

Webサービス基盤の自律制御と動的平衡性

なめらかなシステムのアイデアと実現に向けて

松本亮介 / GMO PEPABO inc.

2016.07.02 はてな・ペパボ技術大会～インフラ技術基盤～@京都



松本亮介 @matsumotory

シニア・プリンシパルエンジニア

技術部技術基盤チーム 研究開発担当

<http://research.matsumoto-r.jp/>

目次

- ・なめらかなシステムのコンセプト
- ・Webサービスの現状と課題と人工知能のアプローチ
- ・Webサーバの自律制御
- ・Webサービス基盤の動的平衡性
- ・まとめ



なめらかなシステムのコンセプト

なめらかなシステムの要件

1. システムにとってのユーザや、システムを構成するサーバ等の要素が、どのようなカテゴリに属するか、いかなる特徴を持つかを、詳細に認識する
2. その際、ユーザやシステム管理者といった人間に労苦を強いるような明示的な操作を課さない
3. 1.および2.で得たより精緻なカテゴリや個別の特徴、あるいはそれらの間の関係性に基づき、その時々状況に応じて最適なサービスを提供する

本発表ではなめらかなシステムというコンセプトをWebインフラへ適用する例



次世代ホスティングサービスを 目指して。

ロリポップ!レンタルサーバーでは、より快適な
ウェブサイト運営が行えるサービスを提供するため、
サーバー環境の改善と革新的な技術の導入に挑戦しています。

Webサービスの状況と課題

Webサービス

ペパボ福岡のWebサービス

**LOLipop! RENTAL
SERVER**

hetemi
WE HOST YOUR CREATIVITY

**MUU
MUU
Domain**

minne



比較的規模の大きいロリポップ！

LOLIPOP! RENTAL SERVER

by **GMO** パパポ

200万サイト以上管理しており、1日のアクセス数は10億以上



Webサービスへの不満

- ・ユーザ・インタフェースが使いにくい
- ・表示が遅い、ダウンロードが遅い
- ・Webサイトにつながらない
- ・このWebサービスは使えない
- ・個人のSNS拡散であっという間に評判が落ちる時代



快適に利用できて当たり前の世界

Webサーバの運用が大変

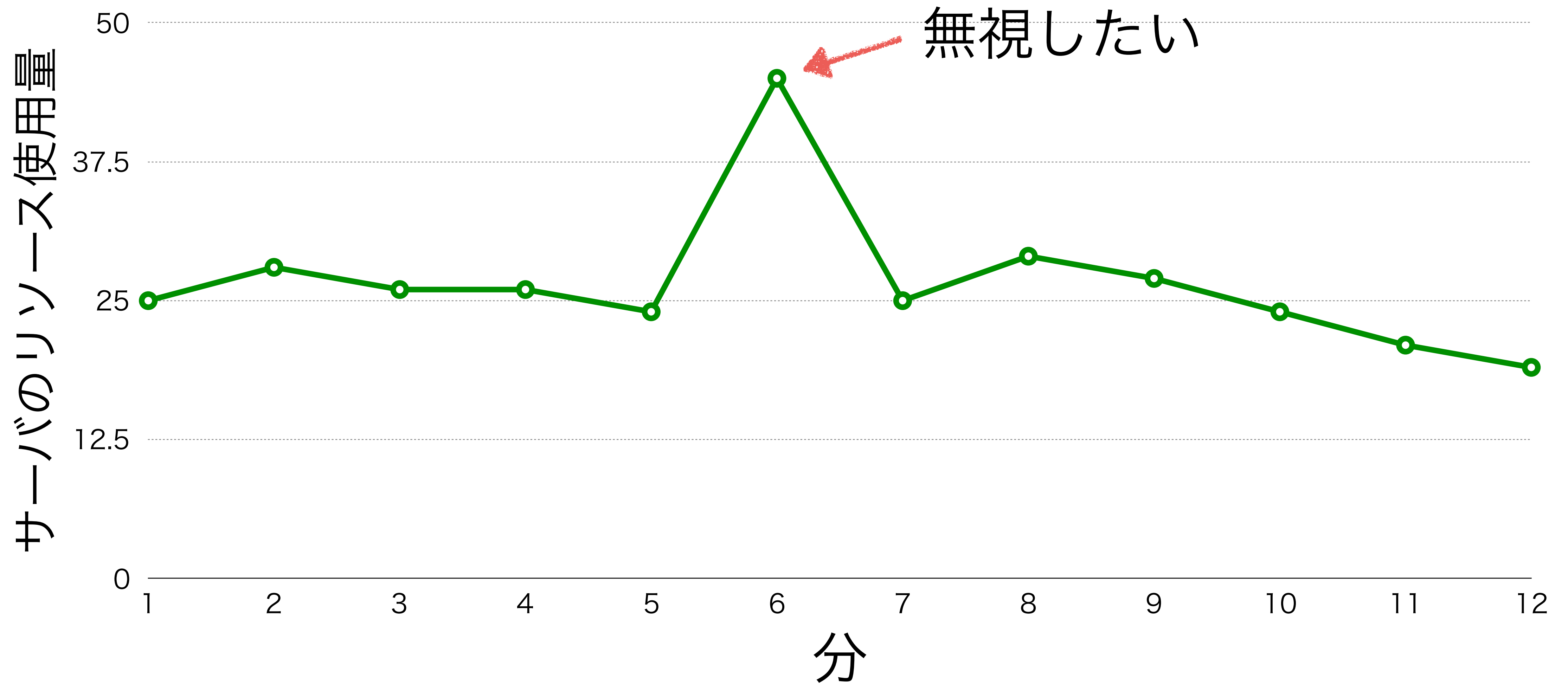
24時間365日のWebサーバ運用とは

- ・高負荷・障害対応
- ・問い合わせ調査・対応
- ・ハードウェア・ソフトウェアのバージョンアップ
- ・サービスや基盤（インフラ）の監視
- ・新規構築や予防措置など



サーバ監視の閾値設定は難しい

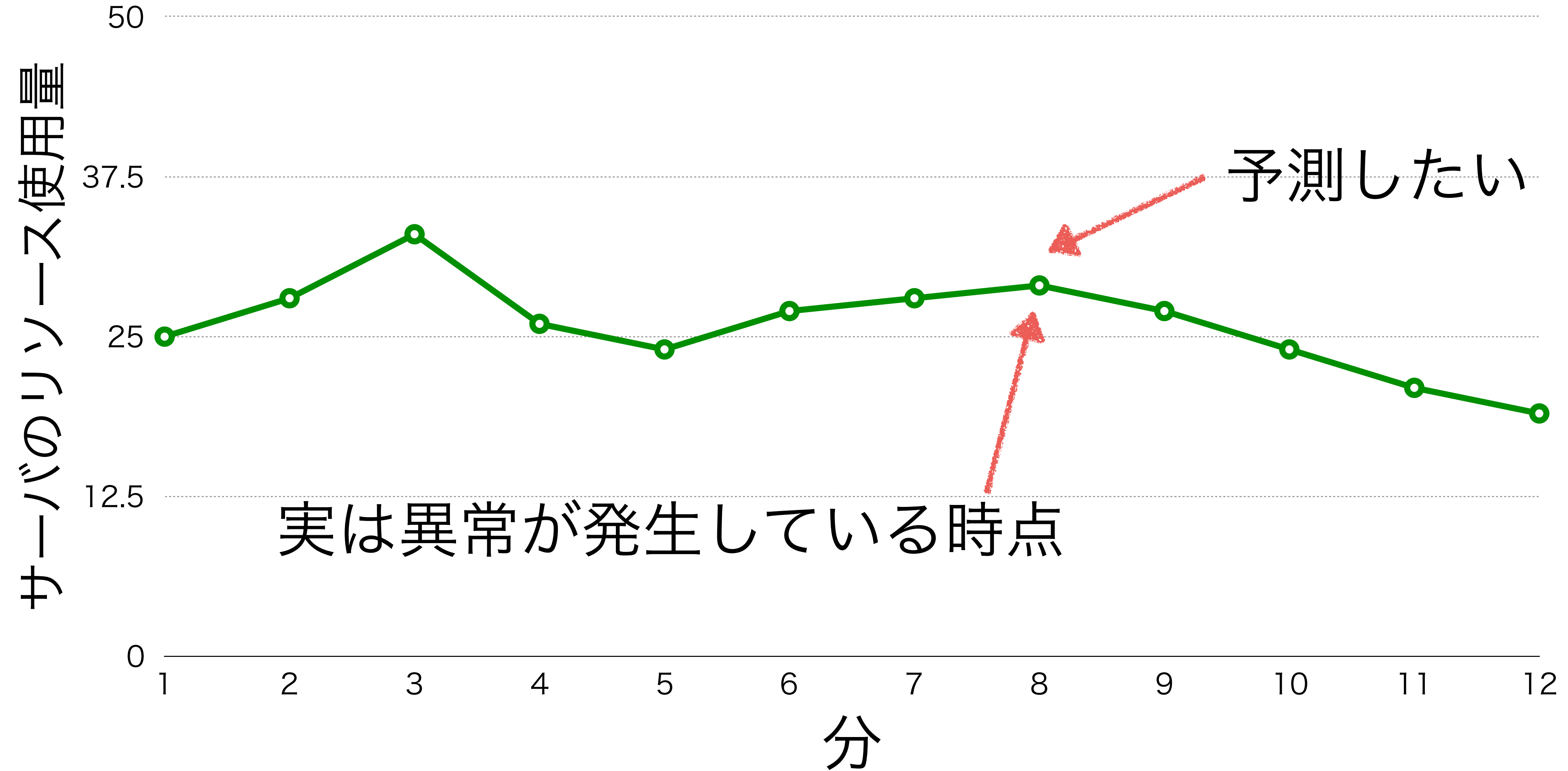
一時的な高負荷は無視したい



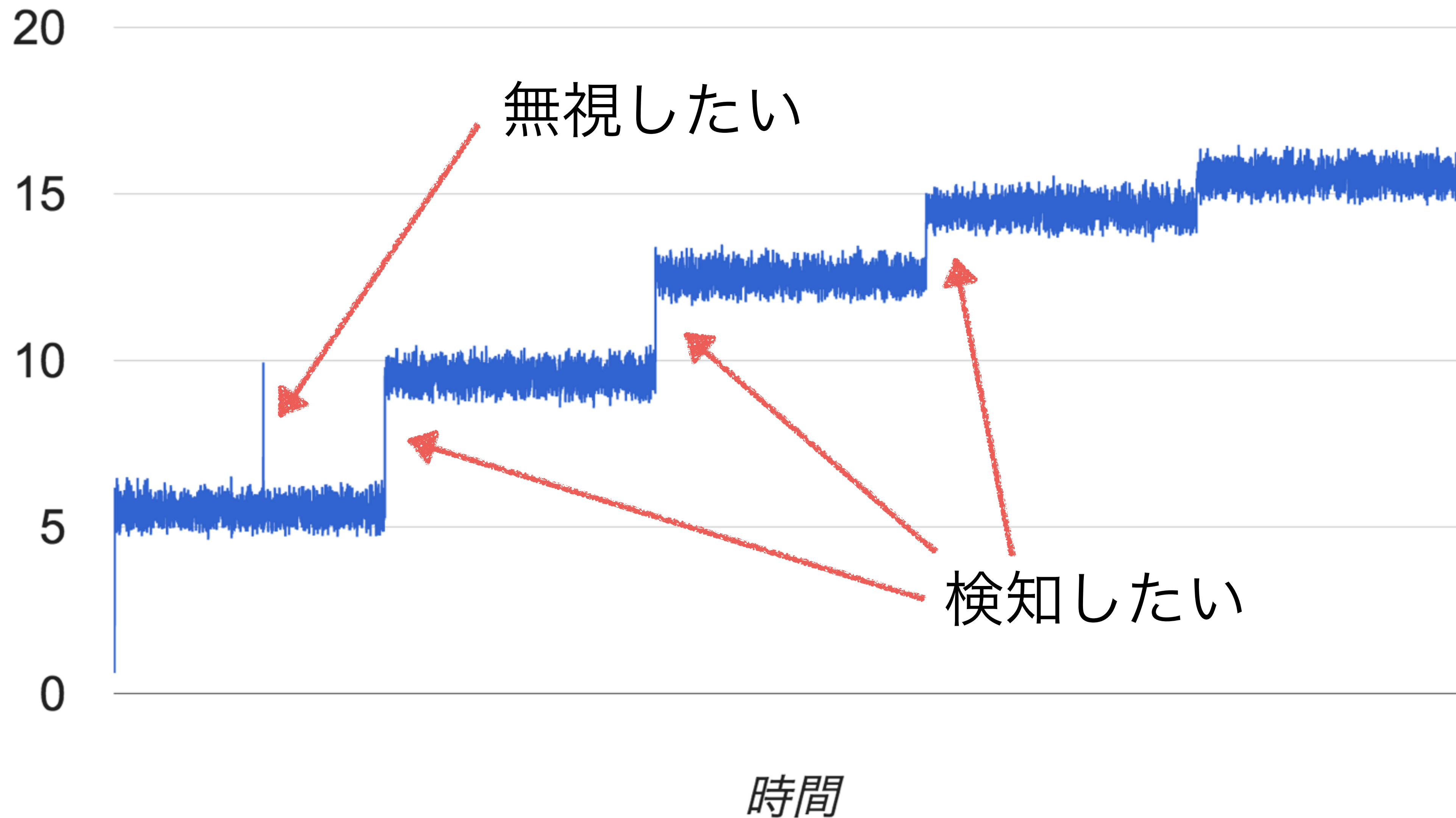
傾向が変化する高負荷は予測したい



正常に見える異常を検知したい



段階的な傾向の変化も検知したい



**詳細な変化とその原因を得るには
人の目による監視運用が必要**

でも自動で制御したい

人の目は対象の振る舞いを捉えている

振る舞いを表す特徴を抽出すれば良い

人工知能のアプローチ

考え方は非常にシンプル

- ・サーバやシステムには特徴があるはず
- ・サーバの特徴の通常状態を学習しておく
- ・通常状態から外れた状態を解析する
- ・連続的に外れた状態を異常のはじまりとする
- ・データマイニングの観点から外れ値・変化点検出



**Webサーバの異常状態を予測して
自動で制御**

各種基盤技術は完成

- Webサーバ設定をプログラム化※1 済
- 設定プログラムとOSのリソース制御を連携※2 済
- 後は上記の技術間で特徴量の定義と解析が必要

※1 松本亮介, 岡部 寿男, mod_mruby: スクリプト言語で高速かつ省メモリに拡張可能なWebサーバの機能拡張支援機構, 情報処理学会論文誌, Vol.55, No.11, pp.2451-2460, Nov 2014.

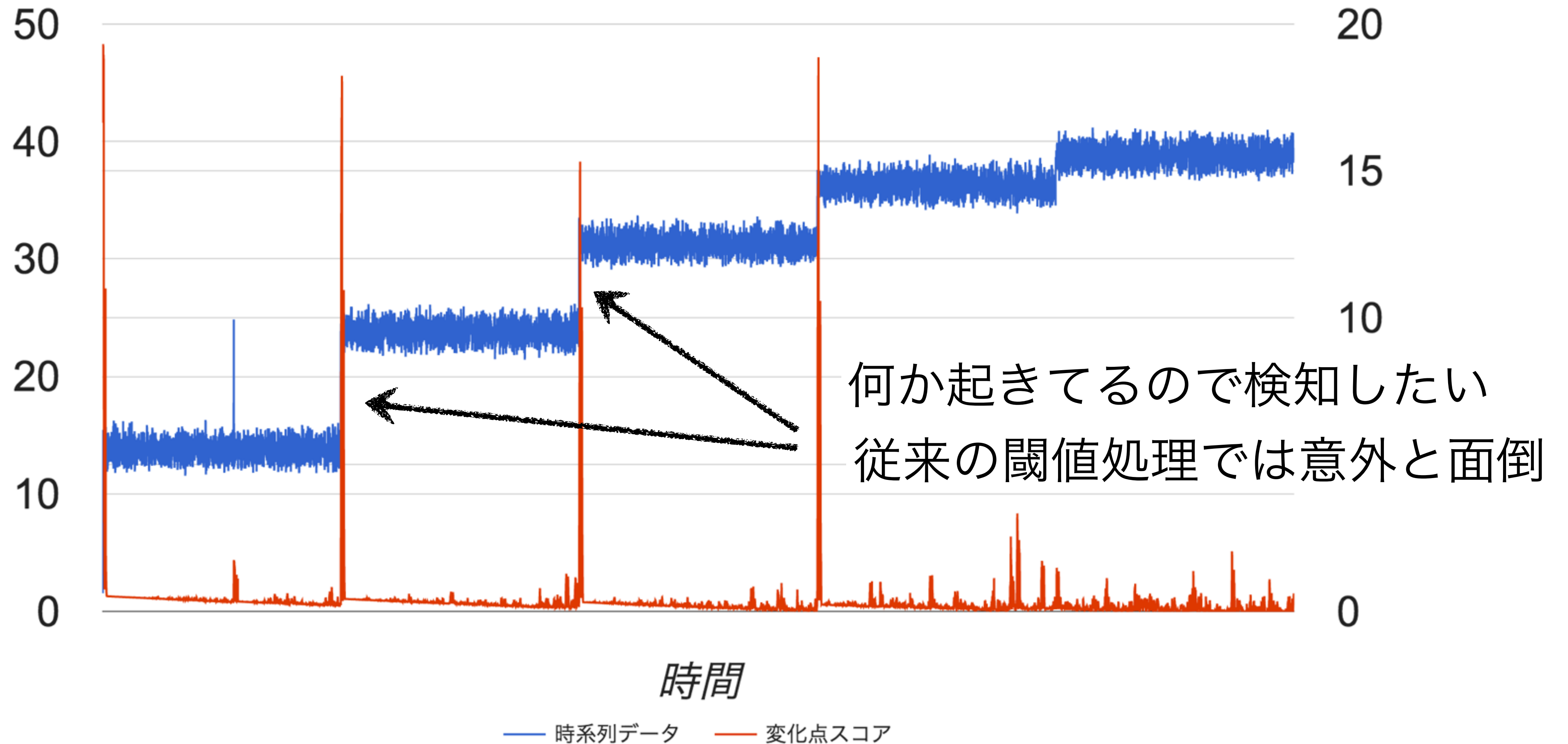
※2 Ryosuke Matsumoto, Resource Control Architecture for a Web Server Separating Computer Resources Virtually at Each HTTP Request, RubyKaigi 2014, Sep 2014.

解析

解析は変化点検出^{※1}やクラスタリングなど 様々な手法がある

※1 J. Takeuchi and K. Yamanishi, “A Unifying Framework for Detecting Outliers and Change Points from Time Series,”
IEEE transactions on Knowledge and Data Engineering, pp.482-492, 2006.

シミュレーションデータ

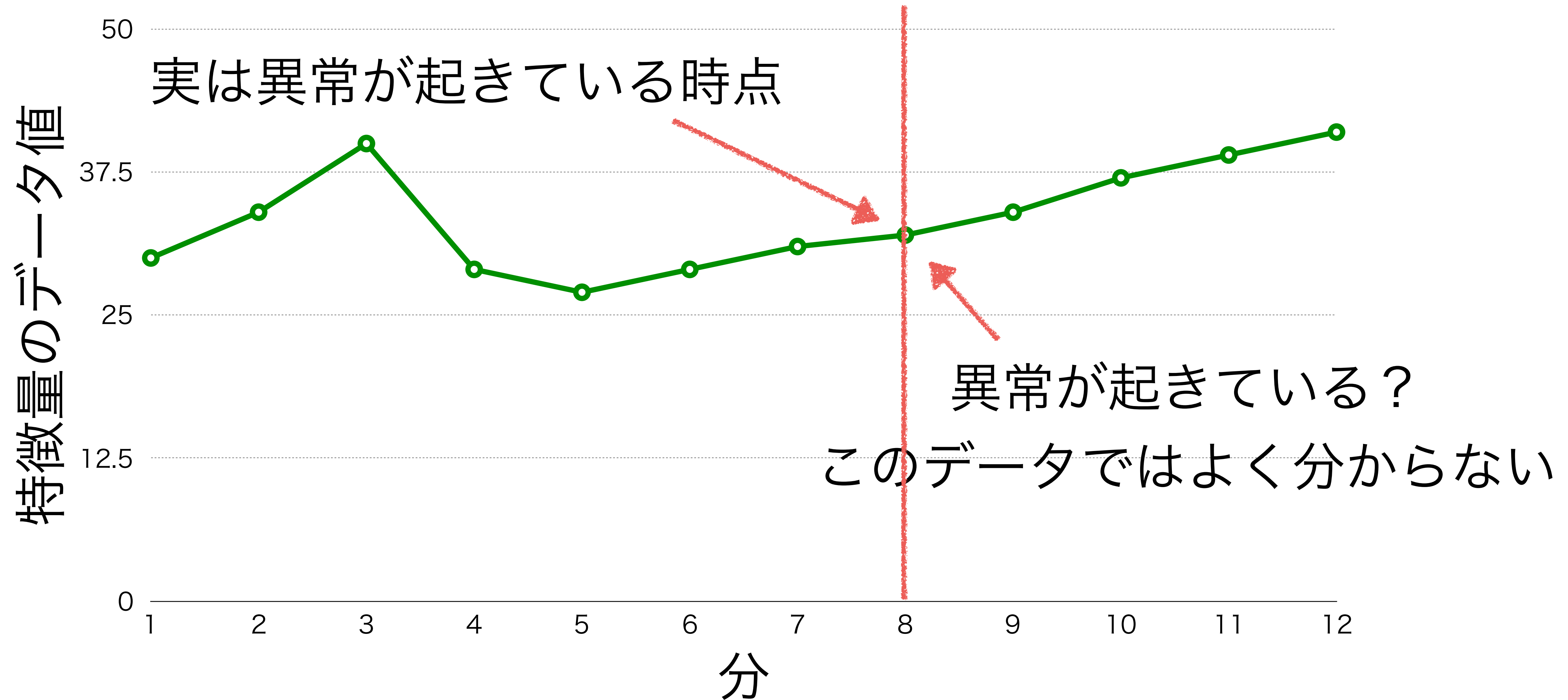


※1 松本亮介, mruby-changefinder, <https://github.com/matsumoto-r/mruby-changefinder>, 2016年3月.

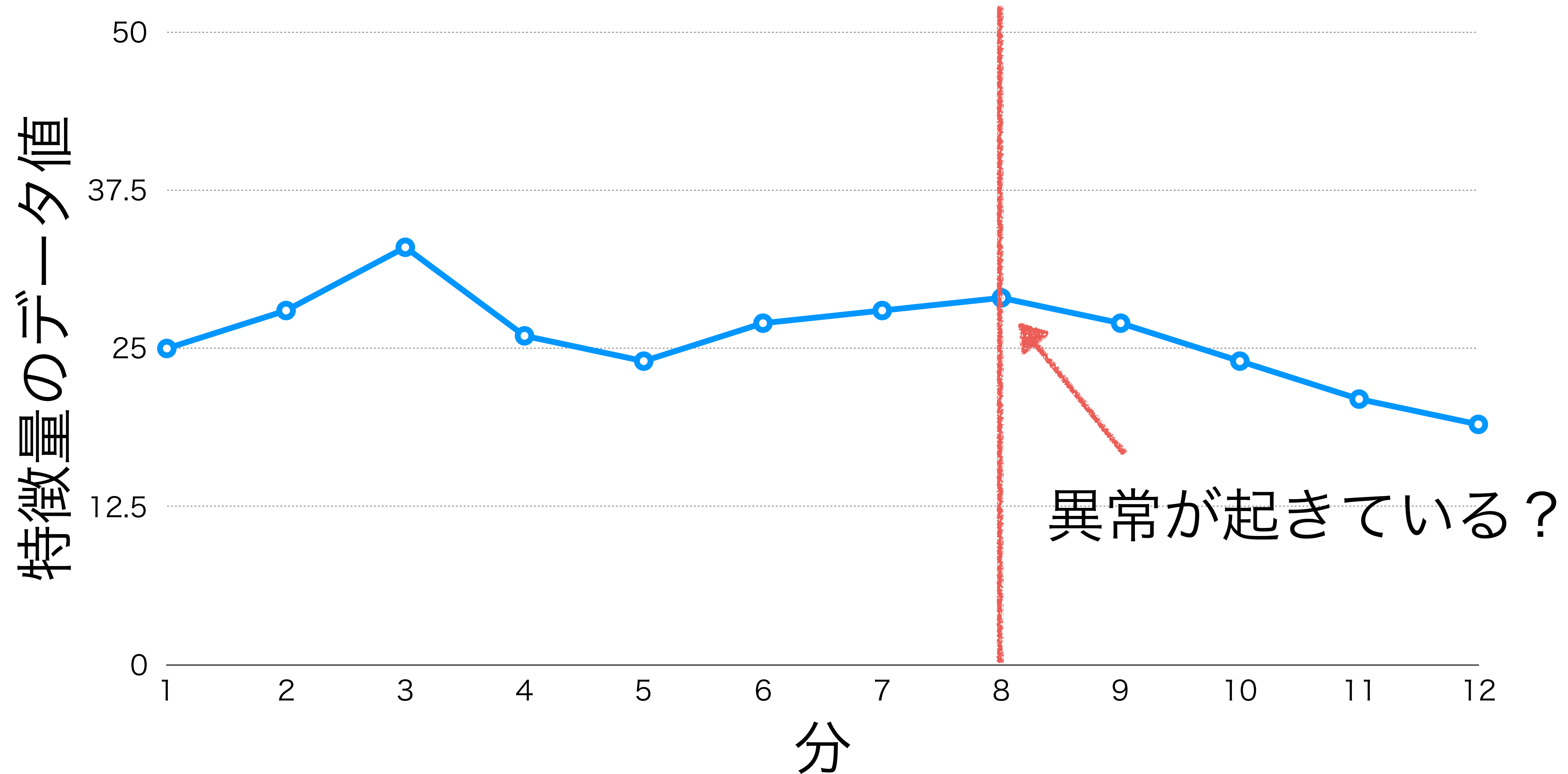
特徴量の定義が重要

例えば複数データの関係性を得る

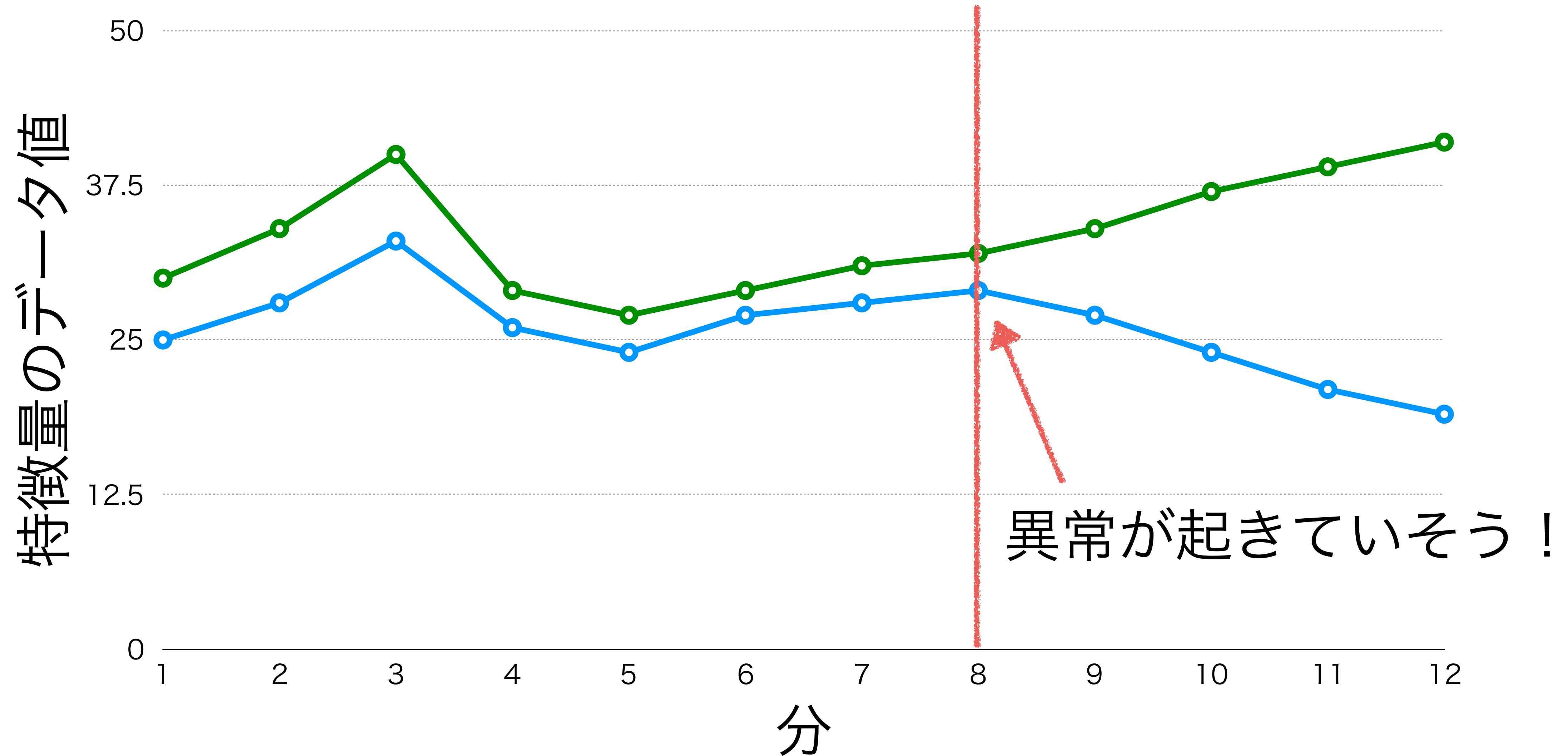
特徴量Aの時系列データ



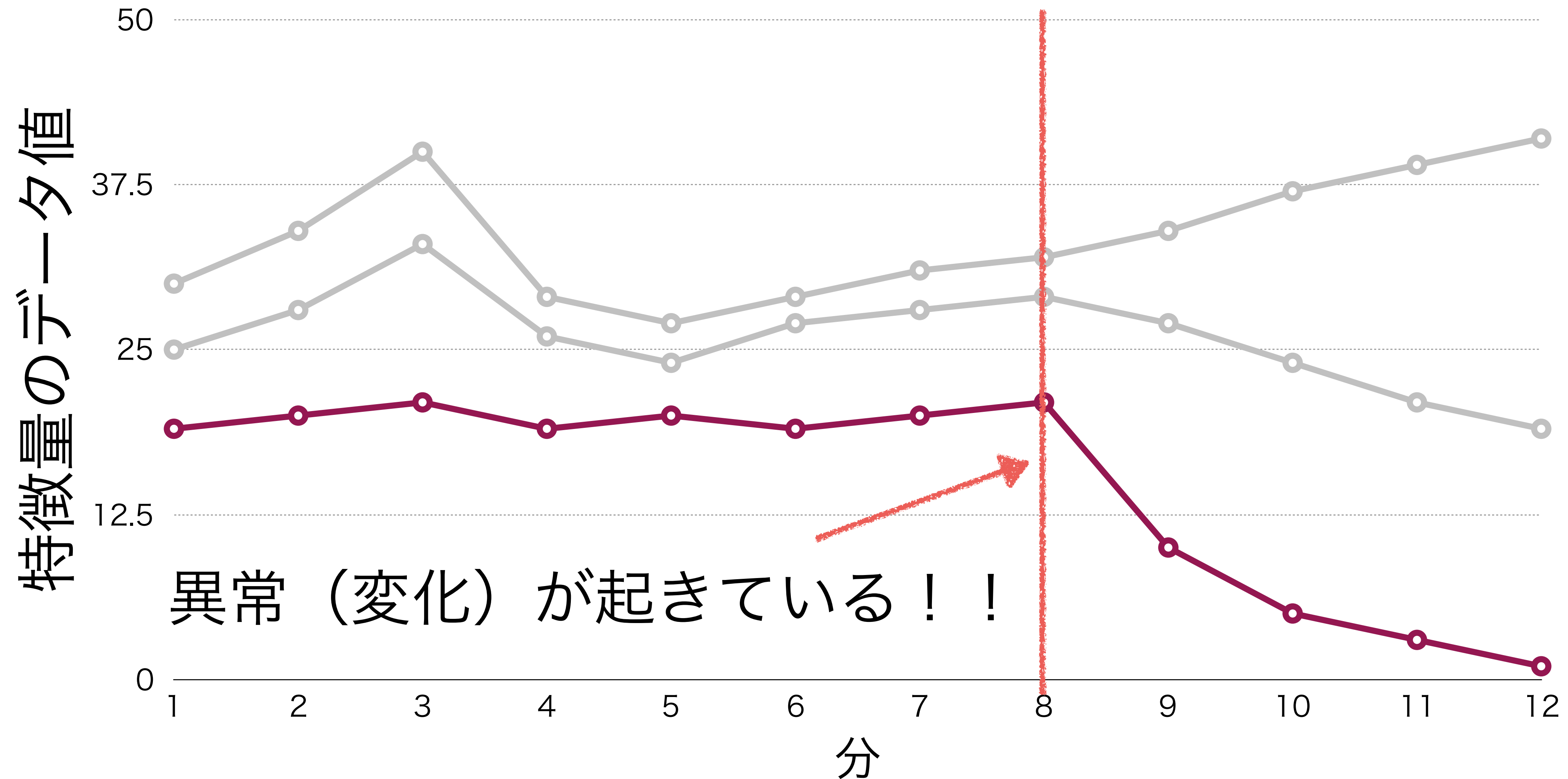
特徴量Bの時系列データ



AとBのデータ間の相関関係を見る



AとBの相関係数の時系列データ

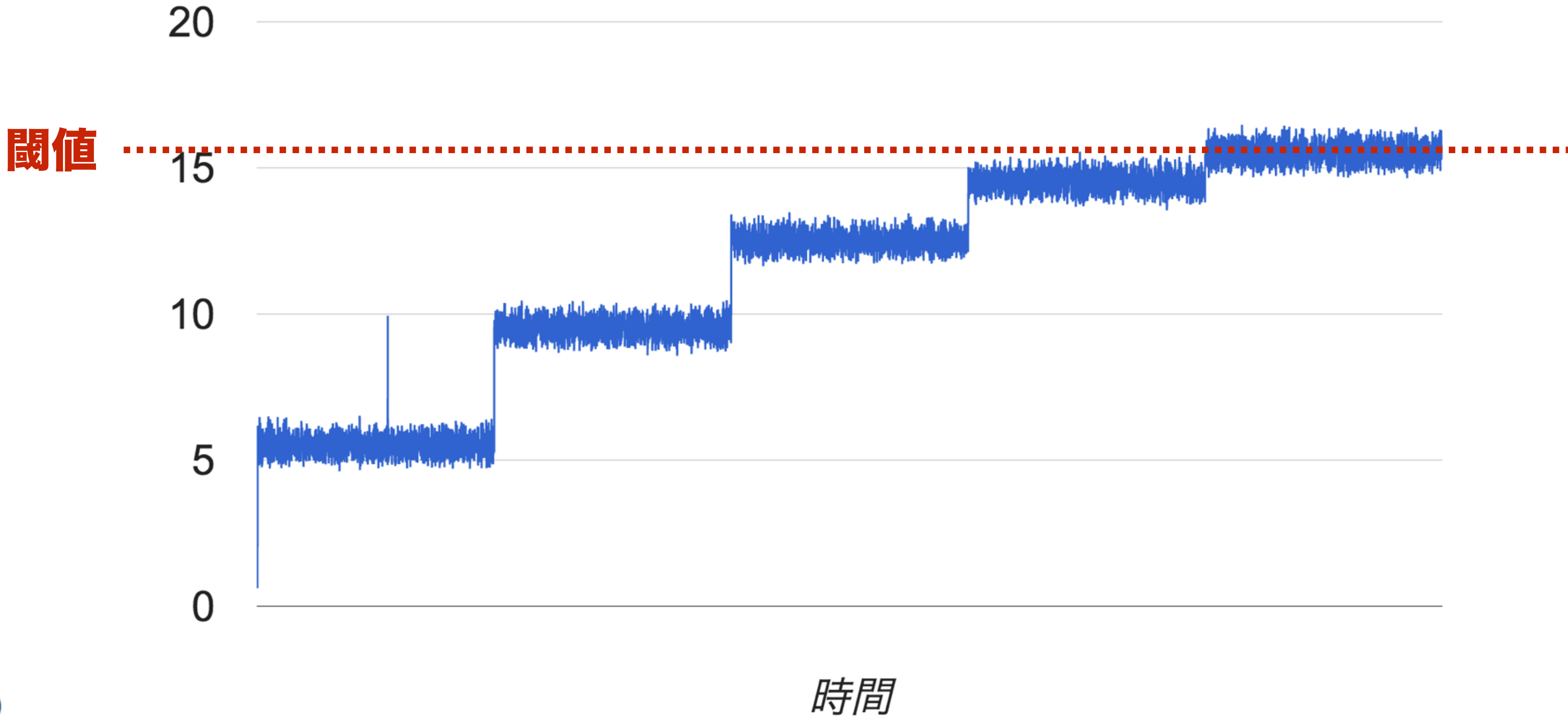


※1 松本亮介, トラフィック特徴量の相関特性を用いた異常検出, 平成19年度 卒業研究発表会, 2008年3月.

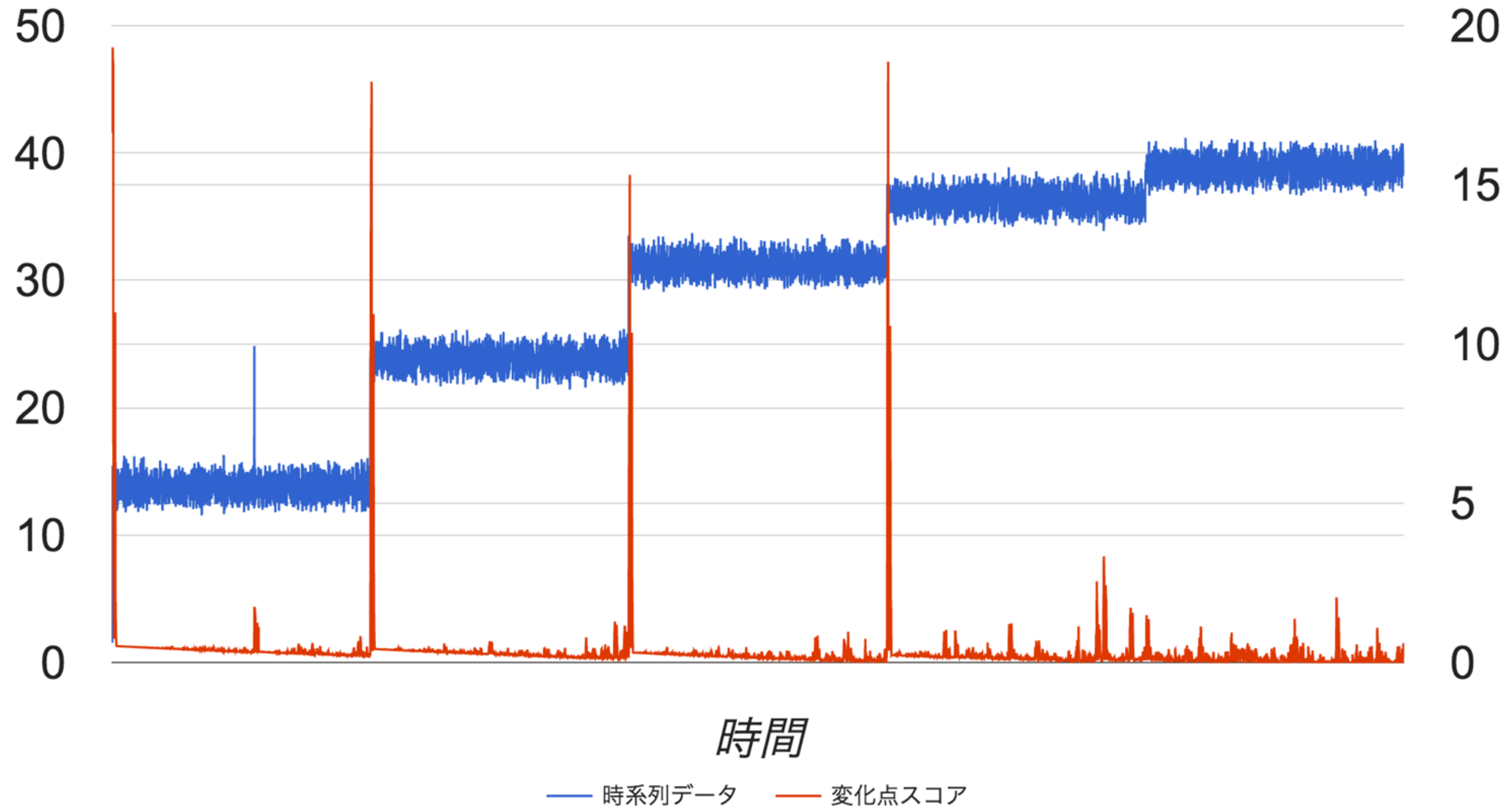
**自己あるいは関係性を監視し
自動でリソース制御する技術**

Webサーバの自律制御

従来の閾値監視



特徴量抽出による検知できる時点の増加



閾値以下でも傾向変化を細かく検知できる

- ・傾向変化の積み重ねにより高負荷になると仮定
- ・高負荷検知時には原因が無数にある状態になる
- ・閾値は超えていないが傾向が変化した時点を知りたい
- ・Webサービス基盤においては変化点や外れ値には何か原因があるはず



細かく傾向変化した時点で何をすべきか

- ・異常検知なのでfalse positive/negativeが重要
- ・検知してすぐ制限をすると誤検知の影響が大きい
- ・とはいえ検出したら原因を特定してアクションすべき
- ・これらの特性を活かす制御手法を考える
- ・非定常なデータの統計値をどう活用するか



バランスをとりたい条件

- 例えば高集積なマルチテナント環境を想定
- 全体のリソースが空いている状況では自由に使える
- リソースが逼迫してきた時のみに制限をかける
- 逼迫時の制限対象は問題となるテナントのみ
- 全体として安定したなめらかな状態を作りたい



レンサバを例に検知・制御を設計

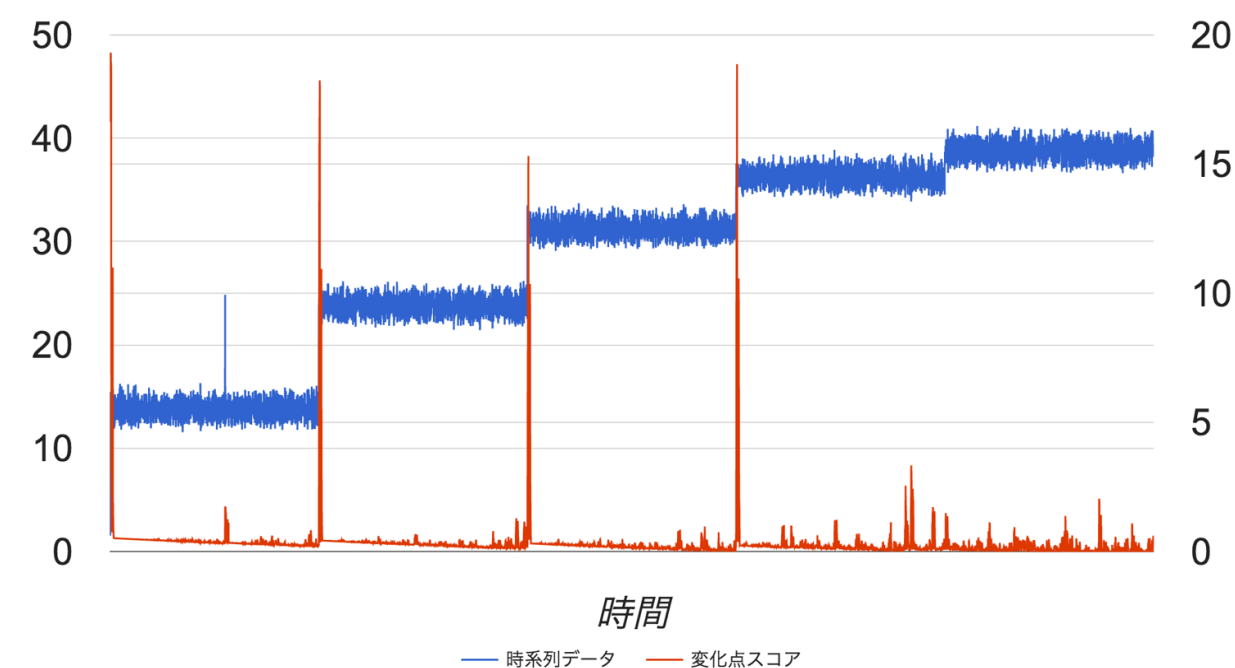
- ・変化点検出時にどのホストが変化点スコアが高いか
- ・各ホストのリソース使用傾向の重み付けを行う
- ・重み付けリストを傾向変化毎に更新していく
- ・全リソースが逼迫してきたらリストを元に自動制御



重み付けリストの生成

Webサーバ
プロセス

ホスト毎にリソース使用傾向
の重み付けリストを更新



重み付けリスト

exmapple.com 95
matsumoto-r.jp 93
example.jp 83
example.org 74
.
.

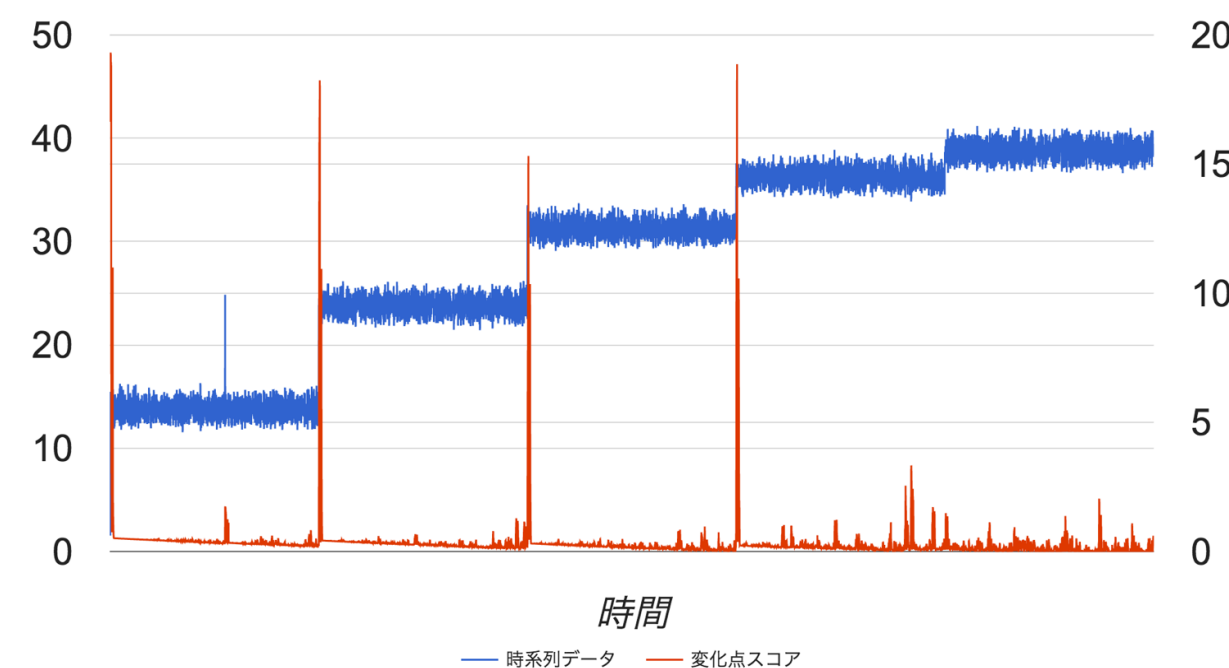
例えば高いスコアを計測した場合のみリストを更新してく等

サーバのリソースが逼迫してきたら制御

サーバリソースが逼迫していない場合はリストの重み付けのみに務める

Webサーバ
プロセス

ホスト毎にリソース使用傾向
の重み付けリストを更新



重み付けリスト

exmapple.com 95
matsumoto-r.jp 93
example.jp 83
example.org 74
.
.

サーバ全体のリソースが逼迫してきたら

● リストを元にWebサーバ自身が制御

時系列データの生成

- ・レスポンス生成に消費したリソース値から生成
- ・関係性のある複数の時系列データから相関関係の時系列データを作ることできる
- ・サーバ全体の变化とホスト単体の变化に相関があればそれはサーバに大きく負荷をかけている、など



データ生成と解析ソフトウェア

- Webサーバーの機能拡張 `mod_mruby/nginx_mruby`
- プロセスのリソース消費値取得 `mruby-resource`
- 変化点検出エンジン `mruby-changefinder`
- 相関係数生成エンジン `mruby-correlation`

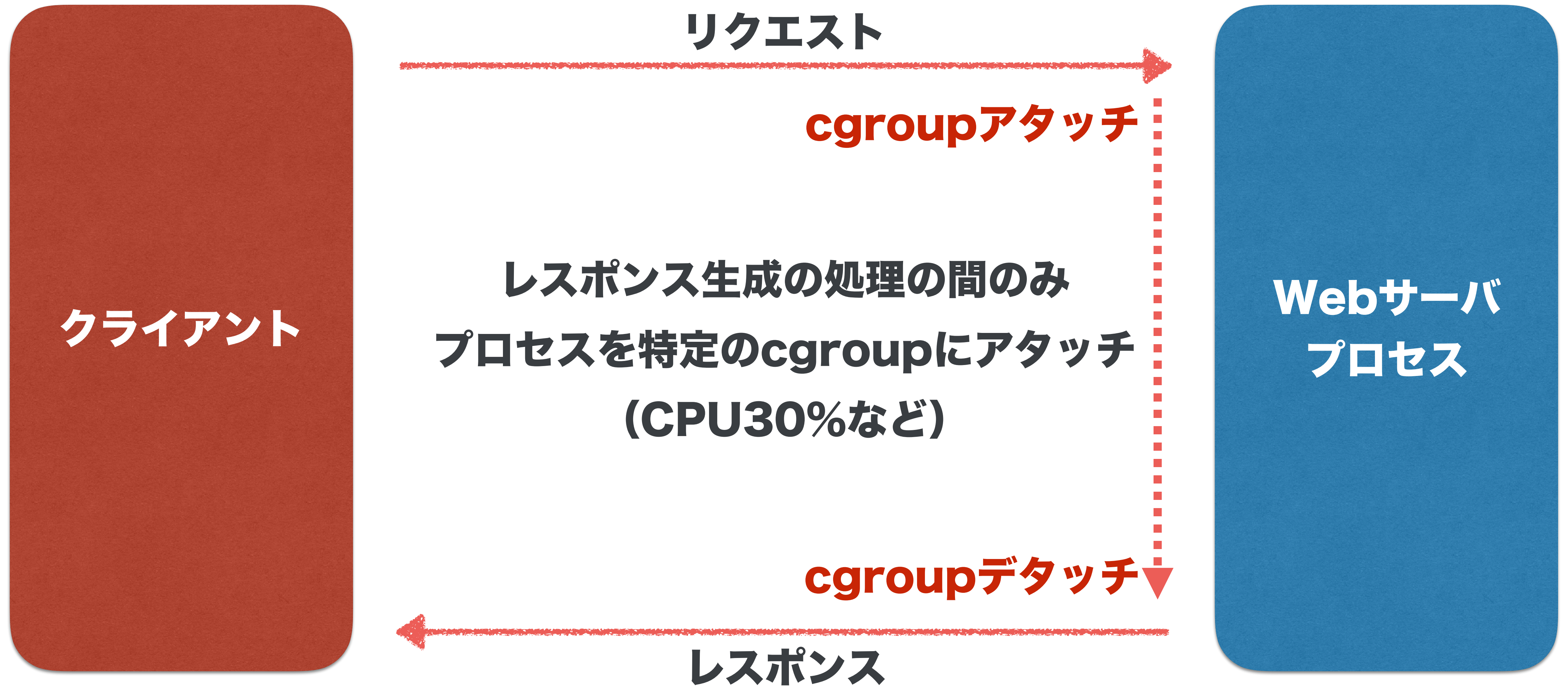


制限の実装方法

- リクエスト単位でCPUやI/Oなどの割り当を変更
- `cgroup(mruby-cgroup)`や`rlimit(mruby-resource)`
- ホスト単位での同時接続数を変更
- `mod_mruby`や`mod_vhost_maxclients`



リクエスト単位でのリソース制御



**ミドルウェアが特徴量抽出・解析・制御
を自律的に行う**

Webサービス基盤の動的平衡性

生命とは

生命とは

- ・動的平衡にある流れ
- ・細胞周期チェックポイント



生命とは動的平衡にある流れ^{※1※2※3}

- ・生体構成成分のエントロピー増大の法則による崩壊
- ・乱雑さが蓄積する速度よりも早く分解して再構築
- ・秩序を守るために絶えまなく壊して作り変える
- ・自らの制御化で平衡状態を作る

※1 W.B. Cannon, "The Wisdom of the Body", (1932) W.W. Norton & Co., New York.

※2 Schoenheimer, Rudolf. "The dynamic state of body constituents." The dynamic state of body constituents. (1946).

※3 福岡 伸一 (2007), 生物と無生物のあいだ, 講談社現代新書, 2007年5月.



細胞周期チェックポイント※1

- ・細胞が正しく細胞周期を進行させているか監視
- ・異常があった場合は周期進行を原則するなど制御
- ・細胞自身がこの制御機構を持っている

※1 細胞周期チェックポイント, [https://ja.wikipedia.org/wiki/](https://ja.wikipedia.org/wiki/%E7%B4%B0%E8%83%9E%E5%91%A8%E6%9C%9F%E3%83%81%E3%82%A7%E3%83%83%E3%82%AF%E3%83%9D%E3%82%A4%E3%83%B3%E3%83%88)

<https://ja.wikipedia.org/wiki/%E7%B4%B0%E8%83%9E%E5%91%A8%E6%9C%9F%E3%83%81%E3%82%A7%E3%83%83%E3%82%AF%E3%83%9D%E3%82%A4%E3%83%B3%E3%83%88>.



生命の特徴をWebシステムに応用

以下のように見立てる

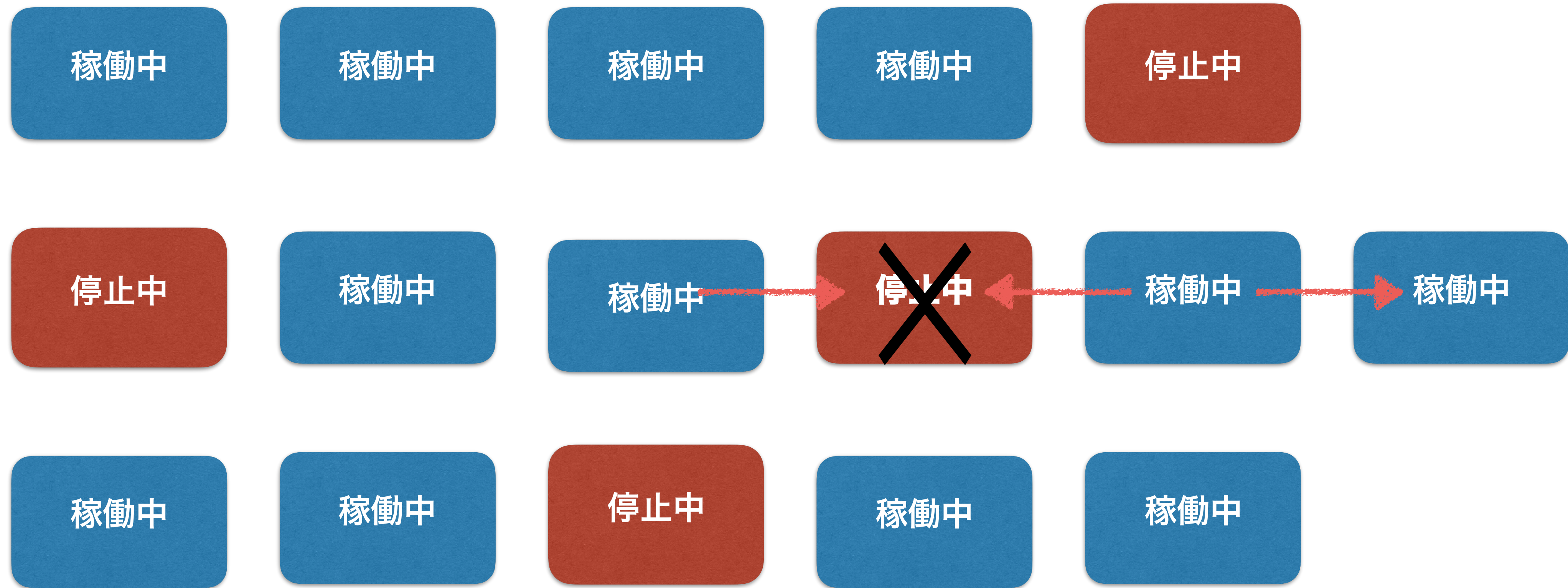
- ・細胞：サーバ（VM、プロセス）
- ・動的平衡：絶え間ないサーバの破壊と再構築
- ・エントロピー増大：異常状態のリスク増大
- ・細胞周期チェックポイント：サーバの自律制御



Webサービス基盤を生命のように扱う

- ・高度なシステムは生命のように見える
- ・コンピュータは長い間動かすと何かからしら異常となる
- ・異常（エントロピー増大）からは逃れられない
- ・自律制御しつつ自らを壊し増殖し再構築すれば良い
- ・動的平衡性により秩序を保つWebサービス基盤





常に12の「稼働中」が存在する前提で、「稼働中」が**任意の時間経過**で「停止中」になった場合、隣接の関係性の監視している「稼働中」が新たな「稼働中」を作り出し定常的に循環させる



VMにライブリバージョンやロケーション属性などを持たせて
VMの振る舞いがおかしいクラスタから異常な属性を抽出するクラスタリング監視

Speaker Deck Published on May 18, 2016

なめらかなシステムのアイデアと設計概要

生命の観点からWebシステムを解釈する

GMOペパボ株式会社
シニア・プリンシパルエンジニア
MATSUMOTO, Ryosuke
@matsumotory

2016/05/18 e-ZUKA Tech Night vol.37 なめらかなシステムをつくる ～AIはエンジニアの敵か味方か?～

◀ ▶ 🔍 share

なめらかなシステムのアイデアと設計概要 / namerakad-idea-design
by MATSUMOTO Ryosuke
Published May 18, 2016 in Technology

動的平衡性について詳しく知りたい場合の資料

https://speakerdeck.com/matsumoto_r/namerakad-idea-design



まとめ

まとめ

- ・人の不完全性やエントロピー増大を許容するシステム
- ・自律制御と動的平衡の系にあるWebシステム基盤
- ・システムそのものの状態や特徴を分類・監視する



今後の研究開発の予定

- ・STNS: プログラマブルな次世代Linux認証基盤
- ・haconiwa: プログラマブルな次世代コンテナエンジン
- ・Webサーバの自律制御
- ・動的平衡性を持つWebサービス基盤
- ・機械学習によるシステムの属性の分類と状態監視

