大規模ネットワークにおける
障害可視化のためのアーキテクチャについて

NTTフィールドテクノ
佐藤 亮介 森野 雄也 野中 建吾, 榊原 寛紀, 重松 勇也, 伊藤 良太

太字:本日登壇

メンバ紹介

・名前: 佐藤 亮介
・現担当: NW運用/保守
・JANOG歴: 初登壇

・名前: 野中 建吾
・現担当: IP装置 開発/検証
・JANOG歴: 登壇2回目

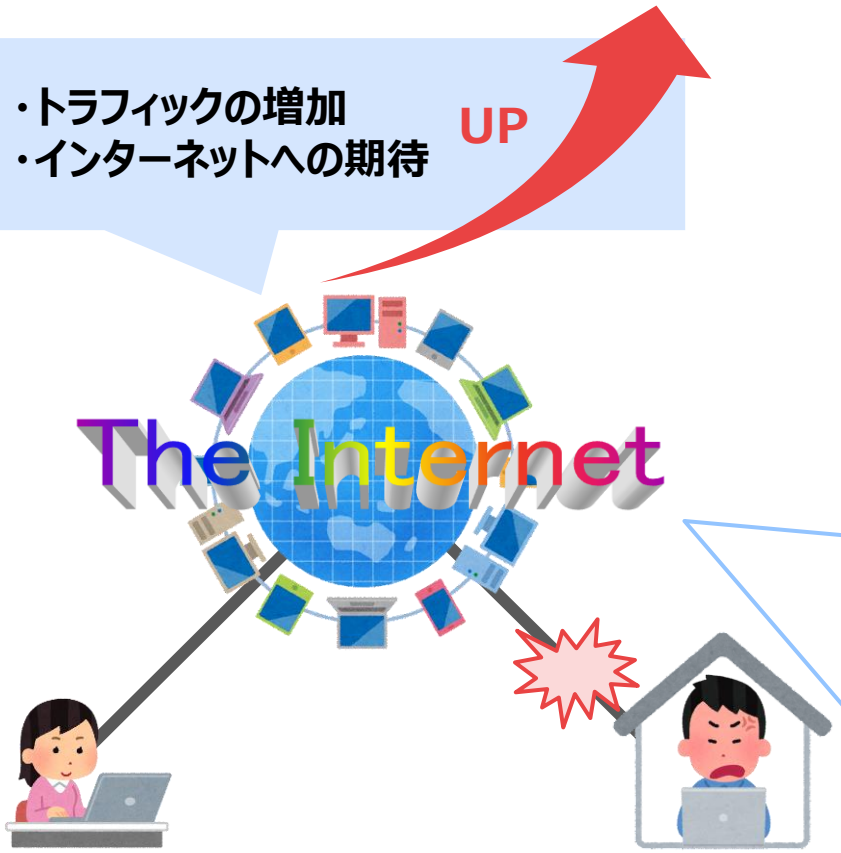
・名前: 森野 雄也
・現担当: NW運用/保守
・JANOG歴: 初登壇

・名前: 榊原 寛紀
・現担当: システム開発
・JANOG歴: 登壇2回目

取り組みの背景

- リモートワークの普及や大規模なネットワーク障害を通してNW保守の重要性が高まっており
  NTTフィールドテクノのNOCでは日々**<u>迅速な障害復旧</u>**に向けて努めています

・トラフィックの増加
・インターネットへの期待 UP

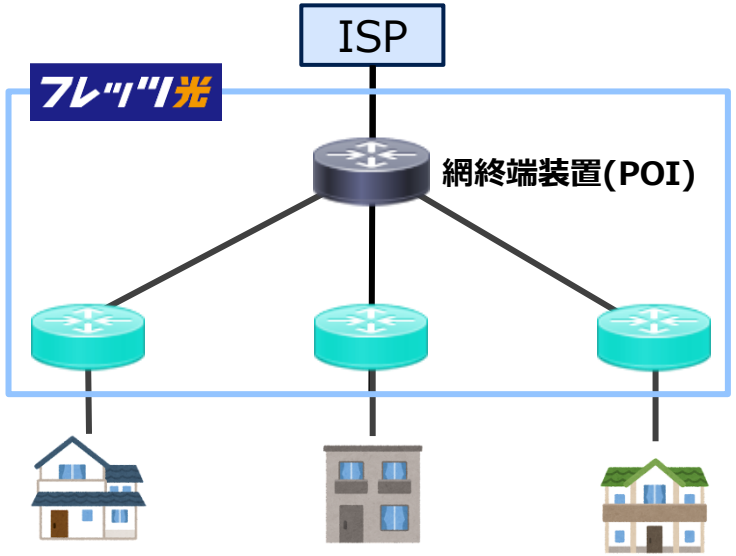


NTTフィールドテクノのNOCでは・・・

**①NTT西日本サービスにおける
NWの運用保守を実施！**

**②迅速な障害復旧に向けた
オペレーション高度化を推進！**

運用装置数は数十万台規模!



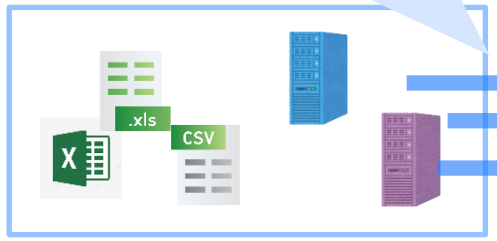
保守に加えて
NW構成図可視化
等の**オペレーション高度化**も実施

大規模NWにおける障害可視化に関する課題

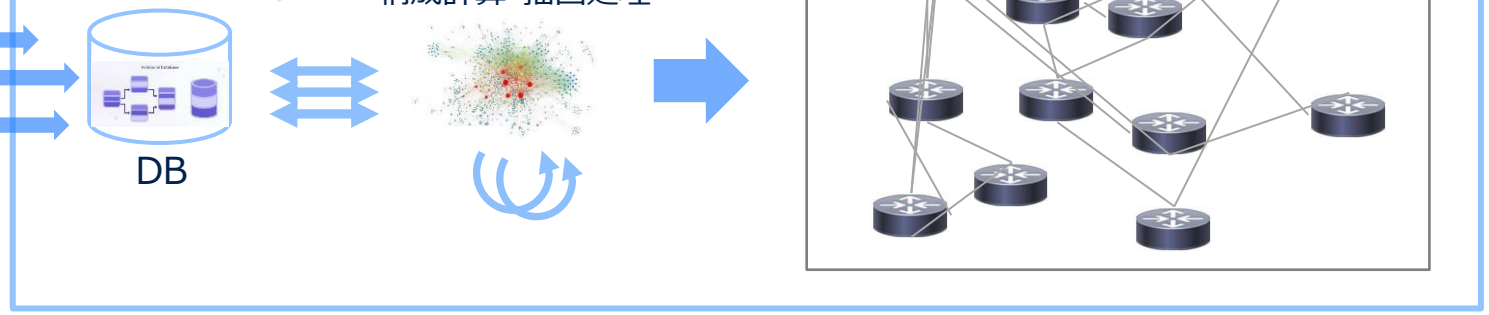
- NW障害の迅速な分析には，**NW装置の構成図と障害情報を関連付けた可視化**が有効．
  しかし，大規模NWの環境においては，OSS等を単純に利用するだけではカバーの難しい課題が存在する．

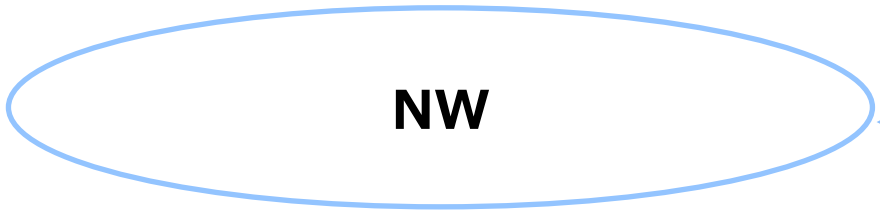
可視化システムアーキテクチャ

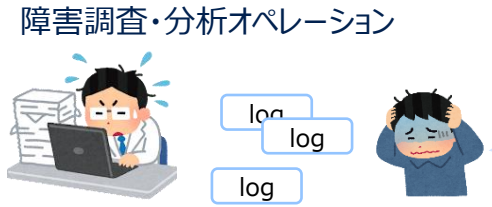
【課題①:SSoTとなる装置情報の確保】
・信頼して取り込める単一情報源
（Single Source of Truth）がない



?



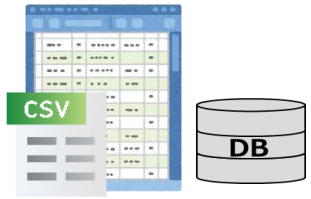


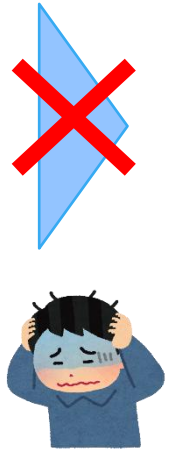


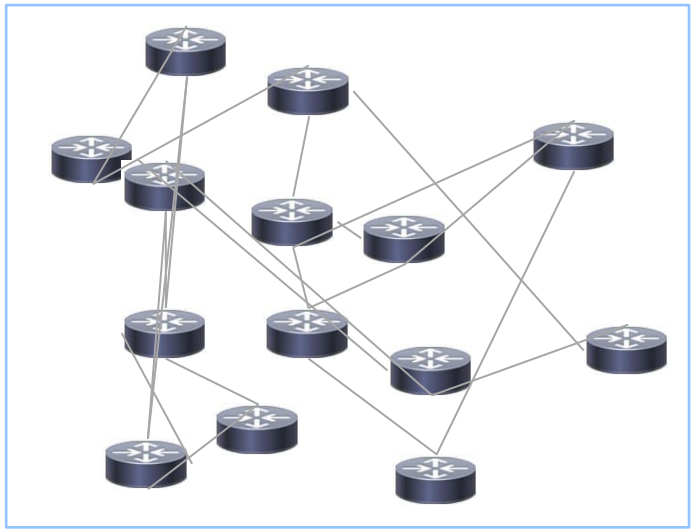
【課題③:障害分析の稼働】
・障害分析に必要な情報の
調査オペレーションに時間がかかる
・データが多く分析が難しい

【課題①】SSoTとなる装置情報の確保

- 可視化に必要な装置情報は信頼できる単一情報源（**S**ingle **S**ource **o**f **T**ruth）となることが望ましい.
- しかし大規模NWでは, **多様な機種の存在や**, 設計から運用フェーズの過程で**管理情報の不統一**となることも多く, 一般的なプラクティスでの**SSoTとなる装置データの確保が難しい**.

過去情報が残っており、管理情報が現用機と一致しない場合がある	機種やOSに差分があり、単一の方法で、必要な情報が取得できない
	
構成描画に必要な情報の管理方法に差異がある	**LLDPのような構成描画に便利なプロトコルが利用できない設定**
	



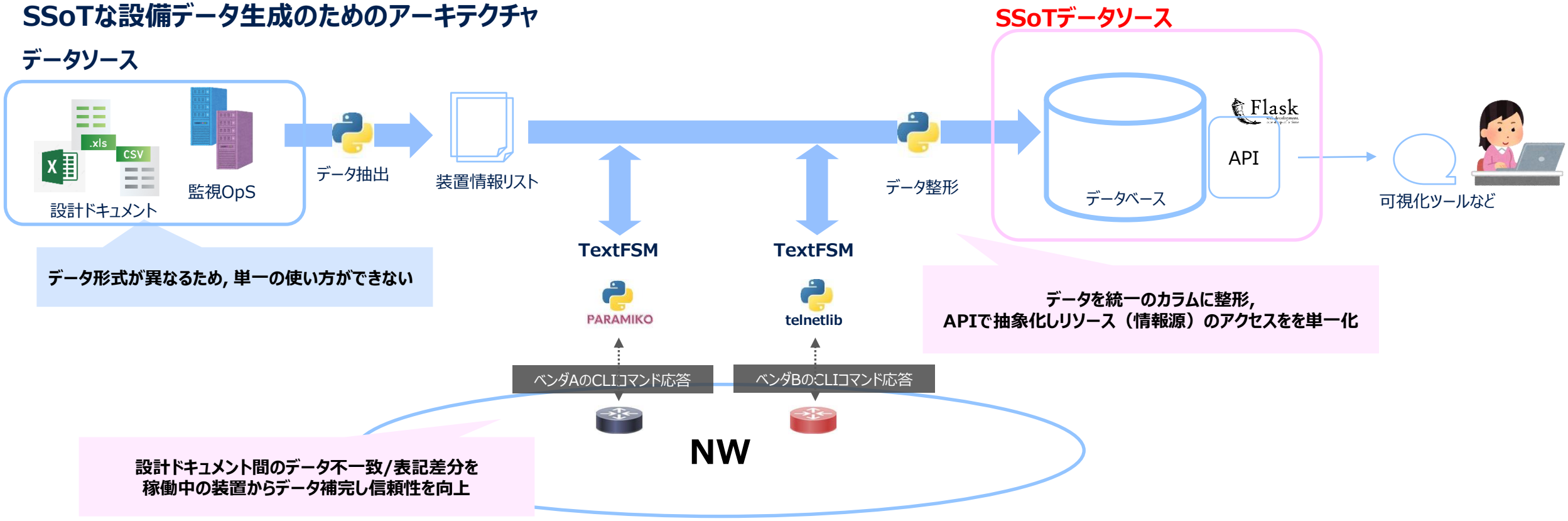


【課題①】SSoTとなる装置情報の確保に関するアプローチ

SSoTな情報源を実現するために

- 管理情報を元に**稼働中の装置から機種ごとの方式で情報を抽出する**ことで信頼性を担保し，統一形式に装置情報を整形・保管。
- **APIを実装する**ことで他システムから同一の方法で利用できるインタフェースを提供。

SSoTな設備データ生成のためのアーキテクチャ



（参考）装置情報（インターフェース）の取得例

- コマンドの出力から，インターフェースに関連する情報を抽出する．
- 機種ごとの出力差分は，パーサの実装で吸収（TextFSMのパーサファイル）

■装置Aのコマンド結果

```
vendorA-host001> show interfaces
Nov 16 16:50:56
Physical interface: lc-0/0/0, Enabled, Physical link is Up
・・・
```

■装置Bのコマンド結果

```
vendorB-host002#show interfaces
Load for five secs: 8%/2%; one minute: 7%; five minutes: 7%
Time source is NTP, 16:55:55.865 JST Thu Nov 16 2023
TenGigabitEthernet1/0/0 is up, line protocol is up
・・・
```

装置A
パーサファイル

装置B
パーサファイル

■DB内に格納されたデータ

hostname	If_type	If_number	Link_state
vendorA-host001	lc-	0/0/0	Up
vendorB-host002	TenGigabitEthernet	1/0/0	up

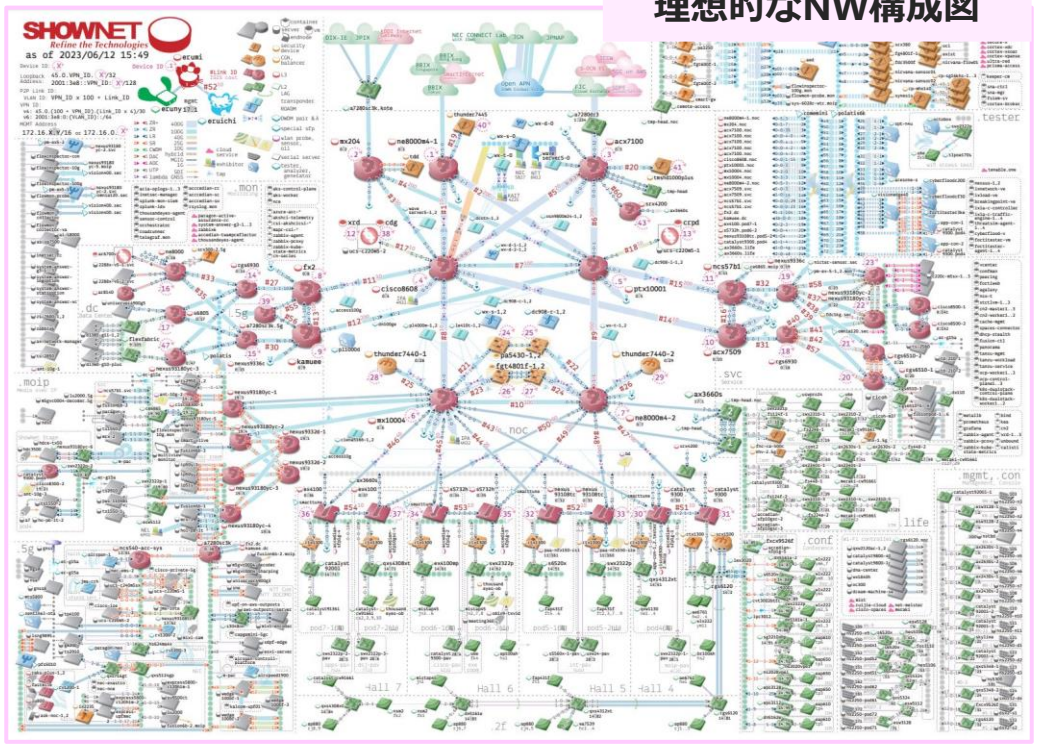
**設計ドキュメント間のデータ不一致/表記差分を
装置の応答をベースに補完処理しDBへ登録することで、
信頼性の高いDBを作成する**

【課題②】構成図の描画処理負荷に関する課題

- 大規模NWにおいては関連装置が多く処理時間の増加や，多方路なサービスパスにより，**現実的な処理時間**と**視認性の高い描画**の両立が必要
- 一方で一般的なRDBを用いた検索処理のみでは，**実装の煩雑さと処理速度の観点から実現が難しい**

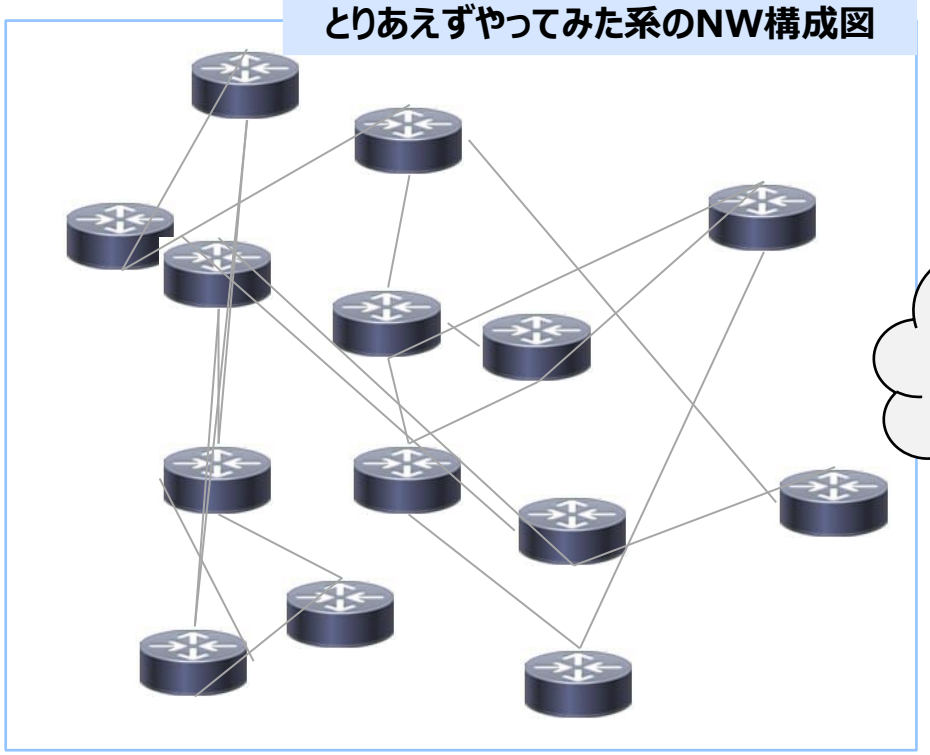
■NW構成図作成の難しさ

理想的なNW構成図



https://www.interop.jp/2023/shownet/topology.pdf

とりあえずやってみた系のNW構成図



【課題②】構成図の描画処理負荷に関するアプローチ

- **処理に応じDB形式を使い分ける**ことで，実装の容易さと処理性能の両立を実現
- 対象装置に**関連性のある装置のみ**を表示することで**描画負荷を軽減**

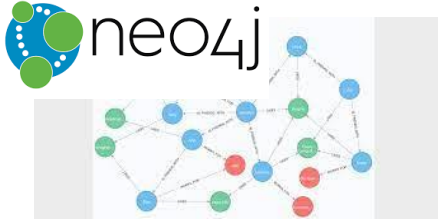
装置情報検索
(ホスト名,機種,アドレス等)

MySQL

リレーショナル DB

接続情報/NW情報検索
(物理・論理接続,IGP等)

neo4j

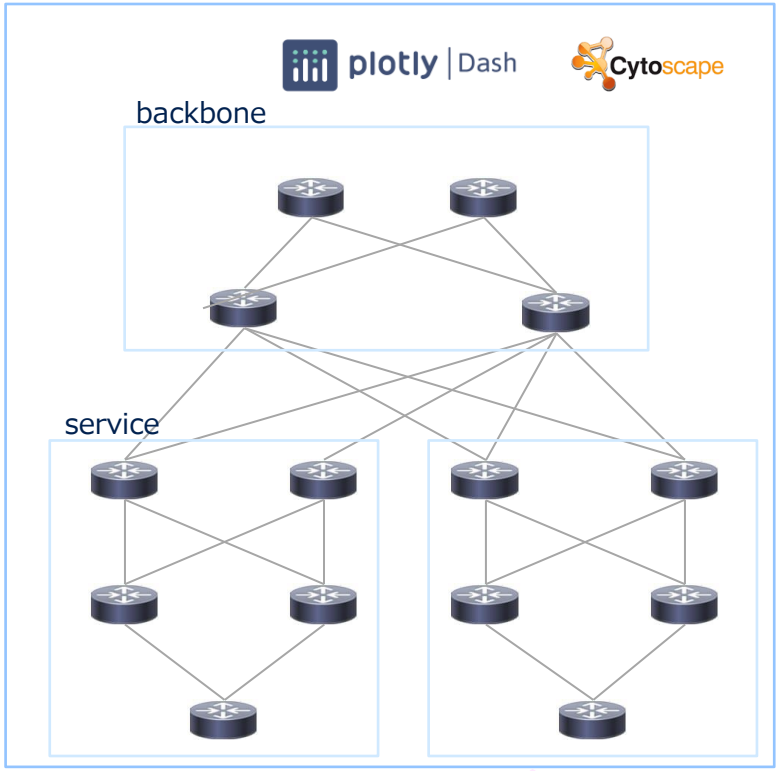


グラフ DB

経路計算処理
(ルートフィルタ,ACL,パス計算)

NetworkX
Network Analysis in Python

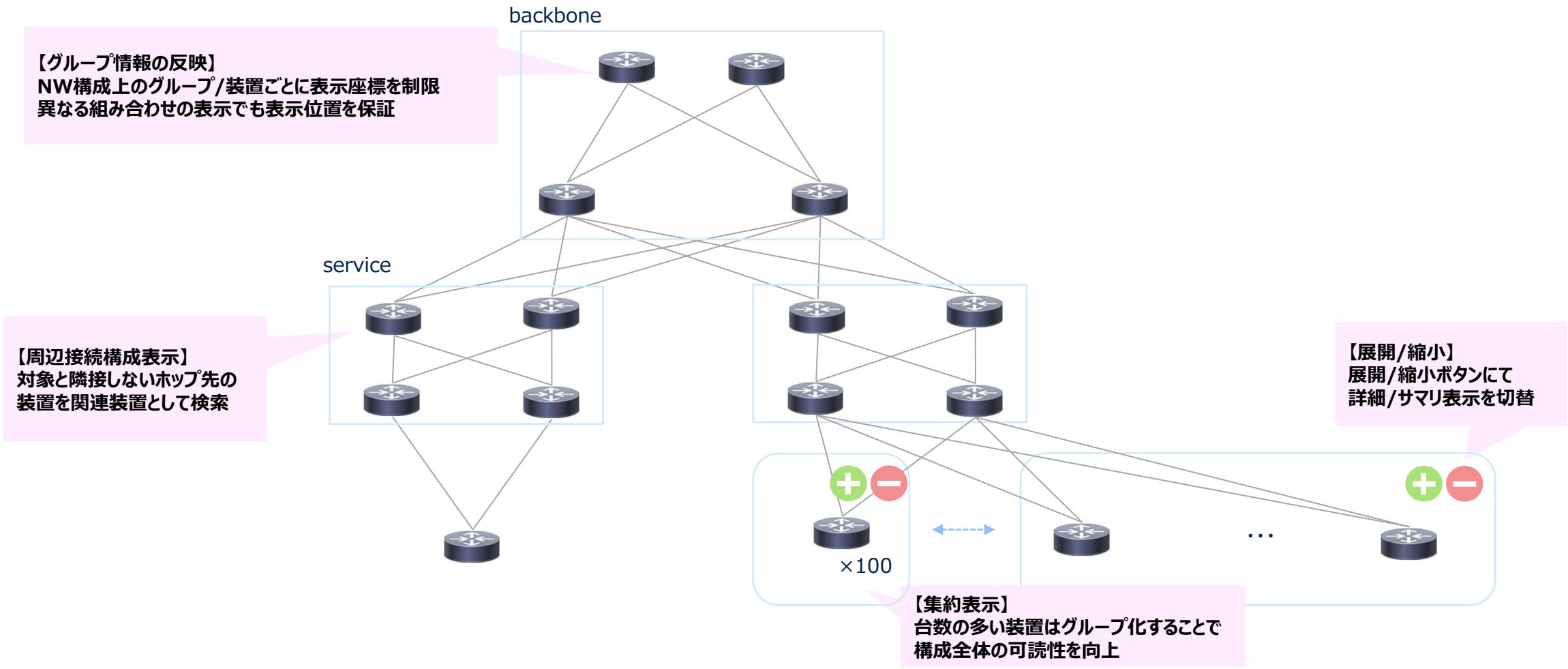
グラフ解析用ライブラリ



関連性のある装置に絞ることで処理の付加を軽減
一定のルールベースで初期表示位置を調整

（参考）NW構成の可視化の具体例

- 指定装置の周辺装置（数ホップ先まで）の接続構成を自動計算し表示
- 分析観点での視認性を高めるロジックをレイアウトに反映



（参考）リレーショナルDBとグラフDBの比較

https://jp.drinet.co.jp/blog/datamanagement/graphdbms_featured

グラフデータベースの検索速度

では、グラフDBはどれほどの速さでクエリを実行できるのでしょうか？
Partner氏とVukotic氏は二人の著書「Neo4j in Action」にて、リレーショナルDBとグラフDBの検索クエリの実行スピードの比較を行っています。実験はソーシャルネットワーク内の「友達の友達・・・」を検索するというもので、各人が50人の友達を持つ100万人規模のソーシャルネットワークを想定しています。

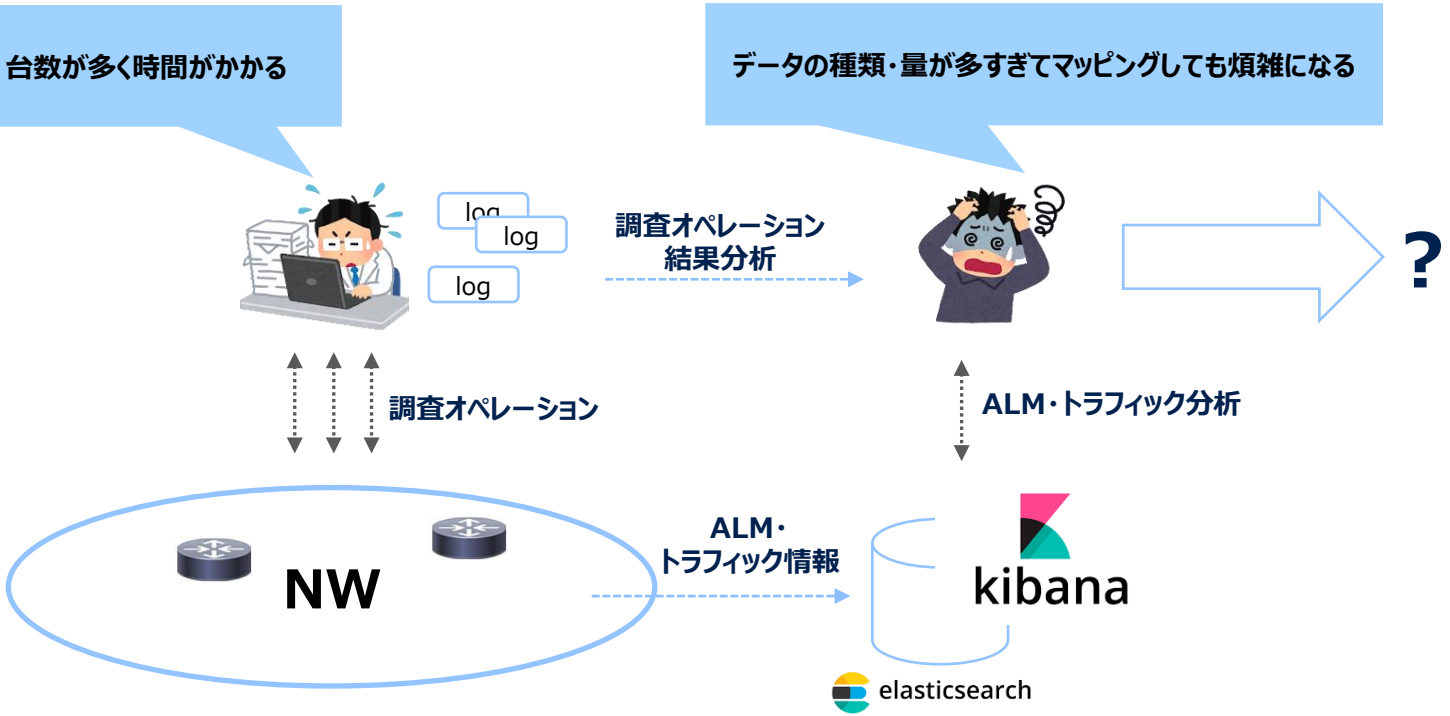
比較結果は以下の通りです。
（下の表の「深さ」は検索の段階を表します。たとえば、深さ"2"は「友達の友達」で、深さ"3"は「友達の友達の友達」を表します。）

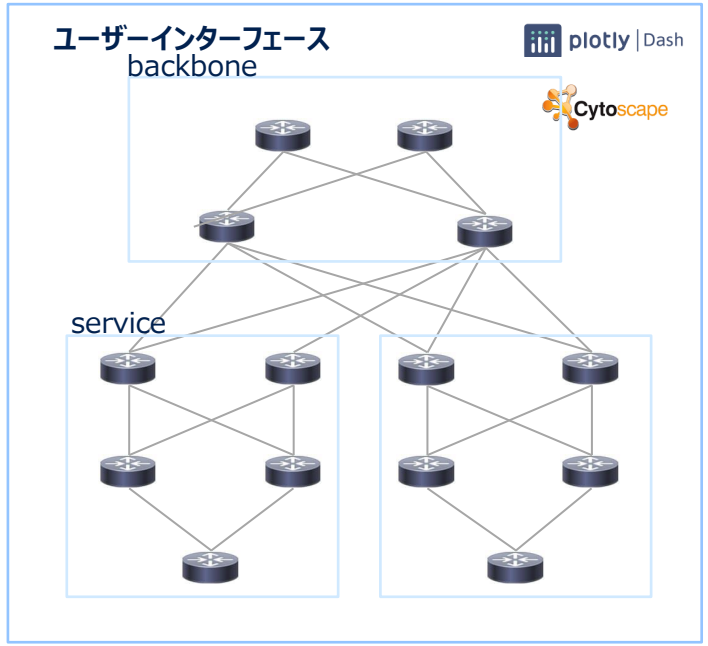
深さ	リレーショナルDB (MySQL)(秒)	グラフDB (Neo4j)(秒)	返されたレコード数
2	0.016	0.01	～2500
3	30.267	0.168	～11万
4	1543.505	1.359	～60万
5	未完了	2.132	～80万

表：リレーショナルDBとグラフDBの検索スピード比較
（出典: Neo4j in Action）

【課題③】障害分析の稼働に関する課題

- NW障害の分析にはALM・トラフィックに加え様々な調査オペレーションの結果データを分析する必要があるが，大規模NWでは**調査オペレーション自体に時間がかかる**ことに加え，多種多量なデータに対し**単純に構成図へのマッピングを行っても煩雑**となり効率的な分析が行えない.

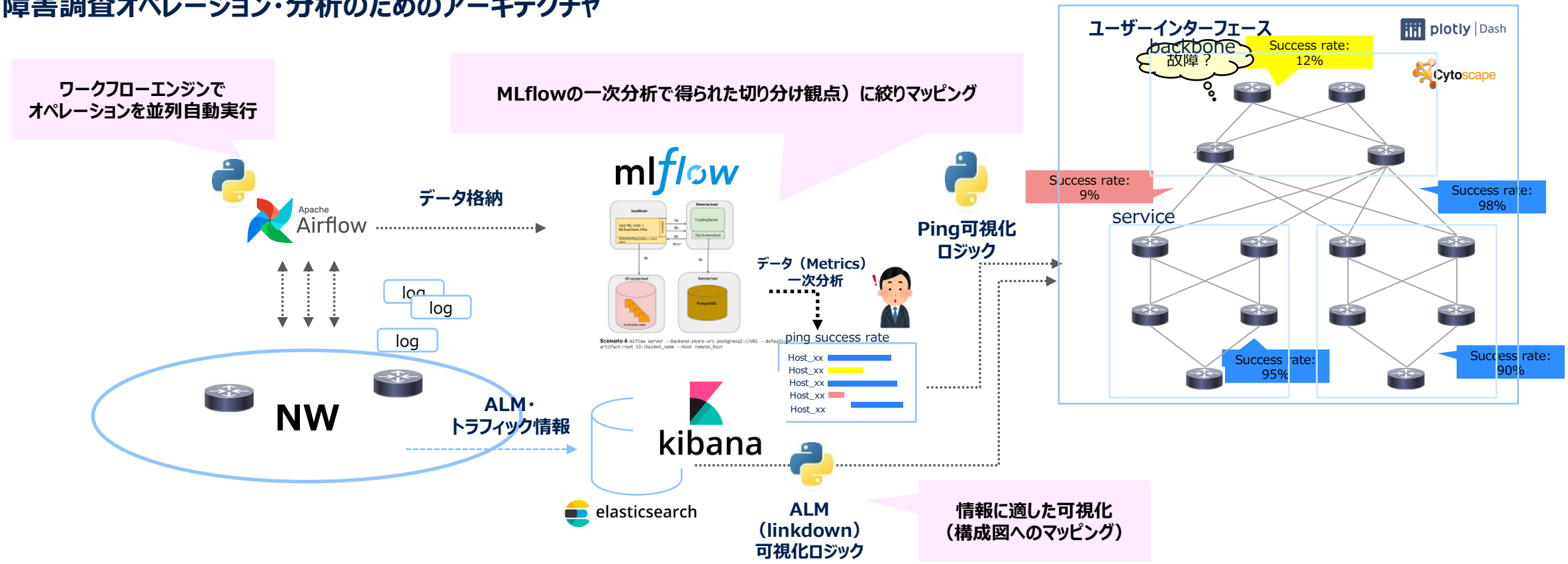




【課題③】障害分析の稼働に関するアプローチ

- 以下のようなアプローチで分析の迅速化を試みている.
  - ワーク**フローエンジンによる実行時間短縮**
  - データを**実験管理DB**による分析の効率化
  - オペレーション情報ごとに，可視化ロジックを個別に実装し**視認性の改善**

障害調査オペレーション・分析のためのアーキテクチャ



(参考)ALM情報（LinkDown）のマッピングによる可視化の例

- NW障害分析においては，LinkDown系のALMからNW構成上実際に接続断となっている区間を早期に把握することが障害原因・影響の特定のために特に重要となる．
- 手動では時間がかかりOSSで機能を賄えないため，**ALM断区間を自動判断し構成図にマッピングする可視化ロジックを個別実装**している．

可視化ロジックのイメージ

elasticsearch

ALM発生時刻	ホスト名	メッセージ	
…	…	…	
2023-12-13-13:01	hostA	Interface 1/1 changed state to down	断発生
…	…	…	
2023-12-13-13:00	hostC	Interface 2/1 changed state to down	
…	…	…	
2023-12-13-11:21	hostD	Interface 1/1 changed state to up	断回復
2023-12-13-11:20	hostC	Interface 1/2 changed state to up	
…	…	…	
2023-12-13-11:10	hostC	Interface 1/2 changed state to down	断発生
…	…	…	
2023-12-13-11:09	hostD	Interface 1/1 changed state to down	
…	…	…	

メッセージをパースし
IFがdown状態の装置部位を
抽出

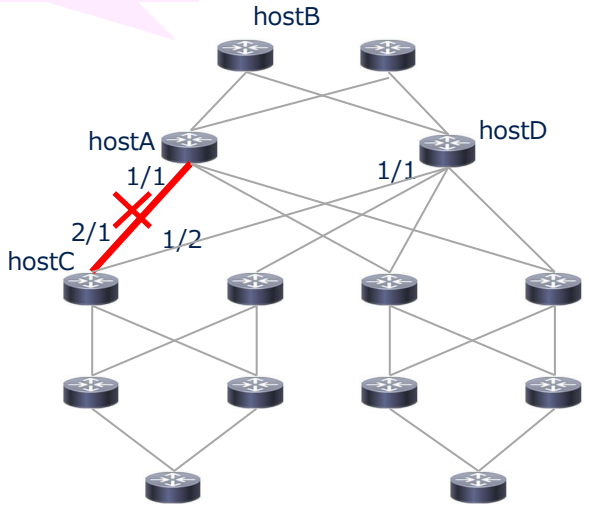
接続構成と紐づけ
断区間を判断

neo4j

回復が確認された区間
は可視化しない

ユーザーインターフェース

構成図上に断区間を可視化
（接続を強調表示）



(参考)MLFlowの一時分析(ping結果)のマッピング可視化のデモ

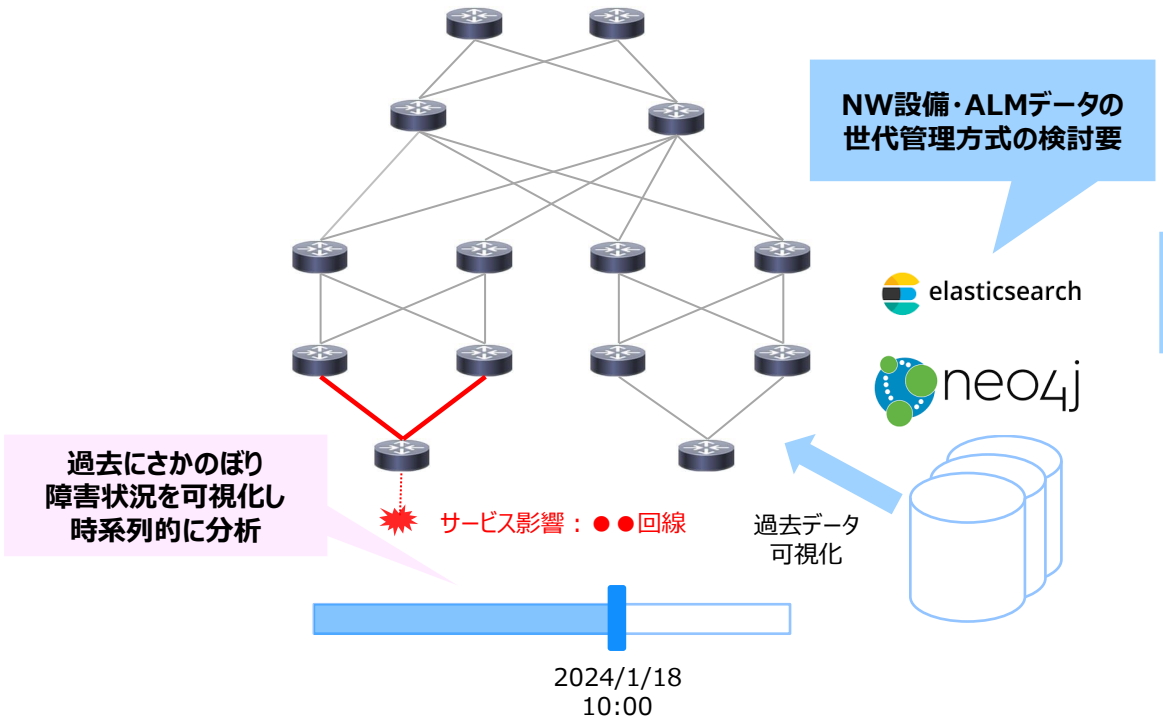
- 原因不明の通信不可申告を契機としNW内の障害分析を行う場合をユースケースとして，可視化アーキテクチャの運用例を示す

本番では，デモをお見せします

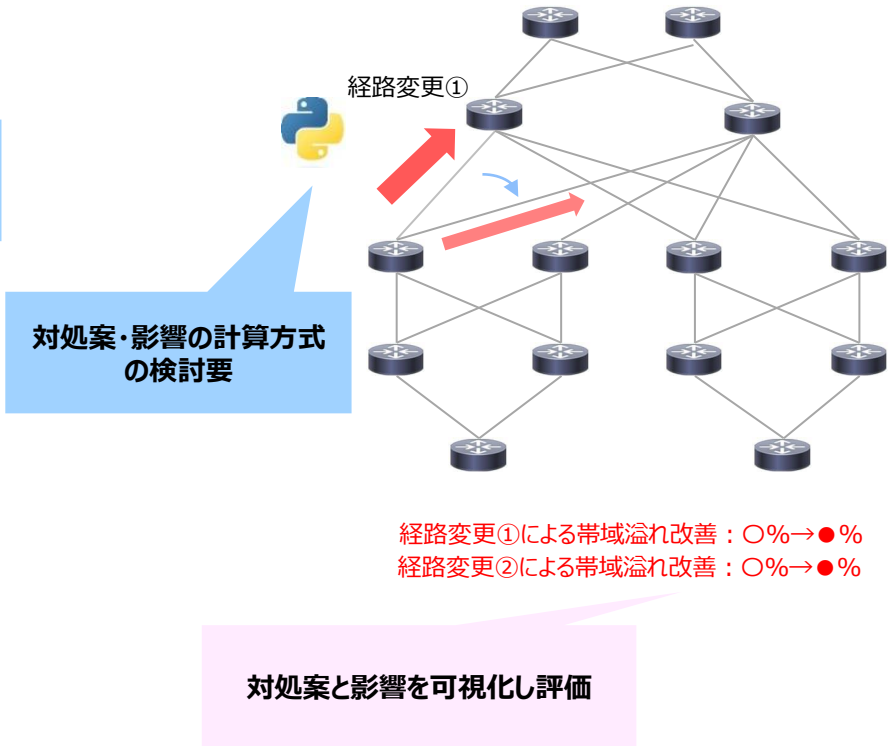
今後の取り組み方針

- これまで**NWの現在状況を分析**するための可視化アーキテクチャを検討してきた．今後は，下記のような拡張に取り組む予定
  - 障害分析の高度化のための，**過去のNWの状況の時系列的分析**の可視化アーキテクチャの検討
  - より迅速な障害復旧のための，**対処案の評価シミュレーション**の可視化アーキテクチャの検討

時系列分析の可視化イメージ



対処案評価シミュレーションの可視化イメージ



会場参加者との議論ポイント

- 可視化のベースとなる，**信頼できる装置・接続情報（Single Source of Truth）をどのように維持**しているか
- 可視化の**スケール（装置台数）**はどの程度か，**レスポンス・視認性**に関する課題感はどのようなものか
- どのような情報を構成図と関連付けて可視化したいか

などの大規模ネットワークにおける運用上のノウハウ

- ネットワーク可視化を実現するアーキテクチャと技術スタック
- 紹介した技術スタック（**cytoscape, plotly, neo4j, networkx, airflow, Mlflowなど**）に関するメリット/デメリット

などの可視化技術に関するノウハウ