AI/ML基盤における800GbE スイッチ導入とその挑戦

JANOG56 Day2

株式会社サイバーエージェント グループIT推進本部 CIU

小障子 尚太朗(Shotaro Koshoji)

疋田 紅樹(Koki Hikida)

小障子 尚太朗 (Shotaro Koshoji)

- **2019/4 株式会社日本レジストリサービス 新卒入社**
  DNSサーバーや周辺ネットワークの運用
- **2021/12 株式会社サイバーエージェント 中途入社**
  CIU Platform Div NWエンジニア
- **JANOG歴: JANOG44～**
- **担当: プライベートクラウドのネットワーク系業務全般**
  - AS24284運用(Peering・拠点間接続)
  - IaaS基盤・ GPU基盤のNW設計・構築・運用
  - 上記の監視運用・自動化など

疋田 紅樹(Koki Hikida)

- **2024年４月 株式会社サイバーエージェント 新卒入社**

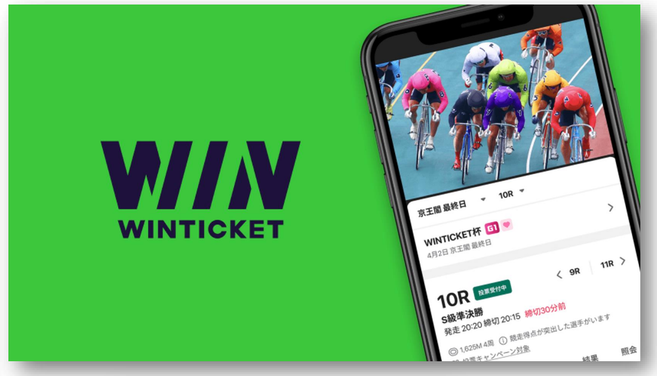
  CIU Platform Div NWエンジニア

- **JANOG歴**
  - JANOG52 長崎
  - JANOG54 奈良
- **業務内容**
  - AS 24284の運用
  - プライベートクラウドに関するDCNWの運用
    - Iaas基盤の構築・運用
    - GPU基盤の構築・運用
  - 上記NWに関わる監視・自動化など

サイバーエージェントの主要事業と関連会社

主な事業

- メディア&IP事業(e.g. ABEMA)
- 広告事業
- ゲーム事業(e.g. Cygames)
- その他



CIU（CyberAgent group Infrastructure Unit）は、
CyberAgentグループ全体のインフラを支える組織です

Cycloud というブランドでプライベートクラウドを展開しており、
IaaSとしてのOpenStackやKaaSであるAKEなど様々なサービスを提供しています

サイバーエージェントのプライベートクラウド

Cycloud

Compute

AKE
GKE相当のKubernetes as a Service

Cycloud Compute
VM、ブロックストレージ、ロードバランサー、ファイアウォールなどのIaaS機能を提供

ML Platform
GPUが利用可能なコンテナプラットフォーム、および機械学習基盤

Cycloud Run
プライベートクラウド Cycloud 向けに提供するサーバーレスサービスです。

Storage

CDB
マネージド MySQL サービス

S4
AWS S3に互換性をもったオブジェクトストレージ

CI / CD

hosted runner
安価かつ強力なスペックを持つGitHub Actions Runner

Container Registry
コンテナレジストリ

Platform Management

IAM
Cycloud Platformに統合されたアクセス制御を提供するサービス

Cycloud CLI
Cycloud Platform のためのコマンドラインインターフェイス

Tools

Terraform Provider
Cycloud Platform のための Terraform Provider

action-configure-credentials
クレデンシャルレスにCycloud Platformを操作するための GitHub Action

action-setup-cycloud
Cycloud CLI をインストールする GitHub Action

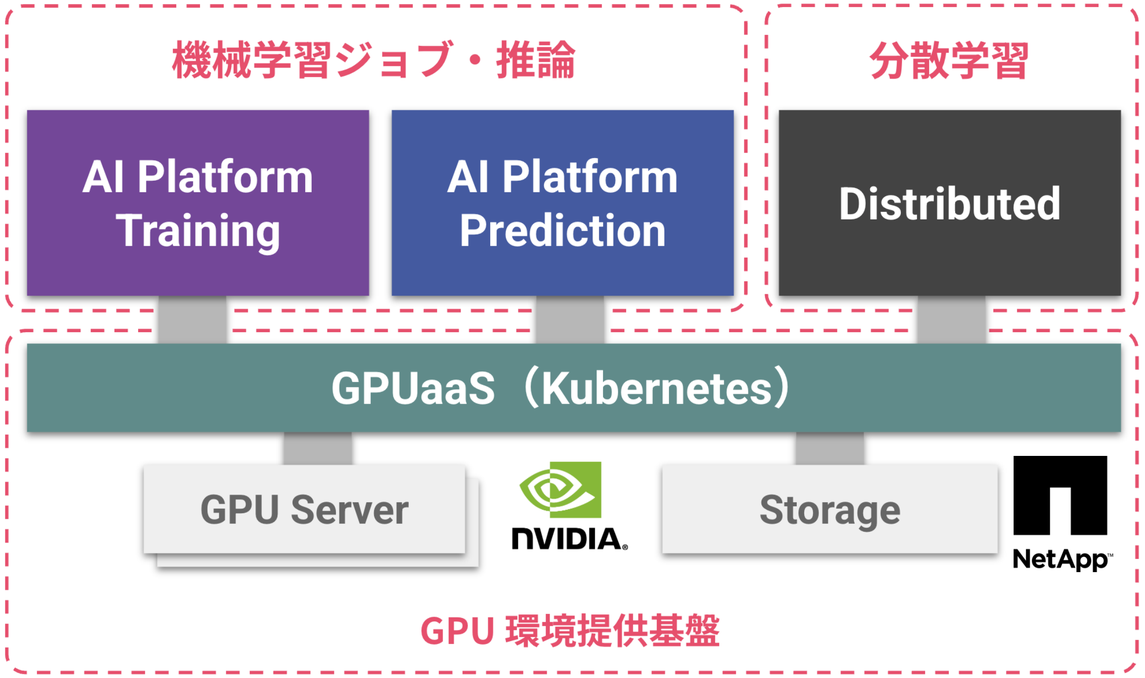
機械学習基盤 ML Platform

CIUが開発する機械学習基盤のサービス総称

- GPU as a Service(GPUaaS): GPU 搭載の Jupyter Notebook 環境やカスタムコンテナ環境のインスタンスを提供
- Training: 機械学習モデル構築基盤
- Prediction: サーバーレス推論基盤
- Distributed: 分散学習基盤

社内のML/DSエンジニア向けに開発

- プロダクト開発・運用コストの削減
- GPU利用ハードルの低減と導入の推進



ML Platform Distributed

2023年7月頃から安定稼働中

GPUサーバーネットワーク構成

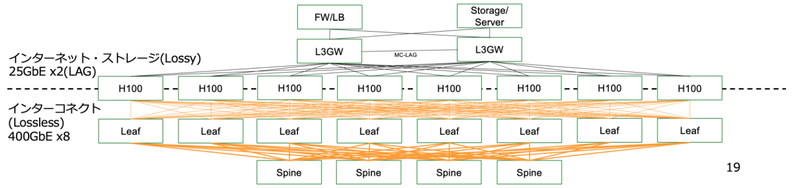
- Internet + Storage: 25GbE x2
- Interconnect: 400GbE x8
  - Rail-optimized Topology
  - Full Bisection Bandwidth
  - Adaptive Routing

GPU需要の増加に伴い 2024/12頃に増設計画開始

Distributedについて
https://cadc.cyberagent.co.jp/2023/sessions/distributed-ml-with-kubernetes/

ネットワーク設計

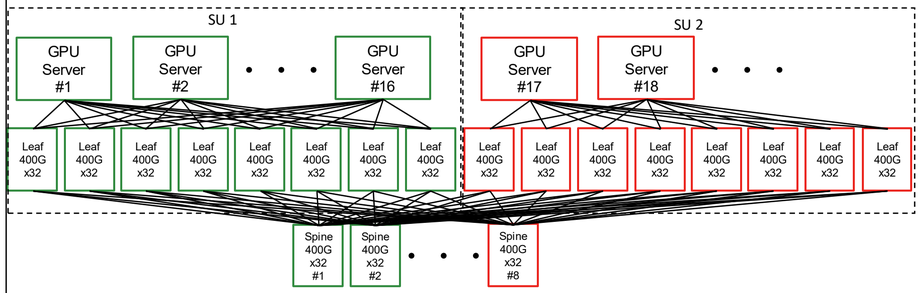
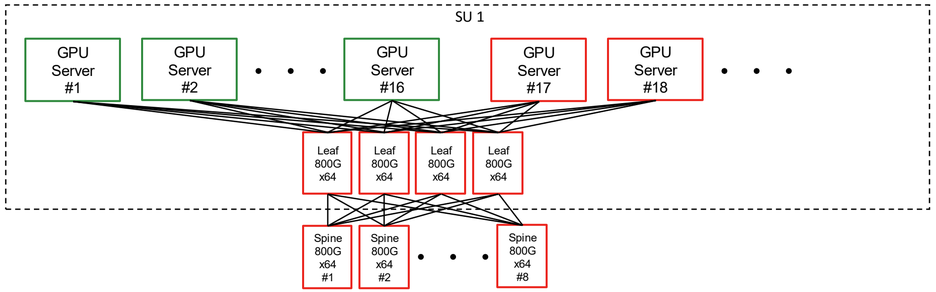
- **ラック構成はEnd of Row(EoR)を採用**
  - 1ラックあたり最大でGPUサーバー3台搭載となるためToR構成は収容効率が悪い
  - サーバーへの配線は全てパッチパネルを経由
- **要件の異なるネットワーク(LossyとLossless)を分離**



JANOG52での発表
https://www.janog.gr.jp/meeting/janog52/aiml400/

増設に向けたインターコネクト構成検討

既存インターコネクト増設と800Gスイッチで新たに再構築する方式を比較

	400Gスイッチ増設	800Gスイッチ導入
方式	• 400GbE 32ポートスイッチを12台増設 • Leaf 16台 + Spine 8台	• 800GbE 64ポートスイッチのインターコネクトを新規構築 • 既存インターコネクト収容機器も全て移行 • Leaf 4台+Spine 3台
メリット	• 既存のスイッチの活用 • 構成変更が最小限	• Scalable Unit(SU)あたりのGPUサーバー収容数増加 • 密度向上によるラックスペースの削減
デメリット	• SUを跨ぐGPU間通信が増える • SUを跨ぐ通信品質の担保が難しい • 多くのラックスペースが必要(最低12U以上)	• 800Gスイッチは当時は新しい技術ではあったので検証必須
構成図 (赤枠: 新規機器)		

→ 400Gスイッチ増設のメリットはほぼ無く、800Gスイッチを導入する方向で検討

増設に向けての課題と対応方針

- **課題1. 800G構成はコストが課題**

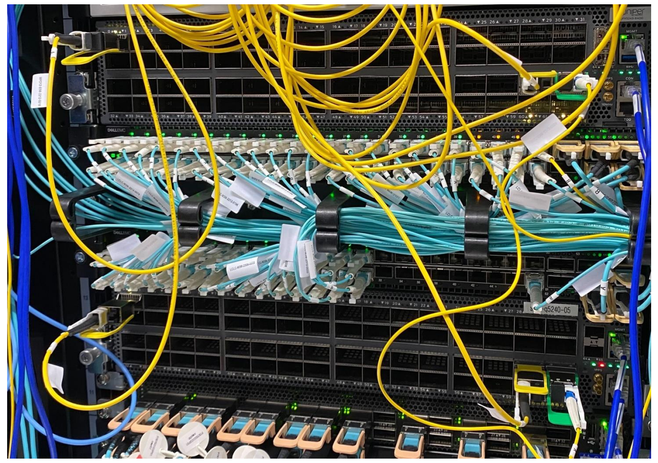
  玉突きで遊休資産となる予定の400Gスイッチを活用したい

- **課題2. インターコネクト用のラックスペースが多く必要**
- **課題3. インターコネクトの利用状況が正確に把握できていないので可視化したい**

- **対応案1. 400G/800G混在構成にしてコストを最適化**
- **対応案2. パッチパネルの高密度化(後半疋田より説明)**
- **対応案3. マイクロバーストモニタリングを実装(後半疋田より説明)**

800Gスイッチの検証

Juniper QFX5240(800Gスイッチ)のベータ機を借用・ベンダーと協力し検証を実施



稼働中のネットワークラックに押し込んで検証

検証での確認項目

- 400Gスイッチ(SN4700) / NIC(ConnectX-7)との接続性
- 400Gスイッチ(SN4700)とのUnnumbered eBGPの接続性
- サーバ間のIP疎通性
- RDMAトラフィックの疎通性
- DCQCN(PFC/ECN)の大まかな動作(ECN・CNP伝搬、 Losless queueing、 PFCカウンタの確認)
- flowlet-based DLBの設定と大まかな動作確認

<u>混在構成が構築・運用可能なことを確認</u>

400G/800G混在構成の課題

Spine-Leaf間のトラフィック制御がかなり難しい

- OS/ASICが変わるのでのDCQCN周りのパラメーターチューニングが難しい(キュー/バッファの扱いなど)
- Adaptive Routing/Dynamic Load Balancingの混在が可能なのか？→ 後半疋田より説明

クラスタの規模的にできるだけ学習通信はSpineを通らないで欲しい

- 1SUのため、Spine経由のトラフィックはほぼゼロにできるはず
- 増設前でもほとんどがLeaf折り返しの通信だった

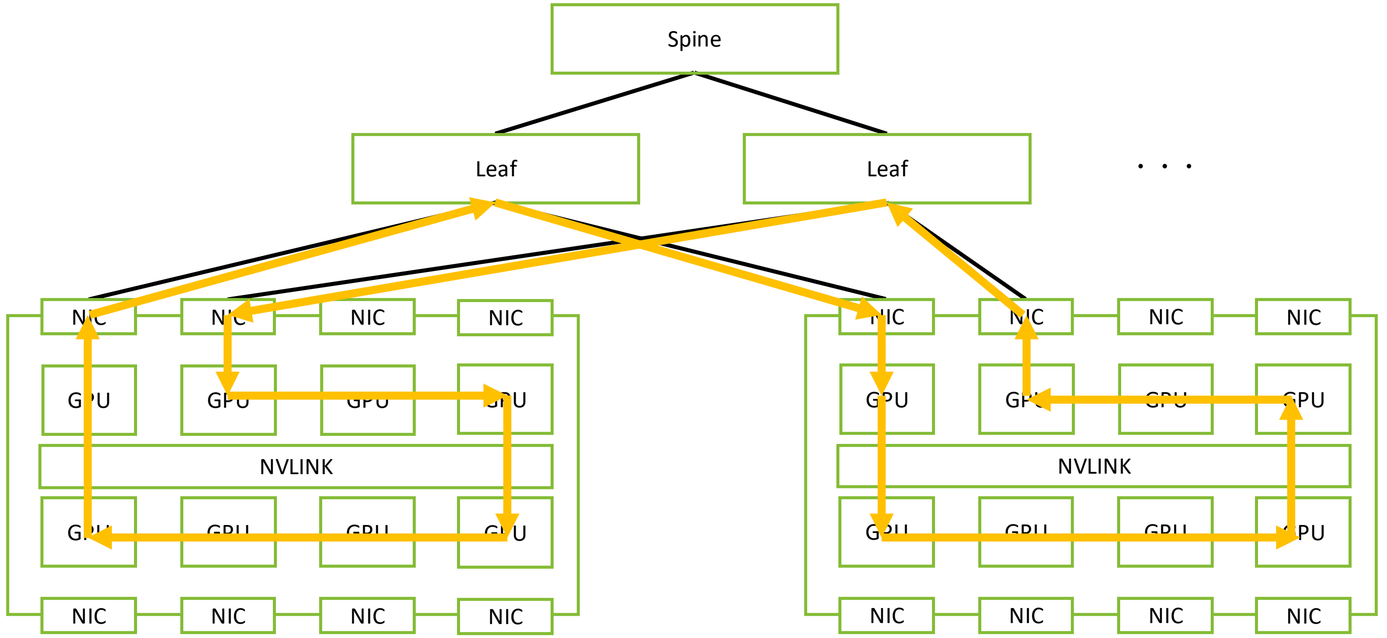
どのような条件で通信がSpineを通るのか調査を実施

分散学習の経路調査(AllReduce 2node)

分散学習実行時にどのようなトポロジが形成されているかをnccl-testsの実行ログから確認

2ノードではLeafに閉じたリングをNCCLは形成する

→ 3ノード以降を確認



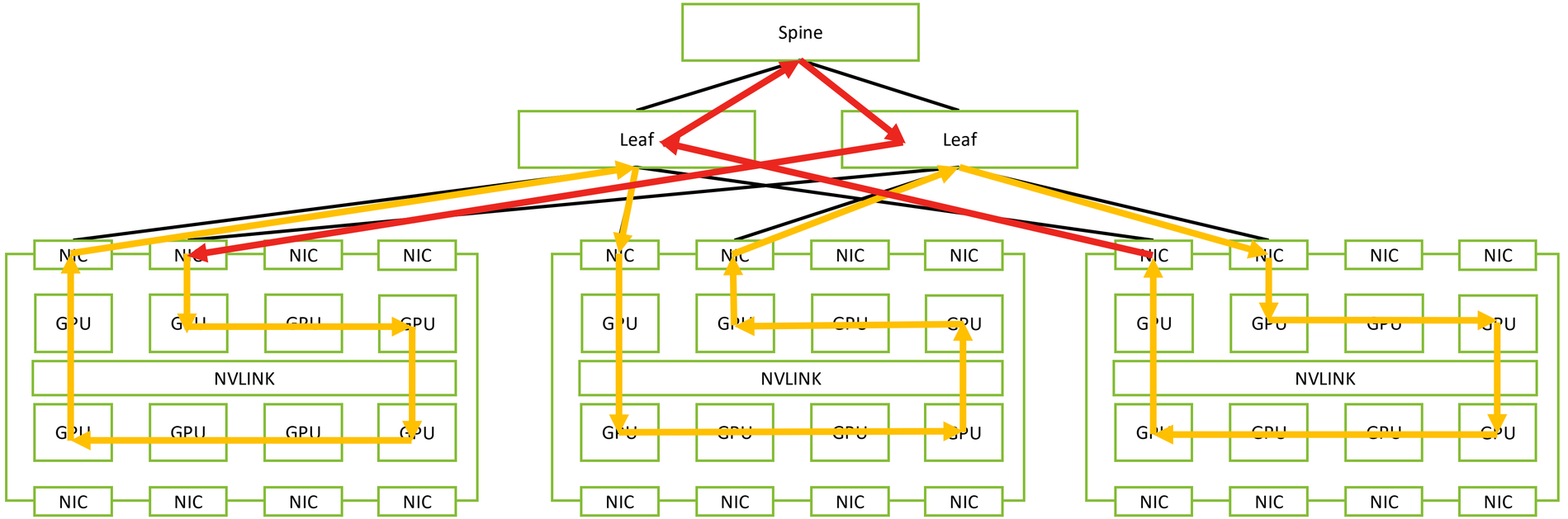
2nodeの場合の通信経路(1リングのみ抜粋)

分散学習の経路調査(AllReduce 3node)

3ノードの場合はSpine経由のリングが形成される

4,5,6...ノードと増やして確認したところ、奇数ノードの場合は3ノード同様の挙動を示した

→ 奇数ノード払い出すパターンがあるので困る😭



3nodeの場合の通信経路(1リングのみ抜粋)

NCCL_CROSS_NIC

NCCL_CROSS_NIC という環境変数でNICの使い方を操作できることが判明

→ 同じリングで同じNICを使ってほしいので0を指定してみる

NCCL_CROSS_NIC

The `NCCL_CROSS_NIC` variable controls whether NCCL should allow rings/trees to use different NICs, causing inter-node communication to use different NICs on different nodes.

To maximize inter-node communication performance when using multiple NICs, NCCL tries to use the same NICs when communicating between nodes, to allow for a network design where each NIC on a node connects to a different network switch (network rail), and avoid any risk of traffic flow interference. The `NCCL_CROSS_NIC` setting is therefore dependent on the network topology, and in particular on whether the network fabric is rail-optimized or not.

Values accepted

0: Always use the same NIC for the same ring/tree, to avoid crossing network rails. Suited for networks with per NIC switches (rails), with a slow inter-rail connection. Note that if the communicator does not contain the same GPUs on each node, NCCL may still need to communicate across NICs.

1: Allow the use of different NICs for the same ring/tree. This is suited for networks where all NICs from a node are connected to the same switch, hence trying to communicate across the same NICs does not help avoiding flow collisions.

2: (Default) Try to use the same NIC for the same ring/tree, but still allow for the use of different NICs if it would result in a better performance.

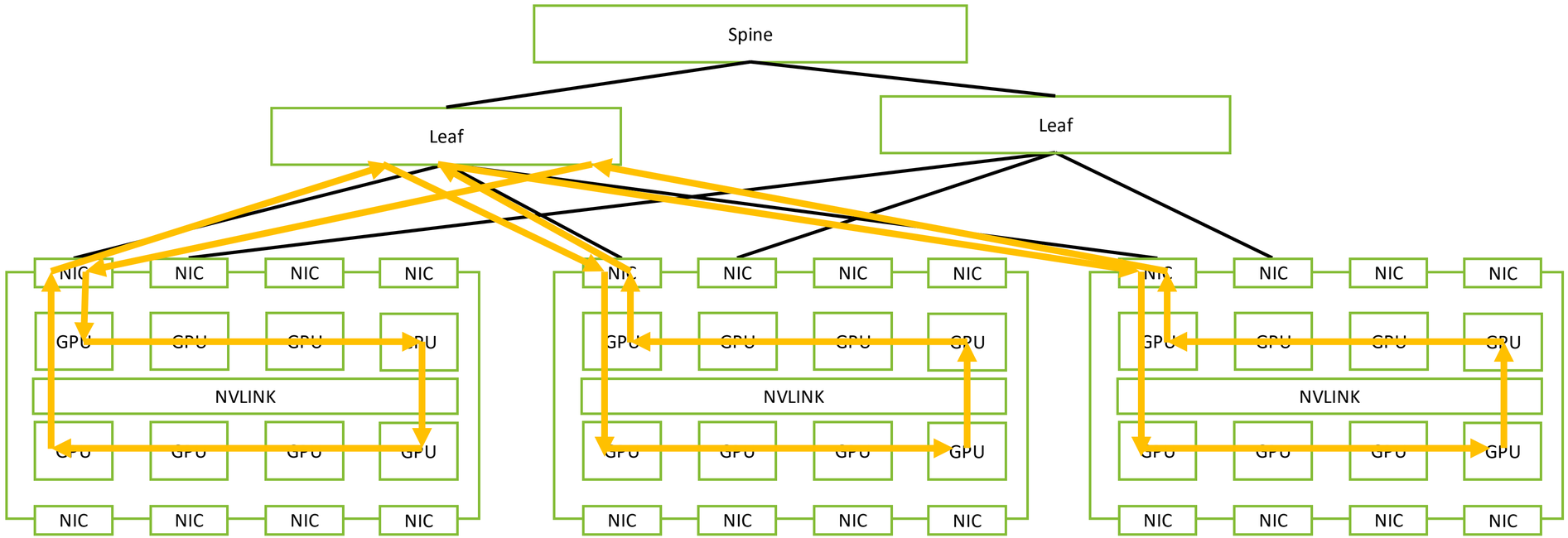
https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/env.html#nccl-cross-nic

分散学習の経路調査(AllReduce 3node,NCCL_CROSS_NIC = 0)

想定した通りリングがLeafに閉じる様になった 🎉

性能測定を行っても性能劣化は見られなかったのでデフォルト設定に組み込むことに

→ Spineは使われなくなるので不要なのでは？



Spineスイッチを排除できるのか

Cycloud MLPlatform Distributedの場合、サービス形態の都合上難しい

→ 分散学習実行時のオプション選択はユーザーに委ねられている

- 環境変数は埋め込んでいるが実際に使われるかどうかはアプリケーションの実行形態に依存
- Spineを通る通信の原因を調査すると適切な環境変数が使われていなかったパターンが多い

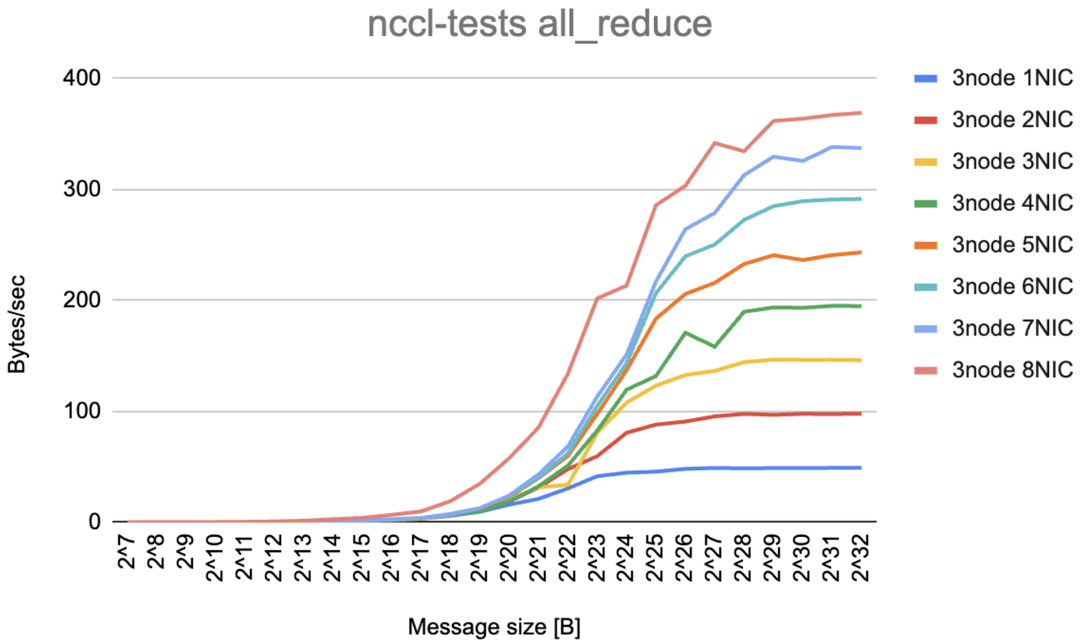
基本はLeafに閉じるつもりでSpineスイッチはコストを抑えつつ用意する方針に

ボトルネックにならないチューニングも実施（後半疋田より説明）

GPUサーバのNIC枚数の削減検討

- **H200 3node NIC1～8枚のnccl-testsの結果を比較**
- **NICは多いほどベンチマークの結果は良い**
  - 性能が出なくなるリスクがあるので8枚を選択
  - ワークロードに合わせてNICを削減できる？

皆さんNICの枚数どうやって決めていますか?



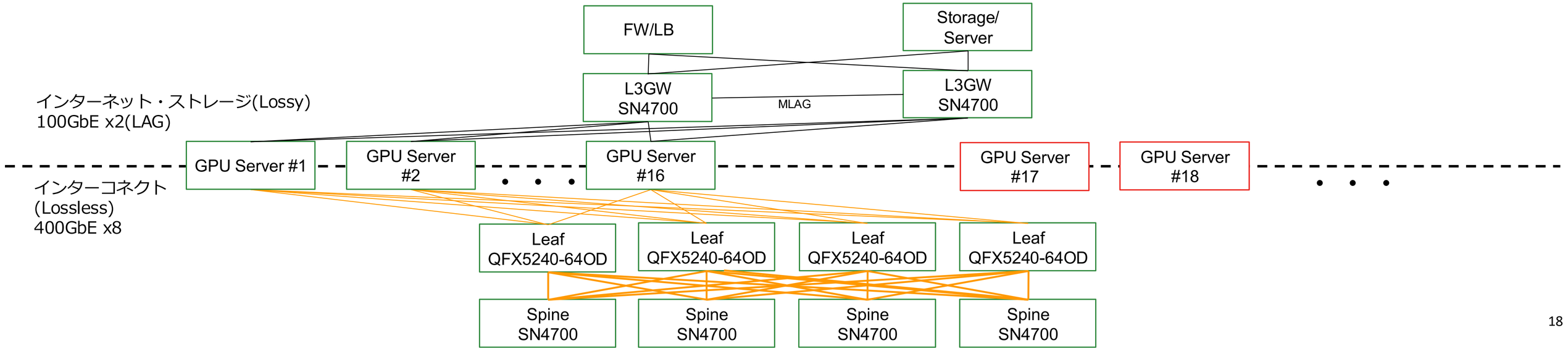
NIC枚数毎のnccl-testの結果比較 ※ 各パターン1回のみの実行

GPU増設後のネットワーク構成

400G/800Gの混在構成でインターコネクトを構築

- **Rail-Optimized Topology**
- **downlink 7: uplink 1の Oversubcription**
- **1SUでH100/H200を56台まで拡張可能**

GPUサーバーの台数/トラフィックが増えた場合はLeaf/Spine/uplinkを増設・増強していく



構築完了！と思いきや性能が出ない問題発生

2種類のGPUサーバーを混在させて分散学習を実行したところ性能劣化が発生

MLPlatformでは調達の都合上、同じGPUでも数種類のGPUサーバーが存在

性能劣化の原因: Spine - Leaf 間に想定以上のトラフィックが発生

Spineは通らないようにしたはずだが、何故…

NVIDIA DGX H100

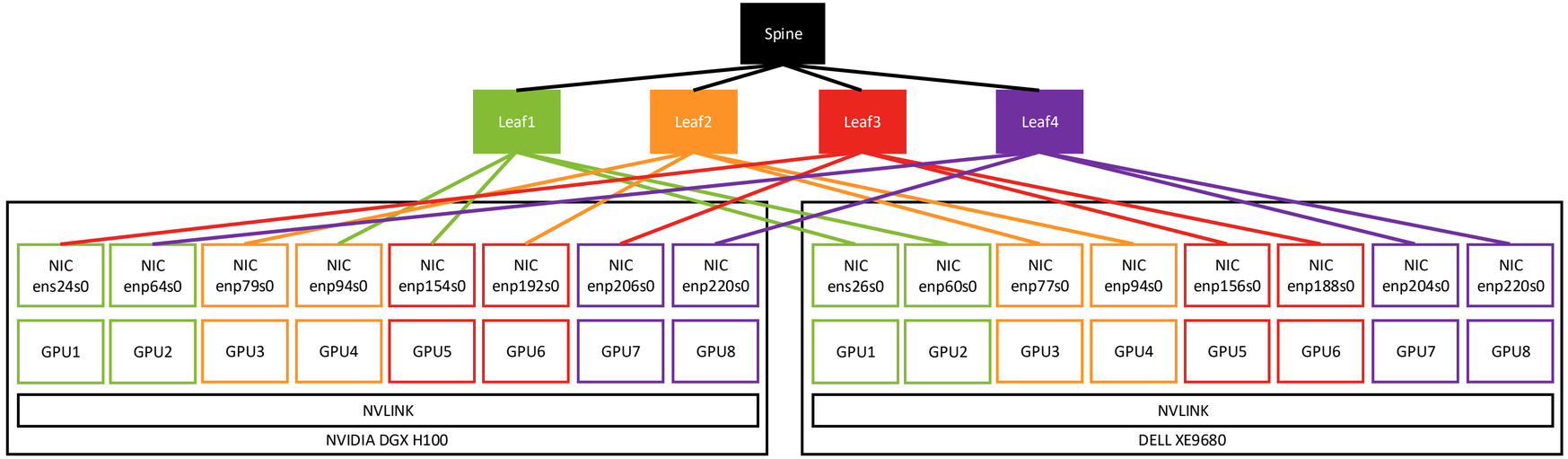
DELL XE9680

インターコネクトの調査

NCCLのログからGPU/NICとLeafスイッチの組み合わせがズレており、Spine経由のトポロジが形成されていることが判明

- **片方のサーバーは物理ポート番号に合わせた結線となっていた**
- **Rail Optimized TopologyはGPU番号毎に同じLeaf(Rail)に接続する必要がある**

  Full bisection時代は帯域に余裕があったため問題になっていなかった

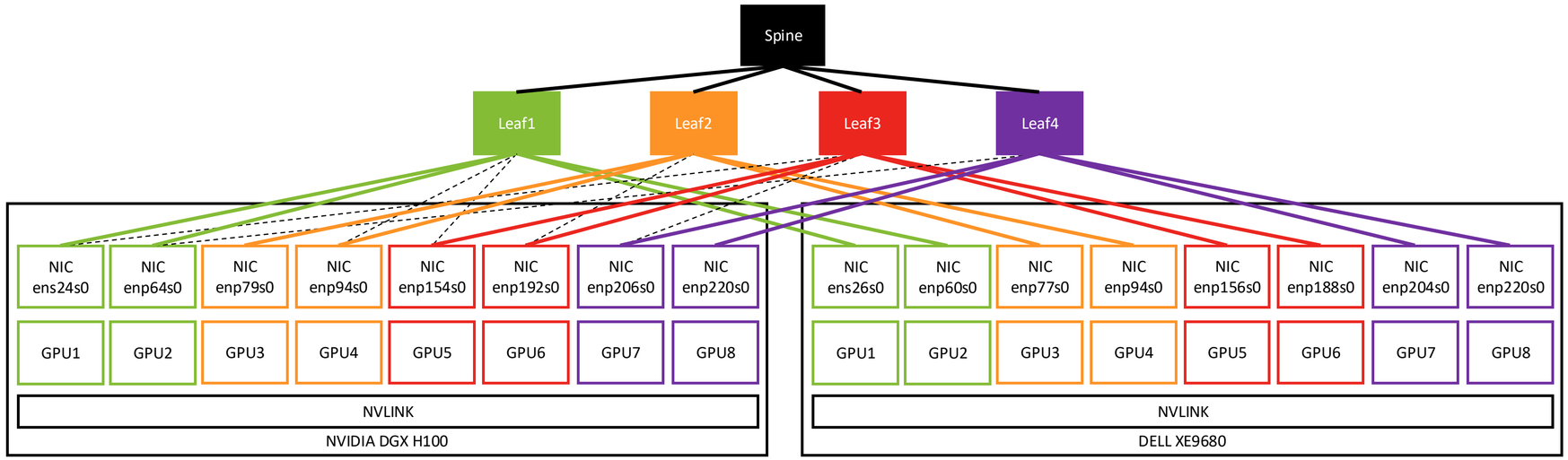


インターコネクトの接続変更

GPUサーバーのトポロジに合わせてインターコネクトを接続変更

→ 2種類のGPUサーバをミックスした際の分散学習性能が想定通りに 🎉

<u>サーバーによってはGPU/NICの見え方が異なる。複数種のサーバーでクラスタを構築する場合はNIC/GPUがどのように認識されるか要確認</u>



無事にTOP500へランクイン

2025年6月に発表された世界的なスーパーコンピューターのランキング TOP500 で 国内15位（世界132位）に登録されました🎉

132	**Cycloud ML Platform** - PowerEdge XE9680, Xeon Platinum 8480+ 56C 2GHz, NVIDIA H100, 400g Ethernet, Ubuntu 22.04.5 LTS, Nvidia CyberAgent, Inc. Japan	37,376	10.82	16.97

実際には...
System: DELL PowerEdge XE9680 / NVIDIA DGX H100
GPU: NVIDIA H100 / H200
Interconnect Switch: Juniper QFX5240-64OD / NVIDIA SN4700

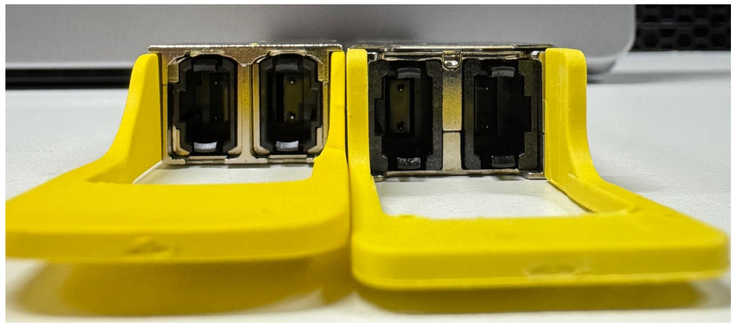
物理とチューニングの話

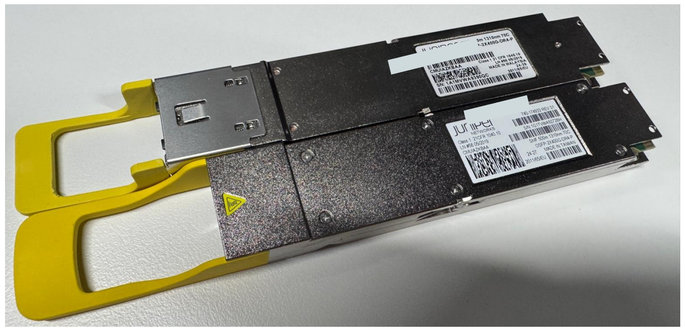
800Gスイッチの導入

H100/H200のインターコネクトではサーバーからの400G 8ポートを多数収容する必要がある

➡ 2U 800G 64ポートのスイッチを、2U 400G 128ポートのスイッチとして活用

トランシーバー1本で400G 2ポートを接続可能な
OSFP 800G-DR8(2x400G-DR4)を採用





800Gスイッチを導入して分かったこと

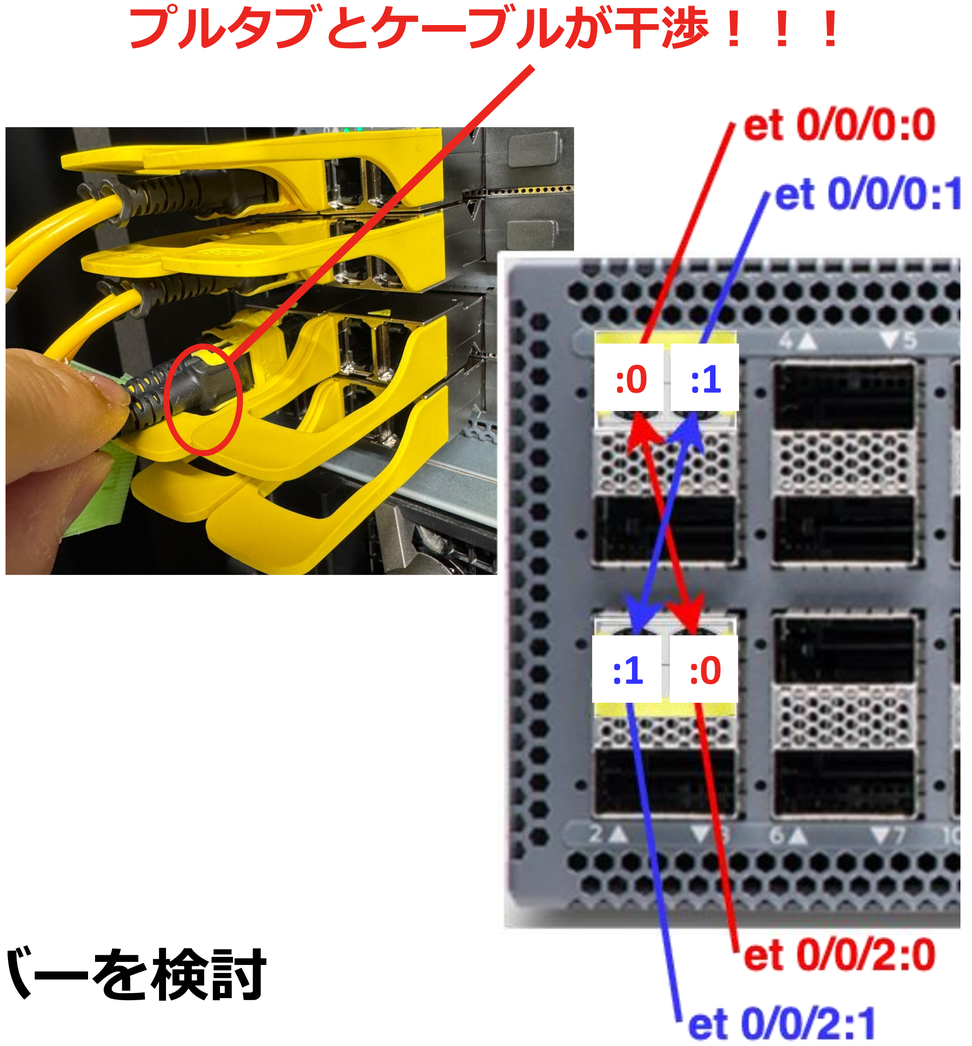
MPOケーブルの抜線時にトランシーバーのプルタブとMPOケーブルが干渉

- **抜線時に意図せずトランシーバーが抜ける事故発生**
- MPOコネクタの限界…？

  今後はMPOコネクタ以外の標準化の検討が必要…？

インターフェースの順番が直感的ではない

- スイッチの上2段と下2段でインターフェース番号が入れ替わっており**配線ミスが発生**



2MPOトランシーバーは運用性に難がある・・・

次回の導入時は 800G-DR8 1MPO トランシーバーを検討

パッチパネルの課題

要件1. ファシリティの都合で、Leaf/Spineスイッチとパッチパネルは1ラックに収容

要件2. GPUサーバーを設置しているラックから1ラックに対しMPO-12を20本配線

1ラックに数百本のMPOケーブルを収容する必要

1U MPO-12 32ポートパッチパネルの占有スペース

100本収容時：4U専有（GPUサーバー 12台 収容可）
200本収容時：7U専有（GPUサーバー 25台 収容可）
300本収容時：10U専有（GPUサーバー 37台 収容可）

800G 64ポート(400G 128ポート)のLeaf 4台では最大56台のGPUサーバーを収容可能

448本のMPOケーブルを収容する必要があり、MPOのパッチだけで14Uを専有
ケーブルマネージャー等を考慮すると20U以上がMPOパッチで埋まる計算に・・・

パッチパネル高密度化の検討

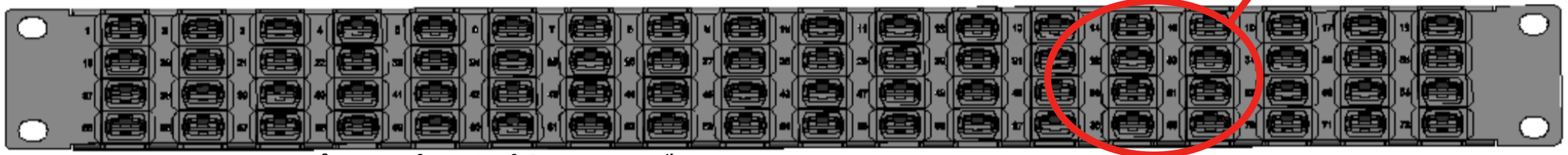
1U MPO-12 32ポートパッチパネルでは拡張性に難があり、高密度のパッチパネルを検討

高密度のMPOパッチパネルは運用性に難あり

- 抜線が高密度のため事故の元
- 清掃時にクリーナーとケーブルが干渉する懸念
- ポート番号の確認が困難

MPOに代わるより小型なケーブルコネクタ導入を検討

MPOコネクタの密度が
高いため操作性に難あり

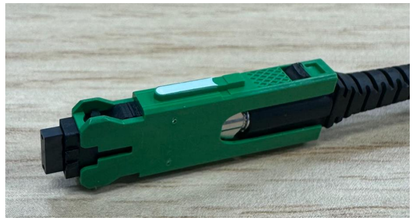


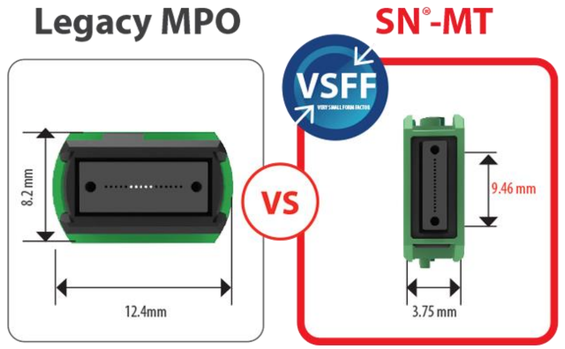
1U MPO-12 72ポートパッチパネルのモデル

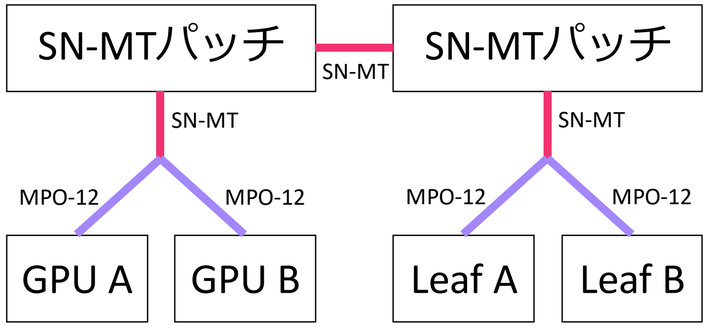
SN-MTコネクタ

特徴

- **Very Small Form Factor (VSFF)の一種**
- **MPOコネクタよりも小型のため高密度化が可能**
- **SN-MT(16芯)-MPO-12*2ブレイクアウト構成が可能**
  - トランシーバーは既存製品を使用可能







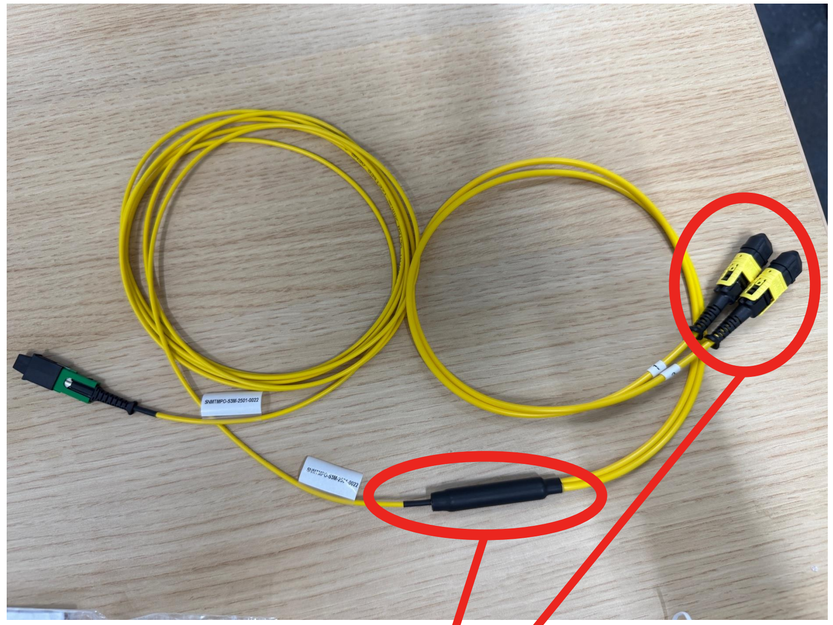
SN-MTコネクタは小型のため
高密度にしても作業性を維持

1U SN-MT 72ポートパッチパネルのモデル

SN-MTコネクタ導入の結果

導入で実現できたこと

- **SN-MTパッチパネルにしたことでポート収容密度を4倍**
- **ラック間ケーブルの本数を1/2**
- **高密度にしたときの作業性が向上**

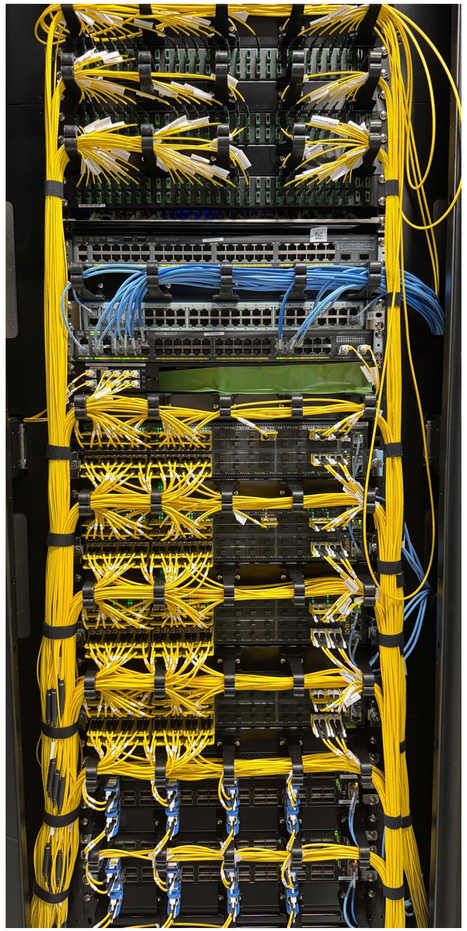


導入してつらかったこと

- **MPOとSN-MTのコネクタクリーナを使い分ける必要があり手間**
- **現場へ清掃や接続方法などの教育周知が大変**
- **ブレイクアウトケーブルのケーブリングが難しい**
  - 分岐部分が大きくラック内での取り回しが困難
  - GPUサーバーの8枚のNICの距離がバラバラで配線が難しい

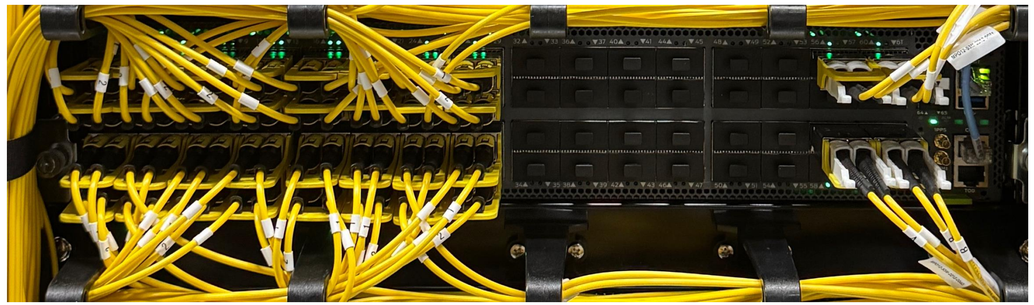
結合部とコネクタの長さを配線時に変更できないのでブレイクアウトケーブルは扱いが難しい

構築したラックの写真

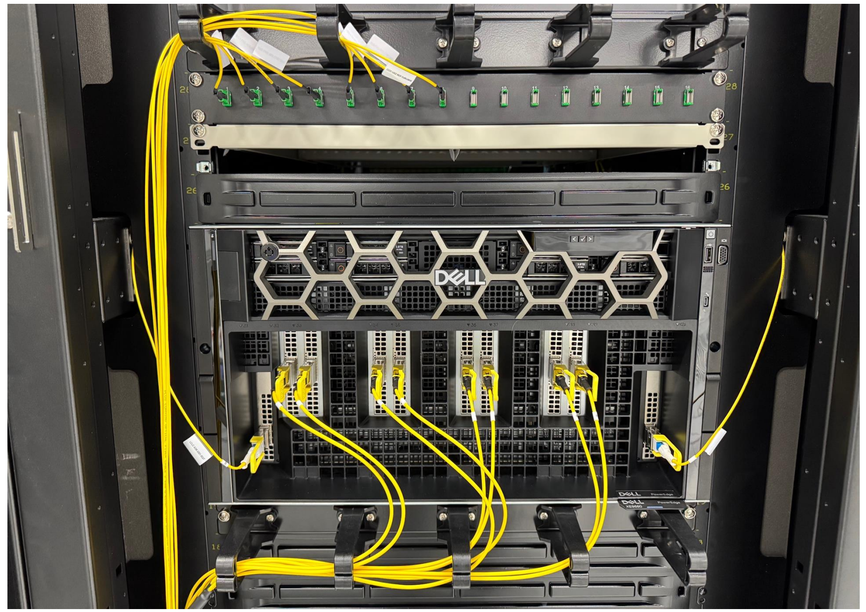


インターコネクトラック全体

SN-MTパッチパネル



800Gスイッチ



サーバー

物理とチューニングの話

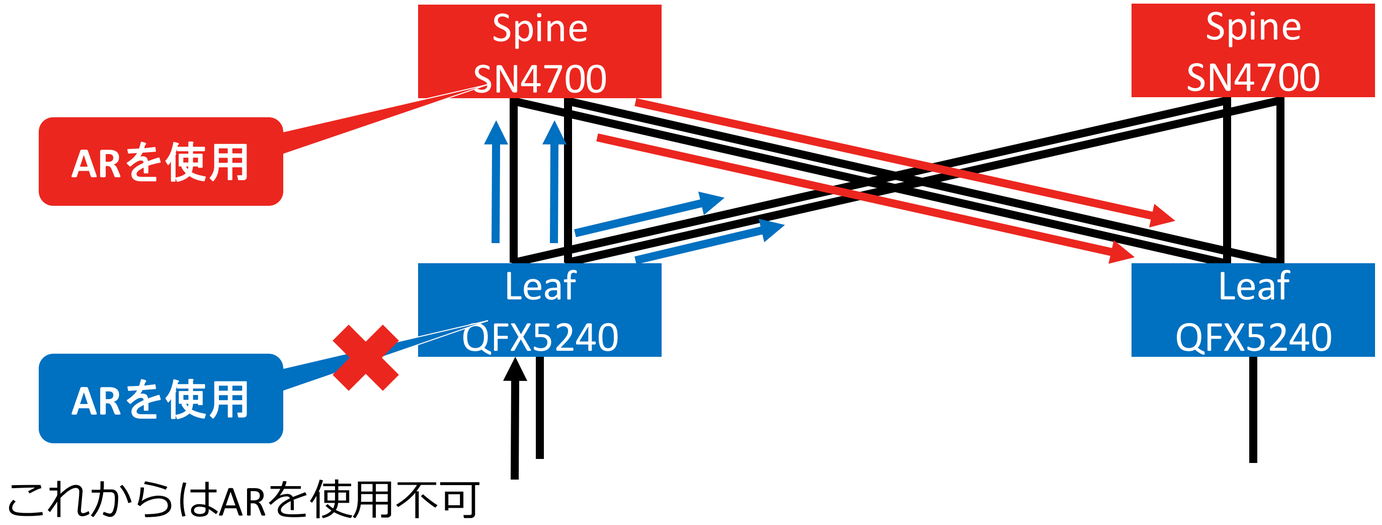
マルチベンダーLosslessネットワークの実現検討

増設前はリンクのフロー偏り防止にAdaptive Routing(AR)を利用

→ 今回導入したQFX5240はBroadcom ASICのため、ARが利用不可

同様の機能としてDynamic Load Balancing(DLB)が実装

これまで同様のフロー偏りの防止を実現するためには、ARとDLBを組み合わせる必要がある
→ <u>SN4700/QFX5240でARとDLBを用いた場合のパフォーマンス影響を確認</u>



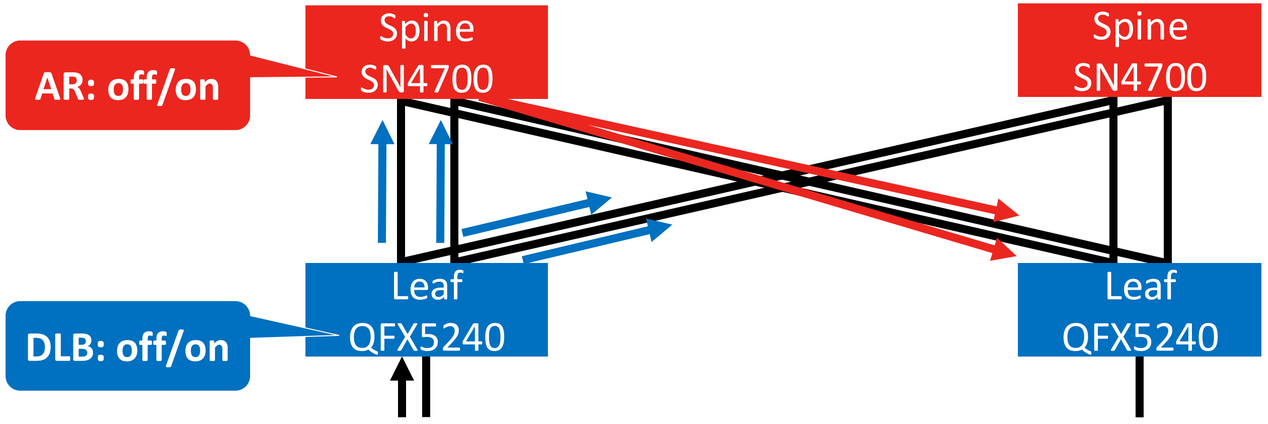
AR・DLB併用のパフォーマンス影響調査

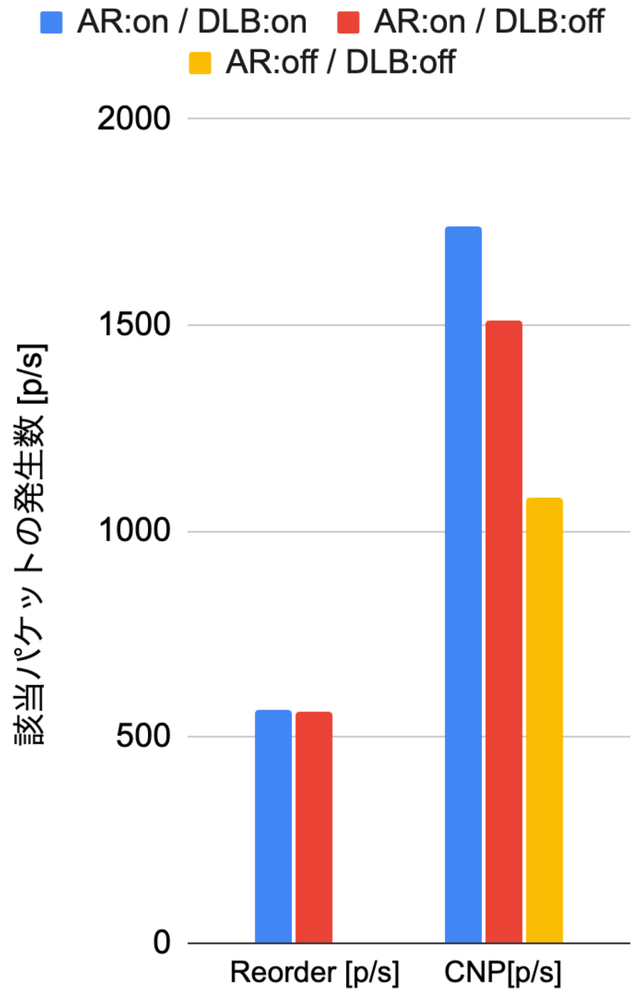
結果

- ① **AR: on DLB: on** → **reorder・CNPが多く発生**
- ② **AR: off DLB: on** → **①と比較して少ないが以前として発生**
- ③ **AR: off DLB: off** → **reorderは未発生、CNPも最も少ない**

AR・DLBのon/offのみでは輻輳が発生

on/offだけでは有効性を確認できなかった





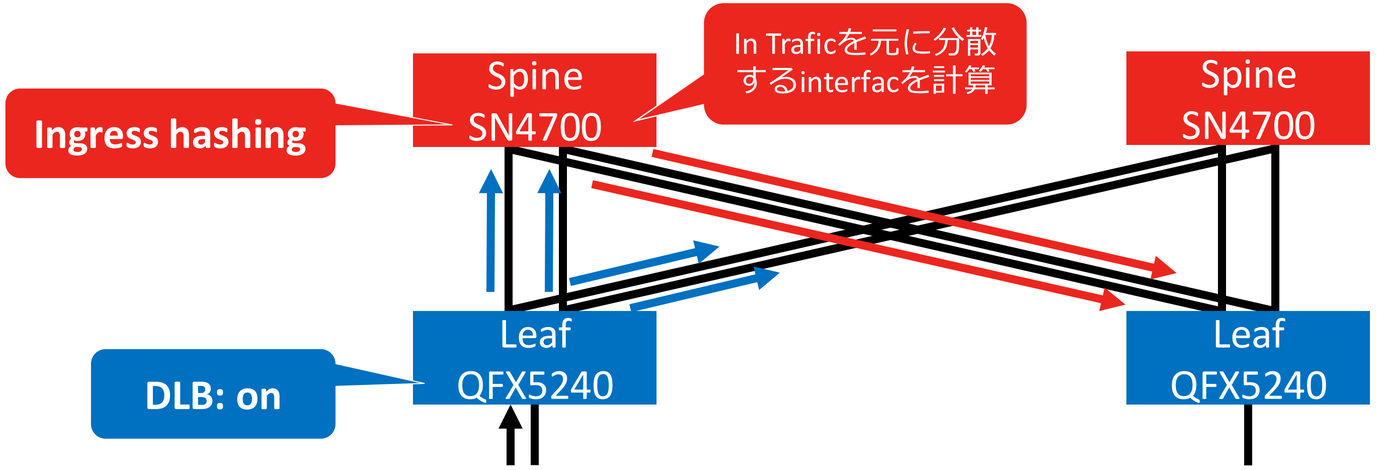
Ingress interface hashing

設定見直す中で生まれた仮説・・・

> **LeafでDLBが正常に動作していれば、Spineはトラフィックが入ってきたインターフェースに応じて分散するだけで十分なのでは？**

SN4700にはIngress interfaceに応じてECMPのhashを計算する方式が実装 (Ingress interface hashing)

→ **Ingress interface hashing とDLBの併用を検討**



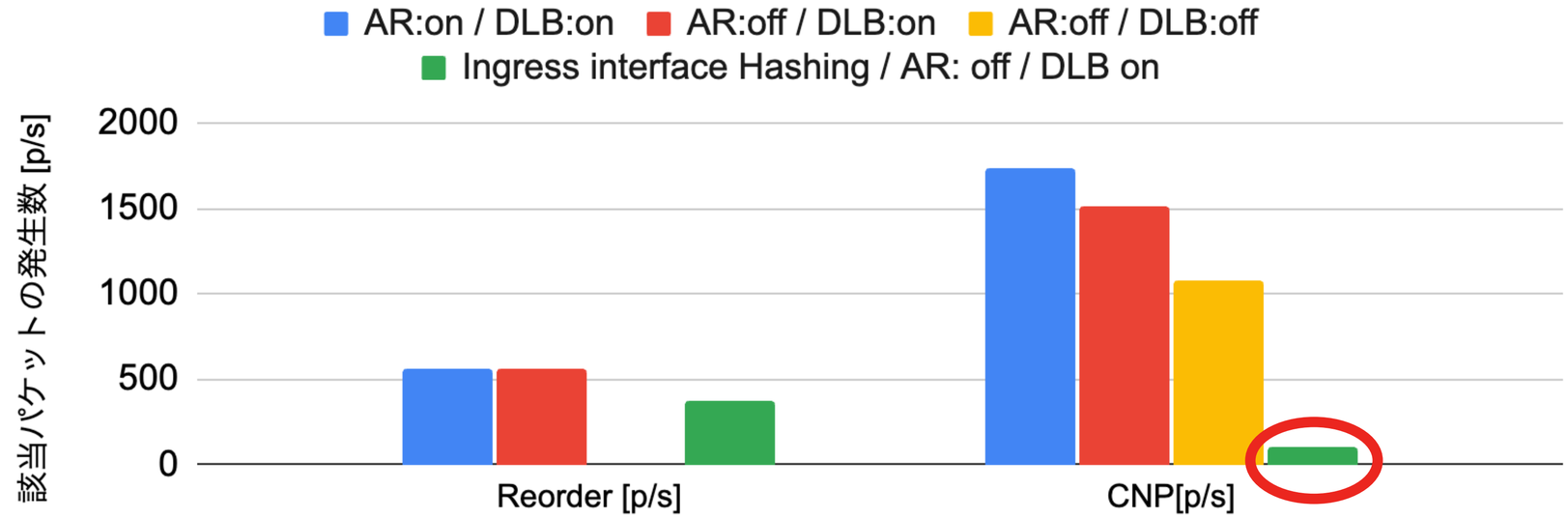
Ingress interface hashing+DLBの効果

結果

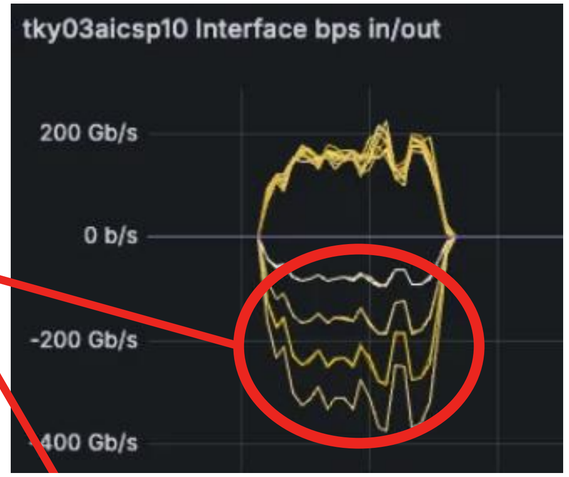
- **Reorderの発生がDLBの使用時と比べ低下**
- **CNPの発生率が大幅に低下**

Ingress interface Hashingの有効性を確認

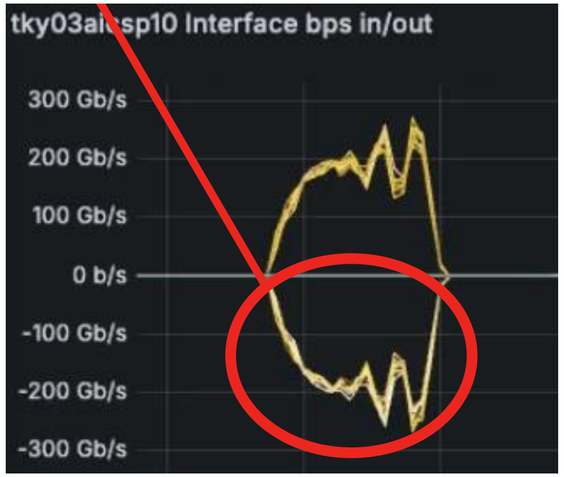
よりパフォーマンスを向上させるためにDLBのパラメータに着目



トラフィックの分散を確認！



DLB 有効/Ingress interface hashing 無効時のSpineのトラフィック



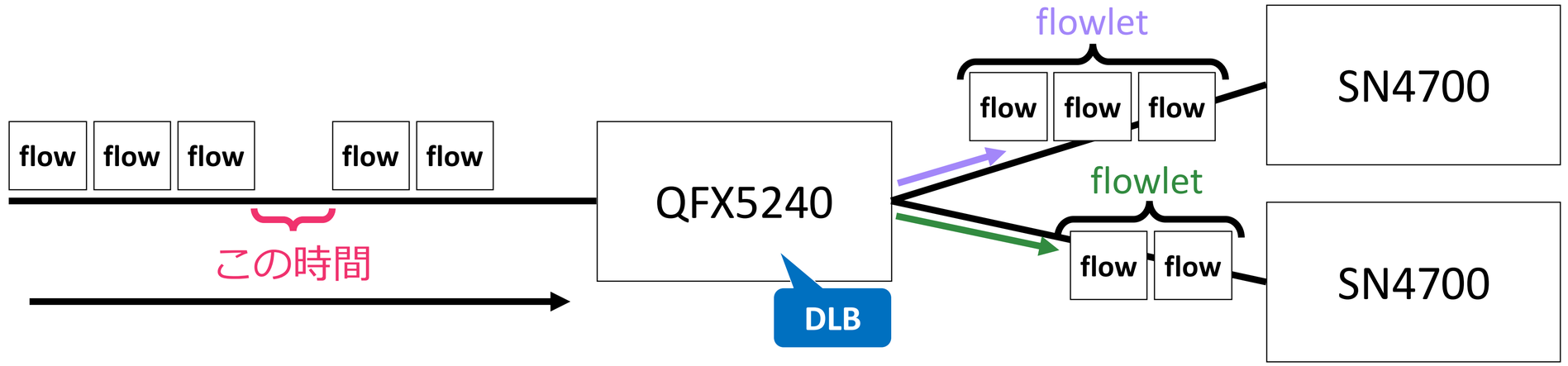
DLB 有効/Ingress interface hashing 有効時のSpineのトラフィック

inactivity-intervalによるパフォーマンス影響

DLBのパラメータのinactivity-interval **(**flowletを識別するための時間間隔**)が**

- **長すぎる場合：負荷分散効果が弱まり、パスが偏る**
- **短すぎる場合：わずかな差で新しい flowlet と判定され、Reorderが増加**

最適な値を見つけることでよりパフォーマンスが向上すると考えた



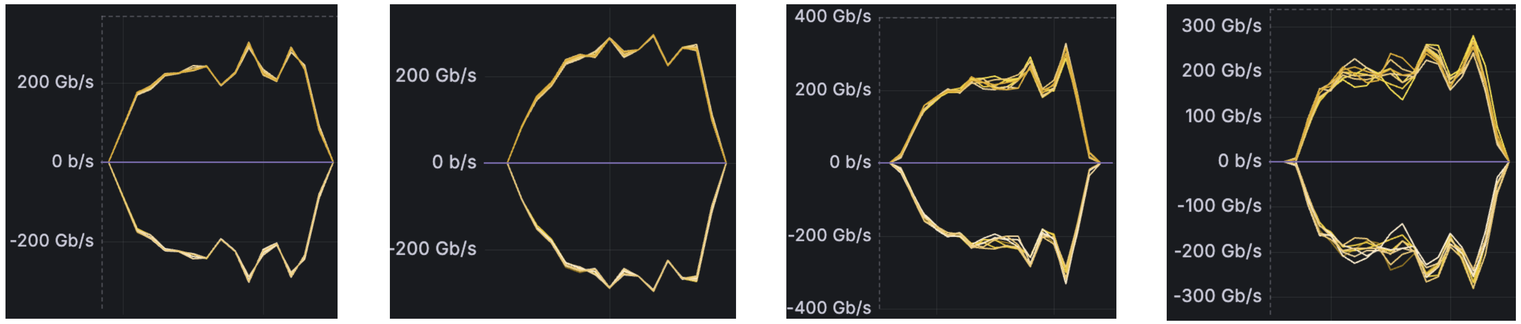
inactivity-intervalのパフォーマンス影響調査結果

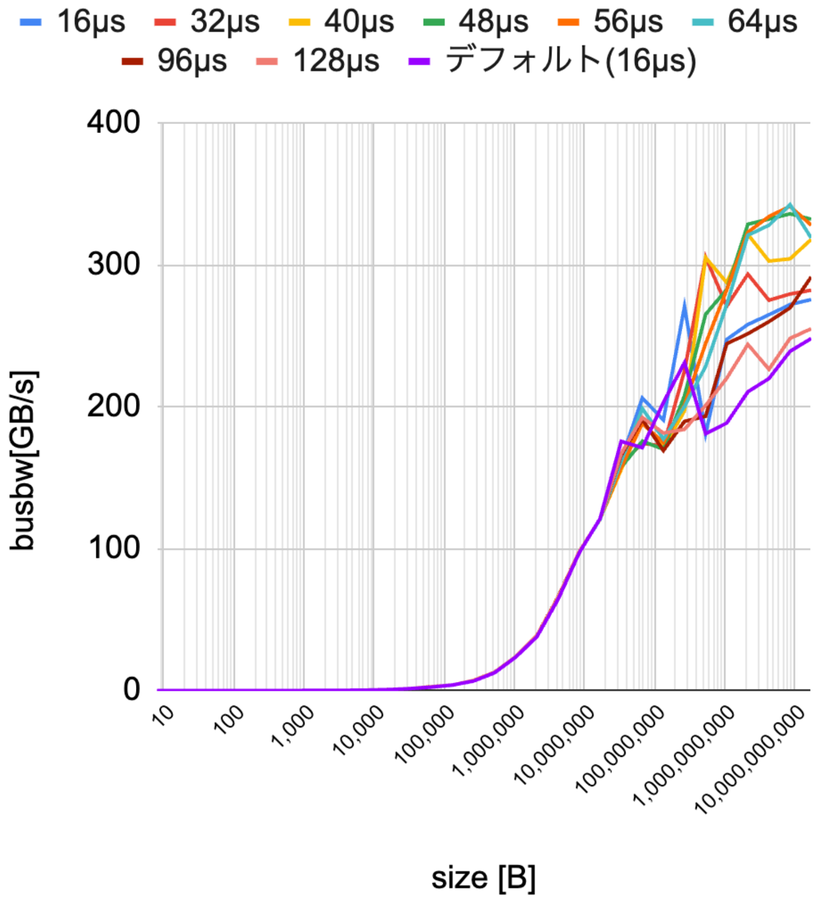
nccl-testsでのパフォーマンス測定結果

- 常に最良の結果を出す値は存在しない

本番環境では各sizeを比較した際に総合的に良い結果が出ている時間を使用

今後、自動的にパラメータを最適化できる機能が実装されたら導入したい





inactivity-intervalを長くした際のトラフィックの変化

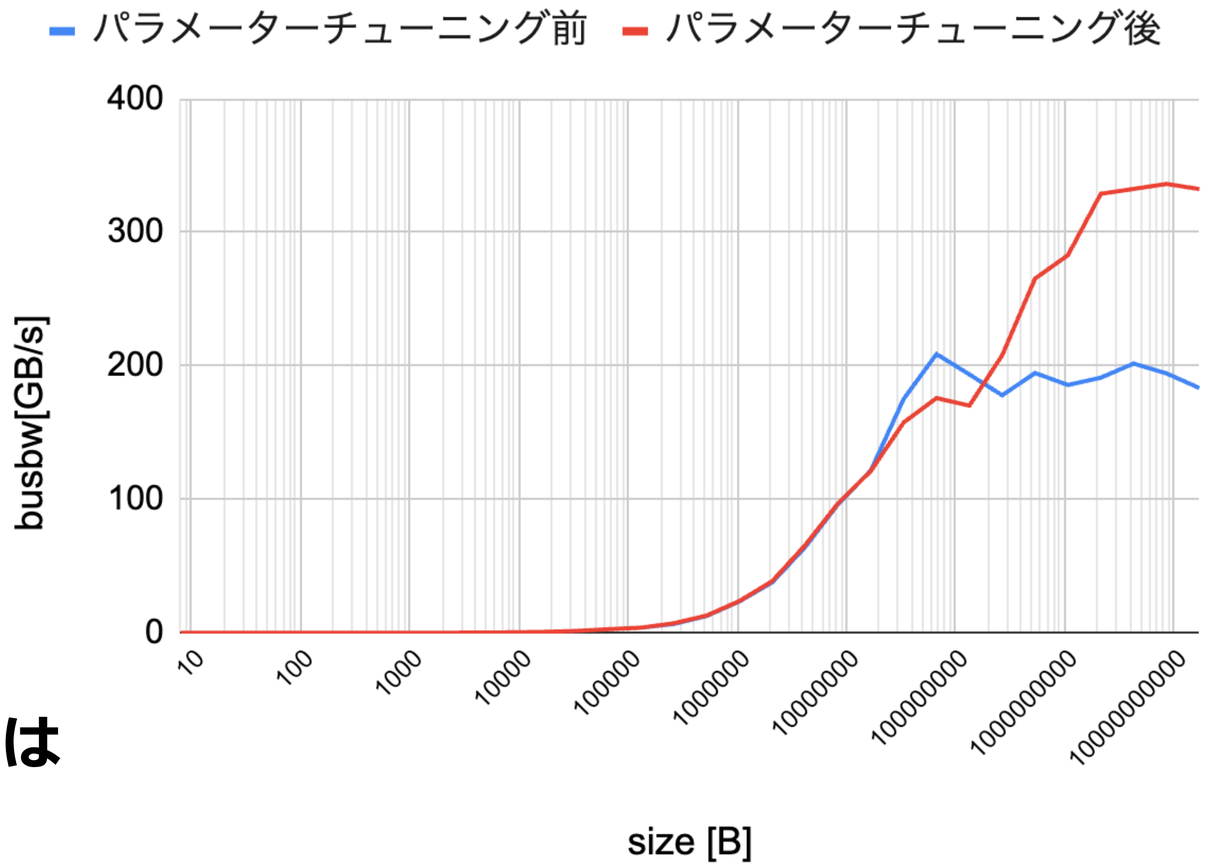
パフォーマンスチューニングの最終結果

デフォルトの設定と比較しチューニングによってパフォーマンスが倍近く改善

課題

- **最適化できていないパラメータもある**
  - DLBの他のパラメータ
  - Ingress interface hashingよりも最適な手法はないか？

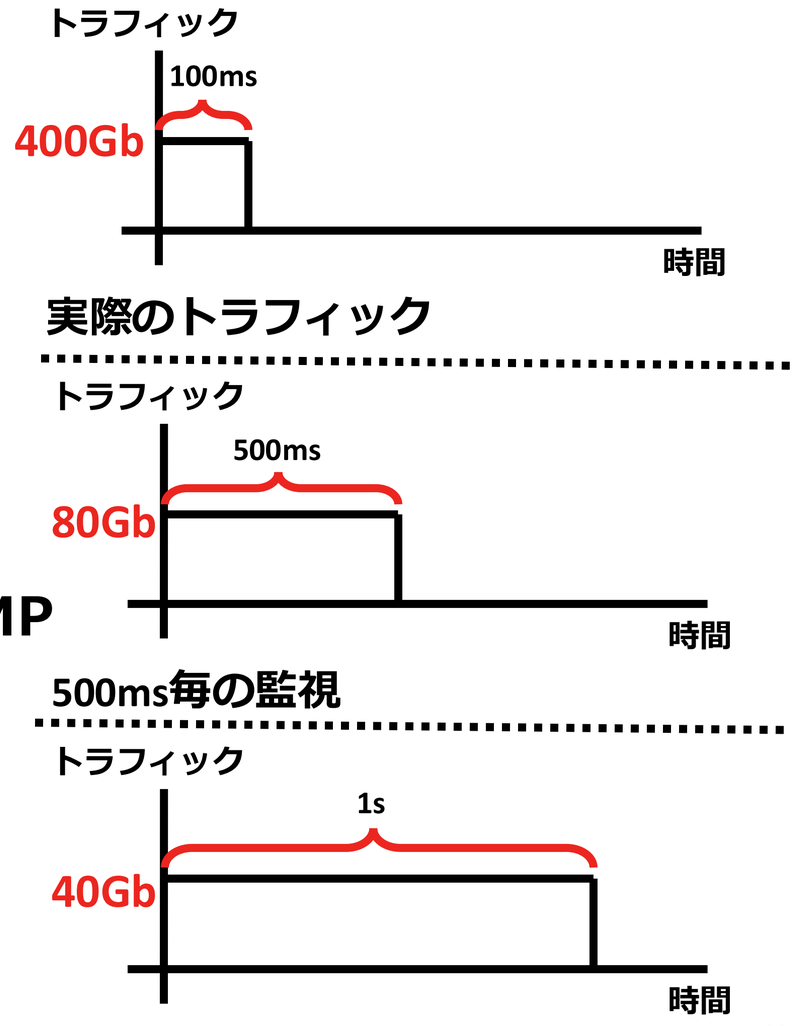
マルチベンダーのLosslessネットワークはパフォーマンスチューニングが難しい



インターコネクトのモニタリング

今後の投資判断のためインターコネクトがどの程度活用されているか確認したい

マイクロバーストを監視するためにはより高頻度にデータを取得する必要



スイッチ毎にNOSが異なるためデータの取得可能なの方法が異なる

- **Spine SN4700: Telemetry(gNMI) / OpenTelemetry / SNMP**
- **Leaf QFX5240: Telemetry(gNMI) / SNMP**

→ 各NOSに最適なモニタリング方法を検討

マイクロバーストモニタリング

Leaf（QFX5240 / Junos）

gNMIでテレメトリデータを収集

最短2秒間隔でデータ取得が可能

データレートが高いとデータの更新が停止する事象を確認

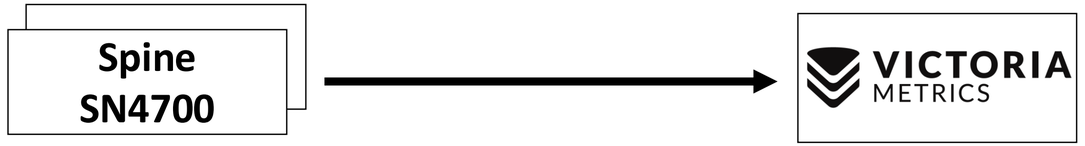
- gNMIcのプロセスを多重化し、データ種別ごとに分離収集することで対応

Spine（SN4700 / Cumulus Linux）

Cumulus Linux ではgNMIでのデータ収集は約15秒間隔が限界

Cumulus Linux 5.11以降でOpenTelemetryをサポート

- 約2秒間隔のデータ更新が可能に
- スイッチから直接TSDBへ書き込み可能



高頻度なデータ収集にも課題あり

・取得間隔の微妙なズレによってレート計算が不安定になるリスク
・監視基盤を見直す必要を感じており今後の課題。皆さんどうしてますか？

まとめ

- マルチベンダーの400G/800Gスイッチを用いたLossless ネットワークを構築
- SN-MTコネクタを用い、既存環境の4倍の密度でパッチパネルを構築
- パラメータチューニングを行い、マルチベンダーでも高性能なネットワークを提供
- インターコネクトのモニタリング環境を整備し、高細度なトラフィック収集を実現

自社に最適化した構成により内製のメリットである

コストの最適化を実現し、社内に低コストなGPUクラスタを提供

今後の取り組み

- 1.6Tスイッチの導入検討
- CPOスイッチの導入検討
- 水冷/液浸の導入検討
- LPOトランシーバーの導入検討
- 新しいGPU(RTX Pro 6000など)に対応したインターコネクト構成の検討
- モニタリング環境の再検討

議論のポイント

- **分散学習環境のNICの枚数はどのように決めましたか**
- **分散学習環境におけるGPUサーバー・スイッチの異ベンダー構成についてどう思いますか**
- **分散学習環境のチューニングはどのように行っていますか**
- **分散学習環境のモニタリングをどのように行っていますか**
- **ラック内の高密度な配線を実現するための工夫などありますか**
- **高電力・水冷時代のネットワーク設計・検証はどのように進めると良いですか**

APPENDIX

LPO(Linear Pluggable Optics)トランシーバー

今後の増設時にトランシーバーコストと消費電力削減を目指して検証実施

- 以下の利用中の機器との相互接続性をそれぞれ確認

  - ConnectX-7: 400G-DR4 純正/TeraHop FRO

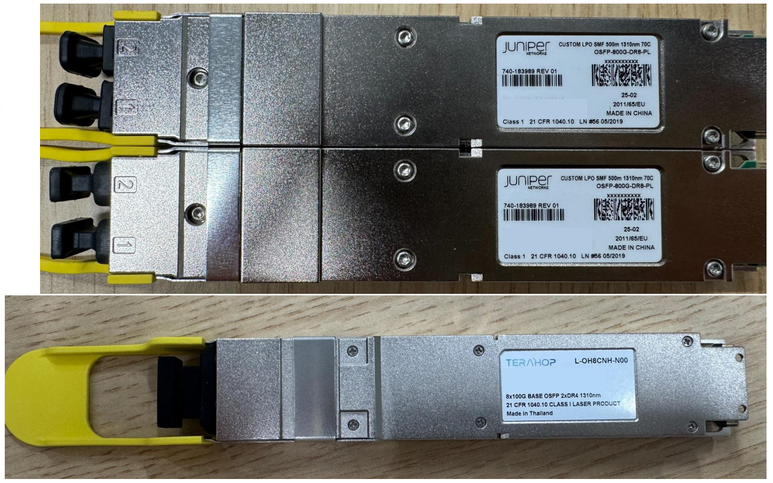
  - SN4700: 400G-DR4 純正/TeraHop FRO

  - QFX5240: 800G-DR8(2xMPO) 純正/TeraHop FRO、純正/TeraHop LPO

検証結果

- ほとんどの接続に問題なし
- Opticsの温度が10～20℃程度低くなる → 故障率低下に期待
- 消費電力は1～2本では測定が難しく効果未確認
- 3rd party LPOと一部の組み合わせでBad Signal integrityとなる組み合わせあり

→<u>LPOの特性上これまで以上に念入りな接続検証が必要</u>



検証したLPOトランシーバ

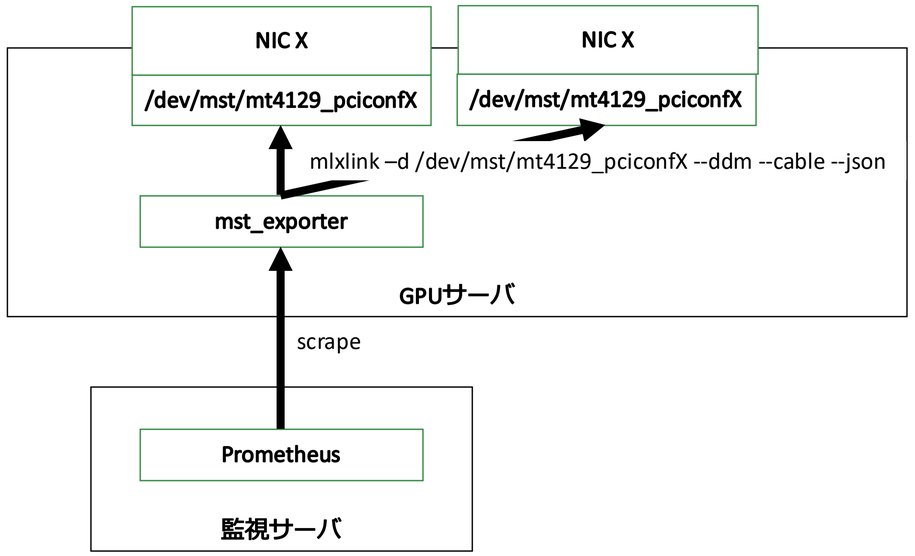
NICのモニタリング

1サーバーあたり10NIC搭載していることもあり、手動チェックは大変

スイッチも合わせると1台あたり20ポート以上確認する必要あり

NICのステータスをPrometheus形式で出力できる exporter を開発し確認作業を効率化

配線時・運用時・保守時に光レベルなどを簡単に確認・通知が可能




```
mst_device_state{device="mt4129_pciconf0"} 1
mst_rx_power_alarm_high{channel="1",device="mt4129_pciconf0"} 0
mst_rx_power_alarm_high{channel="2",device="mt4129_pciconf0"} 0
mst_rx_power_alarm_high{channel="3",device="mt4129_pciconf0"} 0
mst_rx_power_alarm_high{channel="4",device="mt4129_pciconf0"} 0
mst_rx_power_alarm_low{channel="1",device="mt4129_pciconf0"} 0
mst_rx_power_alarm_low{channel="2",device="mt4129_pciconf0"} 0
mst_rx_power_alarm_low{channel="3",device="mt4129_pciconf0"} 0
mst_rx_power_alarm_low{channel="4",device="mt4129_pciconf0"} 0
mst_rx_power_dbm{channel="1",device="mt4129_pciconf0"} 4
mst_rx_power_dbm{channel="2",device="mt4129_pciconf0"} 4
mst_rx_power_dbm{channel="3",device="mt4129_pciconf0"} 2
mst_rx_power_dbm{channel="4",device="mt4129_pciconf0"} 4
```

FIN