NTT DATA
Global IT Innovator

Hinemos Deep Dive
～ 運用アナリティクス編 ～

2018年4月11日

NTTデータ先端技術株式会社
南 温夫

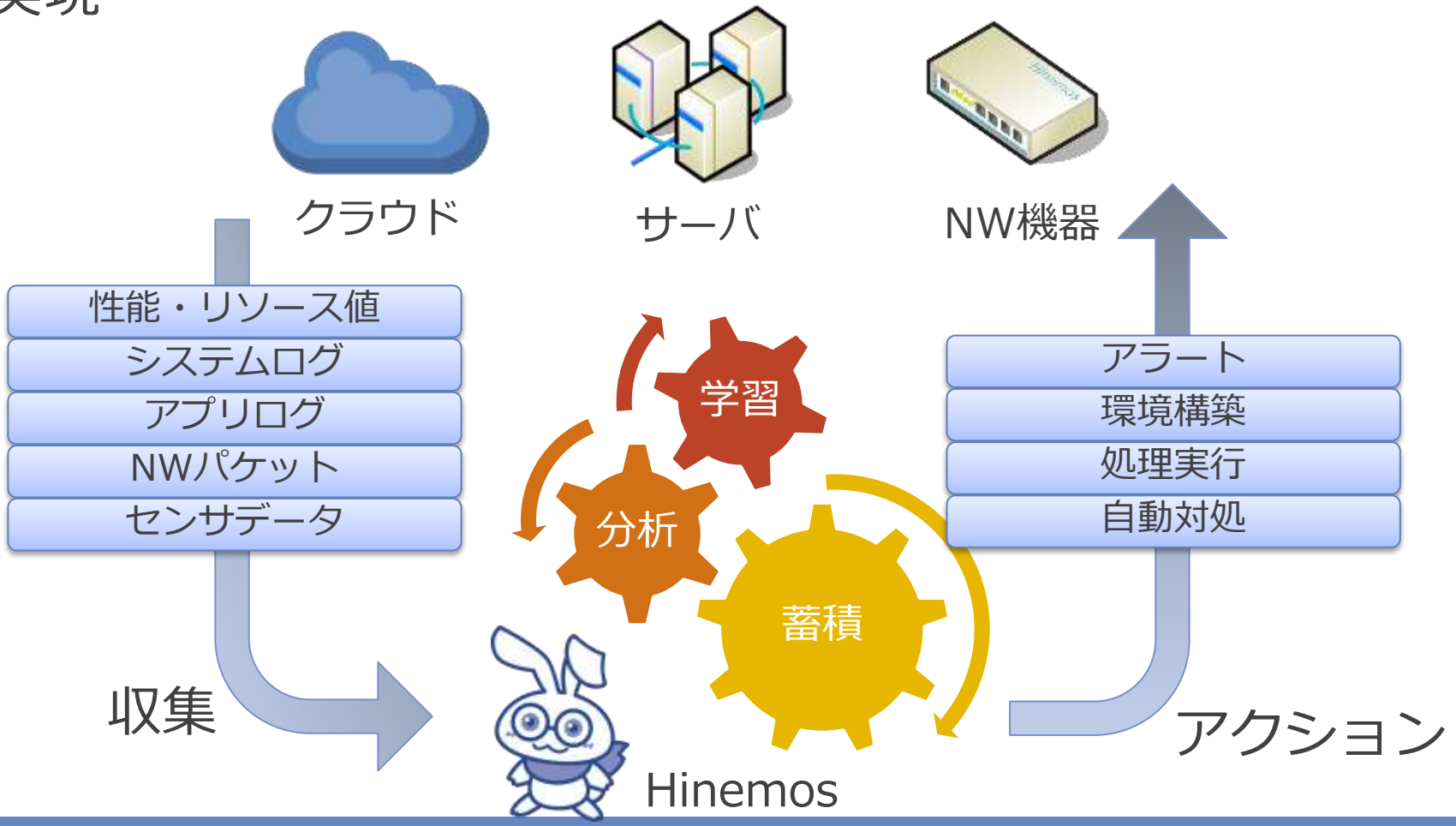
はじめに

本セッションでは、Hinemos ver.6.1で追加された
運用アナリティクスを活用するため、運用アナリティクスを
構成する各機能の詳細情報を深く説明します。

単一情報だけでなく、過去の蓄積情報や複数種別の蓄積情報
を活用する運用アナリティクスを構成する各機能をひとつひ
とつ解説します。

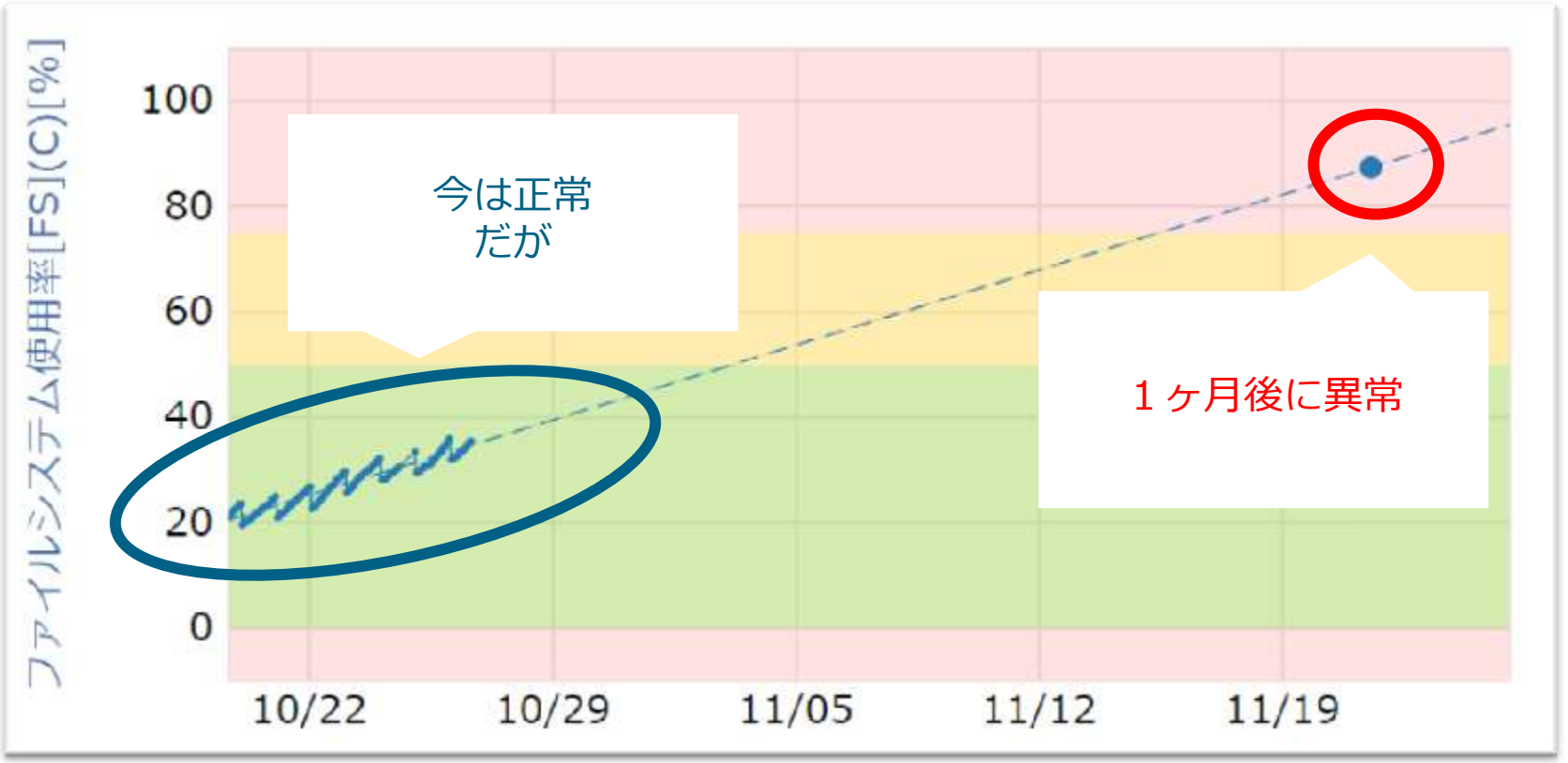
運用アナリティクスとは

Hinemosが収集・蓄積したデータをもとに
リアルタイムなシステム状態把握と**未来を予見した予防保全**
を実現



将来予測監視

将来予測監視



リソース枯渇はアラートが出てからだと手遅れです。
変化の多い環境でも、Hinemosが将来を予測し、いち早く対策を打てます。

将来予測監視

リソース値などは、時間経過で変化する時系列データです

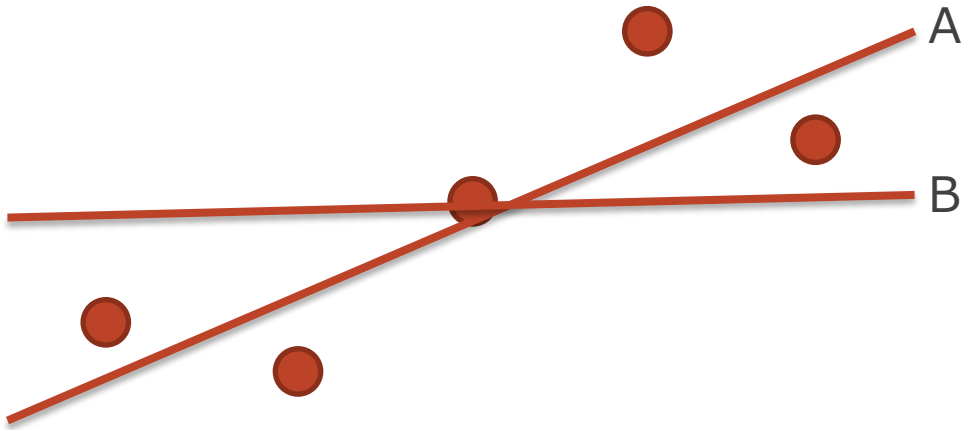
Hinemos 6.1では時系列データの分析における一般的な分析手法である回帰分析をおこない、収集値のトレンドを分析します

Hinemos 6.1は、線形回帰および二次と三次の多項式回帰に対応しています

線形回帰　二次多項式回帰　三次多項式回帰

将来予測

回帰分析ではトレンド線のモデルを作成し、それを元に将来の値を予測します。



線AとBのどちらが
トレンドとして適切か

トレンドはできるだけ誤差が小さいもの、すなわちできるだけ多くの点のそばを通るようなものが適切です。
これを求めるのが最小二乗法です。

最小二乗法

最小二乗法は、時刻t_iに得られた性能値がy_iのときに時刻t_1からt_nまでの値について、y_iと$f(t_i)$の差の二乗の総和が最小になるように$f(t)$のパラメータを求めることを目的とします。

$$\sum_{i=1}^{n}(f(t_i)-y_i)^2$$

例えば線形回帰の場合は、求めるモデルが

$$f(t)=a_0+a_1\cdot t$$

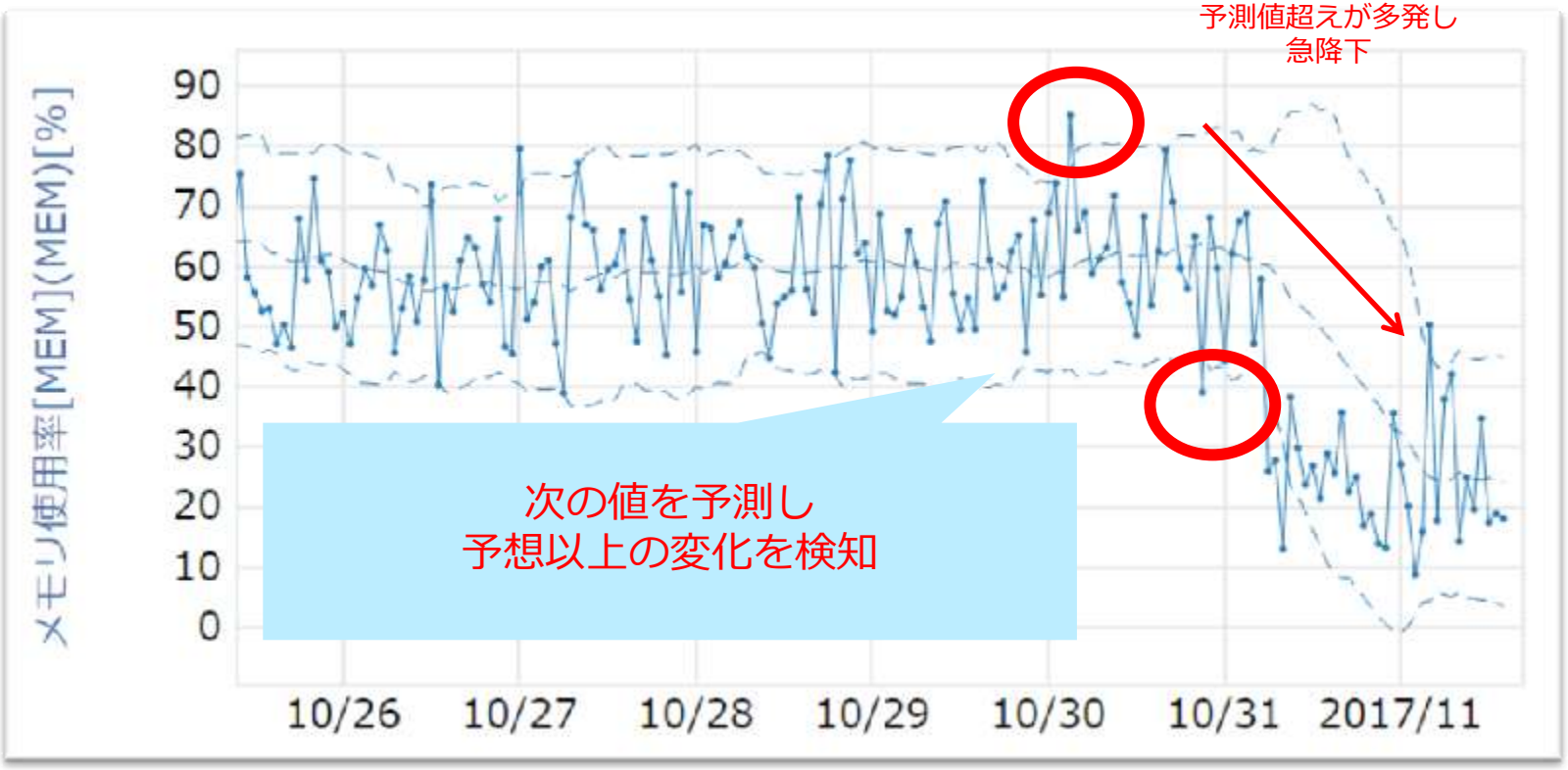
となるので、

$$\sum_{i=1}^{n}(a_0+a_1\cdot t_i\ -y_i)^2$$

が最小になるようなa_0, a_1を求めることでトレンドがわかります

変化監視

変化監視

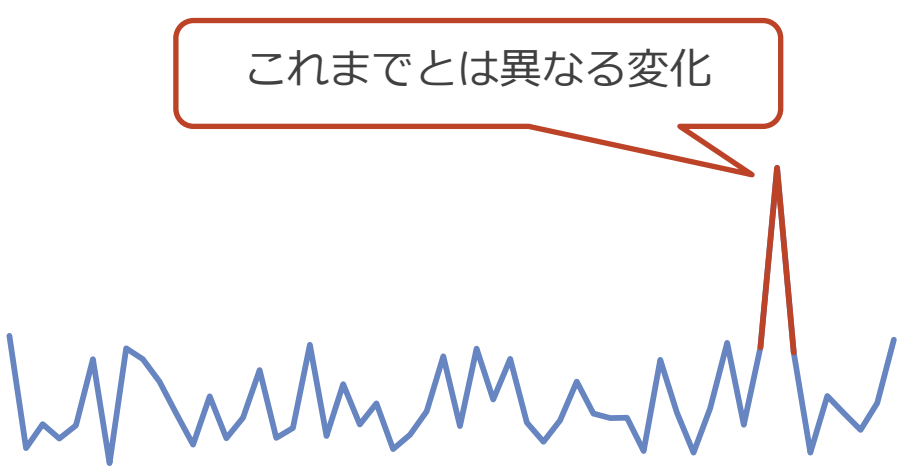


閾値内の変動でも普段と違う動きは異常の前触れかもしれません。
次の動きを予測し監視をすることで、いち早く異常の傾向を察知できます

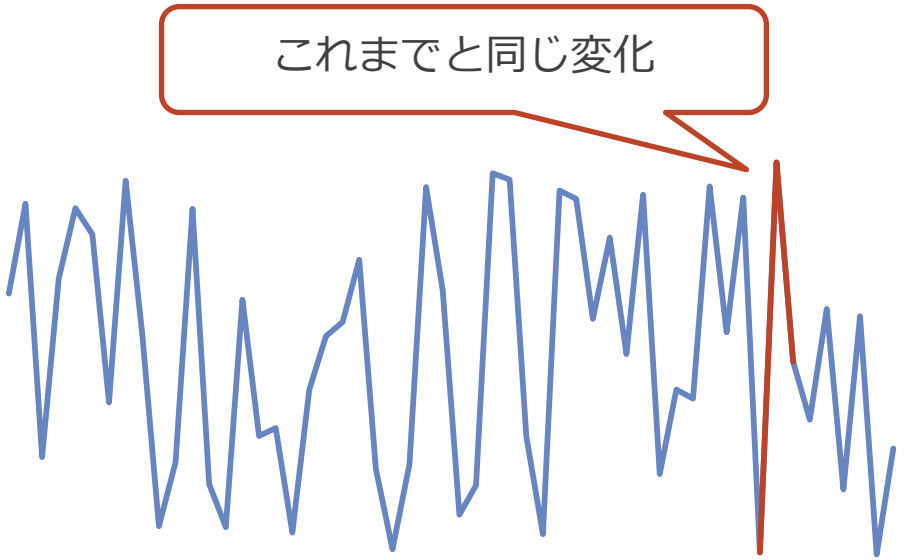
変化監視

値が平均値から離れた値をとった場合でも、これまでの値のちらばり方によって、それがこれまで通りの変化なのか、これまでと異なる変化なのかが決まります

このデータのちらばりを示す指標が、標準偏差です



ちらばりが小さい



ちらばりが大きい

変化監視

過去データがn個あり、その値が、x_1，x_2，x_3，…，x_nのとき

平均μは

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

標準偏差σは

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2$$

で求めることができます

この平均と標準偏差を元に、次の値がとりうる範囲とその確率を求めることができます

変化監視

一般的な確率分布である正規分布では、
データx_1，x_2，x_3，…，x_nによる平均μと標準偏差σをもとに
データx_{n+1}がとりうる値が以下の確率になることが
一般的に知られています

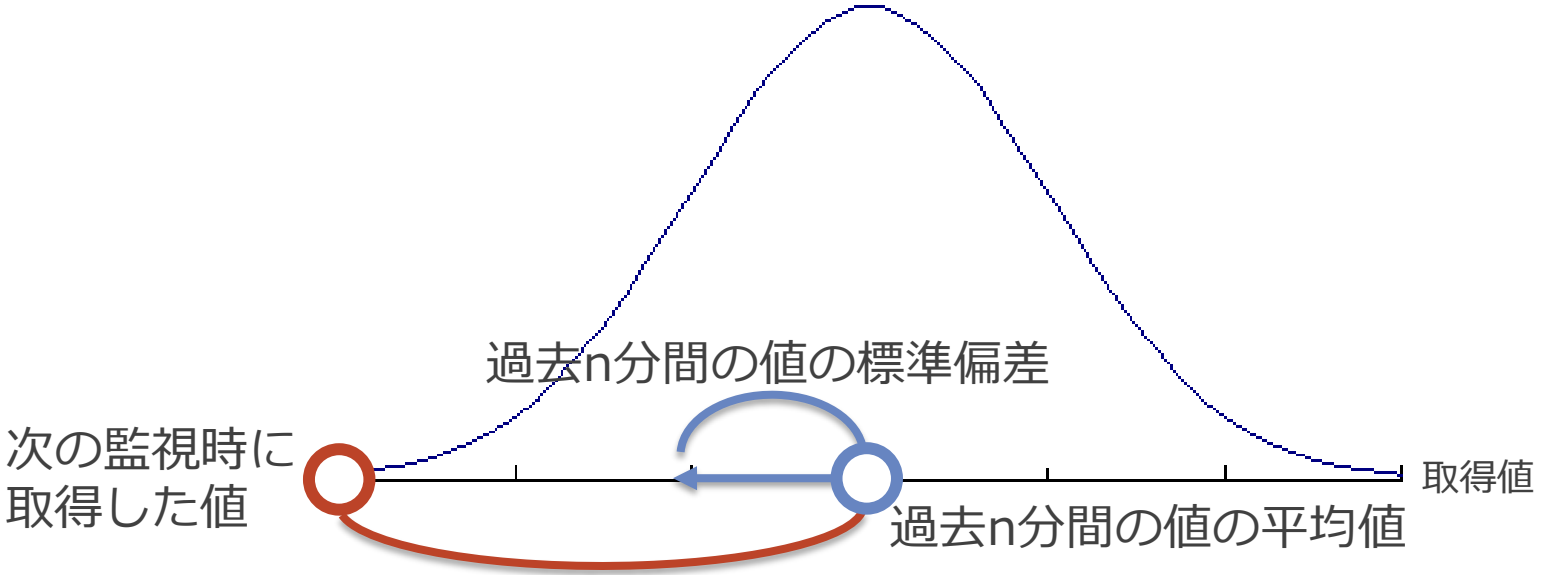
x_{n+1}の値の範囲	左の値になる確率
$\mu-\sigma \sim \mu+\sigma$	68.27%
$\mu-2\sigma \sim \mu+2\sigma$	95.45%
$\mu-3\sigma \sim \mu+3\sigma$	99.73%

これを元に次の値が平均から、標準偏差の何倍離れているか
見ることで、それがどれだけ特異な値なのかを判定できます

これはボリンジャーバンドアルゴリズムと呼ばれ、変化を検
知する一般的なアルゴリズムです

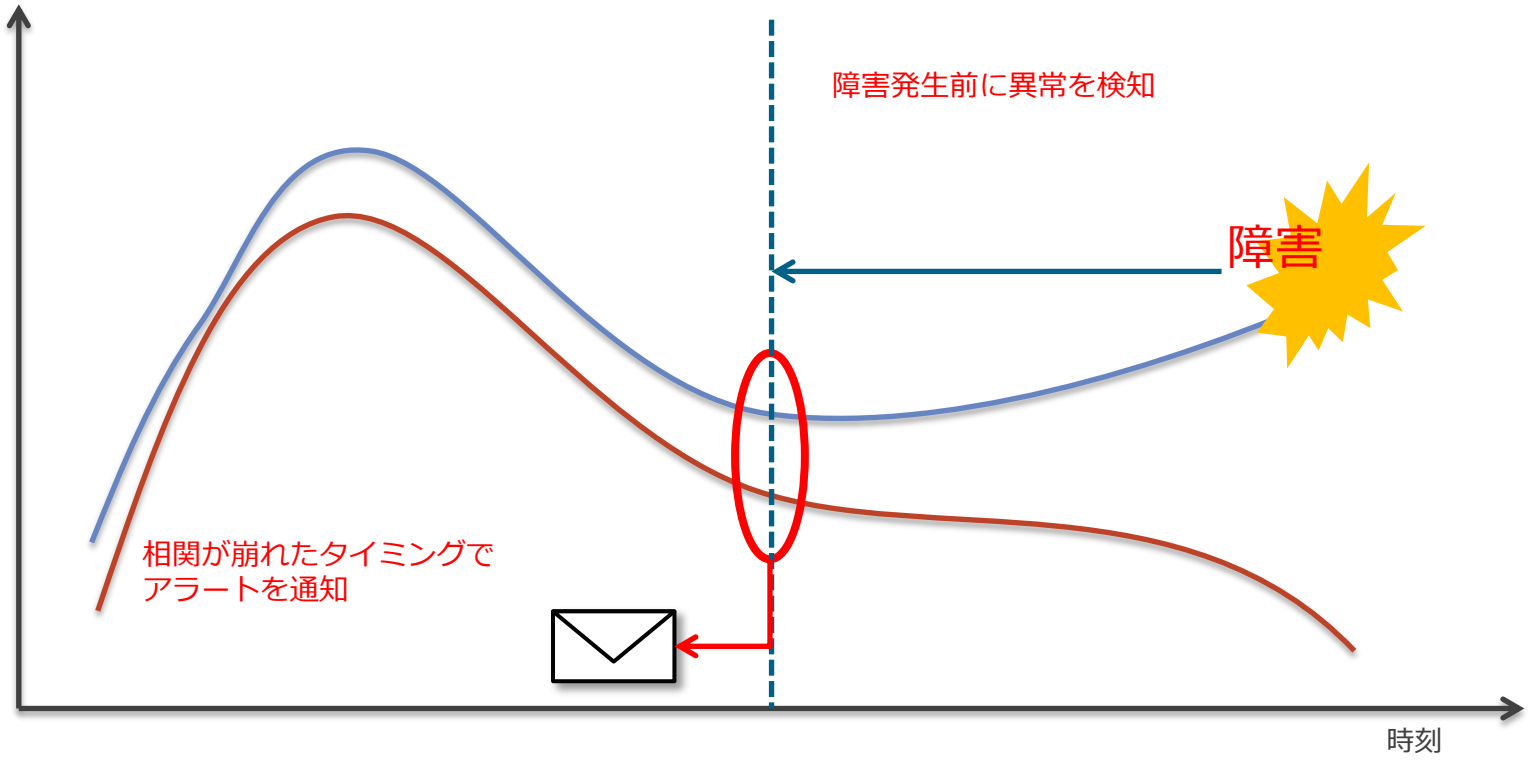
変化監視

Hinemosの変化監視は、判定の閾値として、
「標準偏差の何倍」離れているか、その倍数を指定します



相関係数監視

相関係数監視



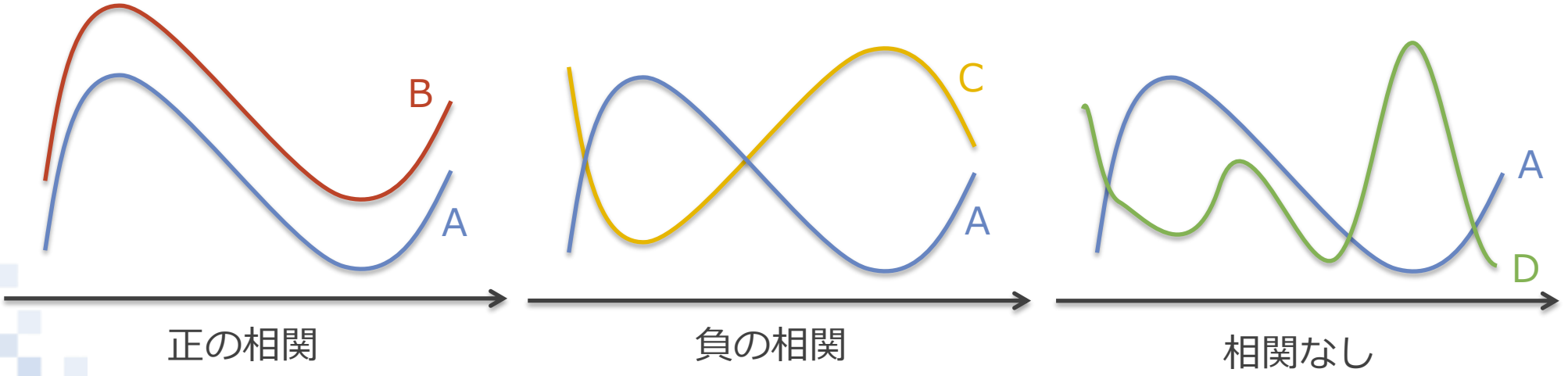
あるデータの異常は、他のデータとの関係性から気づける事があります。
相関係数を使って、関係性の崩れから将来的な異常をいち早く察知できます。

相関係数監視

システムの性能値には、ある性能値Aの値の上昇・下降にあわせて

- 性能値Aの上昇時に上昇、Aが下降時に下降する性能値B
- 性能値Aの上昇時に下降、Aが下降時に上昇する性能値C
- 性能値Aの変化とは関係なく変化する性能値D

があります。
性能値AとBの関係を「正の相関」、AとCの関係を「負の相関」、AとDの関係を「相関なし」と呼びます



相関係数監視

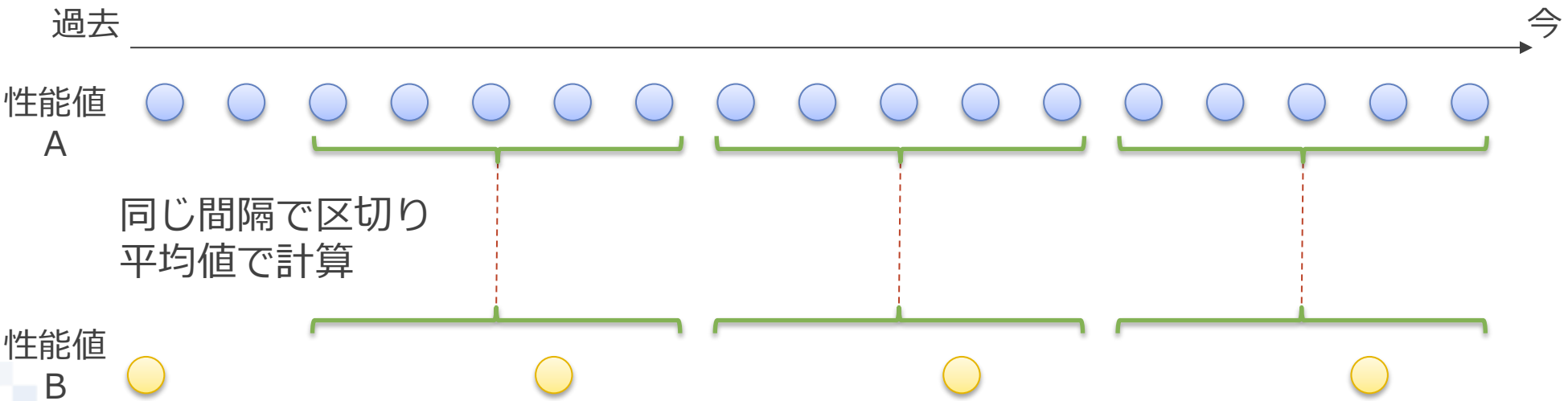
2つのデータの相関がどれだけあるかを示す指数を相関係数と呼び、時刻t_iにx_iとy_iが取得される場合、
データ$x_1,\ x_2, \ldots, x_n$と$y_1,\ y_2, \ldots, y_n$の相関係数 r は以下の式で求めることができます($\bar{x}$は$x_1,\ x_2, \ldots, x_n$の平均を指します)

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

このrは、-1から+1までの値をとり、
rの絶対値が大きいほどxとyは関係性があるといえます

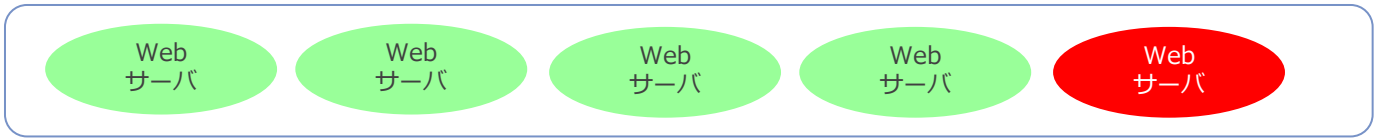
相関係数監視

相関係数を求めるためには同じタイミングのデータを取得する必要があります
Hinemosでは監視設定ごとに監視間隔が異なるものにできるため、相関係数監視の監視間隔で過去のデータを分割し、その平均値を相関係数の計算に利用します。

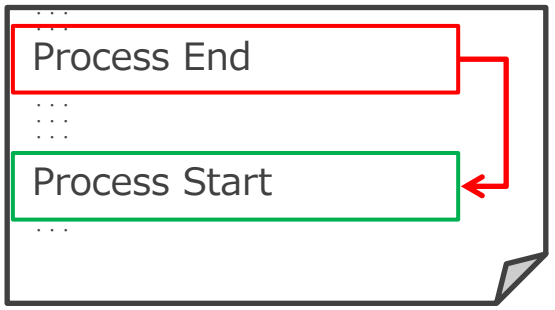


収集値統合監視

収集値統合監視



一台の障害は**警告**レベル
全台の障害は**危険**レベル



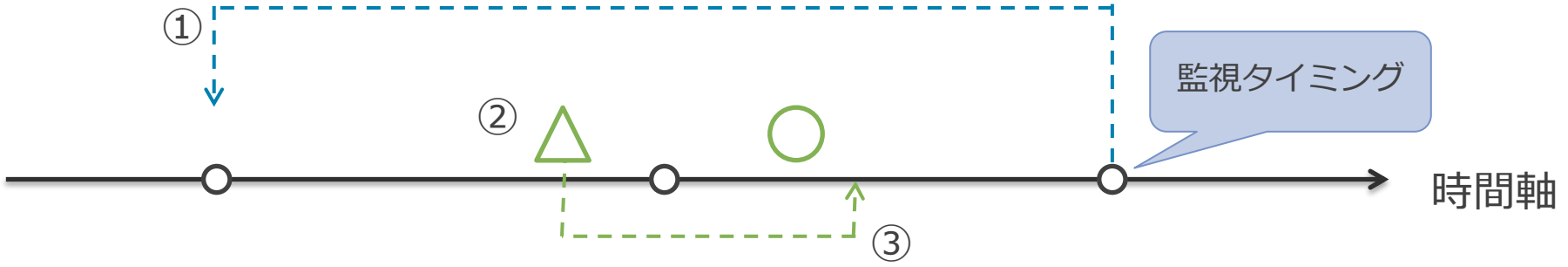
アプリケーションログ

プロセスが停止しても
時間内に再起動すれば
正常レベル

1つのデータでは、重要度を正しく表現することが出来ない場合があります。
複数のデータを組み合わせた監視により障害のレベルを正確に把握できます。

収集値統合監視

収集値統合監視では、以下のアルゴリズムで判定をおこないます
例) 判定条件△○が順番に5分以内に出力されればOK



① 各監視タイミングで監視間隔の2倍分さかのぼり、その期間にひとつ目の判定条件を満たすものがあるか検索する
② 条件を満たす場合、③へ続きすべての条件を満たすか判定に進む。ひとつ目の条件を満たさない場合はOKもNGも通知しない
③ ひとつ目の条件を満たした場合は、そこからタイムアウト期間内にすべての条件を満たすものがあるか検索する。全て満たせばOK、ひとつでも満たさなければNGとして通知される

バイナリファイル監視

バイナリファイル監視

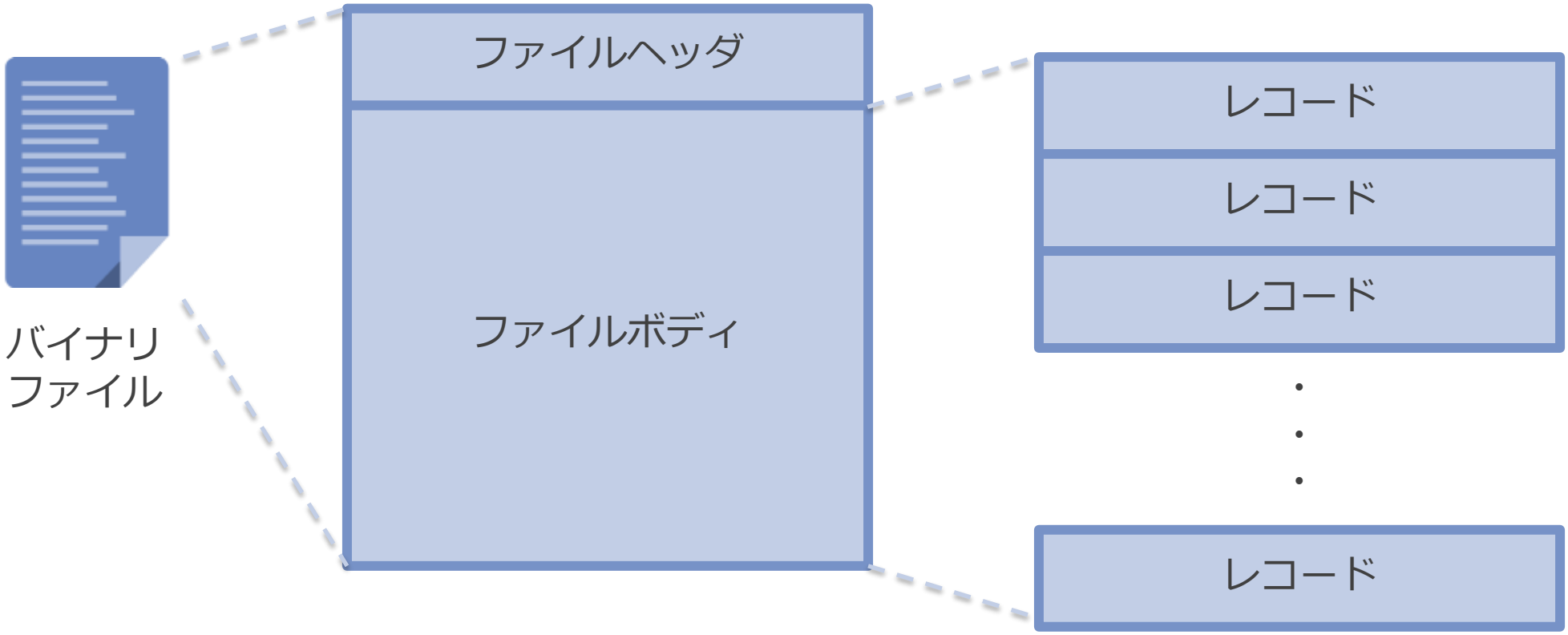
バイナリファイル監視では、監視・収集の単位を、
大きくわけて以下の3つから選択することができます

- ファイル全体
- レコード単位の増分
- 時間区切りの増分

時間区切りの増分は、定期間隔でその間に出力されたものを
増分とみなすため、ファイル構造を意識しません

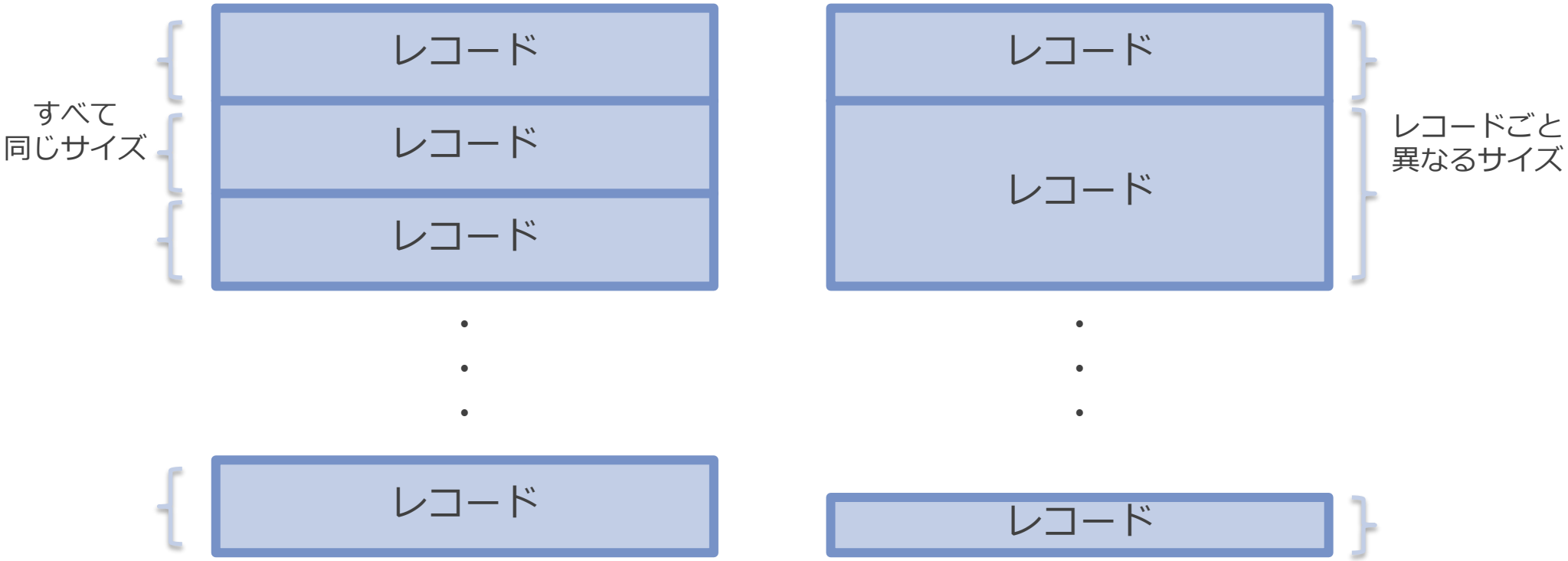
ファイル全体、レコード単位の増分の場合は、
対象のフォーマットを指定することで、バイナリファイルに
追記される個々のレコード単位で監視・収集することが可能
です

バイナリファイルの構造



バイナリファイルは、ファイル自体の情報をもつファイルヘッダと
個々の情報であるレコード群からなるファイルボディから構成されます
ファイルボディを構成するレコードがどのように区切られるかをHinemosでは
指定可能です

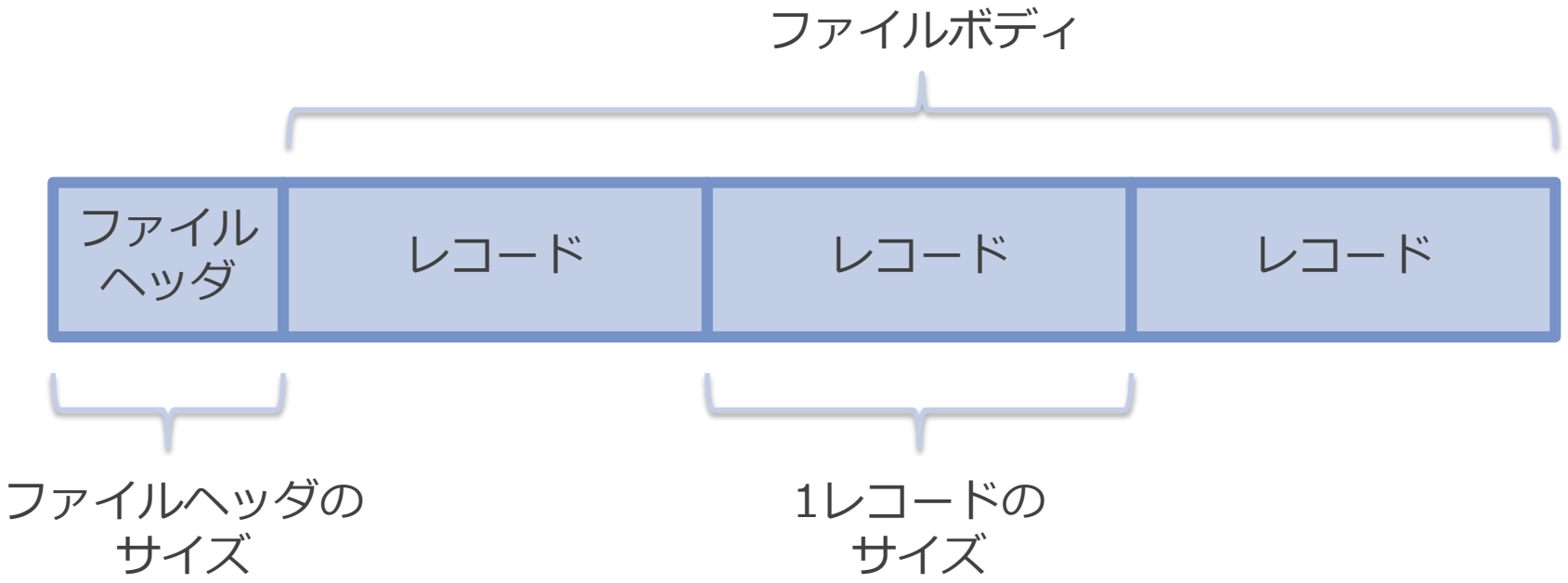
固定長と可変長



バイナリファイルのレコードは、すべて同一の長さである固定長と
レコードごとに流さが異なる可変長の2種類があります
Hinemosでは、このどちらにも対応しています

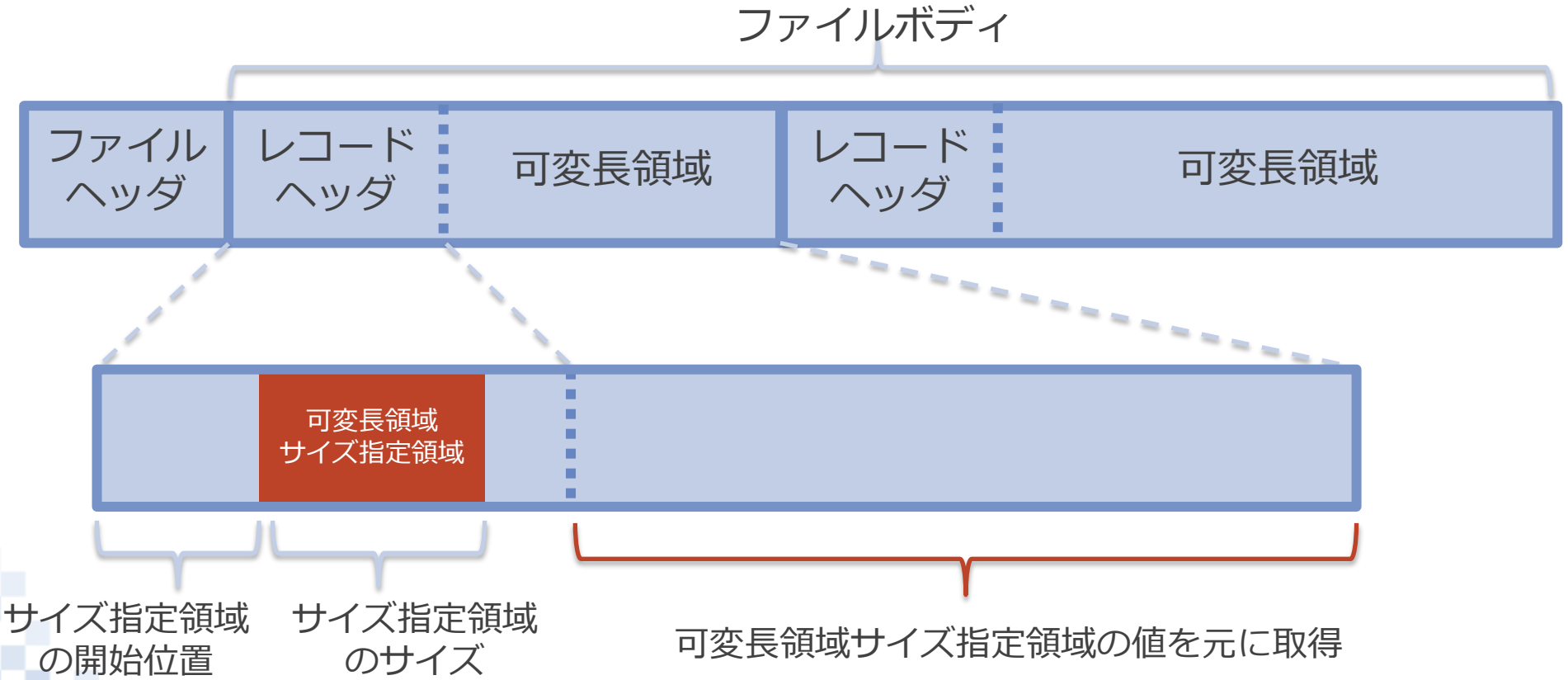
固定長の場合

固定長の場合はファイルヘッダのサイズと1レコードのサイズを指定することで、レコードごとに分割し収集します



可変長の場合

可変長の場合はレコードがさらにレコードヘッダと可変長領域に分かれます
レコードヘッダの特定の位置に可変長領域のサイズを示す領域があるので、その領域の位置を指定します



タイムスタンプ

バイナリファイルのレコードにはタイムスタンプを含むものがあります
Hinemos ver.6.1のバイナリファイル監視は、以下の2種類のタイムスタンプに対応します

- UNIX時間(4バイト)
- UNIX時間(4バイト) + マイクロ秒(4バイト)

※UNIX時間は、協定世界時(UTC)での1970/1/1 00:00:00からの経過秒(エポック秒)です。

これを読み取るため、レコードの先頭からの開始位置を指定します。

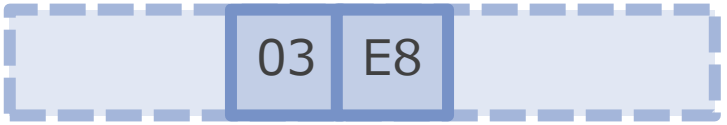
タイムスタンプの開始位置を指定

ビッグエンディアンとリトルエンディアン

バイナリの格納方式には、ビッグエンディアンとリトルエンディアンがあり、どちらの形式にも対応しています。

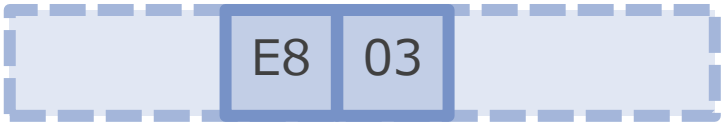
例えば、10進数での「1000」は、16進数にすると0x03E8となりますが、それぞれ以下のようにファイル上格納されます。

- ビッグエンディアン



正順で格納される

- リトルエンディアン

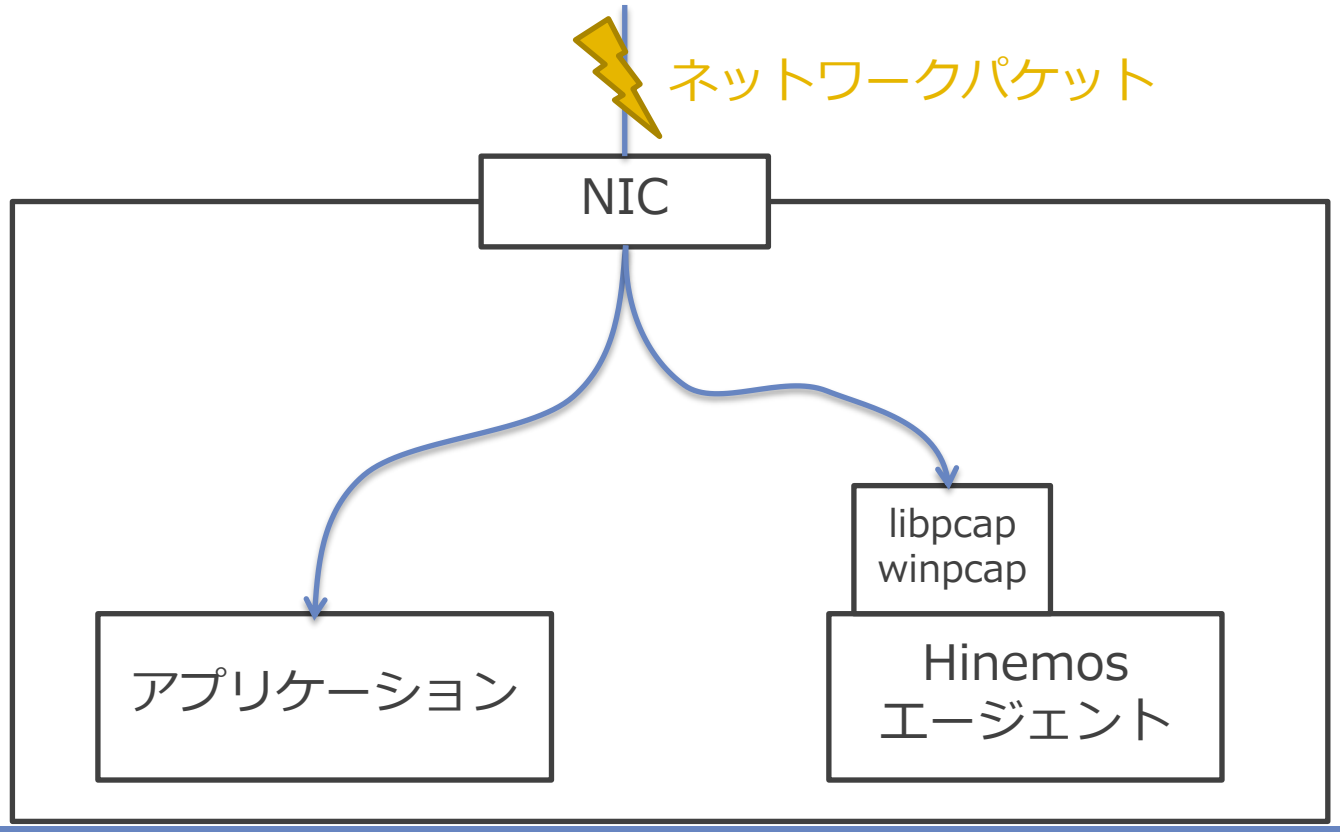


逆順で格納される

パケットキャプチャ監視

パケットキャプチャ監視

パケットキャプチャ監視では、pcapを実装したライブラリ(libpcap/winpcap)を活用し、HinemosエージェントがNICに到達したパケットを監視・収集します。



パケットキャプチャ監視

全てのネットワークパケットを取得するだけでなく、必要なパケットにしぼって監視・収集することが可能

フィルタにはパケットキャプチャにおける一般的な表現であるBPF(Berkeley Packet Filter)記法を利用

例:
192.168.0.1からの通信のみを収集
src host 192.168.0.1

TCP:80の通信のみを収集
tcp port 80

まとめ

本セッションでは、Hinemos ver.6.1で追加された、
以下の運用アナリティクスを構成する各機能を詳細に説明しました。

・将来予測監視
・変化監視
・相関係数監視
・収集値統合監視
・バイナリファイル監視
・パケットキャプチャ監視

各機能の詳細を理解し、運用データを正しく活用しましょう。

NTT DATA

Global IT Innovator