AI/MLデータセンターネットワークでの負荷分散手法 -NW運用者の視点から-

Staff Engineer

Masayuki Kobayashi

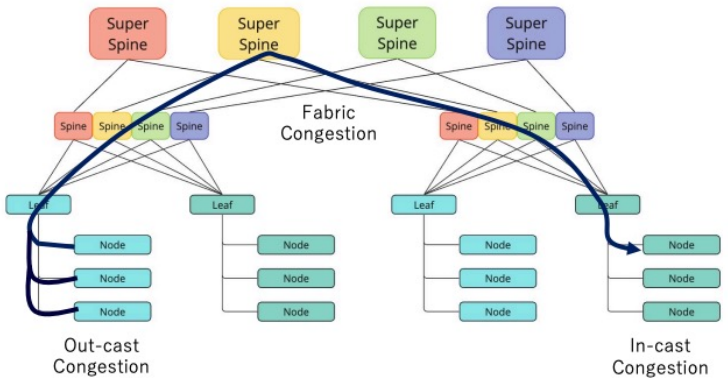
LINEヤフー

（前提）データセンターネットワークと輻輳

JANOG53のおさらい

データセンターネットワークと輻輳

どこで輻輳が発生するのか



- **In-cast congestion**
  - 複数の送信元が1つの宛先にデータを同時に送信する多対一の通信
  - 受信側バッファでtail dropが発生する
  - 現在のデータセンターネットワークで問題になりやすい
  - 発生したときの対応が難しい
- Fabric congestion
  - 不均衡なトラフィック分散やフローの偏りによってファブリック内のバッファでパケットロスが発生する
  - DLBやGLB※での対策が推奨される
- Out-cast congestion
  - 特定の出力キューが総リンク帯域幅を超える速度でデータを送信するときに発生する
  - 送信元が特定されるため対応は比較的容易

※ Dynamic Load Balancing / Global Load Balancing

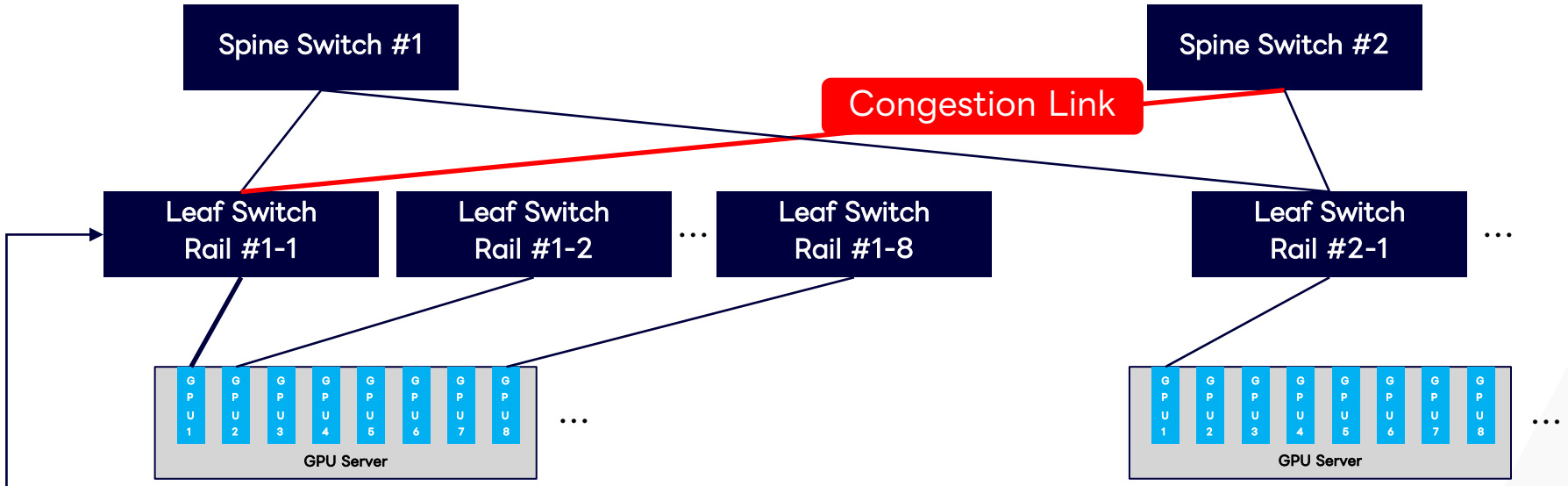
JANOG53ではTCPベースのネットワークでの輻輳制御について
主にIncastの問題に焦点を当てました。
本発表ではLossless Ethernetでの負荷分散手法について話します。

- 本発表で扱う対象領域

“データセンターネットワークでの輻輳対策どうしてる？” (JANOG53)
https://www.janog.gr.jp/meeting/janog53/dcqos/

分散機械学習とElephant Flow

RoCEv2のフローには5-tuple ECMPでリンク分散できるだけのエントロピーが足らない場合が多い



/	iface	Rx	Tx	Total
400G GPU	swp1:	309.90 Gb/s	317.16 Mb/s	310.22 Gb/s
Spine #1	swp31:	158.52 Mb/s	2.84 kb/s	158.53 Mb/s
Spine #2	swp32:	158.64 Mb/s	309.90 Gb/s	310.06 Gb/s
	total:	310.22 Gb/s	310.22 Gb/s	620.43 Gb/s

分散機械学習のRDMA通信が単一のフローにマッピングされ特定リンクに偏っている

主なパケット分散技術

公開情報ベースでのまとめ

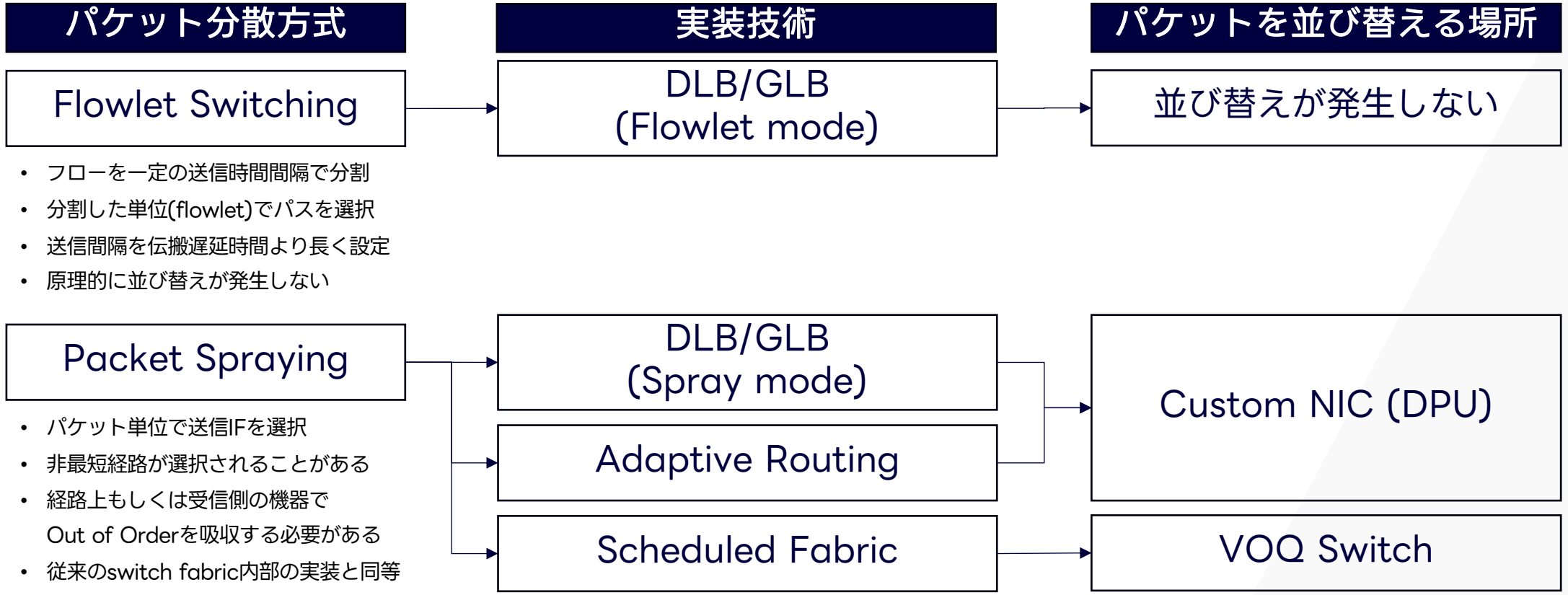
負荷分散手法	品質測定場所	分散要素	備考
Traditional ECMP	Local	5-tuple	RTAG7のハッシュ生成など Static mode と呼ばれる
Traditional ECMP Random Packet Spraying (RPS)	Local	5-tuple (UDP Source Port) (IPv6 Flow Label)	送信元情報のランダム化に よるパケットスプレー実現
★ Adaptive Routing (AR)	Local	Link/Queue 使用率 IB BTH flag	Infinibandの延長 Nvidia Spectrumの機能
★ Dynamic Load Balancing (DLB)	Local	Link/Queue 使用率	flowlet/sprayのmode選択 が可能な実装あり
Global Load Balancing (GLB)	Local + Remote	Link/Queue 使用率	パス品質を隣接機器に伝達 対応製品はまだ入手できず
Fully-Scheduled Fabric (FSF)	Local	パケット単位のスプレー	VOQのスイッチで実装 対応製品はまだ入手できず

Reactive Path Rebalancing, Source IF-based LBなどのDLBの機能拡張技術は記載していません。

★ の技術で動作を検証（どちらも導入予定のため）

Flowlet Switching vs Packet Spraying

負荷分散方式の違いもネットワーク機器と構成選択の判断要素の一つとなる



基本的にスプレー方式はスイッチもしくはNIC側でのOut of Order吸収のためのReorder実装が必要となる

→ 異なる方式を混在させたネットワークを運用することは現実的に厳しい

Adaptive Routing

NICとAR対応スイッチの連携メカニズム

```
% sudo mlxreg -d 0e:00.0 --reg_name ROCE_ACCL --get
Sending access register...

Field Name                                                | Data
============================================================
roce_adp_retrans_field_select                             | 0x00000001
roce_tx_window_field_select                               | 0x00000001
roce_slow_restart_field_select                            | 0x00000001
roce_slow_restart_idle_field_select                       | 0x00000001
min_ack_timeout_limit_disabled_field_select               | 0x00000001
adaptive_routing_forced_en_field_select                   | 0x00000001
selective_repeat_forced_en_field_select                   | 0x00000001
dc_half_handshake_en_field_select                         | 0x00000000
ack_dscp_force_field_select                               | 0x00000001
roce_adp_retrans_en                                       | 0x00000001
roce_tx_window_en                                         | 0x00000000
roce_slow_restart_en                                      | 0x00000001
roce_slow_restart_idle_en                                 | 0x00000000
min_ack_timeout_limit_disabled                            | 0x00000000
adaptive_routing_forced_en                                | 0x00000001
selective_repeat_forced_en                                | 0x00000000
dc_half_handshake_en                                      | 0x00000000
ack_dscp_force                                            | 0x00000000
ack_dscp                                                  | 0x00000000
============================================================
```

動作環境
NIC: NVIDIA ConnectX-7 MCX75310AAS-NEAT
Firmware: 28.39.2048-LTS

```
[+] Internet Protocol Version 4, Src: 192.168.0.10, Dst: 192.168.0.20
[+] User Datagram Protocol, Src Port: 60363, Dst Port: 4791
[-] InfiniBand
  [-] Base Transport Header
        Opcode: Reliable Connection (RC) - RDMA WRITE Only (10)
        0... .... = Solicited Event: False
        .1.. .... = MigReq: True
        ..00 .... = Pad Count: 0
        .... 0000 = Header Version: 0
        Partition Key: 65535
        Reserved: 00
        Destination Queue Pair: 0x0063cb
        0... .... = Acknowledge Request: False
        .100 0000 = Reserved (7 bits): 64 [=]
        Packet Sequence Number: 848230
  [+] RETH - RDMA Extended Transport Header
      Invariant CRC: 0x6e04aa4e
```

NICのPF側でARのflagを設定すると、
IB BTHのReserved Field (7bit)の中の1bitが設定される。
スイッチはこのbitの有無でARの対象パケットとする。

Adaptive Routing

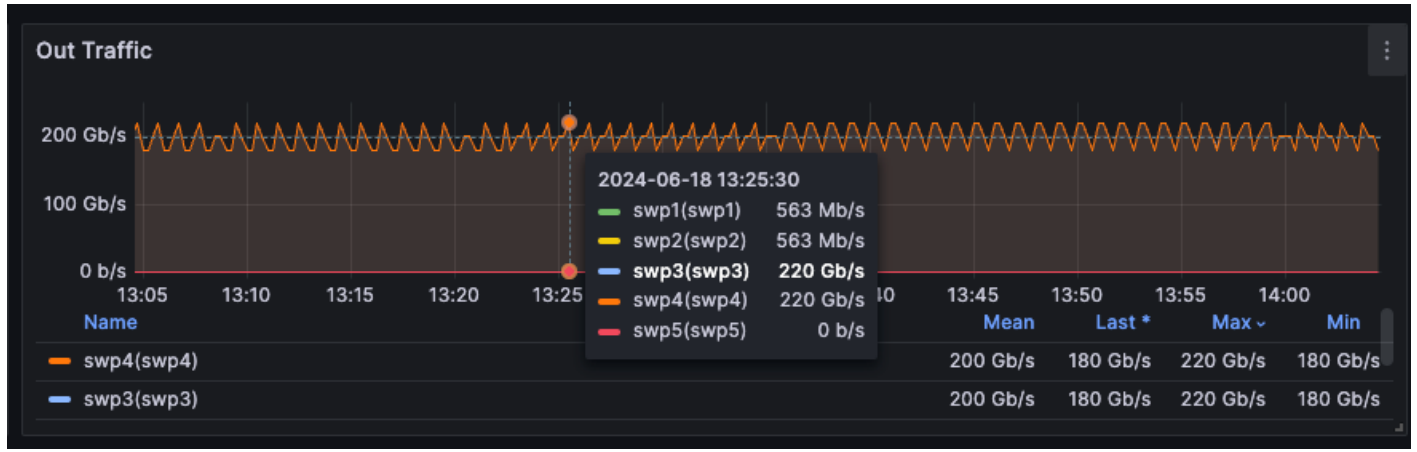
RoCEv2フローの分散

/	iface	Rx	Tx	Total
400G GPU	swp1:	241.95 Gb/s	394.86 Mb/s	242.35 Gb/s
Spine #1	swp31:	197.16 Mb/s	121.01 Gb/s	121.21 Gb/s
Spine #2	swp32:	197.70 Mb/s	120.94 Gb/s	121.14 Gb/s
	total:	242.23 Gb/s	242.23 Gb/s	484.45 Gb/s

AR有効時のRDMA Write

フローが均等に分散されている

(Leafスイッチで分散を確認)



ECMPメンバーでのばらつきが極めて少ない

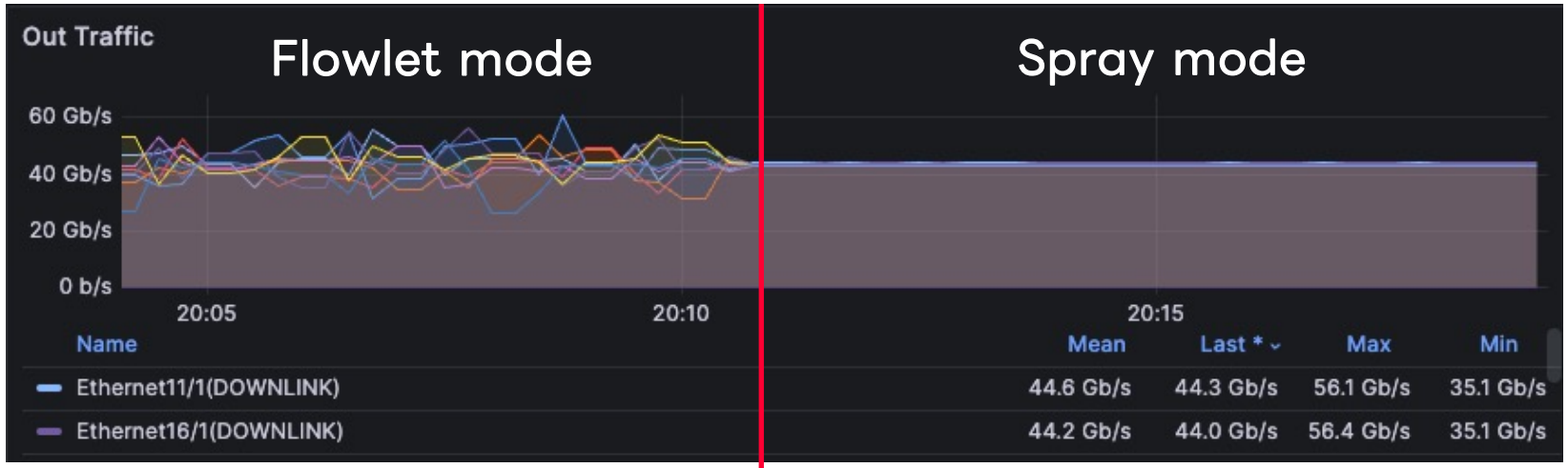
(Spineスイッチで分散を確認)

理想的な動きはするが、IB BTHのflagをNIC側で設定する必要がある

→ スイッチ側でもこのflagをmatchできる必要がある (merchant siliconのスイッチでは実装が無い)

Dynamic Load Balancing (DLB)

ローカルでの輻輳状況に応じたトラフィック割り当て



- 非アクティブ時間(アイドル間隔)による分散
- ECMPメンバーでのばらつきが大きい
  帯域の上ブレはキャパプラで吸収する
- Reorderが発生しない

- パケット単位による分散
- ECMPメンバーでのばらつきが小さい
  リンクの利用効率向上が期待できる
- Reorderが発生する可能性がある

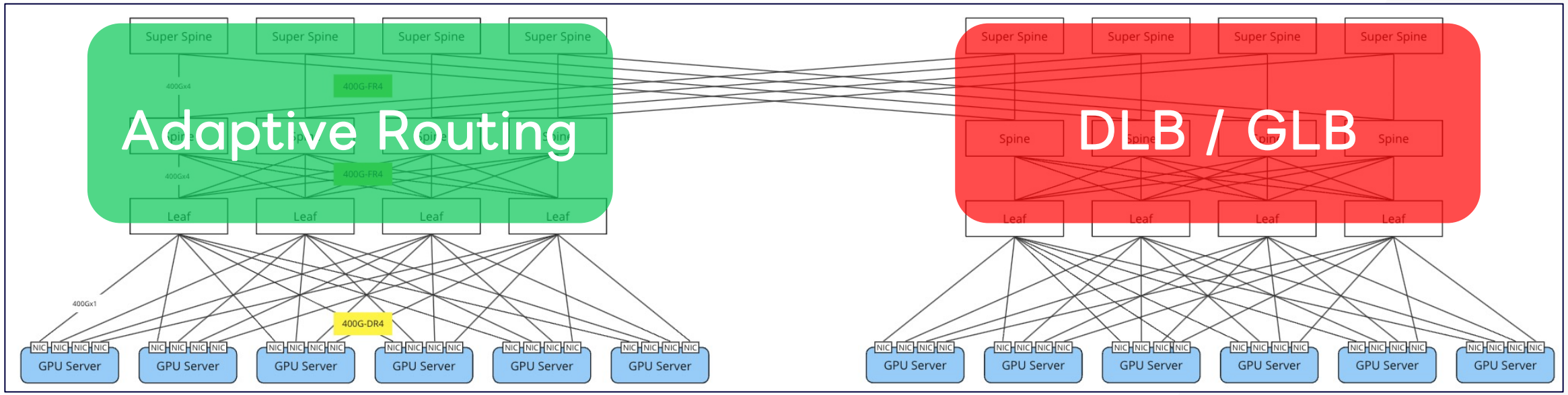
分散モード変更点

⚠ 運用中の分散モード変更は通信影響があるため非推奨です。

なぜ負荷分散技術に着目しているのか？

GPUクラスタを拡張していくと、異なる実装のスイッチが混在することが考えられる

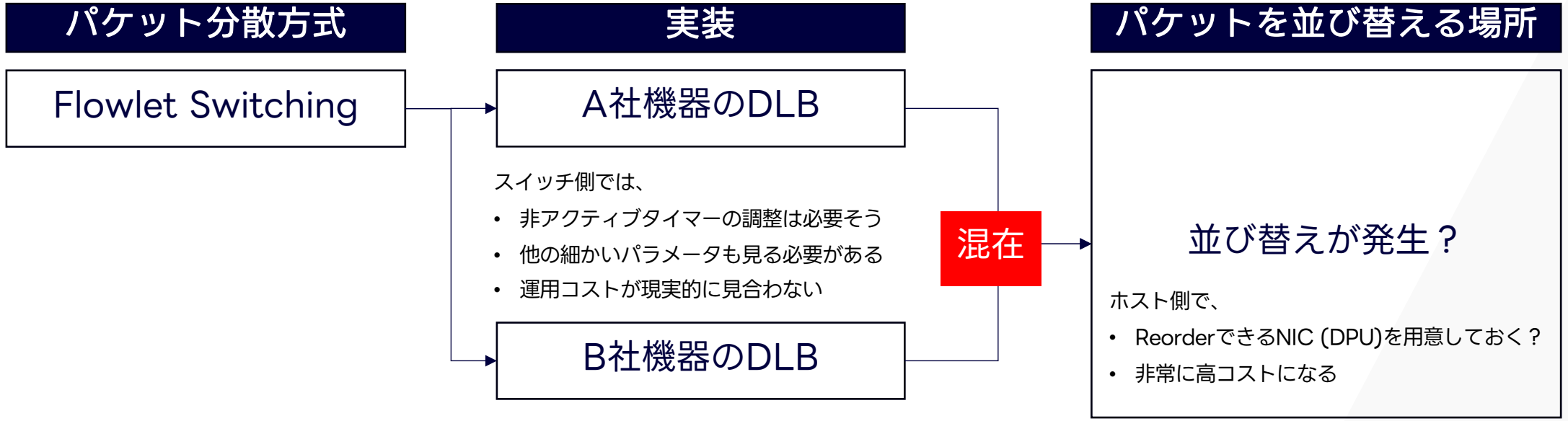
- 負荷分散方式によって内部のアルゴリズムで使用するインテリジェンスが異なる
  - Sampling Rate, Inactivity Timer, Interframe Gap, Packet Flag など
- クラスタ拡張などでマルチベンダ化した際に、これらの違いでパフォーマンスに影響する可能性がある
  - 具体的には受信側でパケットのOut of Orderと並び替え(Reorder)が必要になる可能性



LINEヤフーアメリカDCのGPUクラスタ構成(Rail-optimized Backend Clos): 今後クラスタ拡張で異なる負荷分散アルゴリズムが混在予定

混在構成の場合の留意点

同じ方式であればマルチベンダのスイッチが混在可能か？



悩みポイント

- GPUの世代交代が早く、既存のNWに追加していくか、世代ごとに閉じたクラスタのネットワークにするか
- 既存のNWに追加していく場合、その時々の事情によって異なるベンダのスイッチが入る場合がある
- マルチベンダでLossless Ethernetを構築する場合に、負荷分散方式が技術的な考慮事項になるのではと考えている

→ 次回以降のJANOGで答えをお話できるかもしれません（検証・構築中）

議論したいポイント

会場やSlackでご意見お聞かせいただけると幸いです！

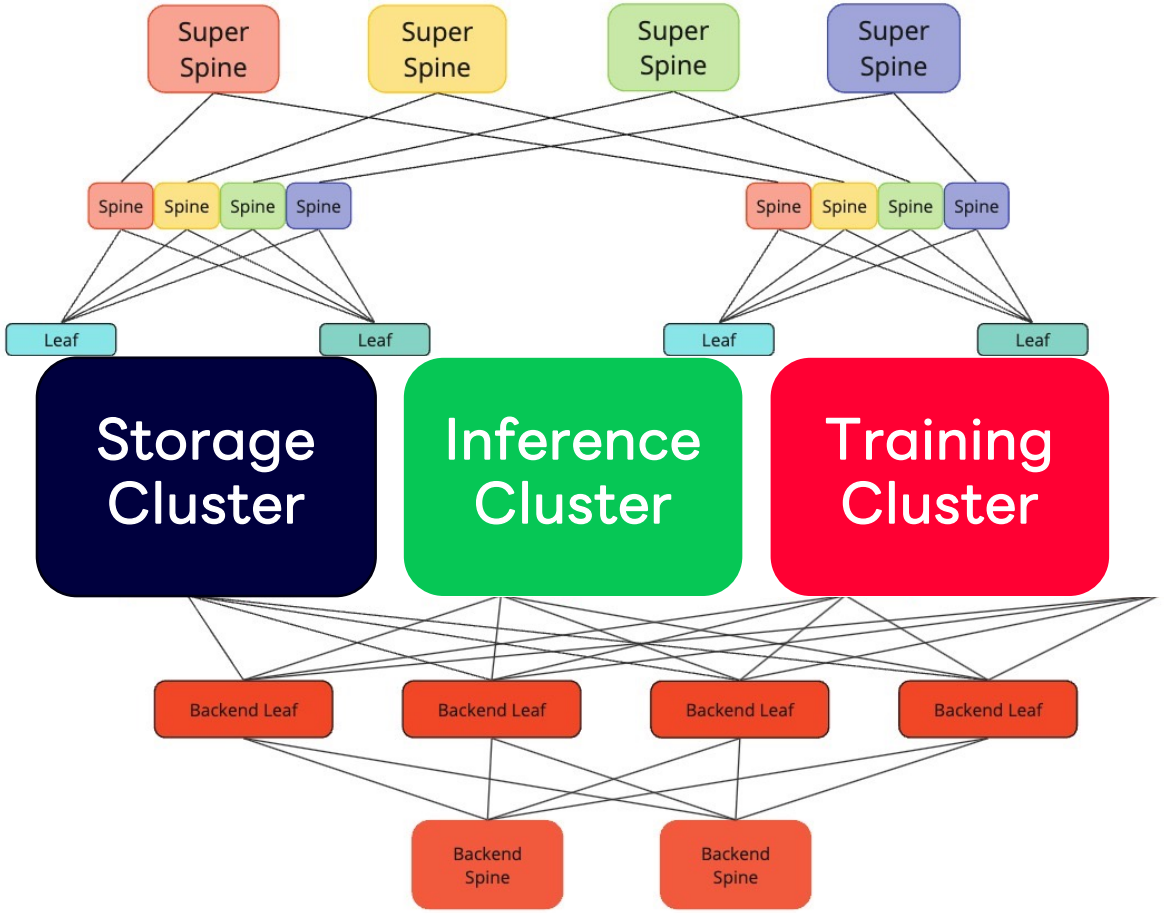
- GPUクラスタのネットワークをどのように作っているか
  - RoCEv2のフローの分散やReorderの影響をどの程度気にしているか
  - 負荷分散技術の選択基準
  - 環境やワークロードに合わせてパラメータを調整しているか
- マルチベンダでRoCEv2ネットワークを組むことを考えるか
  - どのような点を考慮するべきか
  - GPUの世代や用途ごとにネットワークを分離するか
- ネットワーク運用者の立場で、今のAI/MLネットワーク(Lossless)に足りないもの
  - こんな機能があったら嬉しいなど

Appendix

資料をご覧の皆様へ、LTでは話しきれない前提知識や関連情報を記載します。

AI/MLのためのデータセンターネットワーク

CPU-Centric から GPU-Centric へ



Lossy Frontend Fabric

- インターネットサービスに最適化
- Clos Topology
- ベストエフォート
- Lossy (TCP)
- CPU-Centric

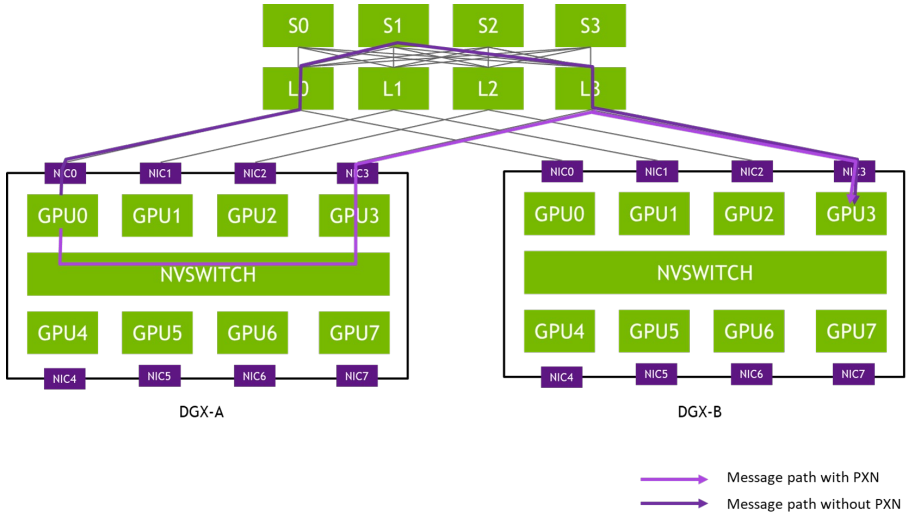
Lossless Backend Fabric

- AI Chip間の通信に最適化
- Rail-optimized Topology
- 超低遅延・超広帯域・非競合
- Lossless (RDMA)
- GPU-Centric

本発表の対象

Rail-optimized Topology

ノード間のGPUが最短経路で通信するために最適化されたトポロジ



- GPUサーバの**NICと同じ数だけのLeafスイッチを用意**し
  各GPU/NICは同じLeafスイッチに接続する構成

- ノード間通信では、**同じ番号のGPU同士で通信する**仕様
  NCCLがAllReduceなどを実行するときにトポロジを識別する
  → この内部バスを含むGPU間の通信経路を“Rail”と呼ぶ

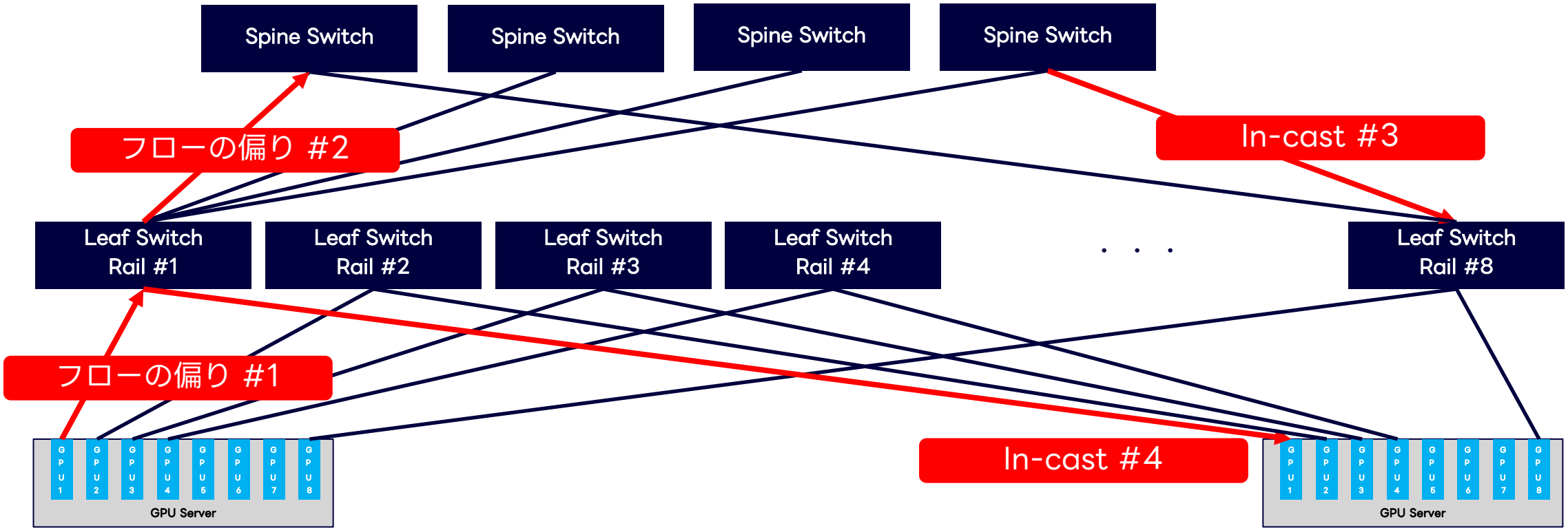
- NCCLの機能(PXN)がパフォーマンスを最大化するため
  トポロジがRail-optimizedであることを前提としている

- 同じ番号のGPU同士がLeaf折り返しで最短距離になるような
  ネットワークを作ることで、Ringアルゴリズムを効率化し、
  Spine経由の競合(輻輳)を回避する

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

Rail-optimized Topologyと輻輳

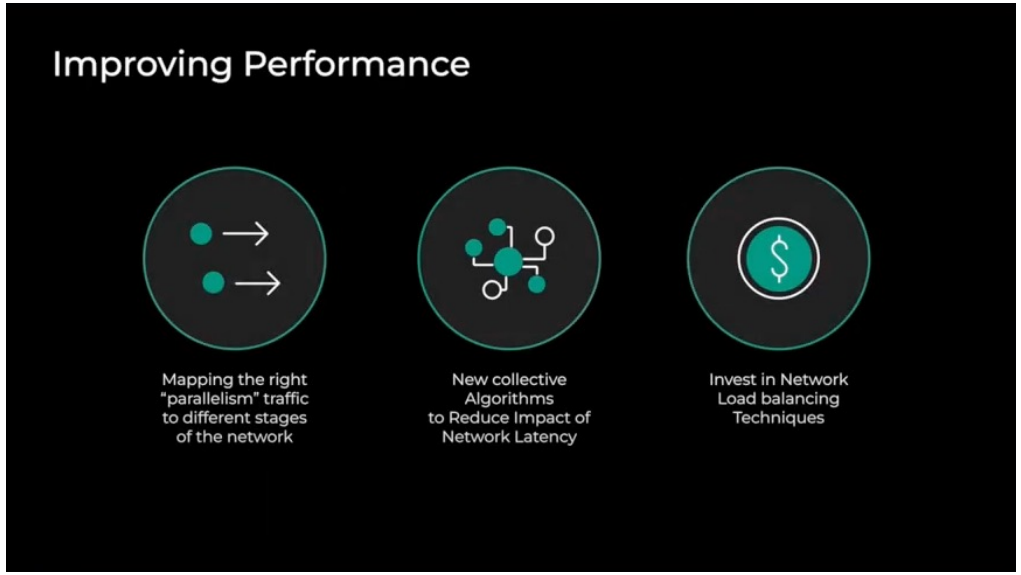
どこでどのような輻輳が発生するのか



- #1 の偏りはRail-optimized Topologyである以上避けることはできない
- #2 の偏りはARやDLBなどのスイッチ側の負荷分散技術によって一定の対処が可能
- #3 のIncastはスイッチ側の負荷分散技術とホスト側の輻輳制御(スイッチでのECN)で一定の対処が可能
- #4 のIncastはホスト側のPFCと輻輳制御(送信元へのCNP)で一定の対処が可能

RDMAネットワークでの負荷分散技術の重要性

ネットワークの性能がトレーニングの性能に直結する

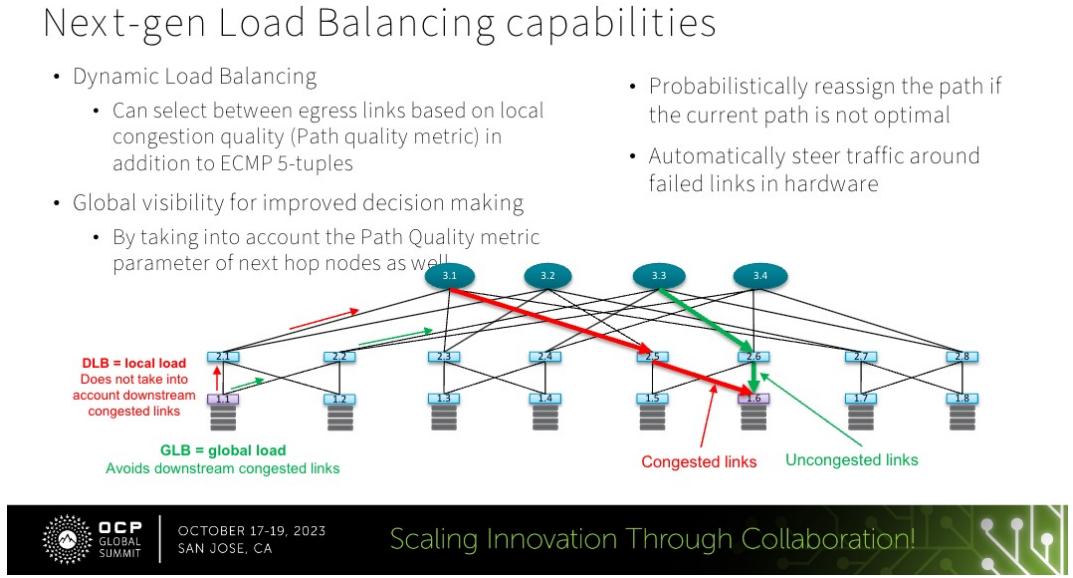


Metaではパフォーマンス改善のために負荷分散技術へ積極的な投資を実施

SYSTEMS @SCALE 2024

“GenAI Training In Production: Software, Hardware & Network Considerations” より引用

https://www.youtube.com/watch?v=1lhrGRqqPWU



AlibabaとBroadcomによる次世代の負荷分散技術への期待

OCP GLOBAL SUMMIT 2023

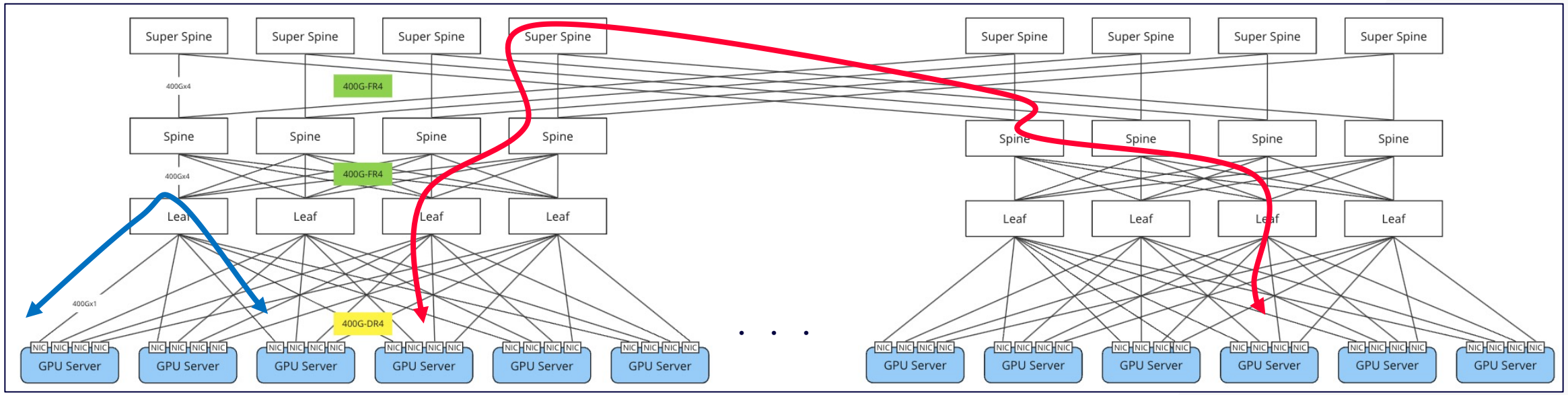
“Alibaba‘s Ethernet based DC deployment for AI/ML workloads using Merchant Silicon” より引用

https://www.youtube.com/watch?v=T5fbbA27rUw

（そもそも）インテリジェントな負荷分散の必要性

中〜大規模クラスタにしか必要ないのでは？

- Rail-Optimized構成の場合、GPU間通信は最短経路のLeafで折り返す
  Spineを経由しない規模の場合、必ずしもこれらの負荷分散技術が必要とは言えないのではないか？
- 現状GPU 1台に400G 1ポートが紐づくため、GPU RailからLeafへの物理パスは一つしかない
- 近い距離のGPUを払い出すスケジューリングコスト vs 負荷分散技術でカバーするコスト?



LINEヤフーアメリカDCのGPUクラスタ構成(Rail-optimized Backend Clos): ARやDLBは赤のフローに対して有効

新しい負荷分散技術の留意点

これまでの運用に適用できないケースがある

- Resilient Hashing との共存ができない
  - 5-tuple に依存しないため、同じフローを一貫性のあるネクストホップに割り当てられない
  - 通常の環境(TCPなどのLossy Network)ではDLBなどを適用すると問題になるケースがある

参考情報

データセンターネットワークでの輻輳制御について (JANOG53)

- https://www.janog.gr.jp/meeting/janog53/dcqos/ (資料)
- https://www.youtube.com/watch?v=hzRyLaE1wCs (動画)

AI(人工知能)の為のネットワーク (JANOG53)

- https://www.janog.gr.jp/meeting/janog53/ainw/

AI/ML基盤の400G DCネットワークを構築した話 (JANOG52)

- https://www.janog.gr.jp/meeting/janog52/aiml400/
- Adaptive Routingの導入事例

LLMとGPUとネットワーク (MPLS Japan 2023)

- https://mpls.jp/2023/presentations/mpls2023-yuyarin.pdf

参考情報

LINEヤフー米国データセンタ技術の最前線：LLMと水冷技術の戦略 (JANOG54)

- https://www.janog.gr.jp/meeting/janog54/dlc/
- 本発表内のGPUクラスタネットワークはこのアメリカデータセンターで運用するものです

参考情報

Adaptive Routing (AR)

- https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/a17f55305bcebb331b95ad79c1e9d3fdc8e185c6.pdf

Dynamic Load Balancing (DLB)

- https://www.broadcom.com/video/6328983519112

Global Load Balancing (GLB)

- https://jp.broadcom.com/blog/cognitive-routing-in-the-tomahawk-5-data-center-switch
- https://www.youtube.com/watch?v=A30nk8_e1WA

On the impact of packet spraying in data center networks

- https://ieeexplore.ieee.org/document/6567015
- Random Packet Spraying (RPS)

参考情報

Hashing Design in Modern Networks: Challenges and Mitigation Techniques

- https://www.usenix.org/conference/atc22/presentation/xu

English Slide.

Load balancing methods in AI/ML data center networks

– From the perspective of network operators –

Staff Engineer

Masayuki Kobayashi

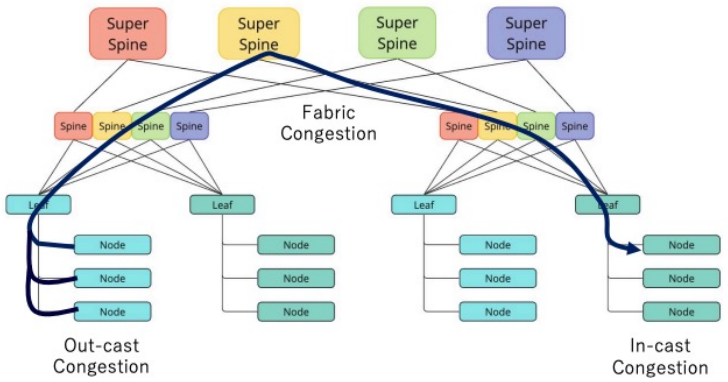
LINEヤフー

Data Center Networks and Congestion

Recap from JANOG53

データセンターネットワークと輻輳

どこで輻輳が発生するのか



- **In-cast congestion**
  - 複数の送信元が1つの宛先にデータを同時に送信する多対一の通信
  - 受信側バッファでtail dropが発生する
  - 現在のデータセンターネットワークで問題になりやすい
  - 発生したときの対応が難しい
- Fabric congestion
  - 不均衡なトラフィック分散やフローの偏りによってファブリック内のバッファでパケットロスが発生する
  - DLBやGLB※での対策が推奨される
- Out-cast congestion
  - 特定の出力キューが総リンク帯域幅を超える速度でデータを送信するときに発生する
  - 送信元が特定されるため対応は比較的容易

※ Dynamic Load Balancing / Global Load Balancing

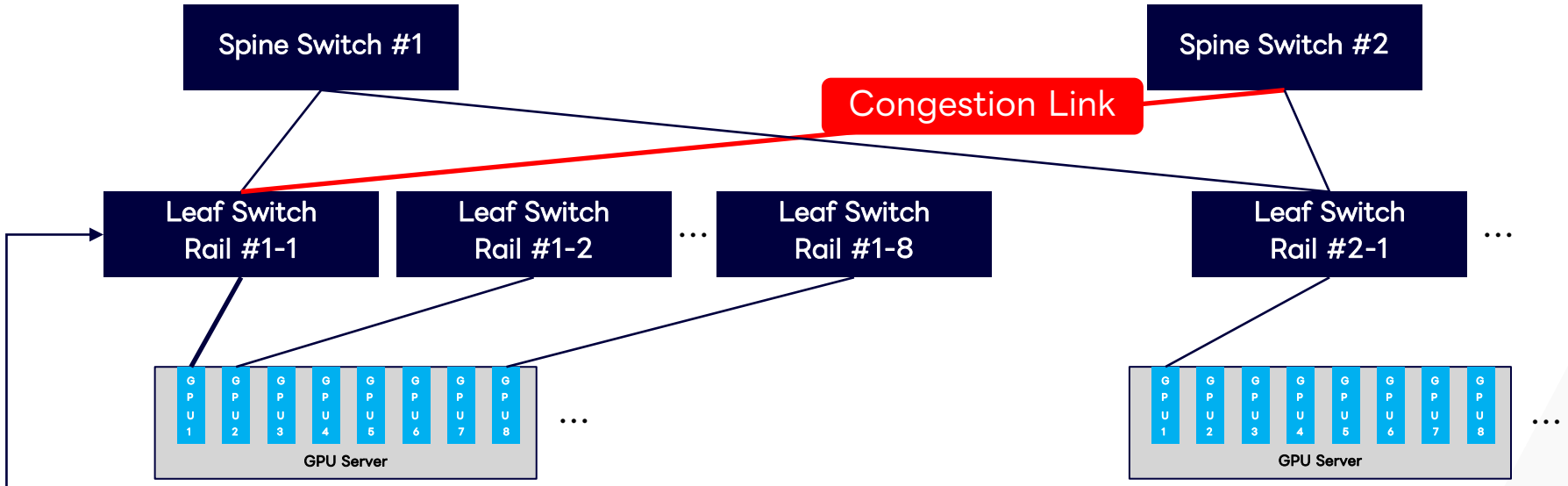
In JANOG53, we focused on congestion control in TCP-based networks, particularly the Incast problem.
In this presentation, we will discuss load balancing techniques in Lossless Ethernet.

- Scope of this Presentation

"How do you deal with congestion in data center networks?" (JANOG53)
https://www.janog.gr.jp/meeting/janog53/dcqos/

Distributed ML and Elephant Flow

RoCEv2 flows often lack enough entropy to distribute links using 5-tuple ECMP



/	iface	Rx	Tx	Total
400G GPU	swp1:	309.90 Gb/s	317.16 Mb/s	310.22 Gb/s
Spine #1	swp31:	158.52 Mb/s	2.84 kb/s	158.53 Mb/s
Spine #2	swp32:	158.64 Mb/s	309.90 Gb/s	310.06 Gb/s
	total:	310.22 Gb/s	310.22 Gb/s	620.43 Gb/s

Distributed machine learning RDMA communication is mapped to a single flow, leading to imbalanced specific links.

Packet Distribution Techniques

Summary based on public information

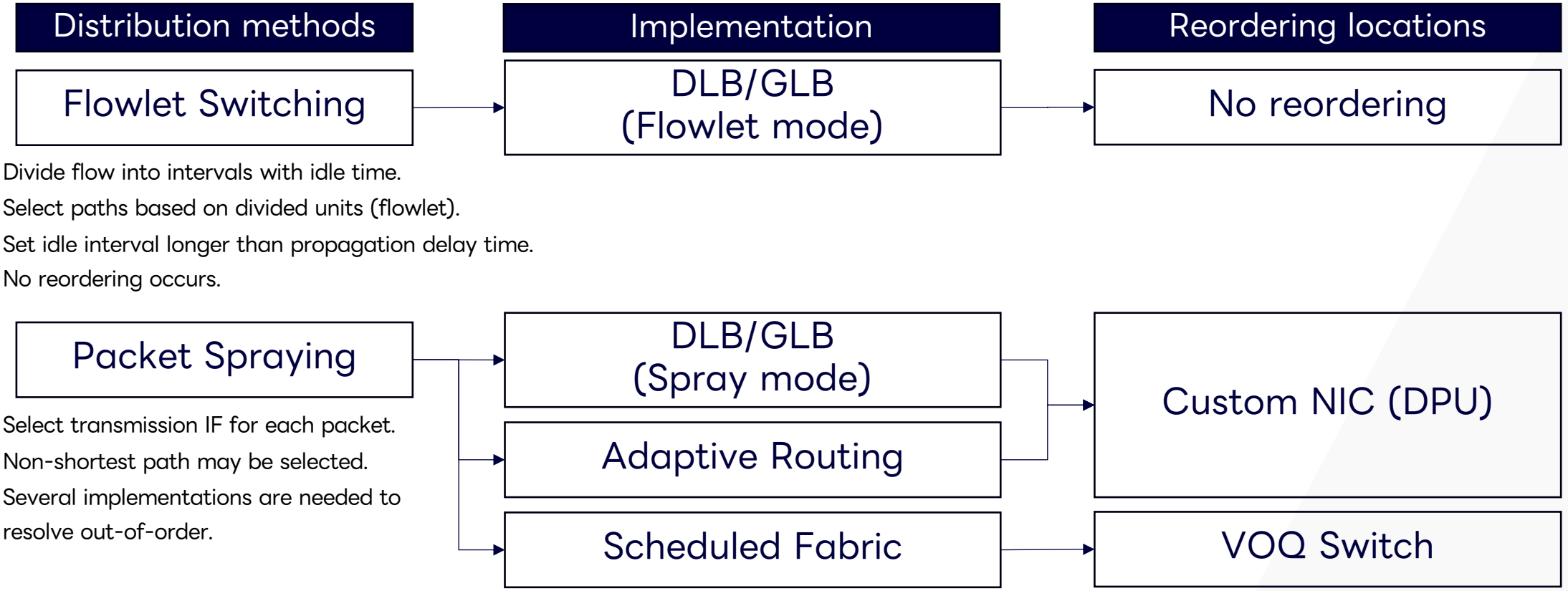
Load Balancing Method	Measurement Location	Distribution Element	Remarks
Traditional ECMP (Static mode)	Local	5-tuple	RTAG7 hash, etc.
Traditional ECMP Random Packet Spraying (RPS)	Local	5-tuple (UDP Source Port) (IPv6 Flow Label)	Packet spray by randomizing source information.
★ Adaptive Routing (AR)	Local	Link/Queue usage IB BTH flag	Extension of Infiniband. Feature of Nvidia Spectrum chip.
★ Dynamic Load Balancing (DLB)	Local	Link/Queue usage	flowlet/spray mode is selectable.
Global Load Balancing (GLB)	Local + Remote	Link/Queue usage	Notify neighboring devices of the quality of the path.
Fully-Scheduled Fabric (FSF)	Local	Packet-level spray	Implemented with VOQ devices.

★ We tested AR and DLB.

Reactive Path Rebalancing, Source IF-based LB, and other DLB feature extensions are not listed.

Flowlet Switching vs Packet Spraying

Load balancing methods as a deciding factor in selecting network equipment and configuration



- Divide flow into intervals with idle time.
- Select paths based on divided units (flowlet).
- Set idle interval longer than propagation delay time.
- No reordering occurs.

- Select transmission IF for each packet.
- Non-shortest path may be selected.
- Several implementations are needed to resolve out-of-order.

Spray methods generally require reordering implementation on the switch or NIC side to absorb out-of-order packets.

→ Operating a network with mixed methods is practically challenging.

Adaptive Routing

Mechanism for collaboration between NIC and AR-compatible switches

```
% sudo mlxreg -d 0e:00.0 --reg_name ROCE_ACCL --get
Sending access register...

Field Name                                                | Data
============================================================
roce_adp_retrans_field_select                             | 0x00000001
roce_tx_window_field_select                               | 0x00000001
roce_slow_restart_field_select                            | 0x00000001
roce_slow_restart_idle_field_select                       | 0x00000001
min_ack_timeout_limit_disabled_field_select               | 0x00000001
adaptive_routing_forced_en_field_select                   | 0x00000001
selective_repeat_forced_en_field_select                   | 0x00000001
dc_half_handshake_en_field_select                         | 0x00000000
ack_dscp_force_field_select                               | 0x00000001
roce_adp_retrans_en                                       | 0x00000001
roce_tx_window_en                                         | 0x00000000
roce_slow_restart_en                                      | 0x00000001
roce_slow_restart_idle_en                                 | 0x00000000
min_ack_timeout_limit_disabled                            | 0x00000000
adaptive_routing_forced_en                                | 0x00000001
selective_repeat_forced_en                                | 0x00000000
dc_half_handshake_en                                      | 0x00000000
ack_dscp_force                                            | 0x00000000
ack_dscp                                                  | 0x00000000
============================================================
```

Operating Environment
NIC: NVIDIA ConnectX-7 MCX75310AAS-NEAT
Firmware: 28.39.2048-LTS

```
[+] Internet Protocol Version 4, Src: 192.168.0.10, Dst: 192.168.0.20
[+] User Datagram Protocol, Src Port: 60363, Dst Port: 4791
[-] InfiniBand
  [-] Base Transport Header
        Opcode: Reliable Connection (RC) - RDMA WRITE Only (10)
        0... .... = Solicited Event: False
        .1.. .... = MigReq: True
        ..00 .... = Pad Count: 0
        .... 0000 = Header Version: 0
        Partition Key: 65535
        Reserved: 00
        Destination Queue Pair: 0x0063cb
        0... .... = Acknowledge Request: False
        .100 0000 = Reserved (7 bits): 64 [=]
        Packet Sequence Number: 848230
  [+] RETH - RDMA Extended Transport Header
      Invariant CRC: 0x6e04aa4e
```

Setting the AR flag on the PF side of the NIC sets one bit in the 7-bit Reserved Field of the IB BTH.

Switches use the presence of this bit to identify AR target packets.

Adaptive Routing

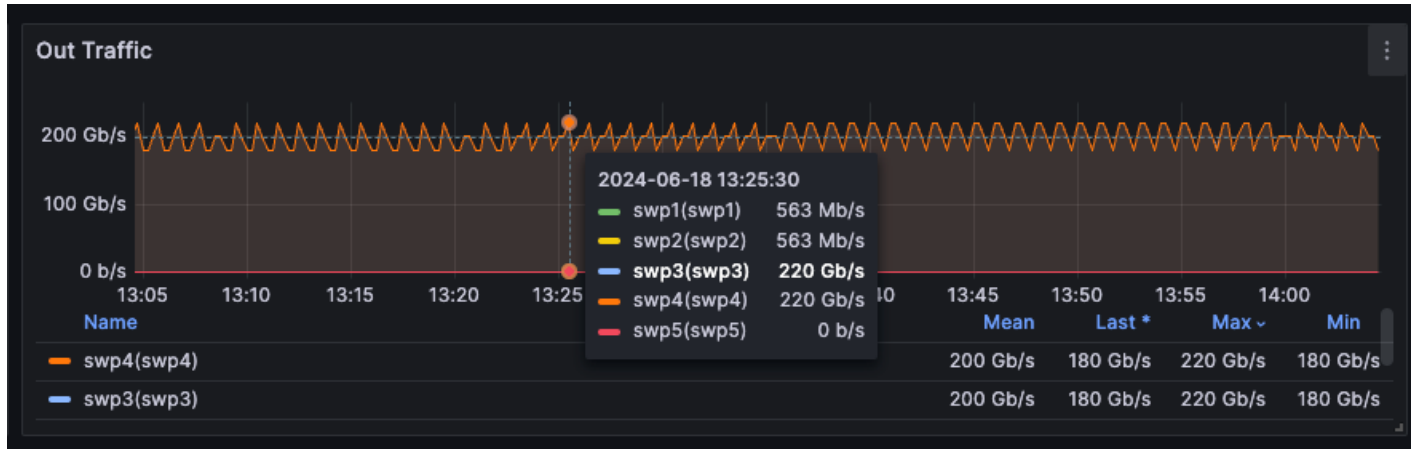
Distribution of RoCEv2 flows

/	iface	Rx	Tx	Total
400G GPU	swp1:	241.95 Gb/s	394.86 Mb/s	242.35 Gb/s
Spine #1	swp31:	197.16 Mb/s	121.01 Gb/s	121.21 Gb/s
Spine #2	swp32:	197.70 Mb/s	120.94 Gb/s	121.14 Gb/s
	total:	242.23 Gb/s	242.23 Gb/s	484.45 Gb/s

RDMA Write with AR enabled

Flow is evenly distributed.

(Confirmed by Leaf switch)



Very little variation among ECMP members.

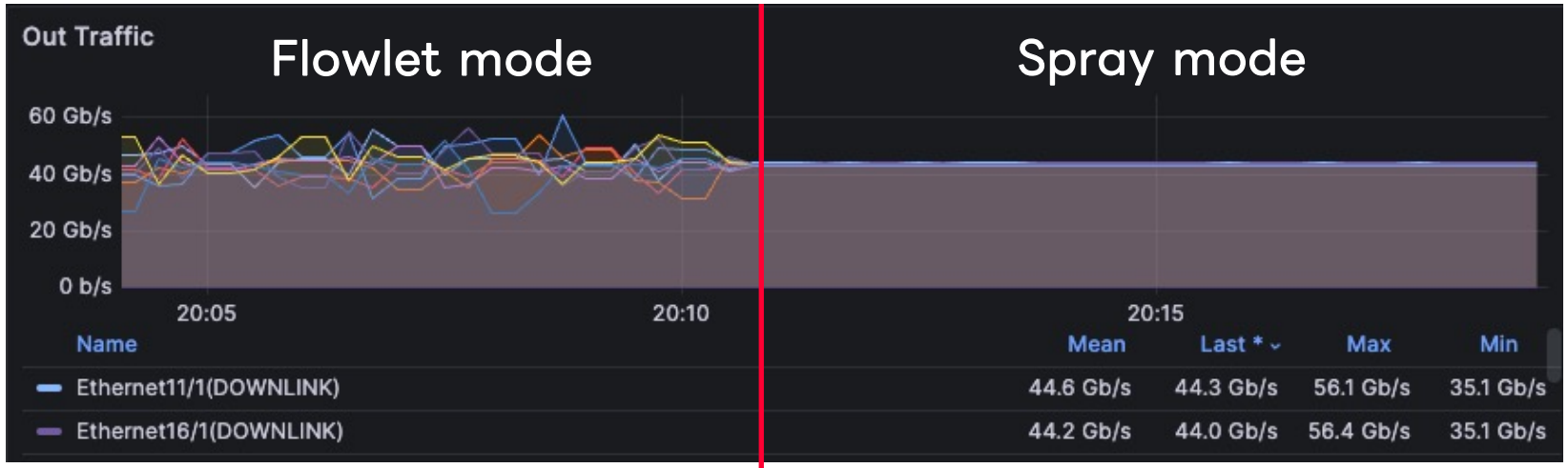
(Confirmed by Spine switch)

It works ideally, but the IB BTH flag needs to be set on the NIC side.

→ Switches also need to be matched to this flag (not implemented in merchant silicon switches).

Dynamic Load Balancing (DLB)

Traffic allocation according to local congestion conditions



- Distribution by inactive time (idle interval).
- Large variation among ECMP members.
- Band width bursts are absorbed by NW design.
- No reordering occurs.

- Distribution by per-packet.
- Small variation among ECMP members.
- Improved link utilization efficiency is expected.
- Reorders may occur.

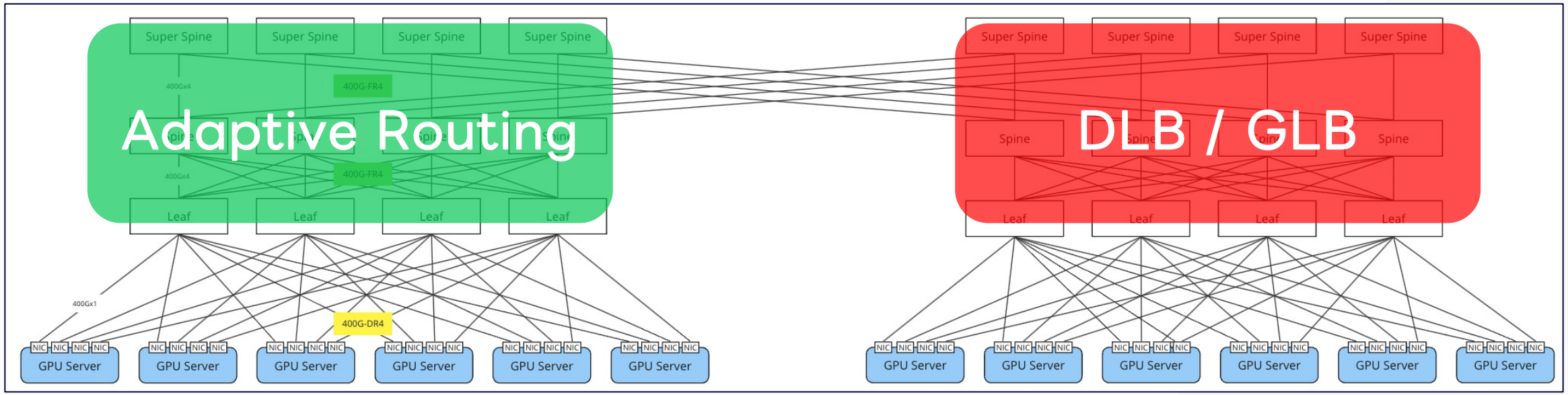
Changed distribution mode

⚠ It is not recommended to change the distribution mode during operation due to service impact.

Why focus on load balancing technology?

When expanding GPU clusters, it is possible that switches with different implementations may be mixed.

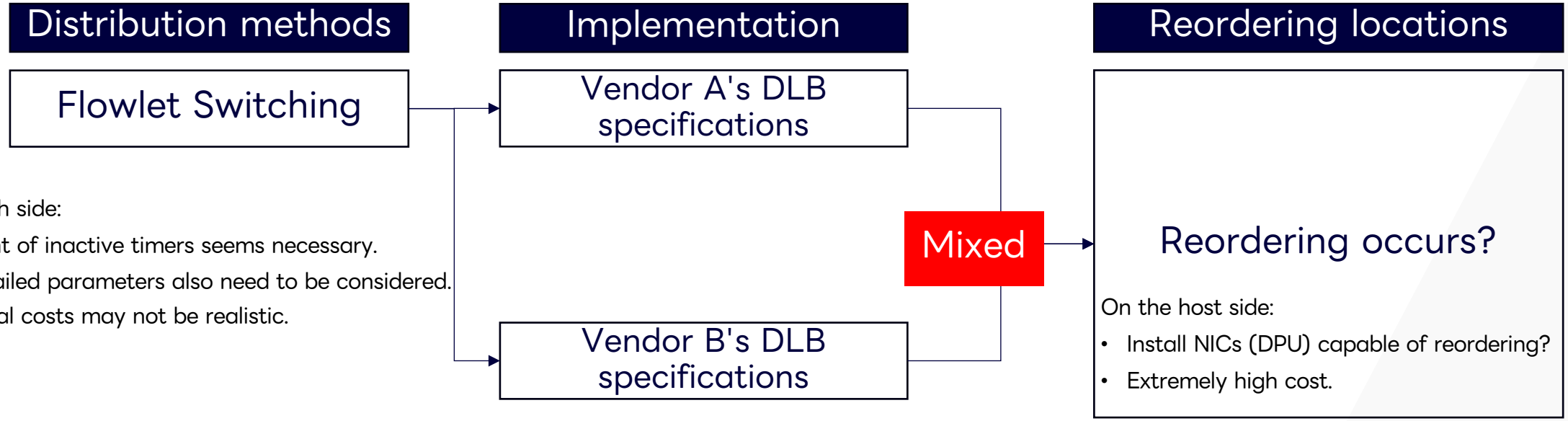
- The intelligence used in internal algorithms varies depending on the load balancing method.
  - Sampling Rate, Inactivity Timer, Interframe Gap, Packet Flag, etc.
- When multi-vendor switches are used during cluster expansion, these differences can affect performance.
  - Specifically, it may be necessary to reorder out-of-order packets on the receiving side.



Configuration of the GPU cluster at LY Data Center (Rail-optimized Backend Clos):
Different load balancing algorithms are expected to be mixed during future cluster expansions.

Considerations for mixed configurations

Is it possible to mix multi-vendor switches if they use the same method?



On the switch side:

- Adjustment of inactive timers seems necessary.
- Other detailed parameters also need to be considered.
- Operational costs may not be realistic.

Points of concern

- GPU generations evolve quickly, deciding whether to add to the existing NW or create separate clusters for each generation.
- When adding to the existing NW, switches from different vendors may be included due to circumstances at the time.
- When constructing Lossless Ethernet with multi-vendors, the load balancing method might become a technical consideration.

  → We may be able to answer this in future JANOG meetings (currently under verification and construction).

Points for discussion

We would appreciate your feedback at the venue or on Slack!

- How the network of the GPU cluster is constructed.
  - How much concern there is for the distribution and reordering of RoCEv2 flows?
  - Criteria for selecting load balancing technology.
  - Adjusting parameters according to the environment and workload.
- Considering multi-vendor RoCEv2 networks.
  - Points to consider.
  - Separating networks based on GPU generations and uses.
- As a network operator, what is missing in the current AI/ML network (Lossless).
  - Functions, observability, etc.