

システムを遠隔でトレースする Web アプリケーションの開発 —ログ収集基盤上で動作するメトリクス管理システム—

1. 背景

サーバリソース(CPU 使用率、メモリ使用量やディスク I/O 等)の収集・可視化・監視を行う為のツールとしては、Zabbix, Nagios, Sensu といったツールが使われてきた。これらのツールでは、メトリクスの収集は各ツール独自のエージェント(メトリクス収集を行うサーバ上で動作し、情報を収集してサーバリソース管理サーバまで送信するプログラム)により行われる。

サーバリソース監視ツールとは別のアプローチとして、近年では汎用のログコレクタである fluentd を用いてサーバから情報を収集・集約し、集約したログを Graphite や GrowthForecast といったグラフ描画ツールで可視化して分析や監視に活用する事例が増えてきている。

fluentd とグラフ描画ツールを組み合わせた可視化に対して、バックエンドに検索エンジンを用いる事でより高機能な可視化と分析を可能にするシステム構成も試みられた。そのような事例の1つが fluentd に Elasticsearch と、Elasticsearch の公式フロントエンドである Kibana 3 を組み合わせた構成である。

2. 目的

fluentd と Elasticsearch と Kibana 3 を組み合わせたシステム構成におけるメトリクスの収集・可視化・テストの 3 段階を一元的に管理し、各段階の設定を 1 つの単位として容易にインポート/エクスポートできる Web アプリケーションを開発する。

メトリクスの収集を一元的に管理する為の機能として、多数台のサーバ上で fluentd の設定を既存の設定と干渉しない形で同期・適用する為の機構を開発する。メトリクスの可視化には、Elasticsearch の公式フロントエンドである Kibana 3 を使用する。また、Elasticsearch 上に保存されたメトリクスをテストする為のツールとして、メトリクスが満たすべき仕様を可読性の高い形で記述できるテストフレームワークを開発する。

これら 3 機能の設定は、バージョン管理システムである git を用いてソースコード共有サイトである GitHub や、リモートのリポジトリへとインポート及びエクスポートを行えるようにする。

3. 開発の内容

本プロジェクトで開発したシステム Andon の全体構成を図 1 に示す。開発した各コンポーネント(図 1 の赤で囲ったもの)はそれぞれ以下のような機能を持つ。

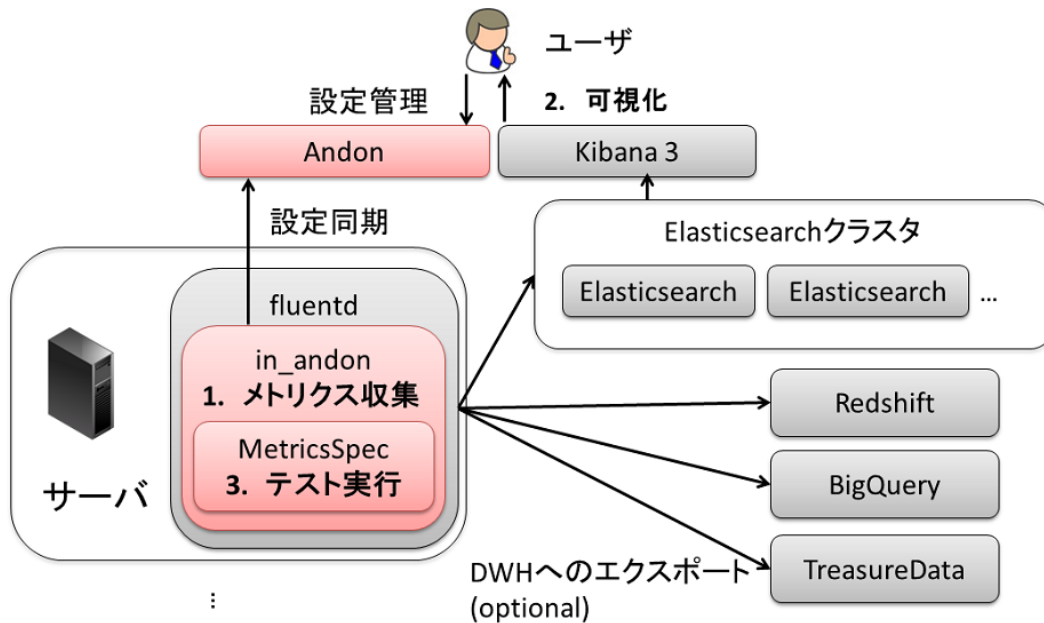


図 1 全体のシステム構成

- Andon (Web インターフェース)

Andon は、ユーザに対してメトリクス収集・可視化・テストに必要な設定を編集・管理する機能や、設定をリモートの git リポジトリ等と共有する機能、メトリクスのテスト結果を一覧する機能を提供する他、後述の in_andon に対して設定を配信する機能を持っている。
- in_andon

fluentd 上でプラグインとして動作し、メトリクスの収集とテストを行う。一定間隔で Andon と設定を同期し、変更があれば適用する。
- MetricsSpec

Elasticsearch 上に保存されたメトリクスが満たすべき仕様を記述し、テストを実行できるテストフレームワークである。Ruby 製のテストフレームワークとして広く使われている RSpec の拡張として記述されている。

Andon の Web インターフェースでは、fluentd と Elasticsearch と Kibana 3 を組み合わせたシステム構成におけるメトリクスの収集・可視化・テストの設定を Role という単位にまとめて一元的に管理する事ができる。

- メトリクス収集・可視化・テスト設定の管理機能

ユーザは Web インターフェースから fluentd 及び MetricsSpec の設定を変更する事ができ、行われた変更は後述する in_andon によって各サーバに同期される。
- 設定のインポート・エクスポート機能

前述の設定管理画面から編集した設定は、ソースコード共有サイトである GitHub 等へエクスポートが可能である。設定を作成したユーザは、簡単な操作で Role を共有で

き、新たに Andon を使いはじめるユーザは既存の Role をインポートするだけですぐに Andon の元でメトリクスの管理を始める事が可能である。

- Role 別、サーバ別のメトリクスのステータス表示機能
収集したメトリクスが MetricsSpec で記述された仕様を満たしているかどうかを Role 別・サーバ別に一覧する事ができる。

in_systemtap は、Linux カーネルやユーザランドで動作するプロセスを動的にトレースする機構である SystemTap を使って、メトリクスを収集する fluentd の入力プラグインである。SystemTap から LTSV (Labeled Tab-separated Values) フォーマットで出力された情報をログとして取得できる。in_systemtap は任意の SystemTap スクリプトを実行できる他、プラグインに同梱されている SystemTap スクリプトライブラリの中から必要なスクリプトを選んで実行する事もできる。

in_andon は、メトリクス収集を行うサーバ上の fluentd 内でプラグインとして動作し、メトリクスの収集とテストを行う Andon のエージェントである。

- fluentd の子プロセスによるメトリクスの収集
in_andon は、Andon サーバから設定を取得してメトリクスの収集を行う。設定は fluentd の設定ファイル形式で記述されており、in_andon は fluentd 内に子プロセスとしてもう1つの fluentd プロセスを立ち上げ、取得した設定を適用してメトリクスの収集を行う。親の fluentd とは別のプロセスでメトリクスの収集を行う事により、親の fluentd が行うログ収集と互いに干渉する事なく動作する事ができる。
- メトリクス収集設定の同期
in_andon は一定の周期で Andon サーバに問い合わせを行い、ユーザが設定に変更を加えていないか確認する。初回の問い合わせ時に、in_andon は自身に関する情報 (ホスト名や設定された Role など) を送信し、Andon サーバに自動で登録されるので、ユーザはサーバを追加する度に登録作業を行う必要はない。
- MetricsSpec の実行
メトリクスのテストツールである MetricsSpec を in_andon 上で実行できる。実行結果は Andon サーバへの定期的な問い合わせ時に送信される他、ログとして fluentd 上で扱う事もできるので、fluentd の豊富な出力プラグインを用いて実行結果の保存や通知を行う事が可能である。

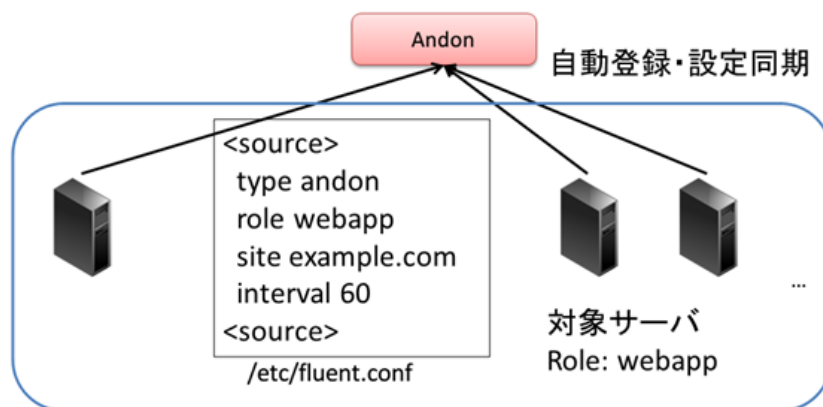


図 2 Andon と複数のサーバの同期

MetricsSpec は、メトリクスのあるべき状態（仕様）を可読性の高い形式で記述して、Elasticsearch に蓄積されたメトリクスが仕様を満たしているかどうかをテストできるツールである。Ruby 製のテストフレームワークである RSpec の拡張として記述されているので、RSpec の経験があるプログラマであればプログラムのテストコードを書く時と同じ要領でメトリクスの仕様を記述し、テストする事ができる。以下に MetricsSpec で記述したメトリクスのテストコードの例を示す。

```

1 describe "cpu usage" do
2   let(:cpu_usr) { metrics("@tag:dstat.cpu_usr").within(1.hour)["value"] }
3   let(:cpu_sys) { metrics("@tag:dstat.cpu_sys").within(1.hour)["value"] }
4
5   it "%usr + %sys should be less than 90" do
6     expect(cpu_usr.avg + cpu_sys.avg).to be < 90
7   end
8 end

```

図 3 MetricsSpec で記述したメトリクスのテストコード

4. 従来の技術（または機能）との相違

- Zabbix, Sensu などの監視ツールに対して

Zabbix や Sensu といった既存のサーバ監視ツールは、専用のエージェントを各サーバにインストールする必要があるが、Andon はログコレクタとして広く用いられている fluentd の基盤上で動作する為、専用のエージェントは不要である。また、既存のツールではサーバ間の通信にキュー等のミドルウェアが必要な場合があるが、Andon ではその必要がない。
- fluentd とグラフ描画ツールを組み合わせたシステムとの差異

既存のグラフ描画ツールではバックエンドをスケールアウトさせる事が難しく、管理するサーバ台数が増えて負荷が増大した場合に対策が取りづらいという難点があったが、Andon の構成でバックエンドとして利用する Elasticsearch は複数台でクラスタを組んで動作させる事が前提となっており、負荷の増大時にも台数を増やす事で容易にスケールアウトさせる事ができる。また、既存のグラフ描画ツールでは時系列グラフ以外の形式での可視化が難しかったが、Andon では Kibana 3 の機能を利用して円グラフや地図、統計情報などの高度な可視化が可能である。さらに、fluentd とグラフ描画ツールを組

み合わせたシステムでは集めたメトリクスをテストする方法がなかったが、Andon には MetricsSpec という可読性の高いテストフレームワークがあり、可読性の高いテストコードでメトリクスが仕様を満たしているかどうかをテストする事ができる。

5. 期待される効果

Andon によってログ収集基盤上でのメトリクス収集が簡単に行えるようになり、既存のツールで運用上の負担となっていた監視対象サーバへのエージェントのインストールや、通信用のミドルウェアの管理の負担は少なくなると考えられる。また、メトリクスの収集には fluentd の豊富な入力プラグインを活用できるので、収集できるメトリクスの種類が増え、より高度な分析や監視が可能となる。

また、今回開発したメトリクスのテストフレームワークである MetricsSpec は、メトリクスのあるべき状態を可読性の高い形式で記述でき、単にアラートを上げる為の条件式としてではなく、運用上満たされるべきメトリクスの仕様を記述及びテストできるツールとして活用する事ができる。

加えて、これまで fluentd とグラフ描画ツール等の組み合わせにおいてはエンジニアがそれぞれの環境に合わせて必要なコンポーネントを設定する必要があったが、Andon ではメトリクスに必要な収集・可視化・テストに関する設定を一元的に管理でき、作成した設定は Web 上のリポジトリ共有サイト等に公開できる。公開された設定を他のエンジニアが適用する事によって、有用な設定の共有が促進される事が期待される。

6. 普及(または活用)の見通し

Andon をオープンソース・ソフトウェアとして公開し、普及を図っていきたい。その為には、Andon でどのような事が実現できるか理解してもらう為のサンプルとなるような Role を作成し、それを使ってもらう事が重要であると考えている。また、各設定を簡単に共有できる点が Andon の特徴であるので、ドキュメントやクイックスタートガイド等を整備してユーザにも Role を作成し、共有してもらえたい。

7. クリエータ名(所属)

松下 正樹(エヌ・ティ・ティ・コミュニケーションズ株式会社)

(参考)関連 URL

成果物のソースコード:

- Andon: <https://github.com/Glasssaga/andon>
- in_systemtap: <https://github.com/Glasssaga/fluent-plugin-systemtap>
- in_andon: <https://github.com/Glasssaga/fluent-plugin-andon>
- MetricsSpec: <https://github.com/Glasssaga/metrics-spec>