

大規模センサデータを処理可能な分散型データ管理システムの提案

松浦 伸彦^{1,†1,a)} 大畑 真生² 太田 賢² 稲村 浩² 水野 忠則³ 峰野 博史⁴

受付日 2012年3月31日, 採録日 2012年11月2日

概要: 近年, Home Energy Management System (HEMS) や Building Energy Management System (BEMS) のようなありとあらゆるセンサデータを収集, 解析することができるシステムを構築することで, 新たなサービスを実現しようとする試みが行われている. このようなサービスではいかにユーザを取り巻く環境を把握するかが重要であり, そのためにはユーザの位置や状況といったコンテキストをセンサデータから推定し, システム自らがコンテキストを考慮して動作するようなコンテキストウェアサービスでなければならない. コンテキストウェアサービスの実現において, センサを用いることで大量のデータを取得してコンテキストを推定可能とする一方, その大量のデータをどのように収集, 蓄積, 解析するのかという課題が発生する. この課題を解決するために, 私たちは分散型センサデータ管理システムを提案する. 本システムは PUCC プロトコル利用によるセンサ差異吸収機能の実現と, データベース管理システムとデータストリーム管理システムの融合による高い書き込み性能とクエリ処理性能を提供する.

キーワード: 分散処理システム, DBMS, EMS

A Proposal of the Distributed Data Management System for Large-scale Sensor Data

NOBUHIKO MATSUURA^{1,†1,a)} MAKI OHATA² KEN OHTA² HIROSHI INAMURA²
TADANORI MIZUNO³ HIROSHI MINENO⁴

Received: March 31, 2012, Accepted: November 2, 2012

Abstract: Context-aware services, such as those for home and building energy management systems, have attracted a great deal of attention. Because services need many sensors to enable us to understand our surrounding environment, the use of sensors has been increasing all over the world. We can obtain vast amounts of information with these sensors, but they raise one serious problem, that is how to collect, store, and analyze data. To solve this problems, we propose a distributed sensor data management system that uses a common format access method, and has a high-write and a high-query-processing performance.

Keywords: distributed system, DBMS, EMS

¹ 静岡大学大学院情報学研究所
Graduate School of Informatics, Shizuoka University, Hamamatsu, Shizuoka 432-8011, Japan
² 株式会社 NTT ドコモ先進技術研究所
Research Laboratories, NTT DOCOMO, Inc., Yokosuka, Kanagawa 239-0847, Japan
³ 愛知工業大学情報科学部
Faculty of Information Science, Aichi Institute of Technology, Toyota, Aichi 470-0392, Japan
⁴ 静岡大学情報学部
Faculty of Informatics, Shizuoka University, Hamamatsu, Shizuoka 432-8011, Japan

1. はじめに

近年, センサの小型化と低コスト化によりセンサを搭載した組み込み機器などが続々と登場しており, 世界中のセンサの数が増加している. センサは工業自動化 [1] や農業支援 [2] のほか, 家電機器の最適化といった身近な目的

^{†1} 現在, 日本電信電話株式会社 NTT 未来ねっと研究所
Presently with NTT Network Innovation Laboratories
a) matsuura@minelab.jp

にも利用される。一方、Home Energy Management System (HEMS) [3] や Building Energy Management System (BEMS) [4] のようなありとあらゆるセンサデータを収集、解析することができるシステムを構築することで、新たなサービスを実現しようとする試みも行われている。このようなサービスではいかにユーザを取り巻く環境を把握するかが重要であり、そのためにはユーザの位置や状況といったコンテキストをセンサデータから推定し、サービス自らがコンテキストを考慮して動作するようなコンテキストウェアサービスでなければならない。

以上のような特徴を持つコンテキストウェアサービスを提供するためには、膨大な量のセンサデータ処理に特化した処理基盤が必要となる。クラウド技術の発展によりコンピュータ資源を容易かつ安価に利用できるようになってきた一方で、その上で動作する処理基盤の多くは業務データのように扱いやすい構造化データを対象としたものである。一方、センサデータはビッグデータの種類であり、多種多様かつ常時大量に非構造化データが発生し、バッチ処理だけでなくリアルタイム処理も要求されるという特徴を持つ。また、複数のセンサネットワークの結果を一様に扱うために、センサデータのフォーマットの違いといった仕様の差異の吸収を行う必要がある。このように、扱いにくい非構造化データであるビッグデータを扱うためには今までとは異なる新たな仕組みが必要となる。そこで、本論文では高効率かつ高可用性を持つ分散型のセンサデータ管理システムを提案する。本システムは、Peer-to-peer Universal Computing Consortium (PUCC) プロトコルの採用による高可用性ならびにセンサデータの差異吸収と、データベース管理システム (DSMS) とデータストリーム管理システム (DBMS) の連携による高効率の大量データ処理機能を持つ。

2. 関連研究

2.1 データ管理システム

2.1.1 Peer-to-peer (P2P) ネットワーク

Peer-to-peer (P2P) ネットワークは、オーバーレイネットワークと呼ばれる仮想ネットワークを物理ネットワークの上に構築し、オーバーレイネットワークに参加する P2P ノードが相互に通信するネットワークである。ネットワークの構築・維持では、クライアントサーバ型アーキテクチャにおけるサーバのような管理者を持たず、P2P ノードが共同で取り組む。P2P ノードは ID と近隣ノードリストを持ち、各ノードが持つ近隣ノードリストから最も近いノードを再帰的に選択し続けることで目的のノードへメッセージを送信する。

P2P ネットワークはネットワークとしての側面を持つ一方、分散型データストアとしての側面も持つ。この分散型データストアを分散ハッシュテーブル (Distributed Hash Table: DHT) と呼び、キーとバリューからなるデータの分

散管理を行う。DHT を構築するためのアルゴリズムとして、Chord [5] や Skipgraph [6] がある。上記アルゴリズムは、ハッシュ関数を用いてオーバーレイネットワークの構築ならびデータの保存を行う。各ノードは IP などノード固有情報のハッシュ値を ID として持ち、ハッシュ値の差をノード間の距離とする考えに基づいた近隣ノードリストを作成する。データを保存する場合は、データのキーのハッシュ値を求め、そのハッシュ値に最も近い ID を持つノードを保持ノードとして選択する。データを取得する場合は、保存するときと同様にキーのハッシュ値から保持ノードを見つけ、保持ノードに対して取得リクエストを送信する。DHT は、ハッシュ値によりランダムにデータが分散するために高いスケーラビリティと負荷分散性を持つ。しかし、センサデータのような複雑なデータを持つことができず、データアクセス時に必ずノード探索のオーバーヘッドが発生する点が課題である。

2.1.2 データベース管理システム (DBMS)

データベース管理システム (Database Management System: DBMS) はデータ管理を一括して行うためのミドルウェアである。従来、DBMS を構築するために RDB が利用され、列と行からなる表形式のスキーマでデータをノード 1 台で集中管理していた。しかし、近年では NoSQL が登場し、複数台でクラスタを構築してデータを非スキーマで分散管理することができるようになった。Google の Bigtable [7] や Amazon の Dynamo [8] の登場をきっかけに注目が集まり、現在では表 1 に示すように数多くのディストリビューションが登場している。NoSQL を特徴付ける指標には様々なものが存在するが、その中でも CAP 理論 [9] が特に有名である。CAP 理論とは、一貫性 (consistency)、可用性 (availability)、分散性 (partition tolerance) の 3 種類の性質について、分散システムは多くても 2 つしか同時に満たすことはできないという理論である。NoSQL は分散管理するために分散性は必須であり、残る一貫性と可用性のどちらを重視しているかが各ディストリビューションの違いをよく表している。NoSQL の開発がさかんに行われる一方で、それらを定量的に評価しようとする研究 [10] も行われている。上記研究ではスループットと遅延の観点で、Cassandra, HBase, Pnuts, MySQL の 4 種類の DBMS を評価している。論文の評価結果では、RDB は読み込みに最適化され、NoSQL は書き込みに最適化されていることが述べられている。そのため、NoSQL はセンサデータ管理システムのように高頻度にデータ格納が発生する場合に向いていることが分かる。しかし、NoSQL はその読み込み性能の弱さからクエリ処理能力に課題が存在しており、特にリアルタイム処理を要求される際に問題となる。

2.1.3 データストリーム管理システム (DSMS)

データストリーム管理システム (Datastream Management System: DSMS) はデータストリームの管理ならび

表 1 NoSQL の種類
Table 1 Kind of NoSQL.

Name	Data model	CAP theorem	Distribute model	Persistence model
Cassandra	Column	AP	Consistent hash	Memtable/SSTable
HBase	Column	CP	Sharding	Memtable/SSTable on HDFS
CouchDB	Document	AP	Consistent hash	Append-only B-tree
Riak	Document	AP	Consistent hash	?
MongoDB	Document	CP	Sharding	B-tree
Tokyo cabinet	KeyValue	AP	Consistent hash	Hash or B-tree
Voldemort	KeyValue	AP	Consistent hash	Pluggable
Redis	KeyValue	CP	Consistent hash	In-memory with background snapshots
Scalaris	KeyValue	CP	Consistent hash	In-memory only

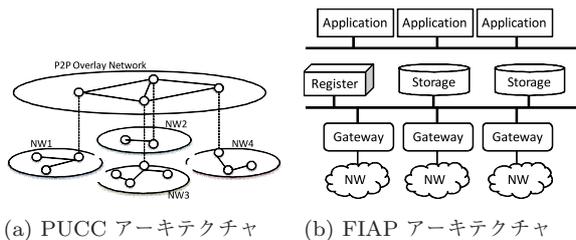


図 1 異種ネットワーク連携プロトコル

Fig. 1 Heterogeneous network convergence protocol.

クエリ処理を行うシステムである。代表的なシステムとして StreamSpinner [11], Auroa [12], Borealis [13] などがあ。DBMS との大きな違いは、データ 1 つ 1 つを見るのではなく、それらを束ねたストリームとして扱う点である。データ取得には DBMS と同様にクエリが用いられ、ストリームに流れてきたデータと逐次マッチングを行い、クエリ発行者に対してマッチング結果が届けられる。クエリは 1 度登録されると永続的にマッチングに利用され続けるため、継続的クエリ (Continuous Query: CQuery) と呼ばれる。データの到着と同時にマッチングを行うためにクエリ処理能力が非常に高く、リアルタイム処理に適している。しかし、データ保存を行わないためにバッチ処理を行うことができず、また後からデータを参照することもできない点が課題である。

2.2 異種ネットワーク連携プロトコル

センサネットワークの利活用ならび処理について議論しているワーキンググループとして P2P Universal Computing Consortium (PUCC) [14] と Facility Information Access Protocol (FIAP) [15] が存在する。両グループでは、センサネットワークのさらなる活用を目指して、センサデータの表現方法や基本的な処理プラットフォームについて提案がなされている。今回、センサデータのフォーマットやプロトコルの違いといった使用差異の吸収とそれを実現するアーキテクチャの参考とするべく、それぞれの特徴を下記に示す。

図 1(a) に PUCC のアーキテクチャを示す。PUCC は P2P ネットワーク技術を利用して異種ネットワーク間連携をサポートする。PUCC に参加する各ノードはサービスと

イベントという概念を持ち、他ノードのサービスを起動する Invoke メッセージ、イベントを受け取る Notify メッセージ、またイベント設定を行う Subscribe メッセージなどの PUCC プロトコルが存在する。PUCC プロトコルを利用してセンサデータのやりとりを行うためには 2 つの方法が存在する。1 つ目はサービスを使う方法で、“getSensorData” や “setSensorData” のようにセンサデータにアクセスするサービスを実装し、Invoke メッセージで起動する方法である。2 つ目はイベントを使う方法で、センサデータを持つノードに対して Subscribe メッセージを利用してイベント設定を行い、センサデータを含むイベントを Notify メッセージで受け取る方法である。上記のように PUCC アーキテクチャを用いることで、センサデータを管理するノードを PUCC ノードとして参加させ、サービスまたはイベントを利用する方法でセンサデータにアクセスすることが可能となる。

図 1(b) に FIAP のアーキテクチャを示す。FIAP は従来のインターネット構造を模した構造となっており、3 つのコンポーネントから構成される。レジスタは誰が何を持っているのかを管理する役割を、ストレージはゲートウェイから送られてきたデータを保存する役割を、ゲートウェイはセンサネットワークなどの背後のネットワークからデータを FIAP ネットワーク上に流す役割を持つ。上記のようにセンサデータはゲートウェイを介してストレージに保存される仕組みとなっているため、アプリケーションは適切なストレージにアクセスしてセンサデータを取得することが可能である。

2.3 要求事項の整理と関連研究のまとめ

センサデータのようなビッグデータを用いたコンテキストウェアサービスを提供するための処理基盤を提案するにあたり、要求事項と関連研究の整理を行う。まず、ビッグデータは下記の特徴を持つ。

- 常時大量にデータが発生する。
- バッチ処理だけでなくリアルタイム処理も要求される。
- 多種多様のフォーマット、データ形式である。

そのため、(1) 処理基盤は常時発生するセンサデータや推測したコンテキストなどを取りこぼすことなく保存し、

表 2 データ管理システム関連研究のまとめ

Table 2 Summary of data management system.

システム	書き込み性能	スケーラビリティ	クエリ処理能力	データ蓄積
P2P	NG	OK	NG	OK
RDB	NG	NG	OK	OK
NoSQL	OK	OK	NG	OK
DSMS	-	-	OK	NG
提案システム	OK	OK	OK	OK

表 3 異種ネットワーク連携プロトコル関連研究のまとめ

Table 3 Summary of heterogeneous network convergence protocol.

プロトコル	ネットワーク構成	センサ仕様差異吸収	スケーラビリティ
PUC	P2P	OK	OK
FIAP	Server Client	OK	NG

また (2) センサデータの差異をプログラマやマネージャが意識しなくてもよい仕組みである必要がある。また、(3) 集まったセンサデータに対してリアルタイム処理を行って低レスポンスタイムで応答が可能である必要もある。よって、大量の書き込み要求を処理できる高い書き込み性能とスケーラビリティ、リアルタイム処理を行うことができる高いクエリ処理能力、センサの仕様を意識せずデータを扱うための仕様吸収機能が処理基盤には必要とされる。当然ながら (4) 後でデータにアクセスするためのデータ蓄積機能も必要不可欠である。

上記の 4 点の要求事項を考慮した場合、関連研究はそれぞれ表 2、表 3 にまとめることができる。表 2 より、既存のデータ管理システムではすべての要求事項を満たすことができない。しかし、NoSQL と DSMS に着目すると互いに短所を補いあう関係にあることが分かり、互いをうまく組み合わせることができれば、NoSQL は高い書き込み性能とスケーラビリティを持つデータ蓄積機能を提供し、DSMS は高いクエリ処理能力を提供することができる。本論文ではこの組合せによる効果に着目し、DSMS と NoSQL を組み合わせた高い書き込み性能とクエリ処理性能を持つセンサデータ管理システムを提案する。また、表 3 より、PUC プロトコルを利用してさらなるスケーラビリティの向上とセンサ仕様吸収機能の実現も行う。

3. 分散データベースを用いたセンサデータ管理システム

センサデータはビッグデータの種類であり、多種多様のデータが常時大量に発生し、バッチ処理だけでなくリアルタイム処理も要求される。従来のアーキテクチャでは RDB を用いてデータベースサーバを構築していたが、RDB は書き込み性能が弱いためにセンサデータの大量の書き込み要求を処理することができない。一方、RDB の代わりに NoSQL を利用することで大量の書き込み要求を処理することが可能となるが、NoSQL の読み込み性能の弱さからリアルタイム性を必要とするクエリ処理に向かない。そこで本論文では、大量の書き込み要求に対応しつつもリアル

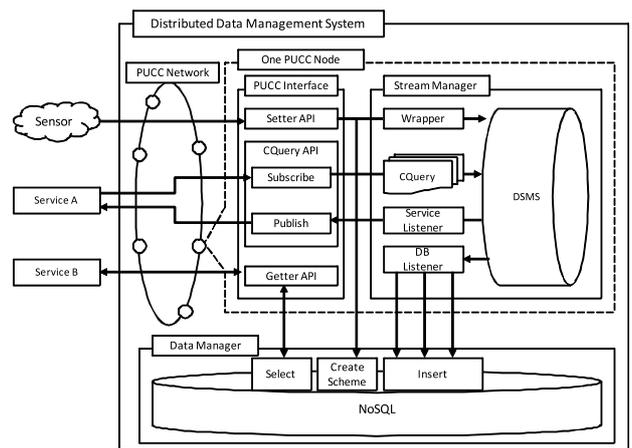


図 2 提案システムアーキテクチャ

Fig. 2 Architecture of proposed system.

タイム処理を行うことができる、分散型のセンサデータ管理システムを提案する。図 2 に提案するシステムのアーキテクチャを示す。本システムは、PUC プロトコルの採用による高スケーラビリティならびセンサデータの差異吸収と、DSMS と DBMS の連携による大量データ処理のための高い書き込み性能とクエリ処理性能を持つ。

提案システムは PUC ネットワーク、ストリームマネージャ、データマネージャの 3 つのコンポーネントで構成される。下にそれぞれの役割を説明する。まず、PUC ネットワークは提案システムのインタフェースを担当する。PUC ネットワークは複数の PUC ノードでオーバレイネットワークを構築しており、センサネットワークから送られてくるデータの分散管理を行う。PUC ノードにはセンサネットワーク単位で処理を割り当てられるため、サービスがデータを取得する際は対応する PUC ノードへリクエストを送ることになる。リクエストは PUC ノードが持つ API に届けられるが、その際にやりとりされるデータは PUC プロトコルに変換されるため、利用者はセンサデータ仕様の違いを意識せずデータを扱うことができる。次に、ストリームマネージャはセンサデータ処理を担当する。センサから送られてきたデータは Wrapper で DSMS 処理に適した形に変換され、DSMS に届けられる。DSMS にはあらかじめサービスから CQuery が設定されており、到着したセンサデータと CQuery を用いたマッチング処理が行われる。マッチング処理はセンサデータが到着すると同時に行われるため、高いクエリ処理能力を備えている。マッチング処理結果は Service Listener に送られ、CQuery を登録したサービスに対して送信される。最後に、データマネージャはセンサデータ保存を担当する。ストリームマネージャは、DB Listener を介してすべてのセンサデータをデータマネージャへと送信している。データマネージャは、受け取ったデータを高い書き込み性能を持つ NoSQL に保存する。なお、Setter API を介してセンサデータが提

案システムに到着した際に、データマネージャは NoSQL のテーブルの作成といった設定を適時行う。

各コンポーネント間のデータフローを、セキュリティサービスを例に説明する。ここで考えるセキュリティサービスは、家庭にあるモーションセンサからデータが送られ、人のいるはずのない時間帯に反応があった場合に通報を行うサービスとする。センサデータは PUCC ノードのいずれかに届けられ、Setter API を介してストリームマネージャへと届けられる。一方でサービスは、CQuery API の Subscribe オペレーションにより異常検出用 CQuery を登録する。異常値検出用 CQuery とは、たとえば人のいない日中にモーションが検出された場合が知りたいといったクエリを指し、1 家庭につき 1 つ発行されてサービスには複数台の PUCC ノードからデータが到着することになる。ストリームマネージャでは、センサデータと異常値検出用 CQuery とのマッチングにより異常発見処理が行われる。マッチングにより異常が発見された場合は、CQuery を登録したセキュリティサービスに対して CQuery API の Publish オペレーションにより処理結果が届けられる。セキュリティサービスは受け取ったデータに基づいて、あらかじめ決められた手段で通報を行う。その際、異常検出した状況をより深く調べるために、Getter API を介してデータマネージャに保存された過去のデータを取得することが可能である。

外部のサービスから要求されるクエリは次の 3 種類に分類でき、それぞれ用途が異なっている。提案システムはデータ取得をするための API を 2 つ持っており、セキュリティサービスの例のようにクエリの種類によって使い分けを行う。

- 最新の単一データ取得クエリ (single data query: SQuery)
このクエリはサービスがイベント駆動を行いたいときの要求である。異常値検出や単に現在の値を知りたいときに利用され、セキュリティサービスや子供や老人の見守りサービスなどに用いられる。サービスは CQuery API を使ってシステムに処理を依頼し、ある間隔で単一のデータを取得することができる。
- 最新の範囲データ取得クエリ (range data query: RQuery)
このクエリはサービスが直近の統計データを取得したいときの要求である。たとえば過去数分の温度平均値など、データマイニングに必要な特徴量を求めたいときに利用され、交通量予測サービスや消費電力量調整サービスなどに用いられる。サービスは CQuery API を使ってシステムに処理を依頼し、ある一定間隔の全データもしくは統計処理されたデータを取得することができる。
- 過去の範囲データ取得クエリ (past range data query:

PQuery)

このクエリはサービスが過去のデータを取得したいときの要求である。データマイニングのモデル作りや詳細情報を取得したいときに利用され、消費電力の見える化やサービスを提供するための前準備などに用いられる。サービスは Getter API を使ってシステムに処理を依頼し、指定した区間の全データを取得することができる。なお、過去の単一データ取得クエリは、用途はあまりに少なく、範囲データ取得クエリで十分補えるため省略している。

提案システムは、データマネージャは後で参照できるように全センサデータの保存に集中し、ストリームマネージャはセンサデータが流れるストリームの処理に集中する。これにより、SQuery や RQuery のようなリアルタイム性の必要な処理はストリームマネージャで行い、大量のデータにアクセスするバッチ処理はデータマネージャで行うことが可能となる。さらに、ストリームマネージャがクエリの大部分を処理することになるため、データマネージャはセンサデータの保存により集中できるようになる。データマネージャは内部の NoSQL にデータを保存しており、NoSQL は書き込みに最適化されているため、結果としてさらなる書き込み性能の向上を見込むことができる。

4. 評価

センサデータ処理基盤には、書き込み要求と前章で説明した 3 種類 (SQuery, RQuery, PQuery) のクエリ要求が届けられる。このうちリアルタイム性を必要とするのは書き込み要求と SQuery, RQuery である。そのため、提案システムと DBMS 単体のシステムとの比較評価により、書き込み性能と SQuery, RQuery の処理性能の評価を行った。

本研究の主目的は、HEMS や BEMS といったコンテキストウェアサービスを実現するためのセンサデータ処理基盤の提案と実装である。そのため、大規模になりやすく負荷の観点でより困難な要求条件を持つ BEMS をもとに評価環境の見積りを行う。我々は、センサネットワークに関わる研究のテストベッドとしてセンサデータ収集環境をすでに構築している。この環境では、静岡大学の情報棟 1 号館 4 階の半フロアをカバーするセンサネットワークを構築してデータの収集を行っている。現在はセンサネットワークはセンサ約 50 個で構成され、毎秒 50 個のデータ (50 ops) が集まっている。したがって、建物は 6 階建てであるため、建物全体をカバーした場合は 600 個のセンサから 600 ops の書き込み要求が発生することになる。これを 1 つの乗り越えるべき壁と考え、上記の環境を評価の指標とする。表 4 に評価に用いたノードの仕様を示す。

評価では Yahoo! Cloud Servicing Benchmark (YCSB) [10] を用いて行う。YCSB はクラウドサービスで利用される様々なデータストアのベンチマークを行うためのツ

表 4 評価に用いたノードの仕様
Table 4 Evaluation environment.

OS	CentOS 5.7
CPU	2.66 GHz
RAM	512 MB
HDD	50 GB

表 5 YCSB パラメータ
Table 5 YCSB parameter.

Key size	10 bytes (max)
Value size	20 Kbyte
Record count	20,000,000
Operation count	100,000

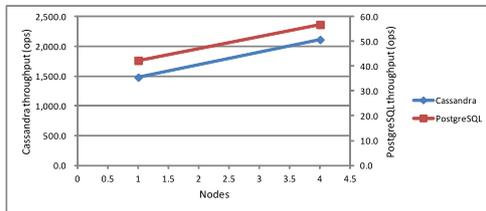


図 3 スケーラビリティと書き込み性能評価結果
Fig. 3 Scalability and write-performance result.

ルであり、RDB や NoSQL のほぼすべてに対応している。YCSB を用いた評価は 2 つのフェイズで構成されており、前準備を行うロードフェイズと実際に評価を行う実行フェイズを持つ。ロードフェイズでは、評価対象の DBMS に対してある一定数のデータのロードを行う。これは、データがある程度ロードしておいて、評価測定中のデータ数の増加による性能劣化の影響を無視できるようにするためである。実行フェイズでは、連続的に書き込みと読み込みリクエストを発生させてスループットと遅延の計測を行う。

4.1 スケーラビリティと書き込み性能評価

図 3 に書き込み性能の評価結果を、表 5 に YCSB のパラメータを示す。PostgreSQL と Cassandra の結果のみ図中に示しているが、これは提案システムのデータストア部に Cassandra を利用しているために Cassandra とほぼ同じ結果が出て見た目では区別できなくなったためであり、図中では提案システムを省略をしている。また、PostgreSQL はシャーディング構成にしており、書き込みデータに付与したユニークな ID をもとに挿入先 DB を選択することで複数ノードに対応している。

図中の Cassandra と PostgreSQL は、ノードを追加するごとの性能上昇率が等しく、同等程度のスケーラビリティを持っている。PostgreSQL のシャーディング構成は、データ管理をクライアント側で行う必要があるという欠点を考えなければ、最もスケーラビリティを向上させるために有効な手法である。そのため、Cassandra はシャーディング構成と同等程度という高いスケーラビリティを持つことが分かる。次にノード 4 台のときのスループット値に着目した場合、PostgreSQL のスループットは 56.8 ops であり、我々の必要とする 600 ops を満たしていないことが分かる。一方、Cassandra のスループットは 2113.0 ops であり要求を満たしている。PostgreSQL に対する Cassandra のスループットは 37.2 倍となっており、Cassandra の書き込

み性能が非常に優れている。上記結果は NoSQL はシャーディング構成の PostgreSQL よりもセンサーデータ管理システムに適していることを意味し、提案システムも同様の性質を持つことが分かる。

4.2 クエリ処理能力評価

クエリ処理能力の評価として、SQuery と RQuery の評価を行った。3 章で説明したように、SQuery と RQuery は非常にリアルタイム性を重視するクエリである。そのため、性能の良し悪しを決める評価基準として、データが発生してから取得できるまでのレスポンスタイムを用いる。書き込むデータとして、10 byte のキーと 100 byte のバリューのペアからなるカラムを 10 件まとめた 1 レコードを生成する。なお、バリューは 100 byte となるようなランダム文字列を生成している。評価では、上記データの書き込み頻度を変更しながら、SQuery では直近のデータを 10 秒間取得し続けたときの、RQuery では過去 10 秒間に発生したデータすべてを 1 度に取得したときのレスポンスタイムを計測する。たとえば、データ 1 からデータ 20 までが 1 秒間隔で発生し、データ 10 が発生した時点でクエリを処理した場合、SQuery はデータ 11 からデータ 20 までが 1 つずつ書き込み後すぐ届くのに対し、RQuery はデータ 1 からデータ 10 までがひと固まりで届く。上記の例のように、今回の評価ではクエリに取得条件を設けず発生したデータはすべて取得するようにし、また RQuery は統計データではなく一定間隔の全データを取得する。各クエリの具体的な処理方法は、PostgreSQL と Cassandra は蓄積データに対してクエリ処理を行い、提案システムでは CQuery を利用して処理を行っている。

図 4 に SQuery の処理結果を示す。図中のレスポンスタイムが急激に上昇している箇所がクエリ処理性能の限界を表している。そのため提案システム、Cassandra、PostgreSQL の順で SQuery 処理性能が良いことが分かる。PostgreSQL の性能限界は書き込みが追いつかなくなることで発生しており、Cassandra の性能限界は読み込みが間に合わなくなることで発生している。以上のように、PostgreSQL と Cassandra の性能の差は書き込み性能と読み込み性能の釣り合いから発生しており、SQuery は特に書き込み性能が重要であったためである。提案システムが最も良い性能であったのは、データマネージャとストリームマネージャの組合せからきている。ストリームマネージャはデータを一

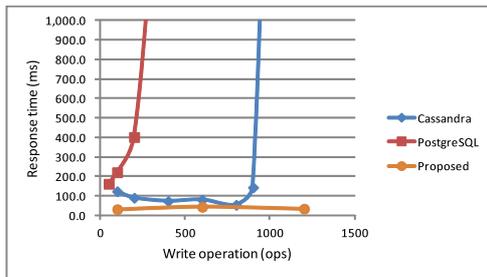


図 4 Single data query 評価結果
Fig. 4 Single data query result.

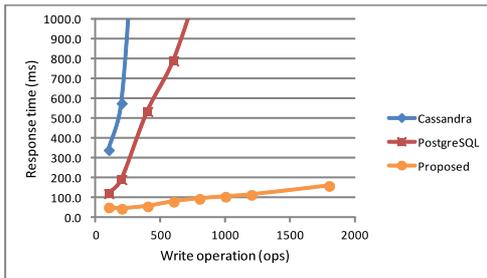


図 5 Range data query 評価結果
Fig. 5 Range data query result.

番初めに処理する箇所であり、仮想的にテンポラリストレージとして動作するため、データが書き込まれるのを待つ必要がないためである。性能限界を迎えた箇所に着目すると、Cassandra と提案システムが要求性能を満たしていることが分かる。

図 5 に RQuery の処理結果を示す。図中のレスポンス時間が急激に上昇している箇所がクエリ処理性能の限界を表している。RQuery は特に読み込み性能が重要であるため、こちらの評価では Cassandra よりも PostgreSQL の方が良い性能を示している。一方、提案システムは最も良い性能を示しており、書き込み処理の増加に対して安定して低い上昇率を見せている。性能限界を迎えた箇所に着目すると、PostgreSQL と提案システムが要求性能を満たしていることが分かる。しかし、PostgreSQL のみのシステムは SQuery の処理において要求性能を満たしていないため、クエリ処理能力に対する要求性能を満たしているのは提案システムのみであると考えられる。

5. 結論

本論文では、PUCC プロトコルの採用による高可用性ならびにセンサデータの差異吸収と、DSMS と DBMS の連携による高効率の大量データ処理機能を持つ分散型センサデータ管理システムの提案を行った。提案システムは PUCC ネットワーク、ストリームマネージャ、データマネージャの 3 つのコンポーネントで構成される。

評価では、書き込み性能とクエリ処理性能の観点から、既存の DBMS のみを使ったシステムと提案システムの比較を行った。評価指標として、すでに構築しているセンサ

データ収集環境からシステムに求められる要求性能として最低限 600 ops のセンサデータを扱う必要があるとして、600 ops に対応できるかという点を重要視した。その結果、書き込み性能とクエリ処理性能において、提案システムのみが性能要求を満たすことができることを確認した。

今後はセンサネットワークの処理ノード割当て方法の最適化を検討する予定である。今回はランダム割当てを行ったが、このままでは負荷の分散性と処理効率が損なわれてしまう可能性が存在する。たとえば、多数の中規模センサネットワークと少数の大規模センサネットワークが混在していた際、ランダム割当てでは規模とそれともなう処理量の違いを考えずに割り当ててしまうため、いくつかのノードの負荷がはねあがってしまう状況が考えられる。また一方で、大学構内で建物ごとにセンサネットワークを構成した場合、ランダム割当てではそれぞれ別々のノードで処理されてしまい、ユーザは建物数分のクエリをそれぞれの管理ノードに送信する必要がある。以上のような問題を解決するために、分散性と効率性のバランスのとれたセンサネットワークの割当て方法を検討する必要があると考えられる。

参考文献

- [1] Tommila, T., Hivonen, J., Jaakkola, L., Peltoniemi, J., Peltola, J., Sierla, S. and Koskinen, K.: Next generation of industrial automation: Concepts and architecture of a component based control system, *The 6th IEEE International Conference on Industrial Informatics (INDIN'08)*, pp.1129-1125 (July 2008).
- [2] Pierce, F. and Elliott, T.: Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington, *The International Journal on Computers and Electronics in Agriculture*, pp.32-43 (Apr. 2008).
- [3] Lu, J., Sookoor, T., Srinivasan, V., Gao, G., Holben, B., Stankovic, J., Field, E. and Whitehouse, K.: The Smart Thermostat: Using Occupancy Sensors to Save Energy in Homes, *The 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10)*, pp.211-224 (Nov. 2010).
- [4] Mineno, H., Kato, Y., Obata, K., Kuriyama, H., Abe, K., Ishikawa, N. and Mizuno, T.: Adaptive Home/Building Energy Management System Using Heterogeneous Sensor/Actuator Networks, *The 7th IEEE Conference on Consumer Communications and Networking Conference (CCNC'10)*, pp.422-426 (Jan. 2010).
- [5] Stoica, I., Morris, R., Karger, D., Kaashoek, M. and Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, *The 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'01)*, pp.149-160 (Aug. 2001).
- [6] Aspnes, J. and Shah, G.: Skip Graphs, *ACM Trans. Algorithms* (Nov. 2007).
- [7] Chang, F., Dean, J., Ghemawat, S., Hsieh, W., Wallach, D., Burrows, M., Chandra, T., Fikes, A. and Gruber, R.: Bigtable: A distributed storage system for structured data, *The 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI'06)*, p.15

- (Nov. 2006).
- [8] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Voshall, P. and Vogels, W.: Dynamo: Amazon's highly available key-value store, *The 21st ACM SIGOPS Symposium on Operating Systems Principles (SOSP'07)*, pp.205-220 (Oct. 2007).
- [9] Gilbert, S. and Lynch, N.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, *ACM SIGACT News*, pp.51-59 (2002).
- [10] Cooper, B., Silberstein, A., Tam, E., Ramakrishnan, R. and Sears, R.: Benchmarking Cloud Serving Systems with YCSB, *The 1st ACM Symposium on Cloud Computing (SoCC'10)*, pp.143-154 (June 2010).
- [11] StreamSpinner, available from (<http://www.streamspinner.org/>).
- [12] Abadi, D., Carney, D., Cetintemel, U., Cherniack, M., Conway, C., Lee, S., Stonebraker, M., Tatbul, N. and Zdonik, S.: Aurora: A new model and architecture for data stream management, *The International Journal on Very Large Data Bases*, pp.120-139 (Aug. 2003).
- [13] Abadi, D., Ahmad, Y., Balazinska, M., Cherniack, M., Hwang, J., Lindner, W., Maskey, A., Rasin, E., Ryvkina, E., Tatbul, N., Xing, Y. and Zdonik, S.: The design of the borealis stream processing engine, *The 2nd Biennial Conference on Innovative Data Systems Research (CIDR'05)*, pp.277-289 (Jan. 2005).
- [14] Ishikawa, N., Kato, T., Sumino, H., Murakami, S. and Hjelm, J.: PUCC Architecture, Protocols and Applications, *IEEE Consumer Communications and Networking Conference (CCNC'07)*, pp.788-792 (Jan. 2007).
- [15] Ochiai, H., Ishiyama, M., Momose, T., Fujiwara, N., Ito, K., Inagaki, H., Nakagawa, A. and Esaki, H.: FIAP: Facility information access protocol for data-centric building automation systems, *IEEE INFOCOM Workshop on Machine-to-Machine Communications and Networking (M2MCN'11)*, pp.229-234 (Apr. 2011).



松浦 伸彦 (正会員)

2012年静岡大学大学院情報学研究科修士課程修了。現在、日本電信電話(株)NTT未来ねっと研究所勤務。分散システム、P2Pネットワーク、データベース等の研究に従事。



大畑 真生

2007年東京大学大学院情報理工学系研究科修士課程修了。同年株式会社NTTドコモ入社。以来、先進技術研究所にてOS、アプリケーション等、端末関連の研究に従事。現在、移動機開発部勤務。



太田 賢 (正会員)

1998年静岡大学大学院博士課程修了。1999年NTT移動通信網(株)入社。現在、NTTドコモ先進技術研究所勤務。モバイルコンピューティング、端末セキュリティ、分散システムに関する研究に従事。訳書『コンピュータネットワーク第4版』等。博士(工学)。電子情報通信学会会員。



稲村 浩 (正会員)

1990年慶應義塾大学大学院理工学研究科修士課程修了。同年日本電信電話(株)入社。1994年から1995年にカーネギーメロン大学計算機科学科にて訪問研究員。1998年よりNTTドコモ。2010年慶應義塾大学大学院開放環境科学専攻後期博士課程単位取得退学。モバイル環境におけるシステムソフトウェア、トランスポートプロトコルに関する研究開発に従事。博士(工学)。電子通信情報学会、ACM各会員。



水野 忠則 (フェロー)

1969年名古屋工業大学経営工学科卒業。同年三菱電機(株)入社。1993年静岡大学工学部情報知識工学科教授。1996年同情報学部情報科学科教授。2006年同創造科学技術大学院教授。2011年より愛知工業大学教授。情報ネットワーク、モバイルコンピューティング、ユビキタスコンピューティングに関する研究に従事。著訳書としては『コンピュータネットワーク』(日経BP)、『モダンオペレーティングシステム』(ピアソン・エデュケーション)等がある。工学博士。電子情報通信学会、IEEE、ACM、Informatics Society各会員。



峰野 博史 (正会員)

1999年静岡大学大学院理工学研究科修士課程修了。同年日本電信電話(株)入社。NTT サービスインテグレーション基盤研究所を経て、2002年より静岡大学情報学部助手。2006年九州大学大学院システム情報科学府博士課程

修了。2011年より静岡大学情報学部准教授。モバイルコンピューティング、ヘテロジニアスネットワークコンバージェンスに関する研究に従事。博士(工学)。電子情報通信学会、IEEE、ACM各会員。