プライベートクラウドを利用したセンサデータ蓄積基盤の試作

川谷, 卓哉
九州大学工学部電子情報工学科

伊東, 栄典
九州大学情報基盤研究開発センター

出版情報：IPSJ SIG Technical Report, 2014-03-05
バージョン：

プライベートクラウドを利用した センサデータ蓄積基盤の試作

川谷卓哉[†1]　伊東栄典[†2]

無線通信技術の発達と通信環境整備が進み，標準規格に基づく無線モジュールを用いことで，容易に無線通信を用いる情報システムを構築できるようになっている．センサを複数，広範囲に分散設置してセンサネットワークを構築し，環境を網羅的に計測することも可能になっている．我々はセンサを用いた情報システムにおける，センサデータの処理基盤と，大規模センサシステムからの知識発見に興味を持っている．本研究では，汎用的なセンサシステムのために，センサデータを蓄積する基盤，センサクラウドを試作した．センサクラウドの要求要件を検討し，検討結果に基づいて，九州大学のプライベートクラウド上に試作センサクラウドを構築した．また試作センサクラウドの性能評価も行った．

A prototype of sensor cloud system on university private cloud

TAKUYA KAWATANI[†1]　EISUKE ITO[†2]

Recently, a lot of sensor system and sensor application are researched. The authors are interesting in general purpose sensor cloud system, which accept data from massive sensor boxes, and archive them in short time. In this paper, they report requirements for sensor cloud system. They constructed a prototype of sensor cloud system on the campus cloud system in Kyushu University. The sensor cloud part consists of data entry part and storage part. This paper also reports results of performance analysis of the prototype sensor cloud.

1.　はじめに

電子センサは，自然現象を電気信号に変換する素子として，様々な計測を行うために利用されている．近年ではセンサを複数，広範囲に分散設置してセンサネットワークを構築し，環境を網羅的に計測することも可能になっている．無線通信技術の発達と無線通信環境の整備が進み，標準規格に基づく無線モジュールを用いことで，容易に無線通信を用いる情報システムを構築できるようになっているため，通信可能なセンサ機器を用いたセンサネットワークの研究が行われている[1]．センサネットワークによる網羅的計測の例として，自然に対するものでは地域環境モニタリングや，農業における圃場モニタリングがある．また，人に対するものでは，独居老人宅に人感センサ，温度センサ，ドア開閉センサを設置して，一定時間の動作がない場合に外部に知らせる見守りシステムが実用化されている[2]．

センサ群を用いて実世界の活動を把握し，把握した状況をサービスや効率化に用いる要求は多い．しかしながら，汎用的に使えるセンサ機器や，センサデータ蓄積・解析基盤は少ない．既存のセンサシステムやセンサネットワークシステムは，目的に特化した専用システムとして構築されているものが多い．HEMS (Home Energy Management System: 家庭内エネルギー管理システム) や BEMS (Building Energy Management System: ビルエネルギー管理システム) と呼ばれる建物内の電力使用量監視のためのセンサシステムは，他の用途，例えば農業圃場モニタリングや，センサによる自宅防犯等に援用することが困難である．

我々は汎用的なセンサデータの蓄積および解析基盤の構築を目指している[3]．本論文では，センサデータの蓄積および解析の基盤システムを「センサクラウド」と呼ぶ．

本論文では，まずセンサクラウドを中心とするセンサシステムの要素および機能について検討した．検討に基づいて，センサクラウドを試作した．また試作センサクラウドの性能評価を行った．

2.　センサシステムの要素と機能要件

本研究では，センサシステムを，センサ機（センサハードウェア），センサクラウド，アプリケーションの３つの要素に分ける．それらを図 1 に示す．

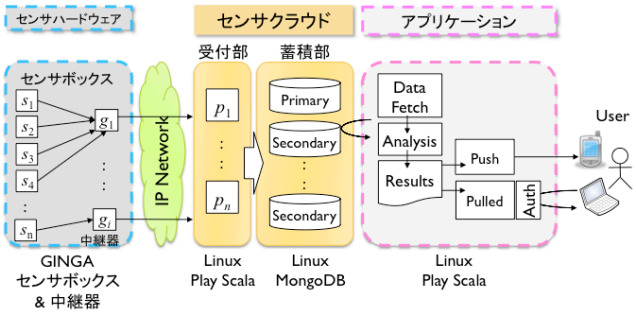

図 1　センサシステムの３要素

†1 九州大学工学部
Faculty of Engineering, Kyushu University
†2 九州大学情報基盤研究開発センター
Research Institute for IT, Kyushu University

2.1 センサ機

センサ機は，環境を計測するセンサ部品と，その計測値を外部に送信する機能を持つ．センサシステムを構成するためには，センサ機はインターネットに接続し IP でのデータ通信可能でなければならない．

この要件を満たすセンサ機として，本研究の共同研究企業社から「GINGA」と呼ぶセンサ機を提供していただいた．

GINGA はセンサボックスと中継器から構成される．センサボックスには加速度センサと温度センサが付属し，中継器との無線通信モジュールを持つ．センサボックスと中継器の間の通信は，通信は ZigBee (IEEE 802.15.4) 規格で実現されている．中継器は，センサボックスとの無線通信モジュール，および Ethernet アダプタを持つ．センサボックスは中継器に対して，一定間隔でセンサデータを送信する．中継器はデータを受信すると，JSON (JavaScript Object Notation) テキストに整形してセンサクラウドに送信する．送信プロトコルは HTTP である．

2.2 アプリケーション

アプリケーションは，センサクラウドに蓄積されたデータを読み出して解析処理を行い，結果を利用者に有用な形で出力する．アプリケーションは分野毎に多様になる．

2.3 センサクラウド

センサクラウドは次の 4 機能が必要である．

- センサ機が送信するデータの受信機能
- 受信データの整形および参照解析用情報の付加機能
- 整形後データの蓄積機能
- アプリケーションのためのデータ読出し機能

データの受信機能がなければセンサクラウドは成り立たない．そのため，センサクラウドには，センサ機が用いる送信プロトコルおよび送信データ形式と適合する API (Application Programming Interface) が必要である．また，センサ機が送信する多数のデータを欠落させることなく受信しなければならない．

情報付加および整形機能は，センサ機が送信するデータを蓄積側の形式への変換し，かつデータ受信時刻を追加する機能である．本研究で用いる GINGA では，センサボックスはボックス内にある複数種類のセンサ計測値を一括送信する．センサクラウドは，受信したデータを蓄積形式に合わせるために，計測データをセンサ種類別に分離する．また，データ受信時刻を追加する．GINGA センサ機は時計を保有していないため，時刻データの付与が必要である．なお，センサ機が時計を有するか否かによらず，受信時刻のデータは時系列整理のために必要である．

データ蓄積機能も重要である．センサが計測したデータを順次蓄積することにより，大規模なデータを用いた解析を行い，知識発見を試みることができる．

整形後データの蓄積機能では，規模適応性が重要である．センサクラウドが単位時間あたりに受信するデータ量は，接続するセンサ機の数に比例して増加する．センサシステムを運用するうちに，センサ機の数が増加することは有り得る．したがって，センサクラウドは，センサ機からのデータ受信処理スループット増大に対処できなくてはならない．その方法の一つは処理システムの分散化である．

図 1 に示した通り，センサクラウドの構成要素は，データ送受信処理を担う受付部と，データベースを格納する部分である蓄積部に大別できる．まず，最小単位の構成として，受付部と蓄積部を一つのマシンに構築することを考える．受信スループット向上には，マシンの処理性能向上で対応してもよい．しかしマシンの性能向上には物理的あるいは金額的な限界がある．別の方法として，処理の分散化がある．受付部と蓄積部を別のマシンに分離し，さらに受付部，蓄積部の内部で，処理をさらに別のマシンに並列分散させることで，システム全体の処理能力を向上できる．

蓄積したデータは，長い期間，保持し続けなければならない．そのため，データ量増大に容易に対応できなくてはならない．データを蓄積し続けると，その量は膨大なものになる．

アプリケーションのためのデータ読み出し機能でも，規模適応性は重要である．図 1 に示す通り，アプリケーションはデータ解析のために，センサクラウドからデータを読み出す．複数のアプリケーションが想定でき，アプリケーションを使う利用者数も膨大になると考えられる．多数のアプリケーションが同時にデータ参照しても，参照性能が低下しないことが望ましい．

3. センサクラウドの試作

3.1 システム構成

図 2 に試作センサクラウドの構成を示す．

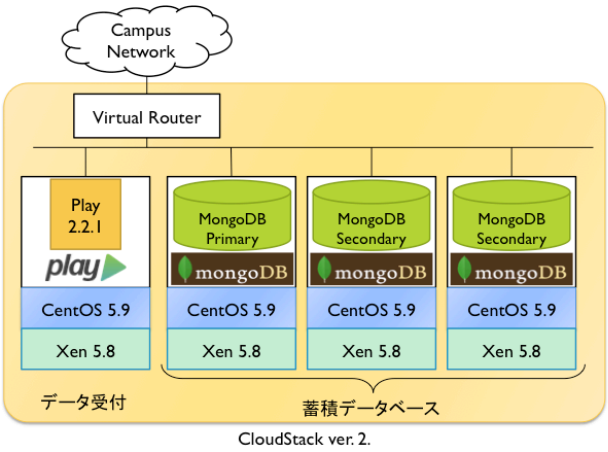

図 2 試作センサクラウドの構成概要

試作センサクラウドは，九州大学情報基盤研究開発センターが提供するキャンパスクラウドシステム（Ctirix

CloudStack 構成）を用いた．仮想マシンを要望に応じて稼働できるため，規模拡張が容易である．

図 2 の大きな四角は仮想マシンを表す．今回の試作では４台の仮想マシンを用いた．左端のマシンはデータ受付サーバで，右側にある残りの３台はデータベースサーバである．データベースサーバのうち，１台がプライマリサーバ，残りがセカンダリサーバとなっている．４台のサーバは１つのプライベートネットワークを構成し，仮想ルータを介して外部の学内ネットワークに接続している．

また，試作で用いた基盤ソフトウェアを表 1 に示す．

表 1　センサクラウドの基盤ソフトウェア

項目	ソフトウェア
Web Framework	Play Framework 2.2.1 with Scala 2.10.2
Data base	MongoDB 2.4.9 (x86_64-2.4.9)
Hypbervisor	Citrix XenServer 5.8
Guest OS	CentOS 5.9

3.2　データ受付部

試作センサクラウドのインターフェイスは，REST (Representational State Transfer) 形式の API を持つ Web アプリにしている．それを実現するために，Play Framework を用いた．Play Framework は Web アプリケーションを作る枠組みの 1 つである[4]．本システムでは，センサ機からのデータを受け付け，整形処理してデータベースに格納する部分の構築に用いた．Play Framework 上のプログラミング言語には，Scala 言語（バージョン 2.10.2）[5]を用いた．

表 2 に，試作システムで用いた REST API を示す．この API は，センサ機 GINGA の通信するデータ形式に適する形に定めた．

表 2　GINGA Web API 仕様

処理	Method	Path, Parameter
データ登録	HTTP GET	/ginga/sol?mode=entry&v=
データ取得	HTTP GET	/ginga/sol?mode=getdata&v=

3.3　データ蓄積部

センサ値データを蓄積するデータ蓄積部は，MongoDB を用いて構築した．MongoDB は，ドキュメント型データベースと呼ばれ，複数の key / value を組としたドキュメントの単位でデータを管理できること，リレーショナルデータベースと比較してデータ追加処理が高速であること，データベース規模拡大が容易に行えることといった特徴がある[6,7,8]．これらの特徴が，データ量が急速に増大する可能性のあるセンサデータの格納に適していることから，本システムのデータベース部分に用いた．

試作の際，データ蓄積部は MongoDB をインストールした仮想マシン 3 台で構成した．1 台をプライマリサーバ，残り 2 台をセカンダリサーバとするレプリカセット構成にしている．レプリカセットとは，MongoDB におけるデータベース多重冗長化の仕組みである．データ書き込みはプライマリサーバに対して行う．セカンダリサーバは定期的にプライマリサーバに問い合わせ，プライマリの更新結果を複製する．プライマリサーバが動作を停止した場合は，セカンダリサーバのうち 1 台が自動的にプライマリサーバに昇格して，処理を継続する．

レプリカセット構成時のデータ読み出しは，設定により「プライマリのみ」,「プライマリ・セカンダリのうち最速応答のサーバ」,「セカンダリのみ」を選択できる．セカンダリサーバからの読み出しを許可する場合，複数の読み出しアクセスの負荷分散を行うことができる．ただし，プライマリ・セカンダリ間の同期に時間差があるため，サーバ間で異なるデータが格納されている事態も起こりうる．本研究の例においては一度蓄積されたセンサデータを更新することはないため，時間差による問題は生じない．

3.4　センサクラウドの動作

この項では，センサクラウドの動作を説明する．図 3 にタイムラインの形で, GINGA センサボックスおよび中継器と，センサクラウドの処理を示す．

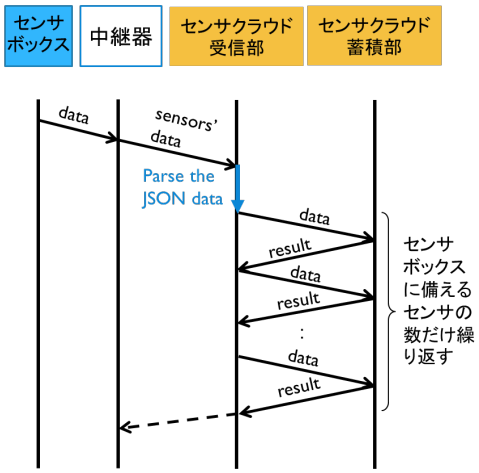

図 3　センサシステムのデータ処理ダイアグラム

センサクラウドのデータ受付部は，センサデータを受信し，データが形式に従っているか調査する．形式が異なる場合，そのデータは破棄する．データ形式が正しい場合，受信データを蓄積側に適した形に組み換えて，そのデータを MongoDB に蓄積する．

4. 試作センサクラウドの性能評価

4.1 性能評価方法

センサクラウドは，多数のセンサデータを実時間で処理できなければならない．多数のセンサ機からのデータ送信およびデータ受信実験は不可能であるため，ソフトウェアによるダミーデータの投入で，試作したセンサクラウドの性能評価を行った．

テストデータを生成するために，Ruby 言語によるデータ取得加工プログラムを作成した．気象庁ウェブサイトの気象観測データを一覧表示するページにアクセスし，得られた HTML ファイルから温度，湿度，気圧のデータを取り出した．次に，取り出した 3 つのデータを GINGA Web API の形式に適合するように，JSON 形式に整形した．すなわち，テストデータは GINGA センサボックスに対して，センサ素子を 3 個装着したものと同等となる．

性能評価実験の環境を図 4 に示す．センサボックスと中継器の部分を自動データ送信プログラムに置き換えている．

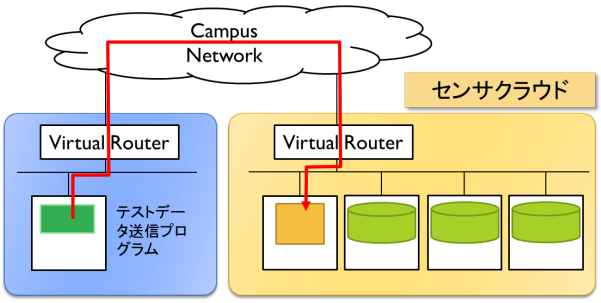

図 4 性能評価実験の環境

処理時間の計測は，図 5 の中引出線で示した区間で行った．センサクラウド受付部にデータが着信した時点から，センサデータ蓄積処理が終わり，その処理結果が受付部に返される時点までである．この実験を，連続して 3 回行った．各試行の間に，データベースを初期化するなどの変更は加えていない．

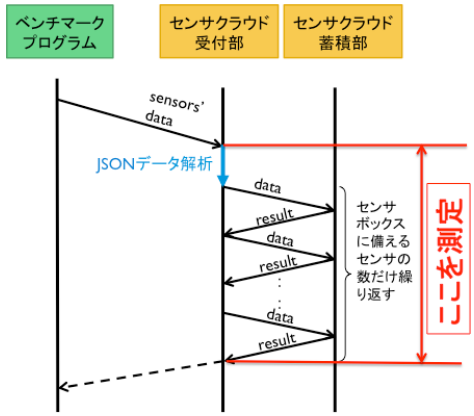

図 5 測定した処理時間

4.2 計測結果

表 3 に，3 回の試行結果を示す．また，図に各試行の度数分布表を

表 3 第 1 回試行の結果

処理時間	データ件数		
	第 1 回	第 2 回	第 3 回
2ms 未満	81,234	79,950	76,336
2…5ms	18,292	19,568	23,069
5…10ms	202	194	277
10…50ms	248	250	288
50ms 以上	24	38	30

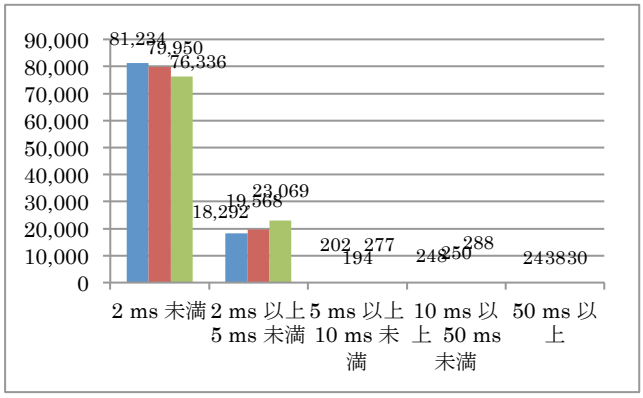

図 6 各試行の処理時間分布

本実験において，センサクラウドは送信データ 1 件あたり平均 1.32 ms でデータ蓄積処理を行うことができた．この平均処理時間から求めた，センサクラウド内部のスループットは，毎秒 757.58 件である．700 個程度のセンサ機を用いたセンサシステムであれば，試作センサクラウドは実時間処理を行うことができると思われる．

しかしながら，実際のセンサデータ通信は，ネットワークの状態によっても通信時間が異なる．一時に多数のセンサ機からのデータ同時に着信すると，処理溢れが発生する可能性もある．より正確なスループット数を計測する必要がある．

5. おわりに

本研究では，センサデータを大量に蓄積するための基盤システムであるセンサクラウドの試作を行った．試作は，九州大学のキャンパスクラウドシステムで構築した．試作センサクラウドは，規模適応性を持つように，分散処理可能な構成にしている．

試作したセンサクラウドの性能評価も行った．実際のセンサ機を多数用意しての性能評価は実現不可能であったため，ダミープログラムを使って性能評価を行った．評価実

験の結果，試作センサクラウドは 1 秒間あたり 757.58 件のスループットを有することを確認した．このことから，センサハードウェアとセンサデータ蓄積処理システムを，IP ネットワークを介する形で分離し，さらに処理システムのハードウェアを仮想化しても，1 秒未満の遅延で計測データを蓄積できることが明らかになった．

ただし，複数の送信元からの着信に対する分散処理や，通信接続・切断処理に必要な時間を考慮して，より実運用に近い形でのスループット評価を行う必要がある．

今後の課題は，センサクラウドに関することと，センサデータに対する応用的研究の二つがある．まずセンサクラウドについては，試作システムを拡張することで，受付部の並列化負荷分散，および蓄積部のデータ書き込み処理の負荷分散を実現し，システム性能の向上を大規模な入力によって試す必要がある．また，実際のセンサ機での実証実験も必要である．将来は，蓄積したセンサデータを解析しての，有用な知識の抽出も研究したい．

参考文献

1) Aggarwal, C.C. “Managing and Mining Sensor Data”, Springer US, S.l., 2013.
2) David Boswarthick, Omar Elloumi, Olivier Hersent (編)，山﨑徳和，小林中 (訳)，「M2M 基本技術書 ETSI 標準の理論と体系」，リックテレコム，2013.
3) Takuya Kawatani, Eisuke Ito, Kensuke Baba, “A propose of real-time seats usage analysis using smart sensors for library marketing”, IIAI AIT 2013, 2013.
4) Play Framework，http://www.playframework-ja.org/，(accessed 2013.12.10).
5) The Scala Programming Language，http://www.scala-lang.org/，(accessed 2014.01.20).
6) MongoDB，http://www.mongodb.org/，(accessed 2013.12.10).
7) 太田洋，本橋信也，河野達也，鶴見利章，「NOSQL の基礎知識 ビッグデータを活かすデータベース技術」，リックテレコム，2012.
8) 松下雅和，桑野章弘，「MongoDB 実践入門」，WEB+DB PRESS vol.75, pp. 48-79, 2013.