情報処理学会論文誌 Vol.54 No.2 672–687 (Feb. 2013)

センサークラウド：センサデバイスを IT資源とする拡張クラウド環境

高橋 ひとみ[1,a)] 串田 高幸[1,b)]

受付日 2012年5月9日, 採録日 2012年11月2日

概要：近年，多種のセンサデバイスが生産され，これらのデバイスを用いたアプリケーション群が多く出現している．しかしそれらの使用方法は資源が乏しくかつ多種多様なプロトコルを使用しているセンサデバイスへ直接接続を行い，センサデータを取得する手法が一般的である．そのため，複数のユーザでセンサデバイスの共有ができない，どのようなセンサデバイスが存在するか分からない，透過的な接続ができないといった問題が発生する．そこで本論文は，クラウド環境下で CPU やストレージ，ネットワークといった通常の IT 資源と同様に，センサデバイスをクラウド上の IT 資源の 1 つとして扱うセンサークラウドを提案する．センサークラウドはクラウド環境の機能である仮想化，自動化，標準化の機能をセンサデバイスへ適用させ，利用者および開発者のセンサデバイスの共有化，センサデバイスの一覧化および透過的な接続を可能とする．本センサークラウド環境では従来の手法であるセンサデータの共有ではなく，センサデバイス自体を仮想化し VM 上で仮想センサデバイスとして扱えるため，利用者，開発者の要求に合わせた柔軟な使用センサデバイスの取捨選択，センサデバイスの共有，状態取得も，複数のユーザ間で可能となる．本論文ではセンサークラウド環境のプロトタイプ実装を行い評価し，センサークラウド環境を用いることで，多くの利用者，開発者へのセンサデバイス共有が可能になることを示した．

キーワード：センサデバイス，クラウドコンピューティング

Sensor Cloud: Extended Cloud Computing Enviroment to Use Sensor Devices as IT Resources

Hitomi Takahashi[1,a)] Takayuki Kushida[1,b)]

Received: May 9, 2012, Accepted: November 2, 2012

Abstract: We propose “Sensor Cloud” which is a cloud computing environment where sensor devices are managed as one of IT resources in the same way as CPU, storages, and networks. In Sensor Cloud, we can adapt three functions of cloud computing to sensor networks; 1) Virtualization, 2) Automation, 3) Standardization and provide sensor network infrastructures which user can share easily. In existing sensor networks, users connect to sensor devices directly and some problems occur that multiple users cannot share a sensor device, acquire information of sensor devices, and connect among various sensor devices seamlessly because sensor devices have various protocols and their resources are much poor. Sensor Cloud can provide three functions to such sensor devices: 1) Multiple users can share one sensor device. 2) Information of sensor devices can be gathered in a catalog. 3) User can connect to many sensor devices seamlessly. In our system, multiple users can select the sensor devices they want to need, and acquire properties and sensing data of sensor device because users share not sensor data but a sensor device itself in VM by virtualizing sensor devices. In our paper, we have implemented prototype of Sensor Cloud and evaluated it. We show that Sensor Cloud enables many users to share one sensor device.

Keywords: sensor device, cloud computing

1. はじめに

近年，IOT（Internet of Things）[1] への注目が高まっており，あらゆるものにセンサデバイスを設置しネットワークと結合させ，利便性の高いアプリケーションが数々と創出されている．IOT のような環境では設置する物，もしくは設置する環境に特化された，数々のセンサデバイスが出現している．資源が乏しく様々な種類のセンサデバイスが存在する環境下では以下の 3 つの問題が発生する．1. 資源が乏しいため 1 人の利用者，開発者がセンサデバイスを占有し，複数の利用者，開発者が同時に使用できない．2. センサデバイスに対する標準化がされておらず，どこにどのようなセンサネットワークが存在するか利用者，開発者は把握できない．3. センサデバイスごとに独自の規格や API が存在し，アプリケーションは各センサデバイスに特化したものとなる．

本論文では上記の問題を解決するためにセンサデバイスを仮想化し，クラウドコンピューティング環境の 1 つの IT 資源として扱うセンサークラウドを提案する．センサークラウドは以下の機能を実現する．1. センサデバイスの仮想化により複数の利用者，開発者が同時にかつ透過的にセンサネットワークへ接続できる．2. センサデバイス群を使用頻度が高い，もしくは意味のあるグループに分け，それに合わせたミドルウェア，アプリケーションをセットでテンプレート化し標準化を行う．さらに，テンプレートをカタログに登録することで，そのテンプレートに登録されているセンサデバイスやセンサネットワークの情報を利用者，開発者が参照できる．3. 仮想マシン設定の自動化により自動的に VM のプロビジョニングを行い，VM 内のセンサデバイスや，アプリケーションの設定を行うことで，利用者，開発者はすぐにセンサアプリケーションの使用や開発が可能となる．

通常のクラウドコンピューティング環境は CPU，HDD，メモリといった一般的な IT 資源に対して仮想化，標準化，自動化の機能を有し，多数のユーザへ透過的にかつ柔軟性の高いマシン環境を提供する [2], [3]．本システムはセンサデバイスを 1 つの IT 資源としてクラウド環境へ取り込み，通常のクラウド環境下で IT 資源に適用される，仮想化，標準化，自動化の機能をセンサネットワークへ適用した．このセンサークラウドの機能により，センサデータのみの共有ではなく，センサデバイス自体を仮想化することで，容易に複数の利用者，開発者へセンサデバイス自身の共有が可能となる．

本論文では役割の異なるロールが複数出てくるため，以下にロールを定義する．

利用者　センサアプリケーションを利用するエンドユーザ

開発者　センサデバイスを使用したアプリケーションを開発するユーザ

クラウド管理者　センサークラウドの管理を行うユーザ

センサネットワーク所有者　センサネットワークを所有しているユーザ

本章以降は次の構成をとる．まず「1. はじめに」で既存のセンサネットワークの問題について述べ，本論文が提案するセンサークラウドについての概要を説明した．「2. 関連研究」では複数の利用者，開発者がセンサネットワークやセンサデバイスを共有する手法の関連研究を述べ，「3. センサークラウドとは」ではセンサークラウドが提供する機能について説明し，「4. センサ管理技術の設計」でセンサ管理技術の設計について述べる．また「5. センサークラウドのプロトタイプ実装」において実装を行ったセンサ管理技術の評価を「6. センサークラウドのプロトタイプ実装による評価」で述べ，このシステムを用い実現できるユースケースを「7. アプリケーション」で説明する．最後に「8. まとめと今後の課題」でまとめる．

2. 関連研究

本章ではセンサークラウドのように，複数の利用者，開発者によるセンサデバイスの共有化が可能となる関連研究をあげる．既存のセンサデバイスの共有化としてネットワークによる共有化，センサデータによる共有化，テストベッドによる共有化，クラウドによる共有化の 4 つに分類し，それぞれの関連研究について説明する．

関連研究を説明するにあたり，本論文で使用するセンサに関する語を以下に定義する．センサデバイス：センサデバイスのハードウェア単体．センサネットワーク：複数のセンサデバイスにより構築されるネットワーク．センサデータ：センサデバイスより取得できるセンシングデータ．

ネットワークによる共有化

ネットワークによる共有化手法は，センサデバイスやセンサネットワークへのアクセス方法を既存のプロトコルに適用させ，複数の利用者，開発者からの接続を可能としている．この手法では API やネットワークプロトコルが共通となるため，利用者，開発者はセンサデバイスの差異を考慮せず透過的な接続が可能となる．まず Gaynor らは，既存の GRID コンピューティングで使用されている規格である OGSA を，センサネットワーク上でも使用可能となるよう拡張した [4]．センサネットワークを使用する利用者は OGSA のアプリケーションを使用することで，センサデータの取得や加工，解析が，センサネットワークを含むグリッドコンピューティング上で行われ，利用者は結果を受信できる．API や接続手法が共通になるため，複数の利

[1] 日本アイ・ビー・エム株式会社東京基礎研究所
IBM Research - Tokyo, IBM Japan, LTD., Koto, Tokyo 135–8511 Japan
a) hitomi@jp.ibm.com
b) kushida@jp.ibm.com

用者が OGSA に対応したアプリケーションを用い，センサネットワークを含むグリッドコンピューティングを利用できる．

次に Dunkels らは，プロキシを用いセンサネットワーク上に仮想的なネットワークを構築した [5]．このプロキシとなるノードは，通常の TCP/IP スタックおよびセンサネットワークのプロトコルを有しているため，利用者，開発者は通常のインターネットで使用するアプリケーションを用い，個々のセンサデバイスへアクセスが可能となる．そのためこのプロキシを介することで通常のネットワークアプリケーションから，複数の利用者，開発者が個々のセンサデバイスへ透過的にアクセスできる．

これらの 2 つの手法はセンサデバイスへの接続を透過的にすることで，ユーザは容易にセンサデバイスへの接続が可能となる．しかし，利用者，開発者はセンサデバイスへ直接接続を行うため，センサデバイスのバッテリや CPU などの資源が乏しい場合，大勢の利用者，開発者によるセンサデバイスの共有はできない．

センサデータによる共有化

次にセンサデータによる共有化手法の関連研究をあげる．センサデータの共有化ではセンサデバイスから取得したセンサデータを DB 上に保持し，その DB へ多くの利用者，開発者がアクセスすることで共有を行う．

Grosky らは，センサデバイスから取得した気象データや地域情報データを収集かつ DB に格納することで，センサデータの共有化を可能にしている．利用者，開発者は地理情報と紐づいた気象データや，センサデータ単体が Web を介して取得できる [6]．

また Rowe らはキャンパスに設置したセンサデバイスのデータを，キャンパスネットワーク内に存在する DB に収集し，キャンパスエリアネットワーク内でセンサデータの共有が可能なシステムを構築した．このシステムでは XMPP を用いたセンサデータの取得手法を採用し，実際にシステムを動作させ実証実験を行っている [7]．

これらの手法はセンサデータのみの共有となり，センサデバイス自体の共有を行う本研究とは異なる．センサークラウドではセンサデバイスを仮想化しクラウド環境下に組み込むことで，経路情報やバッテリ残量，データの送信出力，受信電波強度というような，センサデータではないセンサデバイスの状態を利用者，開発者が取得可能となる．特にマルチホップセンサネットワークでは，今まで受信していたセンサデータが突然配送されなくなる状況が発生し，利用者，開発者はセンサネットワークの現状を確認するため，経路の中間ノードの状態を把握する要求が出てくる．センサネットワークをリモートで利用している場合，直接センサデバイスの状況を確認することは困難な場合が多く，オンラインでセンサデバイスの状況を取得する必要がある．しかし，センサデータ共有のみではセンサデバイス自体の情報は取得できないため，センサデータが突然配送されなくなった場合，その状態がすぐに終了するのか，もしくは永続的に継続するのかまったく判断ができない．本機構は，利用者，開発者が使用しているデバイスのみの経路情報や，その経路に含まれる中間センサデバイスの状態も監視および管理を行い，利用者，開発者からの問合せに対するセンサデバイスの情報を，センサークラウドが応答する．これにより，利用者，開発者はオンライン上から，自分が使用しているセンサデバイスの状態を把握できる．

テストベッドによる共有化

既存の研究では様々な技術を組み合わせ，実際に動作するセンサネットワークのテストベッドを提案している論文がある．Achtzehn らはセンサデバイス上のソフトウェアの開発を，複数の開発者で効率的に行えるためのセンサネットワークテストベッドを提案している [8]．また Werner-Allen らは Web サーバやデータベースを用いセンサデータを複数の開発者で共有し，さらにセンサデバイス自体を共有化することで，センサデバイス上のソフトウェアの開発がリモートから，かつ複数の開発者で可能となるテストベッドを提案している [9]．これらの研究ではセンサデバイス自体の共有化は行っているが，センサデバイス上のソフトウェア開発が目的である．そのため，関連研究ではセンサデバイス上のソフトウェアをコンパイルし再ロードが可能であるが，センサデバイスを開発者がネットワークを介して直接接続するため，資源が乏しいセンサデバイスでは複数の開発者が同時にセンサデバイスを共有できない．本研究ではセンサデバイス自体を仮想化し，複数の利用者，開発者が多種多様なセンサデバイスの状態やセンサデータを透過的に取得可能とすることが目的であるため，センサデバイスにおけるソフトウェアの変更は対応していない．

クラウドによる共有化

本論文と同様にクラウド環境を用い，センサネットワークを共有する関連研究をあげる．まず，Hassan らはクラウド上に存在するセンサアプリケーションのサービスを通じ，利用者がセンサデータを共有するフレームワークを提案している．このフレームワークでは，センサアプリケーションがクラウド管理側に存在する Pub/Sub Broker へ希望するセンサデータを登録し，Pub/Sub Broker が条件に合うセンサデータをアプリケーションへ送信する．利用者はクラウド上に存在するセンサデータを含むセンサアプリケーションをクラウドを介し使用できる [10]．次に Kurschl らは，クラウド上にセンサデータの分析や加工を行うミドルウェアとセンサアプリケーションを含むサービスを用意し，利用者はそれらの希望サービスを選択することで，センサデータを使用できる [11]．

これらの関連研究はセンサデバイスの共有ではなく，センサデバイス上で動作するアプリケーションやサービス，センサデータを共有する手法である．クラウド上ではこのような利用形態を SaaS（Software as a Service）と呼び，センサアプリケーションの使用を目的とした利用者に合致した共有手法となる．一方，本論文で提案するセンサークラウドは，センサデバイス自体を仮想化し利用者へ提供する IaaS（Infrastracture as a Service）である．センサアプリケーションに利用するミドルウェアの開発や，センサアプリケーション自体の開発を目的としている開発者に対しては，センサデバイス自体を仮想化し開発者へ提供する本機構が適している．また，センサークラウドは IaaS として提供できるため，クラウド管理者がセンサデバイスのミドルウェアやアプリケーションをすべて含んだテンプレートを作成することで，SaaS としても利用者に提供が可能である．そのため，本機構はアプリケーションを使用するだけの利用者，センサアプリケーションを作成する開発者のどちらにでも，仮想センサネットワーク環境を提供できる．

センサデバイスおよびセンサデータを複数の利用者，開発者で共有できる機構は数多く提案されているが，本機構のようにセンサデバイス自体を仮想化し，クラウド環境の IT 資源として共有する手法は提案されていない．本手法を用いることにより，複数の利用者，開発者によるセンサデータの取得はもちろん，使用しているセンサデバイス自体の状態も把握できる．センサネットワーク所有者はそれぞれのネットワークで異なるため管理の質も異なる．もし，センサデータの取得が不可能となった場合，使用利用者，開発者がそのセンサデバイスの状態を取得し，使用を続行するか代替を検索するかといった判断をする必要も出てくる．センサークラウドでは実デバイスの状態もプロビジョニングされた VM から把握でき，センサデバイス使用の判断が利用者，開発者に可能となる．さらにセンサデバイスを仮想化し IT 資源として扱うことで，既存のクラウド環境で使用していた監視や課金のフレームワークもそのままセンサデバイスに適用できる．

ただしセンサデータやデバイス状態を，データ要求の問合せを行い取得できるようなセンサデバイスにおいては，多く取得要求を行うとセンサデバイスのバッテリ消費が大きい．センサークラウド側でセンサデータやセンサデバイス情報をキャッシュしデバイスへの負担を減少させているが，開発者，利用者は必ずしもリアルタイムのデータが取得できない場合がある．そのため，センサデータ取得遅延が許容されるアプリケーションが，センサクラウドの適用範囲となる．また，ネットワーク上からセンサデバイスに搭載されたソフトウェアの変更には，センサークラウドでは対応していない．

3. センサークラウドとは

本論文が提唱するセンサークラウドとはクラウドの IT 資源としてセンサデバイスを扱い，クラウド環境の特徴である IT 資源の仮想化，自動化，標準化というフレームワークをセンサデバイスへ適用し，複数利用者，開発者によるセンサデバイスの共有化を実現する．

センサークラウドでは，アプリケーションやミドルウェアを行う開発者，またはセンサアプリケーションを利用するエンドユーザである利用者のどちらも対象ユーザとする．そのため本機構の有用性を考慮した場合，センサクラウドにおいてセンサデバイスを利用する開発者，利用者は，センサデバイスの深い知識の必要なしに，希望するセンサデバイスおよびアプリケーションがすぐに VM 上で使用できる環境を提供する必要がある．一方，センサネットワーク所有者がセンサデバイスをセンサークラウドの IT 資源として提供してもらうためには，センサネットワーク所有者へのメリットが必要となる．そこで，利用者，開発者のセンサデバイス使用代金をセンサネットワーク所有者に還元し，さらにセンサデバイスの監視システムを提供することで，センサネットワーク所有者にもセンサデバイスを IT 資源として提供するメリットが生まれる．これよりセンサークラウドでは，センサデバイス利用環境，およびビジネスモデルを確立できる機能を提供する必要がある．

まずセンサデバイス利用環境の構築のために，センサークラウドでは 3 つの機能を実現する．1. センサデバイスの仮想化を行い，複数の利用者，開発者が同時にかつセンサデバイスの差異を意識せずセンサネットワークへ接続できる．2. センサデバイスおよびセンサネットワークをよく使用されるグループに分け，ソフトウェア群とともにテンプレート化しカタログへ登録することで，利用者，開発者はセンサデバイス情報を集約したカタログから参照できる．利用者，開発者はこのカタログより自分の使用したいセンサデバイスグループやアプリケーションを自由に選択できる．3. VM の設定自動化により，利用したい仮想センサデバイス，センサアプリケーションおよび必要なライブラリを含む VM が自動的にプロビジョニングされ，それらのソフトウェアが自動的に設定されることで，利用者，開発者はすぐにセンサアプリケーションの使用や開発ができる．これらの機能をセンサ管理技術として定義する．

これらの機能は通常のクラウドコンピューティングの機能にあたるが，センサ管理技術において，CPU などの既存の仮想化技術をそのまま使用できない．たとえば，CPU を仮想化するソフトウェアとして VMWare があげられるがこのソフトウェアはセンサデバイスの仮想化には対応していない．また VMWare の設定や処理を命じるクラウドマネージャも一般的なソフトウェアに対する制御や管理のみとなり，センサデバイ

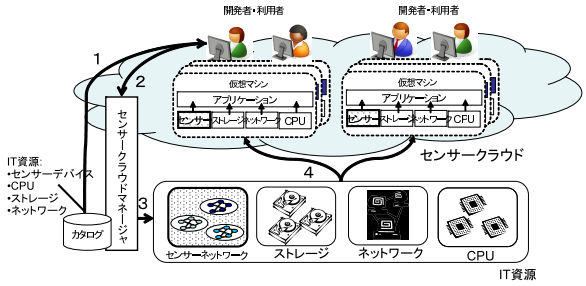

図 1 センサの管理技術
Fig. 1 Management of sensors.

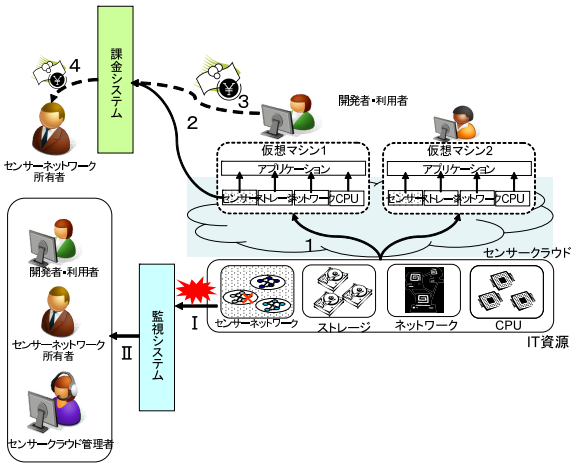

図 2 センサ使用への課金技術および監視技術
Fig. 2 Metering and monitoring of sensors.

ス用ハイパーバイザの管理や制御機能を持たない．そのため，センサークラウドではセンサデバイス用のハイパーバイザ，そのソフトウェアの制御を行う管理モジュール，センサデバイス用の情報を格納する機構が必要となる．

3.1 センサ管理技術

上記のように，センサデバイス用のハイパーバイザを実現し，IT 資源の 1 つとしてセンサデバイスを見なしクラウド環境の機能を適用することで，センサデバイス群を即座にかつ簡便に使用できる．

図 1 にセンサークラウド上で利用者，開発者がセンサデバイスを使用するまでの大まかな手順を図の番号と対応させ説明する．

1. センサークラウド上のカタログにはセンサデバイスの情報（e.g. 取得データ，センサデバイスの種類，取得データフォーマット，ロケーション，ハードウェアの仕様）が集約されており，利用者，開発者はこれを閲覧できる．
2. 利用者，開発者はカタログから VM 上で使用するセンサデバイスを VM のテンプレートとともに，センサークラウドマネージャに通知する．
3. 利用者，開発者からのリクエストを受けたセンサークラウドマネージャは，ユーザが指定した性能どおりの VM をプロビジョニングするため IT 資源の設定を行う．
4. 利用者，開発者が使用要求したセンサデバイスを仮想化し，仮想センサデバイスとして VM 上に追加され VM がプロビジョニングされる．利用者，開発者は他の IT 資源と同様にこの仮想化されたセンサデバイスを実センサデバイスのように使用できる．

3.2 センサ使用への課金技術

センサークラウドでのビジネスモデルを確立するため，センサデバイス所有者へセンサデバイスの使用料金を開発者・利用者から還元する必要がある．センサークラウドではクラウドの管理を行うクラウドマネージャが，プロビジョニングされた VM における IT 資源の利用状態を把握できる．そこで通常のクラウド上で利用者，開発者へ課金を行う既存システムやフレームワークへ，センサデバイスの使用量に応じたセンサ使用の課金機能を追加する．

図 2 に課金技術の手順を図の番号と対応して説明する．

1. センサデータの配送状態を記録する．
2. 各 VM が使用しているセンサデータ量などの情報はセンサークラウドマネージャが収集する．
3. 課金システムがあらかじめ設定されたポリシに従い課金計算を行う．
4. センサデータ使用料金が利用者，開発者に課金される．
5. センサークラウドへ IT 資源としてセンサデバイスを提供した設置者や所有者へ，利用者，開発者から課金した費用を還元する．

3.3 センサデバイスへのモニタリング技術

センサークラウドのビジネスモデルを確立するため，センサデバイス所有者へセンサデバイスの監視機構を提供する必要がある．センサークラウド上ではクラウドの管理を行うクラウドマネージャが，センサデバイスを含む IT 資源の動作状態を把握できる．そこで，クラウド上の IT 資源である CPU や HDD の状態を監視し，異常検知を行う既存システムやフレームワークを拡張し，センサデバイスの状態を監視し異常検知ができるシステムを構築する．もしこのシステムがセンサデバイスの異常を検知した場合，クラウド管理者やそのセンサデバイスを使用している利用者，開発者へ異常を通知する．

図 2 にセンサへのモニタリング動作手順を図の番号と対応し説明する．

I. センサークラウドはセンサデータを配信する際，センサデータが定期的に送信されていることをセンサークラウドマネージャへ通知する．もしくはデータ要求を受信するとセンサデータを応答するようなセンサデバイスであれば，センサマネージャが定期的に生存メッセージを送信する．それらの情報はセンサークラウド

マネージャから監視システムに配送され，監視システムが定期的なデータ通信が発生しているかもしくは生存メッセージへの応答があるか判定を行う．

II. もし監視システムはセンサデバイスが動作していないと判断した場合，そのセンサデバイスを使用している利用者，開発者，センサネットワーク所有者，クラウド管理者に異常を通知する．

4. センサ管理技術の設計

本節ではセンサークラウドの設計について述べる．センサデバイスを IT 資源として扱う場合，通常の IT 資源をクラウド上で扱うのと同様な機構が必要となる．まず IT 資源を仮想化させる機構であるハイパーバイザを，センサデバイス用として構築する必要がある．このセンサデバイス用のハイパーバイザは通常のハイパーバイザと同様に，利用者，開発者が VM 上で使用するセンサデバイスとして，仮想的なセンサデバイスを構築する．利用者，開発者はこの仮想センサデバイスを介し，センサデータの取得やセンサデバイス自体のプロパティを取得できる必要がある．また，利用者，開発者が要求した任意の IT 資源であるセンサデバイスを，動的に仮想センサデバイスとして VM に割り当てるよう，センサークラウドマネージャは自動的にセンサデバイス用ハイパーバイザの設定をする必要がある．さらに，IT 資源として存在するセンサデバイスがどのようなデバイスかを取得し，センサデバイスグループをテンプレート化し，カタログへ追加できる機能が必要となる．

以上の機能要件より，センサークラウドは**図 3** で示すように以下の 3 つの機構に分かれる．1. センサデバイスの仮想化を行うセンサ用ハイパーバイザの機構および，VM の利用者，開発者がアクセスするための仮想センサデバイスモジュール．2.VM へのセンサデバイスの追加，設定の自動化機能を有したセンサークラウドマネージャ．3. 通常の IT 資源に加え，センサデバイスの情報を保持できるよう拡張されたテンプレートおよびカタログ．

さらに，センサデバイスを使用するという点から，センサデバイスを仮想化するセンサデバイス用ハイパーバイザでは，多種多様なセンサデバイスのサポートが簡単に行える必要がある．また，開発者，利用者が増加した場合でもセンサークラウド上でセンサデバイスがストレスなく利用できる必要がある．さらに，センサデバイスの資源は乏しく制約が大きいということを考慮して設計を行う．

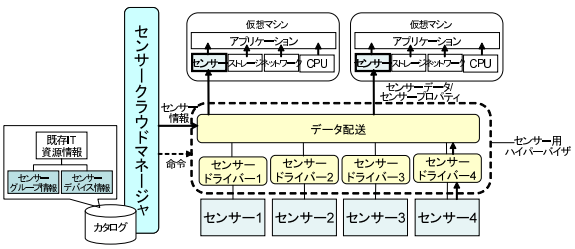

図 3 センサークラウドのシステムアーキテクチャ
Fig. 3 System architecture of sensor cloud.

利用者，開発者はセンサークラウドマネージャが提供する Web の API から，希望するセンサデバイスおよび IT 資源を選択する．その要求に基づきセンサクラウドマネージャは通常のクラウドマネージャの機能である IT 資源の割当てを VMWare といったハイパーバイザを通じて行い，VM を作成する．その際，利用者，開発者が希望したセンサデバイスが VM 上に仮想センサデバイスとして作成されるようセンサ用ハイパーバイザの設定を更新する機能を既存のクラウドマネージャに拡張することで，通常のクラウドマネージャでセンサデバイスの管理が可能となる．

センサークラウドはシングルホップ，マルチホップのネットワークを構成するセンサデバイスを両方サポート可能である．ただし，センサネットワークのゲートウェイやブリッジとなる端末から，TCP/IP のネットワークを介して，センサデバイスの情報やセンサデータを取得できる必要がある．たとえば，一般的に知られている Mote [12] デバイスの場合，ブリッジとなるノードがイーサネットに接続できる，もしくはブリッジへ USB 接続を行っているノードを介して，ネットワークからセンサデータが取得できるため，センサークラウドではサポートできる．また本機構はキャッシュやデータ配送を行うため，データ取得遅延がまったくできないハードリアルタイムを保証するアプリケーションには使用できない．たとえば猛毒ガスの発生を検知した場合，0.5 秒以内に全シャッタを閉めるといった人命にかかわるようなアプリケーションは本機構のスコープ外となる．また，センサデータの値が短時間に大きく変化し，それに追従し制御を行うようなリアルタイム性を求めるアプリケーションにも不向きとなる．各機構の詳細な設計を次に述べる．

4.1 仮想化を実現する機構

本節ではセンサデバイスの仮想化を行う機構であるセンサデバイス用ハイパーバイザについて述べる．まずセンサデバイスの仮想化とは通常の IT 資源の仮想化と同様に，利用者，開発者が直接使用するセンサデバイスを，VM 上でソフトウェアにより実現する必要がある．そこで，利用者，開発者が使用する VM に仮想センサデバイスとしてソフトウェアのモジュールを組み込む．このモジュールを仮想センサモジュールと定義する．

また，一般的なセンサデバイスは利用者，開発者がデバイスへアクセスを行うと，位置情報やバッテリの情報といったセンサデバイス自体のプロパティやセンサデータの取得が可能である．そこで，利用者，開発者が VM 上に存在する仮想センサモジュールへアクセスした際に，利用者，開発者が要求したセンサデバイスのセンサデータおよびプ

ロパティを取得できる必要がある．センサデバイスからのセンサデータを，利用者，開発者が使用している VM の仮想センサモジュールまで配送を行い，また，センサデバイスのプロパティ要求が仮想センサモジュールから送信された場合，対応するセンサデバイスのプロパティをセンサークラウドマネージャが返答する．これにより，利用者，開発者からは通常のセンサデバイスと同様に，仮想センサデバイスモジュールからセンサデータの受信やセンサプロパティの取得が可能となる．

図 3 のセンサデバイス用ハイパーバイザ内に存在するデータ配送モジュールおよびセンサドライバは，各センサデバイスからセンサデータを取得し，そのセンサデバイスの使用を要求した VM に配送する．センサデバイスは多種多様に存在するため，それらのサポートを行う際にプログラムの大きな改変が発生しないよう，センサデバイスの仕様に依存する処理をセンサドライバモジュールとし，センサデバイスに依存しない処理をデータ配送モジュールの機能と定義し設計を行った．

センサデバイスの依存部分であるセンサドライバでは，センサデバイスのアクセスを直接行う必要があるため，センサデバイスの種類ごとに実装する必要があるが，機能の大部分を占めるデータ配送モジュールや VM にインストールされる仮想デバイスモジュールには，いっさいの修正を必要とせずに新しいセンサデバイスのサポートが可能となる．センサークラウドで様々なセンサデバイスをサポートするために，センサドライバにおいて必要な機能は以下となる．

- パケットヘッダの生成：センサデータの要求に独自のデータパケットが必要な場合，そのパケットを生成する．
- センサデータおよびデバイス情報の送受信：センサデバイスからのセンサデータおよびデバイス情報の送受信を行う．
- センサデータパケットからの情報抽出：センサデバイスの独自ヘッダから必要な情報を取得し，センサデータを上位層に渡す．

センサデバイス自体の機能が少ないため，これらの機能のみで広範囲のセンサデバイスがセンサークラウドでサポート可能となる．また SNMP のようにデータ要求の仕様が標準化されているプロトコルでは，センサークラウド側でライブラリなどを提供し開発者の負担を減らす．

次にセンサデバイス用ハイパーバイザの大まかな動作手順を説明する．センサドライバモジュールは各センサデバイスの使用に合わせセンサデータの取得を行う．センサデバイスによるセンサデータの取得手法は大きく 2 つに分けられる．まず，Active に定期的なデータ送信を行うセンサデバイスの場合，センサドライバモジュールは，定期的にセンサデータが取得できるよう，センサデバイスやゲートウェイノードへ接続しセンサデータを取得する．取得したセンサデータのパケットは余分なヘッダが削除され，センサデータのみが取り出されデータ配送モジュールに渡される．次に利用者，開発者がセンサデータを要求しない限り，センサデータを送信しない Passive なデバイスの場合，あらかじめ指定されたデータ取得間隔に従いセンサドライバモジュールが，各センサデバイスの仕様に沿ったデータ要求パケットを作成して送信することで，センサデータの取得を行う．取得したセンサデータは Active なセンサデバイスと同様にセンサデータのみが取り出され，データ配送モジュールに渡される．これより，データ配送モジュールではすべてのセンサデバイスが Active なセンサデバイスとして扱い可能となる．ただし，取得したセンサデータが古くかつセンサデバイスが Passive な場合，例外的に利用者，開発者からのセンサデータ取得要求をデータ配送モジュールが受け付け，同期的なセンサデータの取得ができる．

Passive なセンサデバイスを Active なデバイスとして扱う対象となるセンサデバイスは，AC アダプタからの電源供給が可能，かつシングルホップのネットワークなど，定期的なセンサデータ要求に対しても，センサデバイスやネットワークへの負荷が少ないセンサデバイスが対象となる．このセンサデータ取得手法により，利用者，開発者が高頻度のセンサデータの要求を行っても，クラウド管理者が指定した頻度以上のセンサデータ要求が発生しない処理が簡単に実現できる．また，利用者，開発者が様々なセンサデバイスを使用するセンサアプリケーションを開発する場合，センサデータ取得手法の違いを意識せずに開発ができる．

もし，センサデバイスがバッテリ駆動であり，ネットワークトポロジがマルチホップとなる場合，不必要なセンサデータの要求はネットワークやデバイスに負荷がかかるため極力減らす必要がある．そこで，バッテリ駆動で資源が少ないセンサデバイスでは，利用者，開発者の取得要求に応じた場合にのみセンサデータの取得を行い，クラウド管理者が定めた取得頻度を超える場合はキャッシュの値を返答する処理を行うなど，センサデバイスの特性によりセンサデータの取得手法を変更する．

センサドライバモジュールよりセンサデータを取得したデータ配送モジュールは，データ配送モジュール内のテーブルに存在するセンサデバイスと，そのデータを希望する VM のアドレスをマッピングしてあるテーブルを参照し，配送を希望する VM にセンサデータを転送する．ただしデータの配送を希望する VM が複数存在する場合は，そのセンサデータを複製し希望する VM に配送を行う．また，センサデバイスのプロパティは仮想センサモジュールより取得要求が配送モジュールへ通知され，それを受信した配送モジュールはセンサークラウドマネージャに通知する．センサークラウドマネージャは対応するセンサデバイ

スのプロパティを DB から取得し，データ配送モジュールを介して利用者，開発者へ返答する．もしセンサデバイスに直接プロパティを取得する必要がある場合は，クラウドマネージャより配送モジュールにプロパティ取得要求が通知され，センサドライバモジュールを介して，実センサデバイスよりプロパティが利用者，開発者へ返答される．

利用者，開発者が使用する各 VM にはセンサデータを取得するための仮想センサモジュールが組み込まれる．仮想センサモジュールはデータ配送モジュールが送信するセンサデータを受信して一時的にキャッシュする．利用者，開発者がセンサデータ取得の要求をこの仮想センサデバイスに行うことで，仮想センサモジュールは指定されたセンサデータをキャッシュから利用者，開発者へ渡す．ただしセンサデータが指定されたデータより古くかつセンサデバイスが Passive なデバイスである場合，センサデータを即時に取得するよう，仮想センサモジュールからデータ配送モジュールにデータ取得要求が送信される．

4.2 自動化を実現する機構

センサークラウドマネージャは，利用者，開発者が作成する VM にセンサデータを配送するようセンサデバイス用ハイパーバイザへの設定や命令を行い，作成される VM に仮想センサデバイスモジュールを自動的に組み込む機能となる．既存のクラウドマネージャは，Web GUI より利用者，開発者が VM への操作要求を行うと，IT 資源を設定し VM をプロビジョニングする機能や削除する機能を有する．本センサークラウドマネージャは既存のクラウドマネージャへセンサデバイス用の設定機能を拡張した機構となる．

次に拡張する処理を既存のワークフローとともに**図 4** の番号と対応して説明する．

1. 利用者，開発者の指定により VM 作成が行われると，利用者，開発者が指定したパラメータがセンサークラウドマネージャに渡される．
2. センサークラウドマネージャは通常のクラウドマネージャ同様に渡されたパラメータに従い IT 資源の情報を取得する．
3. 利用者，開発者から要求されたパラメータをハイパーバイザの管理を行うノードへ通知する．
4. 通常の IT 資源用ハイパーバイザが VM の作成を行う．
5. プロビジョニングされた VM の IP アドレスや root のパスワードが設定される．
6. 利用者，開発者が希望したセンサデバイス群と作成された VM の IP アドレスをセンサークラウドマネージャ内のテーブルに追加し，データ配送モジュールにセンサデータの配送開始を通知する．配送通知を受信したデータ配送モジュールは，更新されたテーブルよりセンサデバイスと配送を行う VM のアドレスを取得し，ただちにセンサデータの配送を行う．また，作成された VM にはテンプレート自体に仮想センサデバイスのモジュールが含まれており，仮想センサデバイスモジュールの設定および起動もセンサークラウドマネージャが行う．
7. 利用者，開発者に VM の作成完了を通知する．

通常のワークフローでは，5 のワークフロー後に 7 のワークフローで処理は終了するが，本機構では 5 の処理後に，センサデバイスを設定するワークフローを呼び出すよう 6 のワークフローを追加した．

次に，利用者，開発者の要求により VM の削除が行われる際，VM 作成と同様に既存の VM 削除のワークフローを拡張し，センサデバイス削除のワークフローが呼び出されるよう既存のクラウドマネージャを拡張した．センサデバイスの削除では，該当する VM の IP アドレスとそれらが要求していたセンサデバイスのマッピングをテーブルから削除し，センサデータ配送の停止をデータ配送モジュールに通知する．また，もしそのセンサデータ配送を希望する VM がすべていなくなった場合，センサドライバを停止し，該当するセンサデータの受信を停止する．

新しいセンサデバイスを IT 資源として登録する処理もセンサークラウドマネージャが行う．センサドライバモジュールはあらかじめクラウド管理者がインストールし，もし新たなセンサデバイスの登録要求がクラウド管理者からある場合，センサマネージャは配送モジュールにデバイスの登録要求通知およびサポートするために必要な情報（e.g. IP アドレス，ポート番号）を通知する．その通知を受信した配送モジュールは該当するセンサドライバモジュールを動的にロードし，渡されたパラメータに従い接続可能かどうかをテストする．もしセンサデバイスへ接続が可能であれば，センサデバイス群の情報を DB に追加する．

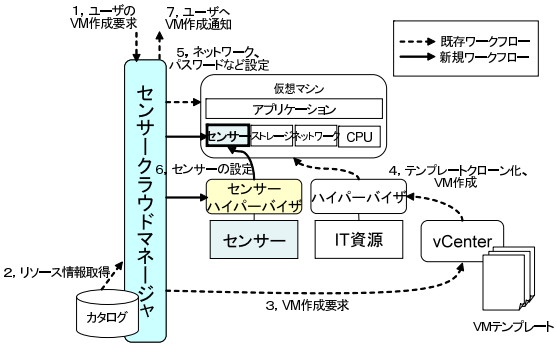

図 4 既存ワークフローへの拡張
Fig. 4 Extension of existing workflow.

4.3 標準化を実現する機構

クラウドマネージャは，IT 資源として追加されたセンサデバイスの情報を保持するため，カタログと呼ばれる DB を保持する．カタログとして格納された IT 資源情報

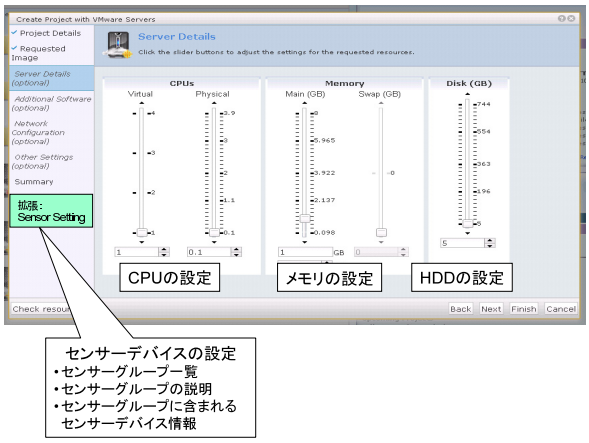

図 5 VM 要求時のカタログ
Fig. 5 Catalog for VM request.

は VM プロビジョニング時に参照および更新される．既存のカタログは CPU のコア数，メモリ量，HDD 量が格納されており，利用者，開発者はその IT 資源の中から自分が使用する IT 資源を選択しプロビジョニングする．利用者，開発者はそのカタログに登録されているテンプレートと呼ばれる頻繁に使用される OS やソフトウェアをセットにした雛形へ，指定した IT リソースを割り当てることで，ソフトウェアや OS がインストール済みの VM をすぐに使用できる．このようなクラウドの機能を標準化と呼び，センサデバイスにも適用する．センサークラウドでは，既存の IT 資源を格納するカタログへ個々のセンサデバイス情報を格納し，よく使用するもしくは意味のある単位でグループ化を行いテンプレートとしてカタログへ登録する．このセンサデバイスのテンプレートと既存のソフトウェア群のテンプレートを紐付けることで，利用者，開発者は標準のテンプレートを選択しプロビジョニングを行うと，センサデバイスも VM に割り当てられる．

まず，センサデバイスの保持者がクラウド管理者に対し，センサデバイスの情報を XML ファイルにて通知する．クラウド管理者はそのデータをカタログへ追加しセンサデバイスへの接続が確認できた場合，利用者，開発者に追加されたカタログ中のセンサデバイス群が利用可能となる．また，クラウド管理者は使用可能となったセンサデバイスからテンプレートを作成し，既存のテンプレートと紐付けを行う．

利用者，開発者は VM 作成時に図 5 に示すようなプロビジョニングを行う VM への CPU やメモリの設定と同様に，センサデバイス群もカタログの情報として登録されるため，プロビジョニング時に利用者，開発者による使用センサデバイスの選択が必要となる．そのため，CPU などの既存 IT 資源と同様に，カタログにセンサデバイスを表示，選択できるメニューを拡張する．しかしセンサデバイスをそれぞれ単独で登録した場合，利用者，開発者は膨大な数から 1 つずつ希望のセンサデバイスを選択することが

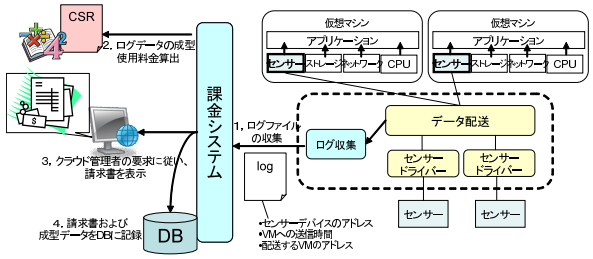

図 6 課金システムのシステムアーキテクチャ
Fig. 6 System architecture for accounting system.

必要になってしまう．そこでセンサデバイスを個別に扱うのではなく，意味のある単位でグループ化し，それをカタログに表示させる．使用するセンサグループ群を GUI で選択し，選択したグループとともに VM のプロビジョニングが可能となる．

4.4 センサ使用への課金システムの設計

本節ではセンサークラウド上でのセンサ使用への課金システムの設計について述べる．図 6 の番号と対応し課金システムの動作手順を説明する．

1. センサデータの配送情報をデータ配送モジュールがファイルへ出力し，出力されたログを定期的に取集，課金システムに送信する．
2. 課金システムは送信されたログを解析し，独自のデータフォーマットに成型し，利用者，開発者が設定した計算式（e.g. 1 センサデータあたりの値段）に基づき各アカウントへの使用料金を算出する．
3. クラウド管理者や利用者，開発者の要求に従い算出された使用料金を GUI で表示する．
4. 成型されたセンサデータ使用量の情報および算出された使用料金は，課金システムが保持する DB に記録される．

センサ用の課金計算に必要な拡張処理は 1 のデータ配送モジュールのログ出力と 2 のセンサ使用料の計算ポリシの設定となる．

4.5 センサへのモニタリングシステムの設計

本節ではセンサークラウド上でのセンサ監視機能について説明する．センサ監視機能はセンサークラウド上に登録されたセンサデバイスが，正常に稼働しているかをチェックする．センサデバイスの異常を検知する情報として，センサデータが定期的に送信されているか，生存メッセージに応答があるか，もしくはセンサデバイスからバッテリの残量が取得できる場合，バッテリの残量が十分か，データ配送モジュールから利用者，開発者が使用する VM までの配送遅延が監視対象としてあげられる．これらの情報はセンサドライバやデータ配送モジュールから取得可能である．

図 7 にモニタリングシステムのアーキテクチャを示す．

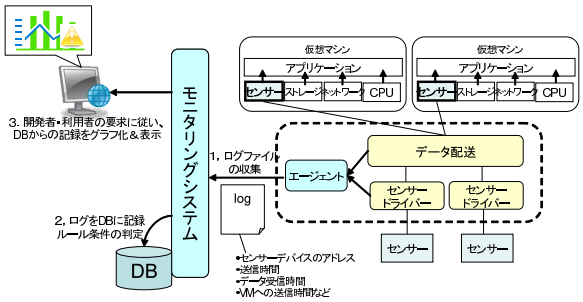

図 7 モニタリングシステムのアーキテクチャ

Fig. 7 System architecture for monitoring system.

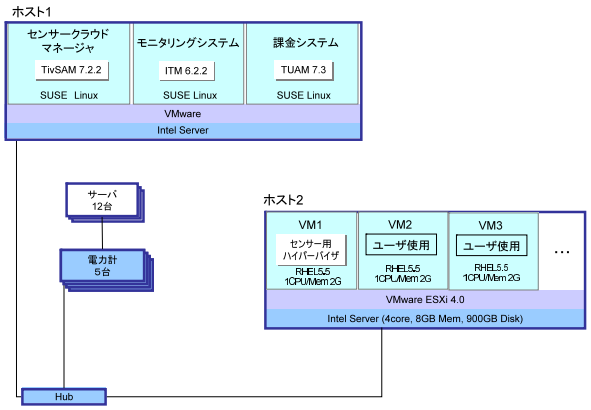

図 8 実装環境

Fig. 8 Implementation environment.

このモニタリングシステムは既存のクラウドシステムで使用されているモニタリングシステムであり，通常の IT 資源である CPU やストレージ，メモリが正常に動作しているかを監視する機構へ，センサデバイス監視機能を拡張させたシステムとなる．この既存機構へのセンサデバイスの異常検知を判断する情報を，センサデバイス用ハイパーバイザが動作しているホストのエージェントが送信することで，センサ監視機能を実現する．次に図 7 の番号と対応し動作手順を説明する．

1. データ配送モジュールやセンサドライバからの異常検知のための情報は，エージェントを通じセンサークラウド用に拡張された監視システムへリアルタイムに送信される．
2. エージェントよりモニタリングシステムへ送信されたログは DB に記録され，利用者，開発者が設定したルールにあてはまるか検査される．
3. 送信された監視情報は，クラウド管理者が Portal Server から GUI で確認できる．

このモニタリングシステムはデータ配送の状況を GUI で確認するだけではなく，利用者，開発者が定義した異常検知の条件にあてはまる事象，たとえば，センサデータがある一定時間以上センサドライバから受信されないルールを異常と設定し，ログがこの条件にあてはまった場合，センサ監視システムはクラウド管理者へ異常を通知することで，センサデバイスの早期異常検知が可能となる．これらのフレームワークは既存のモニタリングシステムで完成されており，センサ用の設定を監視システムへ追加することで，センサデバイス自体の監視が可能となる．

5. センサークラウドのプロタイプ実装

本章では，前章で記述したセンサ管理技術，課金，モニタリングを含むセンサークラウドのプロトタイプ実装について説明する．

まず，本プロトタイプ実装の環境を説明する．本実装で使用したセンサデバイスは消費電力が SNMP により取得可能な電力計である．この電力計は出力コンセントを 5 つ搭載し，そのコンセントに接続された機器のそれぞれの消費電力が SNMP 要求により取得できる．本プロトタイプ実装を行った実装環境を図 8 に示す．ハードウェアはホスト 1 およびホスト 2 が存在し，ホスト 1 ではセンサークラウドマネージャ，モニタリングシステム，課金システムが VM 上で動作し，センサークラウドマネージャよりホスト 2 へ VM がプロビジョニングされる．ホスト 2 ではセンサデバイス用のハイパーバイザの一部であるデータ配送モジュールおよびセンサドライバモジュールが動作する VM1 が存在し，仮想センサデバイスを含み通常の利用者，開発者が使用する VM2 から VM6 の VM がプロビジョニングされる．センサとなる電力計は計 5 台を使用し，電力計には計 12 台の物理サーバが接続している．

4 章で述べたセンサ管理技術，モニタリング技術，課金技術において，センサデバイス用ハイパーバイザ以外は既存のクラウドコンピューティングの管理，モニタリグ，課金システムへセンサデバイスに対応するよう拡張したシステムである．そこで，既存のクラウド管理技術のソフトウェアに対して拡張を行うことで，各システムのプロトタイプ実装を行った．各システムのプロトタイプ実装を説明する．

センサ管理技術

センサ管理技術では既存ソフトウェアである Tivoli Service Automation Manager（TSAM）[13] を用いセンサ管理技術を拡張した．通常のクラウド管理のワークフローにセンサ用管理のワークフローを追加し，既存の IT 資源やプロビジョニングした VM の情報を格納する DB へセンサデバイス，センサネットワークの情報を格納するエントリを追加した．また，センサデバイス用ハイパーバイザのモジュールを含むシステムは，このクラウドマネージャがプロビジョニングした VM 上で動作する．センサドライバモジュールとしては SNMP のプロトコルをサポートしたモジュールを作成し，データ配送モジュールはセンサデバイスが受信したセンシングデータを VM に配送する．セン

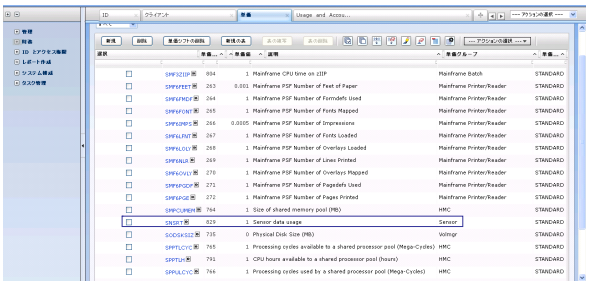

図 9 課金技術の実装
Fig. 9 Implementation for metering system.

サクラウドマネージャはプロビジョニングした通常の VM へ，利用者，開発者が希望したセンサデータを配送するよう，センサデバイス用ハイパーバイザの設定が記載された XML ファイルを更新し，ハイパーバイザに再設定を命令する．図 5 に TSAM のスクリーンショットを示す．本プロトタイプ実装では UI の実装はしていないため表示ができないが，CPU やメモリ，アプリケーションの設定と同様に利用者，開発者が UI よりセンサデバイスの表示を行い，追加や削除が可能となる．

課金技術

センサ課金技術では既存のシステムである，Tivoli Usage and Manager（TUAM）[14] へ，データ配送モジュールより出力されるログのデータ情報を設定ファイルに定義し，各データの要素への課金算出方法をクラウド管理者として設定した．**図 9** に要素への課金方法を行う際のスクリーンダンプを示す．ここでは Sensor data usage に対し，単価 1 の設定を行っている．

モニタリング技術

センサデバイス監視技術では，クラウドコンピューティングにおける通常の IT 資源を監視するソフトウェアである IBM Tivoli Monitoring（ITM）[15] を拡張した．ITM は通常の IT 資源のほかにも，監視情報のデータフォーマットを定義することで様々なデータの監視が可能となる．本プロトタイプの実装ではセンサデバイスとして使用した電力計がサポートする，SNMP の要求および応答のデータを監視データの対象とするよう，クラウド管理者として出力されるログファイルの定義を ITM 上で設定した．また，配送遅延や生存メッセージの返答がない場合を異常と見なすルールを設定した．**図 10** に ITM におけるルール設定のスクリーンダンプを示す．このルールでは SNMP の最も新しい送信時間より，最も新しい受信時間が遅い事象が 2 回連続した場合，アラートを上げる設定を行っている．このように異常のルール設定は式およびデータの要素から GUI で設定できる．

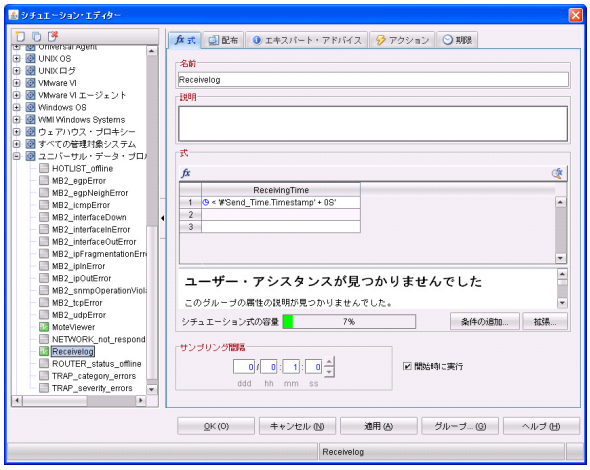

図 10 モニタリング技術の実装
Fig. 10 Implementation for monitoring system.

6. センサークラウドのプロトタイプ実装による評価

本章ではセンサークラウドの評価について述べる．評価環境としてプロトタイプ実装を行った図 8 を用いセンサークラウドを評価した．

まず，設置した電力計 5 台に対し接続されたサーバ 12 台の消費電力をセンサークラウドを通じ，ホスト 2 にプロビジョニングされた VM2 より取得した．この VM2 では開発者や利用者が仮想センサデバイスの API を呼び出すだけで，希望したセンサデバイスのセンサデータが取得できる．取得した消費電力値を**表 1** にあげる．開発者，利用者は VM2 の仮想センサデバイスから任意のセンサデータ取得が可能となった．

6.1 センサ管理技術の評価

本節ではセンサークラウドのセンサデバイス用ハイパーバイザの評価について説明する．ハイパーバイザの仮想化機能である，複数の利用者，開発者が同時に同一センサの共有が可能であるかを評価した．ただし，実験環境に用いたセンサデバイスである電力計は SNMP による要求が必要となる．そこで本評価では**図 11** および具体的な処理の流れを**図 12** で示すような SNMP 要求を行った．

本機構であるセンサークラウド環境では利用者，開発者が使用する VM のセンサデータ受信方式は非同期通信と同期通信が存在する．図 11，図 12 で示されている番号とともに説明する．

1. 非同期通信では，クラウド管理者の指定した間隔でセンサデバイス用ハイパーバイザが SNMP 要求を送信し，受信した SNMP 応答から電力データのみを指定された VM へ配送する．この際，SNMP が要求するセンサデータは，電力計が搭載している 5 つの出力コンセントすべての消費電力を 1 回の SNMP パケット

表 1 取得したサーバの消費電力
Table 1 Power consumption of servers.

サーバ	消費電力	サーバ	消費電力	サーバ	消費電力
1	541 W	2	446 W	3	540 W
4	169 W	5	222 W	6	138 W
7	138 W	8	139 W	9	155 W
10	234 W	11	242 W	12	266 W

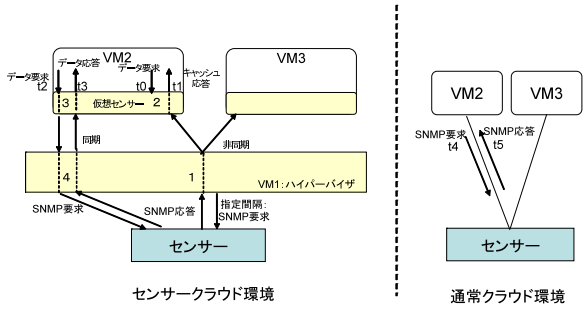

図 11 評価測定
Fig. 11 Evaluation measurement.

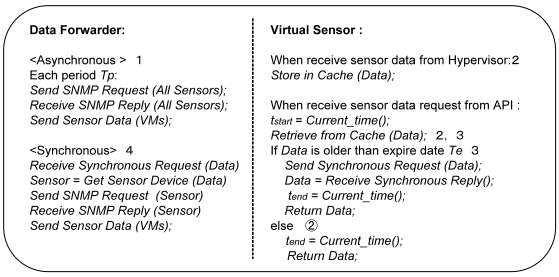

図 12 処理のフロー
Fig. 12 Process flow.

で要求する．

2. それを受信した各 VM の仮想センサデバイスは受信したセンサデータをキャッシュし，もし利用者，開発者よりセンサデータの要求があった場合，キャッシュ内のセンサデータを返答する．このような仮想センサデバイスモジュールのキャッシュから，利用者，開発者へセンサデータの応答をする通信を非同期通信とする．
3. キャッシュ内のセンサデータがクラウド管理者に要求されたデータより古い場合，仮想センサデバイスはセンサデバイス用ハイパーバイザへ，明示的にセンサデータの要求を送信する．
4. 要求を受信したハイパーバイザは，指定間隔以外に SNMP 要求を即座に発行し，センサデータを取得する．その際，取得するセンサデータは該当する 1 センサデータのみであり，出力コンセント 1 口に対する消費電力を 1 回の SNMP 要求で取得する．取得されたセンサデータはハイパーバイザによりそのセンサデータを要求していたすべての VM に対し配送を行う．再び 3 において仮想センサモジュールが受信したセンサデータを利用者，開発者へ返す．このセンサデータの

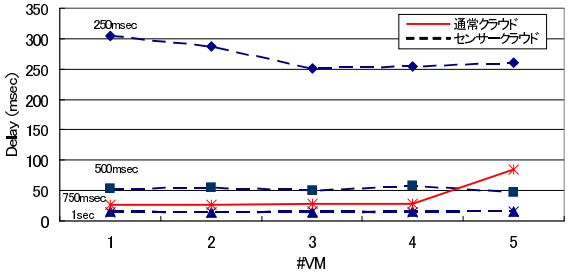

図 13 実験 1：センサデータ取得遅延（500 msec）
Fig. 13 Ex1: Delay of delivering sensor data (500 msec).

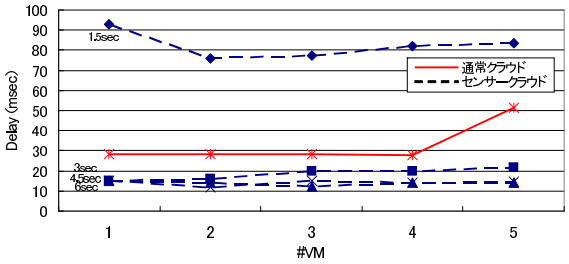

図 14 実験 2：センサデータ取得遅延（3 sec）
Fig. 14 Ex2: Delay of delivering sensor Data (3 sec).

取得方式を同期通信とする．

一方，通常のセンサデータ取得方式であり，センサークラウドの環境を利用しない環境を通常クラウド環境とする．この環境では開発者，利用者は通常の VM からセンサデバイスへそれぞれ SNMP 要求を送信する．

本節では通常のクラウド環境とセンサークラウド環境において，センサデバイスである電力計を 1 台にし，センサデータを取得する VM 数を 1 台から 5 台まで変化させて評価を行った．利用者，開発者が仮想センサデバイスからデータを取得する間隔およびキャッシュ時間を変化させ，センサデータ取得遅延を計測した．ただしセンサデバイス用ハイパーバイザにおいて，定期的に送信する SNMP 要求の時間である Tp は利用者，開発者のセンサデータ要求間隔と等しい．ここで測定した計測時間はセンサークラウド環境においてにおいて，図 11 に示す利用者，開発者がセンサデータを要求しキャッシュから返答する非同期通信の時間である t1 - t0 および同期通信がセンサデータを要求し，SNMP 要求が送信され，SNMP 応答のセンサデータが利用者，開発者へ返される t3 - t2 となり，処理フローでは $t_{end} - t_{start}$ の時間となる．また通常クラウド環境では各 VM が SNMP 要求を送信し，SNMP 応答を受信する時間を計測した．

VM 上の利用者，開発者からセンサデータの要求を 80 回行い，その平均値の遅延測定結果を**図 13**，**図 14** に示す．図 13 の実験 1 では 500 msec 間隔，図 14 の実験 2 では 3 sec 間隔にセンサデータ取得間隔を設定し，仮想センサデバイスから Read を行った．グラフの横軸はセンサデータを取得する VM 数，縦軸はセンサデータ取得までの遅延

$t_{end} - t_{start}$ を示している．また，グラフ内の破線はセンサークラウド環境での評価結果，直線は通常クラウド環境の結果を示している．センサークラウド環境の破線において図 13 では 250 msec，500 msec，750 msec，1 sec と記載があり，図 14 では 1.5 sec，3 sec，4.5 sec，6 sec と記載があるが，これはキャッシュの有効期限である Te を示している．実験 1 において利用者，開発者がセンサデータを要求する間隔である 500 msec は，センサデバイス用ハイパーバイザがセンサデータを取得する間隔である Tp と等しく動作する．もし Te が 250 msec である場合，Tp が 500 msec であると Te が 500 msec 以上の遅延と比較し，キャッシュミスが多く発生し同期通信の回数が増えるため遅延が大きい．一方，Te が 1 sec である場合，利用者，開発者へキャッシュ内のセンサデータが返されるため，同期通信はほぼ発生せず遅延が低いことが分かる．

図 13，図 14 ともに Tp が 250 msec，1.5 sec と同期通信が多く発生する条件下では，それぞれ遅延が 305 msec，93 msec と非常に大きくなる．またセンサークラウド上において，VM の数が 1 における遅延と比較し，VM 数が 2 以上における配送遅延は小さくなるが，これは同期通信によって取得したセンサデータは他の VM にも同時に配信されるため，1 VM あたりの同期通信の発生する回数が減少するためである．たとえば図 13 において Te を 250 msec と設定した場合，VM1 台あたりの同期通信回数は VM 数が 1：922 回，VM 数が 2：682 回，VM 数が 3：597 回，VM 数が 4：584 回，VM 数が 5：523 回となり，VM 数が増えるとともに 1 つの VM で発生する同期通信回数が減少するため，VM が 1 つである結果と比較し VM が 5 つである場合，遅延は 86%と減少する．

また通常クラウド環境において VM 数が 5 である場合，図 13，図 14 とも遅延が 84 msec，52 msec と大きい．これは 1 台のセンサデバイスに対し 5 台の VM が同時に SNMP 要求を送信するため，SNMP 要求が受信バッファに格納されず破棄されタイムアウトが何度も発生したためである．これにより，VM が 5 台以上でかつ Te は同期通信が発生しない Tp の時間より大きい場合，通常クラウド環境と比較しセンサークラウド環境の遅延が最大で図 13，図 14 それぞれ 69 msec，37 msec と減少した．

6.2 センサ監視技術の評価

本節ではセンサ監視技術が異常の早期発見が可能であるかを評価した．センサデータ取得間隔である Tp を 1 秒とし，SNMP の送受信時間をモニタリングシステムへ送信するよう設定を行った．異常検知のルールとして最新の SNMP 応答の受信時間と SNMP 要求時間の差が 3 秒以上になった場合，異常と見なしてアラートを出すようクラウド管理者として設定を行った．

まず電力計をネットワークに接続し，利用者，開発者が使用する VM から非同期通信を開始した．非同期通信の開始後，電力計をネットワークから切断し，センサデバイス用ハイパーバイザが動作する VM1 へのネットワーク接続を人為的に切断した．電力計がネットワーク到達できなくなったことにより，VM1 は SNMP 要求を送信するにもかかわらず SNMP 応答を受信しない状況となる．この状況下における監視結果を**図 15** に示す．このログはハイパーバイザでの SNMP 応答時間を示している．電力計がネットワークへ接続されている場合は 1 秒間隔ごとに SNMP 応答を受信できているが，接続性がなくなり SNMP 応答がなくなったため異常条件のルールに合致しアラートが出力された．アラートはこのセンサデバイスを使用している利用者，開発者やクラウド管理者，センサネットワーク所有者に通知される．本機構によりセンサデバイス自体の早期異常検知が可能となった．

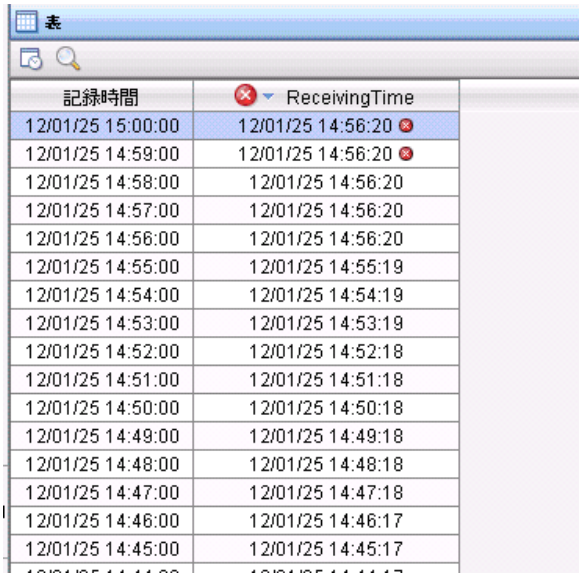

図 15 モニタリングの画面ダンプ
Fig. 15 Screen capture of monitoring system.

6.3 センサ課金技術の評価

本節ではセンサ課金技術の評価について述べる．図 8 において使用料金を算出するために，データ配送モジュールよりセンサデータの使用情報を課金システムへ送信するよう，クラウド管理者として設定した．また例として 1 センサデータにつき 1 円の課金を行うよう課金システムに設定した．課金のポリシはセンサネットワーク所有者から提示される．**図 16** に結果を示す．図 16 は課金システムが表示する請求書であり，使用者，開発者が使用する．請求書ではセンサデータを 20 回受信しているため 1 円 × 20 パケット=20 円の課金が表示されており，利用者，開発者へのセンサ使用料の算出が可能であることを示している．

図 16 課金システムの画面ダンプ
Fig. 16 Screen capture of accounting system.

6.4 評価の考察

6.1 節のセンサ管理技術の評価では仮想化機能に対する性能評価を行った．図 13，図 14 で示したように，センサークラウド環境において頻繁に同期通信が発生する状況は，遅延がそれぞれ 260 msec，83 msec と増大する．遅延の増加を防止するためには同期通信の回数を増加させないようにする必要がある．本実装では同期通信が発生した場合，SNMP 要求を各出力コンセントに対しそれぞれ送信しているため，8 コンセント分のセンサデータを要求するためには 8 回の SNMP 要求，応答が発生する．そこで 1 回の SNMP 要求で複数の出力コンセントのセンサデータを取得するよう，ある一定時間内の SNMP 要求を 1 つの SNMP 要求に集約させ，SNMP 要求の送信回数を削減するといった機能が必要になる．

また，データ配送モジュールが定期的にセンサデータを取得する間隔 Tp とキャッシュの保持期間となる Te の値の関係は同期通信の発生回数に大きく影響を及ぼす．現状では Te の時間はクラウド管理者が手動で設定しているが，利用者，開発者のセンサデータ取得間隔に合わせ，タイムアウトが発生しない Te の値を最適に決定できる仕組みが必要である．

本評価では，センサデバイスを 1 台のみ使用して実験したが，登録されたセンサデバイスの台数が増加した場合，センサークラウドの問題点として，データ配送モジュールのデータ配送遅延があげられる．本評価ではセンサデバイスの台数が 1 台のため配送遅延が少ないが，センサデバイスが増加した際，データ配送モジュール内のデータコピーや各 VM へのセッション管理などの負荷が増加し，センサデータの配送遅延が大きくなる．今回のプロトタイプ実装ではデータ配送モジュールは 1 つのみであるが，複数のデータ配送モジュールを設置し，配送処理の負荷を分散させることで，センサデバイスへのスケーラビリティを確保する．また，データ配送モジュール内のメッセージ配送手法自体もなるべく軽量化し，データ配送にかかる負荷を軽減する予定である．

また，本機構はデータ配送モジュールが利用者，開発者とセンサデバイスの中に存在し，利用者，開発者の取得タイミングによってはキャッシュから返答をしてしまうため，必ずしもリアルタイムのセンサデータが取得できない可能性がある．もし，4 章で述べたような，リアルタイム性が必要なセンサアプリケーションでは，高頻度でキャッシュを破棄する必要があるため，センサデバイスへの負荷が高く非効率となりセンサークラウド上の使用には適さない．しかし，センサデバイスの取得遅延が許容できるセンサーアプリケーションは数多く存在する．たとえば，農地のモニタリング，温度や雨量のモニタリング，室内の空調や照度の制御といったアプリケーションには十分有効に適用可能である．このことより，センサークラウドでは取得遅延が許されるセンサアプリケーションにおいて，通常のクラウド環境と比較し，多くの利用者，開発者におけるセンサデバイスの共有が可能となる．

さらに，センサ管理技術の機能となる仮想化，自動化，標準化の達成について，自動化，標準化は機能要件にあたるため，本実装，評価によりセンサクラウドを実現した時点で目的を達成したと考えられる．また，仮想化ではセンサデバイスを仮想化するといった機能要件のほかに，直接センサデバイスへ開発者，利用者が接続する既存の方式と比較し，より多くの利用者，開発者が 1 つのセンサデバイスを共有できることが非機能要件となる．評価結果から，配送遅延が許容できるようなキャッシュなどのパラメータにおいて，直接センサデバイスに接続した既存の手法とセンサークラウドの結果を比較し，センサークラウドでは，既存手法よりもデータ取得遅延が小さく非機能要件を満たした．今回使用したセンサデバイスは電力計であり，センサデータの変動は緩やかなとなる．そのため，評価で行った取得周期である 500 ミリ秒および 3 秒のサンプリングデータを線形補完するだけで，十分に精度が高いデータとなる．また，センサデータの遅延では，取得周期が 500 ミリ秒の場合，かつキャッシュ時間が 750 ミリ以上，または取得周期が 3 秒の場合，かつキャッシュ時間が 3 秒以上ならば，通常クラウドの評価と比較しセンサークラウドの遅延が小さい．初めに想定した多少のデータ取得遅延が許可できるアプリケーション，たとえば電力モニタのアプリケーションであれば，このデータ取得遅延は十分に許容できると考え，本評価のパラメータは妥当であると考えられる．

7. アプリケーション

本章ではセンサークラウド環境が有効に使用できるアプリケーションシナリオについて説明する．**図 17** はセンサークラウド上で想定されるアプリケーションを示している．センサークラウド上では図 17 で示すように各 VM の利用者，開発者は物理的に離れた空間のセンサデバイスや種類の異なるセンサデバイスを同じ VM 上の仮想センサデバイスとして使用できる．また，センサデバイスを仮想化するため，同一のセンサデバイスを複数の利用者，開発

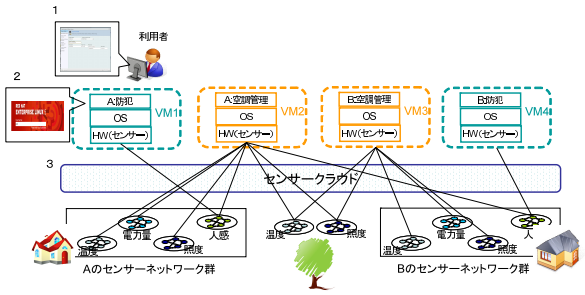

図 17 センサークラウドのアプリケーション
Fig. 17 Application for sensor cloud.

者が同時に共有することも可能となる．さらに，中間ノードを含むデバイス状態の変化に対応した柔軟なアプリケーションの作成，簡単なセンサデバイスおよびセンサアプリケーションの利用，同一センサアプリケーションであるが，異なるセンサデバイスを用いた VM のプロビジョニングが簡単にできる．

図 17 ではアプリケーション例として，室内 A のセンサデバイス群，室外のセンサデバイス群，室内 B のセンサデバイス群が存在する．まず，外気温や照度，および室内の温度，電力量，照度，人感を使用する空調制御のセンサアプリケーションがテンプレートとしてある場合，同様な種類のセンサデバイスが存在すれば，OS を含むソフトウェア群は同一であるが，VM へ割り当てるセンサデバイスを変更するだけで，室内 A 用および B 用の空調管理センサアプリケーションの VM が簡単に作成できる．これらのアプリケーションはデバイスの状態も把握できるため，バッテリの残量などを監視しながらアプリケーションの動作を変更できる．たとえば，バッテリ残量が低下し照度のセンシング頻度が減少した場合，取得した照度のセンサデータをミドルウェアが補完することで，急に低下したセンシング頻度をアプリケーションから隠ぺいできる．また，バッテリの寿命を延命するため，アプリケーションからのセンサデータの問合せを減少させるなど，デバイス状態に適応した処理を行うアプリケーションの開発が可能となる．

もし，利用者が A のセンサデバイス群を利用したい場合，図 17 に示す数字と対応してプロビジョニング手順を説明する．

1. CPU やストレージといった VM パラメータに加え，自分が使用したいセンサデバイスを GUI から要求する．
2. 仮想センサデバイスを含む VM が作成される．
3. プロビジョニングされた VM 上で，利用者は仮想センサデバイスを用いセンサアプリケーションを動作させる．

また，室内の人感センサを用い，留守中に防犯を行うアプリケーションがある場合，プロビジョニング時に異なるセンサデバイスを指定するだけで，A，B それぞれを対象としたアプリケーションが構築できる．利用者が長期家を留守にするときのみ，防犯アプリケーションを使用したい場合，利用者が希望する期間のみセンサデバイスおよびセンサアプリケーションを含む VM を動的にプロビジョニングし，不必要になった際にデプロビジョニングできるため，必要なときのみ VM が使用できる．これにより，アプリケーションやセンサデバイスへの使用料金が利用者へかかる場合，利用者は必要なときのみ VM を使用することで，使用料金を抑えられる．

8. まとめと今後の課題

本論文ではセンサデバイスを IT 資源の 1 つとしてクラウド環境に適用させ，CPU やストレージとセンサデバイスを共有できるシステムであるセンサークラウドについて述べた．センサークラウド環境では，センサデバイスを仮想化し共有できるだけではなく，既存のクラウド環境の機構をセンサデバイスに適用させ，センサの使用料金やセンサデバイスの監視も可能となった．実環境でプロトタイプ実装を行い，センサークラウドの仮想化機能の性能評価を行った．本実験での環境では VM 数が 5 以上かつデータのキャッシュ保持期間が長い場合，直接センサデバイスへデータを取得する通常の使用手法と比較し，センサークラウドよりセンサデータを取得した方が，取得遅延が最大で 81% 減少した．これよりセンサークラウド環境を用いることで，多くの利用者，開発者へのセンサデバイス共有が可能となる．

将来の機能として，1. データ配送モジュールを複数にし，センサデバイス数のスケーラビリティを向上させる．2. 現在サポートするセンサは電力計のみであるため，サポートするセンサデバイスの種類を増やす．3. 各 VM から送信される同期通信の要求を集約させ，センサデバイスへのセンサデータ取得頻度を減少させる．4. 利用者，開発者のセンサデータ要求頻度によりキャッシュの適切な時間を自動設定する機能の追加があげられる．

参考文献

[1] Gershenfeld, N., Krikorian, R. and Cohen, D.: The Internet of Things, *Scientific American*, Vol.291, No.4, pp.76–81 (2004).
[2] Borden, T.L., Hennessy, J.P. and Rymarczyk, J.W.: Multiple operating systems on one processor complex, *IBM Syst. J.*, Vol.28, No.1, pp.104–123 (1989).
[3] Sugerman, J., Venkitachalam, G. and Lim, B.-H.: Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor, *Proc. General Track: 2002 USENIX Annual Technical Conference*, Berkeley, CA, USA, USENIX Association, pp.1–14 (2001).
[4] Gaynor, M., Moulton, S.L., Welsh, M., LaCombe, E., Rowan, A. and Wynne, J.: Integrating Wireless Sensor Networks with the Grid, *Proc. IEEE Internet Computing*, Vol.8, pp.32–39 (2004).
[5] Dunkels, A., Alonso, J., Voigt, T., Ritter, H. and Schiller, J.: Connecting Wireless Sensornets with

TCP/IP Networks, *Proc. 2nd International Conference on Wired/Wireless Internet Communications* (*WWIC2004*), pp.143–152 (2004).
[6] Grosky, W., Kansal, A., Nath, S., Liu, J. and Zhao, F.: SenseWeb: An Infrastructure for Shared Sensing, *Proc. IEEE Multimedia '07*, Vol.14, No.4, pp.8–13 (2007).
[7] Rowe, A., Berges, M., Bhatia, G., Goldman, E., Rajkumar, R., Soibelman, L., Garrett, J. and Moura, J.M.F.: *Sensor Andrew: Large-Scale Campus-Wide Sensing and Actuation*, CMU Technical Report (2008).
[8] Achtzehn, A., Meshkova, E., Ansari, J. and Mahonen, P.: MoteMaster: A Scalable Sensor Network Testbed for Rapid Protocol Performance Evaluation, *Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, pp.1–3 (2009).
[9] Werner-Allen, G., Swieskowski, P. and Welsh, M.: MoteLab: A wireless sensor network testbed, *Proc. 4th International Symposium on Information Processing in Sensor Networks*, *IPSN '05* (2005).
[10] Hassan, M.M., Song, B. and Huh, E.-N.: A framework of sensor-cloud integration opportunities and challenges, *Proc. ACM ICUIMC '09*, pp.618–626 (2009).
[11] Kurschl, W. and Beer, W.: Combining cloud computing and wireless sensor networks, *Proc. ACM iiWAS '09*, pp.512–518 (2009).
[12] Hill, J. and Culler, D.: A Wireless Embedded Sensor Architecture for System-level Optimization, Technical Report, U.C. Berkeley (2001).
[13] IBM Corporation: TSAM: Tivoli Service Automation Manager, IBM Corporation (online), available from ⟨http://www-01.ibm.com/software/tivoli/products/service-auto-mgrx⟩ (accessed 2012-11-06).
[14] IBM Corporation: TUAM: Tivoli Usage and Accounting Manager, IBM Corporation (online), available from ⟨http://www-01.ibm.com/software/tivoli/products/usage-accounting/platforms.html⟩ (accessed 2012-11-06).
[15] IBM Corporation: ITM: Tivoli Monitoring, IBM Corporation (online), available from ⟨http://www-01.ibm.com/software/tivoli/products/monitor/⟩ (accessed 2012-11-06).

高橋 ひとみ （正会員）

2003 年慶應義塾大学環境情報学部卒業．2005 年同大学大学院修士課程修了．2008 年同博士課程修了．博士（政策・メディア）．同年日本アイ・ビー・エム入社，現在に至る．

串田 高幸 （正会員）

1985 年日本アイ・ビー・エム株式会社入社．同年サイエンス・インスティチュート（現，東京基礎研究所）配属．入社以来，超高速ネットワークプロトコル，信頼性マルチキャストプロトコル，モバイルワイヤレス QoS，グリッドコンピューティング，システム管理，クラウドコンピューティングの研究に従事．現在，日本アイ・ビー・エム株式会社東京基礎研究所にシニアリサーチャーとして勤務．情報処理学会マルチメディア通信と分散処理研究会アドバイザー，2011 年より情報処理学会理事および情報処理学会論文誌編集委員長．IEEE，ACM の各会員．