IIJ における異常検知の取り組み

IIJ Internet Initiative Japan

2010/12/13　技術交換会

IIJ アプリケーションサービス部
岩永　義弘

Ongoing Innovation

はじめに

テーマ：「異常」を検知する

IIJ における取組み

- Anode
- リソースの連動性異常検知（鋭意開発中！）

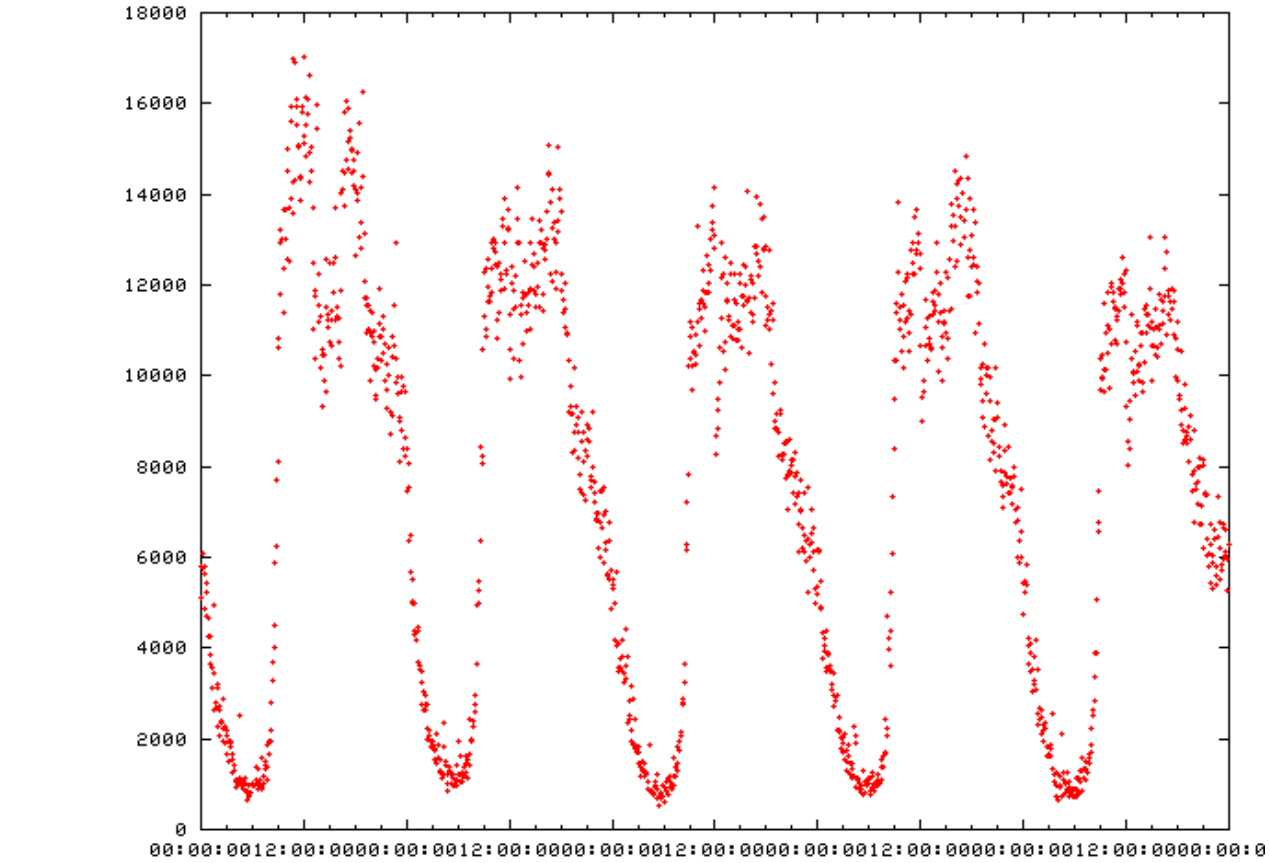
どちらも処理の大枠は同じ

① 平常時の振舞いを学習
② 新しく取ったデータが「平常時と同じかどうか」

Anode のコンセプト

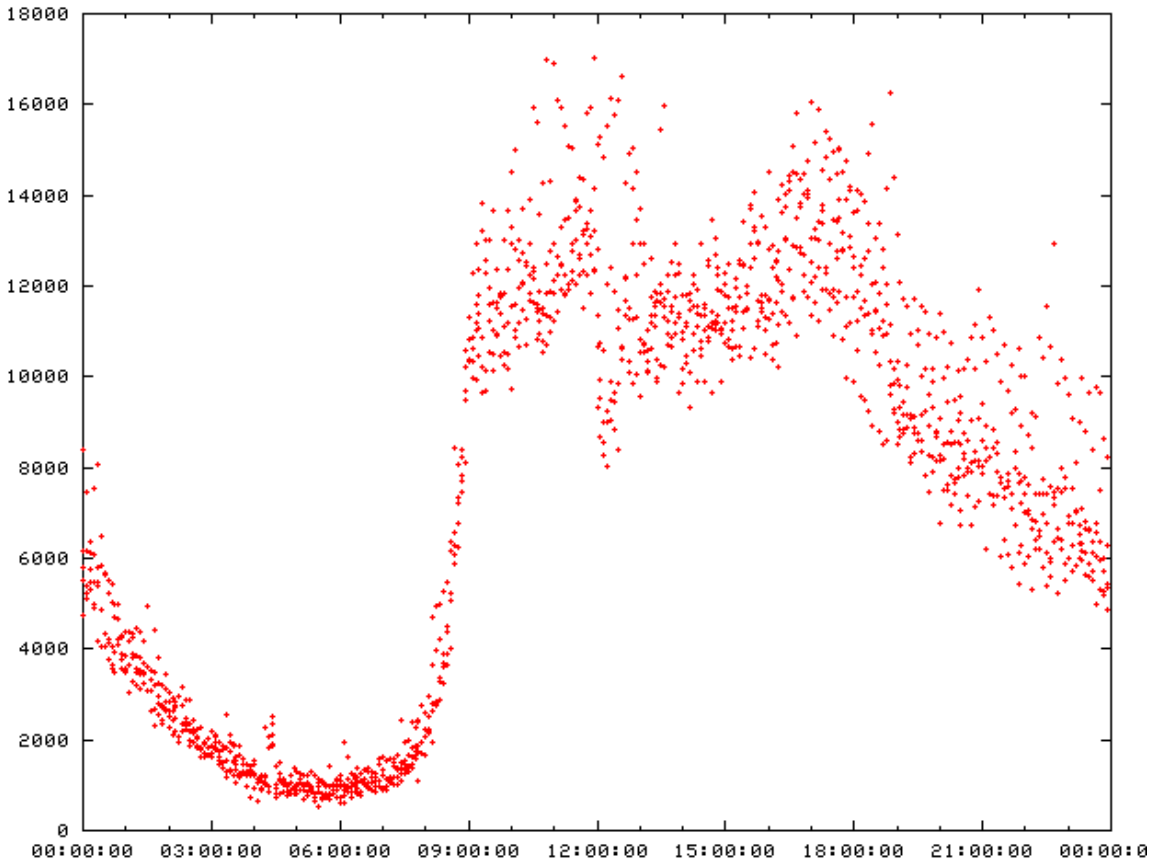
とある顧客の５日間（平日）

FW コネクション数



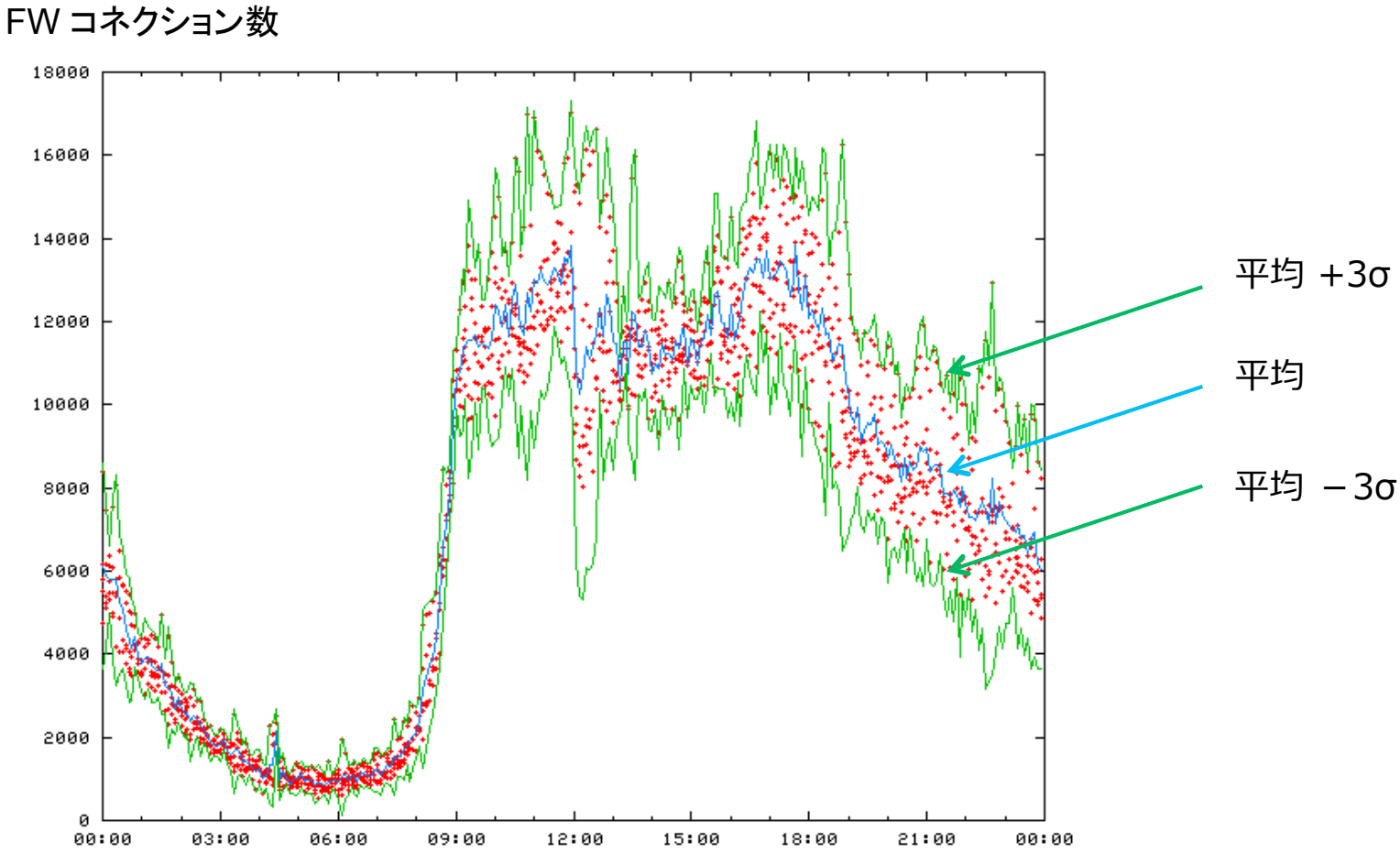
Anode のコンセプト

FW コネクション数



２４時間で重ねてプロット
>>同時刻の値はだいたい同じ

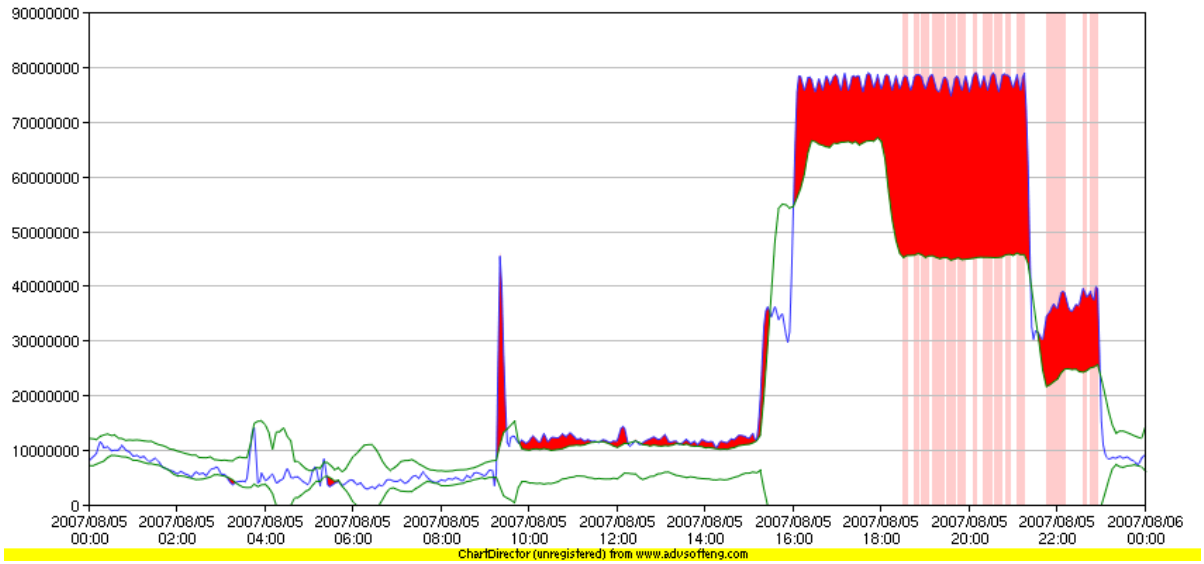
Anode のコンセプト



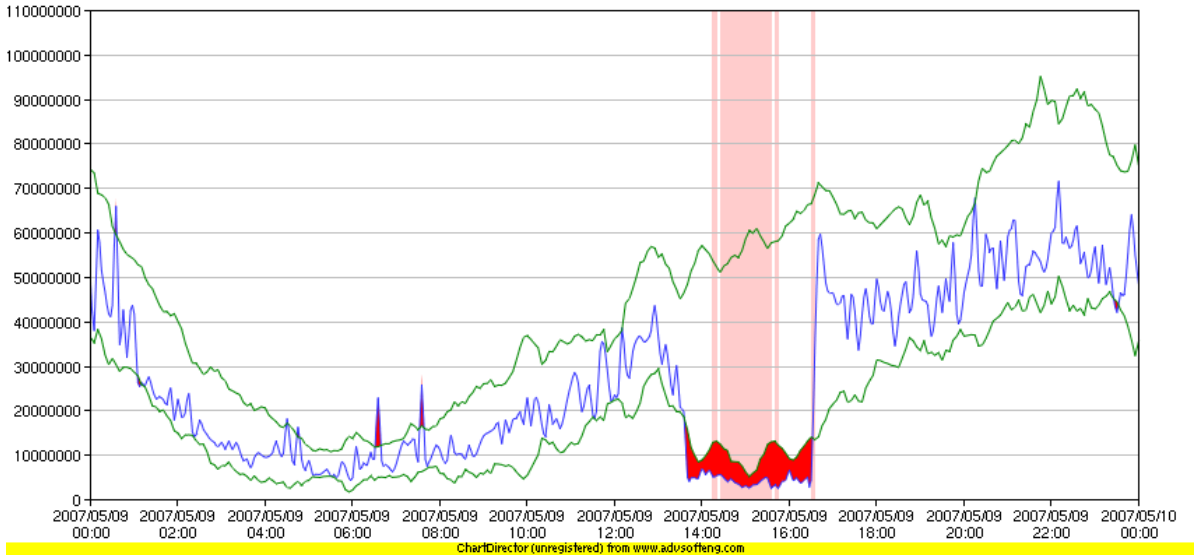
- 99.74%が (平均)±(標準偏差の３倍) の範囲に入る
- 指定した範囲より外側だったら、異常

MFW 検知事例

DoS



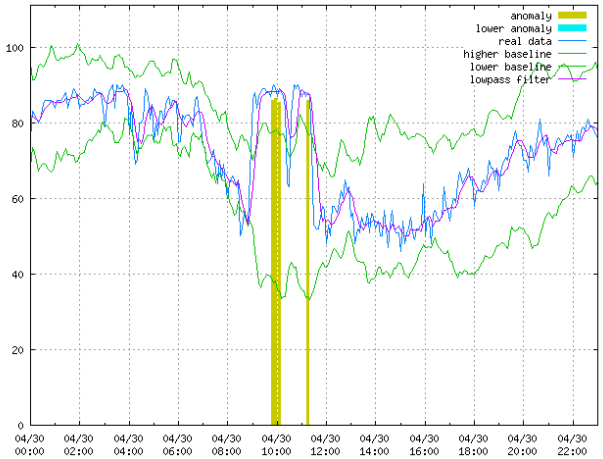
機器障害



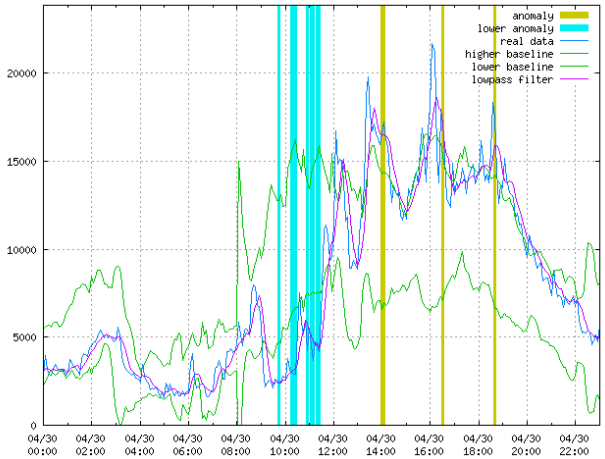
メール 検知事例

誤った SPAM 判定ルールの適応

SPAM 率

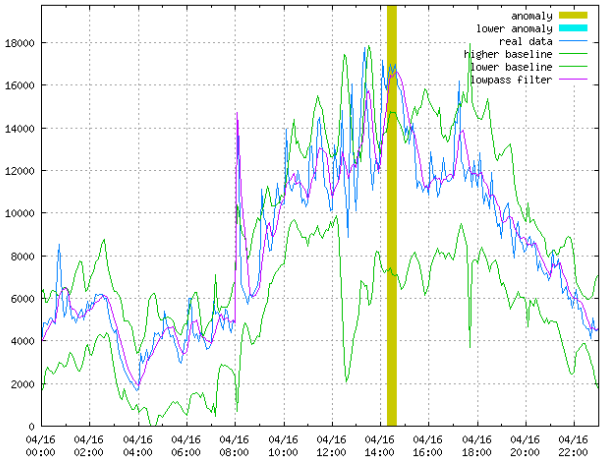


フィルター通過後のメール通数



メールのループ

フィルター通過後のメール通数



Anode まとめ

- いつもより多い・少ないを自動で判定
  - 周期性のあるデータに効果を発揮
- 現在実運用されているサービス
  - ファイアウォールサービス
  - メールサービス

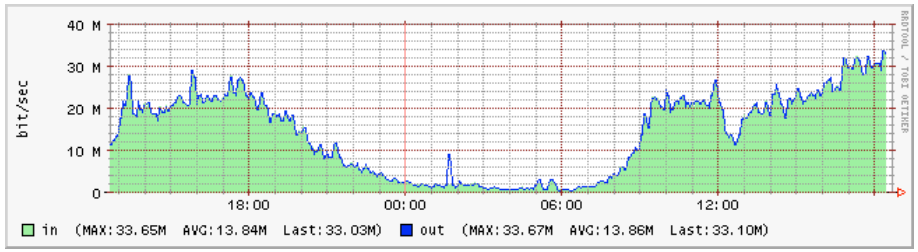
相関異常検知のコンセプト

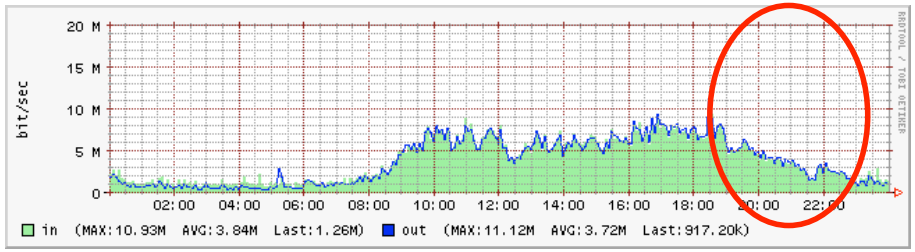
- 連動して機能するリソース同士には相関がある

相関がある状態

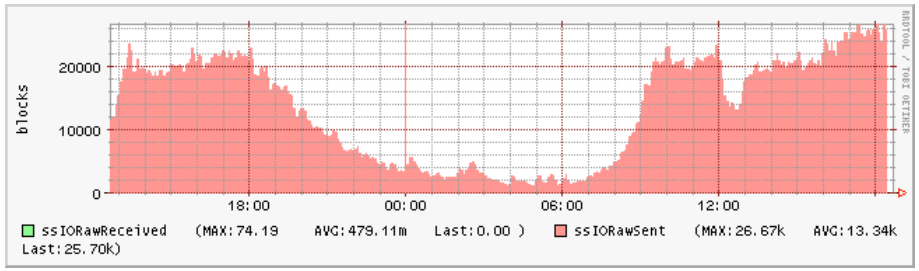
相関が崩れた状態

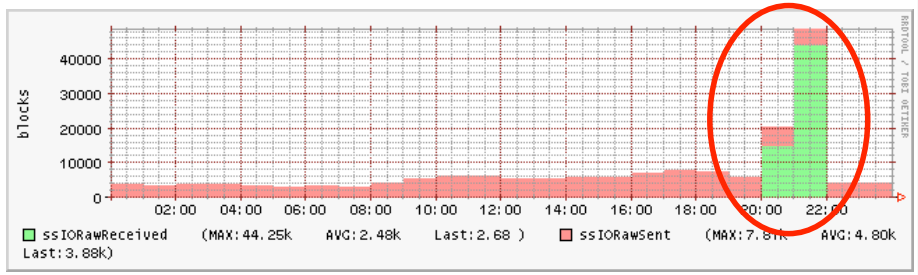
前段ホストのトラフィック





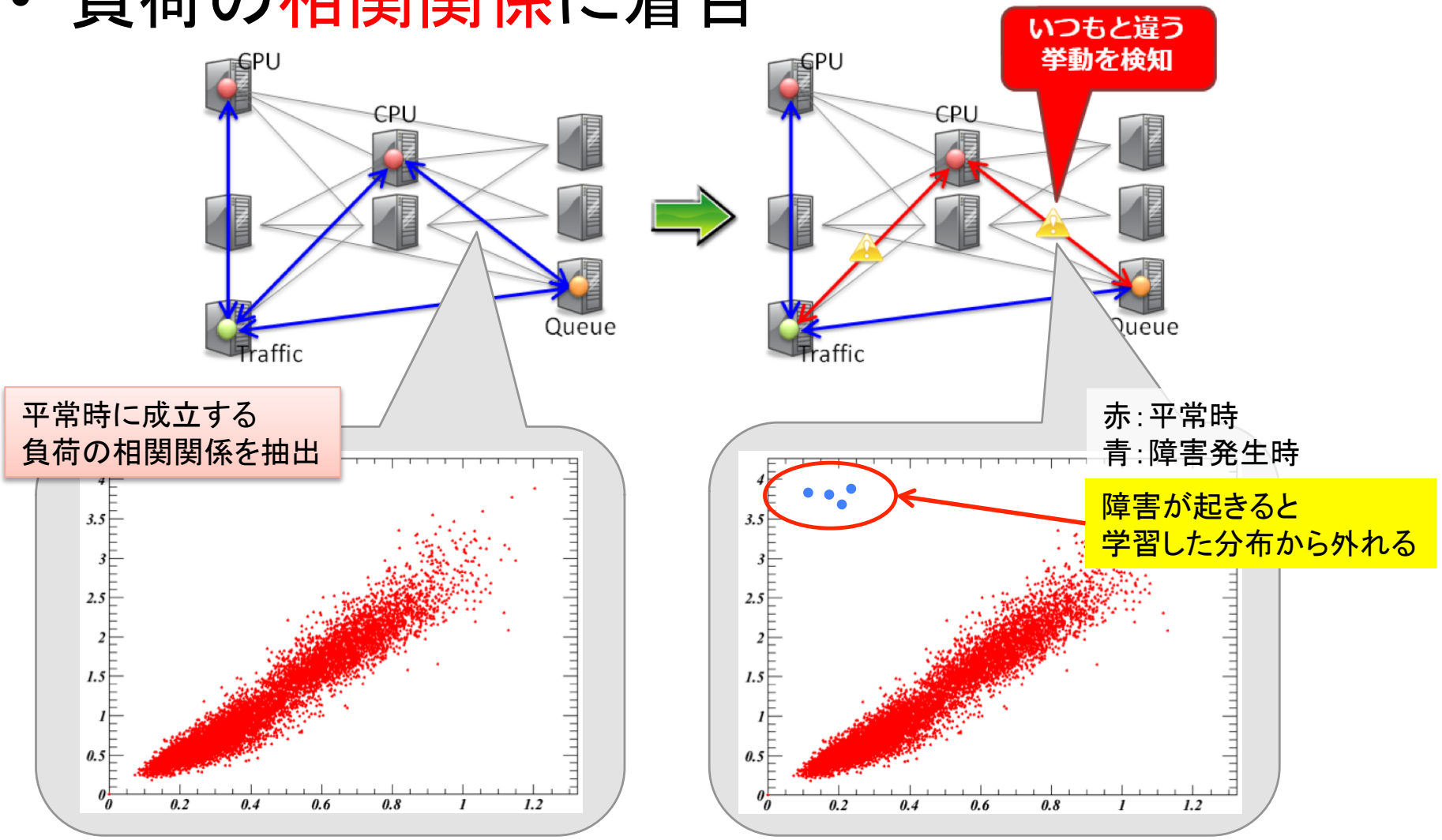
後段ホストのCPU使用率





相関異常検知のコンセプト

- 負荷の相関関係に着目

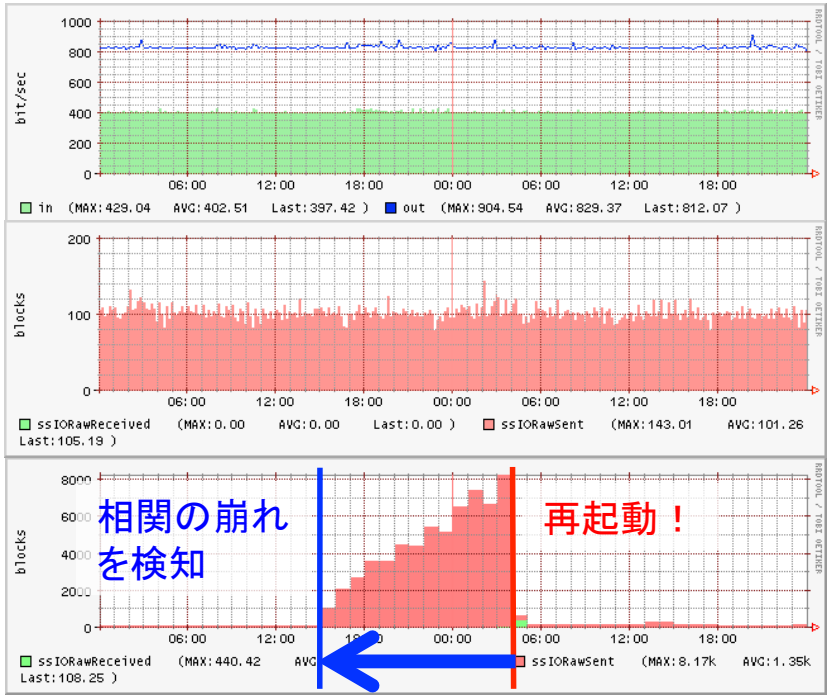


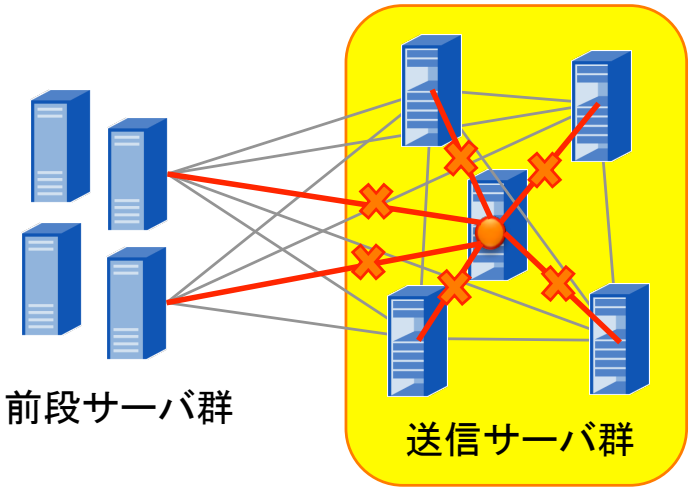
負荷の散布図　⇒ 線形関係になっている

問題が発生したホスト名、リソースを知ることができる

障害検出事例(1)

- ホストが再起動を起こしたケース
  - 該当ホストのCPU使用率に相関の崩れが集中

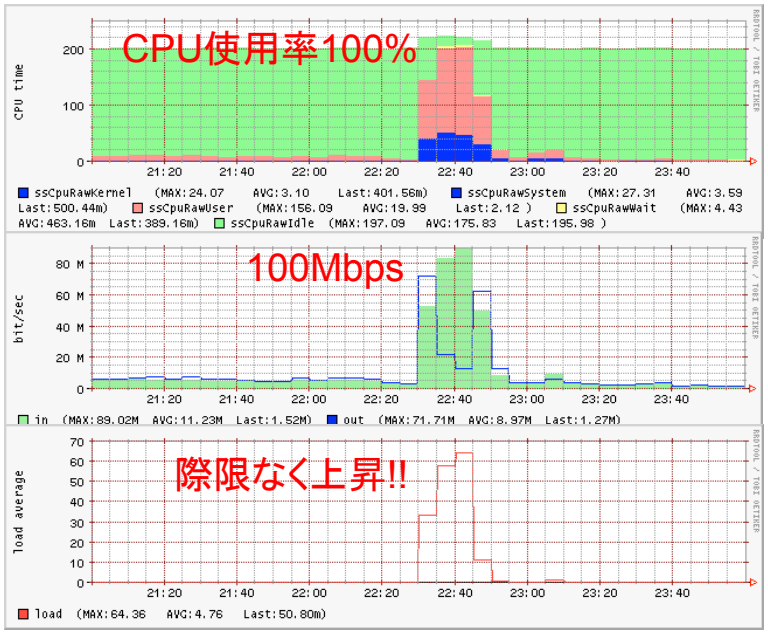


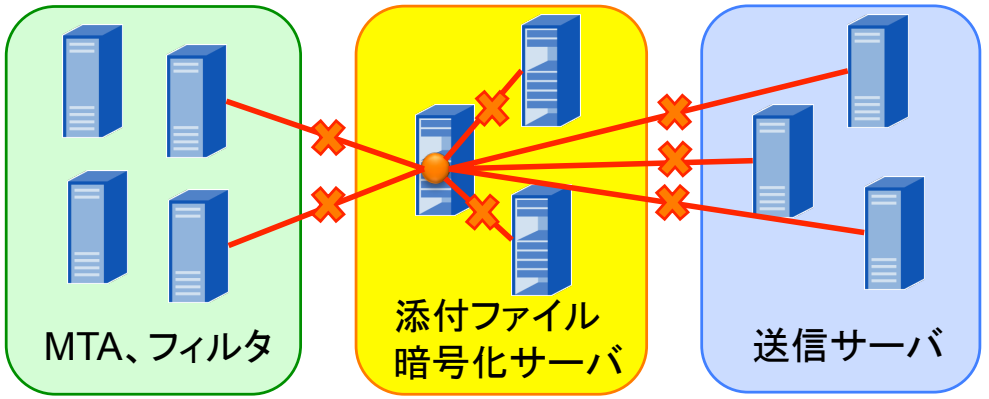


  - そのホストのCPU負荷だけが徐々に増加し、相関が崩れた
  - 兆候を事前に検出
    - 予防できるようになる

障害検出事例(2)

- メールサービスの配送遅延障害
  - 暗号化サーバのロードアベレージに相関の異常





  - 大サイズファイルが添付されたメールを大量に受信
    - 全体の負荷が上昇し、フル稼働状態
    - 一方、暗号化処理が追いつかずロードアベレージだけが上がり続け相関が崩れた
  - サービスレベルの低下を検出
    - 影響を最小限にできていたはず

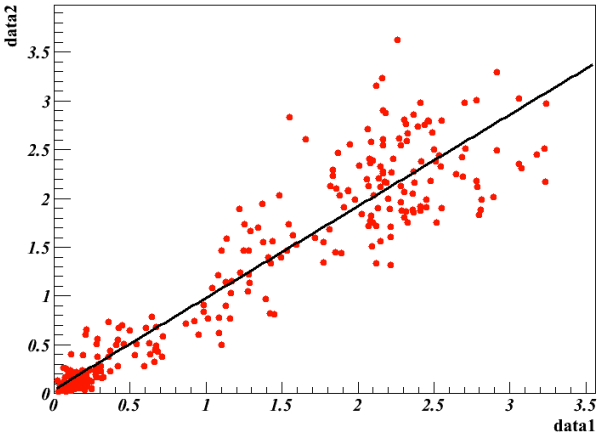
相関異常検知の課題

- 誤検知
  - 「相関が崩れた＝障害発生」とは限らない
- ベイズフィルタによる障害原因推定
  - 学習内容
    - どの役割のホストの、どのリソースか？
    - 障害原因は？
  - 出力
    - 障害原因がAである確率：70%
    - 障害原因がBである確率：25%　　⇒ 原因はAと推測
    - 障害原因がCである確率：　5%

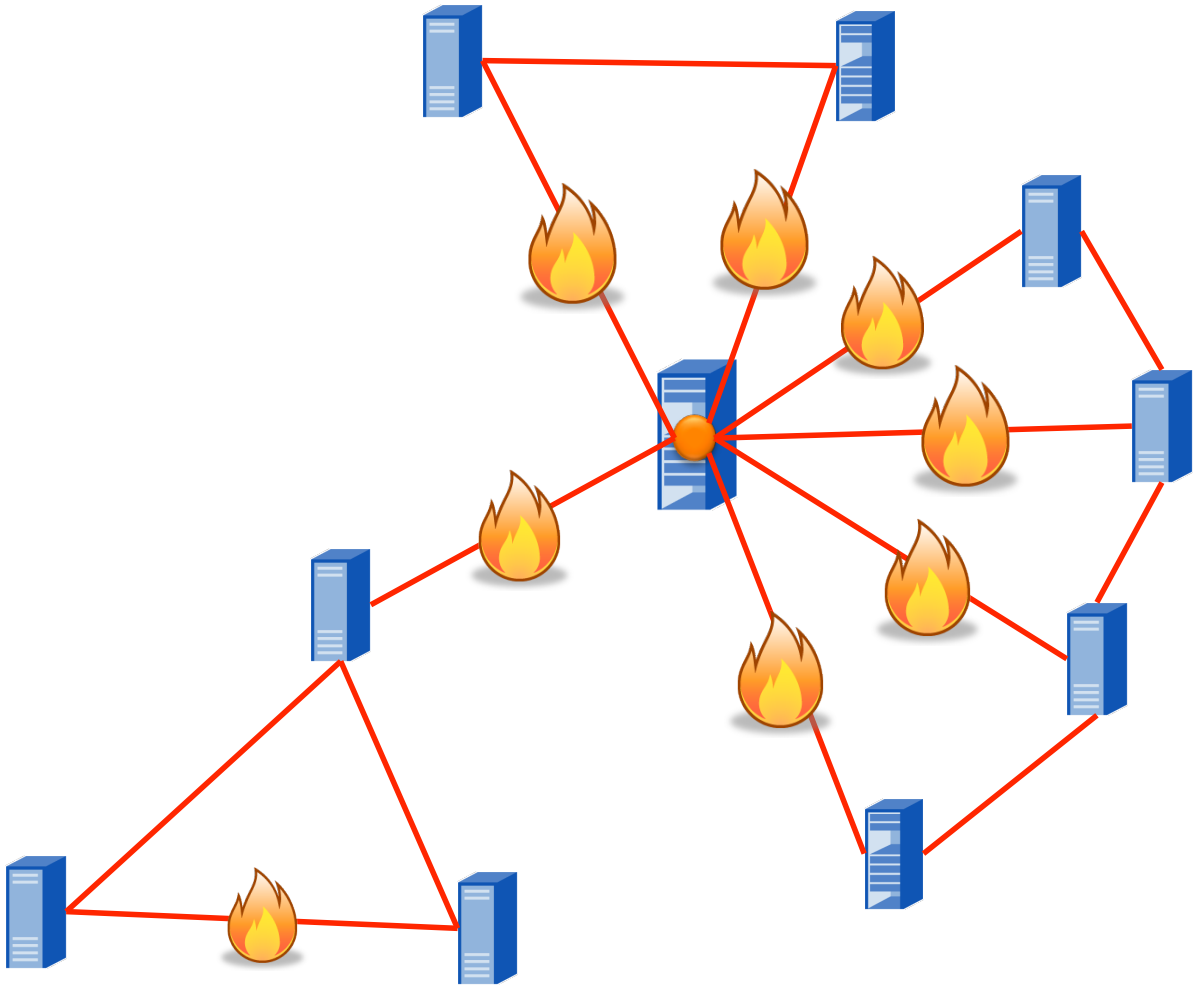
backup slides

モデル化と異常の判定

- 学習対象
  - 全てのリソースの組合わせで相関関係を抽出
  - 相関のあるものだけを選び出し、モデル化
- 検知の方法
  - 学習したモデルから大きく外れていたら異常と判定
  - 通知しないケース
    - 一瞬だけ相関が崩れた場合（スパイク）
    - 相関の崩れた箇所が少ない場合
- 学習期間
  - 2週間
  - 平日と休日を分ける必要は無い

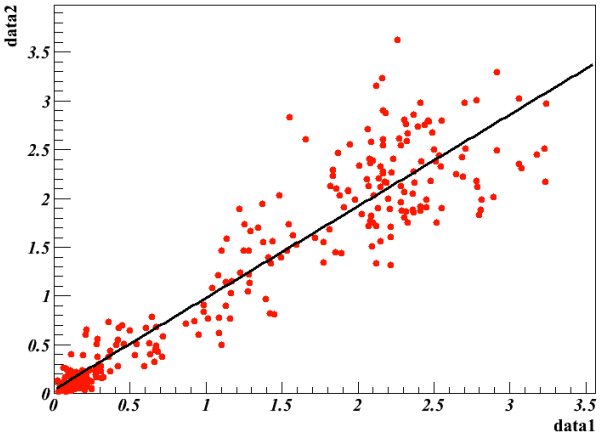


検知しないケース



モデル化と異常の判定

- 学習対象
  - 全てのリソースの組合わせで相関関係を抽出
  - 相関のあるものだけを選び出し、モデル化
- 検知の方法
  - 学習したモデルから大きく外れていたら異常と判定
  - 通知しないケース
    - 一瞬だけ相関が崩れた場合（スパイク）
    - 相関の崩れた箇所が少ない場合
- 学習期間
  - 2週間
  - 平日と休日を分ける必要は無い



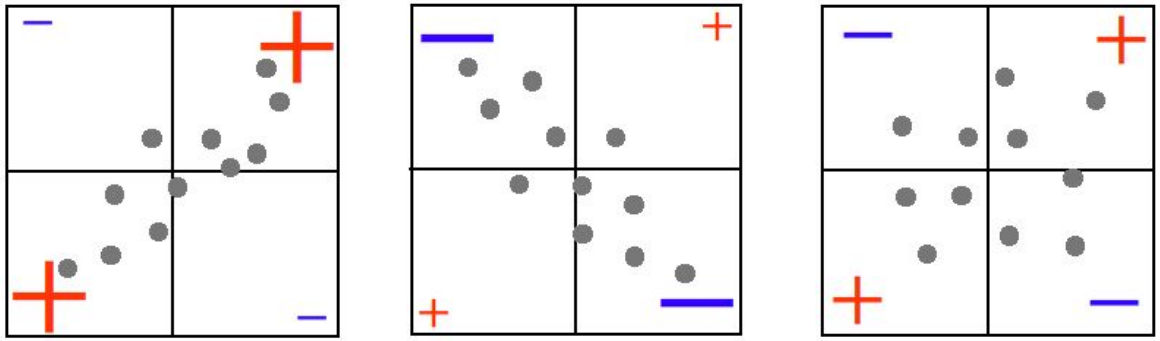
今後の取り組み

- 障害原因の提示
  - 現段階では相関が崩れた場所を知らせるのみ
  - 障害原因を推測するシステムに
    - 「相関の崩れ方」と「障害原因」をセットで学習
    - パターンマッチによって障害原因を提示
    - 誤検知抑制にも効果が期待できる

検知の仕組み

- 相関の有無を判定
  - ピアソンの積率相関係数

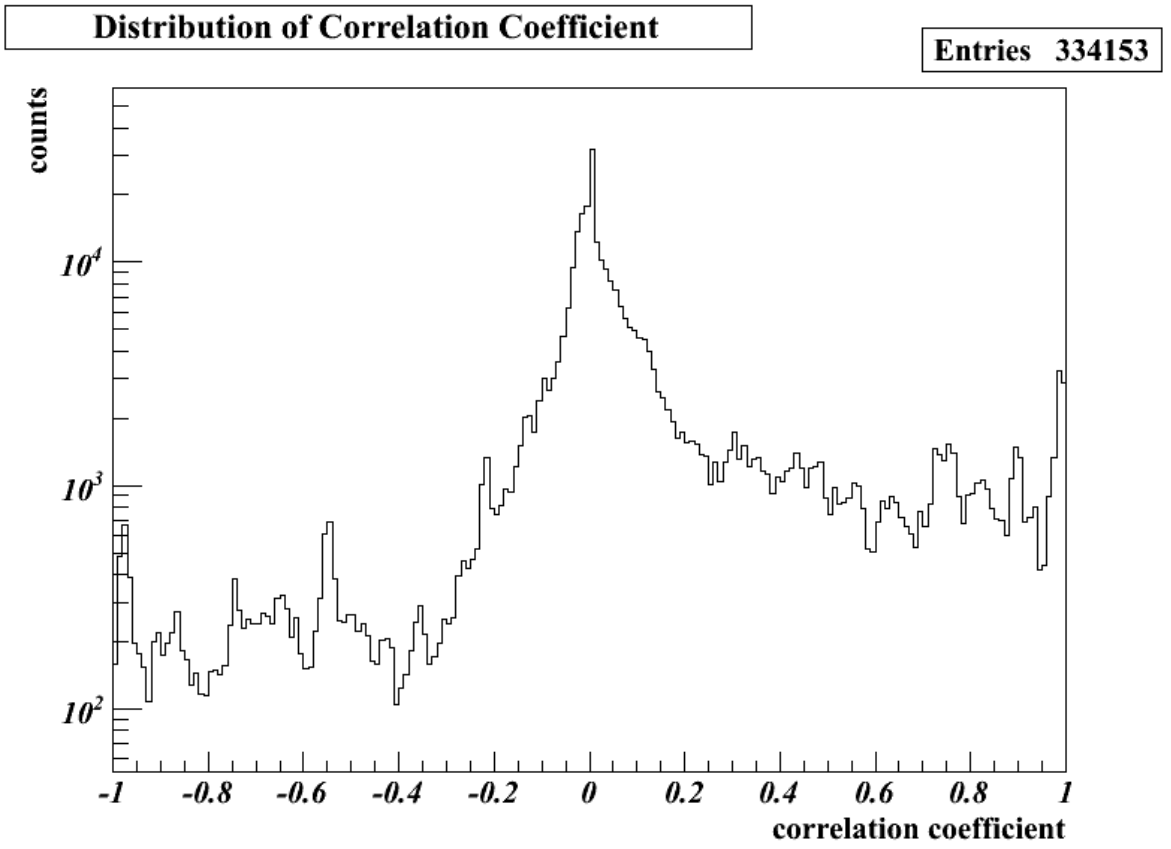
$$r = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$



一般に、| r | > 0.8 のときに強い相関を持つ
⇒ この条件を使って、相関を持ったデータの組合わせを選び出す

相関係数

- 強い相関を持つデータペアを選び出す
  - fabs(相関係数) > 0.8



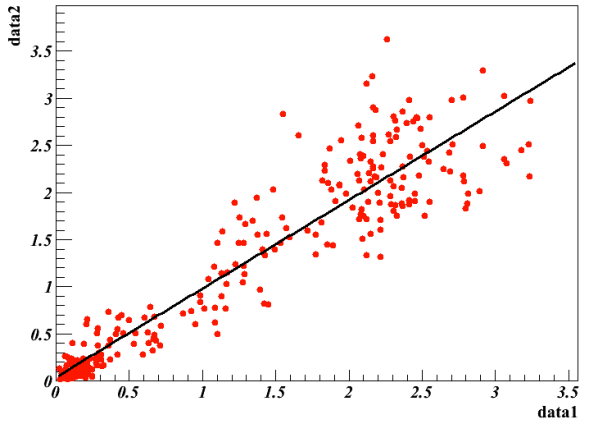
検知の仕組み

- データの規格化
  - データによって取り得る値の散らばりが違う
    - ロードアベレージ : 0 ~ 10 程度の値をとる
    - トラフィック　　: 0 ~ 10,000 の値をとる
  - スケールの違うデータを公平に扱いたい
    - 標準偏差で割る

$$X = \frac{x_i(t)}{\sigma_i} \qquad \sigma = \sqrt{\frac{1}{N}\sum_{i}^{N}(x_i - \overline{x})^2}$$

$$Y = \frac{y_j(t)}{\sigma_j}$$

- フィッティング
  - Y = a X + b でフィッティング
    - データペアの関係を表す
  - a, b を学習する
    - 学習期間 : 二週間



フィッティング

- 最小二乗法

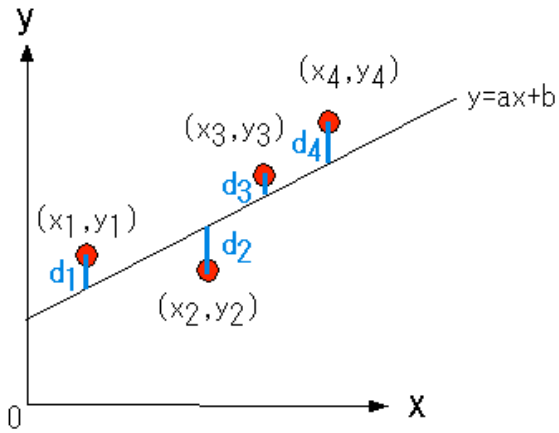
データ集合 $(x, y) = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ を

$y = ax + b$ でフィッティングするとき、

フィッティングパラメータ a, b は下記の通り

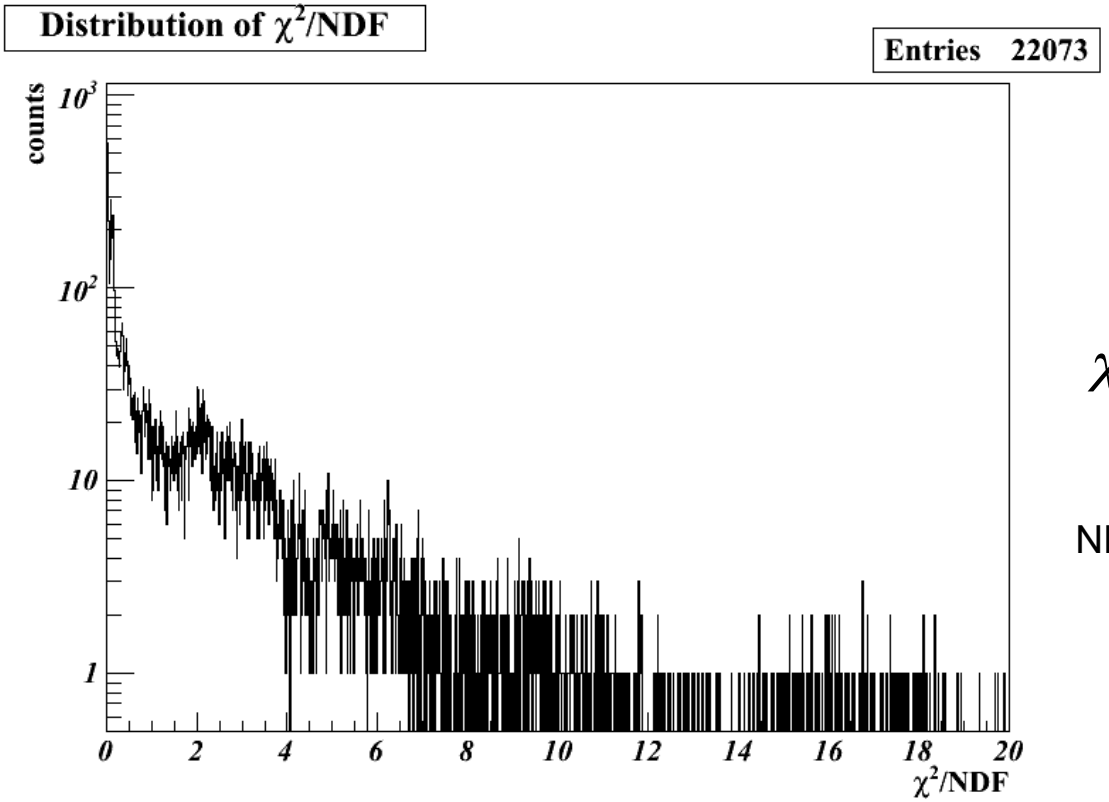
$$a = \frac{n\sum_{k=1}^{n} x_k y_k - \sum_{k=1}^{n} x_k \sum_{k=1}^{n} y_k}{n\sum_{k=1}^{n} x_k^2 - \left(\sum_{k=1}^{n} x_k\right)^2}$$

$$b = \frac{\sum_{k=1}^{n} x_k^2 \sum_{k=1}^{n} y_k - \sum_{k=1}^{n} x_k y_k \sum_{k=1}^{n} x_k}{n\sum_{k=1}^{n} x_k^2 - \left(\sum_{k=1}^{n} x_k\right)^2}$$



フィッティング

- フィッティングの良し悪しを判定
  - 上手くフィッティングできているものだけを選ぶ



$$\chi^2 = \sum_{i=1}^{N} \left(\frac{x_i - \bar{x}}{\sigma_i} \right)^2$$

NDF = 点の数 − パラメータ数

検知の仕組み

- 異常度を計算
  - 部分異常度

$$d_{ij} = \left|(a \times \mathrm{data}_i + b) - \mathrm{data}_j\right|$$

    - マハラノビス距離と等価
  - 総合異常度
    - 部分異常の平均値
    - 10 σ を閾値に設定
  - 異常の継続時間
    - 「一瞬だけ相関が崩れたが直ちに元の正常な状態に戻る」
    - この場合、アラートは不必要
    - アラート送信条件:「相関の異常が15分以上継続している」