

IoT データの収集蓄積可視化のデモシステム構築

清家 巧[†]
TIS[‡]

吉見 真聡[‡]
TIS[‡]

1 はじめに

無線伝送技術と組み込み機器の普及と発展に応じて、モノのインターネット (IoT: Internet of Things) をキーワードに、センシングと情報収集の技術の研究開発が進められている [1]. データ収集のためには収集や蓄積を行うだけでなく可視化するシステムも重要である. クラウドサービス等でその機能を提供するものもあるが, データ構造や接続方法に一定の制限を受けてしまう. そこで本稿では, オープンソースのソフトウェア (OSS) を組み合わせることで, データ形式やデータ転送方法にある程度自力で拡張可能なデモシステムを構築したのでその報告を行う.

2 デモシステムの要件

デモに用いるためのシステム構成を以下のような要件を設定した. 以下の機能要件は過去に類似のシステムを作った際に課題となった内容を基に作成した.

- Web ブラウザで参照可能なクラウドで動作するシステムであること
- 想定するセンサーは 20 データ/秒程度
- リアルタイムにデータ (1 秒以内程度) が可視化できること
- 受信するデータフォーマットに制限がない事, バイナリ値を扱えること

- 過去の情報が参照可能であること
- ダッシュボード (複数の情報を一画面で俯瞰できる機能) を持つこと
- データ可視化の外観やクエリを設定可能であること

多くの非機能的な要件はターゲットとしない. IPA の非機能要求グレード 2018 [2] によると, 「可用性, 性能・拡張性, 運用・保守性, 移行性, セキュリティ, システム環境・エコロジー」が非機能要求である.

3 デモシステムの構成

システムはデータ受信部, データ蓄積部, データ可視化部, 大きく 3 つの機能で構成する.

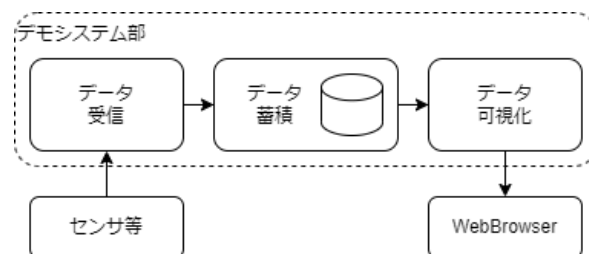


図1 デモシステム構成とデータの流れ

3.1 データ受信部

データ受信部の中心は Kafka[3] で実現している. Kafka の broker に対して Kafka クライアントで接続し, データを投入することでセンサー装置や計測値について Kafka のプロトコルでデータを収集する.

Kafka クライアントを導入できない装置および環境については, mqtt で送受信し, Kafka に投入する proxy や, Kafka の rest クライアントを

Building a demonstration system for collecting, accumulating, and visualizing IoT data

[†] SEIKE Takumi, TIS Inc.

[‡] YOSHIMI Masato, TIS Inc.

用いてデータ投入を可能にした。

また、通常は可用性やスループット向上のために複数の broker を稼働させるが、要件に設定していないため単一構成とした。

3.2 データ蓄積部

データ蓄積部の主要なコンポーネントとしては Kafka Stream と Kafka Connect と PostgreSQL(RDB) がある。

Kafka broker に到着したデータに対して、Kafka Stream で即座に処理を実行できる。この機能を用いて、到着したデータに対して任意の変換などの処理が実施可能となり、送受信するデータフォーマットに制限がなくなった。

処理できる事項は、任意のプログラムによる処理、または、ksql と呼ばれる SQL ライクな問い合わせ言語での Kafka Stream に対する選択、集約関数の適用である。バイナリが到着した際には、受信データやメタデータを元に構造データへの変換を試みる。

さらに到着データに一元的にタイムスタンプを付与することで、複数のデータソースを、同じタイムスタンプを軸にデータ可視化に用いることができる。

Kafka Connect は RDB 等の他のデータソースと Kafka Stream の間でデータの即時連携を実現するための仕組みである。データは可視化のために、構造データを postgresql に蓄積する。

3.3 データ可視化部

データ可視化部の主要なコンポーネントとしては Grafana[4] を用いた。Grafana は Web ブラウザ上でデータ分析及びインタラクティブな可視化を提供するソフトウェアである。

Grafana のもつダッシュボード作成機能は、複数のデータソースを複数のパネルで可視化できる任意の時間範囲や現在時刻等を起点とした相対的な時間範囲を対象とできる、自動更新によるリアルタイム画面更新、と、要件を達成するのに非常に強力なツールであった。

特に 1 秒以内のリアルタイム表示のために

は、データの時間軸に対する差分取得の能力を有すべきだが、Grafana はデータの可視化のための SQL クエリの中でデータを差分的に取得する機能を持ち、この点によって性能問題が発生しなかったと考えている。

4 デモシステムの評価結果

3 章で示したシステム構成に対し、以下の評価を行った。

- 20 データ/秒でバイナリデータを送る実験用センサを作成し可視化基盤での確認
- 1 データを投入し可視化画面で確認できるまでの時間を手作業で計測
- ダッシュボード機能による複数のセンサ情報の可視化

上記の項目で実施し、要件が実現できていることを確認した。機械的な計測を行えていないため、人による計測である。画面のリフレッシュ頻度による画面表示への反映遅延が支配的であった。

5 まとめと今後の展開

本報告では、IoT 機器のデータ収集蓄積可視化をリアルタイムに実現できるデモシステムの構築を行い、要件を満たすシステムを構築した。データ投入から可視化の遅延時間を計測するめい

参考文献

- [1] Jie Lin, et al. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp. 1125–1142, Oct. 2017.
- [2] システム構築の上流工程強化（非機能要求グレード）: ipa 独立行政法人情報処理推進機構. <https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>. (Accessed on 12/26/2022).
- [3] Apache kafka. <https://kafka.apache.org/>. (Accessed on 2023-01-12).
- [4] Grafana: The open observability platform — grafana labs. <https://grafana.com/>. (Accessed on 2023-01-12).