

Redmineデータのグラフデータ化により システム監視・運用業務を効率化

～ 関連チケットの可視化、類似チケットのレコメンド ～

2023年11月4日

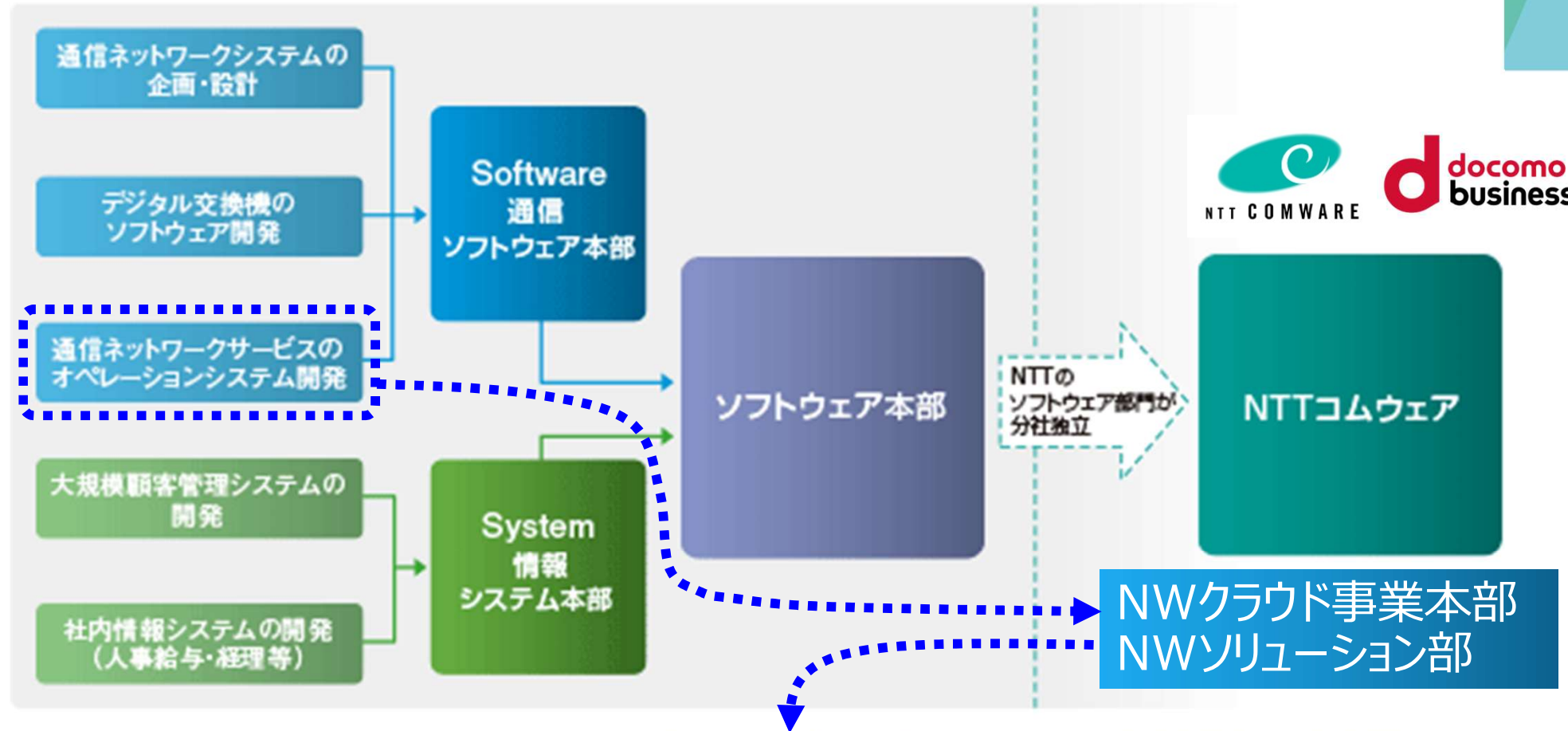
NTTコムウェア株式会社

ネットワーククラウド事業本部NWソリューション部TNX-BU

会社・組織紹介

【NTT時代(分社前)】

1997年(分社)



- NWを知り尽くし、**ミッションクリティカルなシステムを開発・運用できるシステムインテグレーター**
- Redmineは、組織としてもプロジェクト管理等で広く利用されている

本日紹介したいこと

【その1】

グラフデータベースとは

グラフデータベース(グラフDB)の概要、リレーショナルデータベース(RDB)とグラフDBの性能検証結果の紹介

【その2】

RDBとグラフDBの連携ソリューション紹介

RDBから任意のデータをグラフ形式に変換しリアルタイムでグラフDBに反映するソリューション“Graphyde”を少しだけ紹介

【その3】

Redmineデータのグラフデータ化とその活用

Redmineで管理しているシステム監視・運用業務のチケットデータをグラフデータ化し、そのデータを活用して業務を効率化する方法をデモも交えて紹介

【その1】

グラフデータベースとは

グラフデータベース(グラフDB)の概要、リレーショナルデータベース(RDB)とグラフDBの性能検証結果の紹介

1. グラフデータベースとは

- NoSQLに分類
- グラフ理論に基づく、グラフ構造に最適化
- 点(ノード)と線(リレーション、エッジともいう)で構成
- ネットワーク状になっているデータの管理に有効
- Neo4jが圧倒的シェア

ソーシャルネットワーク、経路検索、NWやICTリソースの管理、不正検知等で使われている

※但し、現在、日本国内ではあまり使われていない！

2018年私たちが国内で初めてNeo4jを利用したシステムをリリース

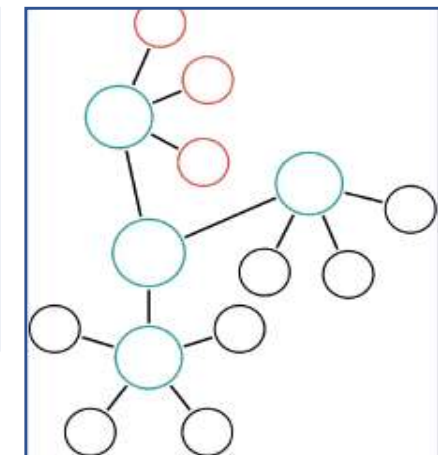


表とグラフどちらが直感的？

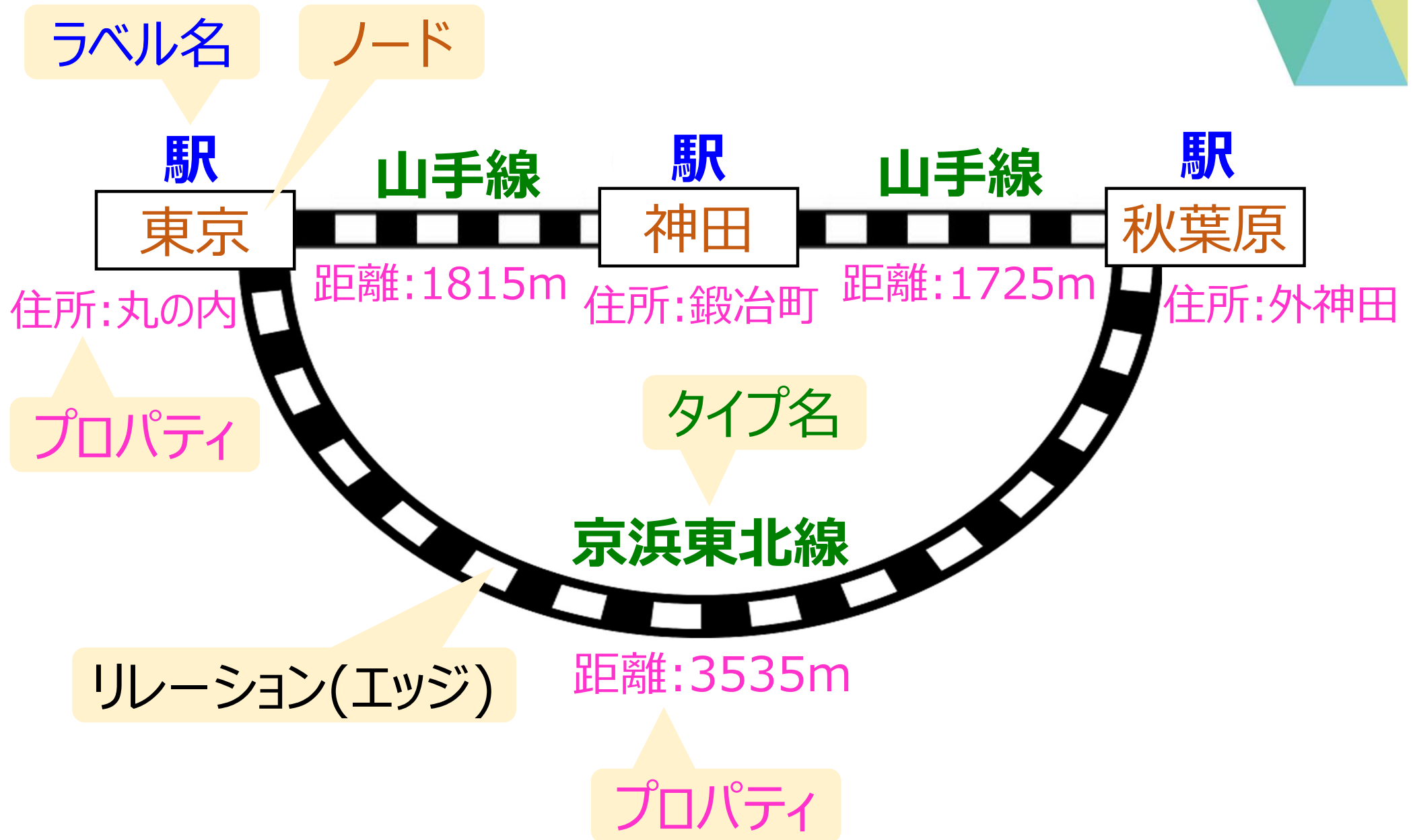
日本でもそろそろブーム到来？

- 2021年3月のガートナーレポートでは、データ解析分野の約10%でグラフ技術が利用されているが2025年には**80%まで利用率が上がる**と予想されている
- Oracle最新バージョン(23c)にGraph機能が実装された

id	name	age	gender	height
1	Paul	25	Male	175
2	Paul	25	Male	175
3	Paul	25	Male	175
4	Mark	30	Male	180
5	Mark	30	Male	180
6	Mark	30	Male	180
7	Luc	35	Male	185
8	Luc	35	Male	185
9	Luc	35	Male	185
10	Luc	35	Male	185
11	Luc	35	Male	185
12	Luc	35	Male	185
13	Luc	35	Male	185
14	Luc	35	Male	185
15	Luc	35	Male	185
16	Luc	35	Male	185
17	Luc	35	Male	185
18	Luc	35	Male	185
19	Luc	35	Male	185
20	Luc	35	Male	185
21	Luc	35	Male	185
22	Luc	35	Male	185
23	Luc	35	Male	185
24	Luc	35	Male	185
25	Luc	35	Male	185
26	Luc	35	Male	185
27	Luc	35	Male	185
28	Luc	35	Male	185
29	Luc	35	Male	185
30	Luc	35	Male	185
31	Luc	35	Male	185
32	Luc	35	Male	185
33	Luc	35	Male	185
34	Luc	35	Male	185
35	Luc	35	Male	185
36	Luc	35	Male	185
37	Luc	35	Male	185
38	Luc	35	Male	185
39	Luc	35	Male	185
40	Luc	35	Male	185
41	Luc	35	Male	185
42	Luc	35	Male	185
43	Luc	35	Male	185
44	Luc	35	Male	185
45	Luc	35	Male	185
46	Luc	35	Male	185
47	Luc	35	Male	185
48	Luc	35	Male	185
49	Luc	35	Male	185
50	Luc	35	Male	185



【参考】グラフDBの構成要素



2. RDBとグラフDBの検索処理

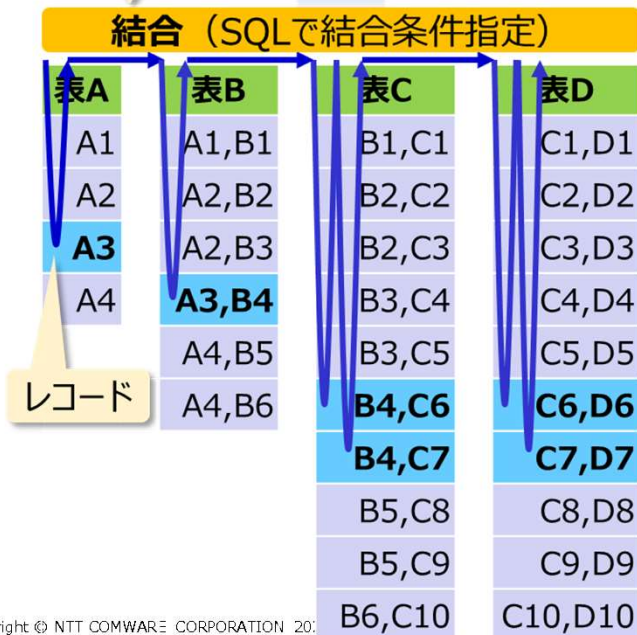
RDBの検索処理

- 表を順次検索、それをSQLで結合して結果を出力
- よって、検索対象の表が多く、結合条件が複雑になればなるほど、パフォーマンスが劣化

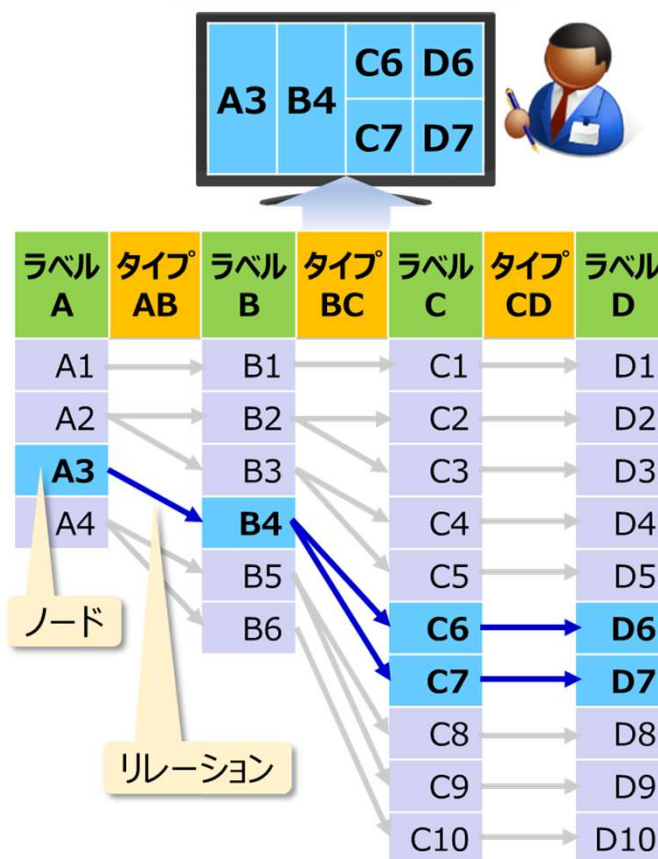
グラフDBの検索処理

- ノードとノードの関係を表すリレーション情報をあらかじめ実態として保持
- そのため、ノードからノード、更にその先にひもづくノードの情報を即座に抽出することが可能

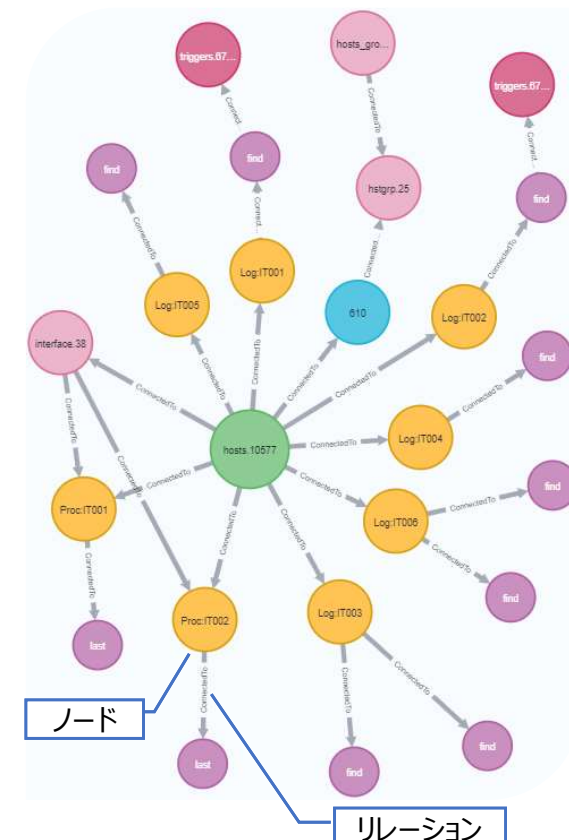
リレーショナルデータベース



グラフデータベース



グラフデータのイメージ

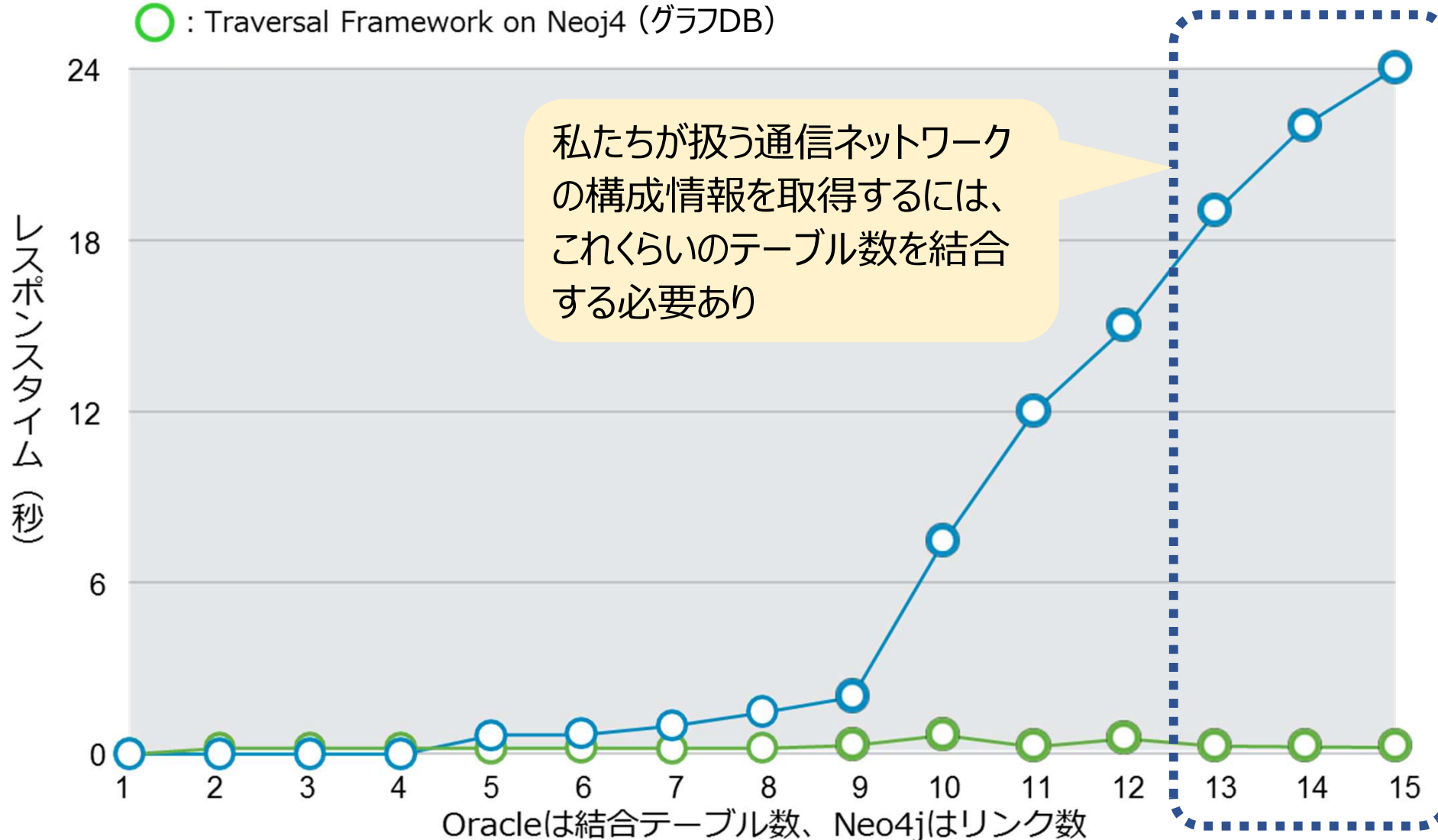


3. RDBとグラフDBの性能検証

3-1. 今回ターゲットにした複雑度

以下はNeo4j社の検証結果

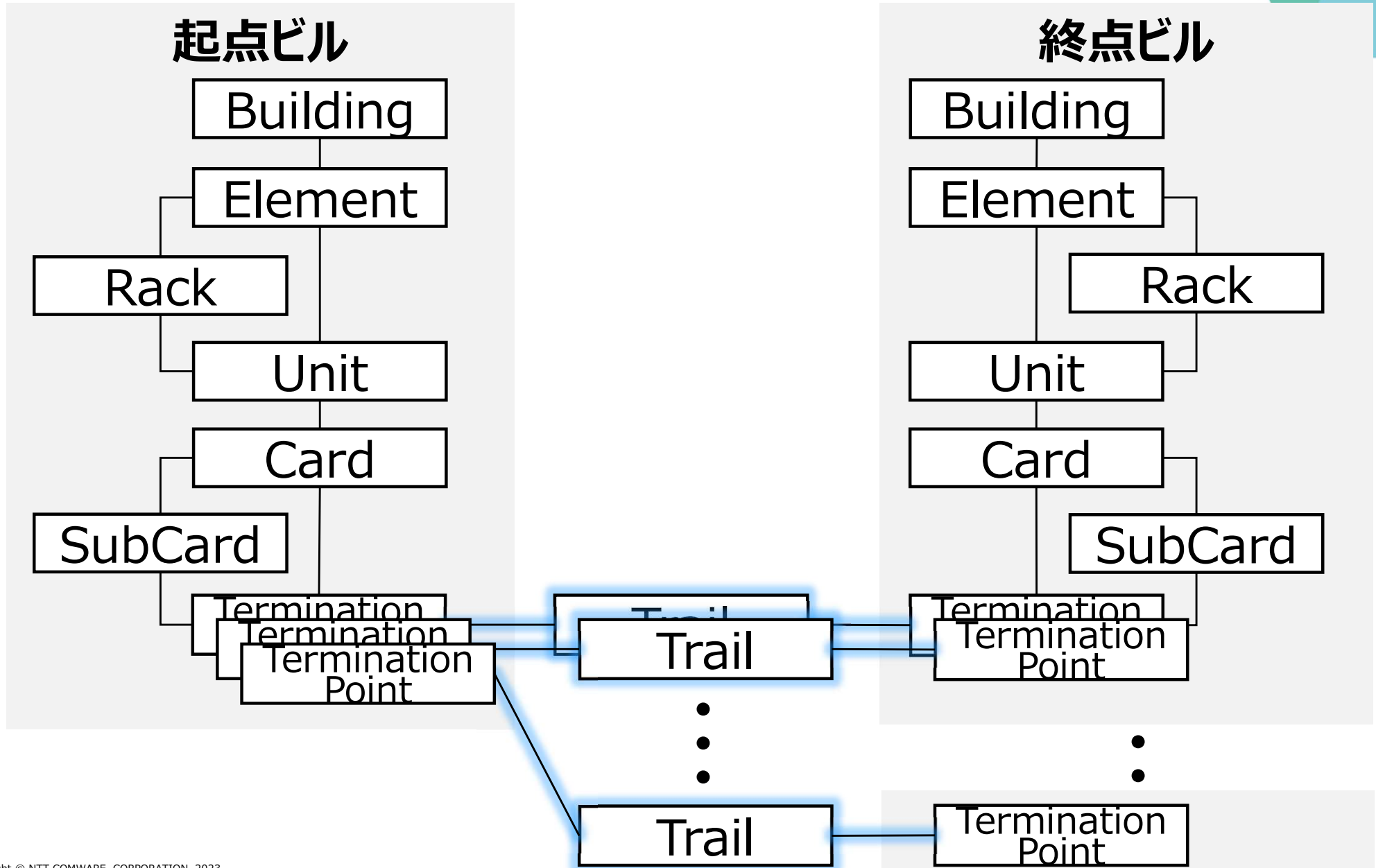
- : SQL on Oracle Exadata (RDB)
- : Traversal Framework on Neo4j (グラフDB)



3. RDBとグラフDBの性能検証

3-2. 検証で利用したデータモデル

- 検証に使用したNW構成を表すデータモデル



3. RDBとグラフDBの性能検証

3-3. 複数データの比較結果

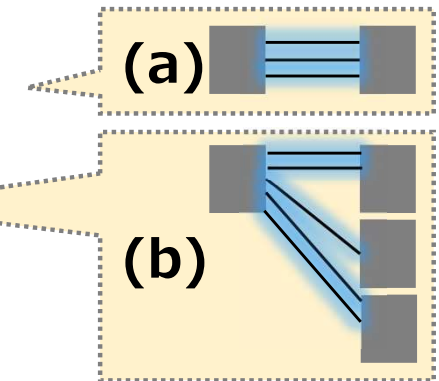
複雑な複数データの検索 (検証合計データ: 12,500,000件)



ビルを跨るネットワークと、そのネットワークを収容するビル内の設備情報を検索

● 2パターンでNW構成を検索

(a) 起点ビルと終点ビルを指定: NW数2,000件抽出

(b) 起点ビルのみ指定: NW数20,000件抽出



パターン	 PostgreSQL	 neo4j
(a)	44.287秒	2.512秒
(b)	4,851,142秒 (約80分51秒)	平均20.626秒

【参考】性能検証で使したCypher Query

性能検証(a)のCypher Query

```
MATCH
  (b1:Build)<-[]-(pd1:PhysicalDevice)<-[]-(trail:Trail)-[]->(pd2:PhysicalDevice)-[]->(b2:Build)
WHERE
  b1.building_id = '13104 ' // 起点ビル
  AND
  b2.building_id = '13106 ' // 終点ビル
RETURN
  b1.building_id, b1.name, trail.trail_id, b2.building_id, b2.name;
```

性能検証(b)のCypher Query

```
MATCH
  (b1:Build)<-[]-(pd1:PhysicalDevice)<-[]-(trail:Trail)-[]->(pd2:PhysicalDevice)-[]->(b2:Build)
WHERE
  b1.building_id = '13104 ' // 起点ビル
RETURN
  b1.building_id, b1.name, trail.trail_id, b2.building_id, b2.name;
```

3. RDBとグラフDBの性能検証

3-4. 単一データの比較結果

単一データの検索 (検証合計データ: 21,000,000件)

- 2つの項目のAND条件に合致する60件のデータを抽出

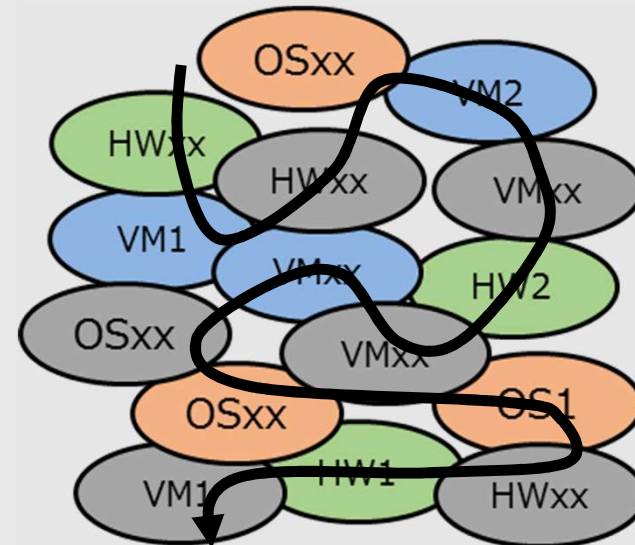


0.000948秒
(0.948ミリ秒)

ID	管理対象	名称
..
3	Server	HW1
4	VirtualMachine	VM1
5	OS	OS1
..
12	Server	HW2
..
21	VirtualMachine	VM2

RDBは単純な表走査なので高速

13.362秒
(13,362ミリ秒)



グラフDBは全体から特定のノードを見つけるのが苦手

4. RDBとグラフDBの使い分け

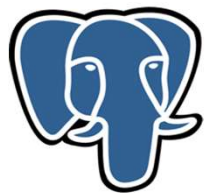
RDBを利用

- 単一データなど比較的シンプルな検索
- 曖昧検索

グラフDBを利用

- 結合データ数が多い等の複雑な検索

RDBとグラフDBのいいとこ取り！



PostgreSQL



neo4j

ハイブリッド利用を推奨！

【その2】

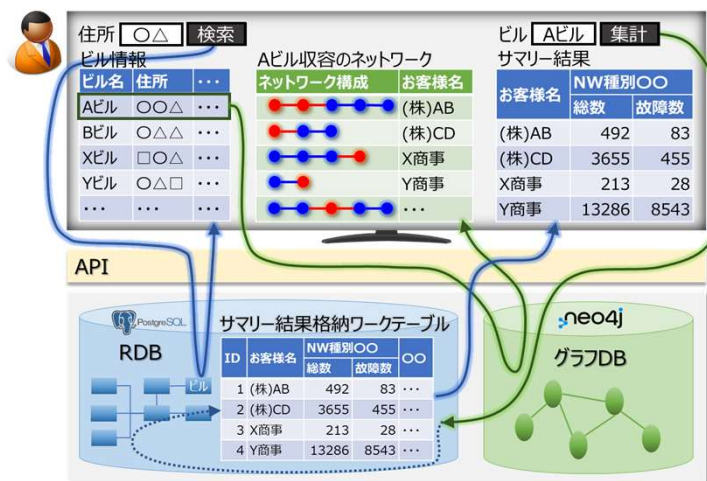
RDBとグラフDBの連携ソリューション紹介

RDBから任意のデータをグラフ形式に変換しリアルタイムでグラフDBに反映するソリューション“Graphyde”を少しだけ紹介

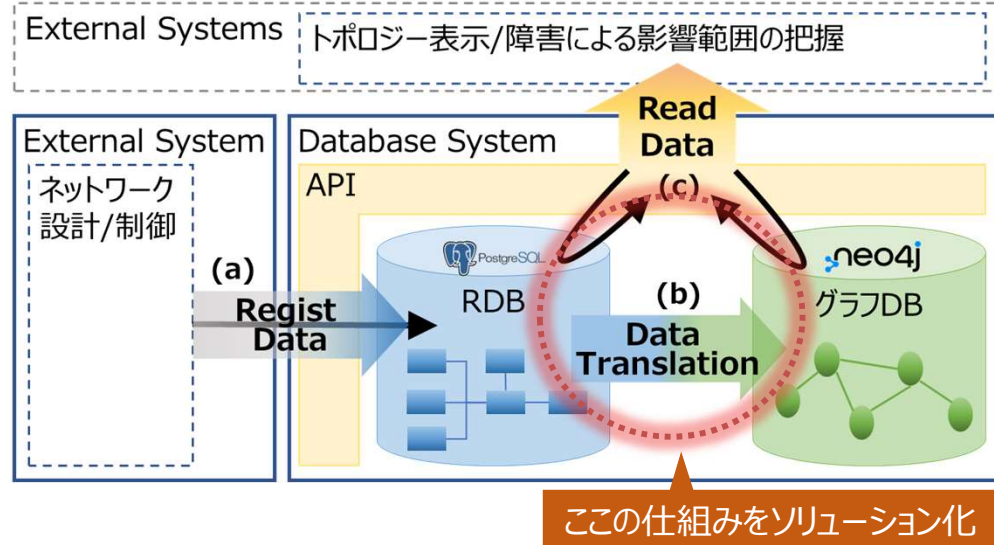
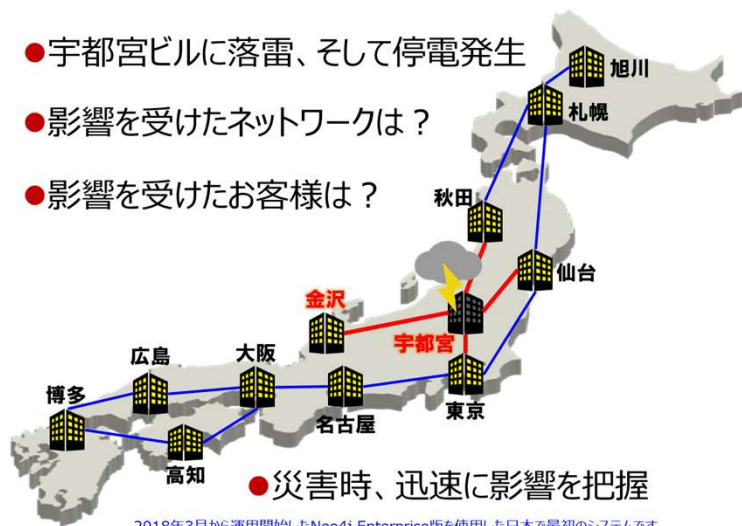
1. 実案件から提唱するハイブリッドDBシステム

- Graphydeは実案件で組み立ててきたノウハウをもとに作成した「簡単・迅速！すぐできるグラフデータ！」がコンセプトのRDBとグラフDBのリアルタイムなデータ連携を可能とするソリューション
- RDBとグラフDBのいいとこ取りをしたハイブリッドDBシステムを簡単・迅速に構築可能

災害時に影響を受けるNWの範囲を把握するシステム



- 宇都宮ビルに落雷、そして停電発生
- 影響を受けたネットワークは？
- 影響を受けたお客様は？



- 整合性の担保、単一データ検索に優れたRDB
- つながり情報の分析、高速抽出に優れたグラフDB
- いいとこ取りのハイブリッドDBシステム構築、提唱

グラフDBを簡単に利用できるようにしたい！

Graphyde Sync

※「Graphyde」は私たちが命名したNTTコムウェア株式会社の登録商標です

2. Graphyde Syncとは

お客様の課題・要求



- RDBのデータ構造が複雑化し、データ抽出に時間が掛かる
- データ分析のためにグラフDBを利用したいが、グラフデータ構築のノウハウがない
- 既にあるデータをグラフ化して、データの関係性を可視化、分析したい
- RDBとグラフDBのいいとこ取りをしたハイブリッドDBシステムを構築したい

Graphyde Sync

- 新規システムだけでなく、既存のシステムにも簡単にグラフDB(Neo4j)を組み込むことが可能
- RDBを操作(登録、更新、削除)するだけで、自動的にリアルタイムでデータがグラフDBに流通

3つの画面機能とリアルタイム同期機能をパッケージング

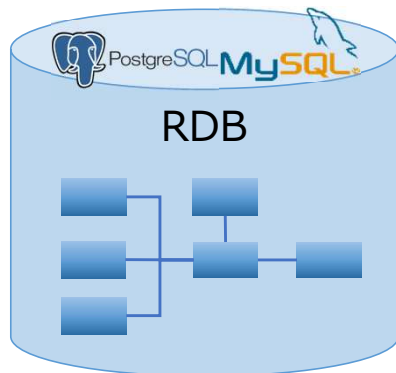


データ同期までの3つのステップを実現するWeb画面

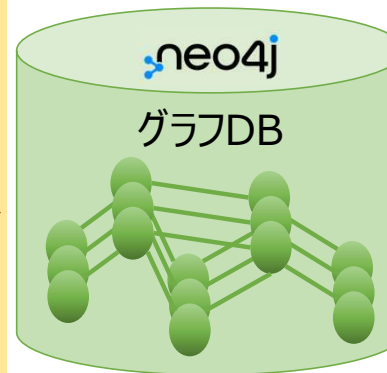
DB接続情報設定

マッピング設定

リアルタイム同期
起動/停止

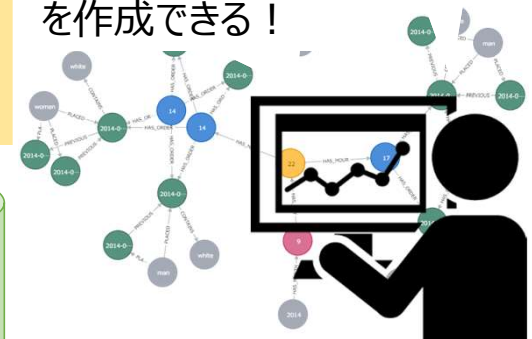


リアルタイム同期



RDBを使いながら
グラフDBも使える！

簡単にグラフデータ
を作成できる！



つながり情報の可視化
データの高速検索
データ活用、分析

【参考】データ同期までの3つのステップ

- ① RDBとグラフDBの接続情報を設定
- ② RDBからグラフDBへ展開したいテーブルやカラム、テーブル間の関係を指定することで、RDBとグラフDBのマッピングを設定
- ③ RDBのデータを即座にグラフDBに反映するリアルタイム同期機能の起動と停止

マッピング一覧 (トップページ)

管理コード	① DB情報設定	② マッピング	③ リアルタイム同期	削除
Location				
PhysicalResource				
test-1				

①接続するDBの情報設定

①DB情報設定画面

RDB情報 NEO4J

PostgreSQL
 MySQL

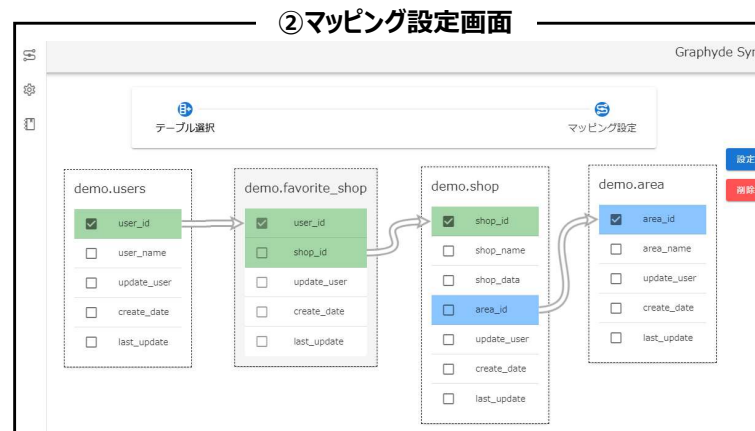
ホスト名
localhost

データベース名
postgres

ポート番号
5432

ユーザー名
postgres

②RDBとグラフDBのマッピング設定



③リアルタイム同期機能の起動

③リアルタイム同期起動/停止画面

リアルタイム同期

起動時に使用するJavaコマンド
/usr/bin/java

スレッド数
1

起動

【参考】Graphyde Syncの動作環境

動作環境

Graphyde Syncは以下の環境で動作することを確認しています。

- Graphyde SyncはLinux(※1)にインストール
- Java(※2)のインストールが必要
- ブラウザは“Chrome”と“FireFox”で動作することを確認

※1:CentOS7、Ubuntu20で動作することを確認しています

※2:Javaのバージョンは8.0以降で動作することを確認しています

対象データベース

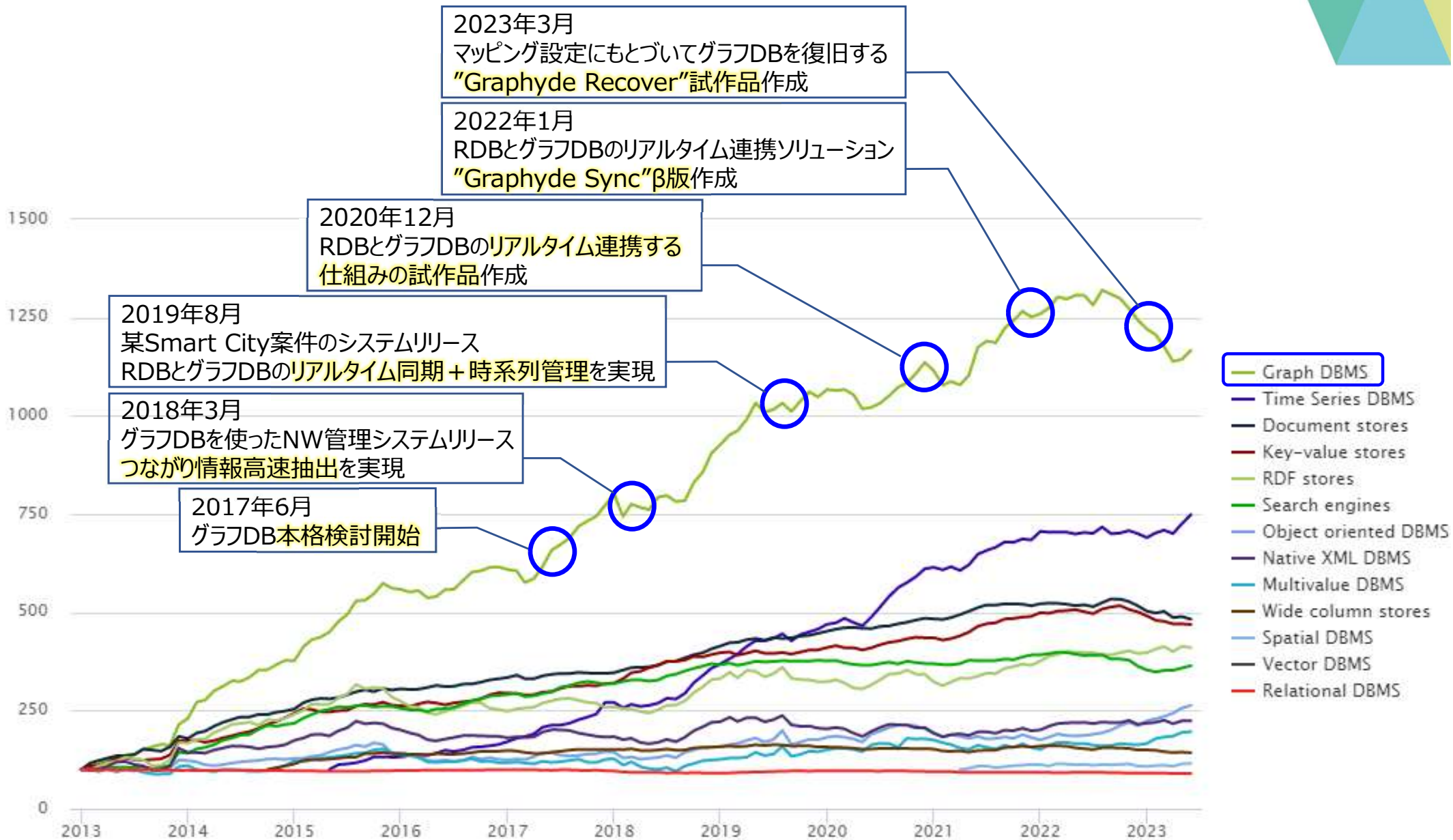
Graphyde Syncが対応するデータベースを以下に示します。

グラフDB	対象のバージョン
Neo4j	4.0以降
RDB ※3	対象のバージョン
PostgreSQL	10以降
MySQL	5.6以降

※3:現在、Oracle、SQL Serverについても動作確認中です

【参考】世界のデータベーストレンドと私たちの歩み

グラフDBは世界で最も注目を集めている



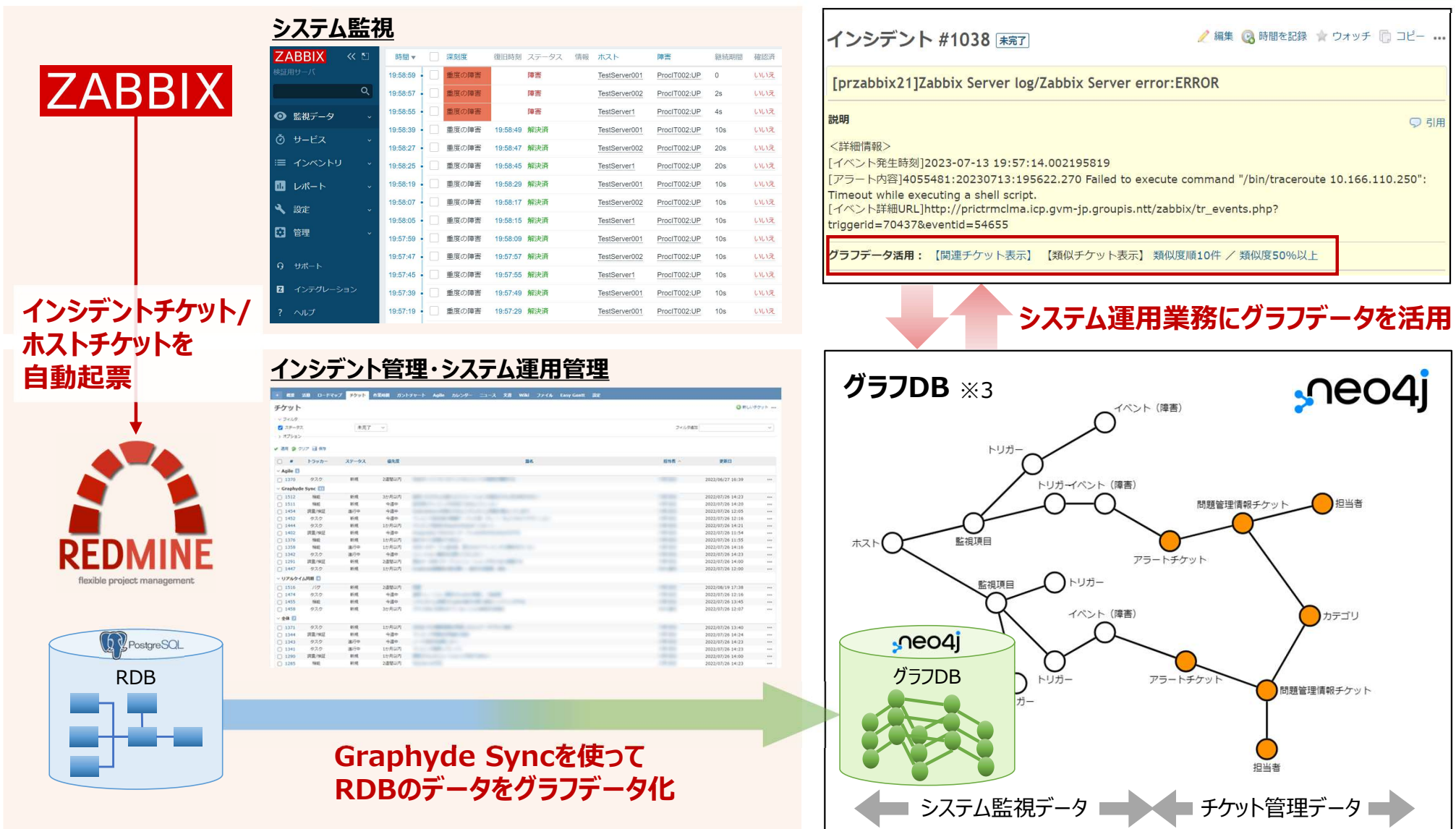
【その3】

Redmineデータのグラフデータ化とその活用

Redmineで管理しているシステム監視・運用業務のチケットデータをグラフデータ化し、そのデータを活用して業務を効率化する方法をデモも交えて紹介

2. Redmineデータのグラフデータ化とその活用

- Zabbixで検知したアラートをインシデントチケットとして、ホストチケットとともに自動起票
- Graphyde SyncによりRedmineのRDBデータをグラフデータ化し、システム運用業務に活用



3. 具体的なアーキテクチャ

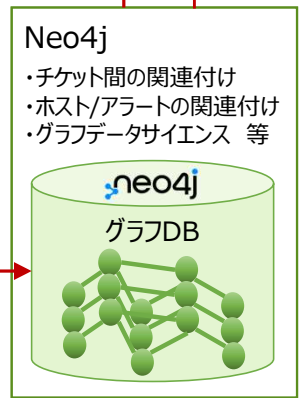
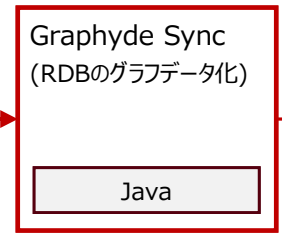
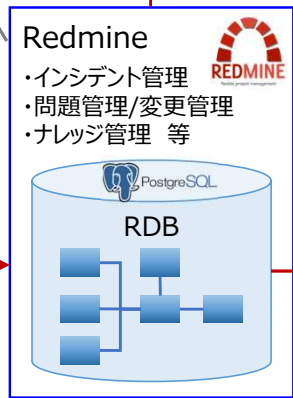
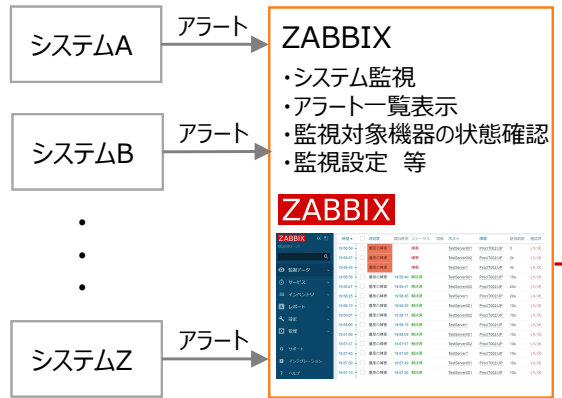
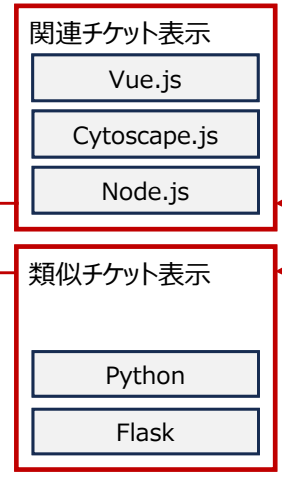
システム運用・保守

```

チケット画面のERBファイル“show.html.erb”に以下を追加
(/app/views/issues/show.html.erb)

<hr />
<b>グラフデータ活用:</b>
<%= link_to “【関連チケット表示】”, “http://xxx.xxx.xx.xx/redmine/issues/#{h @issue.id}”, target:
“_blank”.html_safe %>
【類似チケット表示】
<%= link_to “類似度順10件”, “http://xxx.xxx.xx.xx:5000/similar_issues/#{h @issue.id}”, target:
“_blank”.html_safe %>
</>
<%= link_to “類似度50%以上”, “http://xxx.xxx.xx.xx:5050/similar_issues_50/#{h @issue.id}”, target:
“_blank”.html_safe %>

```



RedmineのAPIを使って以下を実施

- ・インシデントチケットの自動起票
- ・ホストチケットの検索、存在しなければ自動起票
- ・インシデントチケットとホストチケットを自動で関連付け

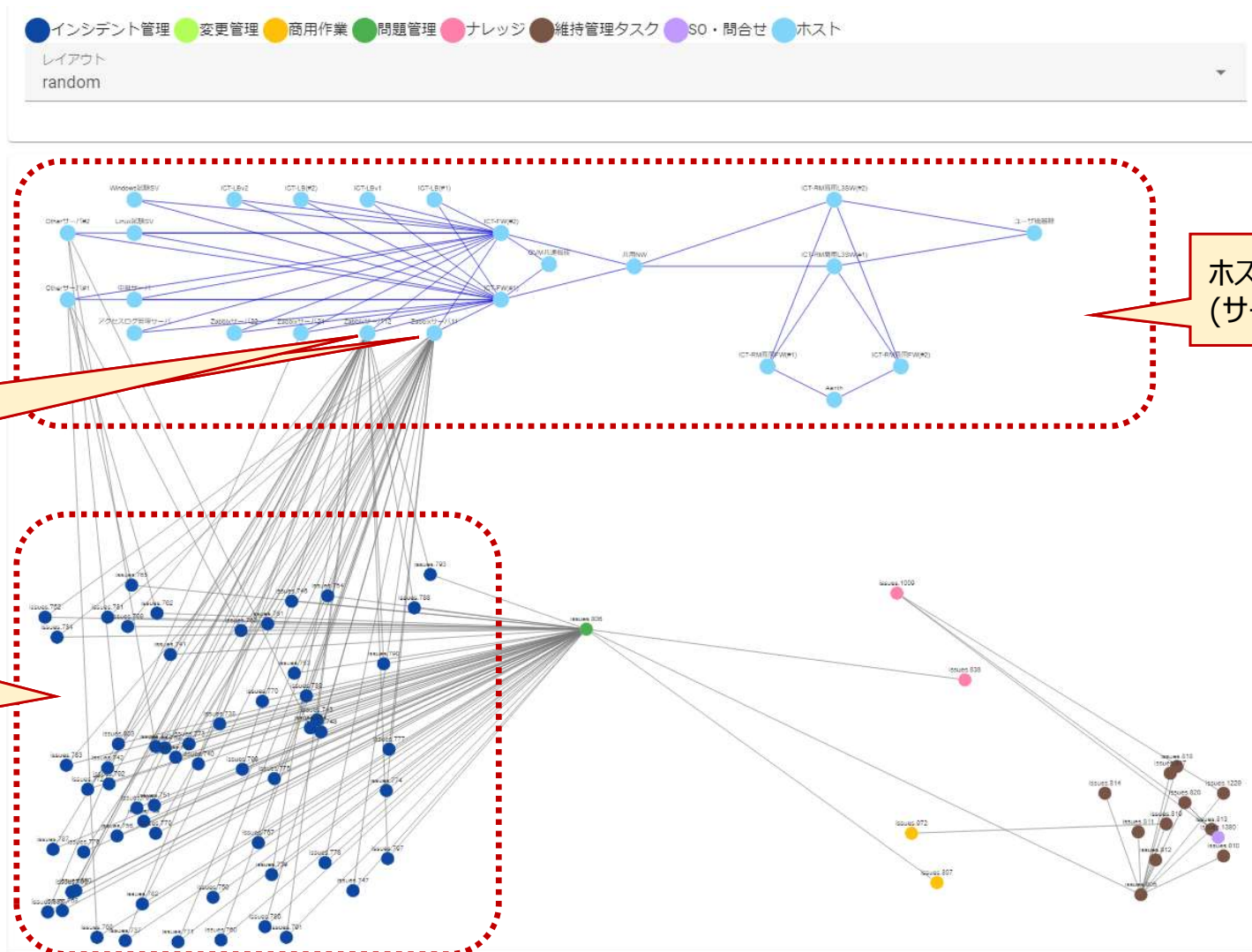
RDBとグラフDBのハイブリッドDBシステム

: 私たちが独自に開発した部分

4. 関連チケットの可視化（つながり情報可視化）

- アラートにより起票されたインシデントチケットとホストチケット、関連チケットをグラフDBで紐づけ
- それにより障害が発生時、どこのホストで発生したアラートか、アラートが集中している個所はどこか、視覚的かつ即座に判別が可能

関連チケット表示



このサーバにアラートが集中していることがわかる

インシデントチケット (発生アラート)

ホストチケット (サーバ・NW機器)

5. 類似チケットのレコメンド

- インシデントの対応時、**グラフデータサイエンス**の活用により過去の類似インシデントのチケットおよび関連チケットを抽出することで、**迅速な原因解析、適切な対処・対応が可能**

類似チケット表示

Issue ID	Issue Subject	Issue Description	Similarity Rate
1038	[]Zabbix Server log/Zabbix Server error:ERROR	* Zabbix Server log/Zabbix Server error:ERROR 1件 <詳細情報> [イベント発生時刻]2023-07-13 19:57:14.002195819 [アラート内容]4055481:20230713:195622.270 Failed to execute command "\$"/bin/traceroute : Timeout while executing a shell script. [イベント詳細URL]http:// /zabbix/tr_events.php?triggerid=70437&eventid=54655	100.00%

Similar Issue ID	Similar Issue Subject	Similar Issue Description	Similarity Rate
1108	[]Zabbix Server log/Zabbix Server error:ERROR	* Zabbix Server log/Zabbix Server error:ERROR 1件 <詳細情報> [イベント発生時刻]2023-07-31 15:53:14.003173708 [アラート内容]4055746:20230731:155249.491 Failed to execute command "\$"/bin/traceroute : Timeout while executing a shell script. [イベント詳細URL]http:// /zabbix/tr_events.php?triggerid=70437&eventid=199622	91.83%

Associated Issue ID	Associated Issue Subject	Associated Issue Description
1153		<ul style="list-style-type: none"> ● 発生時刻 ● アラート内容 ● イベント発生時刻 ● イベント詳細URL

1110	[]Zabbix Server log/Zabbix Server error:ERROR	* Zabbix Server log/Zabbix Server error:ERROR <詳細情報> [イベント発生時刻]2023-08-01 00:21:14.002837815 [アラート内容]4054634:20230801:002112.192 error reason for "\$local:vmware.hv.cpu.usage.perf [{\$VMWARE.URL}, {\$VMWARE.HV.UUID}]\$ changed: Timeout was reached [イベント詳細URL]http:// /zabbix/tr_events.php?triggerid=70437&eventid=256616	64.99%
----------------------	------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------