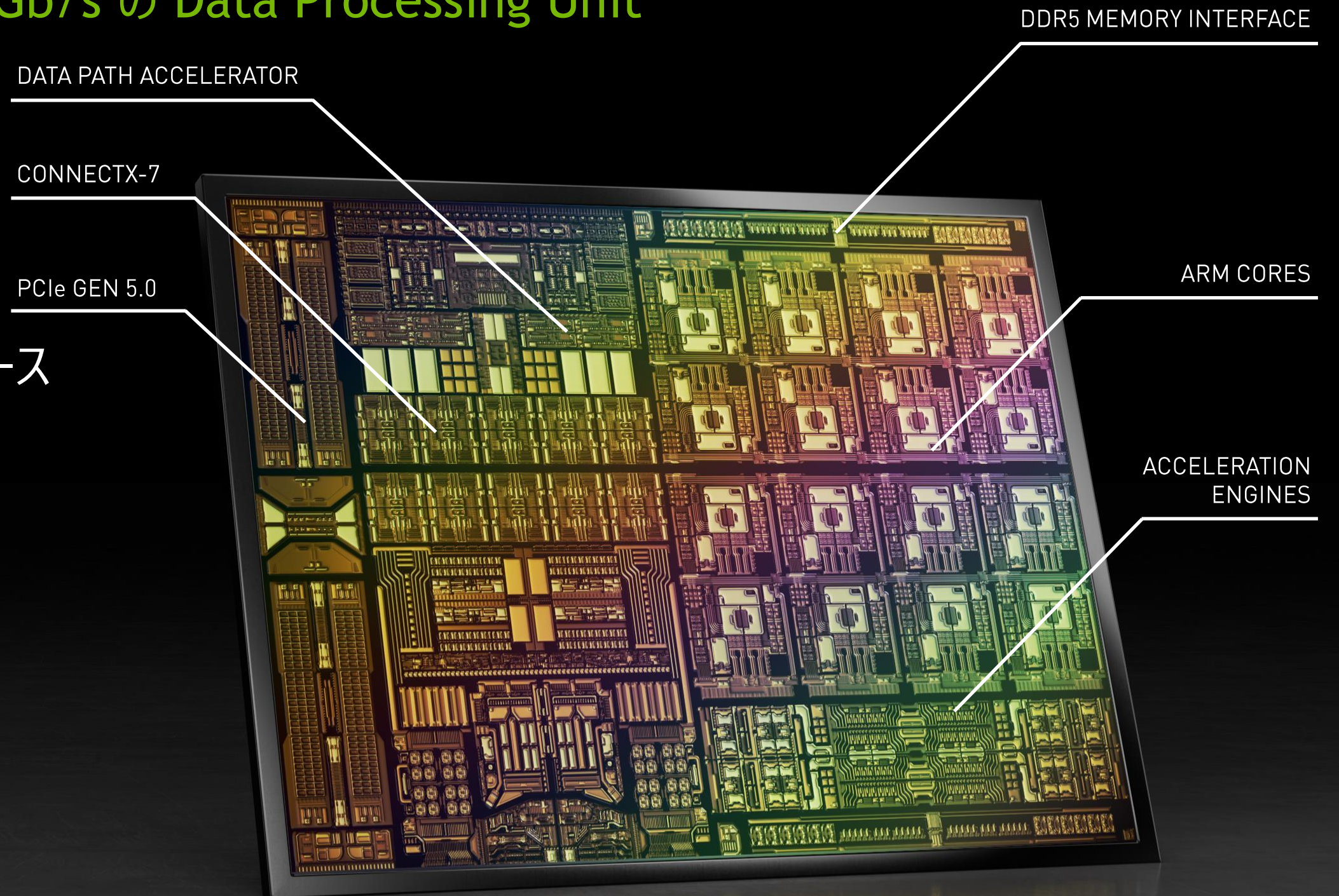
- 侵入検知 (e.g., Suricata)
- ステートフルな L7 ロードバランサ
- L7 ベースの DDOS 検知
- 次世代ファイアウォールの高速化
- 5G vEPC
- Web Application Firewall (WAF)



NVIDIA BLUEFIELD-3 DPU

最速 400Gb/s の Data Processing Unit

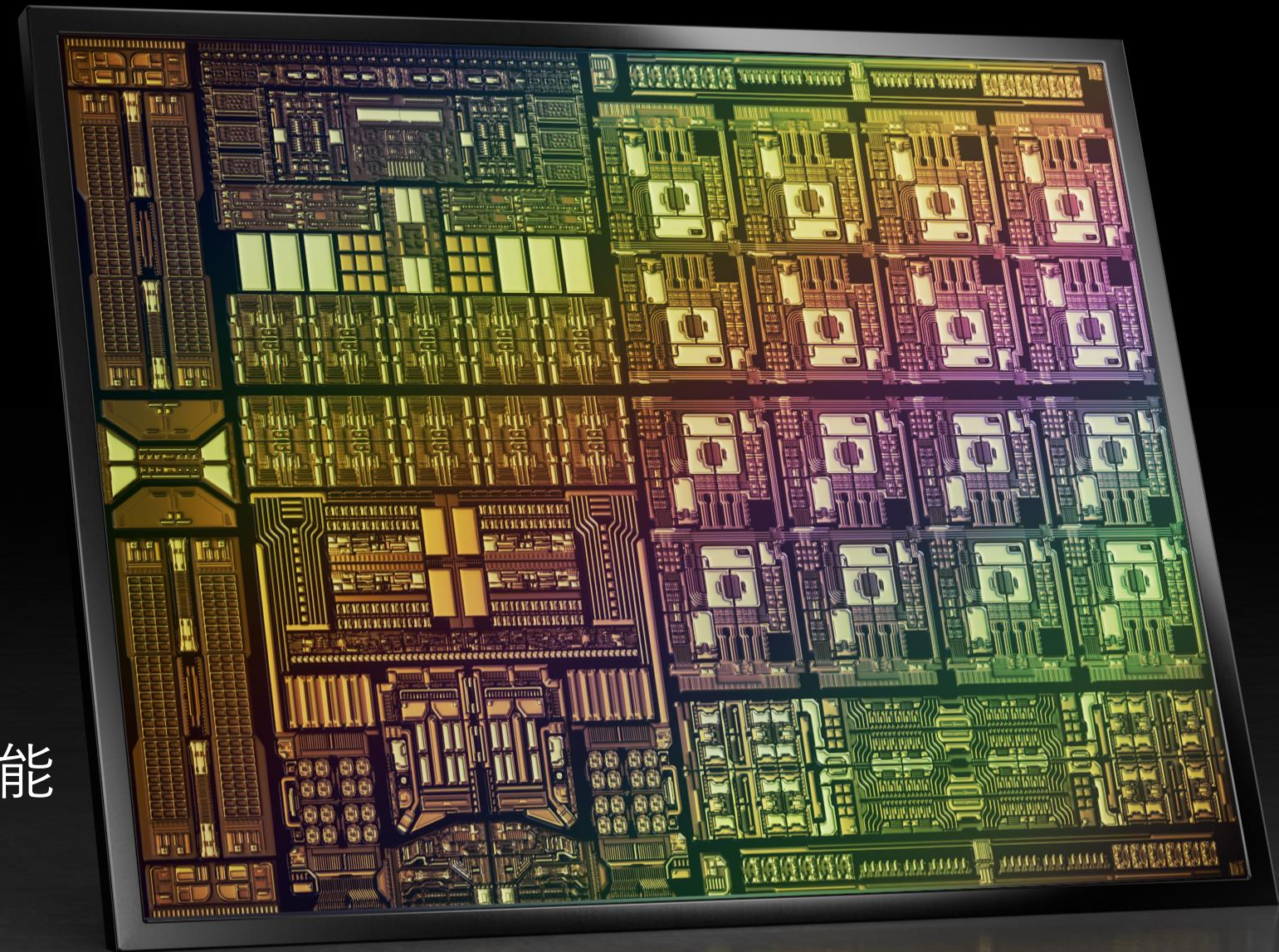
- 220億トランジスタ
- 400Gb/s のEthernet版 InfiniBand版をリリース
- 400Gb/s 暗号化アクセラレーション
- X86 Core 300個と同等
- 18M IOP/s ストレージ性能
- DDR5 Memory



NVIDIA BLUEFIELD-3 DPU

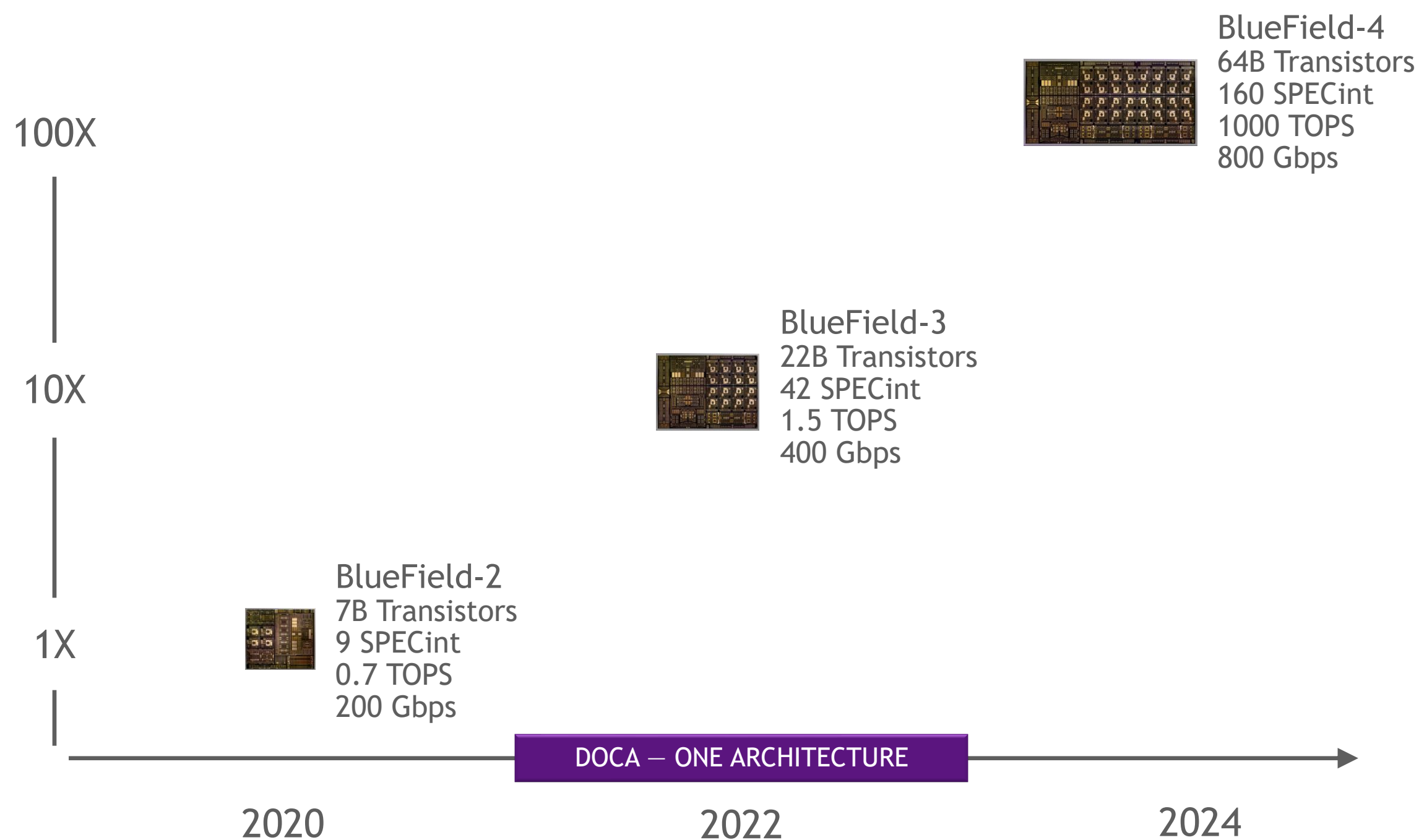
最速 400Gb/s の Data Processing Unit

- データセンター インフラストラクチャーのオフロードと高速化
- ホストのアプリケーションワークロードから
管理機能、制御機能を完全分離
- 強力な CPU - 16x Arm A78 Coresを搭載
- データパスアクセラレータ - 16x Cores, 256 Threads
- ネットワーク, ストレージ, セキュリティを400 Gb/sで処理可能



NVIDIA DPU ロードマップ

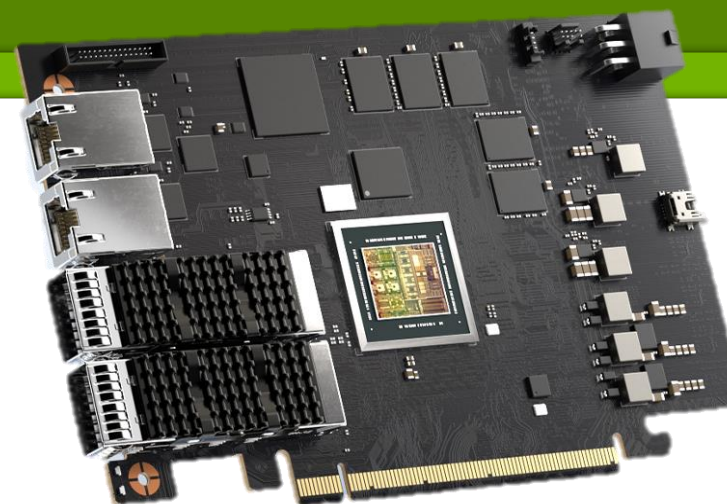
データセンターインフラ処理の飛躍的な成長



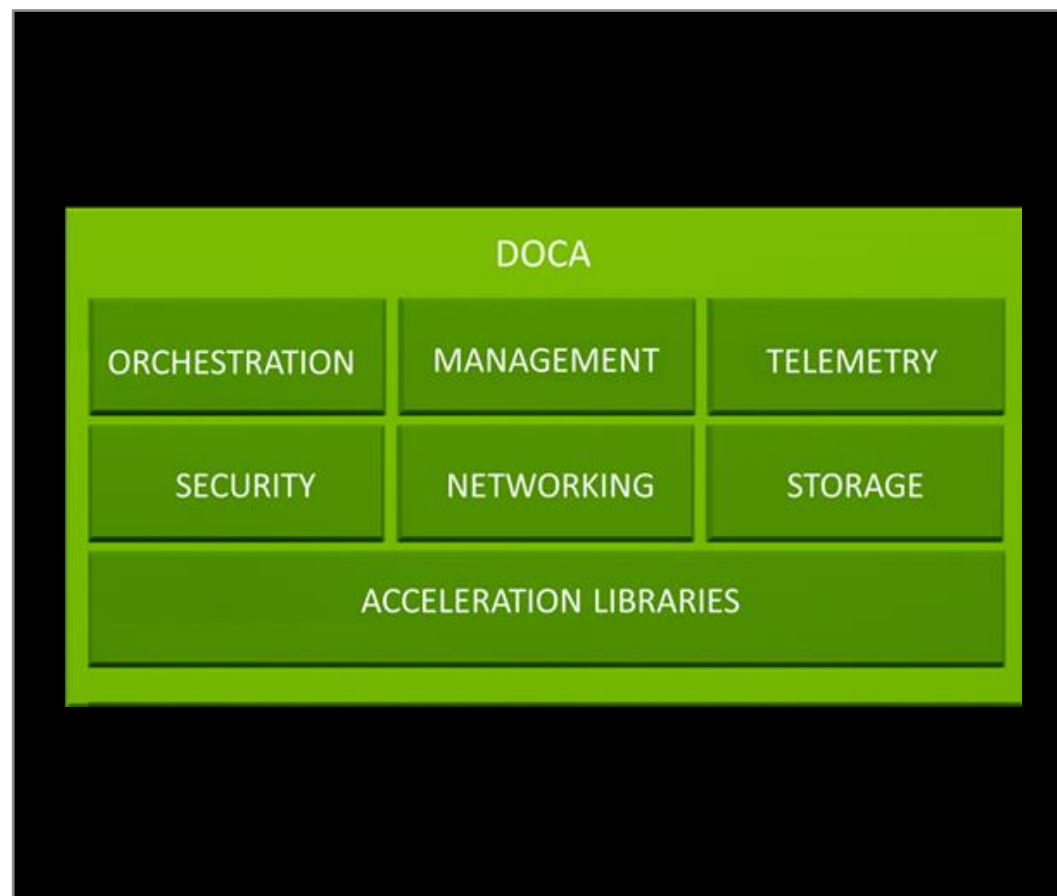
NVIDIA DOCA

広範な DPU パートナーエコシステムの実現

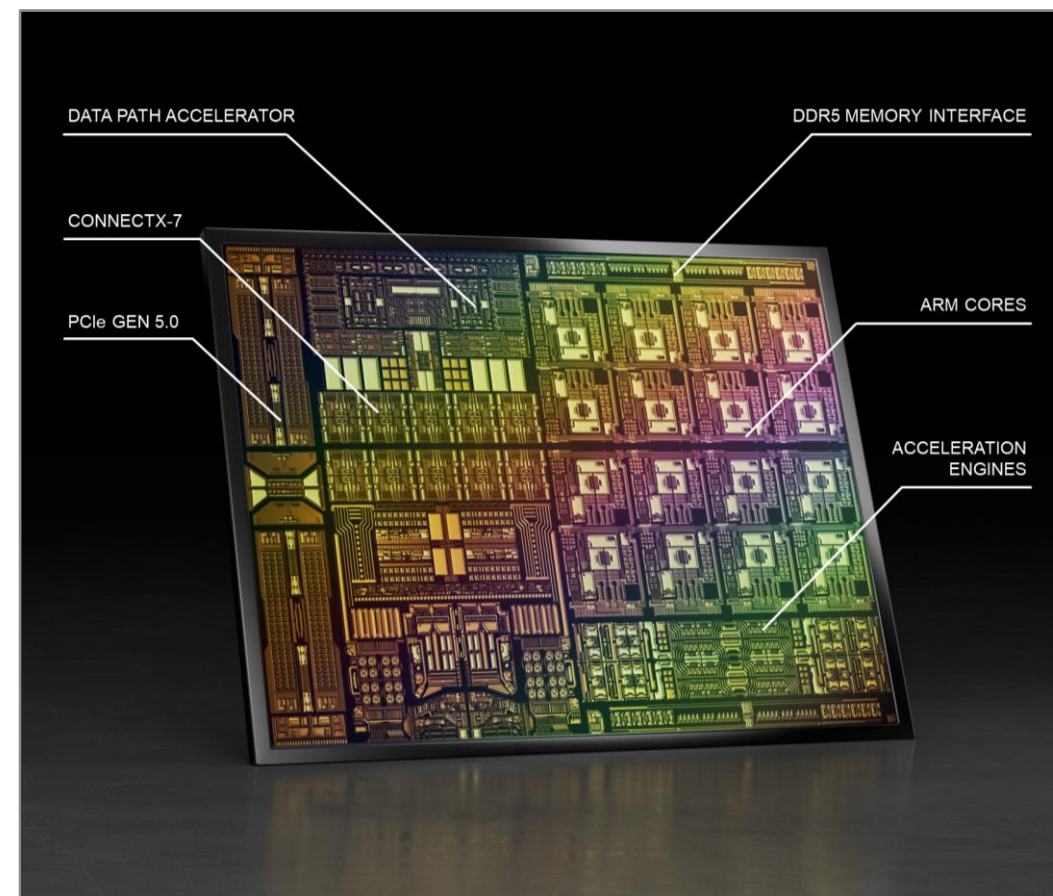
- BlueField DPU のソフトウェア アプリケーション フレームワーク
- GPU には CUDA、DPU には DOCA
- 開発したソフト、習得技術は将来の DPU 上でも動作が可能
- 認定リファレンス アプリケーション, API & パートナー ソリューション
- 業界およびワークロード全体にわたる豊富なパートナー エコシステム



これからの HPC と AI を支えるネットワーク技術



DOCA のパートナーエコシステム



次世代の 400Gbps DPU - Bluefield-3



NVIDIA Quantum-2
NDR 400G InfiniBand

CPU/GPU/DPU のロードマップ

