

# LLMとGPUとネットワーク

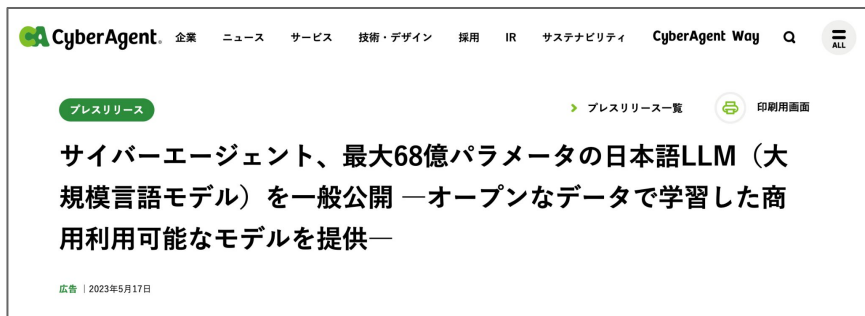
## MPLS Japan 2023

Yuya Kawakami  
Senior Network Architect, SoftBank

# はじめに

- この発表資料は公開情報をもとに作られています
  - 図を自分で書かずに引用してURLを明記しているのはこれを明示するためです
- 間違っている部分があれば遠慮なくお知らせください

# 続々と公開される日本語LLM



CyberAgent 企業 ニュース サービス 技術・デザイン 採用 IR サステナビリティ CyberAgent Way

プレスリリース プレスリリース一覧 印刷用画面

サイバーエージェント、最大68億パラメータの日本語LLM（大規模言語モデル）を一般公開 —オープンなデータで学習した商用利用可能なモデルを提供—

広告 | 2023年5月17日



LINE Engineering Blog Interview Culture Open Source Careers Research

## 36億パラメータの日本語言語モデルを公開しました

Shun Kiyono, Sho Takase, Toshinari Sato(overlast) 2023-08-14  
NLP Foundation Dev Team



松尾研究室 MATSUO LAB THE UNIVERSITY OF TOKYO ABOUT RESEARCH EDUCATION PUBLICATIONS STARTUPS JOIN US CONTACT

## 100億パラメータサイズ・日英2ヶ国語対応の大規模言語モデル“Weblab-10B”をオープンソースで公開しました。

2023年8月18日発表プレスリリース

### 東京大学松尾研究室 100億パラメータサイズ・日英2ヶ国語対応の 大規模言語モデル“Weblab-10B”を オープンソースで公開 —オープンソースの日本語大規模言語モデルで最高水準（注1）—

MENU  
ABOUT  
Missions & Activities  
Members  
RESEARCH  
Major Research Areas  
Web & Business  
Collaborative Research  
EDUCATION  
Overview



stability.ai Stability AIについて モデル API 開発者 ニュース English

## 日本語言語モデル「Japanese StableLM Alpha」をリリースしました

10 Aug

# 生成AIのためのGPU投資が加熱

ITmedia NEWS > 速報 > さくら、生成AI向けのクラウドサービス提供へ | N...

## さくら、生成AI向けのクラウドサービス提供へ「NVIDIA H100」を2000基投入

© 2023年06月16日 12時39分 公開 [ITmedia]

印刷 見る Share B! 26 0

ドメイン駆動設計 | ドメインのモデル設計を手を動かさず任す  
重さ約1.08kgで最大23.5時間駆動!! 高性能モバイルPC「MousePro G4」

さくらインターネットは6月16日、生成AI向けのクラウドサービスを、2024年1月以降に提供すると発表した。同社が北海道で運営する「石狩データセンター」に、米NVIDIAのGPU「NVIDIA H100 Tensor コア GPU」を2000基以上導入。生成AIに必要なコンピューティングリソースを提供していくという。



石狩データセンター

<https://www.itmedia.co.jp/news/articles/2306/16/news117.htm>

## 【独自】政府がソフトバンクの生成AI向けに補助金約50億円

7/6(木) 17:08 配信 41 返信 共有

テレ東BIZ



政府は、ソフトバンクが生成AIを開発するのに必要なスーパーコンピューター事業に対し、補助金約53億円を交付する方針を固めた。明日にも発表する。

ソフトバンクは、独自の生成AIの開発に向けてスパコンを含むクラウドシステムを増強する。生成AIの開発には高い計算処理能力をもつAI向けのスパコンの整備・確保が欠かせないとされ、経済産業省は6月のさくらインターネットへの支援決定に続き、“AIスパコン”への支援を強化していく。



ソフトバンクの宮川潤一社長は5月の決算会見時に独自の生成AI開発を表明しており、ソフトバンクグループの孫正義会長兼社長も4日、「(生成AI開発に向け)あらゆることをやる」と述べていた。ソフトバンクの事業計画では、米・半導体企業のエヌビディアからAI開発に適した半導体GPU(画像処理半導体)を2240基調達するとしていて、これにより国内でトップクラスのAIスパコンを整備することになる。

総事業費は約200億円で、経済産業省はそのうち約160億


<https://news.yahoo.co.jp/articles/f83edd682bfd4b15bcf8e015b6cc84587223758>

# LLM (Large Language Model / 大規模言語モデル)

- 大規模なデータセットとディープラーニングを用いることで精度を高めた自然言語処理 (NLP) のモデル

Default (GPT-3.5)

OP LLMとはなんですか？

 LLMは、「Language Model」の略称で、自然言語処理 (NLP) に関連する技術の一部を指す用語です。自然言語処理は、人間の言語をコンピュータが理解し、生成するための技術分野であり、LLMはその中でも特に重要な役割を果たすモデルの一種です。

LLMは、大量のテキストデータを学習して、言語の文法、文脈、意味などを理解し、テキスト生成や理解のタスクを行うためのモデルです。GPT (Generative Pre-trained Transformer) などのモデルは、LLMの一例です。これらのモデルは、文章の生成、質問応答、文章の分類、要約、翻訳など、さまざまなタスクに使用されます。

GPT-3 (GPTの第3バージョン) は、特に高い表現力と柔軟性を持つLLMの一例であり、多くの人々が驚くほどの自然な文章を生成したり、多くのNLPタスクで優れた成績を収めたりすることができます。

# 最近の主要な世界のLLM

モデル	開発組織	公開年月	パラメータ数	学習データ	モデルの公開先
<a href="#">GPT-3</a>	OpenAI	2020年6月	175B	570GB	-
GPT-3.5	OpenAI	2022年3月	355B	N/A	-
<a href="#">GPT-4</a>	OpenAI	2023年3月	非公開	N/A (推定825TB)	-
<a href="#">PaLM 2</a>	Google	2023年5月	非公開(推定340B)	-	-
<a href="#">LLaMA2</a>	Meta	2023年7月	7B/13B/70B	-	<a href="https://huggingface.co/meta-llama">https://huggingface.co/meta-llama</a>

# 最近の主要な日本語特化型LLM

LLM-jpがまとめを作っています <https://github.com/llm-jp/awesome-japanese-llm>

モデル	開発組織	公開年月	パラメータ数	学習データ	モデルの公開先
<a href="#">Rinna-3.6B</a>	Rinna	2023年5月	3.6B	-	<a href="https://huggingface.co/rinna/japanese-gpt-neox-3.6b">https://huggingface.co/rinna/japanese-gpt-neox-3.6b</a>
<a href="#">OpenCALM</a>	CyberAgent	2023年5月	7B	-	<a href="https://huggingface.co/cyberagent/open-calm-7b">https://huggingface.co/cyberagent/open-calm-7b</a>
<a href="#">japanese-large-lm</a>	LINE	2023年8月	1.7B/3.6B	650GB	<a href="https://huggingface.co/line-corporation/japanese-large-lm-3.6b">https://huggingface.co/line-corporation/japanese-large-lm-3.6b</a>
<a href="#">Japanese StableLM Alpha</a>	Stability AI	2023年8月	7B	750B Token	<a href="https://huggingface.co/stabilityai/japanese-stablelm-instruct-alpha-7b">https://huggingface.co/stabilityai/japanese-stablelm-instruct-alpha-7b</a>
<a href="#">Weblab-10B</a>	東京大学 松尾研究室	2023年8月	10B	-	<a href="https://huggingface.co/matsuo-lab/weblab-10b">https://huggingface.co/matsuo-lab/weblab-10b</a>
<a href="#">PLaMo-13B</a>	Preferred Networks	2023年9月	13B		<a href="https://huggingface.co/pfnnet/plamo-13b">https://huggingface.co/pfnnet/plamo-13b</a>
<a href="#">LLM-jp-13B</a>	LLM-jp (NII)	2023年10月	13B	約300B億 Token	<a href="https://huggingface.co/llm-jp/llm-jp-13b-v1.0">https://huggingface.co/llm-jp/llm-jp-13b-v1.0</a>

# LLMの学習にかかる時間

LINE社のjapanese-large-lmの事例では1.7Bのモデルの構築のために  
NVIDIA社のA100 80GB換算で4000GPU時間を使用

➡ さらに大規模なモデルの構築にはとてつもないGPU時間とメモリが必要

最終的な学習には約650GBのコーパスを利用していますが、英語の大規模コーパスとして一般的に用いられているもの（Pileコーパス）が約800GBであることを踏まえると、我々のデータも遜色ない大きさであると言えます。

本モデルの構築に要した時間について、例えば1.7BモデルについてはA100 80GBで換算し、約4000GPU時間を費やしています。

<https://engineering.linecorp.com/ja/blog/3.6-billion-parameter-japanese-language-model>



# LLMの学習にかかる時間とコスト(Google Cloud)

180Bのモデルの構築をクラウドで行うとH100で55日と24.2億円がかかる

モデルサイズ	トークンサイズ	想定マシンリソースと学習期間	予算例
7B	2B	TPU v5e - 38 Days	\$ 290k
		TPU v4 - 1 Day	\$ 460k
		A100 (40G) - 6 Days	\$ 550k
		H100 - 2 Day	\$ 629k
13B	2B	TPU v5e - 71 Days	\$ 525k
		TPU v4 - 3 Day	\$ 845k
		A100 (40G) - 6 Days	\$ 550k
		H100 - 2 Day	\$ 629k
40B	2B	TPU v5e - 38 Days	\$ 290k
		TPU v4 - 1 Day	\$ 460k
		A100 (40G) - 11 Days	\$ 1M
		H100 - 4 Days	\$ 1.1M
70B	2B	TPU v5e - 384 Days	\$ 2.9M
		TPU v4 - 14 Days	\$ 4.6M
		A100 (40G) - 61 Days	\$ 5.5M
		H100 - 21 Days	\$ 6.2M
180B	2B	TPU v5e - 986 Days	\$ 7.2M
		TPU v4 - 37 Days	\$ 11M
		A100 (40G) - 157 Days	\$ 14.1M
		H100 - 55 Days	\$ 16.1M

435M JPY = 4.35億円

690M JPY = 6.90億円

825M JPY = 8.25億円

930M JPY = 9.30億円

1,080M JPY = 10.8億円

1,650M JPY = 16.5億円

2,115M JPY = 21.2億円

2,415M JPY = 24.2億円

<https://twitter.com/myoshimu/status/1714459784446140723>


※Google Cloud Japan でAI Consultantをしている方の個人的見解

※ 40BのTPU v5e/v4の値は7Bと同じ値になっておりコピペ&編集ミスだと思われる

LLMとGPUとネットワーク / @yuyarin / 2023-10-26 / MPLS Japan 2023

# LLMの学習にかかる時間(Meta)

LLaMA2 70Bの学習には2,000基のA100で1.7M GPU時間(36日)を使用

Recent LLM models from Meta 

MODEL NAME	RELEASE DATE	MODEL SIZE	DATASET SIZE	TRAINING ZETA (IE21) FLOPS	TRAINING HW (COMPUTE)	TRAINING HW (NETWORK)	GPU HOURS (# GPUS X HOURS)
OPT	May 2022	175 B	300 B	430	1K A100	IB 200Gbps per GPU 25.6 TB/s bisection BW	800K
LLaMA	Feb 2023	65 B	1.4 T	600	2K A100	IB 200Gbps per GPU 51.2 TB/s bisection BW	1M
LLaMA2	July 2023	34 B	2 T	400	2K A100	RoCE 200Gbps per GPU 51.2 TB/s bisection BW	1M
LLaMA2	July 2023	70 B	2 T	800	2K A100	IB 200Gbps per GPU 51.2 TB/s bisection BW	1.7M

[LLaMA2-34B](#) is the first large language foundational model trained using RoCE @2K scale

<https://atscaleconference.com/videos/networking-for-genai-training-and-inference-clusters/>

# LLMの学習にかかる時間を短縮するには？

- 利用するGPUの枚数を筐体内で増やして分散・並列化する
- 利用する筐体の台数を増やして分散・並列化する



**筐体内のGPU間の通信や筐体間の通信がボトルネックになる**



**いかにしてGPUの内部バスに近い性能の通信を実現するかが課題**

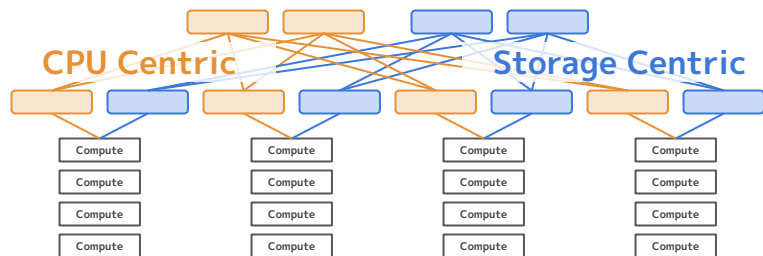
※この分野はNVIDIAの独壇場なのでNVIDIAの宣伝っぽい感じになってしまいますが  
最先端で何が起きているのかを知るためにも紹介します

# AI/MLのためのネットワーク

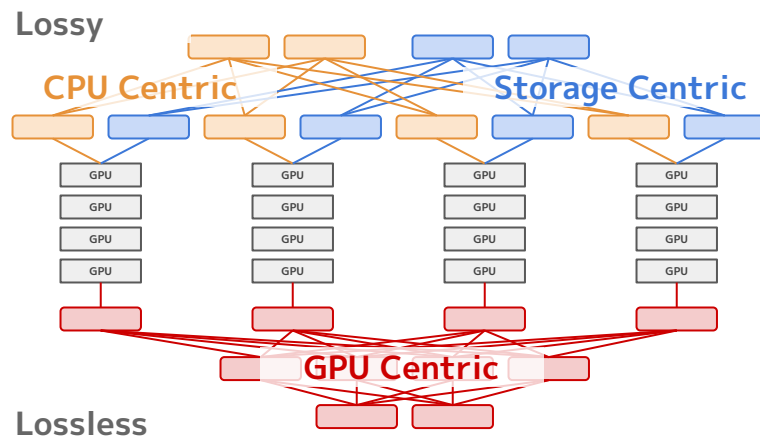
- 学習のパフォーマンスが生命線

➡ GPUのための超広帯域、低遅延、ロスレスなネットワークが必要

いままで



これから



# GPUとGPU通信の高速化

# NVIDIA データセンター向けGPU (NVIDIA Tesla)

## 各アーキテクチャのハイエンドモデルのスペック

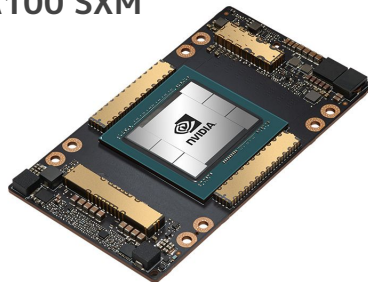
※メモリ帯域幅や消費電力はSXMフォームファクタ

アーキテクチャー	モデル	発表	CUDA コア数	Tensor コア数	メモリ 容量	メモリ 帯域幅	NVLink 帯域幅	PCIe 帯域幅	消費電力
Pascal	P100	2016/4	3,584	-	16 GB	732 GB/s	160 GB/s	32 GB/s	250 W
Volta	V100	2017/5	5,120	640	32 GB	900 GB/s	300 GB/s	32 GB/s	300 W
Turing	T4	2018/9	2,560	320	16 GB	320 GB/s	-	16 GB/s	70 W
Ampere	A100	2020/5	6,912	432	80 GB	2,039 GB/s	600 GB/s	64 GB/s	400 W
Hopper	H100	2022/10	14,592	456	80 GB	3,350 GB/s	900 GB/s	128 GB/s	700 W

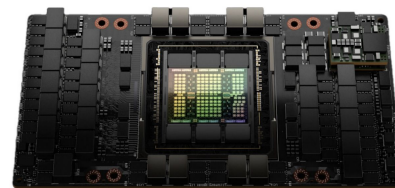
V100 SXM



A100 SXM



H100 SXM



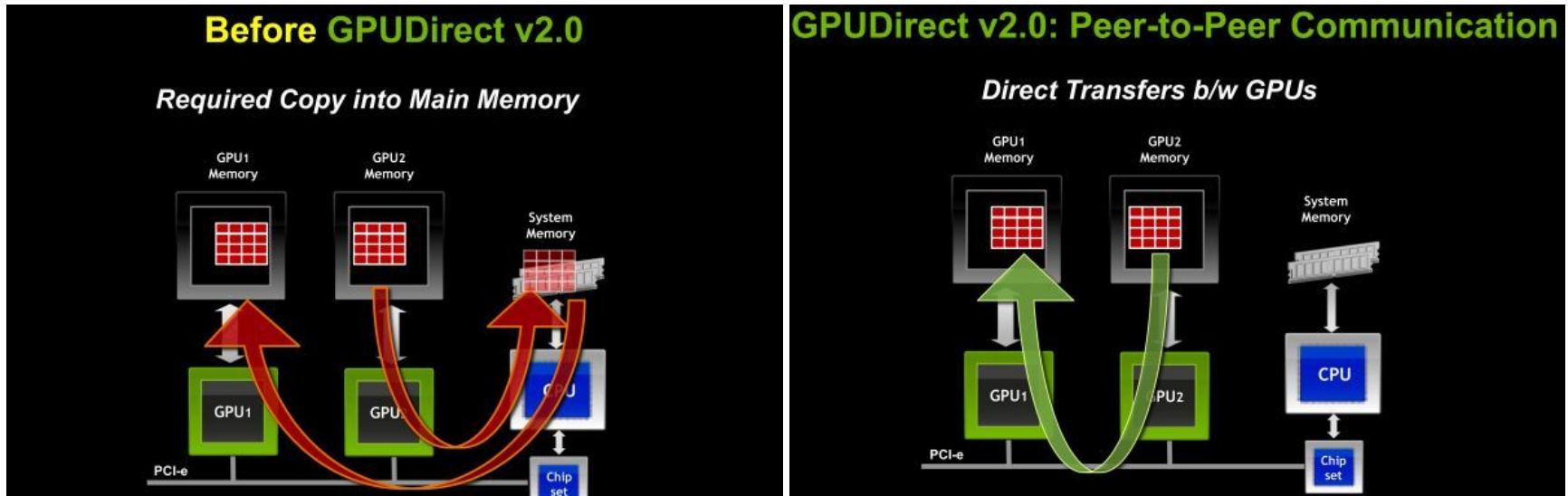
<https://www.nvidia.com/ja-jp/data-center/v100/>

<https://www.nvidia.com/ja-jp/data-center/a100/>

<https://www.nvidia.com/ja-jp/data-center/h100/>

# NVIDIA GPU Direct v2 - Peer-to-Peer

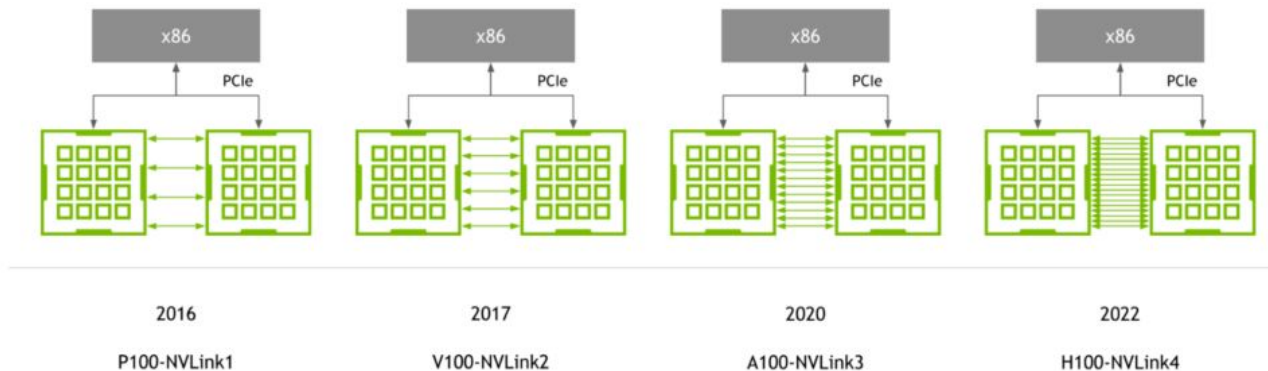
- 通常はCPUと共有メモリを介してデータのやりとりを行うため遅い
- PCIeバス経由でGPUメモリ同士で直接データを転送することで高速化



<https://www.anandtech.com/show/4198/nvidia-announces-cuda-40>

# NVIDIA NVLink

- GPU同士で直接通信するインターコネクットの独自規格
  - 通信規格やプロトコルの詳細は不明
- PCIeバスよりも一桁高速
  - A100のNVLink3は600GB/s (4.8Tbps)
  - H100のNVLink4は900GB/s (7.2Tbps)
- リンクあたり50GB/sでリンク数を増やしている

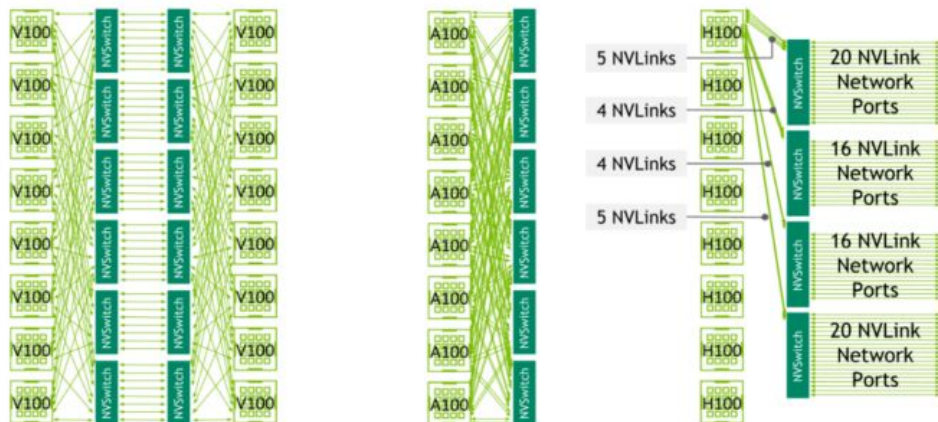


<https://developer.nvidia.com/blog/upgrading-multi-gpu-interconnectivity-with-the-third-generation-nvidia-nvswitch/>



# NVIDIA NVSwitch

- 2つ以上のGPUを相互接続するためのNVLinkのスイッチ
- 筐体内部で複数GPUを搭載するDGXシリーズで利用される



2017

DGX-2 (V100)

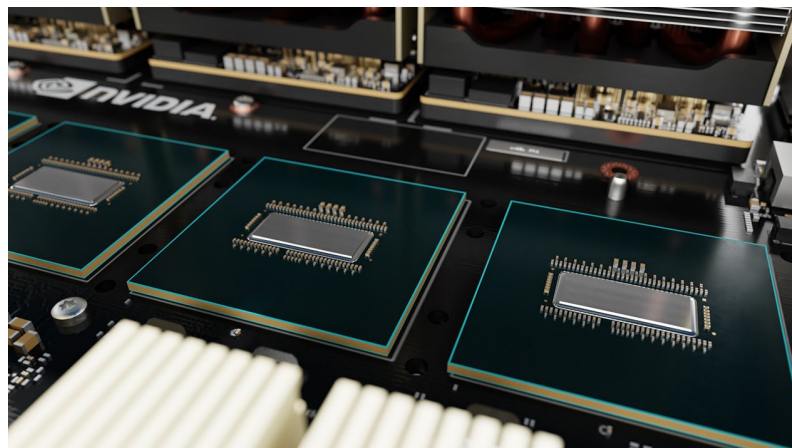
2020

DGX A100

2022

DGX H100

## DGX A100用 NVSwitch (第2世代)

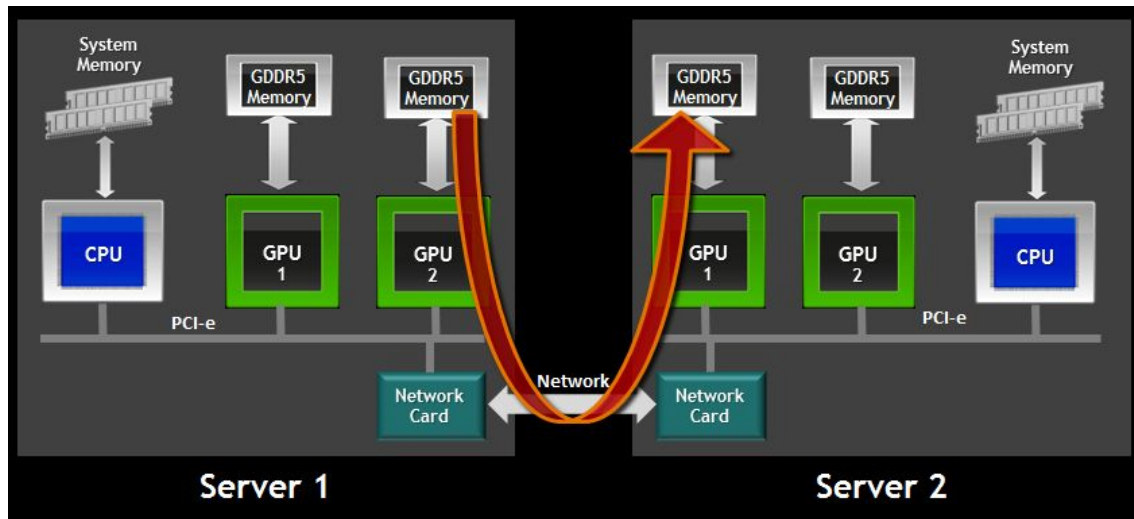


<https://www.nvidia.com/ja-jp/data-center/dgx-a100/>

<https://developer.nvidia.com/blog/upgrading-multi-gpu-interconnectivity-with-the-third-generation-nvidia-nvswitch/>

# NVIDIA GPU Direct v3 - RDMA

- RDMA = Remote Direct Memory Access
- ネットワークを介してリモートホストのGPUのメモリに直接データを送る
- RDMAのために使用される通信規格がInfiniBand



<https://keeneland.gatech.edu/software/gpudirect.html>

# InfiniBandのGPUクラスタ

# InfiniBand (IB)

- HPCやストレージで使われている高信頼性・高可用性の通信技術
- InfiniBand Trade Associationで規格化
- AI/MLではGPU間のRDMAおよび高速ストレージアクセスを実現するために使用される

世代		レーンあたり実効帯域	4レーン(4X)実効帯域	12レーン(12X)実効帯域
SDR	Single Data Rate	2Gbps	8Gbps	24Gbps
DDR	Double Data Rate	4Gbps	16Gbps	48Gbps
QDR	Quad Data Rate	8Gbps	32Gbps	96Gbps
FDR	Fourteen Data Rate	14Gbps	56Gbps	168Gbps
EDR	Enhanced Data Rate	25Gbps	100Gbps	300Gbps
HDR	High Data Rate	50Gbps	<b>200Gbps</b>	600Gbps
NDR	Next Data Rate	100Gbps	<b>400Gbps</b>	1200Gbps
XDR	eXtended Data Rate	200Gbps	800Gbps	2400Gbps

# InfiniBandの特徴

- 広帯域
  - 内部バスを外部に延長するコンセプト
- ロスレス
  - クレジットベースのフローコントロールにより輻輳制御
- 低レイテンシー
  - HCA(ConnectX-6): 600ns
  - スイッチ(QM8700): 90ns
- ノンブロッキング
  - HoL Blockingを防ぐため物理リンク毎に最大16個のバーチャルレーン(VL)がありそれぞれのVLでキューイングが行われる
  - データ用に15個のVL、管理用に1個のVL(VL15)

# NVIDIA DGX

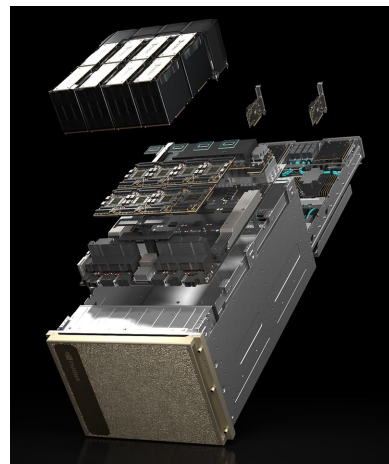
- NVIDIAが提供するGPUサーバアプライアンス
- CPU、OS (DGX OS)、8-GPU HGX、ConnectXがバンドルされている
- A100 80GBが8枚のDGX A100と、H100が8枚のDGX H100がある

DGX A100



<https://www.nvidia.com/ja-jp/data-center/dgx-a100/>

DGX H100



<https://www.nvidia.com/ja-jp/data-center/dgx-h100/>

# NVIDIA DGX SuperPOD

- NVIDIAが提供するDGXのGPUサーバクラスタ
- ネットワーク、ストレージ、管理系まで全部一式で提供される
  - ノード: DGX A100/H100, ストレージ(任意), UFM(IBSM), Mgmt(管理ノード)
  - NW: Compute(IB), Storage(IB), In-Band-Management(Eth), Out-of-Management(Eth)
- 現行はほぼA100。最近H100の導入報道が出てきた
  - [DeepL、ヨーロッパで最大規模のNVIDIA DGX H100 SuperPODを導入](https://www.nvidia.com/ja-jp/data-center/dgx-basepod/) (2023年8月2日)

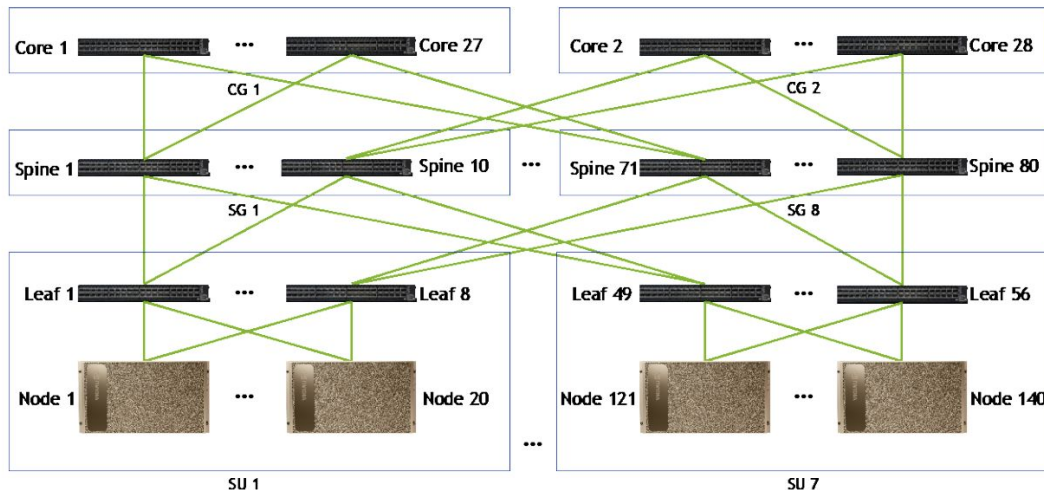


<https://www.nvidia.com/ja-jp/data-center/dgx-basepod/>

# NVIDIA DGX SuperPOD A100のアーキテクチャ

- 20台のDGX A100をSU(Scalable Unit)として140台(7SU)までサポート
- [Reference Architectureの資料](#)にだいたい書いてある
- InfiniBand HDR (200Gbps)のFabric

Figure 4. Compute fabric topology for a 140-node DGX SuperPOD



<https://images.nvidia.com/aem-dam/Solutions/Data-Center/gated-resources/nvidia-dgx-superpod-a100.pdf>

Figure 7. Storage fabric topology for 140-node system

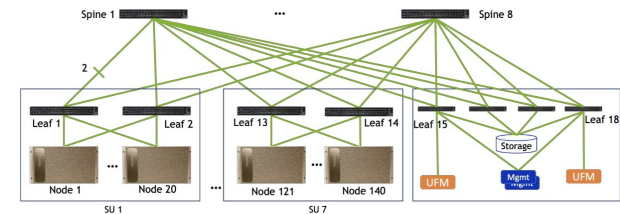
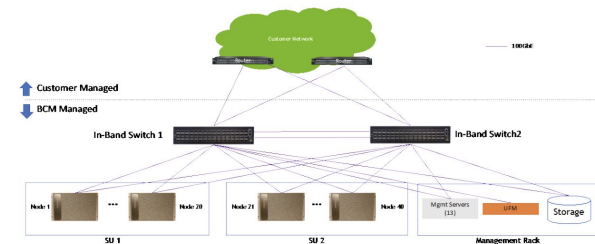


Figure 8. In-band Ethernet topology for one or two SUs





# NVIDIA DGX SuperPOD H100のアーキテクチャ

- 32台のDGX H100をSU(Scalable Unit)として2048台(64SU)までサポート
- [Reference Architectureの資料](https://docs.nvidia.com/dgx-superpod-reference-architecture-dgx-h100.pdf)にだいたい書いてある(2023年9月公開)
- InfiniBand NDR (400Gbps)のFabric

Figure 5. Compute fabric for full 127-node DGX SuperPOD

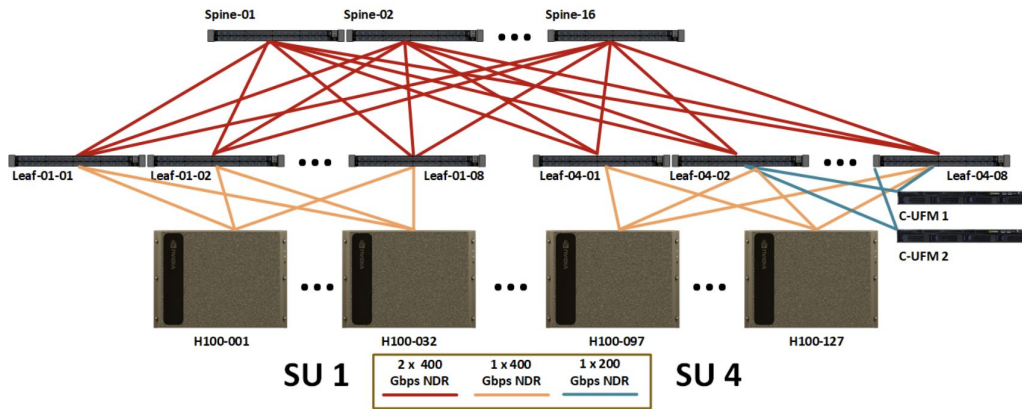


Figure 6. Storage fabric logical design

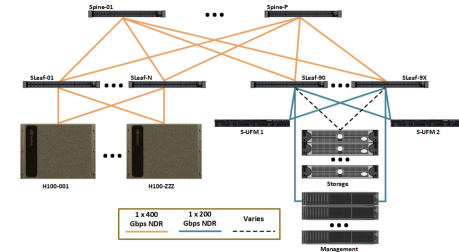
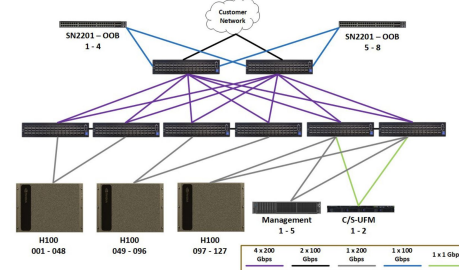


Figure 8. In-band Ethernet network



<https://docs.nvidia.com/dgx-superpod-reference-architecture-dgx-h100.pdf>

# NVIDIA DGX SuperPOD A100のCompute Fabric

- GPU間でRDMAの通信を行うFabric
- Fat-Treeトポロジーで構築されているが、いつものやつと何かが違う
  - 1つのノードが8台のLeafに接続する
  - 7SU (140ノード)のときのスイッチの台数はLeafよりSpineが多くなる！

## ➡ Rail-optimized TopologyとFull Bisection Bandwidthがキーワード

Figure 4. Compute fabric topology for a 140-node DGX SuperPOD

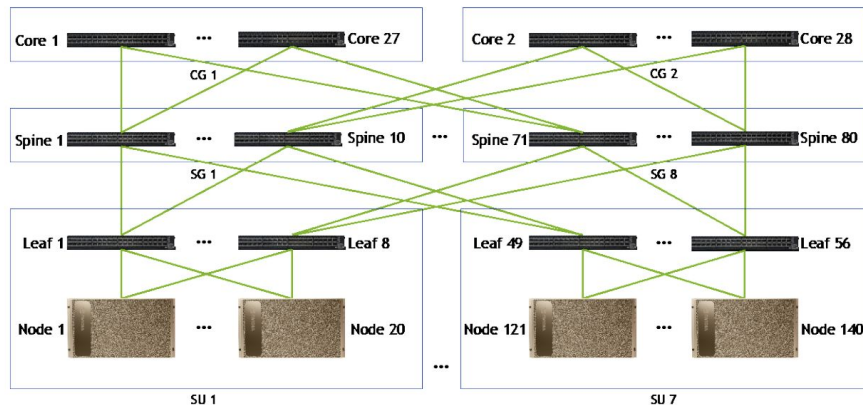


Table 3. Compute fabric switch and cable counts

Nodes	SUs	QM8790 Switches			Cables		
		Leaf	Spine	Core	Leaf	Spine <sup>1</sup>	Core
20 (Single SU)	1	8	5		160	164	
40	2	16	10		320	324	
60	3	24	20		480	484	
80	4	32	20		640	644	
120	6	48	80	24	960	964	960
140 (DGX SuperPOD)	7	56	80	28	1120	1124	1120

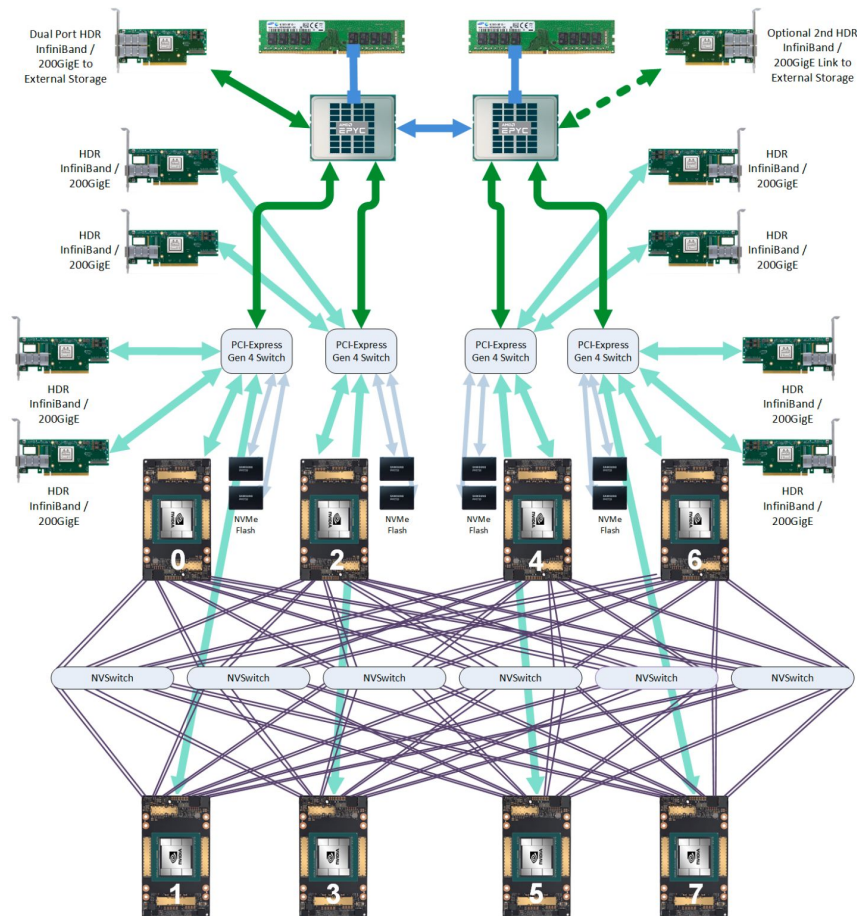
1. UFM Appliance is connected to two different spine switches.

The compute fabric uses NVIDIA Quantum QM8790 switches (Figure 6).

<https://images.nvidia.com/aem-dam/Solutions/Data-Center/gated-resources/nvidia-dgx-superpod-a100.pdf>

# NVIDIA DGX A100 の内部構成

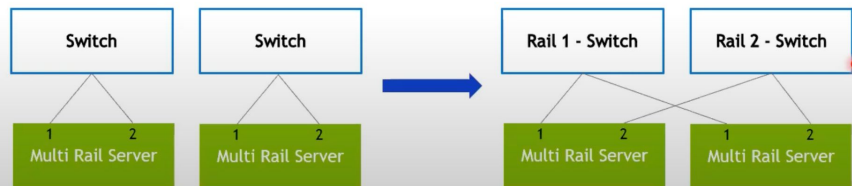
- A100 80GB GPU 8枚に対してそれぞれ200G IB NICが1枚割り当てられている
  - GPUとNICの間はPCIe Gen 4で接続
  - GPU間はNVSwitchで接続



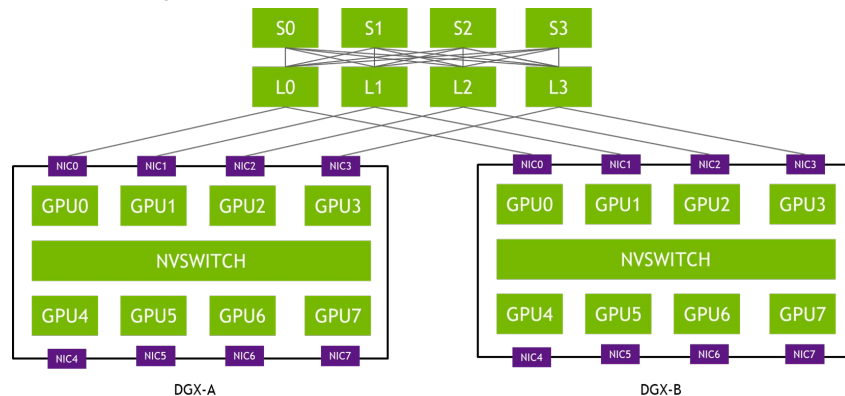
<https://www.microway.com/hpc-tech-tips/dgx-a100-review-throughput-and-hardware-summary/>

# Rail-optimized Topology

- 複数のノードの同じ番号のNICを同じLeafに収容するトポロジー
  - NCCLは複数のノードでAll-reduce演算を行うときに同じGPU番号を使用する
  - パフォーマンスを最大化するためにはGPUの数に対応したNICでMulti-rail構成を取る
  - Multi-rail構成においてRDMAを行うときにはGPUから一番近いNICを使う
  - クラスタサイズを最大化するには**サーバに搭載するGPUの枚数だけLeafスイッチを用意する**
- ToR構成が取りづらいので物理配線が地獄
  - AOCだとパッチパネルも使えなくなるのもっと地獄



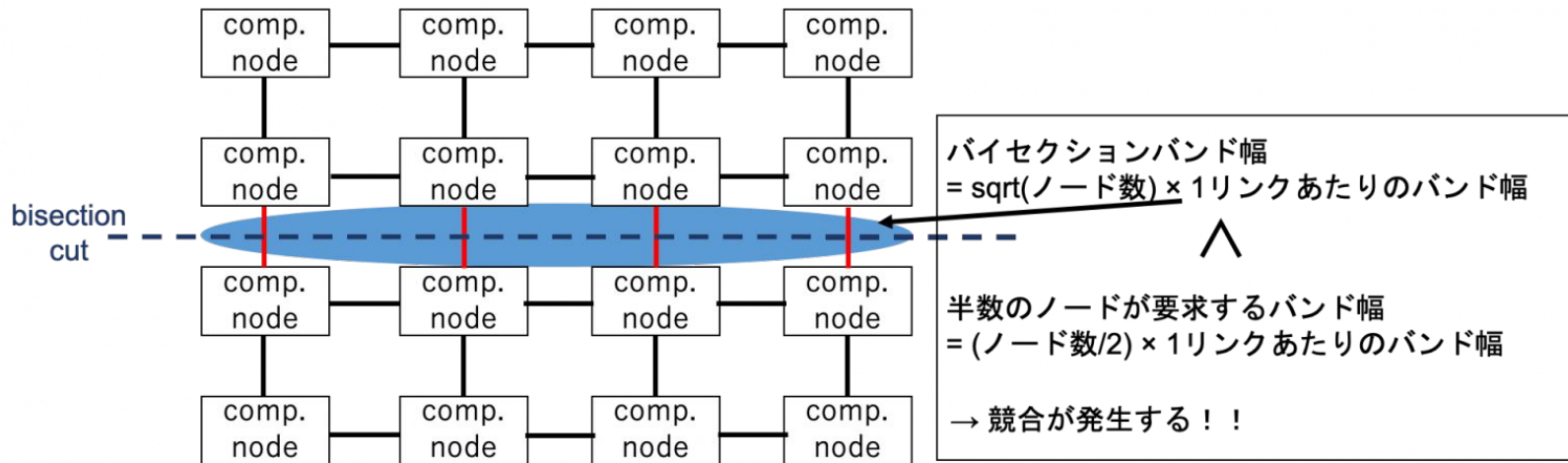
<https://docs.coreweave.com/networking/hpc-interconnect>



<https://developer.nvidia.com/blog/doubling-all2all-performance-with-nvidia-collective-communication-library-2-12/>

# バイセクションバンド幅 (Bi-section Bandwidth)

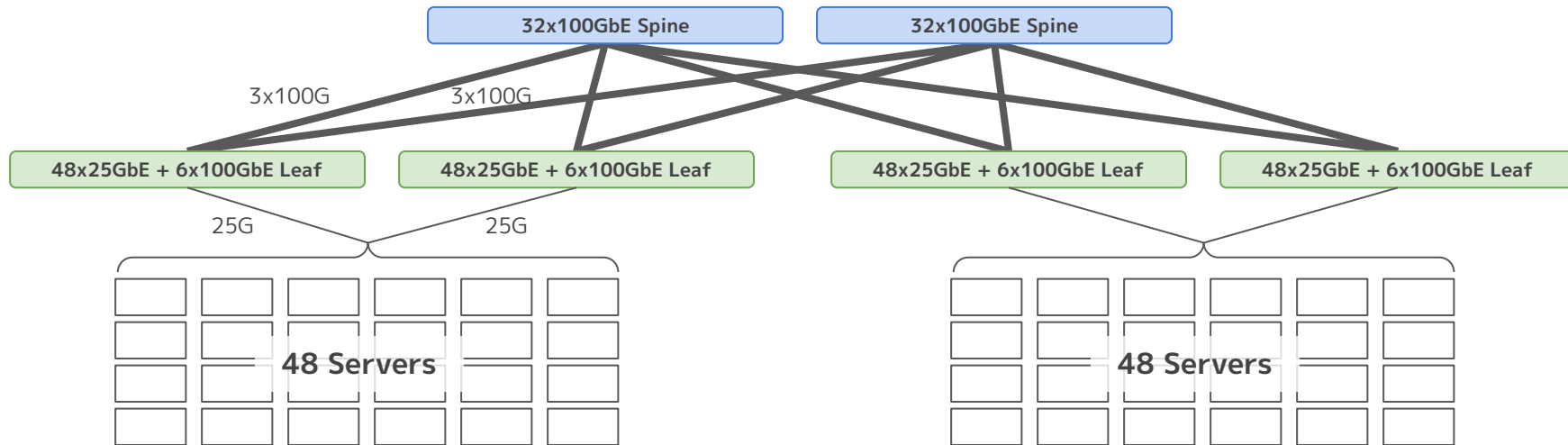
- 全ての計算ノードが全力で通信した時にシステム全体で達成しうる通信性能の下限
- = 計算ノードを二分割したときのグループ間の通信帯域幅の最低値



<https://www.acri.c.titech.ac.jp/wordpress/archives/2870>

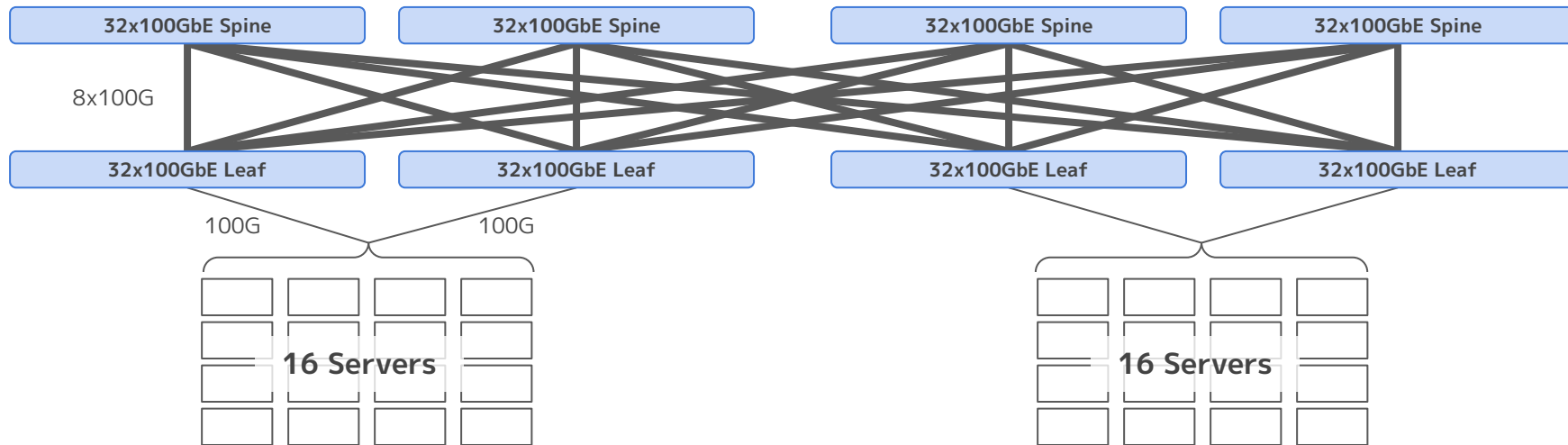
# Constant Bisection Bandwidth(CBB)

- バイセクションバンド幅が定数になるネットワーク
  - LeafでオーバーサブしているようなCLOSトポロジー
- バイセクションバンド幅1200Gbps、2:1 OversubscriptionのCBBの例
  - ダウンリンク:  $2 \times 48 \times 25 \text{Gbps} = 2400 \text{Gbps}$
  - アップリンク:  $2 \times 6 \times 100 \text{Gbps} = 1200 \text{Gbps}$



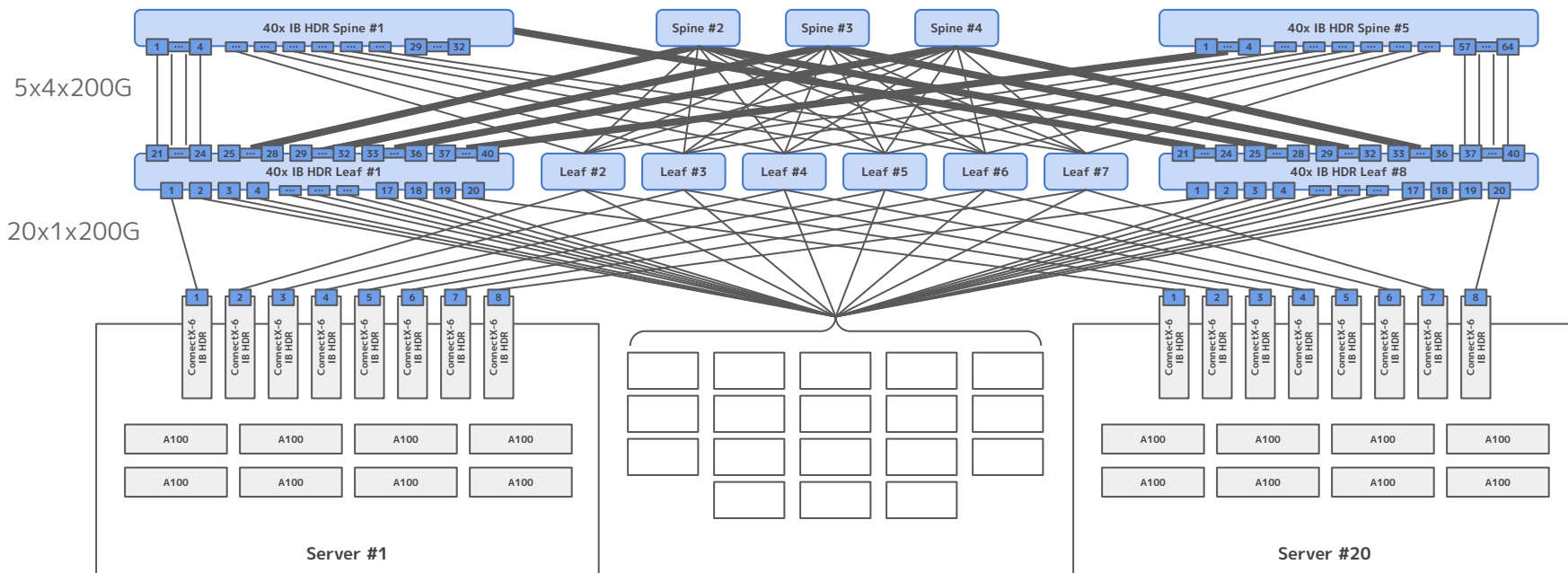
# Full Bisection Bandwidth (FBB)

- クラスタ内の任意の半数のノードが同時に残り半分のノードにデータを送信してもネットワーク内での競合が発生しないネットワーク
  - 各スイッチのオーバーサブ率は1:1以上である必要がある
- バイセクションバンド幅3200Gbps、1:1 OversubscriptionのFBBの例
  - ダウンリンク:  $4 \times 8 \times 100\text{Gbps} = 3200\text{Gbps}$
  - アップリンク:  $2 \times 16 \times 100\text{Gbps} = 3200\text{Gbps}$



# DGX A100 SuperPOD 1SUのフルバイセクション構成

- バイセクションバンド幅32,000Gbps、1:1 Oversubscription
  - サーバからの帯域:  $20 \times 8 \times 200\text{Gbps} = 32,000\text{Gbps}$
  - Leaf/Spine間の帯域:  $8 \times 5 \times 4 \times 200\text{Gbps} = 32,000\text{Gbps}$

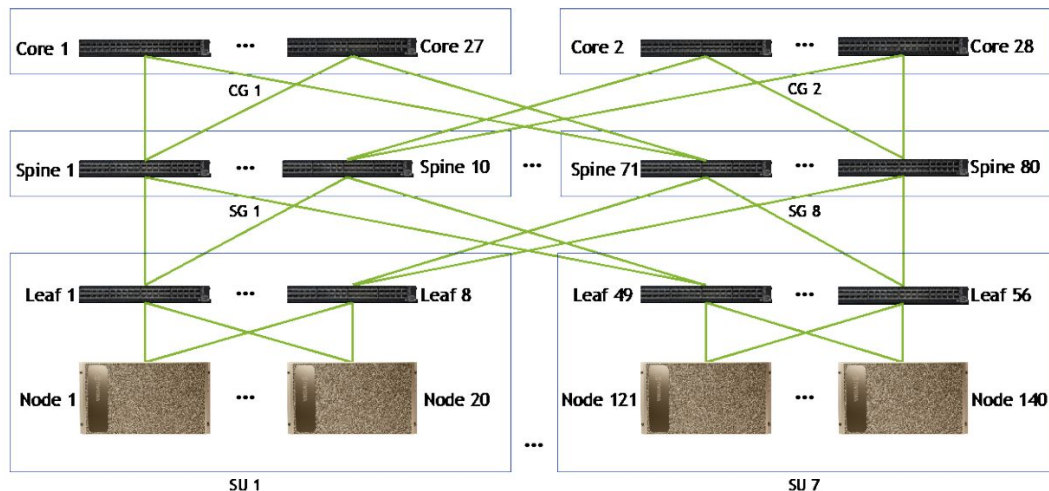




# DGX A100 SuperPOD 7SUのRail-optimized Topology

- SU間でもRail-optimizedな構成にするためにSpine Groupを構成する
  - SUをまたぐノード間の同一GPU番号の通信をSpineで折り返せるようにする
  - 各SUのLeaf 1番をSpineのグループ1番に収容する→8グループ必要
  - フルバイセクション構成にする→各Spineグループには10台のSpineが必要
  - 全部でSpineは80台必要

Figure 4. Compute fabric topology for a 140-node DGX SuperPOD



<https://images.nvidia.com/aem-dam/Solutions/Data-Center/gated-resources/nvidia-dgx-superpod-a100.pdf>

# InfiniBand Adaptive Routing

- 負荷が一番少ないポートからトラフィックを転送することで特定のリンクの輻輳を回避する
  - InfiniBandのStatic Routingでは特定のデータフローは特定のパスだけを通る
  - 複数の広帯域データフローが1つのリンクに集中してしまうと輻輳が発生してしまう
- Out-of-Orderが発生するのでHCA(サーバのNIC)でリオーダーする



<https://developer.nvidia.com/zh-cn/blog/accelerating-your-network-with-adaptive-routing-for-spectrum-ethernet/>

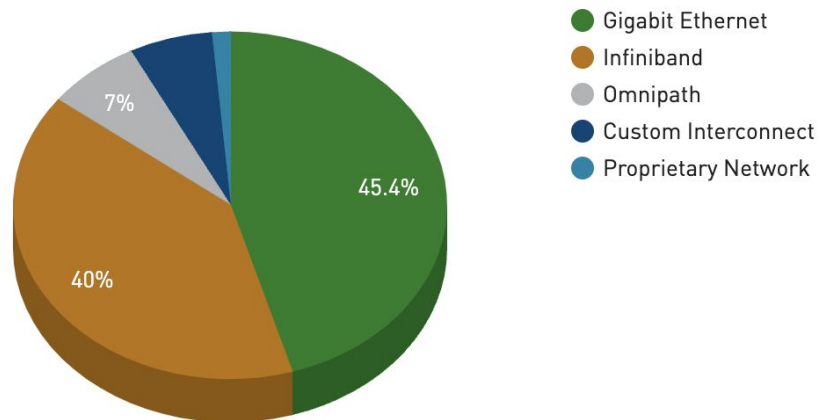
※図はEthernetのAdaptive Routingですが概念やイメージは同じ

# Ethernet-based Network

# HPCにおけるEtherent

- HPCのTOP500のInterconnectのシェアはEthernet 45% vs Infiniband 40%
- それなりのパフォーマンスのシステムを比較的安価につくることができる
  - TOP500の上位はカスタムとかInfiniBand

Interconnect Family System Share



<https://www.top500.org/statistics/list/> (2023/June)

# Ethernetを採用するモチベーション

- コスト
  - 安く作りたい
- **マルチテナンシー**
  - トラフィックを分離してセキュリティを確保したい
- **複数ジョブのパフォーマンス隔離**
  - 複数のJobの間でパフォーマンスの影響を排除したい
- **SDN**
  - コンピュートとネットワークの動的なプロビジョニングを行いたい
- **既存のサービスとの連携**
  - クラウドプロバイダ内で提供されている他のサービスと連携を行いたい
- **セキュリティとコンプライアンス**
  - パケットブローカーやIDSとの連携を行いたい
- **オープン化**
  - SONiCなどのオープンなソフトウェアを使って構築したい

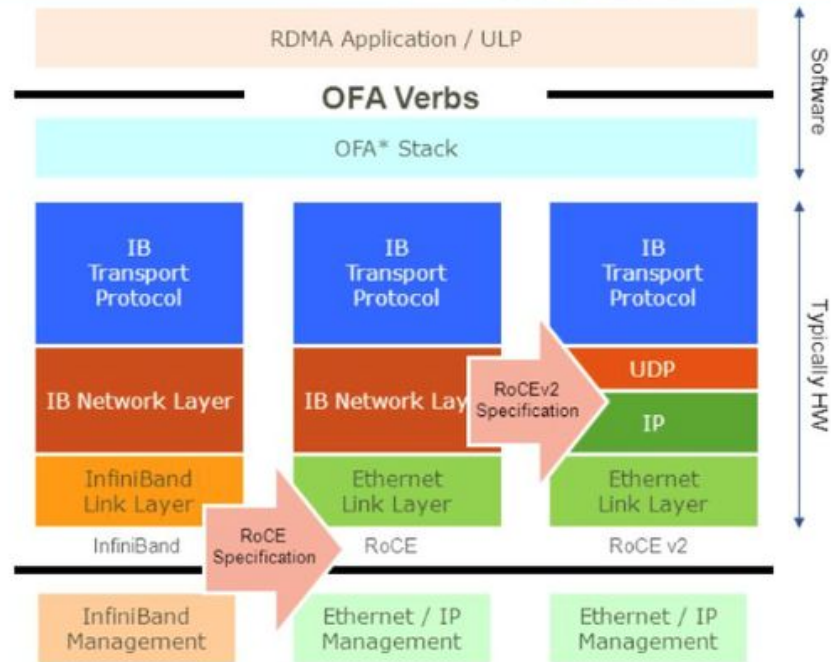
# EthernetでAI/ML学習用ネットワークを作るには

InfiniBandのような特性を持ったネットワークを作る必要がある

- **Lossless:** パケットをDropしないようにすること
  - RDMAはInfiniBandのロスレスを基本としているので再送に弱い(Go-Back-N)
  - QoSと輻輳制御でパケットを落とさないようにする
  - EthernetはCSMA/CDのプロトコル
- **Effective Load Balancing:** 帯域を有効活用して輻輳を防ぐこと
  - そのまえにPure L2ではマルチパスできないのでIP Fabricが必要
  - InfiniBandのAdaptive Routingに相当する機能が必要

# RoCEv2 (RDMA over Converged Ethernet)

- InfiniBandのRDMAをEthernet/IPネットワークで実現するための技術
- IBのトランスポート(BTH)をUDPの上に乗せる



<https://www.keysight.com/blogs/tech/traf-gen/2021/04/19/understanding-real-rocev2-performance>

# RoCEv2のためのLossless Ethernet Configuration

- **ETS** (Enhanced Transmission Selection)

- ホスト側の技術
- 送信トラフィックにQoSを行う

- **PFC** (Priority Flow Control)

- スイッチ側の技術
- VLAN CoSまたはIP ToSで優先度制御を行い、優先度キューごとにXoff/Xonで制御する。

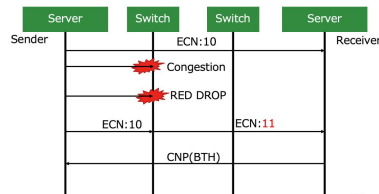
- **ECN** (Explicit Congestion Notification)

- スイッチ側+ホスト側の技術
- IP ToSのECNの2-bitを使用して輻輳を制御する
- 途中のスイッチが輻輳を検知するとビットを立ててCongestion Experienced(CE)を通知
- CEを受け取った受信側ホストはCongestion Notification Packet(CNP)を送信側ホストに通知し、トラフィック送信を抑制してもらう
  - ※CNPはInfiniBandのBTH内で示される

## ECN (Explicit Congestion Notification)

- IP ToS(ECN)の2bitで輻輳制御を実施
- 輻輳を検知したらRED (Random Early Detection): [REDはoptionであることが多い]
- 輻輳時は全てのパケットにECN CE(11)をマーキング
- ReceiverはSenderにCNPを送信、Senderは送信レートの制限を実施
- エンドノード間の帯域制御によって輻輳を根本的に解決

bit	Description
00	ECN非対応
01	ECN対応 ECT[1] (CNPの他に利用CNPはBTHで定義)
10	ECN対応 ECT[0] (一般的に利用)
11	輻輳発生 / CE



32

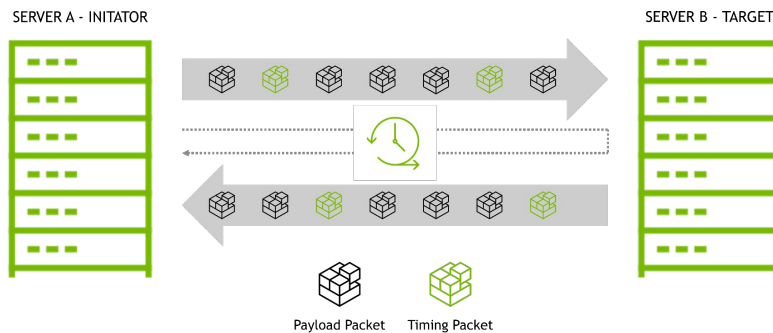
<https://www.janog.gr.jp/meeting/janog52/wp-content/uploads/2023/06/janog52-aiml400-uchida-koshoji.pdf>



# NVIDIA ZTR (Zero Touch RoCE)

- DCQCN (PFC+ECN)の設定を行わずにLossless Ethernetを実現する仕組み
  - NVIDIA独自の仕組み
  - DCQCNの手動設定の98-99%程度のパフォーマンスをゼロタッチで実現できる
  - Round-Trip Time Congestion Control (RTTCC)と一緒に使う
  - NIC側=ConnectXに設定する
- MicrosoftがAzure Stack HCIで採用している

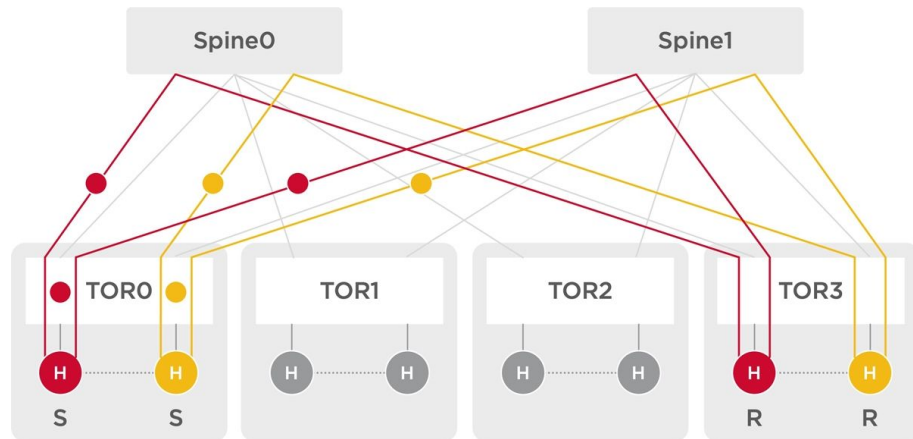
LATENCY MEASUREMENT FOR NETWORK CONGESTION CONTROL



<https://developer.nvidia.com/blog/scaling-zero-touch-roce-technology-with-round-trip-time-congestion-control/>

# Adaptive Routing / Dynamic Load Balancing

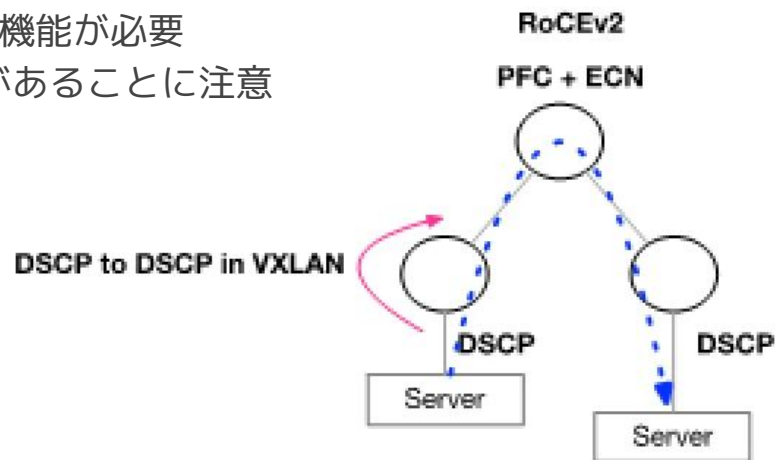
- ポートの使用率やキューの使用状況に基づいて動的に経路を選択する
- RoCEv2のECMP+ハッシュによるフローの偏りをなくすることができる
  - Spectrum + Cumulus Linuxの場合ポートごとの帯域使用率のしきい値で発動を制御
  - Flowlet (Flowの時分割)単位ごとに振り分ける
- BroadcomのTomahawk 4では Dynamic Load Balancingという機能
  - リオーダーをどこでやっているのかはよくわからない



<https://www.keysight.com/blogs/tech/traf-gen/2021/04/19/understanding-real-rocev2-performance>

# マルチテナンシーRoCEv2の実現

- マルチテナンシーを実現するためにはネットワークのIsolationが必要
  - クラウドサービスの学習環境が提供可能になる
- 従来の方で実現可能
  - EVPN-VXLANによるL2VPN
  - VLAN+VRF (やりたくない)
- EVPN-VXLANによるL2VPNで実現する場合
  - PFCのDSCPをOuterのIPヘッダにコピーする機能が必要
  - サーバ側で設定されたDSCPを信用する必要があることに注意



<https://docs.nvidia.com/networking/display/onyxv3102202/vxlan>

# Ethernet GPU Fabric - ベンダーの動向

※個人の印象です

- NVIDIA、Cisco、Broadcom系の3大勢力を感じる
- NVIDIAはSpectrumシリーズ
  - 100G/200G/400G/800Gと揃っていて、ライブラリからNICまで全部持っているのでRoCEv2でもNVIDIA Spectrumを使って垂直統合で組む人が多い印象
    - 特にA100用の200GはService Provider系のベンダーはあまり製品を持ってない
- CiscoはNexusシリーズ
  - 現在はCloud Scale ASICがほとんど。これからSilicon Oneでも出てくる？
  - Preferred Networksは[Nexus 9000シリーズでRoCEv2を使用](#)しMN-2を構築
- Broadcom系
  - Arista: [MetaのAIプラットフォームで7800シリーズが使われている](#)らしい
  - Juniper: [SambaNova社でQFX5200を使用](#)してMLクラスタを構築
  - WBS+SONiC: 公開事例なし？

# CyberAgentさんの事例

- DellのPowerEdge XE9680にHGX H100を搭載
- NICはConnectX-7で400GbE (RoCEv2)
- スイッチはNVIDIA SN4700でフルバイセクション Fat-Treeトポロジー
- Lossless Ethernet + Adaptive Routing



<https://twitter.com/yaemonsan/status/1685934499425828869>

## インターコネクト設計

- 構築・運用しやすい構成
  - ネットワーク構成はFat-Treeを採用
  - BGP Unnumberedを採用
  - メンテナンス時はG-shut communityによる迂回
  - L2 延伸しない(L2は1台のLeafで閉じる)
- フローの偏りの抑止
  - LAGを使わない
  - Adaptive Routing によるフローの偏りを解消
- RoCEv2への対応
  - フルバイセクション
  - Lossless Ethernetへの対応
    - PFC, ECN(CNP), ETS
  - 400GbEへの対応



20

<https://www.janog.gr.jp/meeting/janog52/wp-content/uploads/2023/06/janog52-aiml400-uchida-koshoji.pdf>

# Scheduled Fabric

- Fat-TreeトポロジーのFabricをEthernet/IPベースではなくVOQベースで実装したFabric
  - シャーシ型スイッチのアーキテクチャ
  - Leafがラインカード、Spineがスイッチファブリック
- パケットまたはそれより細かい単位(Cell等)でスイッチングする
- クレジットベースで転送するのでロスレス
- リンク負荷に応じて均等に振り分けるので輻輳が発生しない
- 製品・導入事例はまだなさそう？

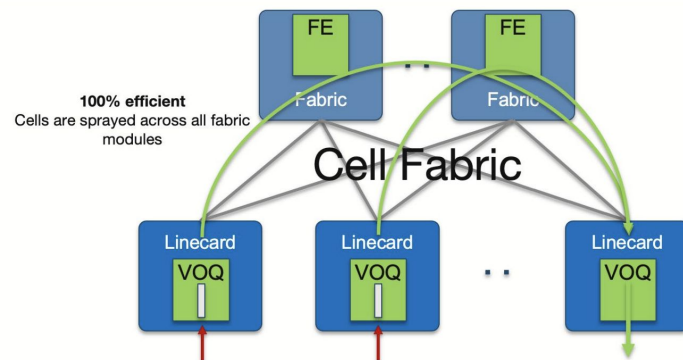


Figure 13: Cell-based Fabric Architecture

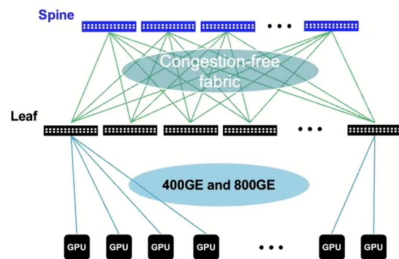
<https://www.arista.com/assets/data/pdf/Whitepapers/AI-Network-WP.pdf>

# BroadcomのScheduled Fabric

- LeafにJerico系、SpineにRamon系を使用したFabric
  - 現行はJerico2C+ & Ramon
  - 次世代はJerico3-AI & Ramon3
- 実装はDriveNets DDCが有力？
  - 2023-10-17にJerico3-AI & Ramon3の“Network Cloud-AI”の[プレスリリース](#)

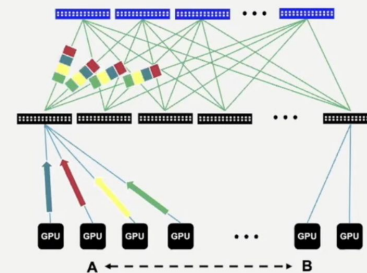
## Jericho3-AI Ethernet Fabric

- Switch scheduled fabric
  - Standard Ethernet I/O
  - Leaf: switching, forwarding, queuing, scheduling
  - Spine: forwarding at low power
  - Receiver based scheduling
- Leaf deployment options
  - ToR/MoR - in the GPU racks
  - In the network rack, with spines



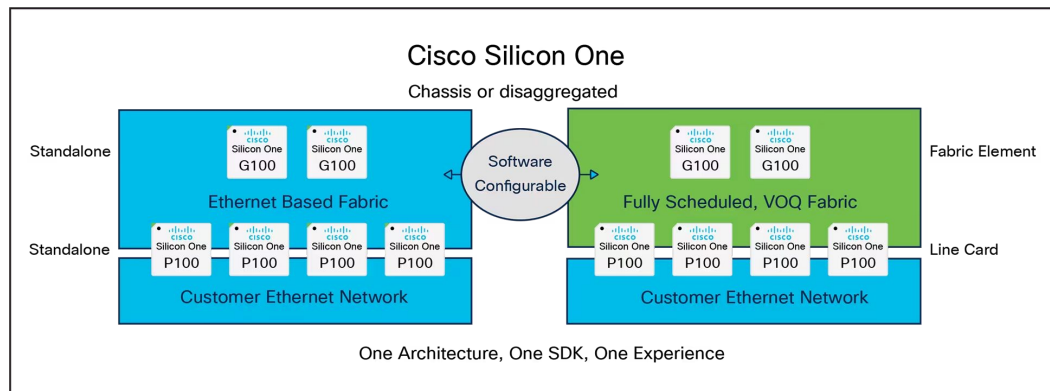
## Perfect Load Balancing

- Equal spraying over all links of the fabric, independent of flow size
- Uniform link utilization – avoids hot spots
- Consistent high performance at all network loads

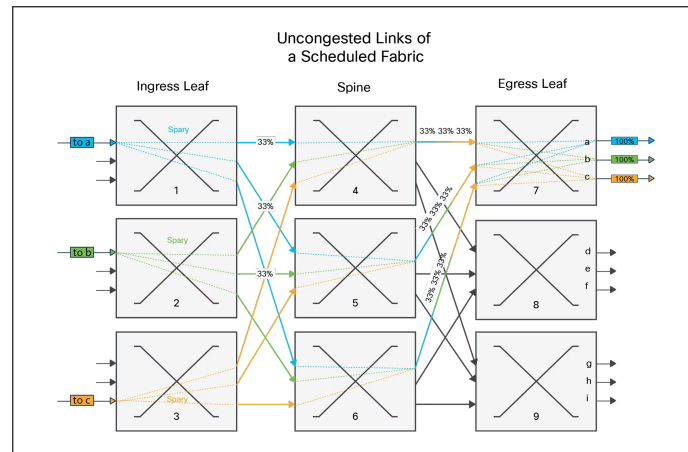


# CiscoのScheduled Fabric

- Silicon OneのスイッチでLeaf-SpineをVOQ Fabricモードで動作させる
  - Unscheduled Ethernet fabric に対して Fully scheduled fabric と呼んでいる
- Spray and Reorder
  - Ingressではパケットごとに出カリンクに対して均等に分散して送出される(Spray)
  - Egressでパケットを並び替える(Reorder)
- ボックス型のSilicon One Nexusで対応？



<https://www.cisco.com/c/en/us/solutions/collateral/silicon-one/silicon-one-wp.html>



<https://www.cisco.com/c/en/us/solutions/collateral/silicon-one/evolve-ai-ml-network-silicon-one.html>



# シャーシ型スイッチによるクラスタの構築

- Pizza Box SwitchでScheduled Fabricを組むぐらいならシャーシ型で組んでしまえばいいのでは？
- Arista 7816R3 (576 x 400G) 2台で576 GPU (DGX 72台)のクラスタ
  - BroadcomのJerico 2C+ラインカード & Ramonファブリックの構成

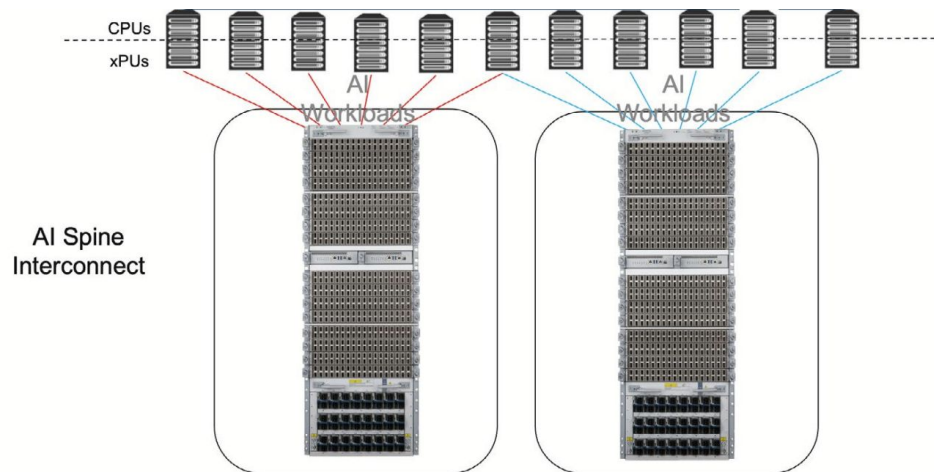


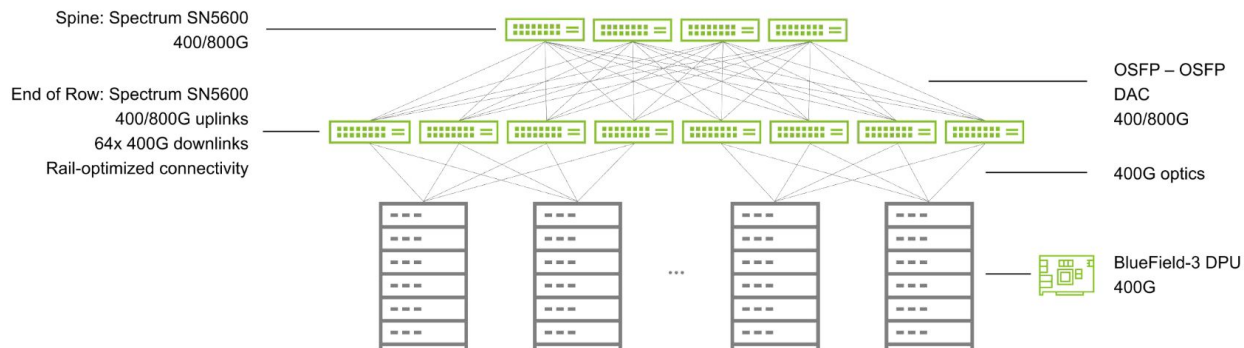
Figure 20: AI Spine Interconnect

<https://www.arista.com/assets/data/pdf/Whitepapers/AI-Network-WP.pdf>

# NVIDIA Spectrum-X Network Platform

- NVIDIAが提供するEthernetベースのAI/MLクラスター用ネットワークプラットフォーム
  - SuperPODのようなリファレンスアーキテクチャはまだ公開されていない
  - [ホワイトペーパー](#)は入手可能
- Spectrum-4 (400G/800G)とBlueField-3 DPUの組み合わせ

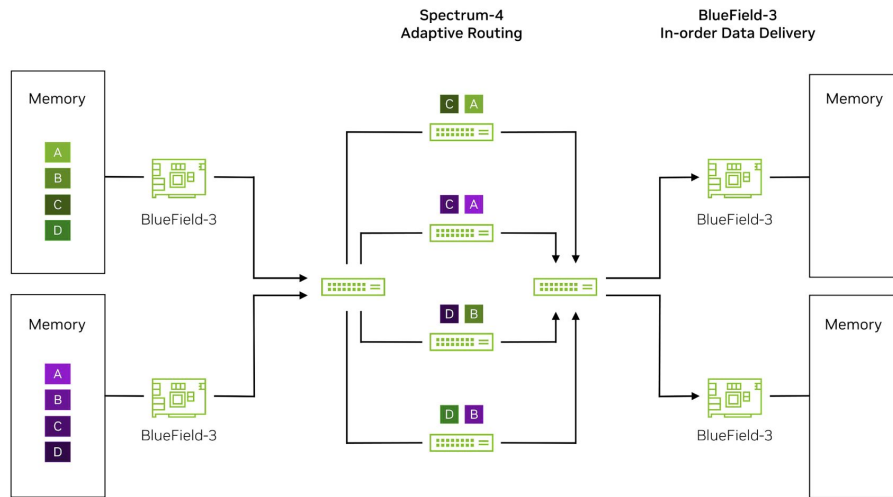
Figure 4. Typical Spectrum-X Network Topology



<https://nvdam.widen.net/s/6lmkmc8lqg/nvidia-spectrum-x-whitepaper>

# NVIDIA Spectrum-X Network Platform

- Flowletやポート使用率ベースではなく、Packet単位でロードバランスする
  - Spectrum-4 Adaptive Routing
- BleField-3 DPUで受信したパケットを並び替える
  - NVIDIA Direct Data Placement (DDP)



<https://nvdam.widen.net/s/6lmkmc8lqg/nvidia-spectrum-x-whitepaper>

# NVIDIA Israel-1

- NVIDIA Spectrum-X Network Platformを使用してイスラエルで建設中のスーパーコンピュータ
  - 2023年末頃にできあがるらしいので、そのあとちょっとしてからリファレンスアーキテクチャが公開されるかもしれない



<https://www.timesofisrael.com/nvidia-taps-into-israeli-innovation-to-build-generative-ai-cloud-supercomputer/>

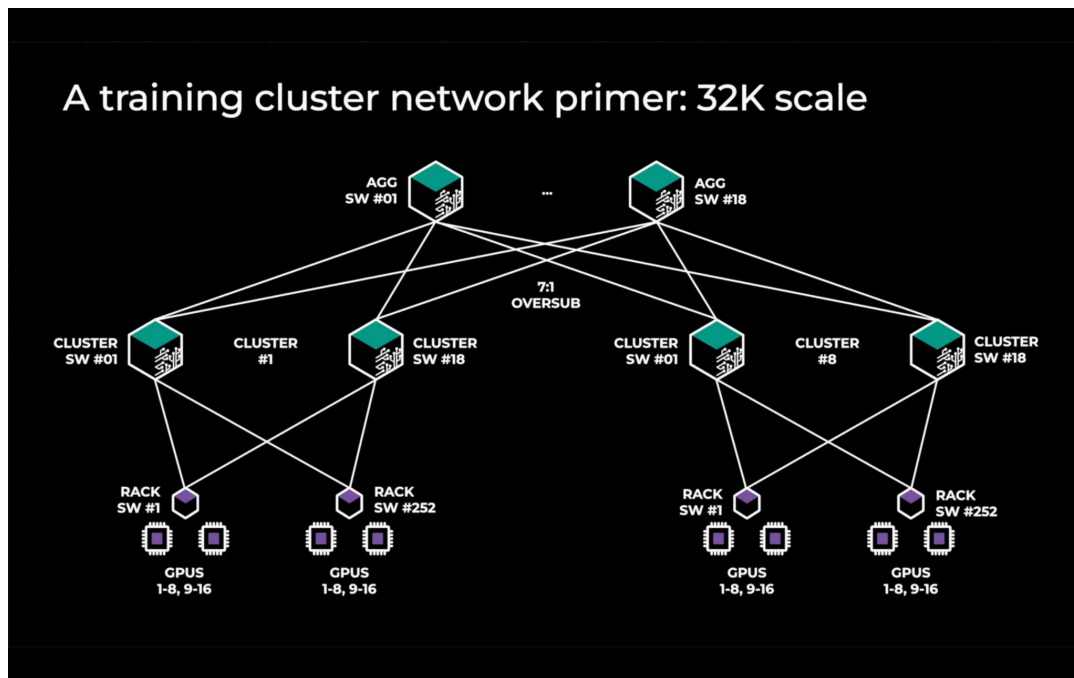
# Ultra Ethernet Consortium (UEC)

- AI/MLなどのHPCのためのEthernetの通信規格の実現を行う団体
  - RoCE プロトコルを Ultra Ethernet Transport に置き換えることを目指している
  - Multipath/Spraying, Congestion Notificationなどにより効率化を行う
- 活動内容
  - Ethernet通信におけるプロトコル、電氣的/光学的信号特性、APIおよびデータ構造
  - 既存のリンクおよびトランスポートプロトコルを拡張または置き換えるための、リンクレベルおよびエンドツーエンドのネットワークトランスポートプロトコル
  - AIやマシンラーニング、HPC環境に適したリンクレベルおよびエンドツーエンドでの輻輳、テレメトリ、信号のメカニズム
  - さまざまなワークロードおよび動作環境を促進するソフトウェア、ストレージ、セキュリティの構成
- 設立メンバー
  - AMD、Arista、Broadcom、Cisco、Eviden(an Atos Business)、HPE、Intel、Meta、Microsoft

# 最新の研究動向

# フルバイセクションバンド幅は必要なのか？

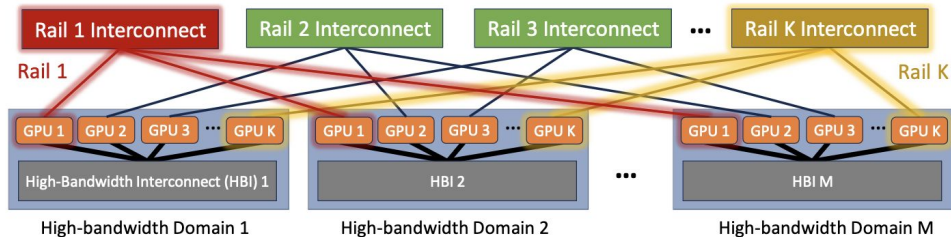
- Metaの32K GPUクラスタ構想では7:1のオーバーサブスクリプション
  - LLMの学習においては実際にフルバイセクション構成が必要ない



<https://atscaleconference.com/videos/networking-for-genai-training-and-inference-clusters/>

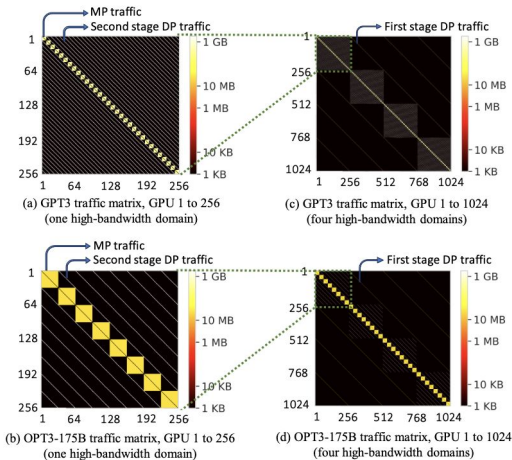
# フルバイセクションバンド幅は必要なのか？

- 2023-07-22 に投稿された論文でMetaが“rail-only” ネットワークを提案
  - [“Optimized Network Architectures for Large Language Model Training with Billions of Parameters”](#)
  - Model Parallel と 1st/2nd Stage のData Parallelのデータ転送量の計測からRail間の通信がそれほど多くないことを発見したためRail間の通信をネットワークから除去
  - ネットワーク機器コストを最大75%減らせた



**Figure 6: Our proposal: replace the any-to-any connectivity with a *rail-only* connection.**

<https://huggingface.co/papers/2307.12169>



**Figure 4: Traffic heatmaps for GPT3 and OPT3-175B.**



# ネットワークトポロジー

- Fat Tree (folded CLOS)トポロジーではノード数が増えるとホップ数が増えるし、リンクの本数が多いためコストや消費電力の面で課題がある
- Dragonfly+トポロジーではホップ数が均一になりリンクの数も減らすことができる

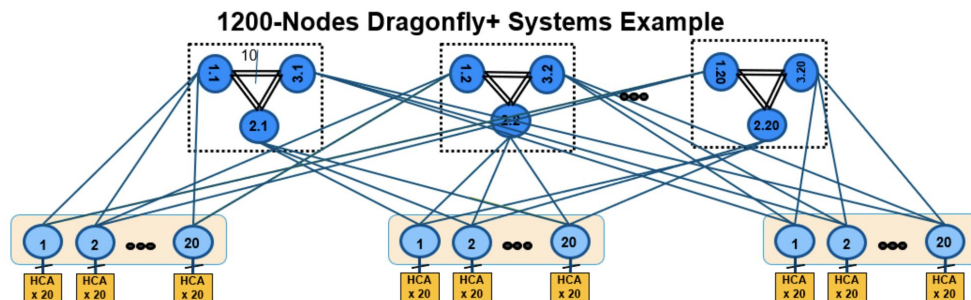
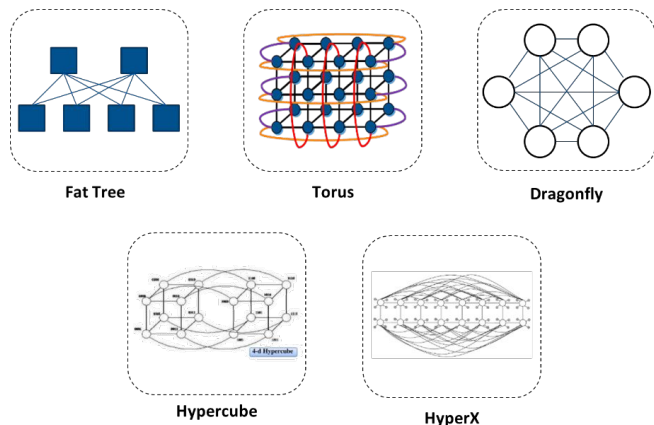


Figure 3. Dragonfly+ topology: reduces total cost of ownership, with fewer long Active Optical Cables

<https://network.nvidia.com/files/doc-2020/wp-saving-power-in-the-modern-datacenter.pdf>

<https://www.hpcwire.com/2019/07/15/super-connecting-the-supercomputers-innovations-through-network-topologies/>

# Routing in Dragonfly+ Topology

- 2023-07-10 に IETF の rtgwg に [Routing in Dragonfly+ Topologies](#) というInternet-DraftがSubmitされた
  - InfiniBandでは実現可能だったDragonfly+を、IPルーティングでも実現するための実装方式について提案している
  - 2023-10-14に開催された[JANOG52.5](#)では「[draft-agt-rtgwg-dragonfly-routingを試してみた](#)」というタイトルのLTがあり、さっそくこのDraftの内容を動かした人がいた

## Dragonfly+ってなあに？

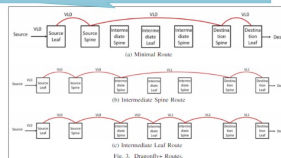
■当該トポロジに対応したパス選出とロードバランシングが必要

□IP Networkにおけるルーティング ⇒ draft-agt-rtgwg-dragonfly-routing

➢VRF/BGPを活用することで、ホップ数が異なるパスを等コストとして扱う

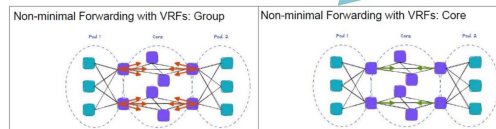
1. High priority : Minimal route (LGL) [Minimal path]
2. Medium priority : Intermediate spine route (LGGL) [Non-minimal path]
3. Low priority : Intermediate leaf route (LGLGL) [Non-minimal path]

Dragonfly+では異なるホップ数となる3種類のパスが存在



Source: [Dragonfly+: Low Cost Topology for Scaling Datacenters](#)

VRFを用い参照テーブルを使い分けることによってDragonfly+でマルチパスとループ回避の両立を実現



Source: <https://datatracker.ietf.org/meeting/117/materials/slides-117-rtt-dragonfly-routing-00>

[https://www.janog.gr.jp/meeting/janog52.5/doc/lt1\\_dragonfly-routing.pdf](https://www.janog.gr.jp/meeting/janog52.5/doc/lt1_dragonfly-routing.pdf)

# まとめ

- LLMのブームにより従来のデータセンターネットワークエンジニアもHPCネットワークを理解して構築できるようになる必要がでてきた
- InfiniBandからEthernetへ、オープン化・コモディティ化の動きがあり現在進行系で進化している

Special Thanks!!

LINEヤフー 小林さん、CyberAgent 高橋さん & 内田さん & 小障子さん  
ネットワンシステムズ 平河内さん