

## エッジにおけるセンサデータの集約手法の実装と評価

## Implementation and Evaluation for an aggregation technique of sensor data in Edge Computing

吉見 真聡\* 清家 巧\*  
YOSHIMI Masato SEIKE Takumi

## 1. はじめに

センサを含むIoT(Internet of Things) デバイス数は年率17%の増加が見込まれており、従来のデータセンタやクラウドの外で生成され、処理されるデータ量は、2018年での約10%から、2025年には75%に到達すると予測されている[1]。取得されたセンサデータは、その場で利用されたり、ネットワークを介してクラウドに蓄積されたりすることになる。膨大な数のセンサから発せられるデータに対し、適切な時間と負荷で処理を継続するためには、様々な工夫が行われることになる[2]。また、様々なセンサを利活用したサービスが展開されるようになった現在における問題のひとつに、ベンダによるロックインが挙げられる。ベンダごとにIoTセンサの通信方式が異なっていたり、ひとたびセンサネットワークが敷設されるとその後の変更にはコストを要したりと、サービスが固定されて汎用性に欠けることが、普及促進をためらわせる一因となってしまっている。

IoT時代に生じつつあるこれらの問題への解決策のひとつとして、センサ製品やクラウドのようなネットワークの端点だけではない計算資源を活用する、エッジコンピューティング技術の導入が検討されている[3]。ここで課題となるものは、ベンダごとに受け入れ可能なデータ構造と演算方法の一般化であると考えられる[4]。

本研究報告では、主に生体センサを対象とした汎用性のあるセンサネットワークを実現する技術的として、エッジコンピューティングの計算方法のうち、多数のセンサデータを集約して転送する仕組みを取り上げ、その予備評価を行った結果について報告する。

## 2. エッジコンピューティングにおけるデータ構造と演算手法

IoTにおける大きな課題は、サービスベンダやセンサ製品メーカーによってデータ構造が大きく異なることである。エッジコンピュータは、MECのようなタスクがオフロードされる計算機であったり、周囲のセンサデータを集めるだけの小型のマイコンであったりと、大小様々な計算資源が利用される。これらがサービスごとに個別に決定されてしまっており、センサベンダとサービス開発者が密に結合されてしまっている。そのため、エッジにおいてサービスの拡張や複数サービスの相乗りが難しい。そこで、以下のようなデータ構造[5]と演算手法を提案する。

**データ構造** Listing 1に、センサデータの構造を示す。センサ

データをPayload、取得時刻、Payload長、センサの個体識別子を属性として持つ構造を採用した。Payload構成はベンダや製品ごとに異なるが、識別子を用いてPayloadを識別し、エッジやクラウドでの演算を行う。

**演算手法** 図1に、エッジで行う演算のためのコンポーネントを示す。前項で述べた構造を持つセンサデータを入力とし、演算処理を行った上で、Payloadに結果を格納した上で、同様の構造を持つデータとして出力する。入力(Ingress)と出力(Egress)の時間間隔には差があることが多いので、入出力間は複数のFIFO型のバッファで接続する。このバッファ間のデータの移動はソフトウェア・プログラムで記述されるが、その際に、データの加工や集約を行う。図1のコンポーネントは、前後に接続することができるとともに、センサデバイスやゲートウェイ、クラウドなどの計算資源に配置できる。

Listing 1 センサデータのデータ構造

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

## 3. センサデータ集約システム

## 3.1 実験環境

前章で述べたデータ構造と演算手法を組み合わせた評価システムとして、TI社のCC3220SFを使用してセンサデータ集約システムを実装した。CC3220SFは、出力用にWiFi接続用のネットワークコントローラ(Simplelink)と、センサ素子からの入力を受け付けるI2C/SPI端子を持つ、32-Bit ARM Cortex-M4を組み込んだプロセッサである。

図2に、構築した実験環境を示す。I2C接続したセンサからデータを取り込み、WiFi経由でクラウド上に構成したMongoDBに転送して蓄積するシステムを構築した。

## 3.2 センサデータの集約

センサデータとは、センサデバイスによって感知された生体信号や環境シグナルを、ADコンバータなどを介してデジタル値に変換されたものである。このデジタル値は、連続的に取得される信号情報であるので、サービスに適した形に変換または集約する必要がある。

図2の実験環境において、エッジプロセッサに図1のコンポーネントを組み込む場合、センサから取得された値の加工は

\* TIS 株式会社 戦略技術センター TIS Inc.

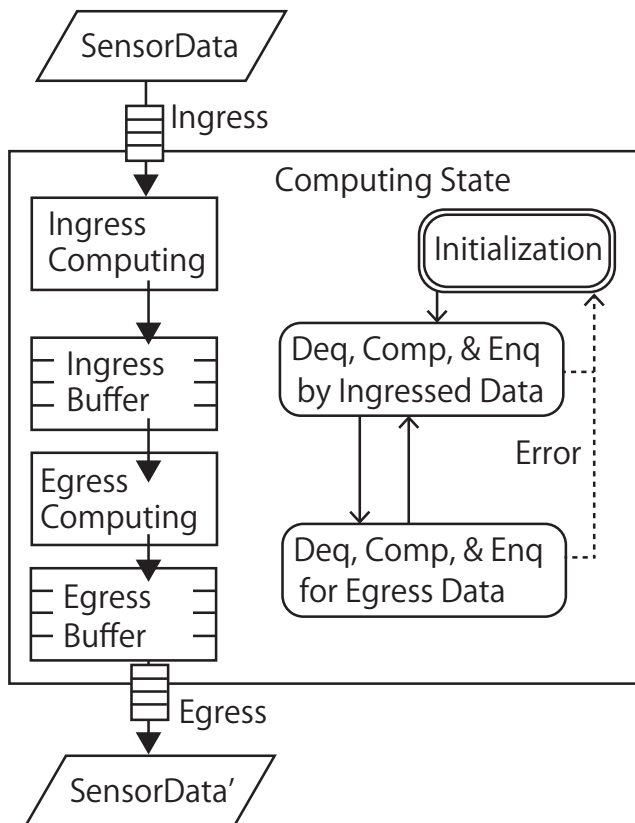


図1 エッジにおける演算コンポーネント

Ingress 部で、集約部は Egress 部で処理する方法が考えられる。Ingress 部では、バイト列であるセンサデータを Listing 1 の構造にラップして、バッファに格納する。その一方で、Egress 部では、Listing 1 の時刻変数やデータ数に基づき、Payload のセンサデータに対して集約演算（平均値、最大値、最小値、データ数など）を行い、出力する。

#### 4. エッジ計算の実装

図1に示した演算コンポーネントを、CC3220SF用に実装した。リアルタイムOSを使用せず、C言語でプログラムを記述した。TI社が提供するSimplelink SDKを使用し、WiFi機能を実装している。Ingress部では、取り込まれたセンサデータを Listing 1 の構造に組み込み、Ingress Buffer に書き込む。Egress部では、Ingress Buffer からデータを一定数取り出し、集約値を計算した上で Egress Buffer に書き込み、クラウド上の MongoDB インスタンスに転送する。OSによるサポートがないので、スレッド処理やメモリ管理の機構は利用できない。そのためプログラムでは、(a) 図1の右側に示す状態、(b) 処理を行う各関数の実行を管理するキュー、(c) 事前に確保したメモリ領域の3つを使用してジョブスケジューラを実装し、擬似的に複数の処理を並列実行できる仕組みを実装した。

クラウドでは、受け取ったデータ構造をJSONフォーマットに変換してから蓄積するサーバ・プログラムを実装した。

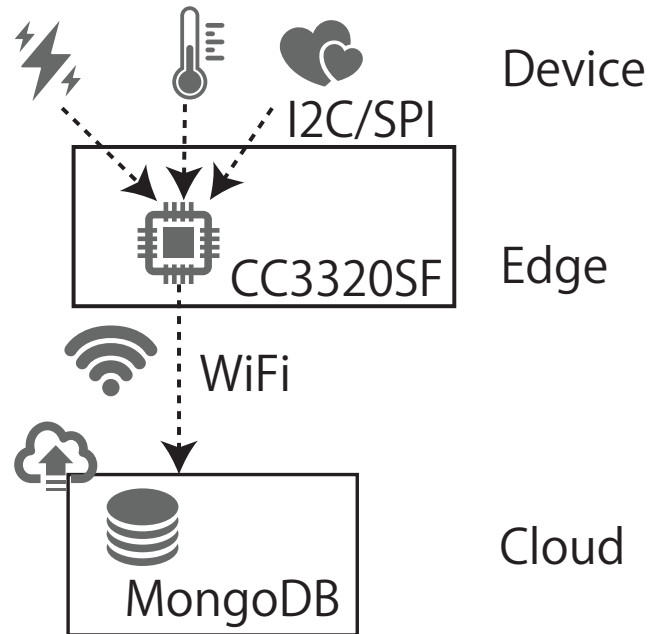


図2 実験環境（概要）

#### 5. 予備評価

発表では、図1と図2で示した構造に基づき、前章で示した集約システムの実験結果について検討し、有用性について議論する。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。

#### 6. まとめと今後の展開

本研究報告では生体、環境センサからの情報を集めるためのシステム構成について議論し、センサデータを集約して転送する仕組みを提案した。今後は、実際のサービス混在型のエッジ処理について検討し、構造を明らかにすることを予定している。

#### 参考文献

- [1] Rov van der Meulen: Edge computing promises near real-time insights and facilitates localized actions, <https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders/> (2018).
- [2] LFNetworking and LFEdege: Edge Networking: An Introduction, <https://www.lfnetworking.org/edge/> (2020).
- [3] LF Edge: WHITE PAPER: Sharpening the Edge: Overview of the LF Edge Taxonomy and Framework, Technical report, LF Edge (2020).
- [4] 吉見真聡: IoTセンサ情報収集システムの提案と予備評価, 技術報告 C-001, 情報科学技術フォーラム FIT (2020).
- [5] 吉見真聡, 清家巧: 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, 技術報告 6D-05, 情報処理学会第83回全国大会 (2021).