

センサネットワークにおけるデータストリーム処理に関する検討

山田典生[†] 峰野博史[†] 水野忠則[†]

[†] 静岡大学情報学部

1 はじめに

近年のセンサデバイスの進展により、環境データやウェアラブルセンサによるバイタルデータなどが、より高密度でリアルタイムに取得できる環境が実現されてきている。そのような環境においては大量なセンサによって高頻度に生成された大量なデータの処理方法やマイニングといったことが重要になってくる。センサデータ処理の特徴を以下に示す。

- センサデータを分析することによって生成されるイベントは、元データとして複数のセンサデータを用いることがある
- 高頻度で生成されるセンサデータは時系列データとして扱われることがある
- センサのデータ生成頻度は、センサデータの信頼性や用途に応じて決定されるため、容易に変更できない
- 蓄積したいデータとリアルタイムに処理したいデータとがある

センサによって収集されたデータから生成されたイベントは、様々なアプリで利用されることが想定される。特に本研究も参加する PUCC を用いることで、センサからのデータをトリガーとして様々な機器を連携させることが可能になると考えられる。本論文では PUCC ネットワーク上でセンサネットワークにて取得された大量のデータを効率よく処理しイベントを生成するシステムの検討を行う。

本稿は次のように構成される。2章ではセンサデータの特徴について述べる。続く3章では関連研究としてデータストリーム処理を対象としたシステムである DSMS と異種ネットワークの連携を目的とする PUCC について述べる。4章では検討しているデータ処理システムについて述べる。そして、5章で本稿のまとめと今後の課題を述べる。

2 センサ

センサのデータ生成頻度は高低様々である。体重のようなものであれば日単位で十分であるが、気温や室温では分単位である。また、同じ気温を測る場合でも、要求に応じてデータ頻度は異なってくる。これらデータ取得頻度の多くはアプリに依存して決定し、自由に変更することはできない。例えば、表 1[1] のように、医療センサは高頻度でデータを生成している。しかし、これらのデータ取得頻度は、データの信頼性を保証するためには、下げることはできない。

よりセンサが普及した環境では医療センサのような高頻度にデータ生成するセンサが家庭内に入ってくるのが考えられる。そのような場合、大量で高頻度に到着するデータの処理が必要になる。また、医療センサによる異常検知からイベントを発生させるためにはリアルタイム性も必要になる。

表 1: 医療センサのデータ生成頻度

センサデバイス	データ生成頻度
SpO2 (動脈血酸素飽和度)	3Hz
EKG (心電図)	256Hz
加速度センサ	100Hz
EEG (脳波計)	256Hz
EMG (筋電図)	1024Hz

データの長期的な監視からリアルタイム処理までを扱うことが求められる。

3 関連研究

関連研究として、連続してデータが到着し続けることを考慮したデータストリーム処理システムと、異種ネットワークの連携を実現する PUCC について述べる。

3.1 データストリーム処理

近年、継続して到着するデータをデータストリームとして、このデータストリームを処理対象としたシステムが研究されてきている。既存のデータ処理システムのひとつに DBMS(Database Management System) があるが、この DBMS ではデータ蓄積のためにディスクへのアクセスによる遅延が発生する。そのため、先に述べた医療センサのように高頻度データをリアルタイムに処理することは DBMS では困難である。また、既存の DBMS では到着するデータを DB に登録することにより、センサデータを時系列として扱うことを困難にしている。

このような背景からデータストリーム処理システムが考えられている。そのひとつに DSMS(Data Stream Management System)[2] がある。この DSMS は到着したデータのリアルタイム処理に焦点を当てているため、DB 機能を有しない。DBMS と DSMS のイメージを図 1 に示す。DBMS ではデータを蓄積したうえでクエリーを発行していたが、DSMS ではあらかじめ登録したクエリーにデータを渡して結果を得るようなイメージになる。DSMS では事前登録したクエリーを連続的に実行し続けることで、到着したデータを処理し続ける。

3.2 PUCC

現在、世界中では様々な種類のネットワークが利用されている。それらのネットワークを跨いで互いに通信できれば、既存のネットワークよりも広いネットワークを利用でき、より多くのデバイスを連携したサービスが可能になる。PUCC (P2P Universal Computing Consortium) [3] ではオーバ-

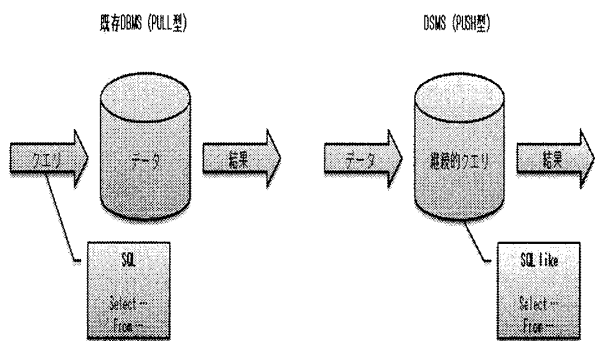


図 1: DBMS と DSMS の比較イメージ

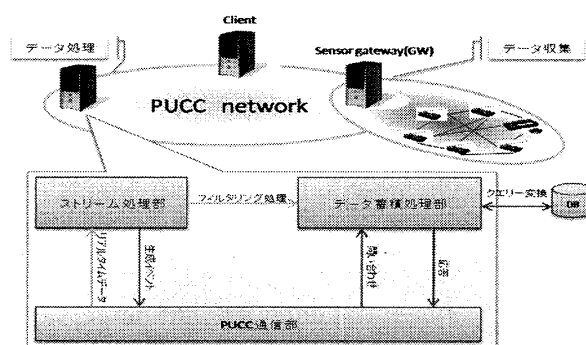


図 2: 提案システムの概要

レイネットワークの技術を利用して、個々のネットワーク間の差異を吸収して、異種ネットワーク間通信をシームレスに行うことができる環境を実現している。実装としては、Pucc のミドルウェアを利用することで異種ネットワーク間での通信を可能にしている。

4 提案システム

Pucc ネットワーク上でセンサなどにより生成されたデータを蓄積・分析処理をするデータ処理部が必要になると考えられる。本研究では、様々なデータからイベントを生成する、異種ネットワーク連携向けデータ処理システムの開発を行う。

4.1 目的

2章で挙げたように、今後のセンサの普及に従って、既存のシステムでは対処しきれなくなることが想定される。現在、センサネットワークと Pucc ネットワークの中継には、本研究室で開発したセンサゲートウェイ (GW) を介している。このことで、センサネットワークと Pucc ネットワークを連携させ、センサによる Web カメラ制御などの環境を実現している。しかし、現在の環境には大きく 2つの課題がある。

- データストリームのような高頻度データのリアルタイム処理が困難である
- 時系列データの処理を行うことができない

本研究では上に挙げた 2つの課題に対応したシステムの実現を目的としている。ひとつは、センサデータのリアルタイム処理の問題である。既存のセンサ GW ではイベント生成に DBMS を用いているため、高頻度に到着するデータではディスク処理による遅延が発生する。このシステムでは医療センサまでを想定しているため、データのリアルタイム処理やイベントの信頼性も求められてくる。

二つ目に、時系列データを扱うことが困難であることを挙げる。これも DBMS を用いることに起因する問題であるが、センサデータは一度 DB へ登録されるため、データの時系列という面が失われてしまう。本稿では PCUCC ネットワーク上のデータを集中的に処理するためのデータ処理部を生成する。

4.2 システム概要

アーキテクチャを図 2 に示す。本システムでは、センサ GW はセンサデータを収集する役割とし、センサなどから得られる Pucc ネットワーク上のデータを処理・蓄積することはデータ処理部として別に作成する。センサデータ受信部

と処理部を分散させることでスケーラビリティの向上にもつながる。

図 2 の左側のデータ処理部では、データストリーム処理とデータ蓄積処理を並列に行う。ユーザはデータ蓄積とリアルタイム処理を必要に応じて選択的に使い分けることが必要であると考えられる。場合によっては、リアルタイム処理した結果を蓄積ということも行われる。また、蓄積した長期データを解析するということが行われることが考えられる。

現在、オープンソース・ストリーム処理システムである Esper[4] を用いた実装を行っており、Wii リモコン (センサ) から取得した毎秒 100 のデータからのイベント生成を実装している。

5 本稿のまとめと今後の課題

本稿ではセンサデータの収集からイベント生成までを検討した。従来は一つのシステム内で行っていた、センサデータ収集と分析を分散させることによって、スケーラビリティが向上し、負荷分散が可能になると考えられる。また、データストリーム処理と蓄積処理をユーザが選択できることで可用性が高まることが期待される。

今後はストリーム処理とデータ蓄積処理の連携の検討を行い、アプリ側からの使いやすさやユーザのイベント設定の容易さについても踏まえながら実装していく。

参考文献

- [1] Marion Blount, John Davis, Archan Misra, Daby Sow, Min Wang, "A Time-and-Value Centric Provenance Model and Architecture for Medical Event Streams", HealthNet 2007, pp. 95-100, 2007.
- [2] Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Keith Ito, Itaru Nishizawa, Justin Rosenstein, and Jennifer Widom, "STREAM: The Stanford Stream Data Manager (Demonstration Proposal)", ACM SIGMOD 2003, p.665, 2003.
- [3] Norihiro Ishikawa, Takeshi Kato, Hiromitsu Sumino, Shingo Murakami, and Johan Hjelm, "Pucc Architecture, Protocols and Applications", IEEE CCNC 2007, pp. 788-792, 2007.
- [4] Esper: <http://esper.codehaus.org/>