

## 多段階エッジ計算手法を導入した センサ情報収集システムの予備評価

吉見 真聡<sup>†</sup>  
TIS Inc.<sup>†</sup>

清家 巧<sup>‡</sup>  
TIS Inc.<sup>‡</sup>

### 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]. 年率 17% の増加が見込まれる IoT デバイスに対応するため、センサ情報を収集するには、データ転送経路でデータを結合したり、集約するような、エッジ計算の技術が求められる。様々なエッジ計算手法が提案されているなかで、エッジの位置と計算リソースに応じて、多段階の演算が有望と考えられる。

本研究報告では、主に生体センサを対象としたセンサネットワークを実現する技術的解決として、センサ、エッジ、クラウドの3段階におけるエッジコンピューティングを導入した試作システムを提案し、その予備評価を行った結果について報告する。

### 2 多段階エッジ計算構造の提案

#### 2.1 エッジ計算における課題

多くの IoT システムにおけるデータフローは、センサデータが、センサが接続された計算装置を起点に、オンプレミスやクラウドのサーバに送信され、蓄積される。蓄積されたデータに対して演算を行い、サービスが構成される。エッジ計算は、センサ情報を集約、結合したり、

データ生成源近くで分類や検知を行ったりと、ネットワーク通信量の削減や低遅延な反応を実現するために利用される。ここで、解決が求められる問題が2つ挙げられる。

第1に、センサが検知した物理情報はデジタル値として計算機に取り込まれるが、多様な生体/環境センサ製品 (素子を含む) が利活用されている現在では、そのデータ構造はセンサ製品/素子のベンダによって様々であり、統一されていないことである。この不統一性がロックインを生み、センサ製品やセンサデータを活用したサービスの発展を抑制する一因となっている。

第2に、エッジ計算が行える計算機が、センサデータを取得するスマートフォンやルータ、またはクラウドの入り口に限定されてしまっていることが挙げられる。ネットワークの端点でのエッジ計算は一定の効果を上げるものの、アプリケーションの要求に柔軟に応える構成とは言い難い。

#### 2.2 多段階エッジ計算の構造

前節で述べた問題に対応するために、ネットワーク経路における多段階のエッジ計算を行う仕組みを提案する。図1に、コンテナをセンサからクラウドに至る一連の通信経路で流通させる3階層に分けて図式化した構造を示す。

もっとも左側 (センサ側) に位置するセンサ製品や素子群は、センサが検知した物理情報をデジタル信号 (センサデータ) として、近隣のエッジ計算機に送信する。このとき送信されるセンサデータは、センサベンダにより多様であり、必ずしも構造化されていない。

Preliminary Evaluation for a Sensor Data Collection System with Multi-Level Edge Computing

<sup>†</sup> YOSHIMI Masato, TIS Inc.

<sup>‡</sup> SEIKE Takumi, TIS Inc.

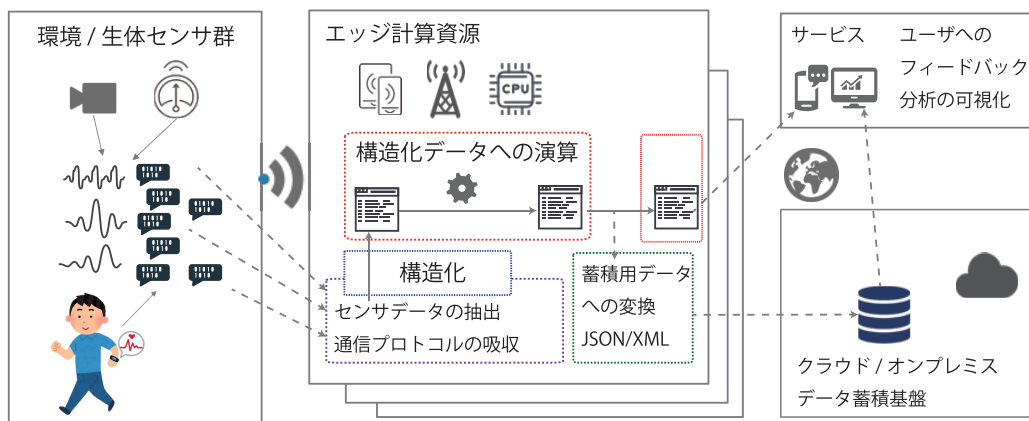


図1 多段階エッジ計算の構造

多段階のエッジ計算を実現するためのキーとなる仕組みは、共通化されたデータ構造である。異なるデータ構造を持つセンサデータを、共通化された構造でラップする方法が考えられる。端点となるエッジ計算機でデータを共通構造にラップし、以降の計算はその構造を入出力とするコンポーネントとすることで、柔軟な演算を適用することが可能となる。これは、エッジ計算の演算コンポーネントを複数接続して演算規模を拡張したり、異なるエッジ計算機に演算を分散したりと、高い柔軟性を実現する方法である。

### 3 プロトタイプシステムの実装

前章の仕組みの効果を検証するため、センサ素子、組み込みマイコン、シングルコンピュータ、クラウドからなる、多段階のエッジ計算機システムを実装した。各段階におけるソフトウェアは、C言語またはそれに準じるプログラミング言語で実装できるため、ソースコード1に示す構造を使用して、エッジ計算機およびコンポーネント間のインタフェースを行った。

プロトタイプシステムは、生体/環境センサと Arduino, エッジ計算機として RaspberryPi, データ蓄積基盤として FIWARE Orion+Elasticsearch を使用して、時系列データを蓄積、可視化する実験を行った。

#### ソースコード 1 共通センサデータ構造

```

1 typedef struct SensorData {
2     uint8_t Length;
3     uint32_t TimeStamp;
4     uint32_t SensorId;
5     uint8_t* Payload; // センサデータ
6 } SensorData_t;
    
```

試作的な共通データ構造として、センサデータを Payload, データ取得時刻, Payload 長, センサタイプを属性として持つ構造を採用した。発表では、実システムのデモンストレーションに加え、プログラムの動き、性能に対する影響について説明する。

### 4 まとめと今後の展開

本研究報告では、センサデータを共通のデータ構造でラップすることで、多段階でのエッジ計算を可能とし、高い柔軟性を実現する方法を提案した。今後は、実際のセンサ製品を使用して、エッジ計算機、クラウドを接続した実用的なシステムの研究開発に取り組む予定である。

### 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.