

Telemetry-WG 第3回Meeting 情報共有 from ユーザサイド

Telemetry導入における課題

～Prometheusによるモニタリングを導入する中での気づき～

株式会社IDCフロンティア

見崎 徳仁、佐久間 充

2018年9月14日



- IDCフロンティアのモニタリングシステム
- Telemetry導入における課題

IDCフロンティアの モニタリングシステム



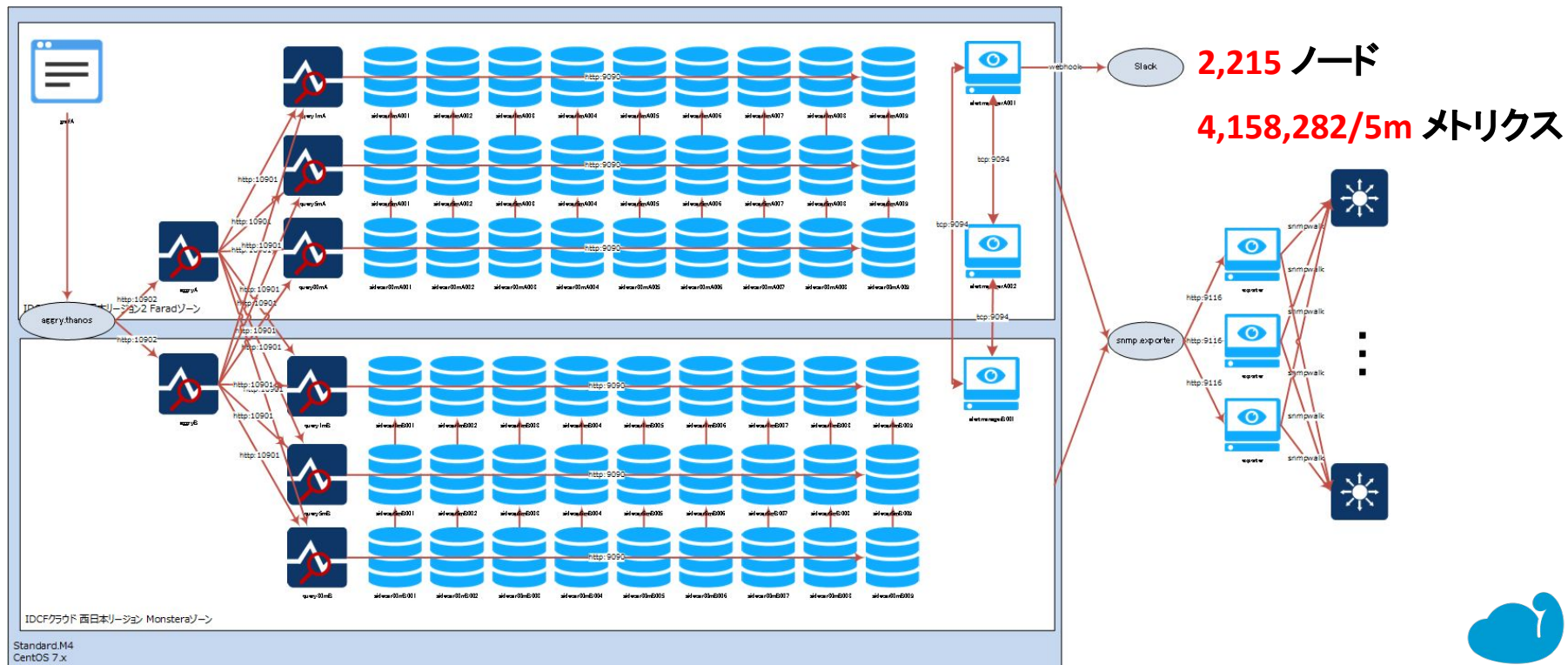
大量のデータを一つのモニタリングシステムで捌きき って難しい...

- 利用者的には、一つのUIで見たいデータを簡単に素早くグラフ表示できてほしい
- 運用者的には、耐障害性に優れてノード登録簡単にできてほしい
- スケールのには、ノード数が1,000台を超えてもサクサクであってほしい
- コスト的には、ハイスpek・大容量システムを低コストで実現してほしい

そんな無茶な、、、

モニタリングシステムを開発している皆さんに頭がさがります

分散とスケールアウトにこだわったシステムを目指した



機能を分散させることでマイクロサービス化っぽくして見た

表示  Grafana, 検索  Thanos, 保存  Prometheus , 取得  Exporter

システム作る際に考慮した点

- ・1(Thanos): 多数(Prometheus)のクラスタ構成で、分散したデータを一元的に可視化できる設計にした
- ・マシンのメモリ、ストレージに応じたメトリクス数と解像度と保持期間のバランスを踏まえて設計を行った

システム作ってよかった点

- ・異なる三つのモニタリングシステムをこれに集約(とりあえずこのGrafana見ることから始められる)
- ・単一障害であれば切り替えも不要
- ・Prometheusのスペック(CPU,メモリ,ストレージ容量・I/O)はそんなに高くなくて大丈夫
- ・Exporter 足すことでトラフィック以外のデータもどんどんモニタリングできる

大量のデータを捌ききるモニタリングシステムはできたが snmpでのデータ取得は限界が来ていることも実感

本当は、データ取得を10秒間隔&システムを3冗長にしたかった

ネットワーク機器は、F数の関係から10秒間隔ではデータ取得が終わるわけではない

3つ同時に来るsnmpポーリングによってCPU使用率が高騰して危険極まりない状態になる機器も出た

泣く泣く断念

機器にデータを取得しに行く、snmpの限界を感じた瞬間でもある※多分プロトコル的な特性もあるんだと思う

そして、**機器からデータを送付してくる**Telemetryに期待した瞬間でもある

(Telemetryだと、変更があったデータのみ出すとかの定義をいい感じに機器側で出来る？

なら機器の負荷を上げずに短間隔で欲しいデータのみを送付可能？)

Telemetry導入における課題



データがなかなかエグい量になる

$1,000_{[台]} \times \underline{54_{[ポート]}} \times 2_{[方向]} \times 6_{[メトリクス]} \times 60_{[サンプル/分]} \times 60_{[分]} \times 24_{[時間]} \times 12_{[bytes]}$

= 約672[GBytes]

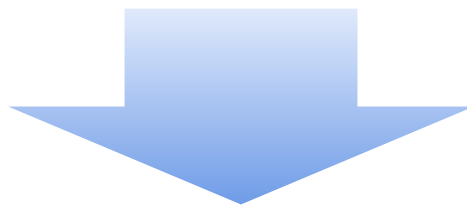
- In/Outのメトリクスを1つ追加すると+約112[GByte]

[計算の前提]

- ToRスイッチ1,000台からデータを収集
- ToRスイッチはよくある48SFP&6QSFP(54ポート)
- 取得メトリクスはOctet, Unicast Pkt, Multicast Pkt, Broadcast Pkt, Discards, Errorsとする
- 1秒間隔でデータを収集し、24時間分貯める
- 1サンプルあたり12バイトとする ※OpenTSDBの例



**Telemetryを利用するためには
多かれ少なかれシステムの更新は必須**



何をする？どこまでやる？

Telemetryを使う目的を決める

- 目的によって必要なシステム構成は変わる
 - 高解像度なデータを活用したい
 - 高解像度データの収集・保管・評価を行うシステムが必要
 - そこから何ができるか？費用対効果は？
 - SNMPを置き換えるだけ
 - 高解像度なデータは不要だがSNMPは嫌という人
 - コレクタの置換に留め投資を抑えるという選択肢も

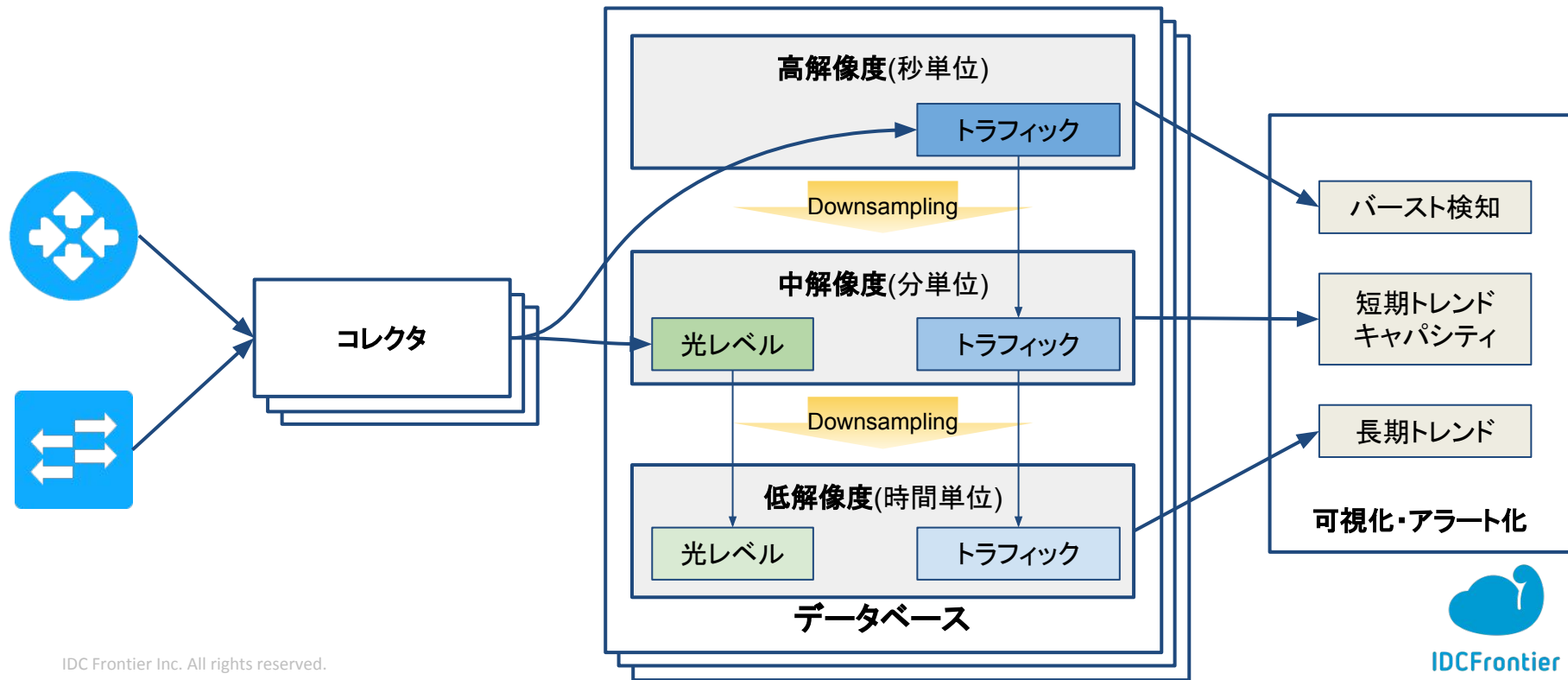
データ収集、データ保持の戦略

- 利用目的に応じてデータの収集方法、保持期間を変える
 - 利用目的が決まれば適切な解像度が決まる
 - 高解像度が必要なメトリクス、不要なメトリクスという割り切りもできる
- Downsampling Strategy
 - 「短期間用は高解像度、長期間用は低解像度」
 - 秒単位の解像度でデータを長期間保持しようとするシステムのリソースがいくらあっても足りない
 - 「短期間」「高解像度」をどう設計するか？がキモ

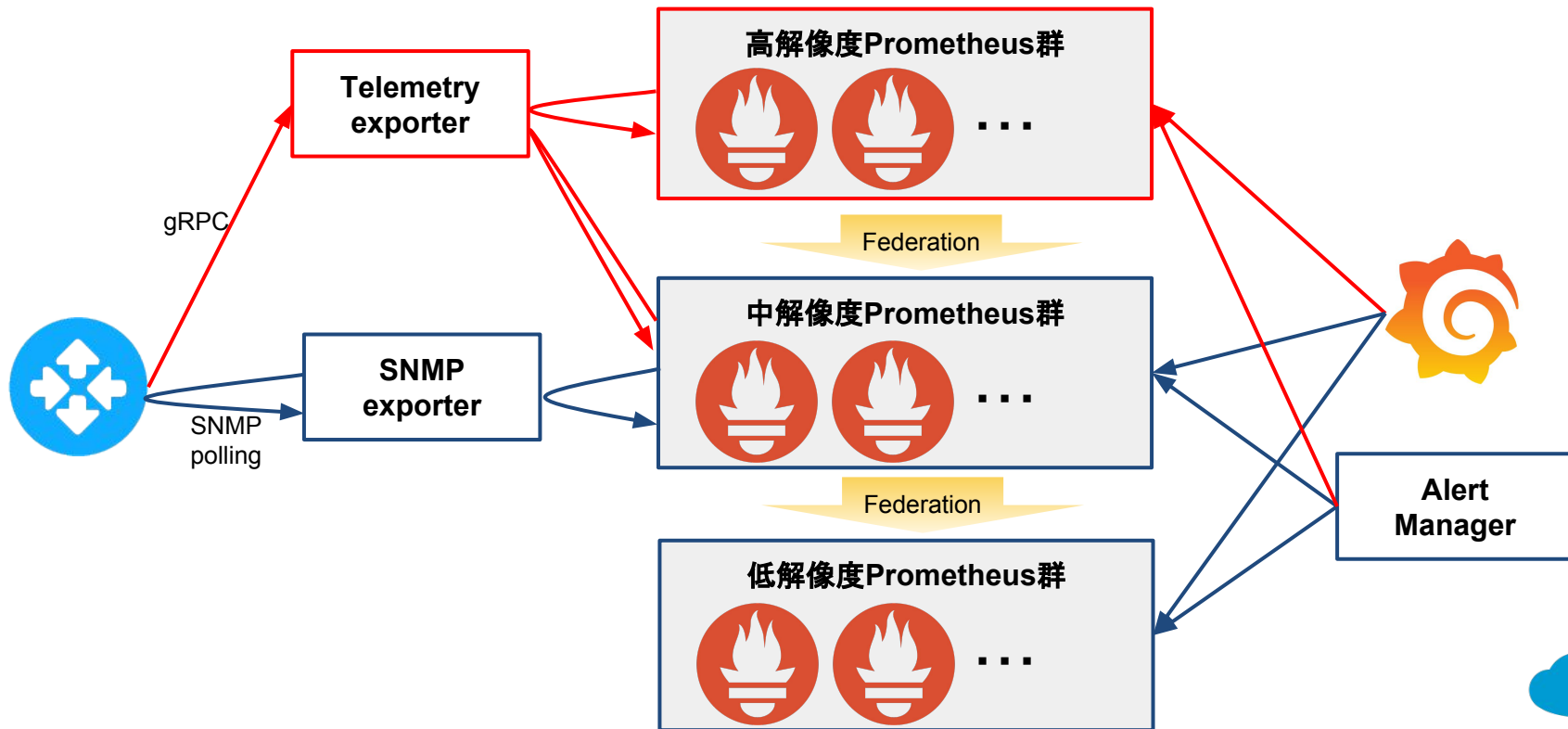
モニタリングシステムのスケーラビリティ

- ネットワーク機器はインターフェースに紐づくメトリクスが多い
 - 取得メトリクスが1つ増えただけでも影響が大きい
 - 高解像度なデータであればなおさら
- 最初は不要としていたデータでも突然必要になるケースは往々にして存在する
 - 後から慌てないよう、モニタリングシステムもスケールアウトできる構成を考えておくことが重要

必要なデータを必要な解像度で利用する



Prometheusでやろうとするとこんな感じ？ ※想像



Telemetryの導入を検討する際の課題

- Telemetryを使う目的を明確にしよう
- 大量のデータを扱うための仕組みを整えよう
 - データ収集、データ保持の戦略を考える
 - モニタリングシステムのスケーラビリティを考える

