

AWS IoT を用いたセンサアプリケーションの開発方法の評価

1210379 宮尾 将史 【分散処理 OS 研究室】

1 はじめに

近年, IoT の普及により様々なモノがインターネットに繋がるようになってきており, 各デバイスから集められた膨大なデータを処理するためのプラットフォームの需要が高まってきている. そのため, 拡張性に優れたクラウド技術を用いることによって, 開発コストが抑えることが期待される. 本研究では, Amazon Web Services(以降:AWS と略す) が提供する AWS IoT Core を用いた土砂滑り検知システムの開発を通して, センサアプリケーションの開発方法の評価を行う.

2 AWS IoT Core を用いた開発

2.1 AWS IoT Core

AWS IoT Core はインターネットに接続されたデバイスを各種 AWS のサービスや他のデバイスに接続することで, データと通信を保護し, デバイスデータに対する処理やアクションを実行する事を可能にする AWS のサービスである [?].

2.2 センサアプリケーションの開発

AWS を用いたセンサアプリケーションの構成を図1に示す. IoT デバイスは AWS IoT Device SDK for Python を使用することで AWS IoT Core と接続を可能とする. また, AWS IoT Core との間の通信プロトコルは MQTT 通信を使用し, IoT デバイスから送られてきたデータを IoT topic で認証とアクセス管理を行う. 受け取ったデータは IoT rule で設定したルールアクションに従い DynamoDB にデータの格納を行う. DynamoDB との接続には IoT rule 上でルールクエリステートメントで SQL に似た構文を記述する必要がある. 本研究で用いたルールクエリステートメントを以下に示す. AWS を用いるとサーバ側の処理のほとんどを UI を用いて構築することができる.

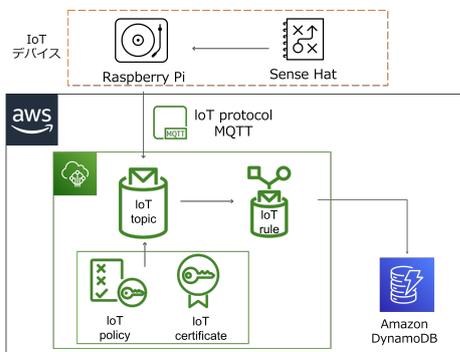


図1 センサアプリケーションの構成図

ルールクエリステートメント

```
SELECT ID () AS ID, Temperature() AS Temperature, Humidity() AS Humidity, Pressure() AS Pressure, Timestamp() AS Timestamp, Acceleration_X() AS Acceleration_X, Acceleration_Y() AS Acceleration_Y, Acceleration_Z() AS Acceleration_Z * FROM 'office/#'
```

表1 ステップごとのソースコード比較

	クライアント側			サーバ側
	データ取得	データ送信	その他	データ格納
AWS	40行	38行	16行	1行
ローカル	54行	6行	16行	38行

3 評価

AWS を用いた構築とローカル環境でのソースコードの行数を各ステップごとに比較することで AWS IoT Core の開発方法の評価を行う. 図1のセンサアプリケーションを VMware 上で MariaDB10.3.27 を用いて構築する. 通信方式としては MQTT 通信が TCP/IP をベースとして作られているため, ソケット通信を使用する. 評価結果を表1に示す.

表1からソースコードの総数はローカル環境では114行, AWS 環境では97行であり, 総数での比較だと大きな差が出ていないことが分かる. この要因として, AWS に接続してデータを送信するステップに差が出ていることが挙げられる. しかし, この38行の工程は一度構築してしまえば, 別のセンサ機能を使用した際にもほとんど同じコードが使用できる. このことを考慮すれば AWS を用いることでコードの削減が期待できる. また, AWS 環境を用いることで図1のサーバ側の機能のほとんどが UI を用いて構築することが可能となる為, ローカル環境での開発と比べて, 学習コストとソースコードの削減が期待できる.

4 まとめ

本研究では, パブリッククラウドである AWS の AWS IoT Core 機能を用いた開発方法の評価を行った. 本研究では使用するサービスが1つだった事から大きな差が現れなかったが, 複数の AWS サービスを用いた開発を考慮すると AWS の有用性が明らかになる.

参考文献

[1] "AmazonIoTCore の特徴", <https://aws.amazon.com/jp/iot-core/features/?nc=sn&loc=3>, 2020年12月閲覧.