NVIDIA コンピューティング プラットフォーム

2022/02/10
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

GPU

データセンター向け GPU 製品

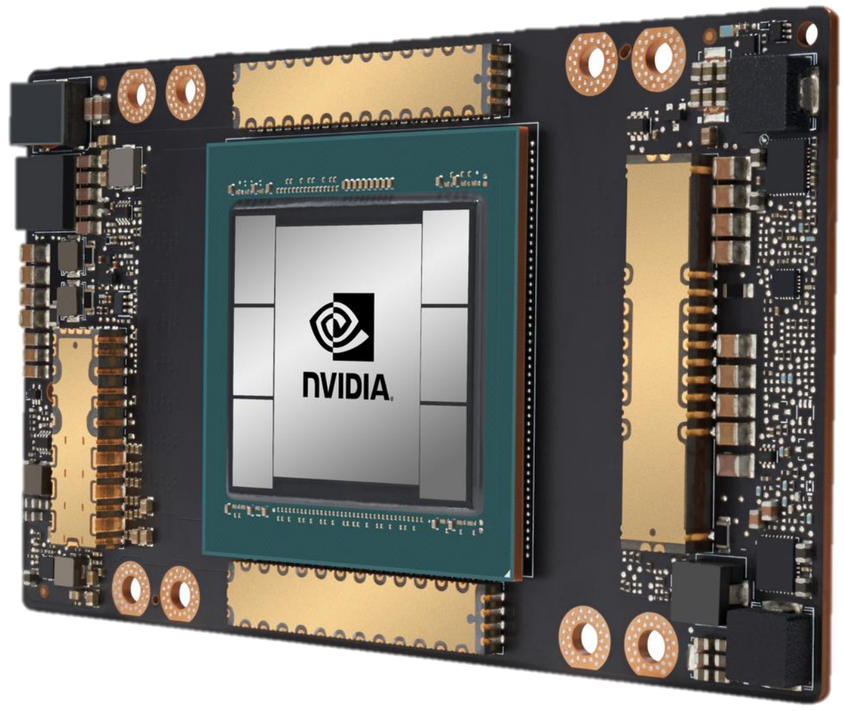
	A100	A30	A2	A40	A16	A100X	A30X
Design	Highest Perf Compute	Mainstream Compute	Entry-Level Small Footprint	High Perf Graphics	High Density Virtual Desktop	High Perf Converged Accelerator	Mainstream Converged Accelerator
Max Power	300W	165W	40-60W	300W	250W	300W	230W
Form Factor	x16 PCIe Gen4 2 Slot FHFL 3 NVLink Bridge	x16 PCIe Gen4 2 Slot FHFL 1 NVLink Bridge	x8 PCIe Gen4 1 Slot LP	x16 PCIe Gen4 2 Slot FHFL 1 NVLink Bridge	x16 PCIe Gen4 2 Slot FHFL	x16 PCIe Gen4 2 Slot FHFL 3 NVLink Bridge	x16 PCIe Gen4 2 Slot FHFL 1 NVLink Bridge
GPU Memory	80GB HBM2e	24GB HBM2	16GB GDDR6	48GB GDDR6	4x 16GB GDDR6	80GB HBM2e	24GB HBM2e
Multi-Instance GPU (MIG)	Up to 7	Up to 4	-	-	-	Up to 7	Up to 4
Media Acceleration	1 JPEG Decoder 5 Video Decoder	1 JPEG Decoder 4 Video Decoder	1 Video Encoder 2 Video Decoder (+AV decode)	1 Video Encoder 2 Video Decoder (+AV1 decode)	4 Video Encoder 8 Video Decoder (+AV1 decode)	1 JPEG Decoder 5 Video Decoder	1 JPEG Decoder 4 Video Decoder
Ray Tracing	-		Yes	Yes	Yes	-	
Fast FP64	Yes		-	-	-	Yes	
Graphics	For in-situ visualization (no NVIDIA vPC or RTX vWS)		Good	Best	Better	For in-situ visualization (no NVIDIA vPC or RTX vWS)	
vGPU	Yes		Yes*	Yes	Yes	Yes*	
Hardware Root of Trust	Yes		Yes	Yes	Yes	Yes	
Integrated DPU	-	-	-	-	-	BlueField-2	

NVIDIA

NVIDIA A100

かつてない飛躍 - Volta 比最大 20 倍のピーク性能

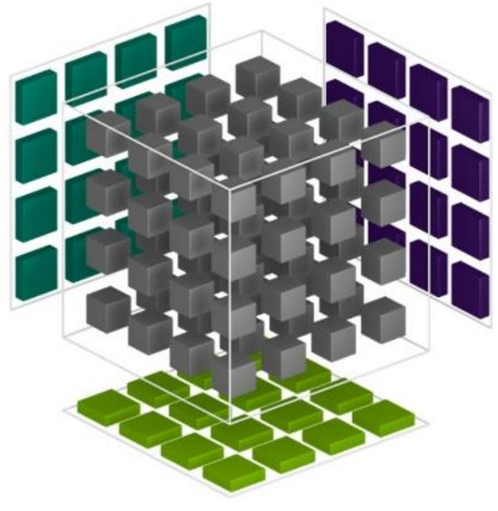
	ピーク性能	V100 比
FP32 トレーニング	312 TFLOPS	**20X**
INT8 インファレンス	1,248 TOPS	**20X**
FP64 HPC	19.5 TFLOPS	**2.5X**
Multi-instance GPU (MIG)		**7X GPUs**



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

Tensor コア

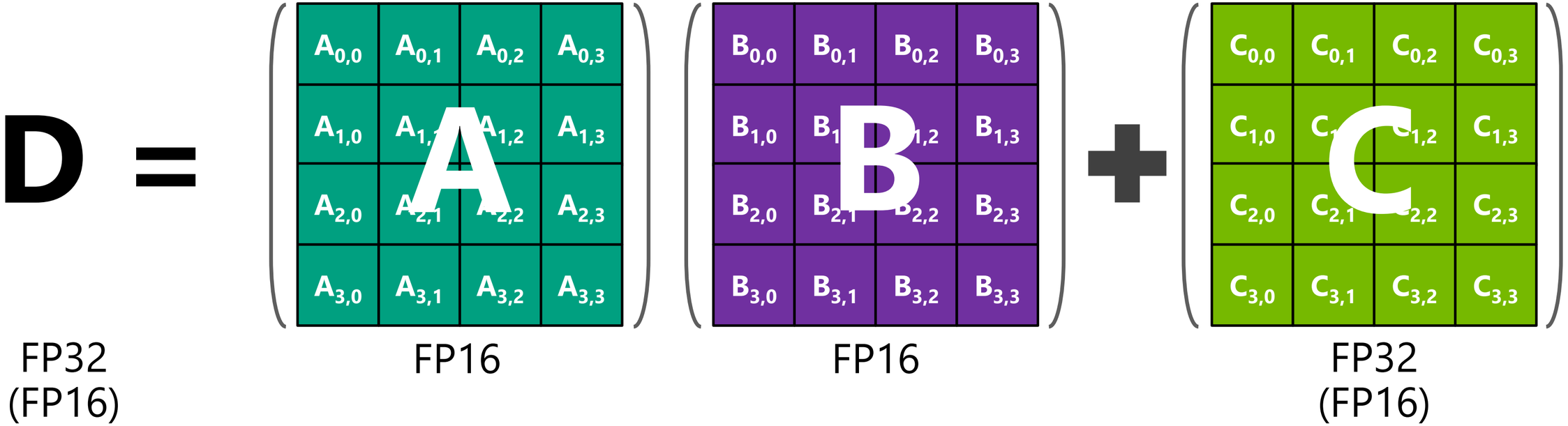
行列演算ユニット



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



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

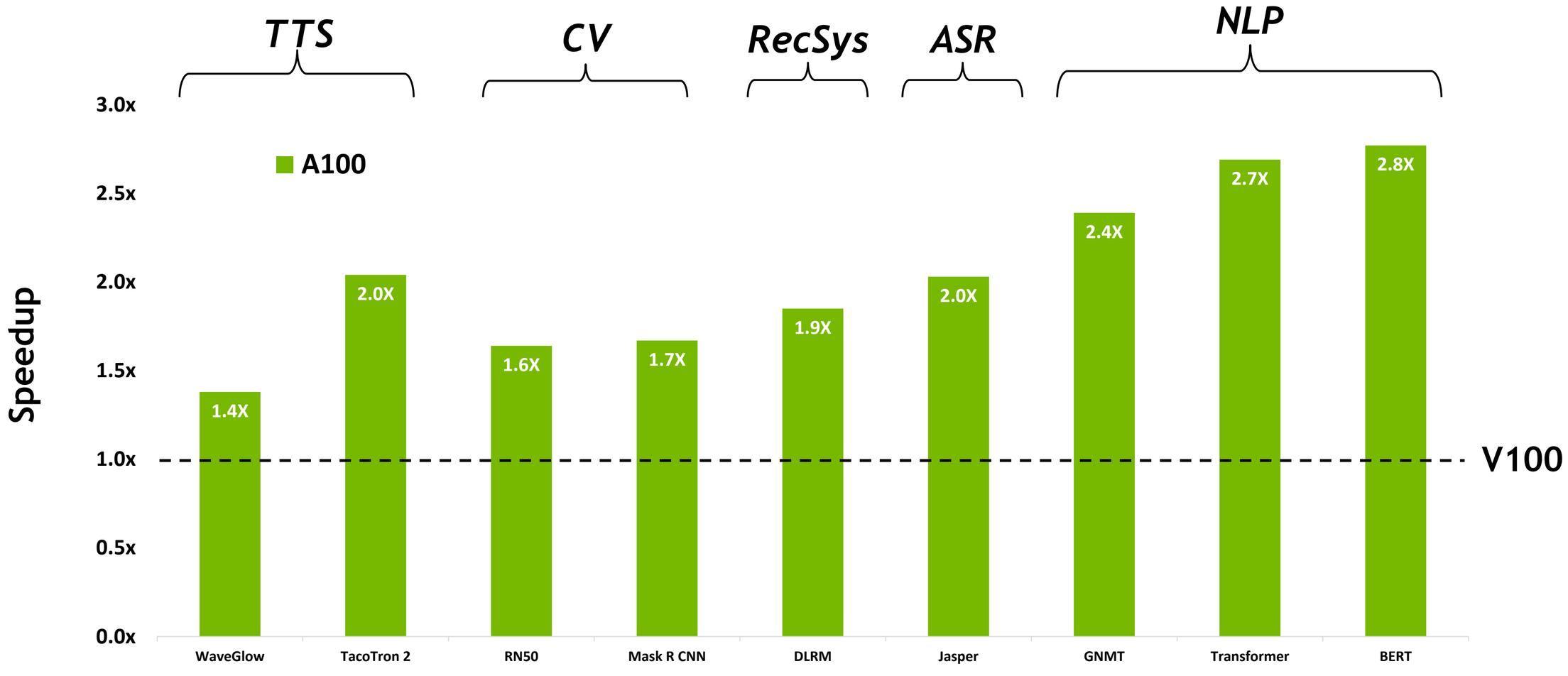
TensorFlow	**NVIDIA NGC コンテナイメージ 19.07 以降、TF 1.14 以降及び TF 2 以降では、オプティマイザのラッパーが利用可能:** **opt = tf.train.experimental.enable_mixed_precision_graph_rewrite (opt)** Keras mixed precision API in TF 2.1+ for eager execution https://tensorflow.org/api_docs/python/tf/train/experimental/enable_mixed_precision_graph_rewrite
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

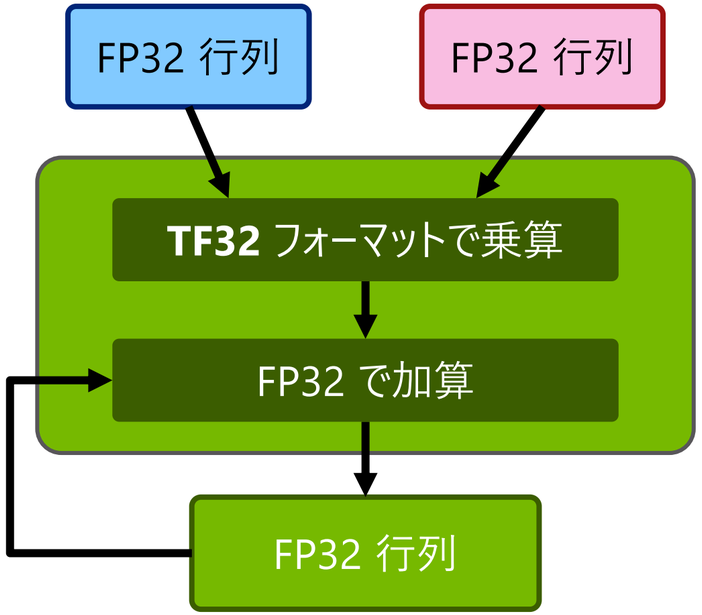
V100 (FP16) と A100 (FP16) の比較



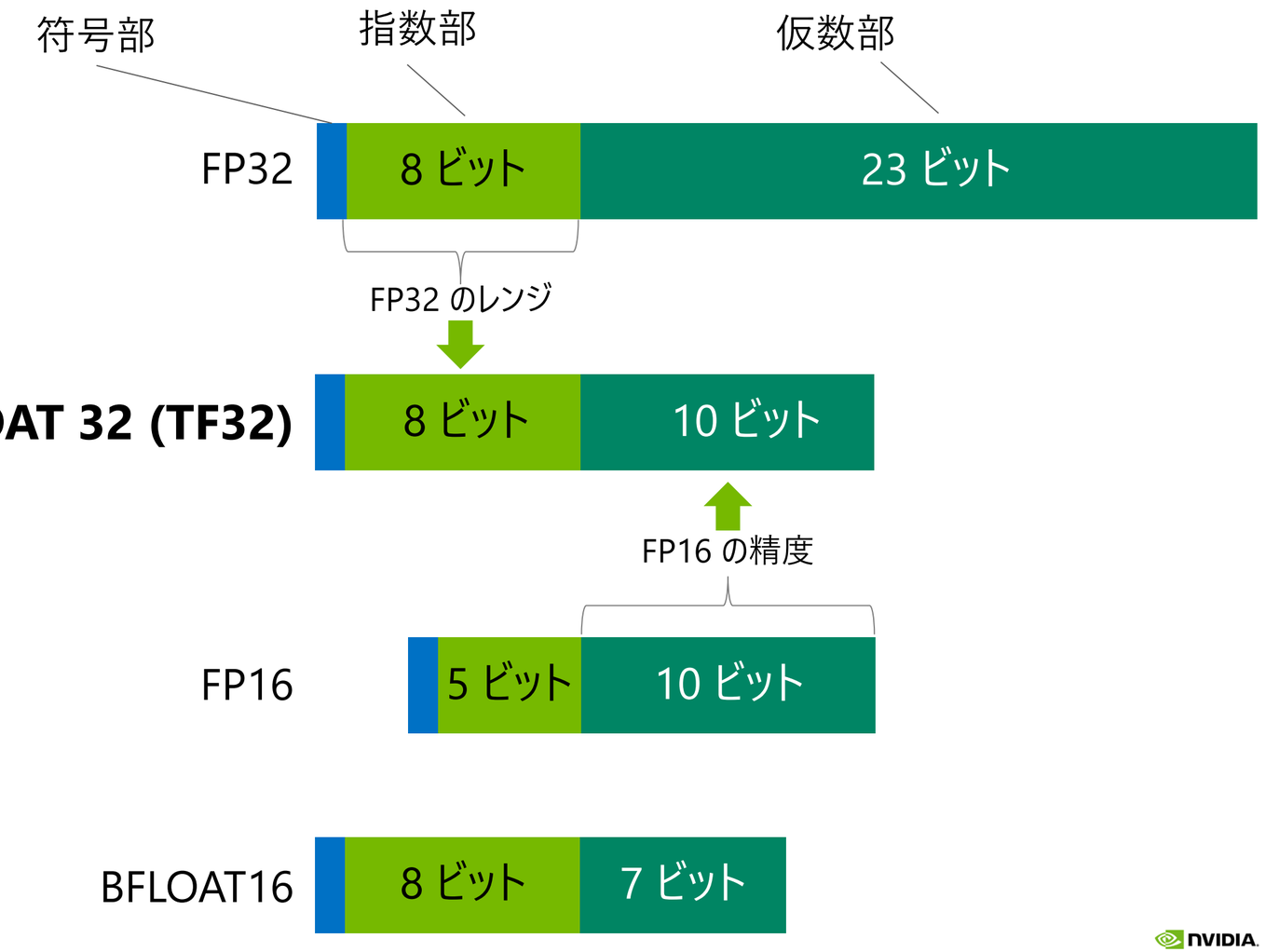
All results are measured
V100 used is DGX-1 (8xV100 16GB). A100 used is s DGX A100 (8xA100 SXM4), except DLRM which uses 1xV100 and 1xA100; all use FP16
RN50 uses MXNET Batch size =192, Mask R CNN uses PyTorch BS = 4 (V100) and BS=16 (A100), DLRM uses PyTorch and BS=32768, Jasper uses PyTorch and BS=32 (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

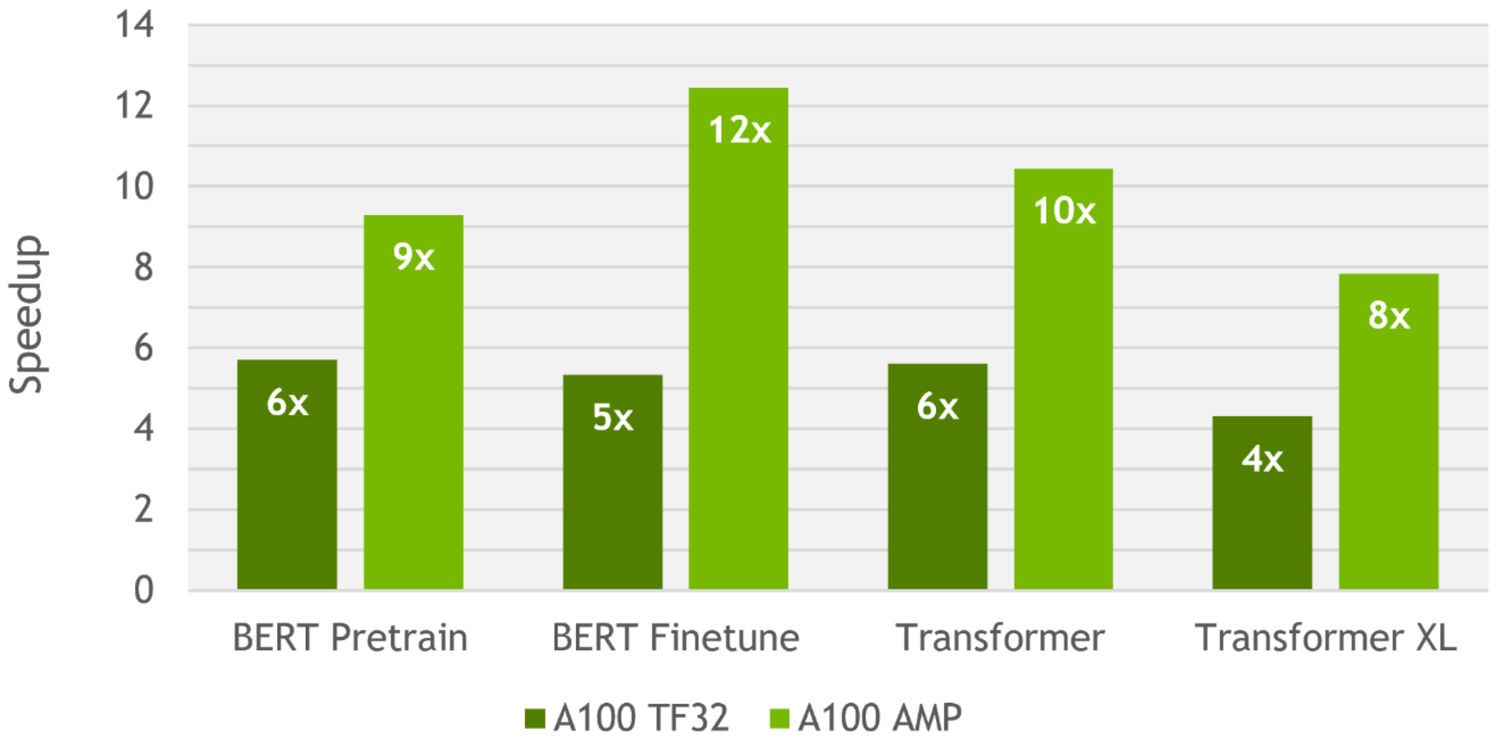


- FP32 の指数部、FP16 の仮数部
- FP32 を受け取り、**TF32** で乗算して FP32 で加算
- コード変更不要でモデルのトレーニングを高速化



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



V100 FP32 と比較

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 >= 8.0:

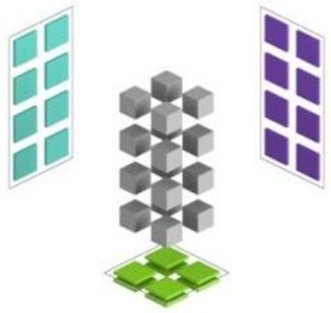
default: FP32 行列積を **TF32** で加速

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

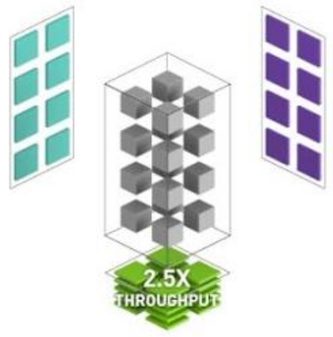
倍精度演算のピーク性能が 2.5 倍に

A100 の Tensor コアは FP64 に対応

NVIDIA V100 FP64

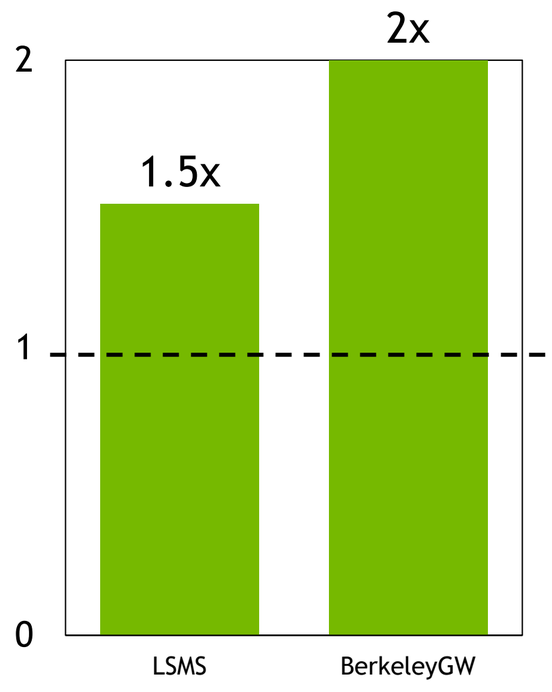


NVIDIA A100 Tensor コア FP64



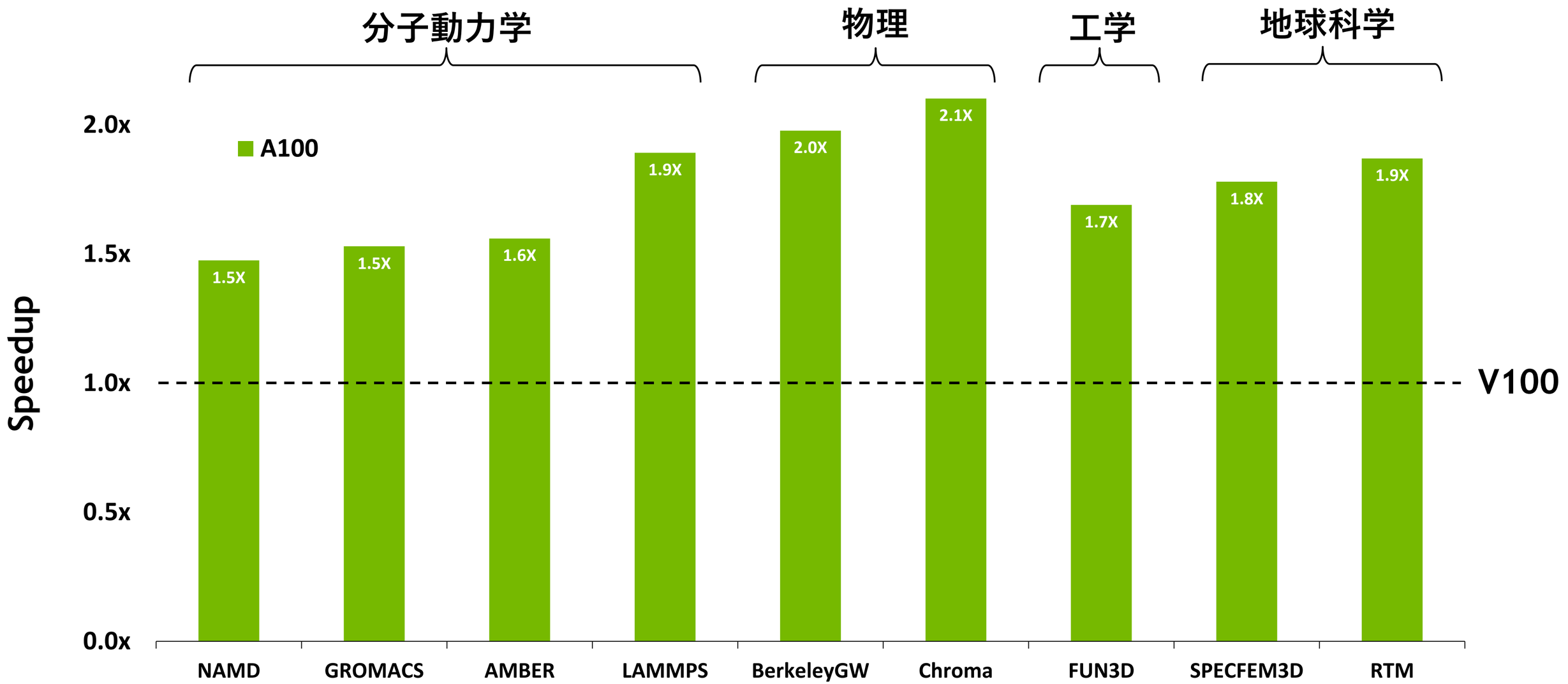
- IEEE 754 準拠の倍精度浮動小数点数
- cuBLAS, cuTensor, cuSolver 等のライブラリで対応

A100 Speedup vs. V100 (FP64)



Application [Benchmarks]: BerkeleyGW [Chi Sum + MTXEL] using DGX-1V (8xV100) and DGX-A100 (8xA100) | LSMS [Fe128] single V100 SXM2 vs. A100 SXM4

HPC アプリケーションの性能向上

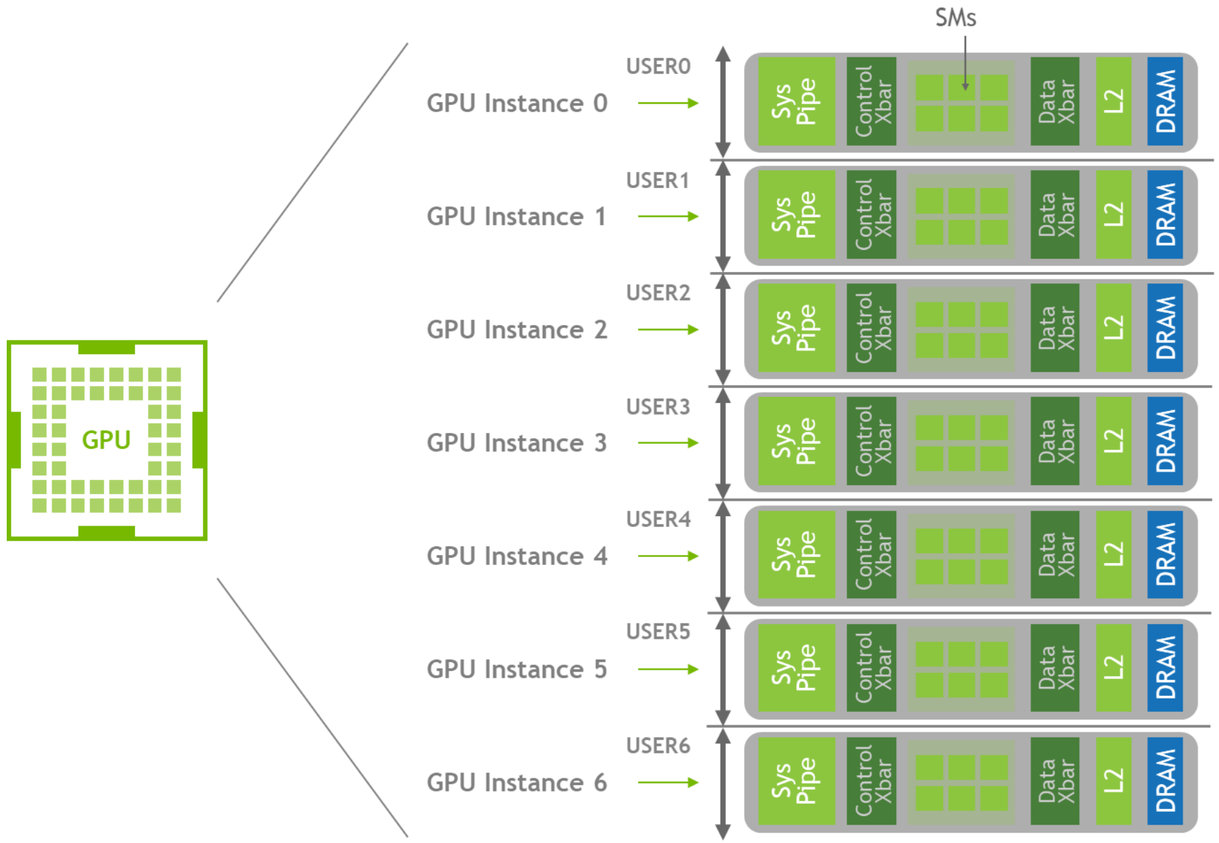


All results are measured
Except BerkeleyGW, V100 used is single V100 SXM2. A100 used is single A100 SXM4
More apps detail: AMBER based on PME-Cellulose, GROMACS with STMV (h-bond), LAMMPS with Atomic Fluid LJ-2.5, NAMD with v3.0a1 STMV_NVE
Chroma with szscl21_24_128, FUN3D with dpw, RTM with Isotropic 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

NVIDIA

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

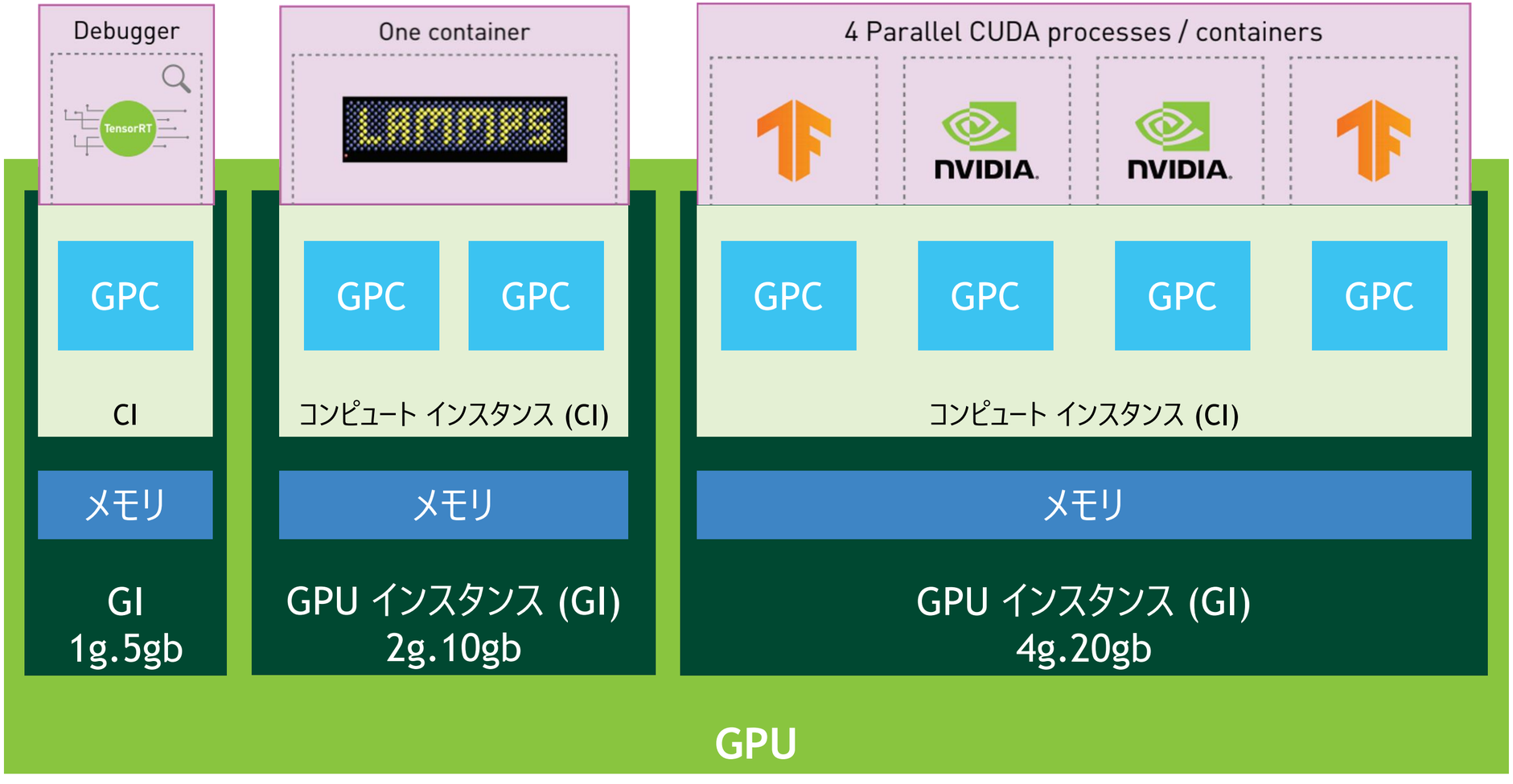
適切な GPU 割り当て:
ターゲットワークロードに応じて適切なサイズのインスタンスに分割可能

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

MIG User Guide: https://docs.nvidia.com/datacenter/tesla/mig-user-guide/index.html

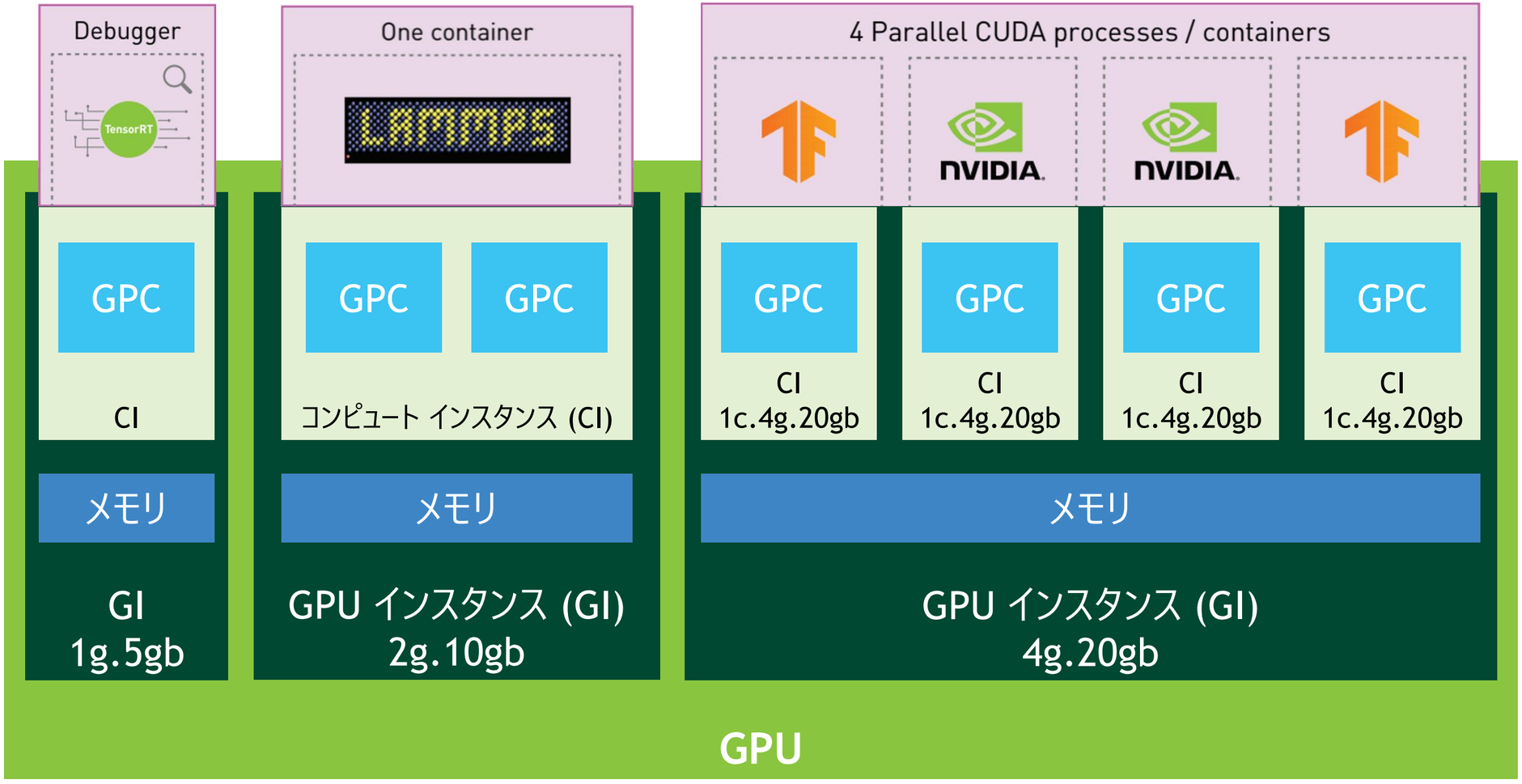
GPU インスタンスとコンピュート インスタンス

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		

GPU インスタンス プロファイル

For A100-SXM4-80GB

```
+-----------------------------------------------------------------------------+
| GPU instance profiles:                                                      |
| GPU   Name             ID    Instances   Memory     P2P    SM    DEC   ENC  |
|                              Free/Total   GiB              CE    JPEG  OFA  |
|=============================================================================|
|   0  MIG 1g.10gb       19     7/7         9.50       No     14     0     0  |
|                                                              1     0     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 1g.10gb+me    20     1/1         9.50       No     14     1     0  |
|                                                              1     1     1  |
+-----------------------------------------------------------------------------+
|   0  MIG 2g.20gb       14     3/3         19.50      No     28     1     0  |
|                                                              2     0     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 3g.40gb        9     2/2         39.50      No     42     2     0  |
|                                                              3     0     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 4g.40gb        5     1/1         39.50      No     56     2     0  |
|                                                              4     0     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 7g.80gb        0     1/1         79.25      No     98     5     0  |
|                                                              7     1     1  |
+-----------------------------------------------------------------------------+
```

GA100 と MIG

GA100 全体

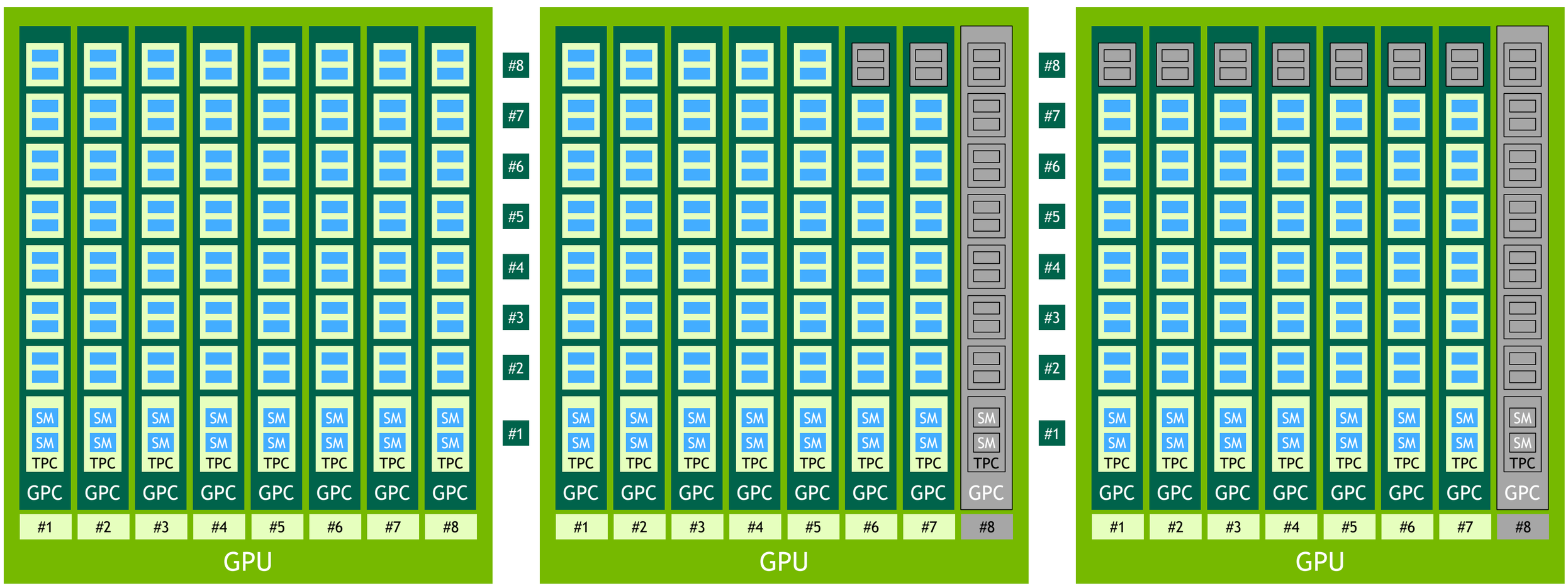
8 GPC, 8 TPC/GPC, 2 SM/TPC, 128 SM

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



Slurm は 21.08 から MIG に対応

MIG Management §

Beginning in version 21.08, Slurm now supports NVIDIA *Multi-Instance GPU* (MIG) devices. This feature allows some newer NVIDIA GPUs (like the A100) to split up a GPU into up to seven separate, isolated GPU instances. Slurm can treat these MIG instances as individual GPUs, complete with cgroup isolation and task binding.

To configure MIGs in Slurm, specify `AutoDetect=nvml` in *gres.conf* for the nodes with MIGs, and specify `Gres` in *slurm.conf* as if the MIGs were regular GPUs. An optional GRES type can be specified to distinguish MIGs of different sizes from each other, as well as from other GPUs in the cluster. This type must be a substring of the "MIG Profile" string as reported by the node in its slurmd log under the `debug2` log level. Other MIG attributes will be displayed as well, including MIG UUID, GPU Instance (GI) ID, Compute Instance (CI) ID, GI and CI minor numbers, the associated MIG device files, and "UniqueID" (the value Slurm will use to select MIGs via `CUDA_VISIBLE_DEVICES`).

The sanity-check AutoDetect mode is not supported for MIGs. Slurm expects MIG devices to already be partitioned, and does not support dynamic MIG partitioning.

For more information on NVIDIA MIGs (including how to partition them), see the MIG user guide.

https://slurm.schedmd.com/gres.html#MIG_Management

NVIDIA A2

NVIDIA A2

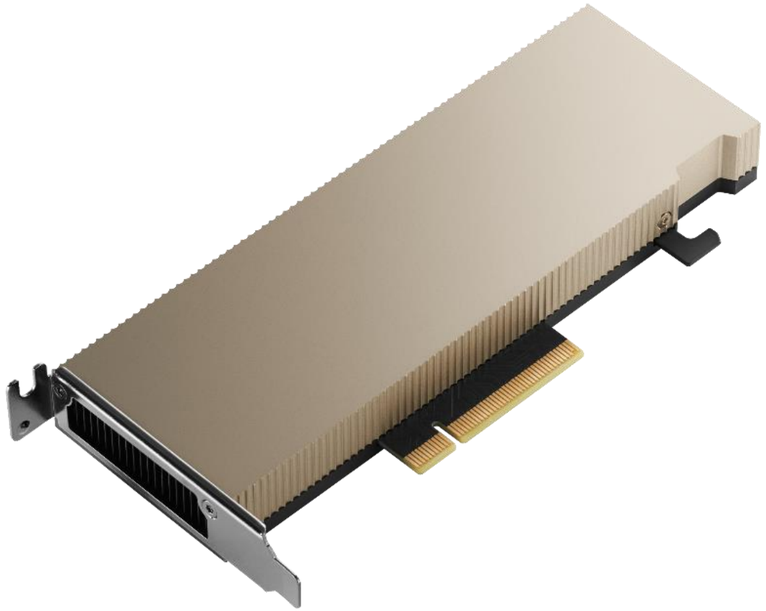
様々なサーバーに搭載可能なエントリーレベル GPU

Compact, Entry-Level Inference
Single slot LP, lower power – fits any server
& optimal for thermally constrained systems

Latest Ampere Architecture Features
3rd gen Tensor cores, 2nd gen RT cores, Secure RoT

Higher Intelligent Video Analytics (IVA) Performance
1.3X better performance vs T4

Up to 20X Higher Performance versus CPU
Speedups for AI inference and cloud gaming

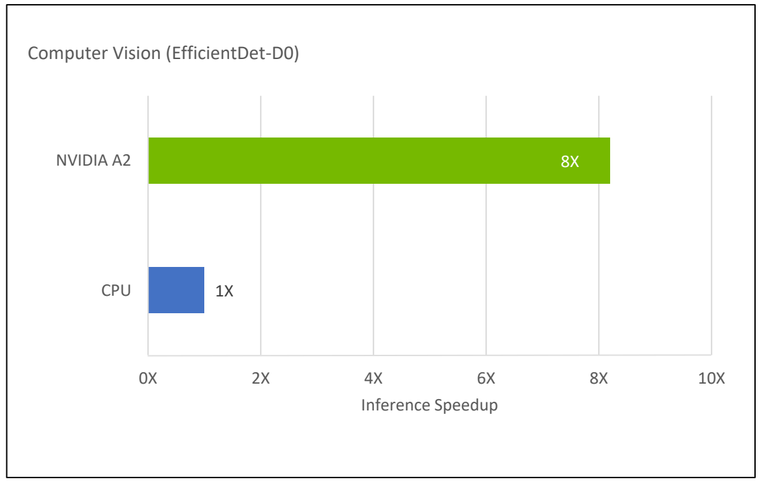


**Server availability 1-3 months later*

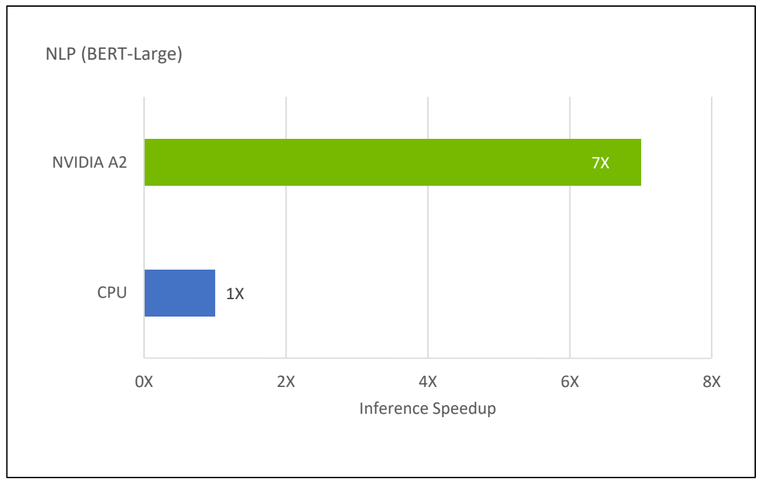
A2 が推論を高速化

CPU サーバーに対し最大 20 倍の高速化

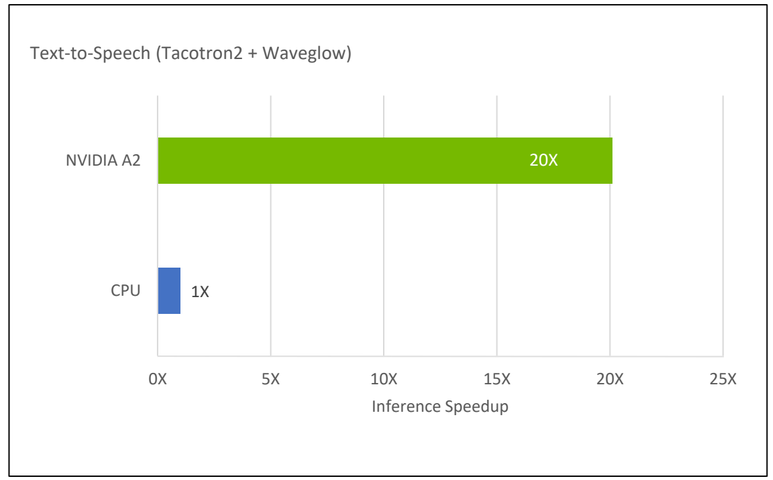
Computer Vision



Natural Language Processing



Text-to-Speech



Comparisons of one NVIDIA A2 Tensor Core GPU versus a dual-socket Xeon Gold 6330N CPU

System Config: [CPU: HPE DL380 Gen10 Plus, 2S Xeon Gold 6330N @2.2GHz, 512GB DDR4]
Computer Vision: EfficientDet-D0 (COCO, 512x512) | TensorRT 8.2, Precision: INT8, BS:8 (GPU) | OpenVINO 2021.4, Precision: INT8, BS:8 (CPU)
NLP: BERT-Large (Sequence length: 384, SQuAD: v1.1) | TensorRT 8.2, Precision: INT8, BS:1 (GPU) | OpenVINO 2021.4, Precision: INT8, BS:1 (CPU)
Text-to-Speech: Tacotron2 + Waveglow E2E pipeline (input length: 128) | PyTorch 1.9, Precision: FP16, BS:1 (GPU) | PyTorch 1.9, Precision: FP32, BS:1 (CPU)

NVIDIA A100X と NVIDIA A30X

DPU と GPU のコンバージド アクセラレーター

A100 / 30 Tensor Core GPU

BlueField-2 DPU

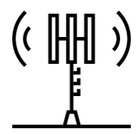
ConnectX-6 Dx

8 ARM A72 Cores

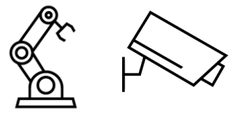
Integrated PCIe Gen4 Switch

2 slot FHFL

300 / 230 W

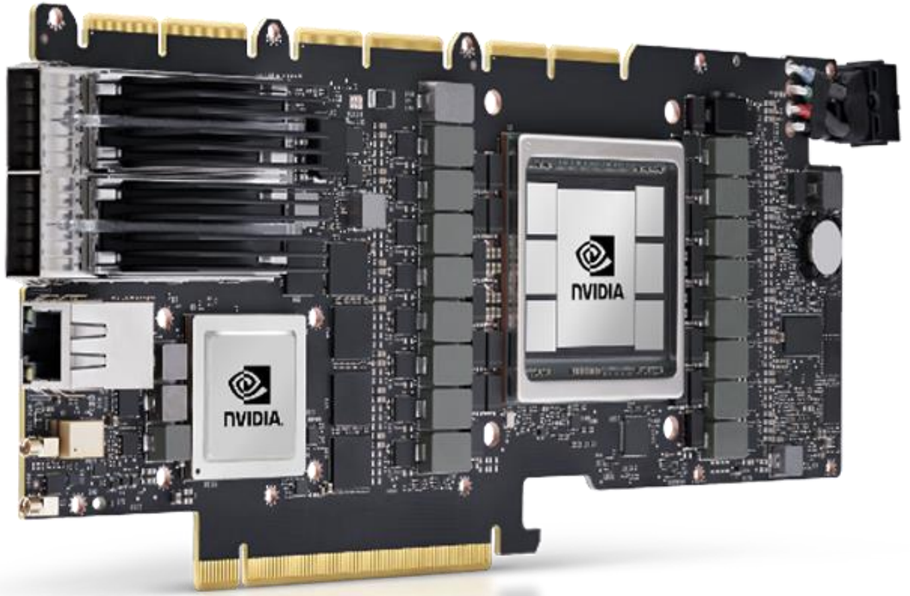


5G vRAN



AI on 5G

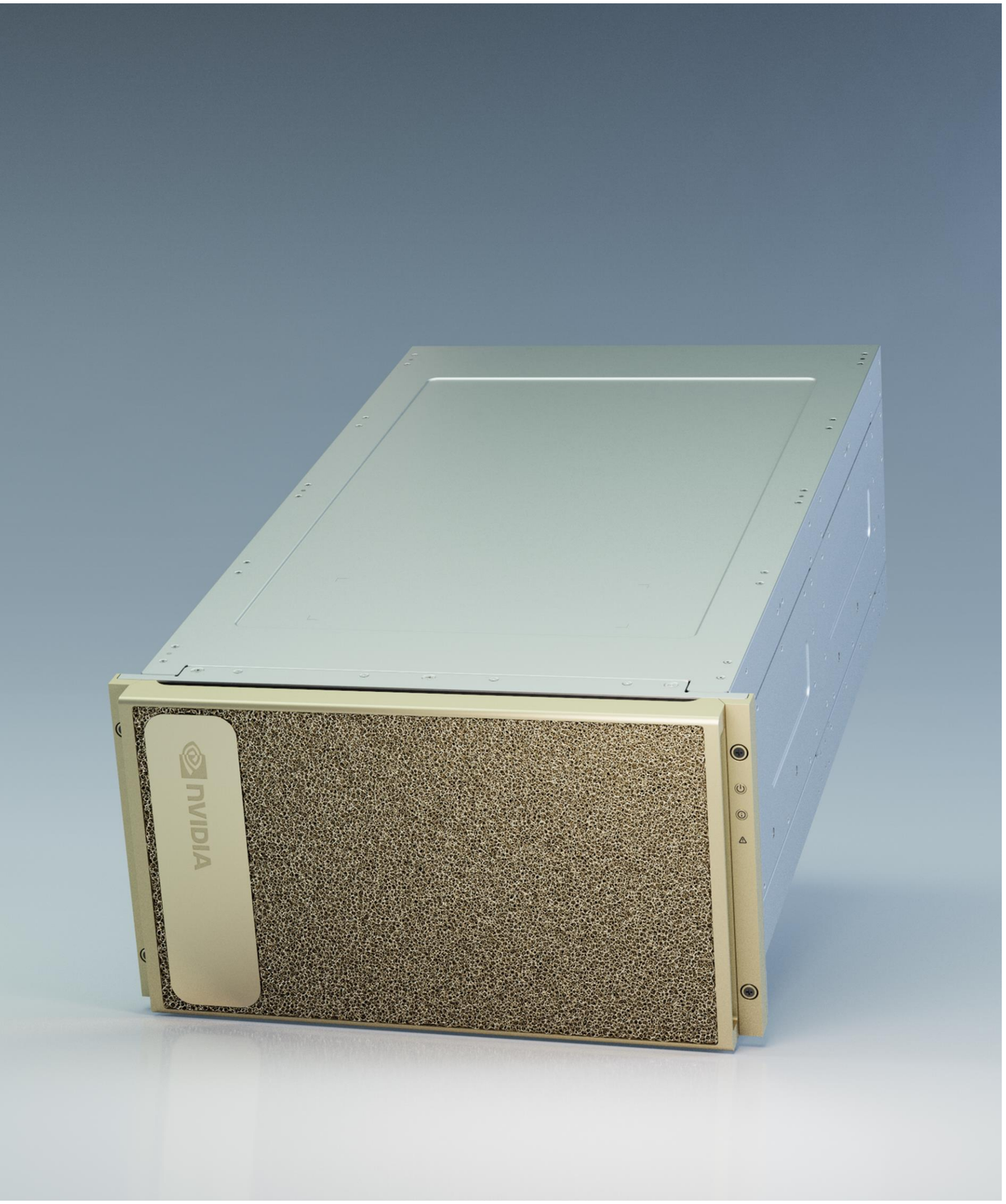
AI-Based Security



データセンター向け GPU 製品

	A100	A30	A2	A40	A16	A100X	A30X
Design	Highest Perf Compute	Mainstream Compute	Entry-Level Small Footprint	High Perf Graphics	High Density Virtual Desktop	High Perf Converged Accelerator	Mainstream Converged Accelerator
Max Power	300W	165W	40-60W	300W	250W	300W	230W
Form Factor	x16 PCIe Gen4 2 Slot FHFL 3 NVLink Bridge	x16 PCIe Gen4 2 Slot FHFL 1 NVLink Bridge	x8 PCIe Gen4 1 Slot LP	x16 PCIe Gen4 2 Slot FHFL 1 NVLink Bridge	x16 PCIe Gen4 2 Slot FHFL	x16 PCIe Gen4 2 Slot FHFL 3 NVLink Bridge	x16 PCIe Gen4 2 Slot FHFL 1 NVLink Bridge
GPU Memory	80GB HBM2e	24GB HBM2	16GB GDDR6	48GB GDDR6	4x 16GB GDDR6	80GB HBM2e	24GB HBM2e
Multi-Instance GPU (MIG)	Up to 7	Up to 4	-	-	-	Up to 7	Up to 4
Media Acceleration	1 JPEG Decoder 5 Video Decoder	1 JPEG Decoder 4 Video Decoder	1 Video Encoder 2 Video Decoder (+AV decode)	1 Video Encoder 2 Video Decoder (+AV1 decode)	4 Video Encoder 8 Video Decoder (+AV1 decode)	1 JPEG Decoder 5 Video Decoder	1 JPEG Decoder 4 Video Decoder
Ray Tracing	-		Yes	Yes	Yes	-	
Fast FP64	Yes		-	-	-	Yes	
Graphics	For in-situ visualization (no NVIDIA vPC or RTX vWS)		Good	Best	Better	For in-situ visualization (no NVIDIA vPC or RTX vWS)	
vGPU	Yes		Yes*	Yes	Yes	Yes*	
Hardware Root of Trust	Yes		Yes	Yes	Yes	Yes	
Integrated DPU	-	-	-	-	-	BlueField-2	
Server Availability	In Production	In Production	Q1 ’22	In Production	In Production	Q1 ‘22	

**Coming soon*



NVIDIA DGX A100

5 ペタフロップスの混合精度演算性能

8 基の NVIDIA A100 GPU で合計 640 GB の HBM2e メモリ

- GPU 毎に V100 の 2 倍となる 600GB/s の NVLink
- PCIe Gen4 の最大 10 倍の帯域幅

6 基の NVSwitch で全ての GPU を接続

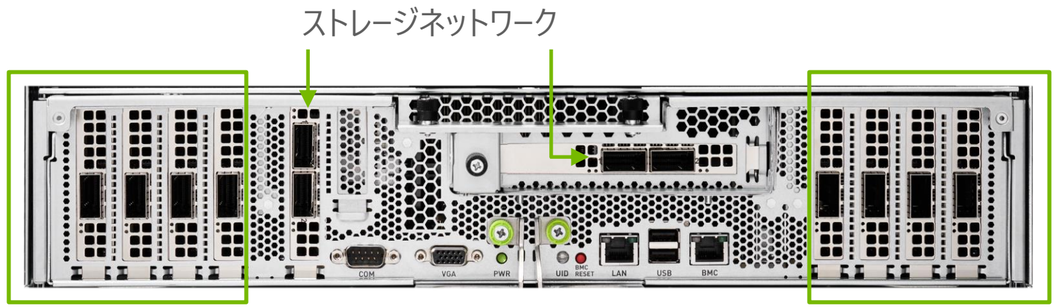
- 4.8TB/s のバイセクションバンド幅
- HD ビデオ 426 時間分に相当するデータを 1 秒で転送

2 基の AMD EPYC 7742 – 合計 128 コア

- PCIe Gen4 128 レーン
- 2 TB のメモリを標準搭載

MELLANOX ネットワーキングによる比類なき拡張性

ノード間通信とストレージアクセスに最高の性能を



シングルポート
CX-6 NIC

クラスターネットワーク:

- 8 枚のシングルポート Mellanox ConnectX-6
- HDR/HDR100/EDR InfiniBand と 200GigE をサポート

データ/ストレージネットワーク:

- 2ポートの Mellanox ConnectX-6 を 2 枚
  - Supporting: 200/100/50/40/25/10Gb Ethernet default or HDR/HDR100/EDR InfiniBand

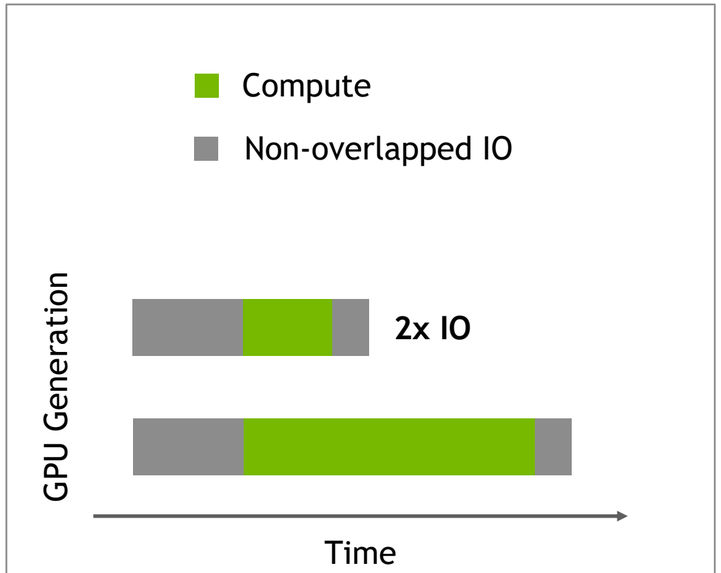
450GB/sec のバイセクション バンド幅

全ての I/O を PCIe Gen4 化、Gen3 の 2 倍高速

複数の DGX A100 ノードを Mellanox Quantum スイッチでスケール可能

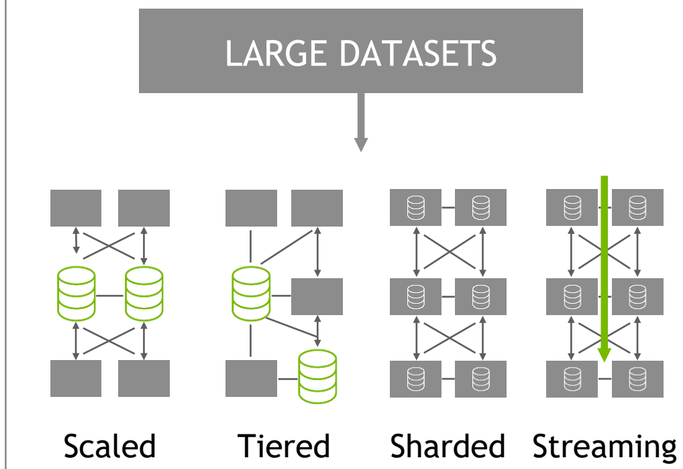
GPU Direct Storage (GDS)

EXTREME-SCALE AI REQUIRES EXTREME IO



LATENCY

IO latency dominates as compute shrinks



THROUGHPUT

Extreme throughput required as microservices scale across data center

180

Industry-Leading Organizations

2,500

Active Contributing Members

ECOSYSTEM

Diverse set of standards and solutions in the ecosystem

GPUDIRECT STORAGE GA AVAILABLE WITH CUDA 11.4

6.6X perf benefit in DL inference

Low latency, high throughput

Multi-node storage acceleration library

Supported by a strong ecosystem

Available on DGX

OS: DGX BASEOS 5.0, Ubuntu 18.04, 20.04, RHEL 8.3

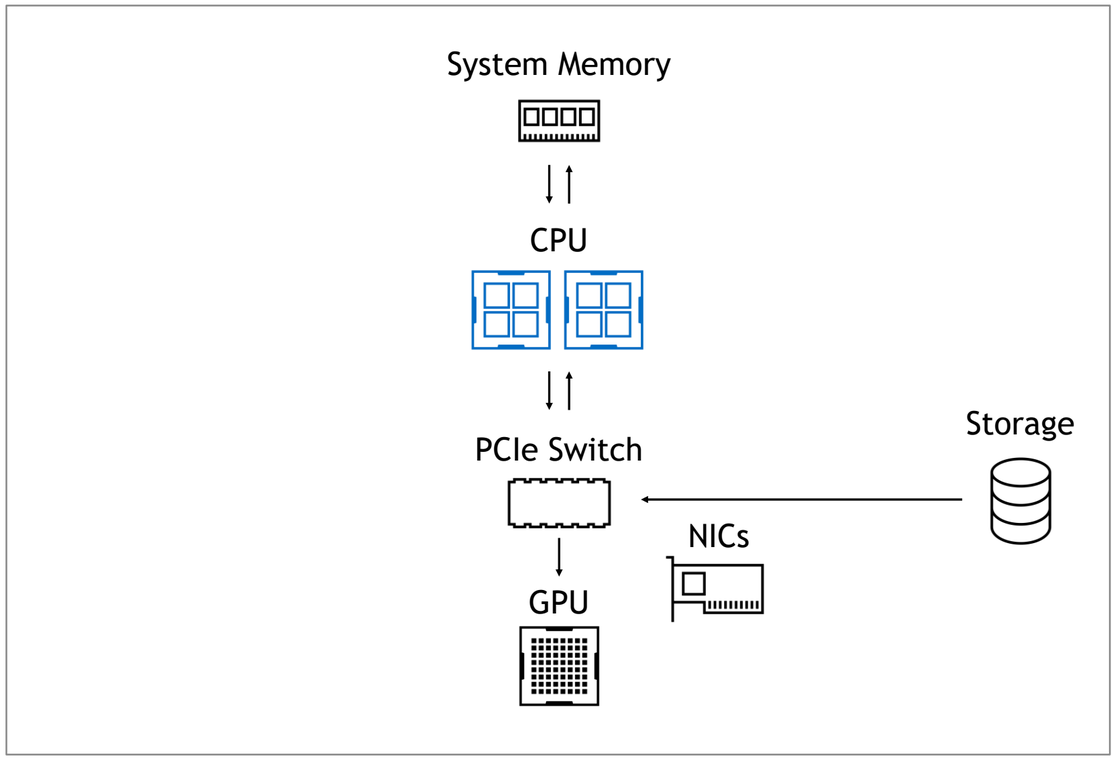
DL frameworks: PyTorch, MXNet

Data analytics frameworks: DALI, RAPIDS cuDF

CUDA 11.4 toolkit integration

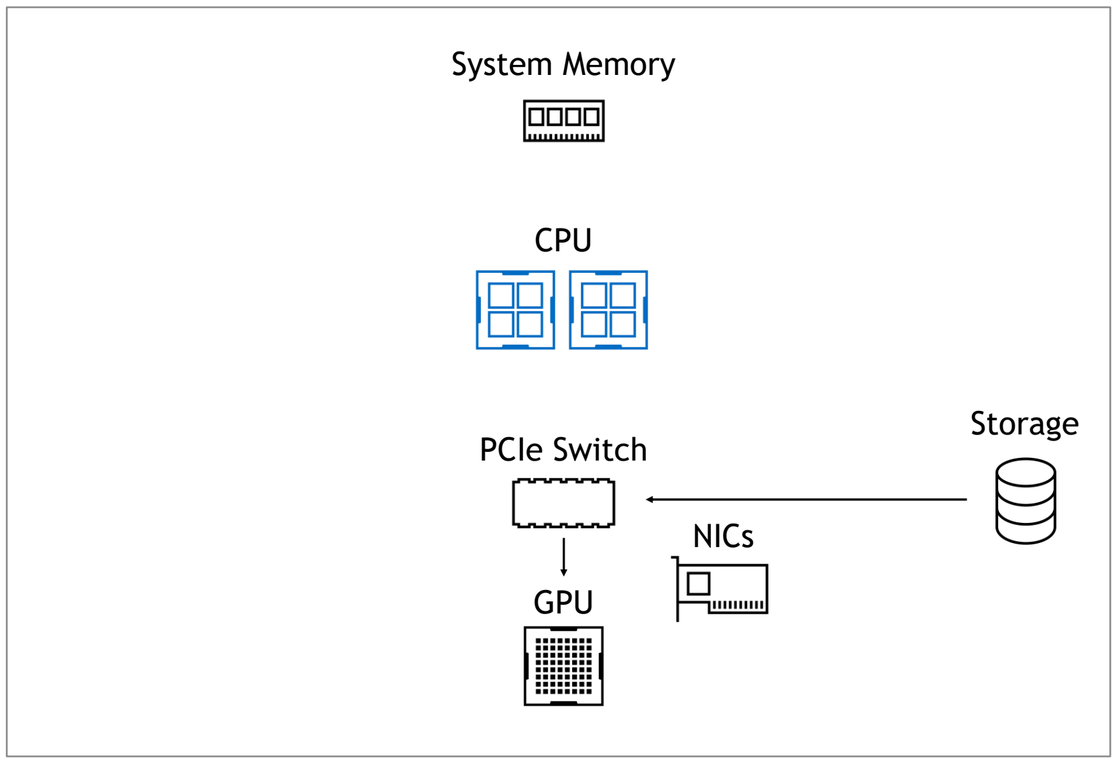
GETTING DATA TO GPUs: IO

WITHOUT GPUDIRECT STORAGE



Low Bandwidth | High Latency | Limited Capacity

WITH GPUDIRECT STORAGE

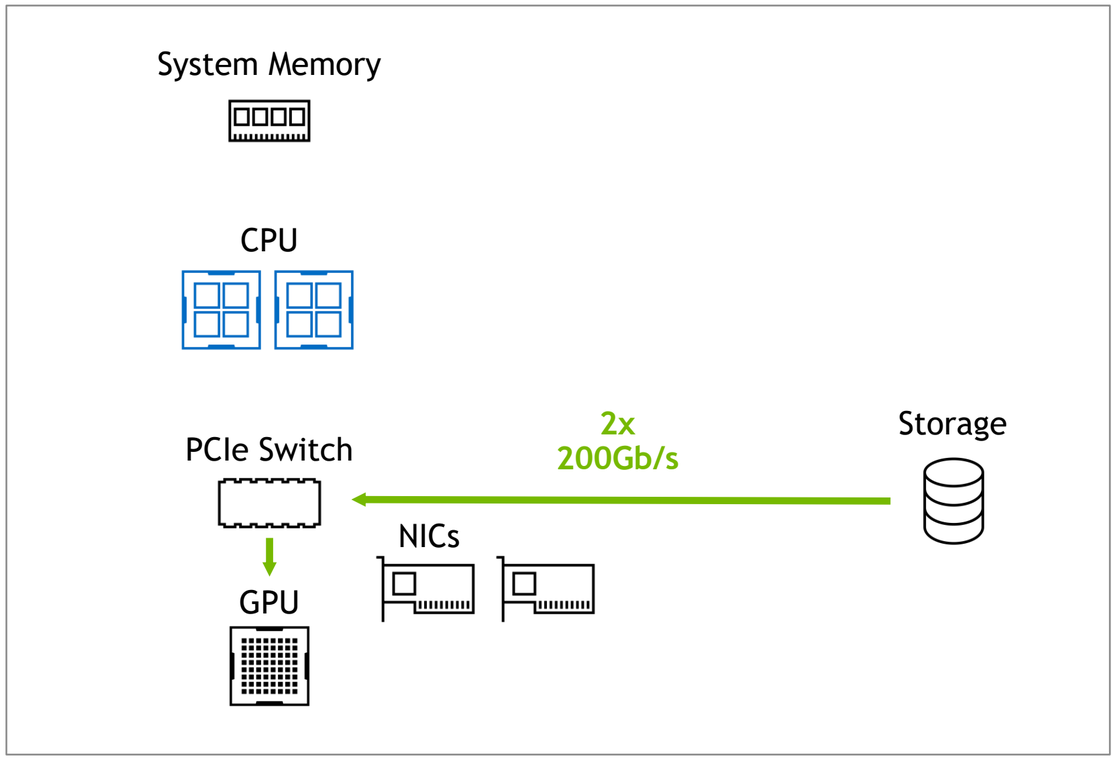


Higher Bandwidth | Lower Latency
O(PB) capacity | CUDA programming model

GDS CONFIGURATIONS ON DGX

Lower Latencies and Increased CPU Efficiency

STANDARD GDS CONFIGURATION IN DGX A100

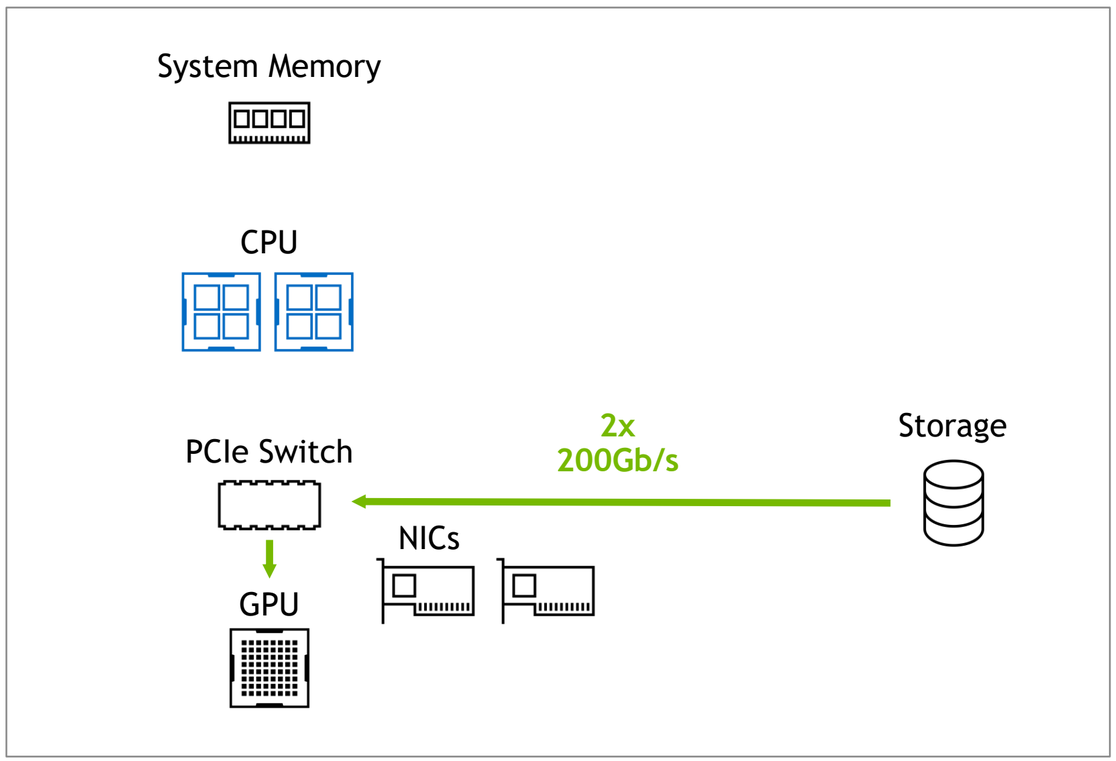


North-south storage configuration adapters 4 and 5 used for GDS

GDS CONFIGURATIONS ON DGX

Lower Latencies and Increased CPU Efficiency

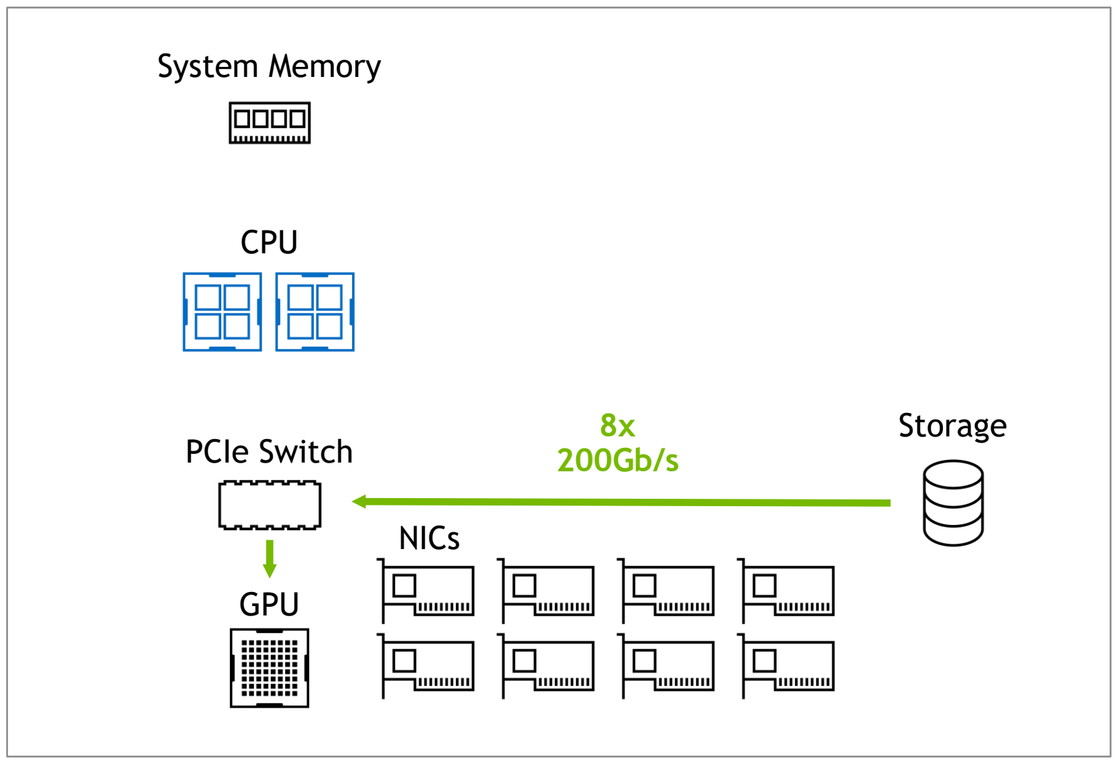
STANDARD GDS CONFIGURATION IN DGX A100



North-south storage configuration adapters 4 and 5 used for GDS

HIGH-THROUGHPUT GDS CONFIGURATION

only recommended for approved LHAs



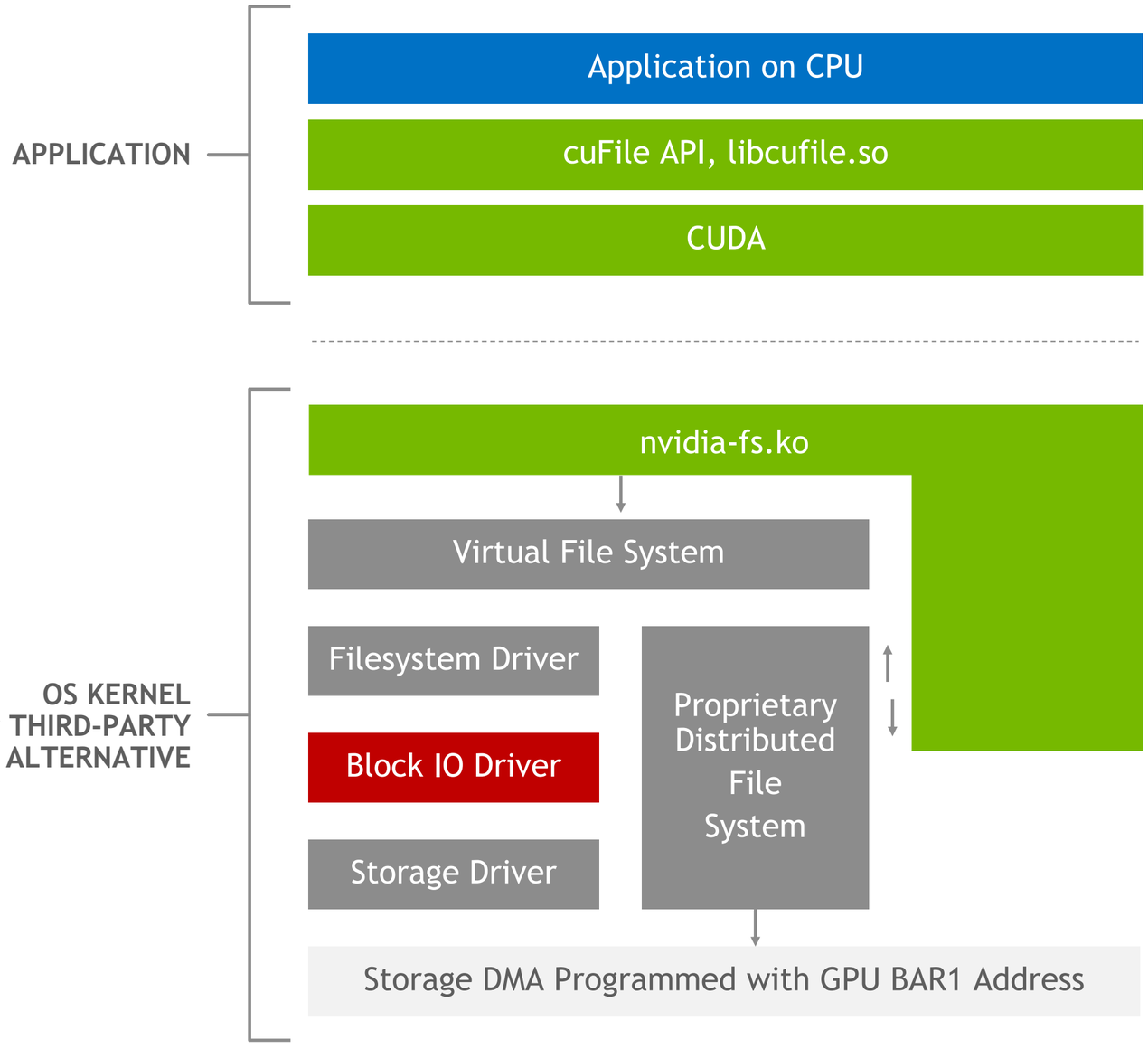
Converged east-west configuration adapters 0-3, 6-9 used for GDS

GPUDIRECT STORAGE SW ARCHITECTURE

cuFile user API
Enduring API for applications and frameworks

nvidia-fs driver API
For file system and block IO drivers
Vendor-proprietary solutions: no patching avoids lack of Linux enabling

NVIDIA is actively working with the community on upstream first to enable Linux to handle GPU VAs for DMA



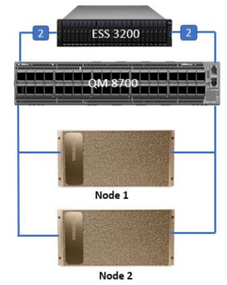
USAGE EXAMPLE

Write from GPU Memory to Local or Remote Storage

```
int main(void) {
        …                                                   // set GPU device, etc.

        CUfileError_t status;
        CUFileDescr_t cf_descr;                             // general enough to support Windows
        CUFileHandle_t cf_handle;

        status = cuFileDriverOpen();                        // initialize

        fd = open(TESTFILE, O_WRONLY|O_DIRECT, 0644);       // interop with normal file IO
        cf_descr.handle.fd = fd;
        status = cuFileHandleRegister(&cf_handle, &cf_descr);// Check support for file at this mount

        cuda_result = cuMemAlloc(&devPtr, size);            // User allocates memory

        status = cuFileBufRegister(devPtr, size);           // performance optimization
        assert(cudaMemset((void *) devPtr, 0xab, size) == cudaSuccess);
        ret = cuFileWrite(cf_handle, devPtr, size, 0, 0);  // ~pwrite: file handle, GPU base address, size,
                                                            //  offset, GPU buffer offset

        status = cuFileBufDeregister(devPtr);               // optional cleanup for good hygiene
        cuFree(devPtr);
        cuFileHandleDeregiser(cf_handle);
        close(fd);
        cuFileDriverClose();
        return 0;
}
```

GDS THROUGHPUT

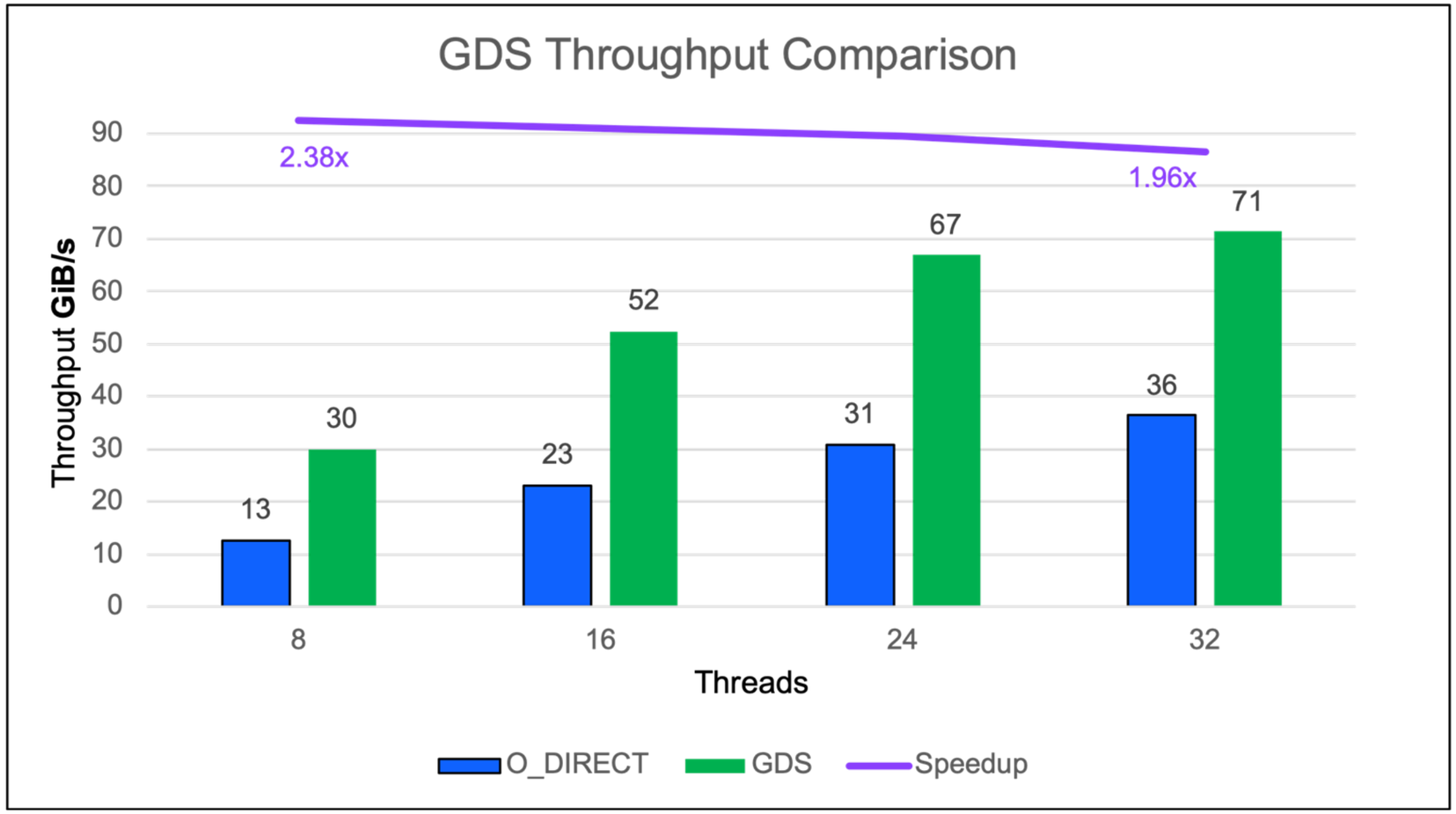
1X ESS 3200 AND 2X NVIDIA DGX A100



GPUDirect Storage removes the system bottlenecks to deliver almost full wire speed

Typical improvements are 2x when the storage and network can support the throughput

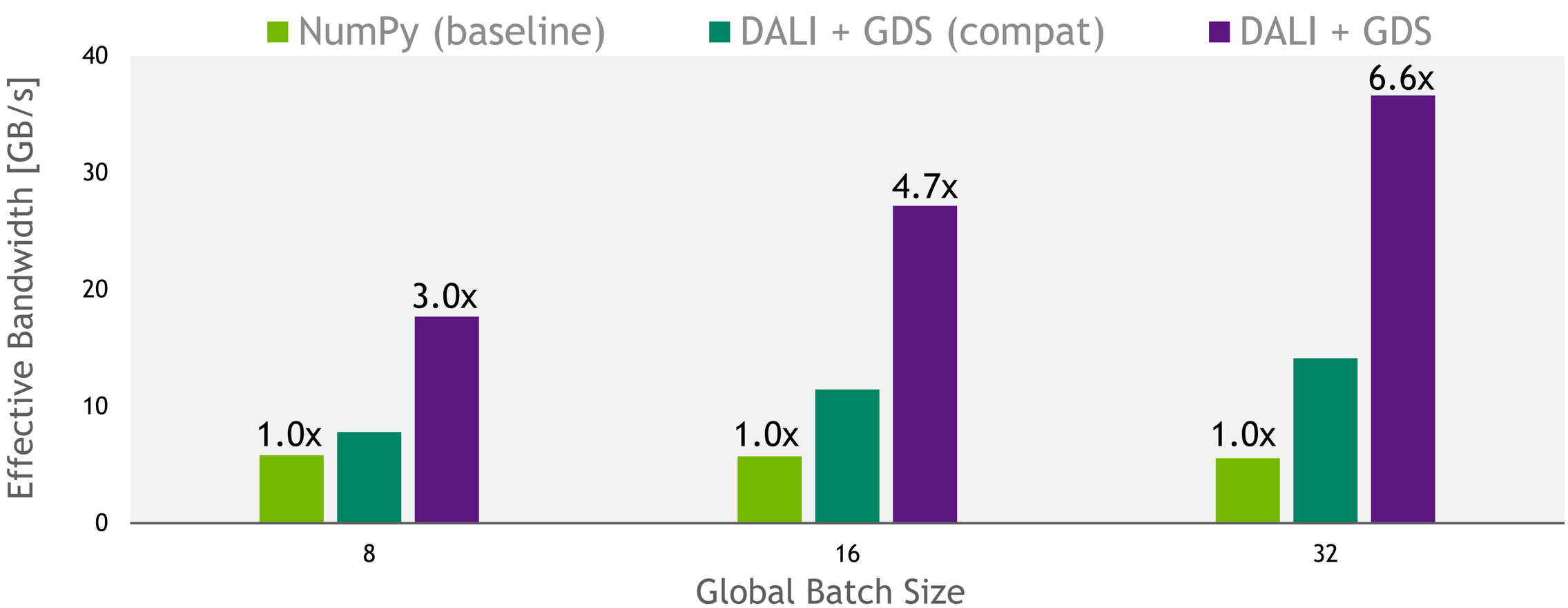
A single ESS 3200 delivering 77GB/s (71 GiB/s) over the storage fabric to two DGX A100s 16 GPUs (4x HDR)



IBM Lab Testing: GDS_DirectIO read measurements 1M I/O: One ESS 3200 running IBM Spectrum Scale 5.1.1, with four InfiniBand HDR to two NVIDIA DGX A100 servers using storage NICS

CASE STUDY : ACCELERATING DEEPCAM INFERENCE

DALI 1.3; DeepCAM is a component of MLPerf HPC



Performance benchmarking done with PyTorch DeepCAM using standard GDS configuration in DGX A100, UB 20.04 MLNX_OFED 5.3 GDS 1.0 DALI 1.3, EXAScaler 5.1.1 , 2.12.3_ddn29.
Application for batch size >= 32 limited by GPU compute throughput

ddn NVIDIA

GPU Direct Storage の要件

HOW TO DEVELOP WITH GDS?

What SW Is Needed to Get GDS to Work with DGX A100

USAGE SCENARIO	MECHANISM OF ACTION
C++ CUDA application	Install GDS (UML + KMD) Code change required to use cuFile API*
Python application w/ NumPy reader	Install GDS (UML + KMD) Import DALI reader (open-source)*
Spark/RAPIDS application	Install GDS (UML + KMD) GDS integrated

*ONE CHANGE FITS ALL

EXT4 with local (NVMe) or direct attached storage (NVMe-oF)
NFS (VAST, Isilon/PowerScale) | DFS (EXAScaler, WekaFS)

ストレージ要件の詳細はこちら: https://docs.nvidia.com/gpudirect-storage/o-direct-guide/index.html

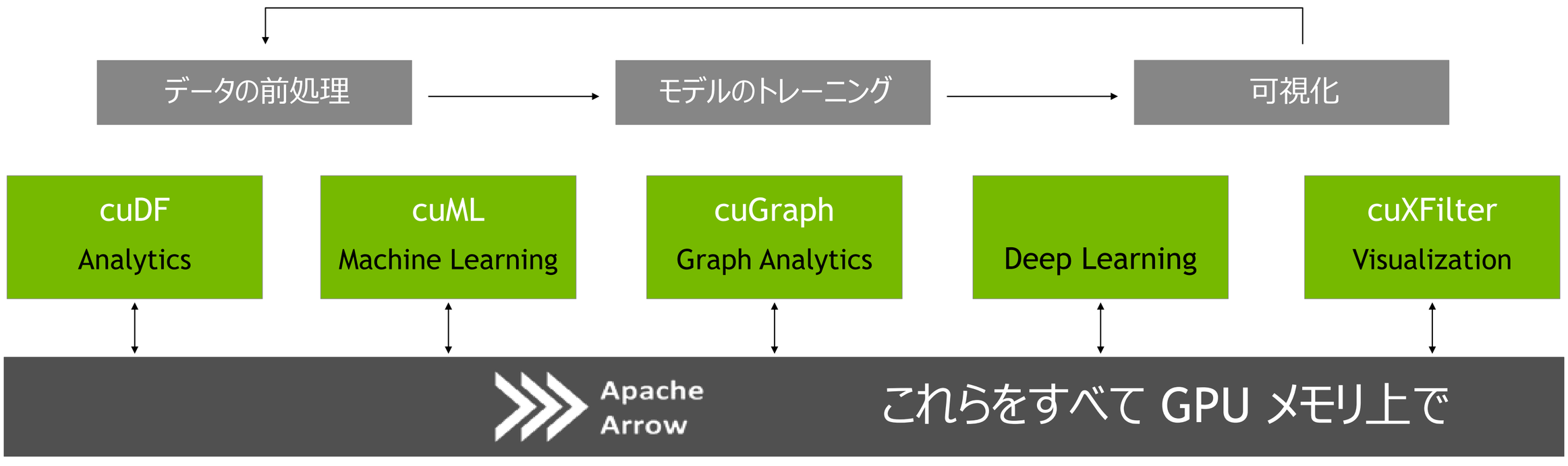
RAPIDS

RAPIDS

Open GPU Data Science

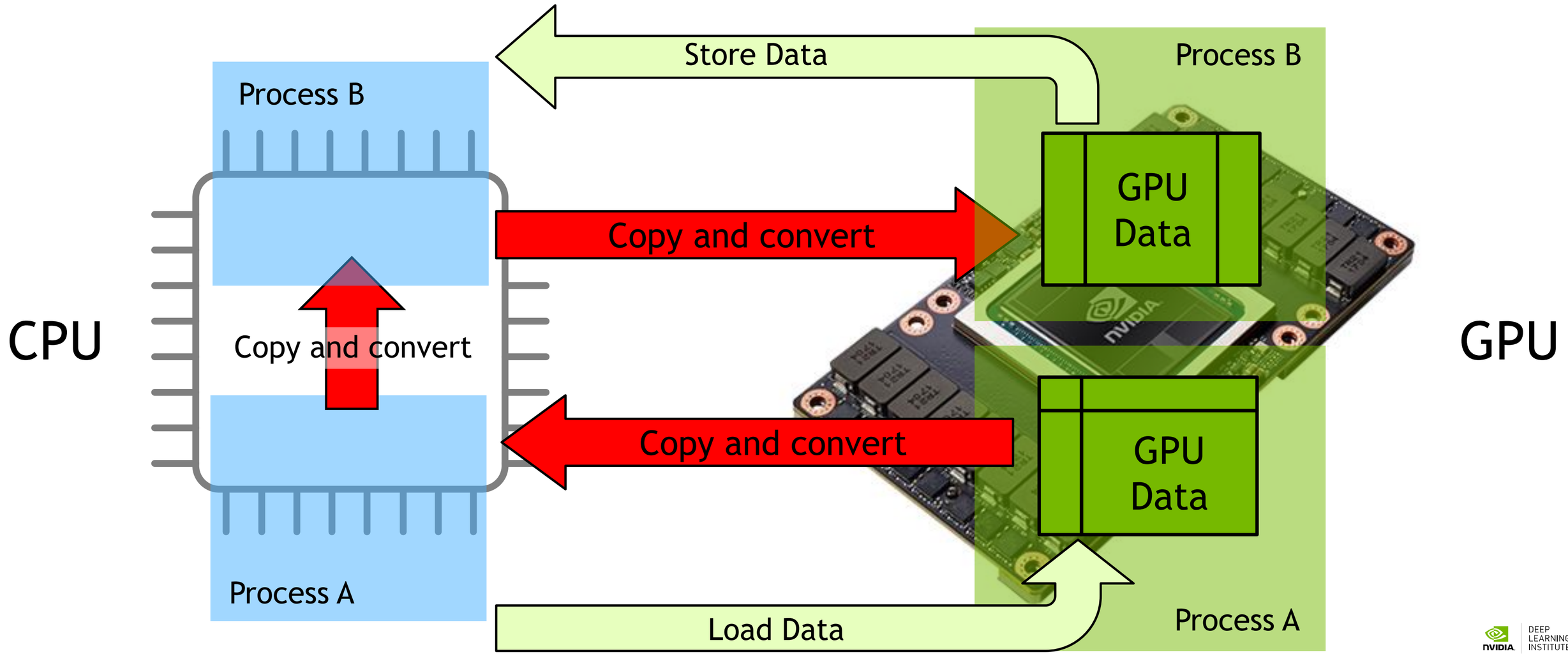
https://rapids.ai

データ分析作業の全体を GPU 上で高速に実行するためのオープンソース ライブラリ



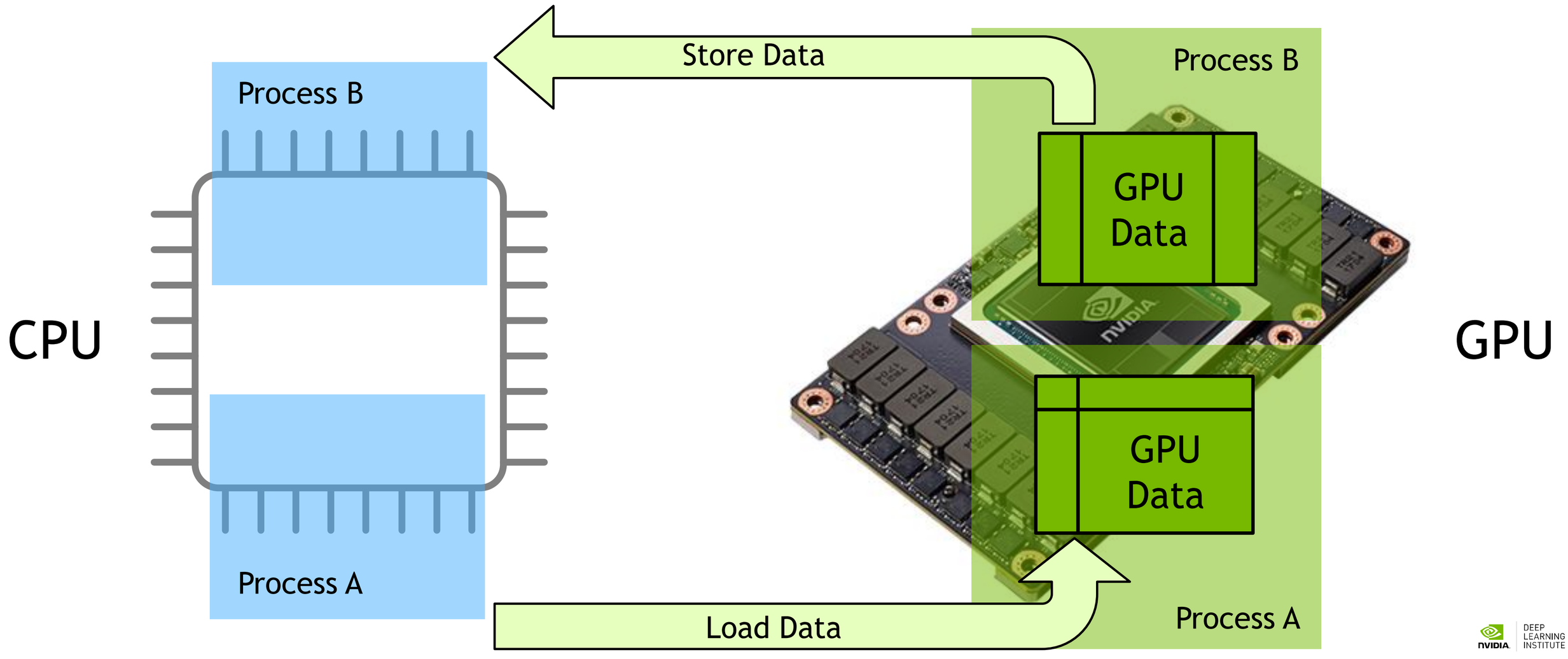
データの移送と変換

パフォーマンスと生産性に影響



データの移送と変換

データを共通形式で GPU 上にキープ



CUDF

GPU データフレーム ライブラリ

	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitude	...	Y2004	Y2005	Y2006	Y2007	Y2008
0	AF	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.71	...	3249.0	3486.0	3704.0	4164.0	4252.0
1	AF	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.71	...	419.0	445.0	546.0	455.0	490.0
2	AF	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.71	...	58.0	236.0	262.0	263.0	230.0
3	AF	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.71	...	185.0	43.0	44.0	48.0	62.0
4	AF	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.71	...	120.0	208.0	233.0	249.0	247.0
5	AF	2	Afghanistan	2514	Maize and products	5142	Food	1000 tonnes	33.94	67.71	...	231.0	67.0	82.0	67.0	69.0
6	AF	2	Afghanistan	2517	Millet and products	5142	Food	1000 tonnes	33.94	67.71	...	15.0	21.0	11.0	19.0	21.0
7	AF	2	Afghanistan	2520	Cereals, Other	5142	Food	1000 tonnes	33.94	67.71	...	2.0	1.0	1.0	0.0	0.0
8	AF	2	Afghanistan	2531	Potatoes and products	5142	Food	1000 tonnes	33.94	67.71	...	276.0	294.0	294.0	260.0	242.0
9	AF	2	Afghanistan	2536	Sugar cane	5521	Feed	1000 tonnes	33.94	67.71	...	50.0	29.0	61.0	65.0	54.0
10	AF	2	Afghanistan	2537	Sugar beet	5521	Feed	1000 tonnes	33.94	67.71	...	0.0	0.0	0.0	0.0	0.0

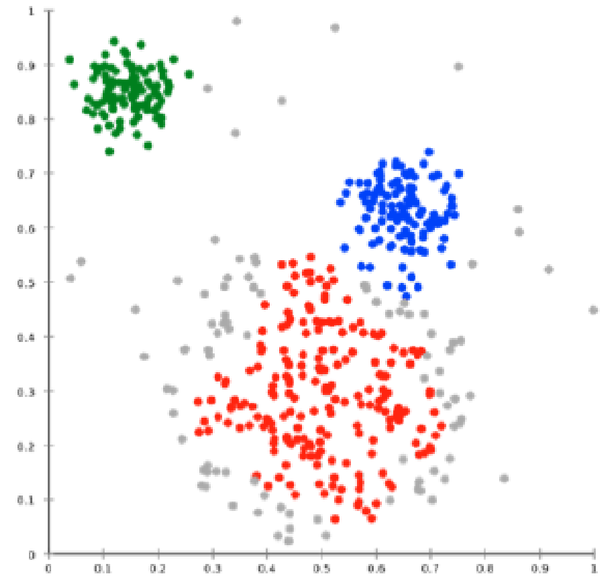
Apache Arrow のデータフォーマット

Pandas 的な API

- Unary and Binary Operations
- Joins, Merges
- GroupBys
- Filters
- File readers
- User Defined Functions
- Etc.

CUML

GPU 対応 scikit-learn 的なライブラリ



- **Classification / Regression**
  - Decision Trees / Random Forests
  - Linear Regression
  - Logistic Regression
  - K-Nearest Neighbors
  - Kalman Filtering
  - Bayesian Inference
  - Gaussian Mixture Models
  - Hidden Markov Models
- **Clustering**
  - K-Means
  - DBSCAN
  - Spectral Clustering
- **Decomposition & Dimensionality Reduction**
  - Principal Components
  - Singular Value Decomposition
  - UMAP
  - Spectral Embedding
- **Timeseries Forecasting**
  - ARIMA
  - Holt-Winters
- **Recommendations**
  - Implicit Matrix Factorization
- **Cross Validation**
- **Hyper-parameter Tuning**

CPU vs GPU

PCA
(Principal Component Analysis)

System: AWS p3.8xlarge
CPUs: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz, 32 vCPU cores, 244 GB RAM
GPU: Tesla V100 SXM2 16GB
Dataset: https://github.com/rapidsai/cuml/tree/master/python/notebooks/data

sklearn (CPU)

```
# import algorithm
from sklearn.decomposition import PCA

# helper functions
from helper import *

# data loading
x = load_data(2**22, 400)



# algorithm parameters
n_components = 8
whiten = False
random_state = 42
svd_solver = “full”

# training
pca = PCA(n_components, svd_solver,
        whiten, randome_state)
_ = pca.fit_transform(x)
```

57.1 seconds

training time

cuml (GPU)

```
# import algorithm
from cuml import PCA

# helper functions
from helper import *

# data loading
x = load_data(2**22, 400)

import cudf
x = cudf.DataFrame.from_pandas(x)

# algorithm parameters
n_components = 8
whiten = False
random_state = 42
svd_solver = “full”

# training
pca = PCA(n_components, svd_solver,
        whiten, randome_state)
_ = pca.fit_transform(x)
```

4.3 seconds

CPU vs GPU

KNN
(*k*-Nearest Neighbors)

System: AWS p3.8xlarge
CPUs: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz, 32 vCPU cores, 244 GB RAM
GPU: Tesla V100 SXM2 16GB
Dataset: https://github.com/rapidsai/cuml/tree/master/python/notebooks/data

sklearn (CPU)

```
# import algorithm
from sklearn.neighbors import
NearestNeighbors as KNN

# helper functions
from helper import *

# data loading
x = load_data(2**17, 40)



# algorithm parameters
n_neighbors = 10

# training
knn = KNN()
knn.fit(x)
_ = knn.kneighbors(x, n_neighbors)
```

9 minutes

training time

cuml (GPU)

```
# import algorithm
from cuml.neighbors.nearest_neighbors
import NearestNeighbors as KNN

# helper functions
from helper import *

# data loading
x = load_data(2**17, 40)

import cudf
x = cudf.DataFrame.from_pandas(x)

# algorithm parameters
n_neighbors = 10

# training
knn = KNN(n_gpus=1)
knn.fit(x)
_ = knn.kneighbors(x, n_neighbors)
```

1.12 seconds

NETWORKING

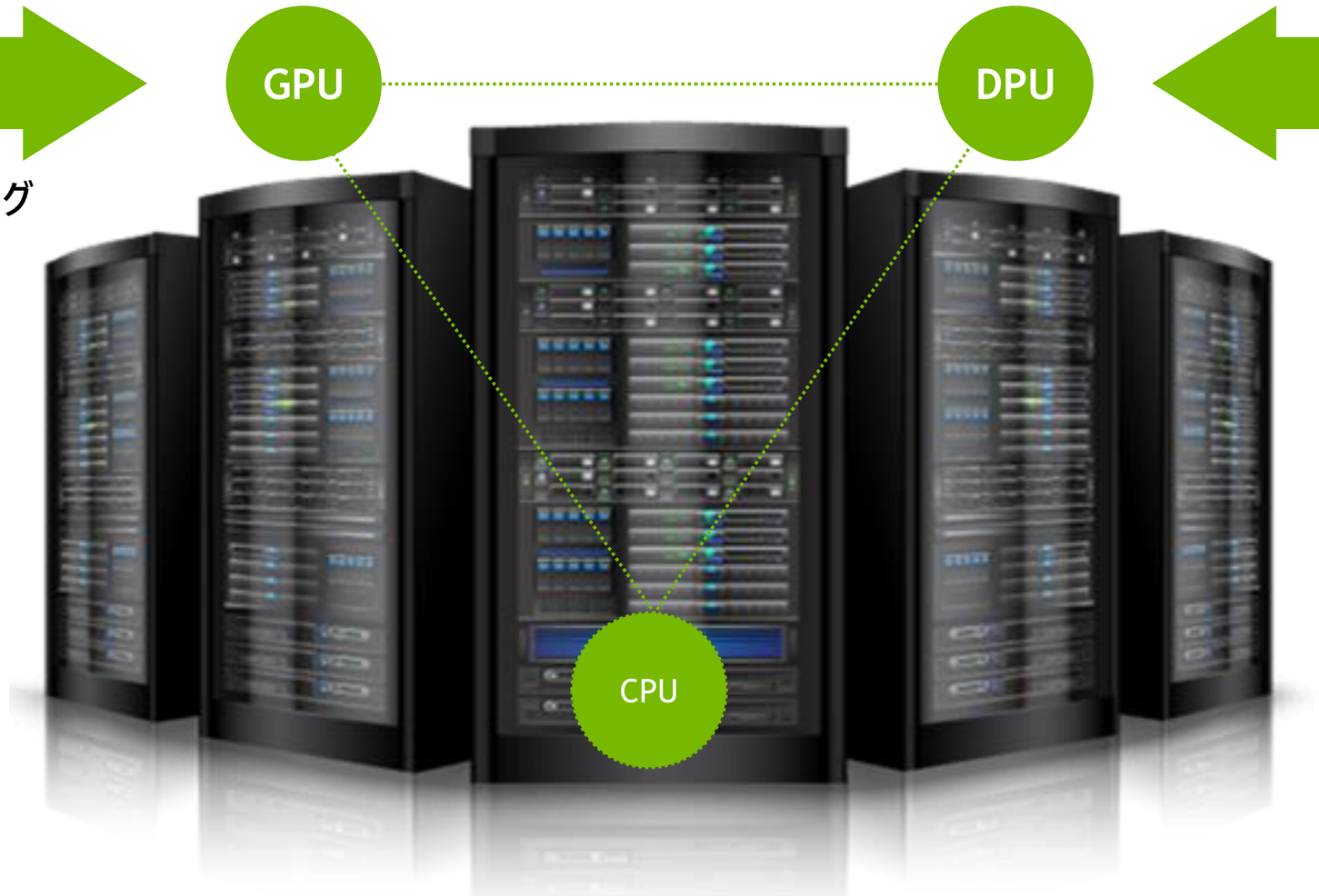
IN-NETWORK がスーパーコンピューティングを飛躍的に加速

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

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



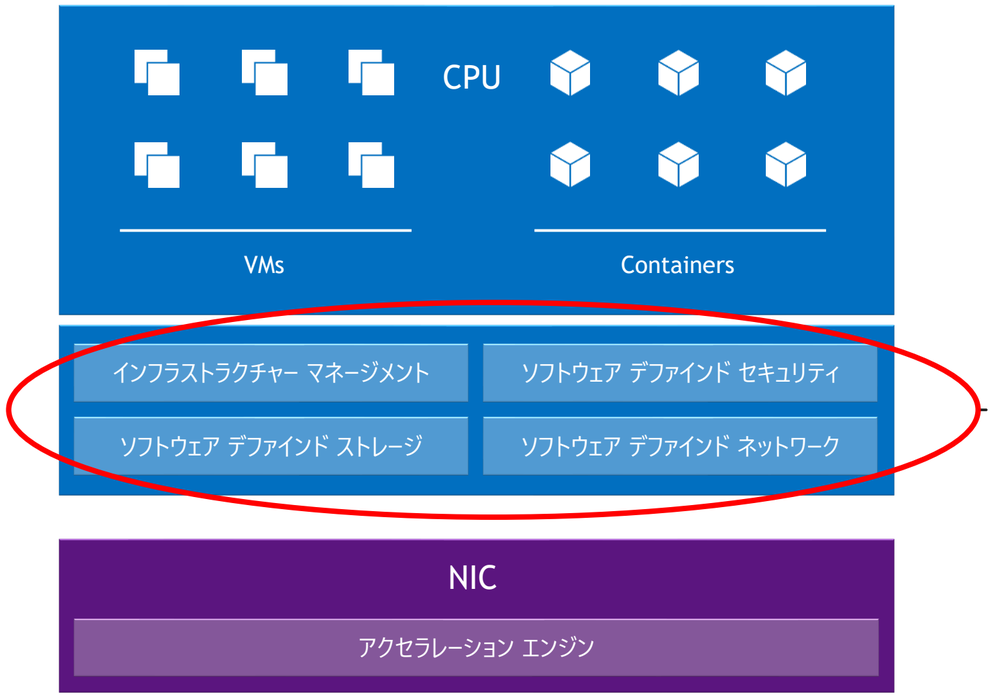
Data Processing Unit

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

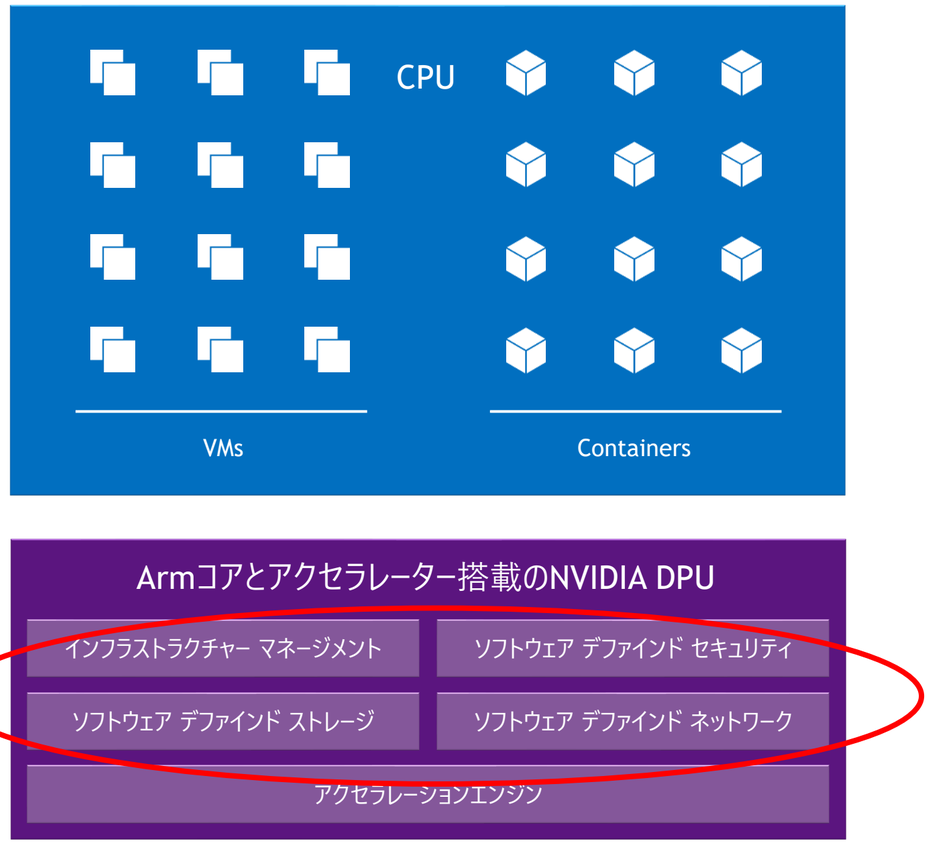
DPUはデータ集約型タスクを加速
ネットワーク、セキュリティ、ストレージ

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



手動インフラストラクチャ管理| セキュリティ アプライアンス| ストレージシステム| 静的ネットワーク| マイクロサービス|エンドツーエンド接続| ストレージアクセス| ゼロトラストセキュリティ



NVIDIA BLUEFIELD-2 DATA PROCESSING UNIT

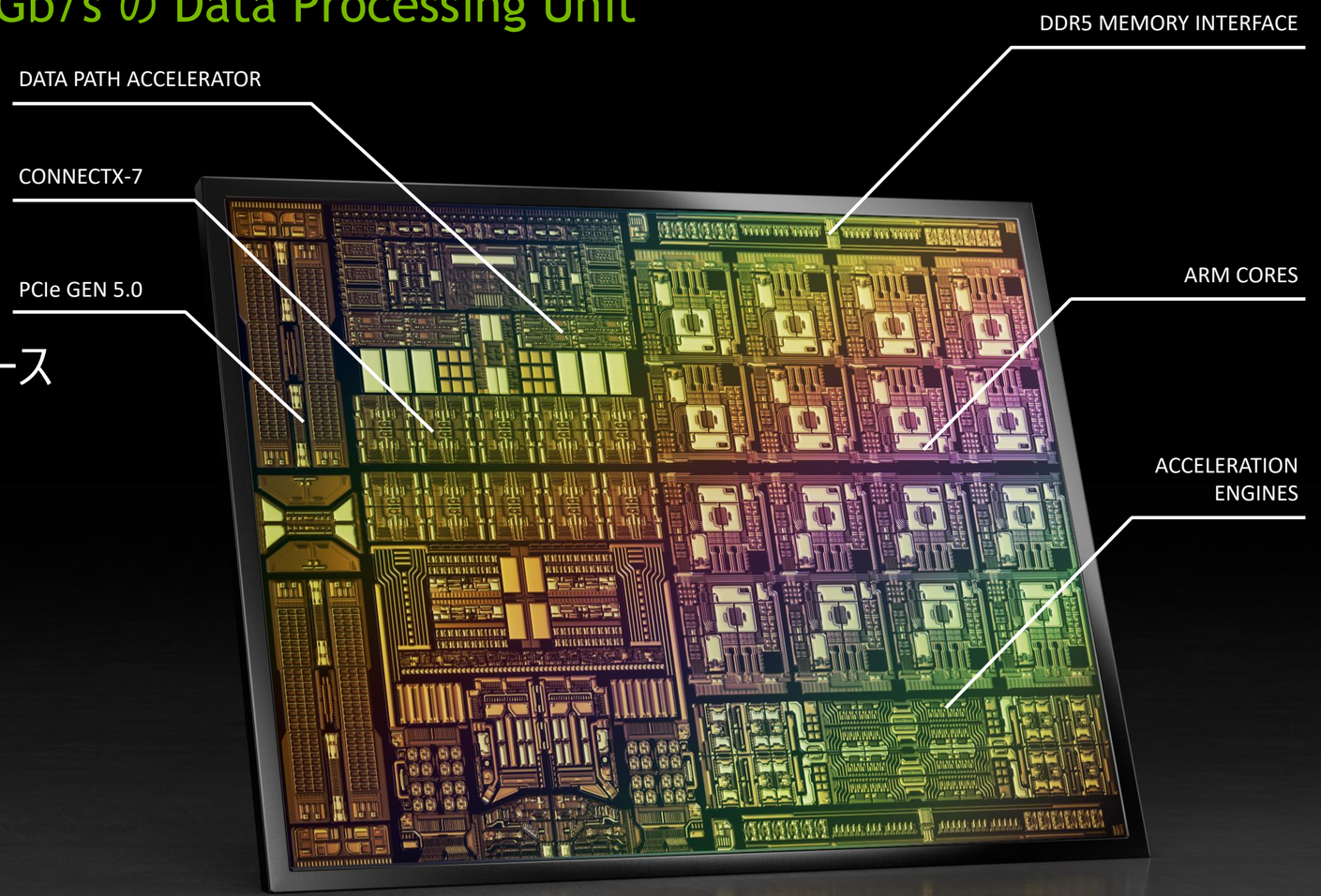
データセンター インフラストラクチャをチップにオフロード

- 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

NVIDIA BLUEFIELD-3 DPU

最速 400Gb/s の Data Processing Unit

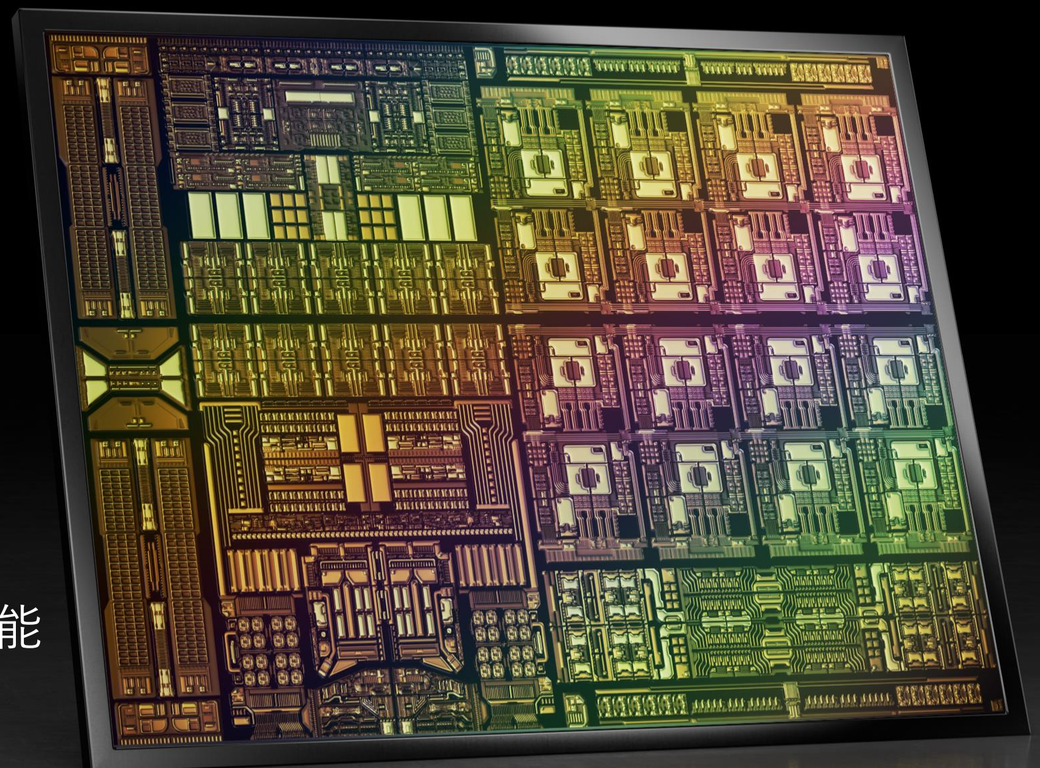
- 220億トランジスタ
- 400Gb/s のEthernet版 InfiniBand版をリリース
- 400Gb/s 暗号化アクセラレーション
- X86 Core 300個と同等
- 18M IOP/s ストレージ性能
- DDR5 Memory



NVIDIA BLUEFIELD-3 DPU

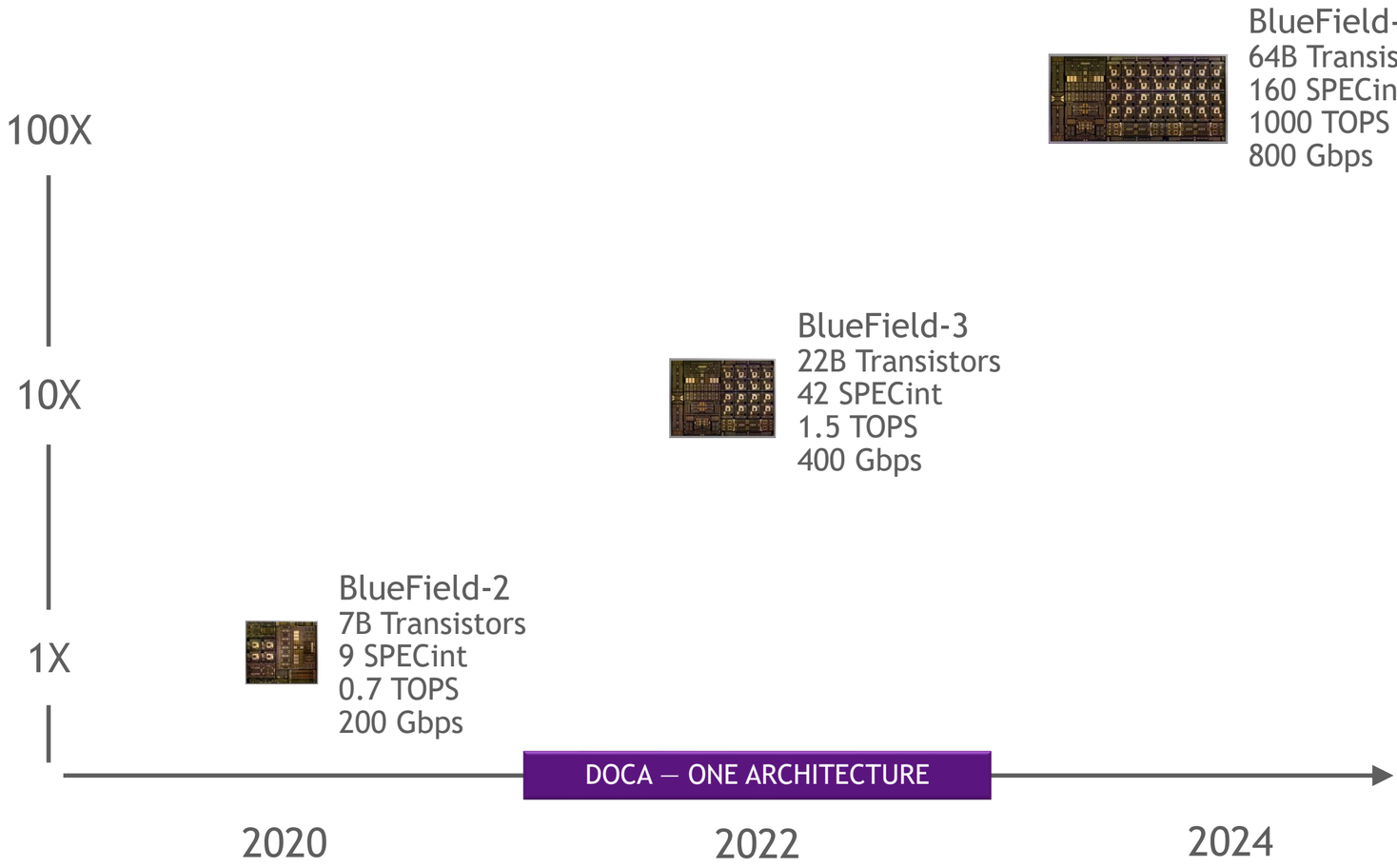
最速 400Gb/s の Data Processing Unit

- 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
- データパスアクセレラレータ – 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

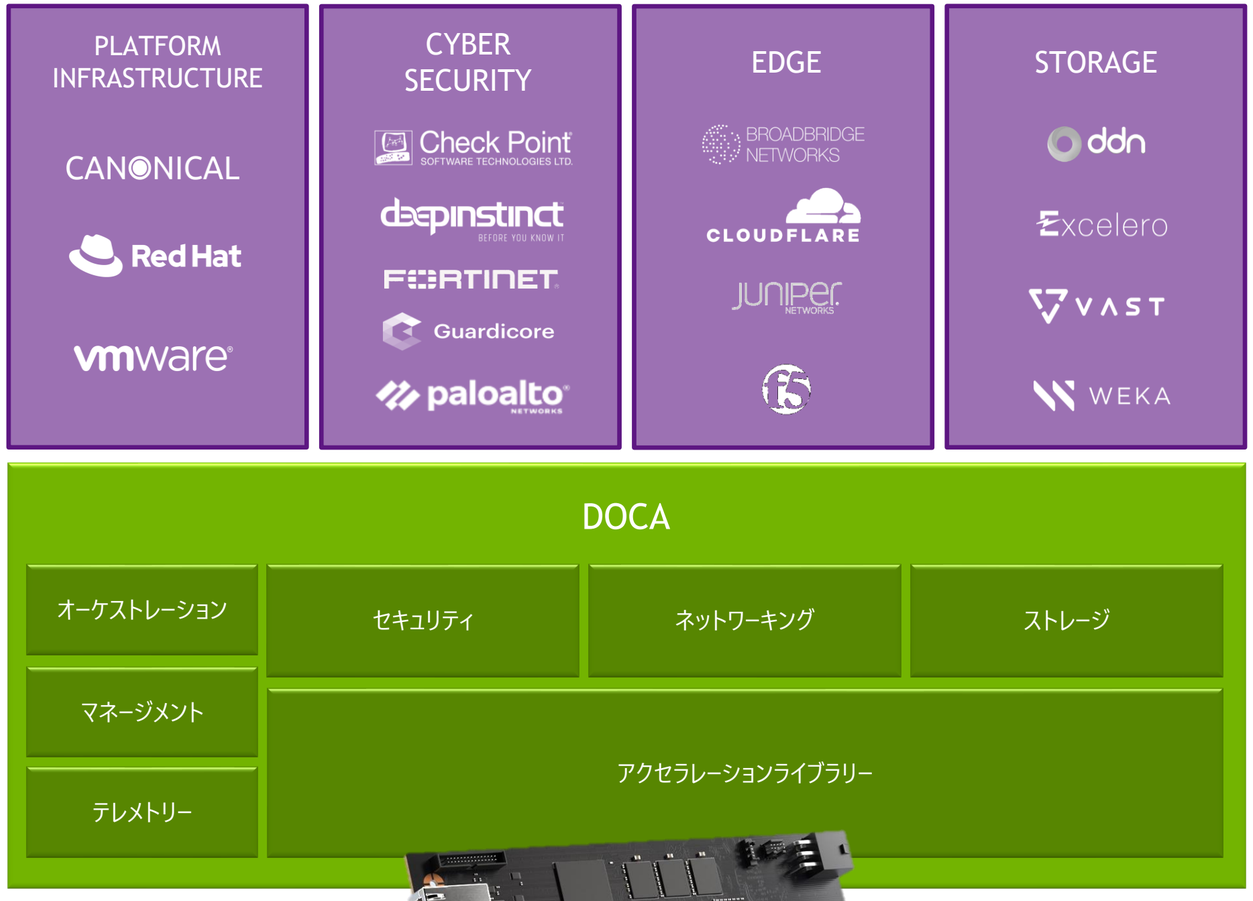
データセンターインフラ処理の飛躍的な成長

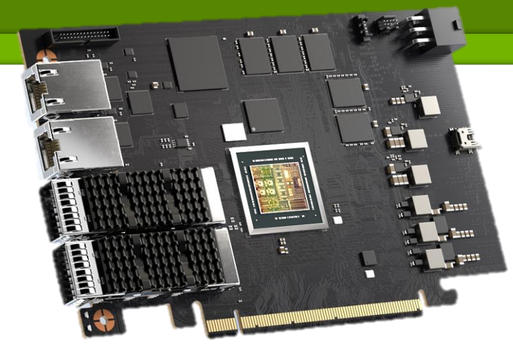


NVIDIA DOCA

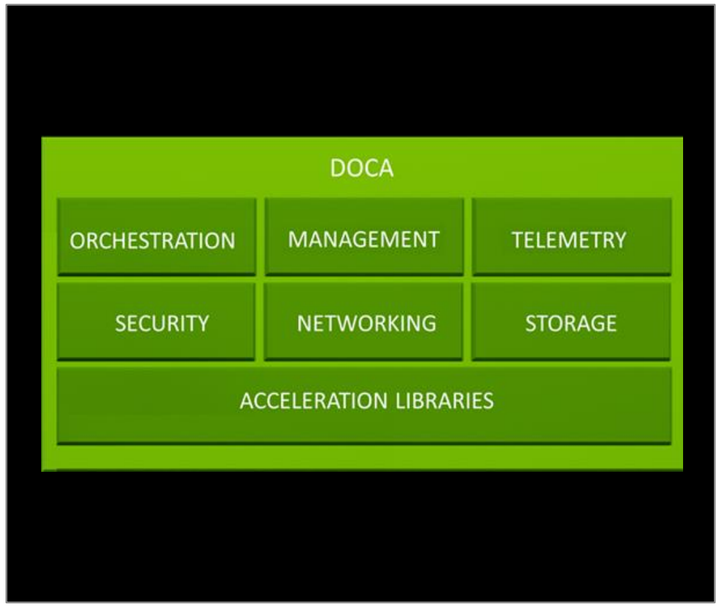
広範な DPU パートナーエコシステムの実現

- 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

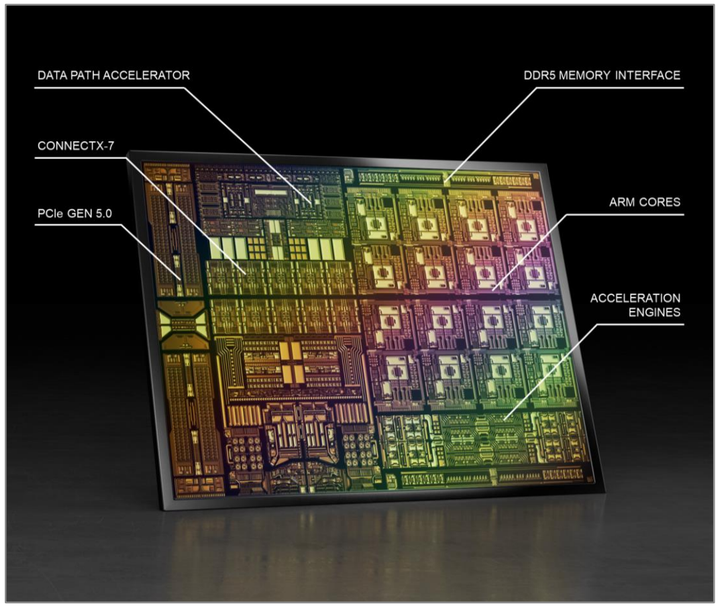




これからの HPC と AI を支えるネットワーク技術



DOCA のパートナーエコシステム

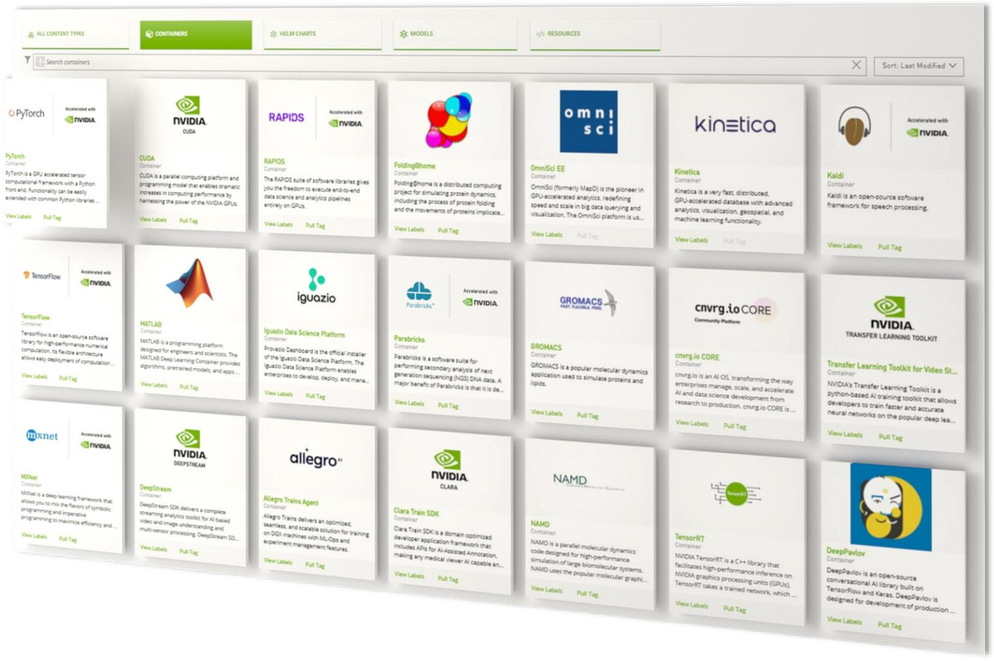


次世代の 400Gbps DPU - Bluefield-3

NVIDIA Quantum-2
NDR 400G InfiniBand

NGC

NGC コンテナが AI 開発を効率化



企業ユース向けの品質

セキュリティ検査

信頼性テスト

エンタープライズ サポート

パフォーマンス最適化

スケーラブル

月例更新

最適化による性能向上

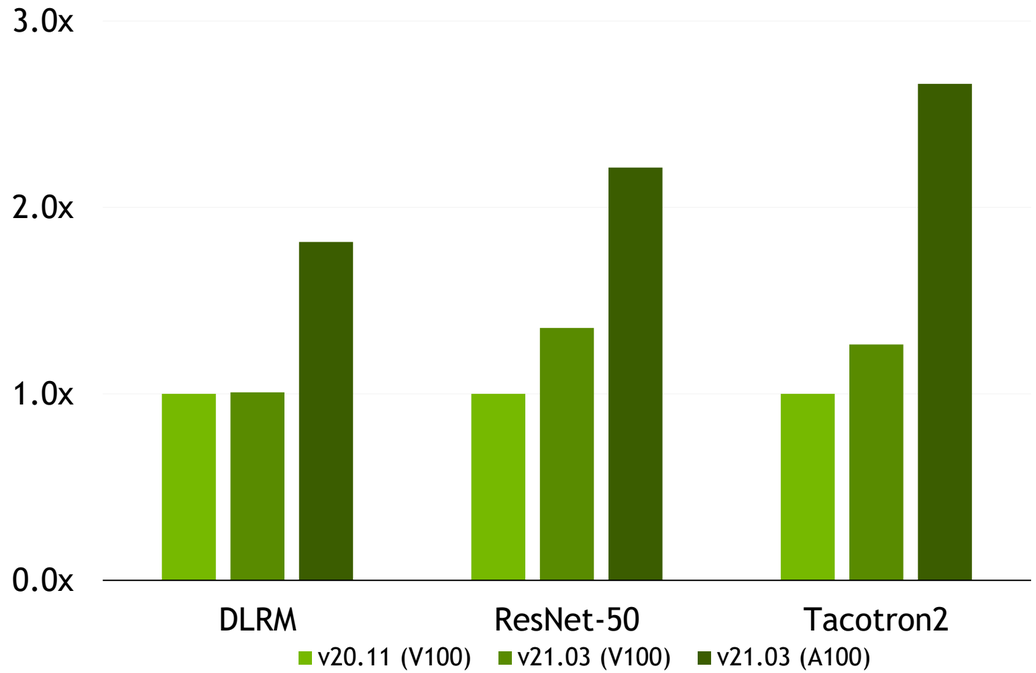
様々な環境で利用可能

Docker | cri-o | containerd | Singularity

ベアメタル、仮想マシン、Kubernetes

オンプレミス、クラウド、エッジ環境

常に最高の性能を



企業ユース向けの品質

セキュリティ検査

信頼性テスト

エンタープライズ サポート

パフォーマンス最適化

スケーラブル

月例更新

最適化による性能向上

様々な環境で利用可能

Docker | cri-o | containerd | Singularity

ベアメタル、仮想マシン、Kubernetes

オンプレミス、クラウド、エッジ環境

Tacotron2 Training with PyTorch container on 8x V100 (32GB) & A100 (40GB). Mixed Precision. Batch size: 104 (V100). 128 (A100). DLRM training with PyTorch on 1x V100 (32GB) & A100 (40GB). Mixed Precision. Batch size: 32,768. ResNet-50 v1.5 training with TensorFlow on 8x V100 (32GB) & A100 (40GB). Mixed Precision. Batch size: 256

NGC プライベート レジストリ

安全で迅速なコラボレーションを促進

Build and Secure
- コンテナイメージのセキュリティを確保するために既知の脆弱性に対する自動的なスキャンを実施

Share
- ユーザー自身が作成したコンテナイメージや Helm チャート、学習済みモデル、モデルスクリプトを組織内で安全に共有

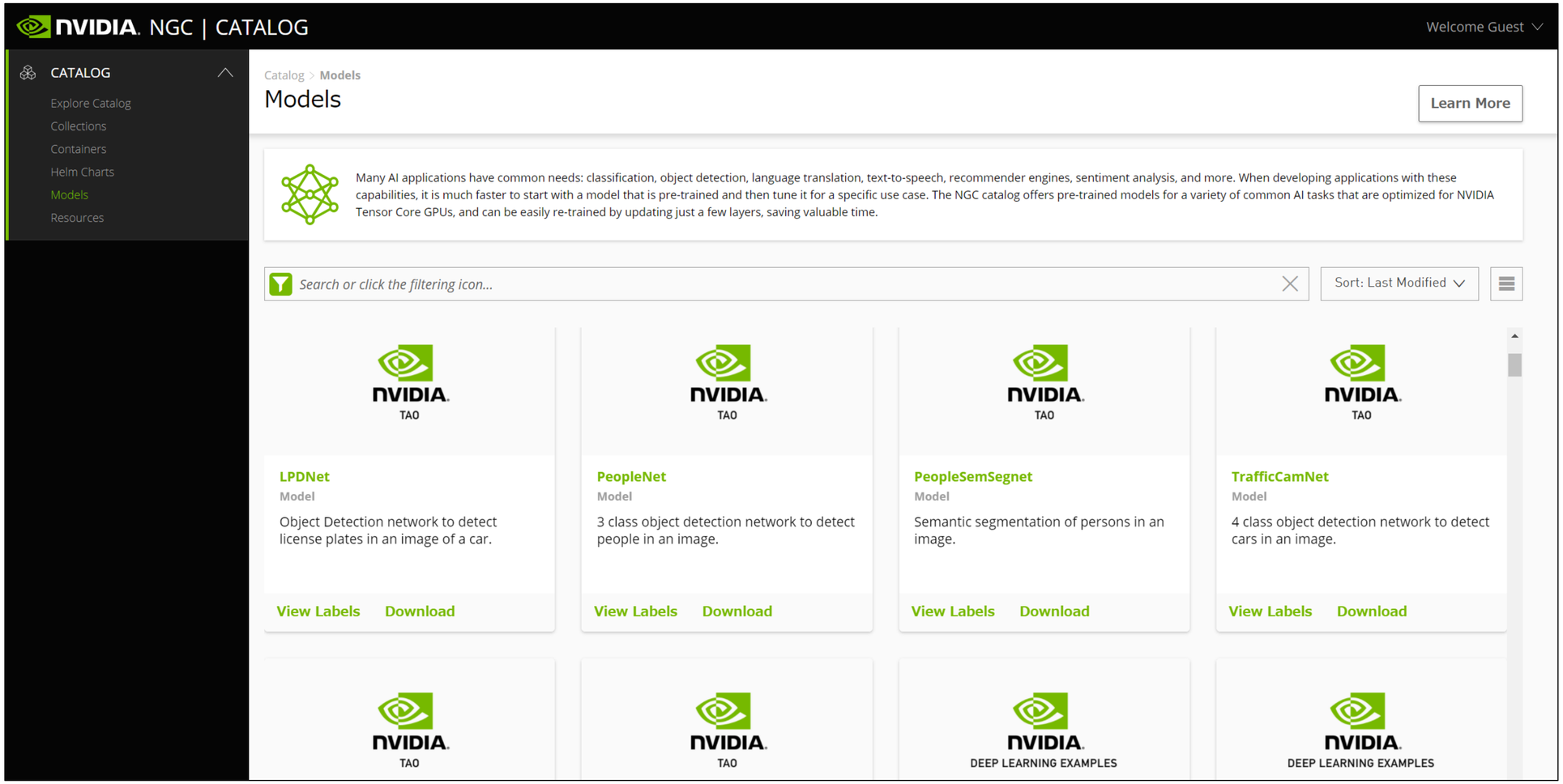
Manage
- モデルのバージョン管理が可能

Protect your IP
- 安全なストレージ、ユーザーとチームの管理、API キーの管理

プライベートレジストリのドキュメントはこちら

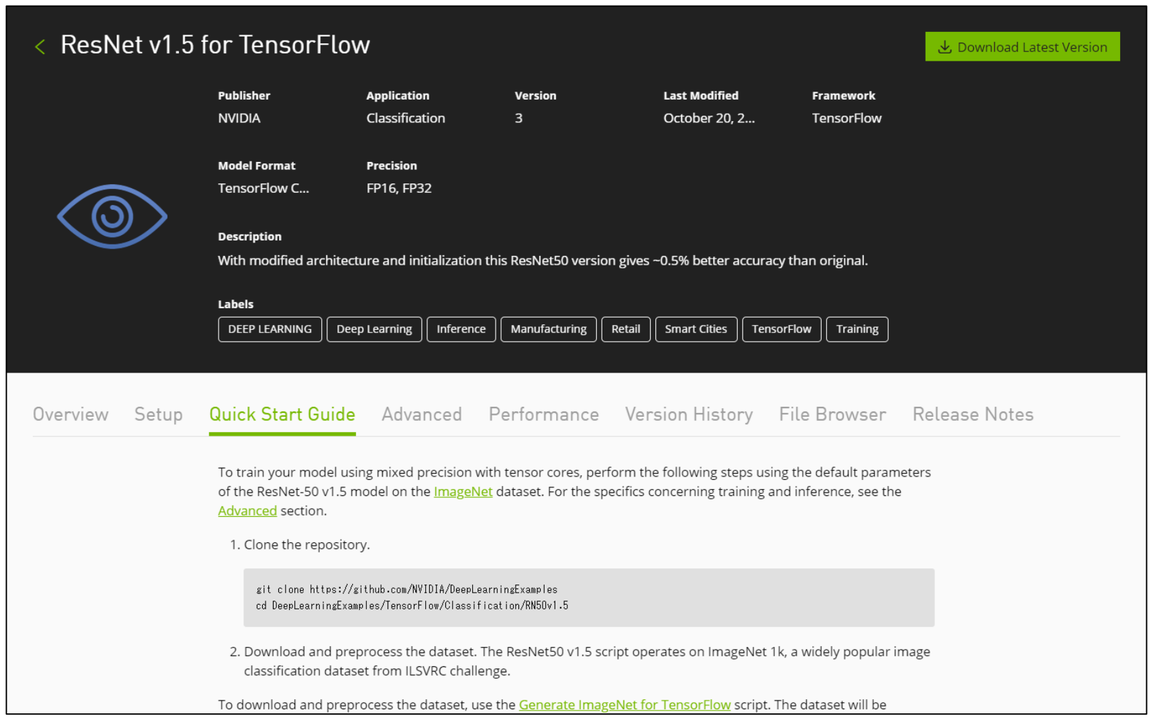
DGX, EGX, NGC サポートサービスのユーザー

トレーニング済みモデル



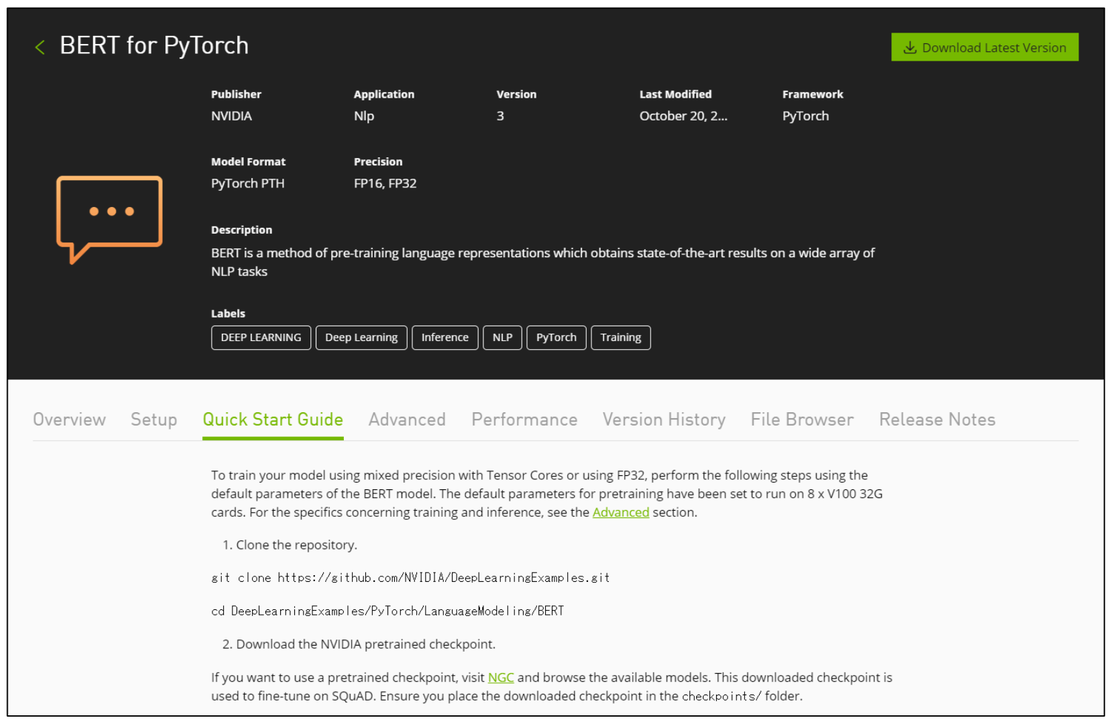
NGC のモデルスクリプトを活用

ResNet-50 for TensorFlow



https://ngc.nvidia.com/catalog/model-scripts/nvidia:resnet_50_v1_5_for_tensorflow/quickStartGuide

BERT for PyTorch



https://ngc.nvidia.com/catalog/model-scripts/nvidia:bert_for_pytorch/quickStartGuide

NVIDIA GTC

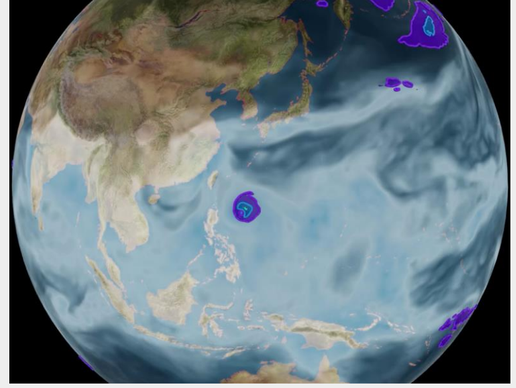
YOUR MOST BRILLIANT WORK STARTS HERE

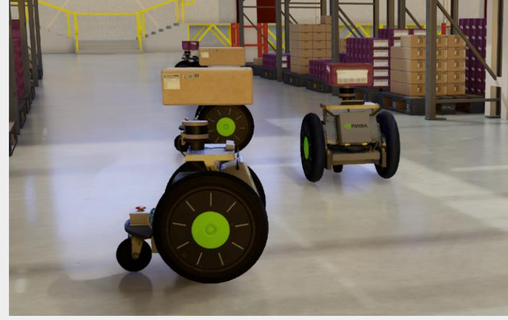
The Developer Conference for the Era of AI

NVIDIA GTC is more than a game-changing AI developer conference. It's a global community committed to decoding the world's greatest challenges, transforming every major industry workflow, and exploring tomorrow's next big ideas—together. Join us this March and discover how to accelerate your life's work.

MARCH 21—24, 2022 | www.nvidia.com/GTC

参加登録 (無料) はこちら





NVIDIA