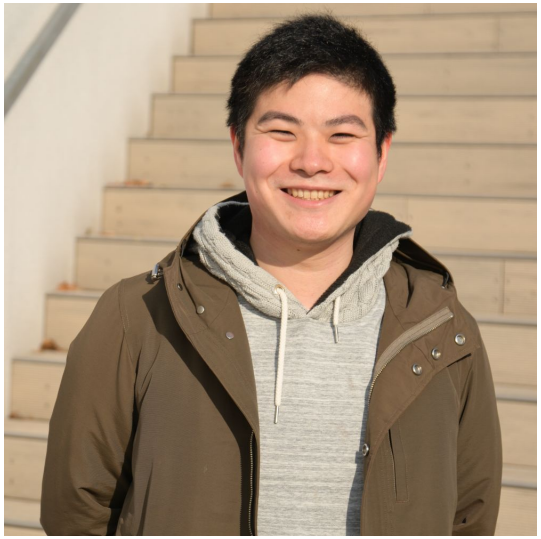
Preferred Networks

アクセラレータ間通信の実際

MPLS Japan 2024

Yuichiro Ueno / Preferred Networks, Inc.

自己紹介：上野 裕一郎 (Yuichiro Ueno / @y1r)



- むかし: 東工大 横田理央研究室 修士課程
  - 「多数のGPUで深層学習を効率的にやる」
  - 集団通信アルゴリズムの開発・評価とか
  - グランドチャレンジ(2048GPUs学習)とか
- 2021/04~: Preferred Networks に新卒入社
  - Kubernetes クラスタの開発 & 運用
  - GPU, MN-Core, NIC 周りなどに興味
  - 分散キャッシュシステムの開発
  - MPLS 未経験です...

Preferred Networks

Preferred Networks（PFN）会社概要

ミッション： 現実世界を計算可能にする

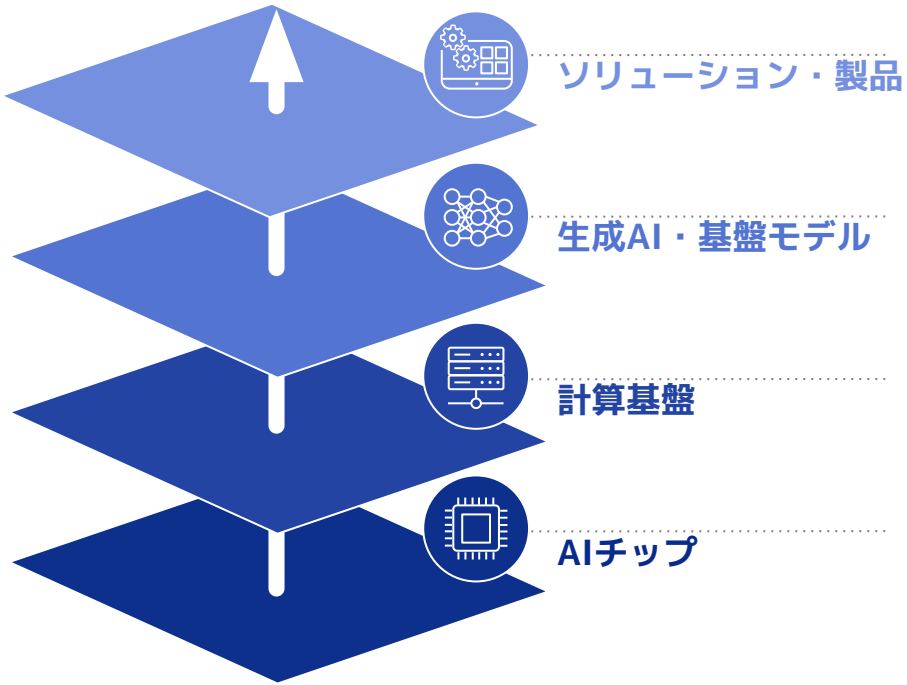
設立	2014年3月26日
本社	東京都千代田区
代表取締役	西川徹（最高経営責任者）、岡野原大輔（最高研究責任者）
従業員数	約350名（2024年9月）
事業内容	AIチップ、計算基盤、生成AI・基盤モデルなどのAI関連技術を活用したソリューション・製品の開発・販売および研究開発
主要子会社	株式会社Preferred Computational Chemistry（2021年6月） 株式会社Preferred Robotics（2021年11月） 株式会社Preferred Elements（2023年11月）
出資企業	トヨタ自動車株式会社　ファナック株式会社 日本電信電話株式会社　ENEOS ホールディングス株式会社 中外製薬株式会社　株式会社博報堂DYホールディングス 株式会社日立製作所　三井物産株式会社　みずほ銀行株式会社 東京エレクトロン株式会社

https://www.preferred.jp

Preferred Networks

PFNの事業: AI技術のバリューチェーンを垂直統合

PFNは、チップ、計算基盤、生成AI・基盤モデル、ソリューション・製品まで、AI技術のバリューチェーンを垂直統合し、ソフトウェアとハードウェアを高度に融合することで、競争力の高い技術の開発および産業応用を進めています。



ソリューション・製品

様々な産業・消費者向けのソリューション・製品群

PreferredAI　MiseMise　MATLANTIS　PFN 3D Scan　くりあこ rypko　kachaka　Preferred Networks Visual Inspection

生成AI・基盤モデル

PLaMo

PLaMo Prime（今秋提供予定のLLM）
PLaMo Lite（エッジ向けSLM）

PFP

物質のエネルギー計算モデル

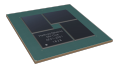
計算基盤

GPUクラスタ

MN-3
(MN-Core™
クラスタ)

MN-Core™ 2のクラウド提供
（2024年開始予定）

AIチップ



MN-Core™

MN-Core™ 2

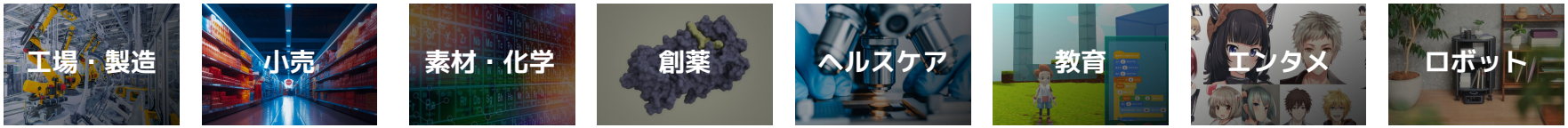
LLM向け
推論チップ
（2026年提供予定）

MN-Core
第三世代

Preferred Networks

PFNの事業: AI技術の水平展開

PFNは、AI技術のバリューチェーンを垂直統合し、産業、コンシューマー、社会に向けて様々な領域でソリューション・製品を水平展開しています。



産業
生産性向上・品質改善
属人化回避・人手不足解消

社会
安心・安全な社会
高度な教育・医療

消費者
人間の能力の拡張
新しい創作表現・娯楽体験

生成AI・基盤モデル

計算基盤

AIチップ

Preferred Networks

PFNの計算基盤

他にもいくつかありますが
大規模なものだけ抜粋してご紹介させてください

	時期	アクセラレータ	アクセラレータ間通信
MN-1	2017/09 ~ 2022/07	NVIDIA P100 x 8	InfiniBand FDR x 2
MN-1b	2018/07 ~ 2022/07	NVIDIA V100 x 8	InfiniBand EDR x 2
MN-2A	2019/07 から運用中	NVIDIA V100 x 8	100 GbE x 4
MN-2B	2022/07 から運用中	NVIDIA A30 x 6 NVIDIA A100 x 4	100 GbE x 2
MN-3	2020/05 から運用中	MN-Core x 4	100 GbE x 2 MN-Core DirectConnect
MN-Core 2 クラスタ	2024 から運用中	MN-Core 2 x 8	100 GbE x 4
H100 クラスタ	2024 から運用中	NVIDIA H100 x 8	400 GbE x 4

InfiniBand

RoCE v2

Preferred Networks

今日の話: RoCEv2 を使ったアクセラレータ間通信

- なぜアクセラレータ間通信が必要なのか
  - 分散深層学習、とくに昨今のLLMの分散学習について
- 集団通信（アクセラレータ間通信）を支える技術
  - なぜ集団通信ライブラリが必要か
  - ノード内、ノード内からネットワークへ、ネットワーク
- H100 クラスタ構築でのトラブルシューティング事例
  - ノード内の課題と解決、ネットワーク内の課題と解決
- まとめ

今日の話: RoCEv2 を使ったアクセラレータ間通信

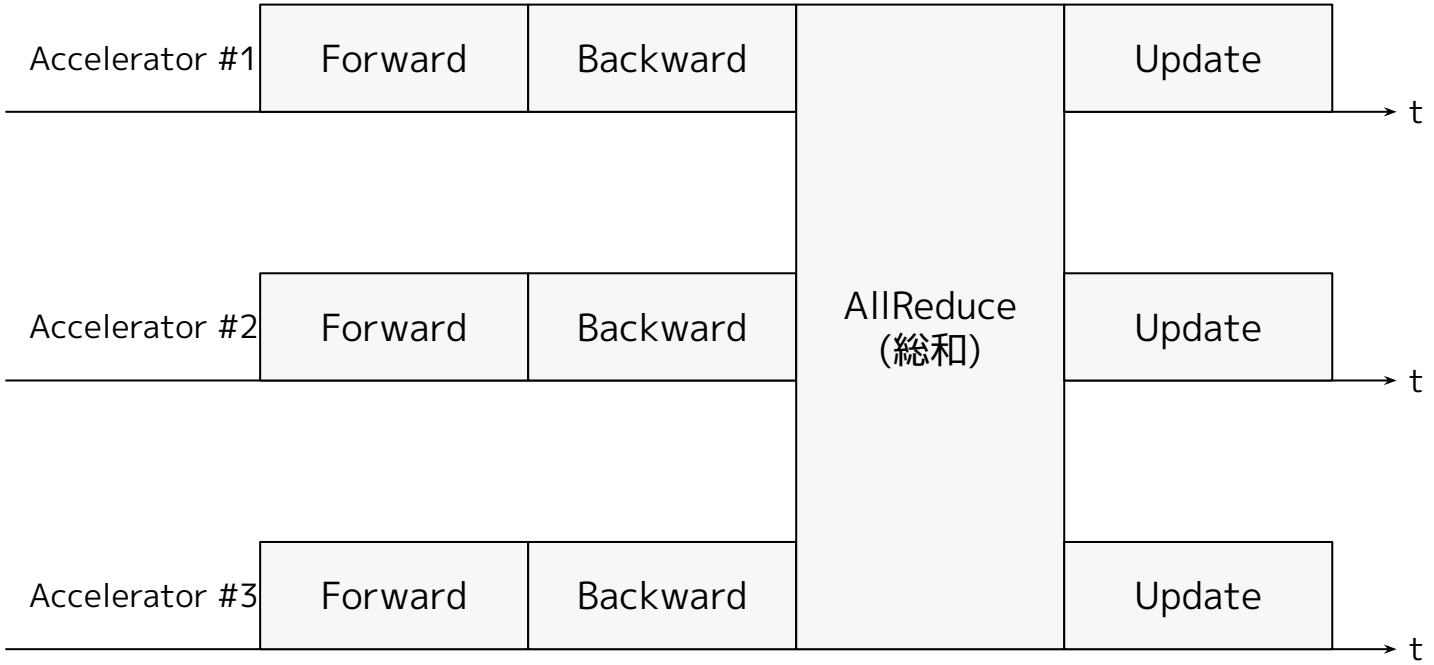
- **なぜアクセラレータ間通信が必要なのか**
  - 分散深層学習、とくに昨今のLLMの分散学習について
- 集団通信（アクセラレータ間通信）を支える技術
  - なぜ集団通信ライブラリが必要か
  - ノード内、ノード内からネットワークへ、ネットワーク
- H100 クラスタ構築でのトラブルシューティング事例
  - ノード内の課題と解決、ネットワーク内の課題と解決
- まとめ

分散深層学習のモチベーション

- 深層学習の計算の流れ
  - Forward 計算　：データをNNに通して推論して、誤差Lを計算
    - $\mathbf{x}$=入力, t=教師信号, l=損失関数, $\boldsymbol{\theta}$=パラメータ
      - $L = \Sigma\, l(t, f(\mathbf{x}; \boldsymbol{\theta}))$
  - Backward 計算：誤差Lのパラメータによる微分を計算
  - 得られた勾配を使って、誤差が小さくなるようにパラメータを更新

- 上の計算を高速化したい → データ並列型分散深層学習
  - Σ の内側の部分を別々の計算機で並列に計算する
  - Σ を計算する
    - AllReduce と呼ばれる集団通信パターンが使用される
  - あとは通常の計算と同じ

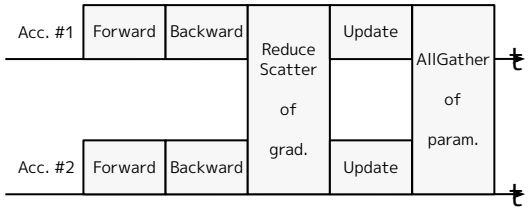
データ並列型分散深層学習 の 1反復



Preferred Networks

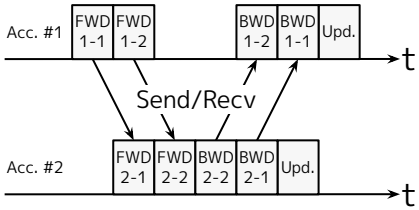
メモリ消費を抑えつつ高速化するための並列化手法

FSDP, ZeRO
Fully Sharded Data Parallel



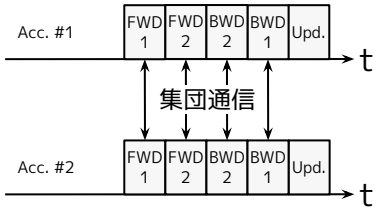
- データ並列化の拡張
- AllReduce を2つの集団通信に分割
  - 勾配の Reduce-Scatter
  - パラメータの AllGather
- パラメータを Scatter (分散) して、パラメータによるメモリ消費量を減らす

パイプライン並列



- モデルをパイプライン的に分割して計算機でそれぞれ役割分担する
- パイプラインステージをまたぐところで必要なデータを Send/Recv する
- 自分が担当しているステージに関するパラメータだけメモリにもつ
  - ほかのステージは持たない

テンソル並列



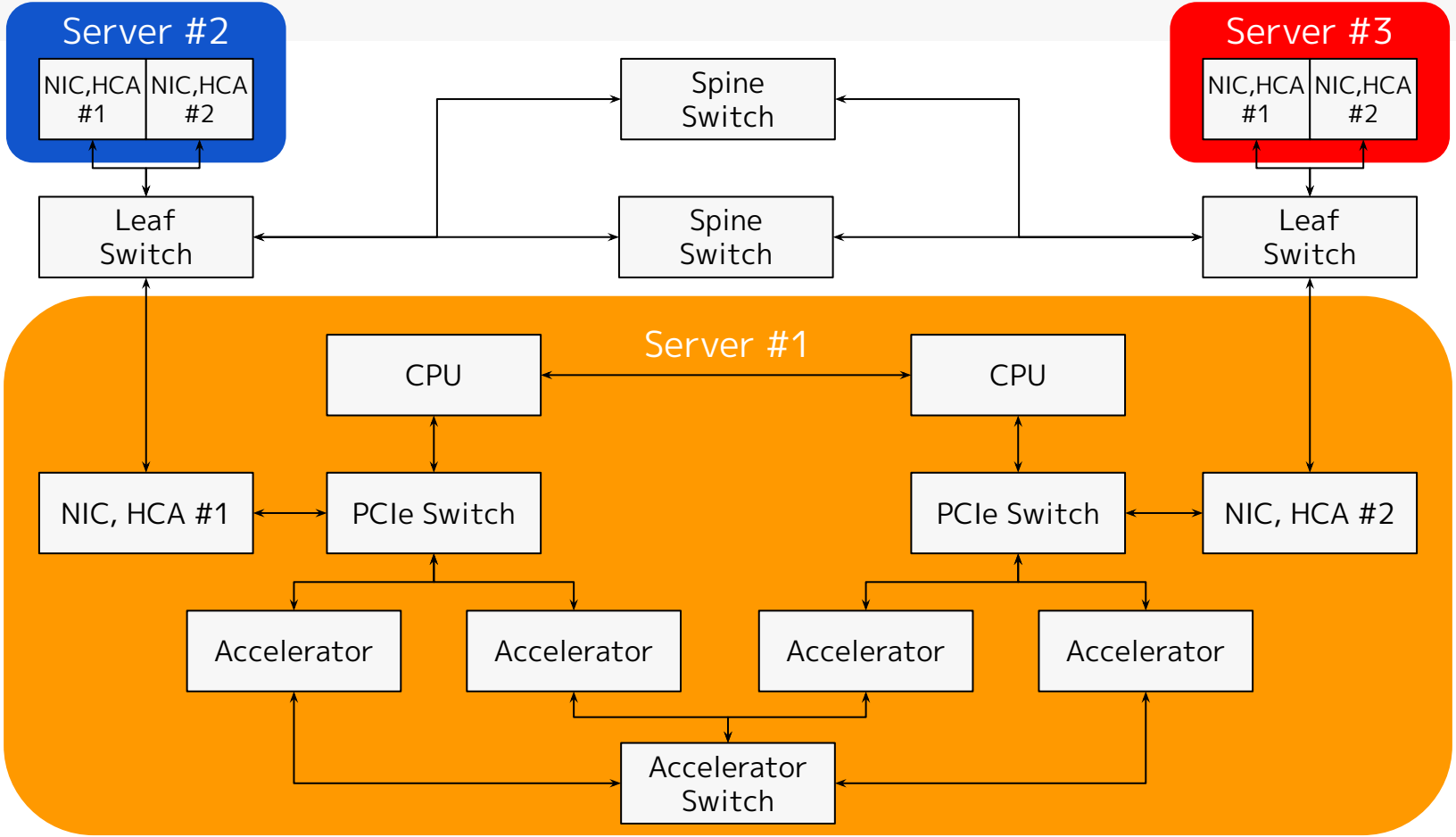
- 各レイヤのパラメータを分割する
- パラメータに入力を掛ける部分を分散行列積など効率的な方法で行う
- 自分が担当しているパラメータの一部だけメモリにもつ
  - ほかの部分は持たない

LLMの文脈では、AllReduceだけではなくReduce-Scatter, AllGather, Send/Recv も高速な必要がある

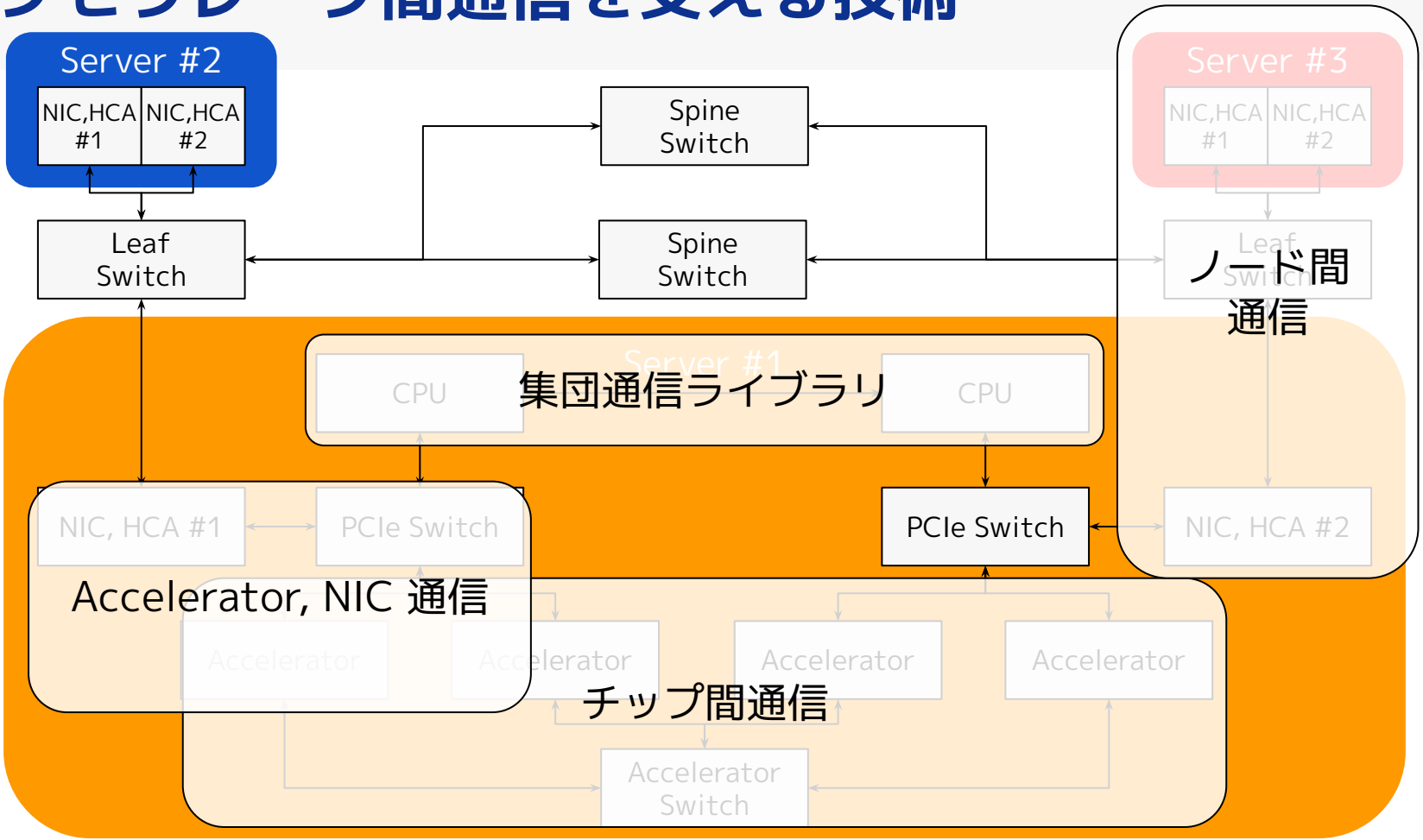
今日の話: RoCEv2 を使ったアクセラレータ間通信

- なぜアクセラレータ間通信が必要なのか
  - 分散深層学習、とくに昨今のLLMの分散学習について
- **集団通信（アクセラレータ間通信）を支える技術**
  - なぜ集団通信ライブラリが必要か
  - ノード内、ノード内からネットワークへ、ネットワーク
- H100 クラスタ構築でのトラブルシューティング事例
  - ノード内の課題と解決、ネットワーク内の課題と解決
- まとめ

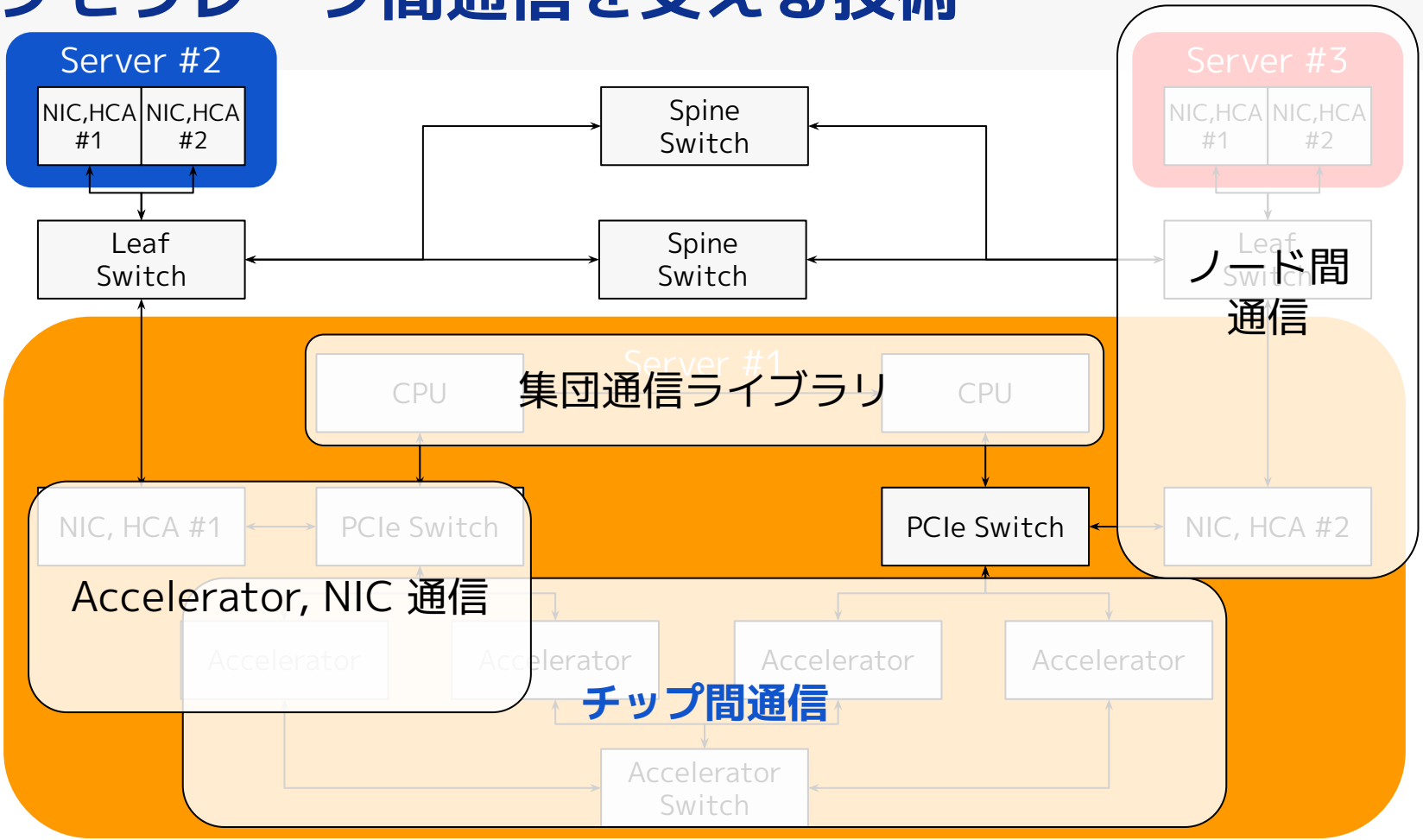
4 GPUs, 2 NICs のサーバ x3 の クラスタ



アクセラレータ間通信を支える技術

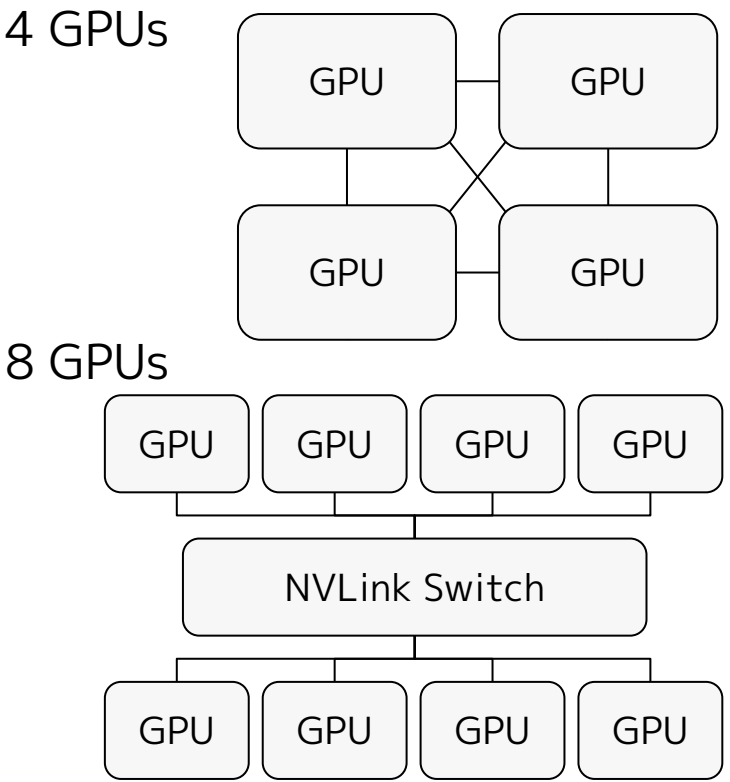


アクセラレータ間通信を支える技術



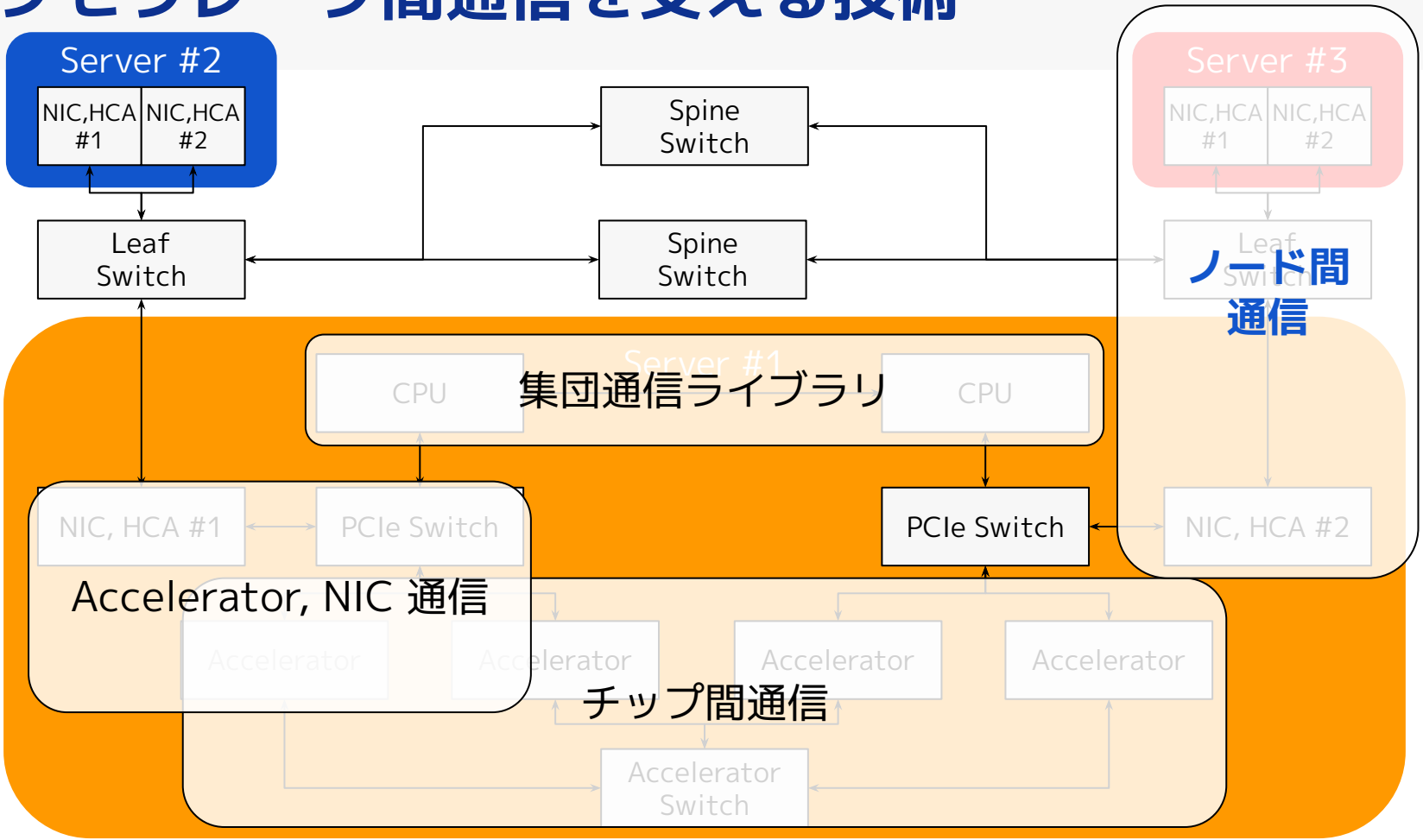
NVIDIA GPU のための NVLink と NVLink Switch

- NVLink
  - GPU の いくつかのインターフェイス
- ノード内 GPU 数によって繋ぎ方が変わる
  - 4 GPUs: GPUに直結
  - 8 GPUs: NVLink Switch 経由で接続
- CUDA の機能で、メモリコピーが可能
  - NVLink があれば NVLink 経由
  - NVLink がなければ PCIe 経由
- それ以上の細かい API はない（はず）



Preferred Networks

アクセラレータ間通信を支える技術



RoCEv2 / InfiniBand

- Remote DMA (別のサーバに対するDMA) の仕組み
  - InfiniBand プロトコル を UDP 上で転送する → RoCEv2

- NICに Ether, IP, UDP, IBのプロトコルがオフロードされている
  - データは CPU/kernel をバイパスして、デバイスにDMAされる
  - NIC が、RDMA と DMA の載せ替えをしてくれるイメージ

- InfiniBand Verbs から使う：
  - ibv_reg_mr(): memory pinning
  - ibv_post_send/recv(): 通信要求を queue に積む
  - ibv_poll_cq(): completion queue を poll して完了確認

DCQCN によるロスレスネットワーク

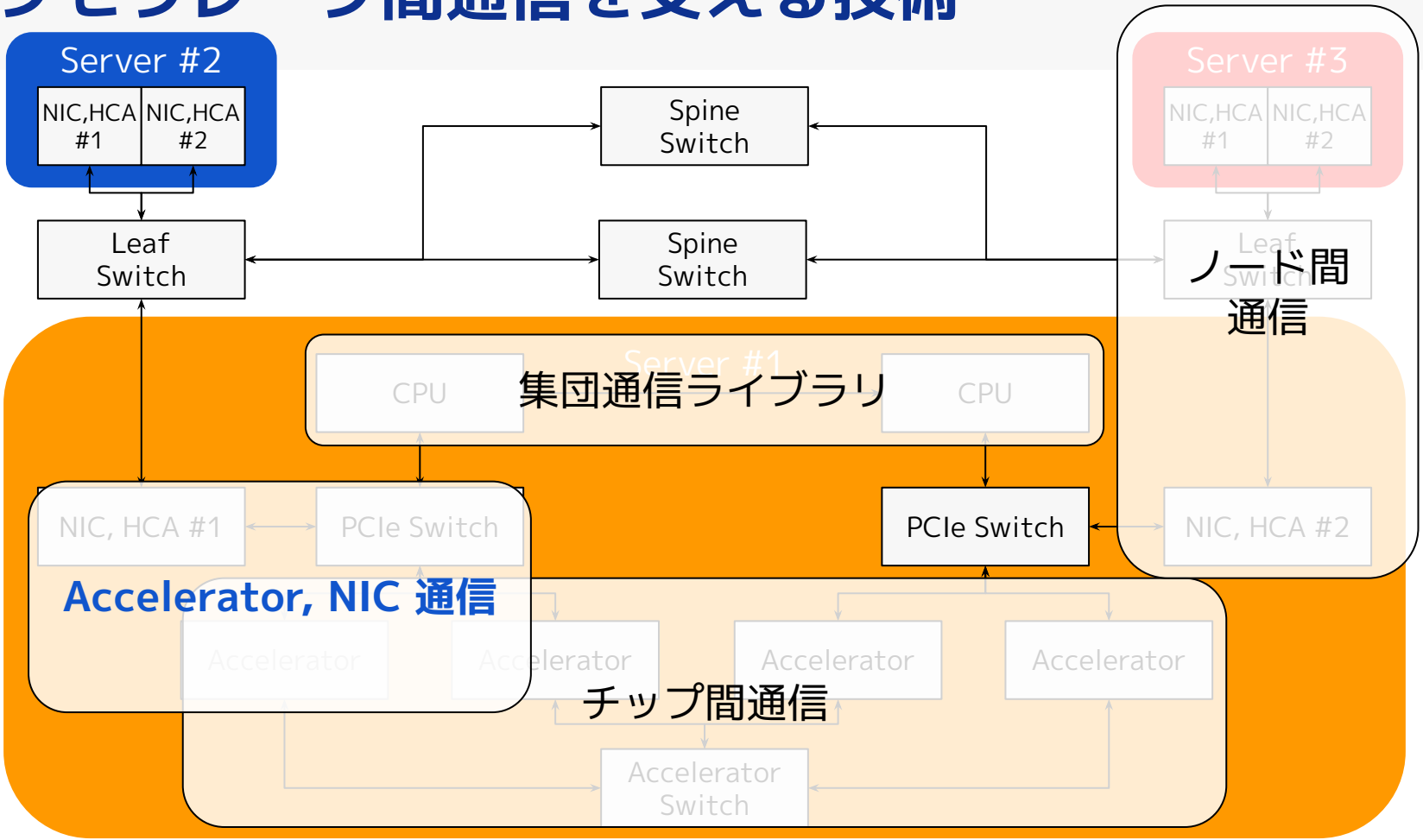
- Ethernet は、ロッシーなネットワークである
  - InfiniBand はロスレスなので、RoCEv2 のためにロスレス化が必要

- DCQCN: Data Center Quantized Congestion Notification
  - ECN と PFC の組み合わせで、ロスレスにすること

- ECN: Explicit Congestion Notification
  - 経路上のスイッチがパケットをいじって受信側に輻輳を通知
  - CNP (Congestion Notification Packet) で受信側が送信側に通知
  - CNP を受け取った送信側 (NIC) が送信ペースを下げる

- PFC: Priority-based Flow Control
  - ECN が不十分な時に、キュー単位で pause させる

アクセラレータ間通信を支える技術



Peer Memory Direct によるアクセラレータサポート

- カーネル空間で、ib_register_peer_memory_client() を使って、あるデバイス用の Peer Memory Client を登録しておく

- ibv_reg_mr() で、デバイスメモリを登録しようとすると：
  - 当該の Peer Memory Client が memory pinning する

- ibv_post_send(), ibv_post_recv() が呼ばれると：
  - 当該の Peer Memory Client が DMA アドレスを解決する
  - 実際にデバイス/NIC間のDMAを実行する

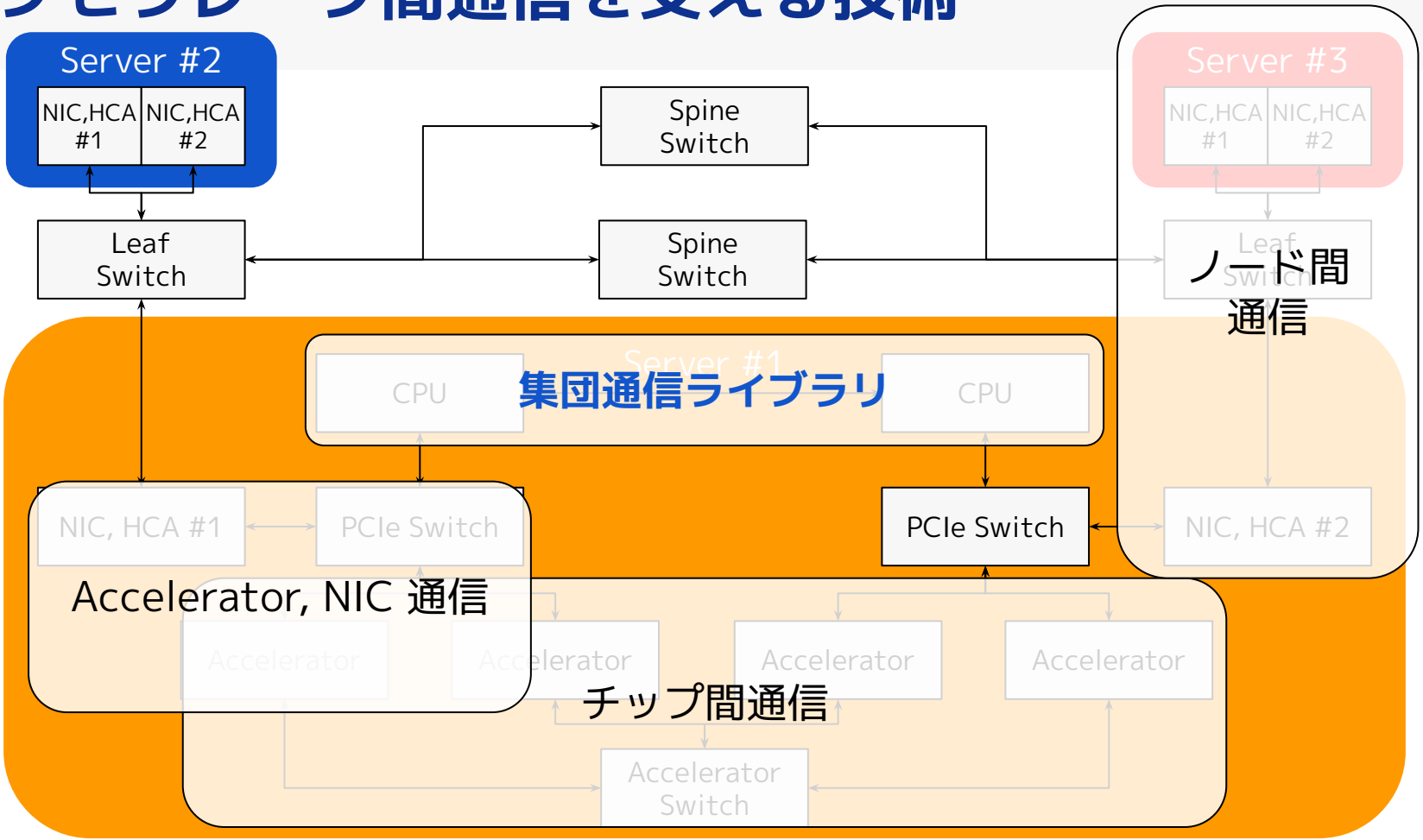
- 最近は DMA-BUF に移行が進んでいる。

Peer Memory Direct を使う上で注意すること

- 当該の Peer Memory Client がきちんとロードされているか？
  - NVIDIA の場合: modprobe nvidia_peermem したか
- PCIe Switch で DMA の折り返しができるか？
  - できない場合、Root Complex 折り返して無駄に帯域を消費する
  - いわゆる Access Control Service を無効にする
- システムのブロック図をみて、どういう経路を通るべきか確認する
  - GPU/NIC の affinity が悪いと、へんなところを通って遅くなる

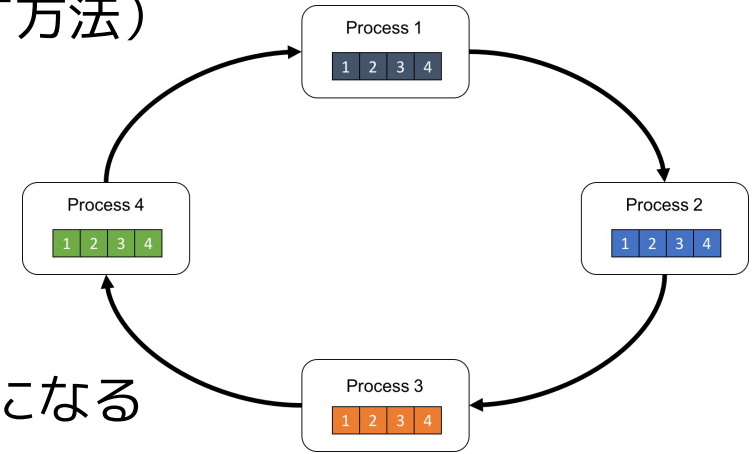
ブログ「KubernetesクラスタにおけるGPU-NIC割り当ての改善によるRDMAの高速化」で実例を紹介しました

アクセラレータ間通信を支える技術



Ring-AllReduce のアルゴリズム

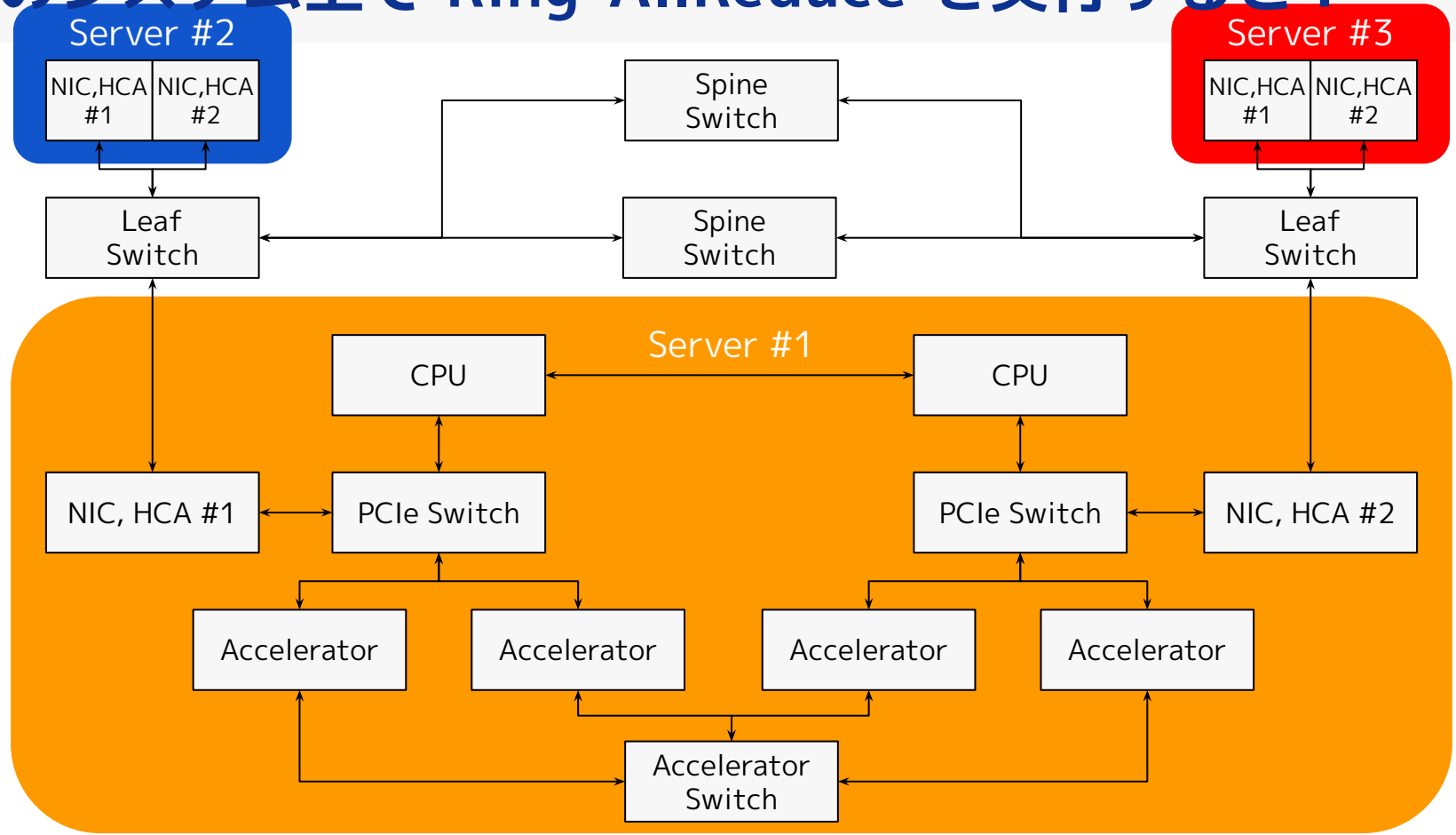
- Ring アルゴリズム（データをぐるぐる回す方法）
  - プロセスをリング状に並べる
  - Reduce-Scatter フェーズ
    - 配列を回しながら値を加算
  - AllGather フェーズ
    - 加算が終わった配列を回す
  - これで、全プロセスが総和を得たことになる



  - 1受信、1送信 なので、どのトポロジでも性能が出やすい方法

- Tree アルゴリズムもある
  - プロセス数に対し配列が小さいときに有利

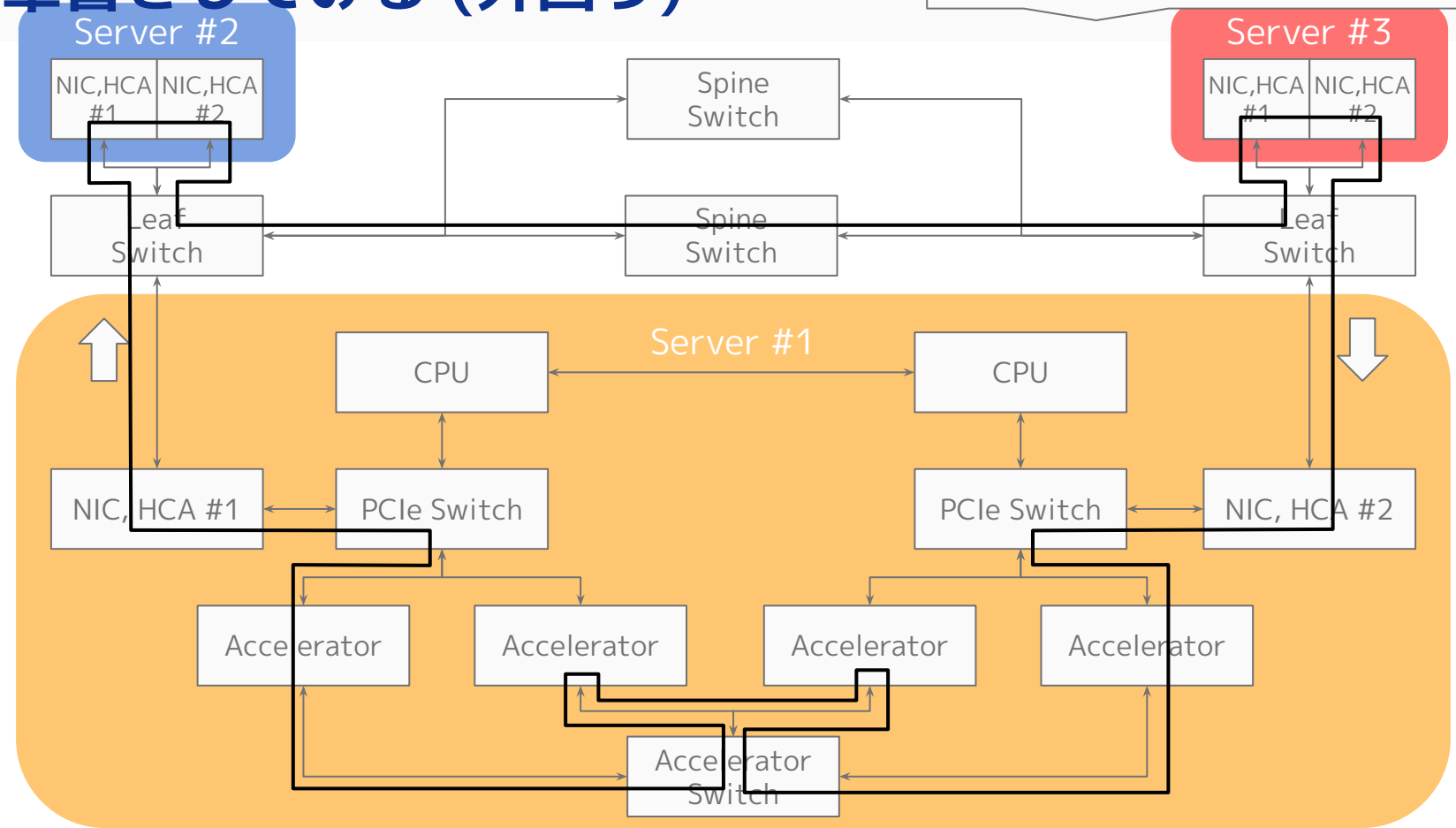
このシステム上で Ring-AllReduce を実行すると？



集団通信ライブラリ

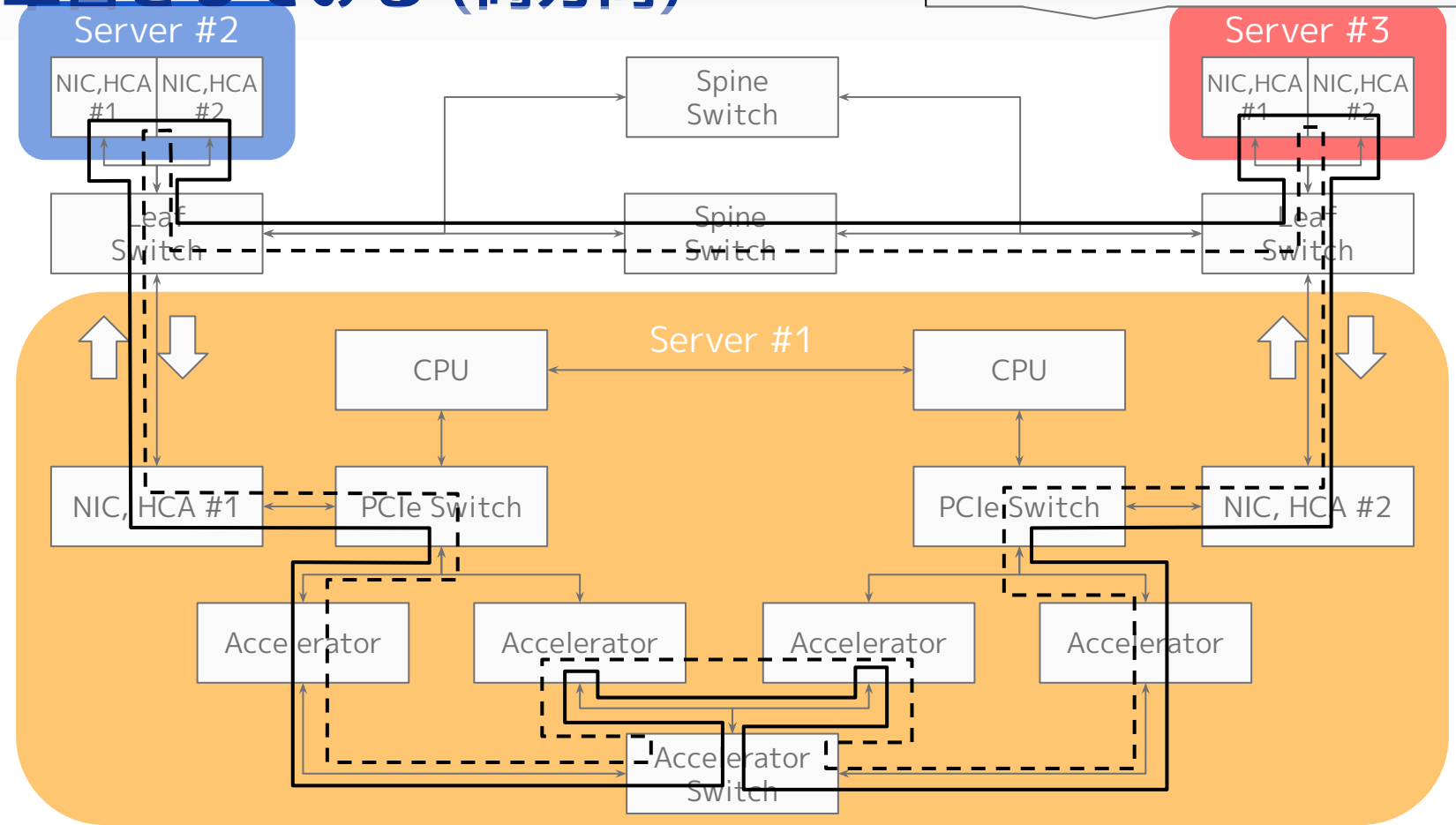
一筆書きしてみる (外回り)

ノード内にスイッチがあるとき
NICは片方向しか使っていない



集団通信ライブラリ
一筆書きしてみる (両方向)

Accelerator Switchは十分速い必要があるが
これでNICの全二重通信を使えた



NCCL の使い方

- ユーザがNCCLを初期化する
  - NCCL が、ノード内のトポロジや使えるハードウェアを検出する
  - デバッグに便利なログが出るので眺める（NCCL_DEBUG=INFO）
    - IB 有効で NIC 全て検出したか？ GPUDirect RDMA 有効か？

- ユーザが集団通信を開始する
  - サイズ、プロセス数によって適切なアルゴリズムが選ばれる（Ring/Tree, low-latency kernel 有無など）
    - 気に入らない場合は、環境変数やtunerで上書き設定できる

- 性能を理論値と見比べる
  - nccl-tests で表示される busbw を見るのが便利

今日の話: RoCEv2 を使ったアクセラレータ間通信

- なぜアクセラレータ間通信が必要なのか
  - 分散深層学習、とくに昨今のLLMの分散学習について
- 集団通信（アクセラレータ間通信）を支える技術
  - なぜ集団通信ライブラリが必要か
  - ノード内、ノード内からネットワークへ、ネットワーク
- **H100 クラスタ構築でのトラブルシューティング事例**
  - ノード内の課題と解決、ネットワーク内の課題と解決
- まとめ

生成AI向け機械学習クラスタ

構築自体の話は、Cloud Native Days Summer 2024
生成AI向け機械学習クラスタ構築のレシピ 北海道石狩編 へ！

- PFN / PFE では生成AIを学習するため、
  さくらインターネットさまの「高火力 PHY」を利用中
  - 高性能なGPUサーバ
    - NVIDIA H100 Tensor コア GPU x 8 搭載
      - 80 GB VRAM / GPU
  - ベアメタル
    - 低オーバーヘッド
  - インターコネクト（広帯域ロスレスネットワーク）
    - 400 Gbps x 4 (RoCEv2 対応)
    - シャーシスイッチ Arista Networks 7800R3

Preferred Networks

運用を開始したところ、問い合わせが発生...

- 実際にLLM学習中に発生した課題：
  - パイプライン並列化を実施すると、
    NCCL が自動選択する NIC が偏り、性能が出ない
    → ノード内のPCIe トポロジが怪しそう？ NCCL のバグ？

  - AllGather 集団通信性能が安定しない
    - カタログスペック通りの時もあれば、¼ 程度まで遅い時もある
      - AllReduce (Tree) については概ね良好であった

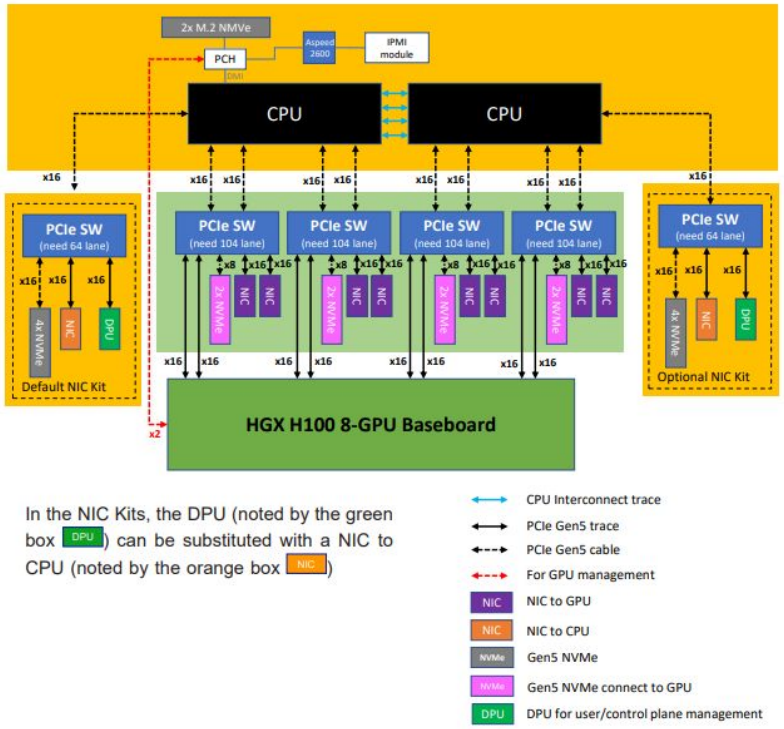
    → 原因不明

  - InfiniBand のタイムアウトエラーが頻発し、学習を継続できない
    → 原因不明

NCCL が自動選択する NIC が偏り、性能が出ない件

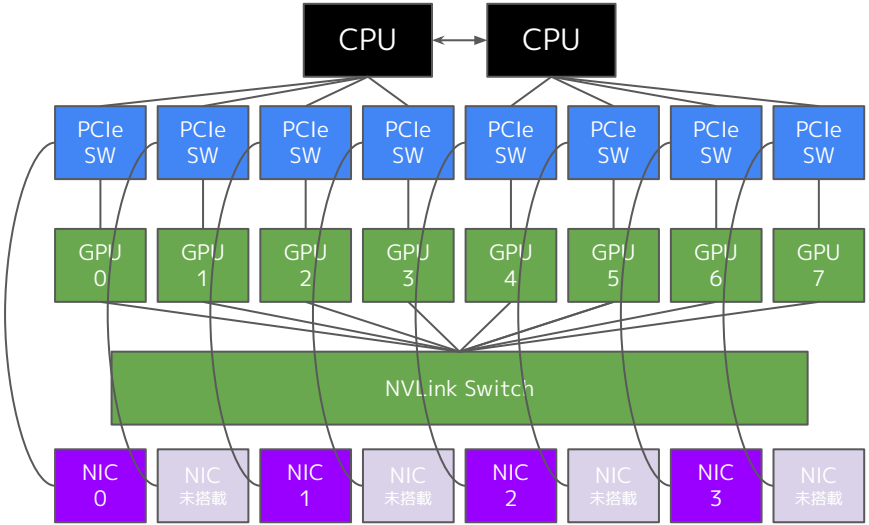
ノード内のPCIe トポロジを nvidia-smi で確認してみる

SMCI SYS-821GE-TNHR のドキュメント



実際の認識状況

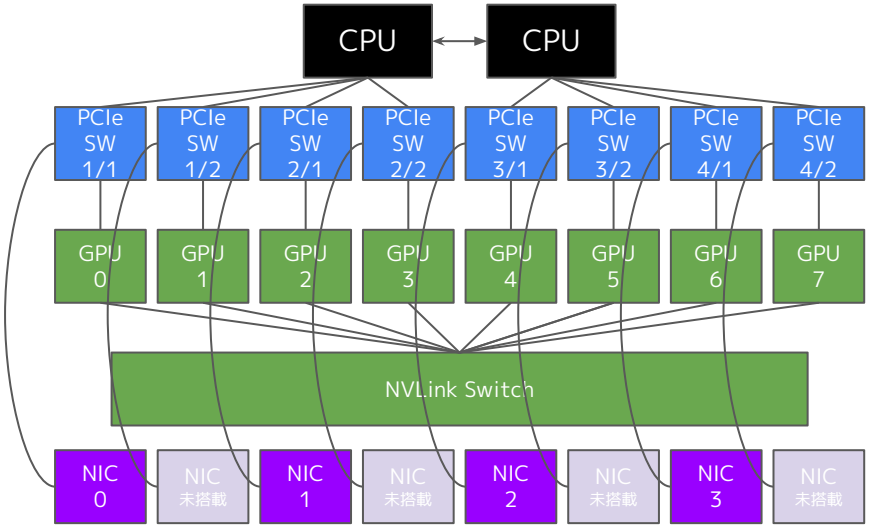
PCIe SWの数が 8個 で、ドキュメント (4個) と合わない



Preferred Networks

PCIe SW の数がなぜドキュメントと合わないか？

- ドキュメントでは 4個、実際には 8個 に見えている
  - **PCIe SW の論理的な分割機能による**
  - おそらくこういう理由による：
    - NIC 8枚搭載 を前提として設計しているため
    - CPU/GPU間の通信帯域を稼ぐため
      - PCIe SW の uplink は上限 x16
      - 分割で GPU の uplink を x16 にできる



- これにより、NCCL が自動選択する NIC が偏ってしまう
  - GPU1, GPU3 から見て NIC0, NIC1 の距離が同じ
  - GPU5, GPU7 から見て NIC2, NIC3 の距離が同じ
  - これにより、どちらも NIC0, NIC2 が選択される

- 今回のシステムはNIC 4枚搭載なため、論理分割すると NIC/GPU の間で RC 折り返しが必要になる
  - ワークアラウンドとして、**PCIe SW を統合して扱うファームウェアをリリースしてもらい適用した**
  - PCIe SW あたりの uplink 上限 が x16 なため、GPU あたりの uplink が半減するトレードオフがある
    - PCIe SW 間 link を発見して使えるようにする解決策を Broadcom が開発中らしい
      - Add new module to discover inter-switch P2P links

考えられる可能性を列挙してみる

- ノード内に問題がある可能性：
  - NVLink や PCIe のリンク速度の変化、省電力機能？
  - プロセスとCPU, Memory の割り当て（NUMA）周り？
  - メモリアドレスによって PeerDirect できないとか？

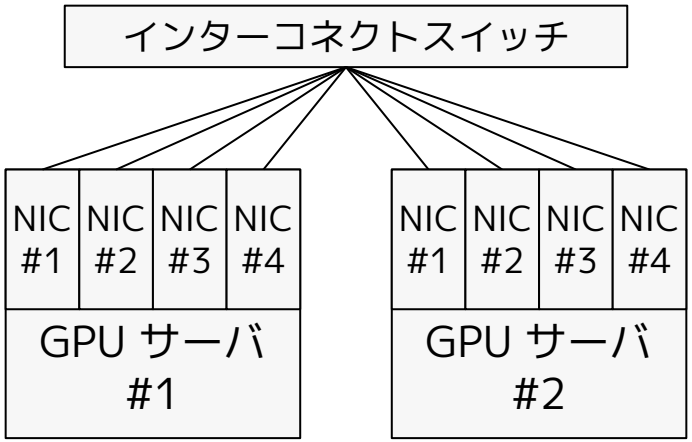
- インターコネクトに問題がある可能性：
  - DCQCN 関係の問題？
  - L1 の問題？

まずは、ノード内, インターコネクトの切り分けを実施することにした

ノード内, インターコネクトの切り分けと問題解決

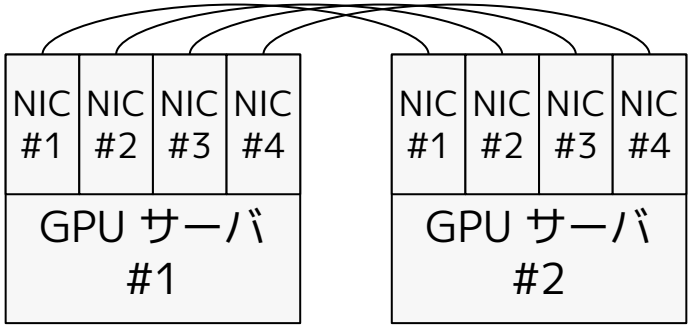
通常の構成

スイッチ経由で接続されている



切り分け用に準備した構成

2台のサーバをレール単位で直結してみる



- 2台のサーバを直結した結果、問題が再現しなくなった
  - インターコネクト側になんらかの問題があると考え Arista さまと相談したところ、**クレジット管理に関する修正コンフィグを提供いただき、問題が解決**

Preferred Networks

今日の話: RoCEv2 を使ったアクセラレータ間通信

- なぜアクセラレータ間通信が必要なのか
  - 分散深層学習、とくに昨今のLLMの分散学習について
- 集団通信（アクセラレータ間通信）を支える技術
  - なぜ集団通信ライブラリが必要か
  - ノード内、ノード内からネットワークへ、ネットワーク
- H100 クラスタ構築でのトラブルシューティング事例
  - ノード内の課題と解決、ネットワーク内の課題と解決
- **まとめ**

まとめ

- アクセラレータ間通信を使って、分散深層学習を高速に実施可能
  - LLM の台頭で、必要な通信パターンが複雑化してきている
    - AllReduce のみならず、複数の集団通信が重要に

- 高い並列化効率を達成するためには、高速な集団通信が必要だが…：
  - 関連技術を理解して、正しくシステム全体を設計・設定する
    - PCIe, NVLink, NCCL, 集団通信アルゴリズム, ECN/PFC, …
  - 実際のワークロードで検証を実施、地道なチューニングをしていく
    - 集団通信ライブラリのログ
    - 各種カウンタ
    - マイクロベンチマークと性能モデルとの比較

謝辞

- H100 クラスタのインターコネクト性能が安定しない件につきまして、以下のみなさまのご協力のおかげで解決できました。
  - さくらインターネット株式会社 さま
    - 定期的な情報共有、切り分け協力、ベンダさまとの連携、スイッチのカウンタ情報の提供をいただきました。
  - Super Micro Computer, Inc. さま
    - PCIe Switch のファームウェアの提供をいただきました。
  - Arista Networks, Inc. さま
    - クレジット管理に関するコンフィグの提供をいただきました。
- 本当にありがとうございました。

Preferred Networks

Making the real world computable