# 自己紹介

名　　前　■　木村 裕（きむら　ゆう）
所　　属　■　株式会社QTnet 技術部 監視システムグループ

**【経歴】**

・ 1997年　九州通信ネットワーク株式会社(現：株式会社QTnet)入社
  - 電話交換サービスの保守業務
  - ISP事業用設備構築業務
  - ISPサービス設備の保守・運用業務に従事
・ 2014年から現在の技術部に所属

**【Zabbixとの関わり】**

（省略：Zabbixは1.4の頃から利用）

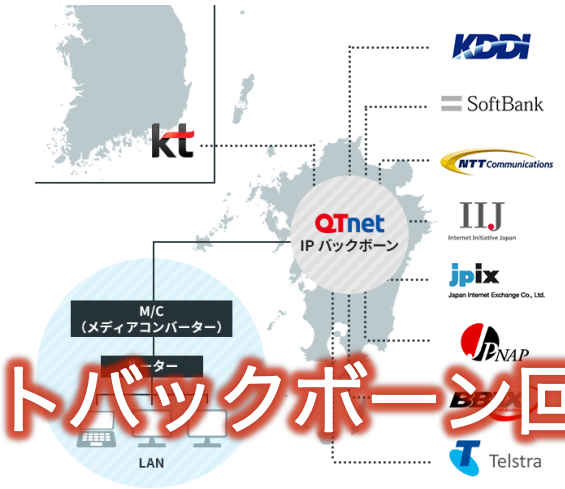
2015年　ZabbixカンファレンスJapanのライトニングトークで発表
2016年　ZabbixカンファレンスJapanで発表
　　Zabbix3.0 スペシャリスト・プロフェッショナル取得
2017年　休憩（カンファレンス後の懇親会ビンゴで景品を初ゲット）
2018年　Zabbixサミットで発表しようとするが、諸々の事情で断念

# 会社紹介

| | | |
|---|---|---|
| 社名 | ■ | 株式会社QTnet（平成29年7月に社名変更、旧社名は九州通信ネットワーク株式会社） |
| 設立 | ■ | 昭和62年7月1日（事業会社への移行は平成元年11月1日） |
| 所在地 | ■ | 福岡県福岡市 |
| 株主 | ■ | 九州電力株式会社 |
| 主な事業 | ■ | 電気通信事業、一般放送事業、電力の購入・販売 |

総延長約10万kmの
光ファイバ網（九州内）

24時間365日の
技術エキスパートによる運用体制




充実のインターネットバックボーン回線

# アジェンダ

機械学習による異常の予測検知について
～素人が機械学習をやってみた～

1. Zabbix監視内容紹介
2. 機械学習をやってみた
3. 分析のやり方
4. 今後のお話

# 1.Zabbix監視内容紹介

2. 機械学習をやってみた

3. 分析のやり方

4. 今後のお話

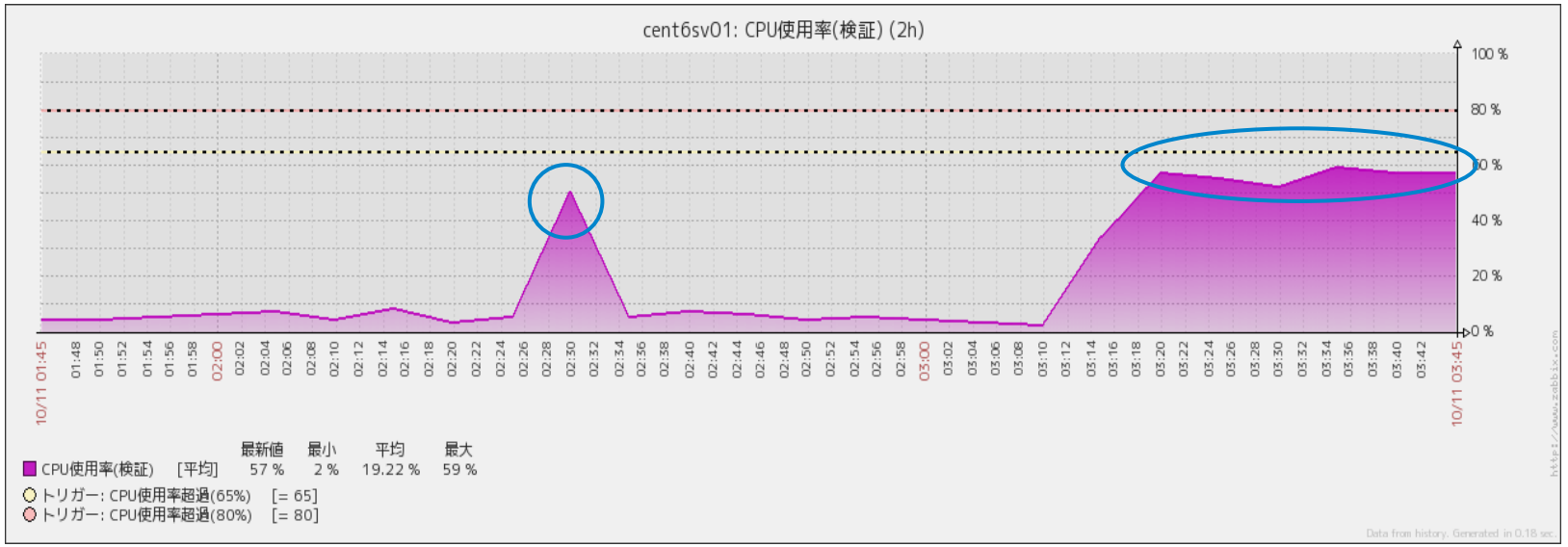
# Zabbixでの監視内容紹介

- 従来はSNMP Trapによる監視のみ行っていたが、より詳細な監視を行うよう、Zabbixにてリソース監視等も行っている。
- サービス別にZabbixサーバーを構築し、監視を行っている。
- ZabbixサーバーはActive-Activeのデュアル構成で冗長構成を行っている。
  - サービス A (Zabbixサーバー1 & Zabbixサーバー2)
  - サービス B (Zabbixサーバー3 & Zabbixサーバー4)

| | サービス A | | サービス B | |
|---|---|---|---|---|
| | Zabbix サーバー1 | Zabbix サーバー2 | Zabbix サーバー3 | Zabbix サーバー4 |
| ホスト数 | 214 | 214 | 378 | 378 |
| アイテム数 | 25,897 | 25,897 | 106,054 | 106,054 |
| トリガー数 | 3,686 | 3,686 | 14,647 | 14,647 |
| 1秒あたりの監視項目数 | 167.3 | 167.3 | 354.98 | 354.98 |

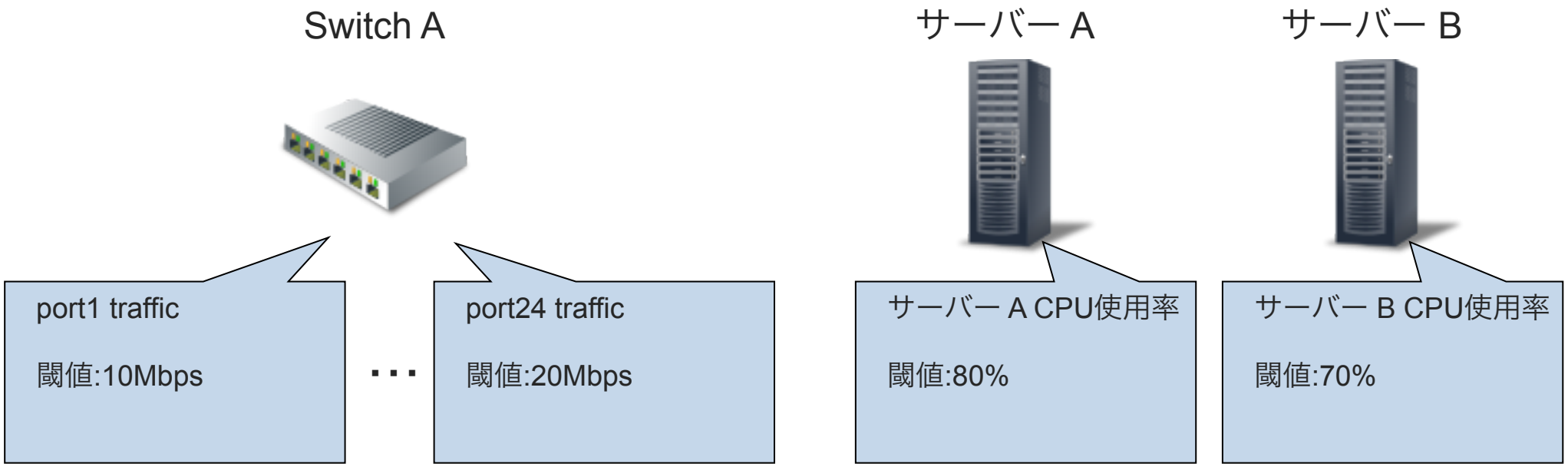
# 監視設定の問題点

- **閾値を設定しても、閾値内の通常と異なるデータの動きを検知できない。**
  - 検知するにはトレンドを都度確認したり、何度もフィッティングをかける必要がある。



# 監視設定の問題点

- ZabbixのテンプレートやDiscovery機能で容易に監視・閾値設定はできるが、**ネットワーク機器の各ポート単位のトラフィック閾値等を設定するには手間がかかる。**



# 問題点を解決するには

**問題点**

閾値を設定しても、閾値内の通常と異なるデータの動きを検知できない。

**解決**

Zabbixで収集したデータを用いて機械学習を行い、データの値を予測分析する。

**問題点**

ネットワーク機器の各ポート単位のトラフィック閾値等を設定するには手間がかかる。

**解決**

予測分析した値を逸脱したものを異常検知することで、閾値を設定することなく監視を行う。
（いつもと違う動きを捉える）

1. Zabbix監視内容紹介

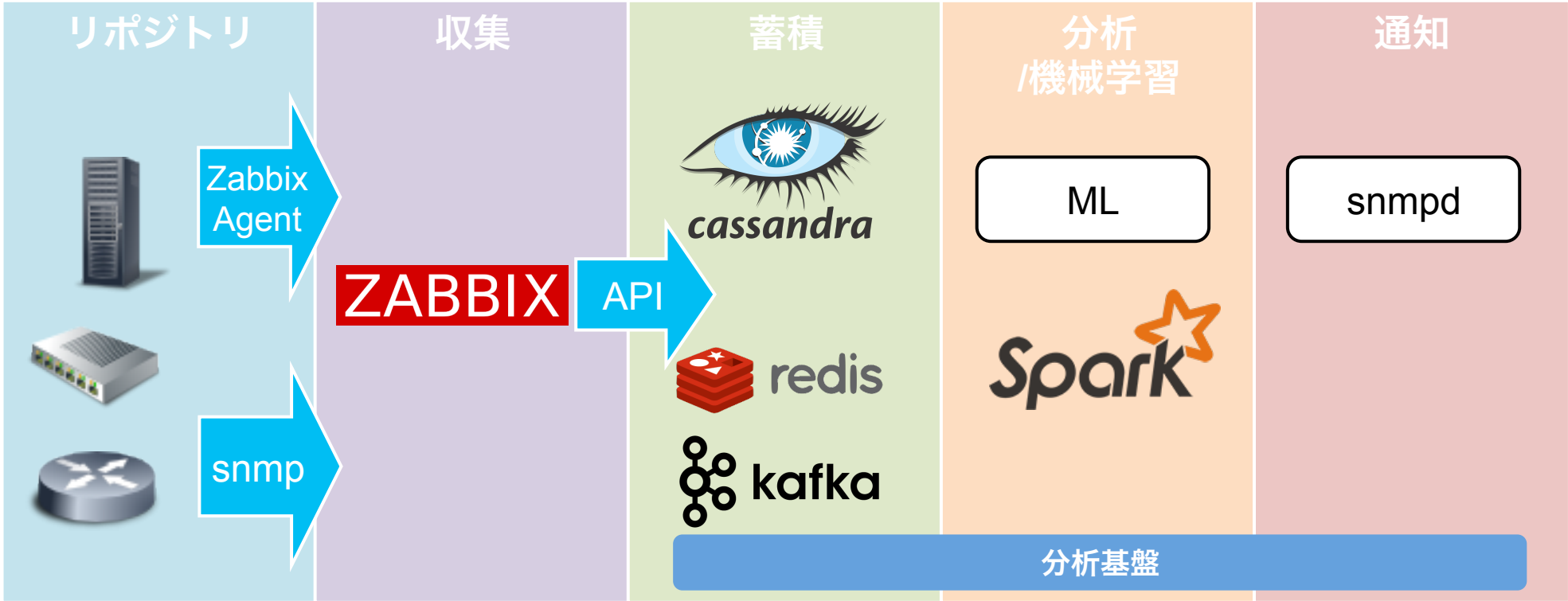
# 2.機械学習をやってみた

3. 分析のやり方

4. 今後のお話

# 機械学習・分析基盤構成

- ネットワーク、サーバー機器のデータをZabbixにて収集。
- Zabbixで収集したデータをAPIを利用して分析基盤のcassandraに蓄積。
- 蓄積したデータを機械学習、分析。
- 分析結果から異常をSNMP Trap通知。



# 学習方法

- 通常時の「正常なデータ」から逸脱した「外れ値」を異常として捉える「教師なし学習」を実施。
  - ✓ 検知定義（シグネチャ）を考える必要がないので、「何か異常がおきているっぽい」という判断の実装はすぐに可能。
    →サイレント障害や、過去に発生した事がない事例にも対応できる。
  - ✓ 教師データを保存しておく必要がないので、ストレージ容量的には優しい。
  - ✓ 教師データとなる、過去の障害時のデータ（ナレッジ）が少なかった。。。

- ただし、信憑性に若干欠ける。
  - ✓ 教師データからの検知定義であれば過去の類似障害として判断できる。（正当性が高い）
  - ✓ なぜ異常検知したのかを後々確認する必要がある。
    →明らかに異常なデータ値を示していれば異常として認識できるが、判断に困る場合もある。

# 学習データの最新化

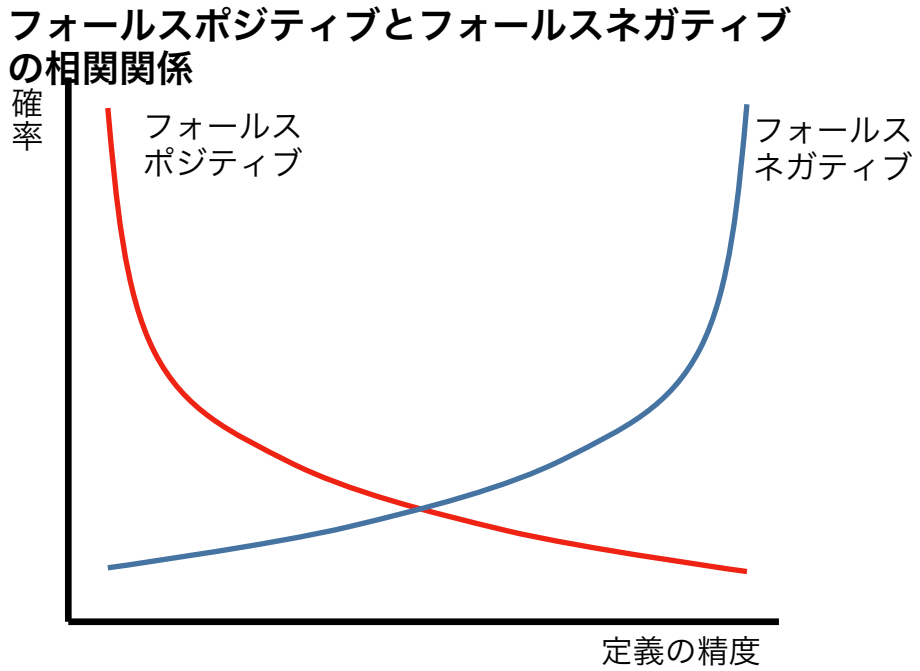
- トラフィック等、データが徐々に増減してカレント値と学習データの差が広がっていき、検知頻度が多くなる。
（誤検知の増加）

- そのため、学習データを再シュミレーションし、現在のトレンドデータに沿った形で分析できるようにする。
（学習データの最新化）

- トレンドが変わったかどうかは異常検知した数とその連続性で自動判断。
※決め打ちで、**1日5回以上、2日連続**で検知したらトレンド変更と判断。
→ただし、日またがりの場合は、トレンド変更という判断をしてしまう。

# 誤検知と見逃し

- フォールスポジティブ（False Positive）
  →**「正常」**を**「異常」**として検知してしまう。（誤検知）
- フォールスネガティブ（False Negative）
  →**「異常」**を**「正常」**として見逃してしまう。（見逃し）




FPの確率を下げるために定義の精度を上げると、FNの確率が上がる。

→妥協するポイントを設定することが大事。

→運用では**「ある程度の誤検知はしょうがないよ」**という大らかな気持ちが大事。

（ポジティブ）

→現場の心の声：余計な検知しやがって（ネガティブ）

1. Zabbix監視内容紹介

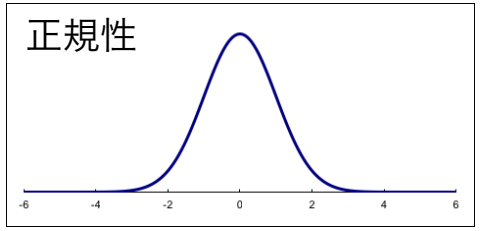
2. 機械学習をやってみた

# 3.分析のやり方

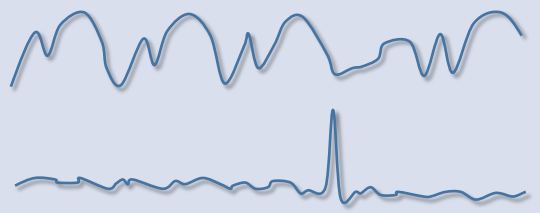
4. 今後のお話

# 分析モデルの作成方法

- 分析はデータの特性に適したアルゴリズムを利用する必要がある。
  - ✓ 周期性があるか
  - ✓ 正規性があるか

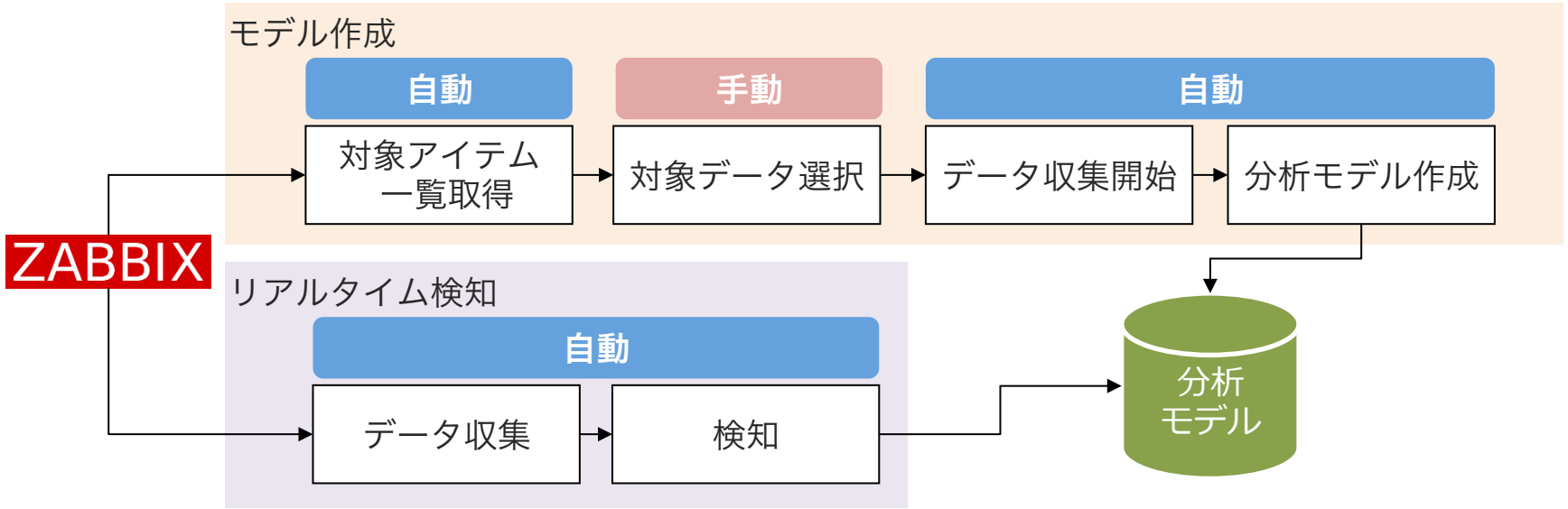

- 分析には以下のアルゴリズムを利用。
  - ✓ SparseCoding
  - ✓ ARIMA

| アルゴリズム | 適用シーン | データのイメージ |
|---|---|---|
| SparseCoding | • 周期性の強いネットワークトラフィックの監視<br>• ネットワークトラフィックにおける急増急減の監視 |  |
| ARIMA | ディスク使用量、メモリ使用量の上限到達予測 |  |

# 分析モデルの作成方法

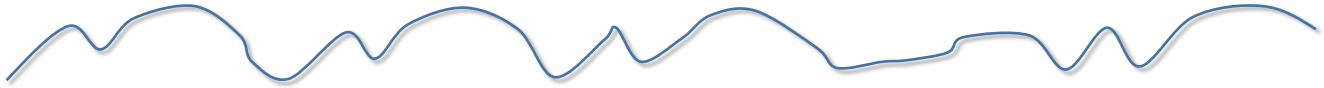
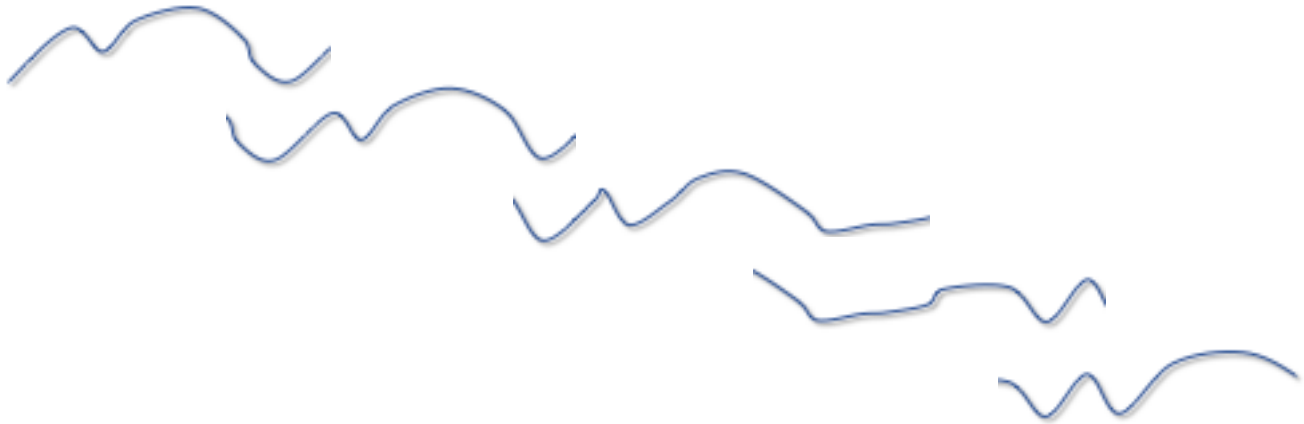
- Zabbixアイテムのキー名で、適用するアルゴリズムを決定。
  - ✓ トラフィック(net.if.in、 net.if.out)
    →SparseCoding
  - ✓ CPU・メモリ・Disk使用率(system.cpu.util、vm.memory.size、vfs.fs.size)
    →ARIMA（ただし、周期性が高ければ、SparseCodingを利用することもある）

- データ収集から分析モデルの作成、リアルタイム検知までを自動化。



# SparseCodingの特徴

- 過去のデータのパターンを辞書化し、現在のデータを同辞書で復元、類似パターンと比較することで「異常」を検知する手法。

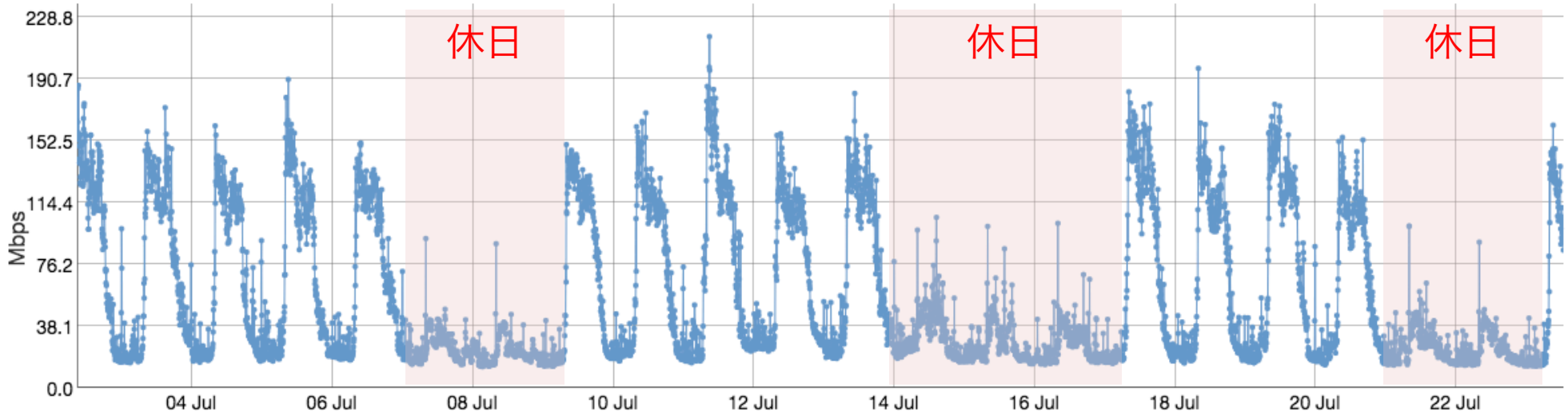
| | |
|---|---|
| 元データ |  |
| ランダムウィンドウで複数<br>パターン化<br>（辞書化） |  |

- SparseCodingは過去のパターンを見ているが「時系列」の情報を見ていない。

- 動きの少ないデータは予測出来ない。

# SparseCodingを選んだ理由

- トラフィックは平日と休日で時間によるデータのパターンが異なる場合がある。



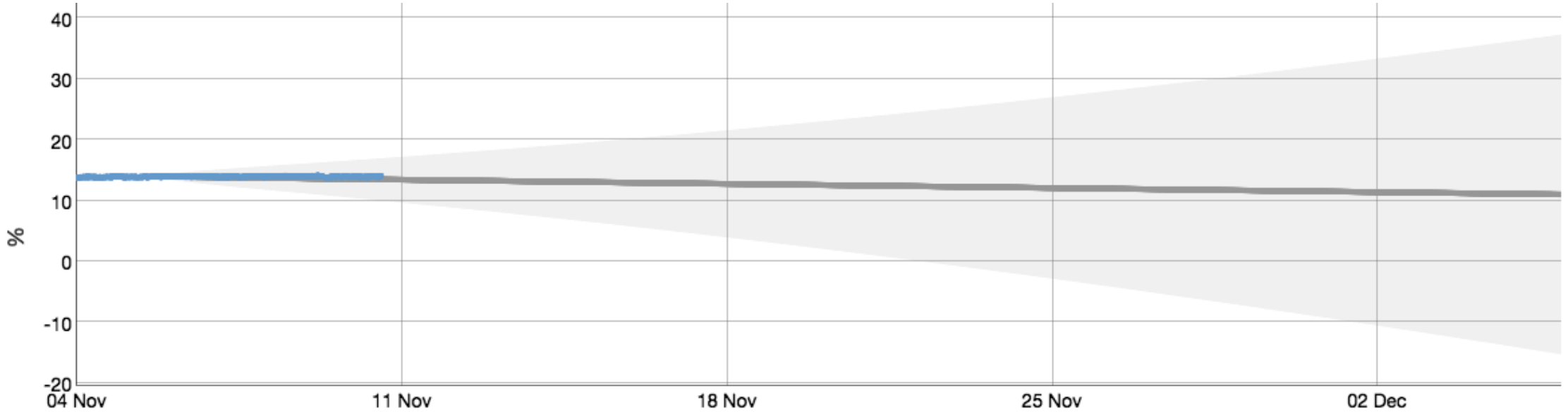

- 平日のデータと休日のデータを合わせて時系列分析を行った場合、誤った予測値を返す場合があり、誤検知が多く発生した。
- SparseCodingでは値を予測してからの異常検知ではなく、形(パターン)にマッチしないものを検知するため、平日と休日のデータパターンを学習させておくことで、正しい分析を実行できた。

# ARIMAの特徴

- 過去の時点との相関を算出し、現在に及ぼす影響度をパラメータ化、95%で取り得る値の範囲を外れることで「異常」を検知する手法。

- ARIMAは「X時間前」との関係性で現在値を評価する。

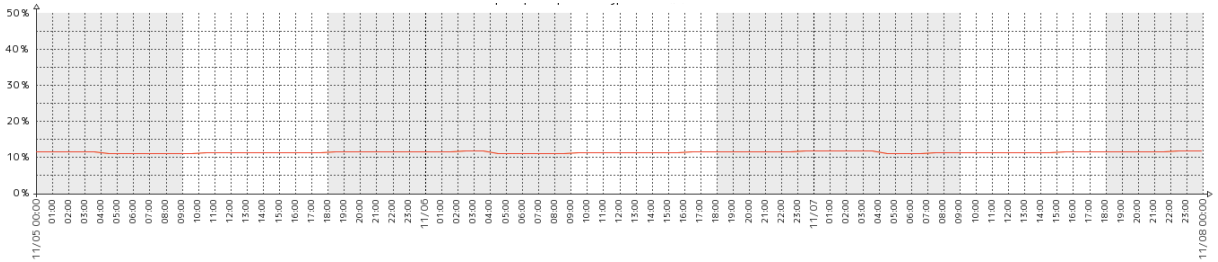
- Zabbix3.0でも類似の分析？



# ARIMAを選んだ理由

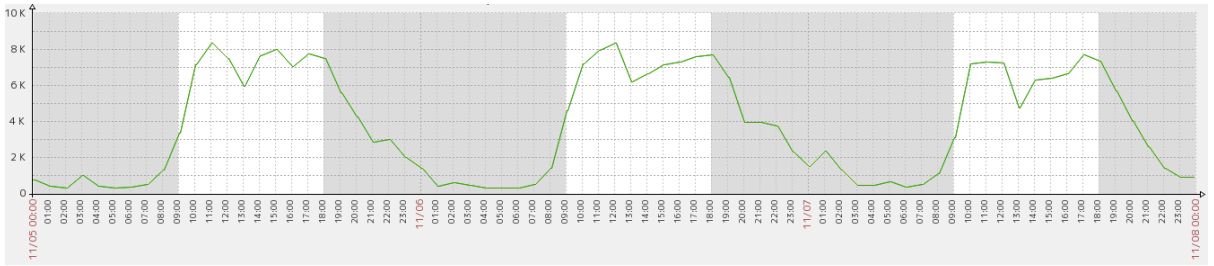
- データの変動が少ないもの(形のパターンが一定)はSparseCodingでは分析に適しておらず、異常検知できなかった。

**■ARIMAの分析に適したデータ**



絶対値の変動が少ない（ほぼ一定）データは、ARIMAでの分析に適しているデータの形式。

**■SparseCodingの分析に適したデータ**



データの変動が大きい（周期性がある）データはSparseCodingでの分析に適しているデータの形式。

- そのため、データの変動が少ないものについては、ARIMAによる分析（時系列）を行い、現在値の予測を行うことで、異常として捉えることができた。

# 機械学習による異常の予測検知をやってみて

- 結果が出なくても**くじけない心**が大事。
  - ✓ 検証で上手くいっても、本番でダメとかいう結果がしばしば。。。
- 誤検知、見逃しが多くて周囲からダメ出しを受けても、「100%正解なら気象予報士とかいらない」という**悟りを開く**こと。
- 機械学習で楽するつもりが、**実際は稼働が増えていることに気づかないようにする。**
  - ✓ 実際はトレンドデータとにらめっこ。。。
  - ✓ 何だかんだでPDCAサイクルで分析を回す必要がある。
- なんとなくでもアナリストになれた気になる。
- **Zabbixで標準機能として実装を熱烈に希望。**

1. Zabbix監視内容紹介

2. 機械学習をやってみた

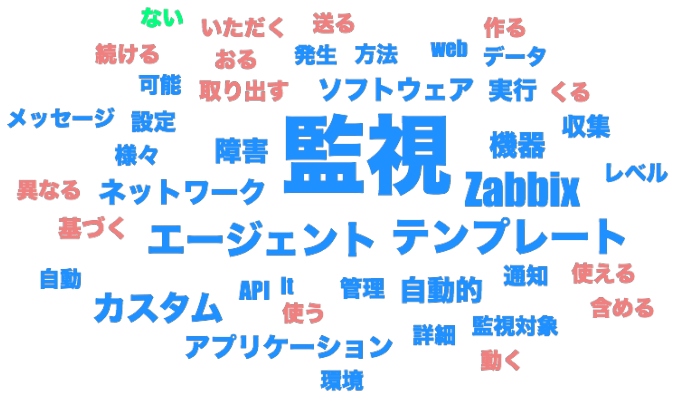
3. 分析のやり方

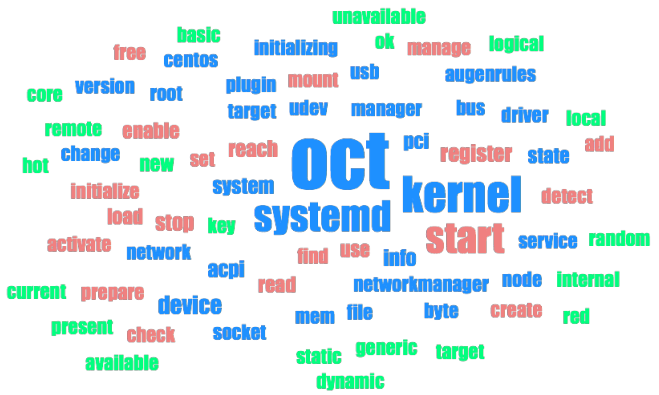
# 4.今後のお話

# 今回できたこと、今後やってみること

- 数値型の分析
  - ✓ 数学的な統計分析による異常検知

- Zabbixで各ログデータも収集できるため、今後はテキスト型の分析を検討中
  - ✓ システムログ、アプリケーションログなどの文字列の特徴を分析評価
    - ◆ 何らかの文字列（FAIL等）が増加傾向に無いか？（ワードクラウド分析）




https://textmining.userlocal.jp

  - ◆ 本来出力されるべきログが出力されているか⇔通常出力されないログが出力されていないか
  - ◆ Zabbix3.4.5からElasticSearchと連携できるので期待。

# ご清聴ありがとうございました