クラウドとセンサの統合システム管理方式

串田 高幸[1]

概要：クラウドは，IT で自動化や省力化において中心のサービスとして位置づけられている．また，IoT では，センサを使った多くのアプリケーションやソリューションの実用的な利用がはじまっている．クラウドにおいて，センサをプロダクション・システムの一部として利用するとき，センサ・データの信頼性を確保する管理やアプリケーションやソリューションからアクセスを容易に可能にするためのインターフェイスが必要になる．この論文では，クラウドの拡張機能として，センサの管理を統合することによってクラウド上でセンサを扱いやすくするための統合管理アーキテクチャと，その実装方式について述べる．クラウド・ユーザは，本提案方式を使うことでセンサをプロダクション・システムに利用することができる．

1. はじめに

クラウドは，IT システムの自動化や省力化において，必要不可欠なサービスになってきている．多くの企業や組織は，商用としてクラウド・プロバイダから提供されているクラウドのサービスをプロダクション・システムの一部として利用するようになってきた．さらにクラウドは，Docker, Kubernetes, Istio, Knative に代表されるコンテナ，サービスメッシュやサーバーレス技術が追加されたことによって，以前よりも高度化している [1].

一方，IoT では，センサを使った多くのアプリケーションやソリューションが，研究開発から実用システムとして利用がはじまっている [2]．現在のセンサや IoT の研究開発の多くは，センサデータの収集や処理が中心である．このセンサデータを利用して医療，建設，土木，農業のような様々な分野のアプリケーションやソリューションに IoT が適用されはじめている．また，ワイヤレス・センサネットワークの分野では，センサノードの自律的制御やセンサノード間のネットワークの研究が行われている [3].

今まで筆者らの研究で仮想センサと仮想センサグループと，それらを実現するセンサ・クラウドの提案と実装を行ってきた [4] [5]．そこでは，物理センサを仮想化することによってセンサをクラウド資源の一部として利用する方式の基礎を確立した．しかし，クラウド一部としてセンサを管理するとき，センサとクラウドの統合管理やアプリケーションからセンサにアクセスするための共通 API が必要になってくる．この論文では，プロダクション・システムにおいて，論理センサを利用するときに必要となるクラウドとセンサの統合管理と，センサにアクセスするときに必要となる共通 API に関するアーキテクチャとその実装方式について述べる．クラウドのユーザは，この提案方式を利用することによって，論理的なセンサをクラウド・サービスの一部としてプロダクション・システムに利用することが可能になる．この論文は，以下の構成になっている．「関連研究」では，クラウドの拡張とセンサ管理に関連した研究について述べる．また，「アーキテクチャ」では，提案アーキテクチャについて述べる．「実装について」では，クラウドの拡張したセンサ管理の実装について述べる．「おわりに」では，論文のサマリと今後取り組むべきことを述べる．

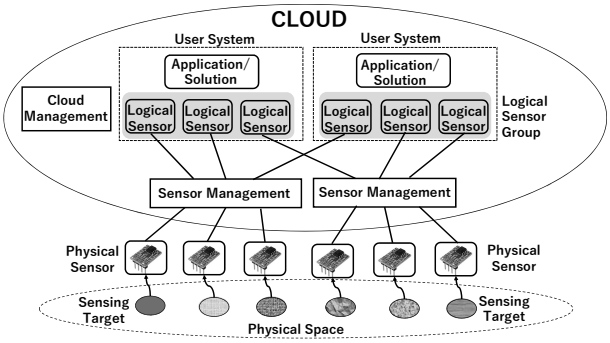

図 1 アーキテクチャ

2. 関連研究

IoT を使いやすくするためにソフトウェア定義 IoT の方式を提案している [6]．この研究では，提案方式を実現するためにソフトウェア定義のゲートウェイを実装して，このゲートウェイを経由して外部からのアクセスを一元管理し

[1] 東京工科大学コンピュターサイエンス学部 (kushida@acm.org)

ている．また，IoT の資源や構成を API により定義できるようにして，従来よりも IoT 資源を使いやすくしている．別な研究では，開発，デプロイ，運用，最適化，プロダクションのそれぞれのフェーズにおいて，IoT システムの 7 つの原則がどのように適用されるかを定義して，iCOMOT と呼ばれるツールセットを IoT システムとして提案している [7]．このツールセットは，IoT ユニットからのデータ転送や IoT ユニット自身のガバナンスを可能にしている．これらの関連研究は，IoT のセンサを使いやすくしたり，データを適切に転送することにフォーカスしている．しかし，クラウドの一部としてセンサを統合管理することについて十分に言及されていない．

筆者らの以前の研究では，物理センサを仮想化して利用することと，その仮想化によって実現されるセンサクラウドを提案した [4] [5] [8]．この研究の提案では，センサをクラウドのリソースの一部として使用することにより，ユーザは，センサを CPU，メモリ，ディスクのようなクラウドリソースと同様に要求によって使用することができる．

この論文では，以前の研究成果を拡張して，プロダクションシステムに利用することを目的にクラウドとセンサの統合システム管理を提案する．この提案では，センサクラウドに実際の運用をするとき必要になるライフサイクル管理，ロギングモニタリング，高可用性についても提案する．

3. アーキテクチャ

この章では，センサデバイスをクラウドリソースの一つとして取り扱うためのアーキテクチャについて述べる．図 1 が，クラウドとセンサの統合システム管理のアーキテクチャである．この図において，物理センサ（Physical Sensor）は，実世界 (Physical Space) にある対象 (Sensing Target) を測定している．データをデジタル化して取得している．この物理センサは，ネットワーク機能を持っていて，センサ管理 (Sensor Management) に接続している．また，この図において論理センサ (Logical Sensor) は，センサ管理と接続して，物理センサからのセンサデータを共通 API によってアクセスすることを提供するソフトウェア・コンポーネントである．アプリケーション・ソリューション (Application/Solution) は，共通 API にアクセスすることによってセンサデータやセンサの管理情報を取得することができる．また，この論理センサは，プロビジョニングの要求によって，クラウド管理 (Cloud Management) によって生成されるソフトウェア・コンポーネントである．ソフトウェア・コンポーネントであるため，同じ論理センサが複数のユーザシステム (User System) に同時に生成して存在することができる．

物理センサは，所有者がいてそのセンサが設置された場所で管理されている．また，最近の物理センサは，ネットワークに接続されて取得したデータ転送機能を持っている．しかし，現在の研究開発の主な目的は，センサでデータを取得して転送することと，転送された後のセンサデータを一箇所で集めて検索できるようにすることであって，物理センサを共有することについて，特に注目はされていない．また，通常の IoT の研究開発では，センサデータを一括処理することで，目的に応じて，データ解析をしている．そのため，複数のオーナーが所有する物理センサを組み合わせた構成によるアプリケーションやソリューションを構築することは，難しくなっている．現在の IoT で使用されている物理センサは，色々なユーザからの同時利用に対して，十分なシステムの設計になっていないため，使いにくいか利用用途に制限があることが多い．

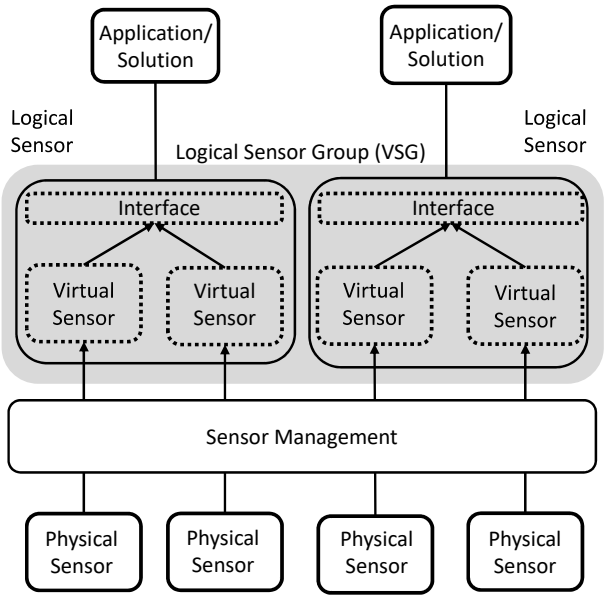

図 2 論理センサと論理センサグループ

この論文で提案しているクラウドとセンサの統合管理を導入するによって，センサの物理的な制約に起因する問題をなくすことができる．例えば，物理センサであれば，それぞれの物理センサが所有者によって管理されていて，データの提供方法や物理センサへのアクセスは，所有者によって異なっている．また，物理センサの情報について，それぞれの所有者に問い合わせることが必要になる．一方で，論理センサを導入して，その一般のユーザが，自分の用途に合わせた論理センサを生成することで，所有者以外の一般のユーザであっても，また遠隔地であったとしても，物理センサと同等の機能をアプリケーションやソリューションから利用することができる．また，複数の異なったセンサが複数の所有者になっていたとしても，同じアクセス方法で論理センサをプロビジョニングして，共通の API 経由でアクセスすることができる．

図 2 が，論理センサと論理センサグループ (Logical Sensor Group) である．論理センサは，1 つ以上の仮想センサとインターフェイスから構成されている．2 つ以上仮想セ

ンサの場合，それぞれのセンサが同じ機能であって高可用性の構成である場合は，どちらかが落ちた時にもう一つの仮想センサが，継続してデータを転送する．また，それぞれのセンサが異なるセンサである場合，論理センサは，複合したセンサをもつ論理センサとして動作する．例えば，それぞれの仮想センサが，温度センサと湿度センサであったときにこの 2 つをもった論理センサは，実質的には温度・湿度センサになり，それらのデータを同じ論理センサのインターフェイスから提供することができる．このように論理センサを導入することによって，センサを複合センサとすることができ，サービスに多様性を持たせることができる．

また，論理センサグループは，1 つ以上の論理センサから構成されたグループ機能である．個別の論理センサは，CPU, メモリ，ディスク，ネットワークようなクラウドのリソースに比べて小さく，一つのアプリケーション・ソリューションに多くの論理センサを必要とするため，論理的にまとめてグループ化して管理できるようにしている．これが論理センサグループである．そのため，クラウド管理機能は，論理センサグループごとに管理を行うことができる．

	物理センサ	論理センサ
実装方式	ハードウェアとソフトウェア	ソフトウェアのみ
データソース	物理量から取得	物理センサからのデータ
データ提供方式	独自方式	共通 API
管理方式	個別の管理	共通の管理
同時アクセス	データを共有	複数の論理センサ

表 1　物理センサと論理センサの比較

表 1 は，物理センサと論理センサを比較を示したものである．実装方式，データソース，データ提供方式，管理方式，同時アクセスについて，物理センサと論理センサをそれぞれ比べて示している．論理センサは，ソフトウェアによって構成されているので，ユーザの要求によって生成，更新や削除をすることができ，またセンサデータやデバイス情報は，共通 API によって提供される．

論理センサは，ソフトウェアなので一つの物理センサであったとしても同時に複数の論理センサを作って動作させることができる．この機能を使うことによって同時に複数のユーザが，1 つ物理センサに対応する論理センサを個別に所有して使うことができる．図 3 は，一つの物理センサが 3 つ論理センサとして構成されて動作しているところを示している．この図で 3 つの論理センサは，同じ機能をもっているが異なったユーザが，User System I, II, III の異なるアプリケーション・ソリューションで，それぞれ独立して使用していることを表している．論理センサによって，物理センサの共有と独占を実現することができるようになる．

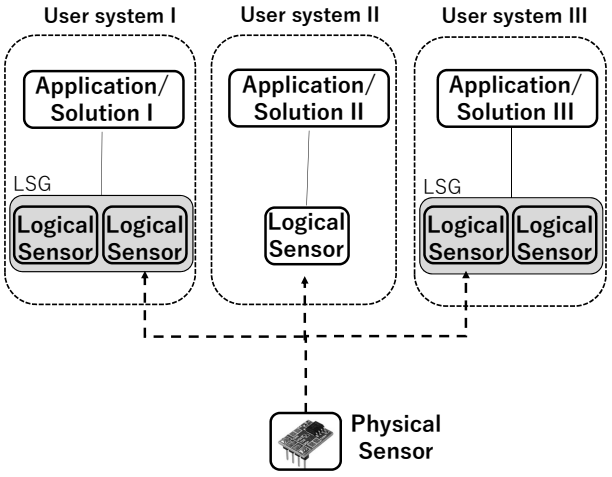

図 3　一つの物理センサと複数の論理センサ

3.1　温度センサ

図 3 をもとに，物理センサとして温度センサを例に論理センサを説明する．温度センサは，温度を測定して温度データをディジタル化して提供する機能を持っている．また，この温度センサは，ネットワーク経由で測定した温度データを転送する機能を持っている．この温度センサを物理センサの例として，論理センサである「論理温度センサ」を説明する．

論理温度センサは，温度センサと同じセンサデータを提供するようにソフトウェアとして構成される．しかし，実際の温度センサは，設置された場所で温度を測定してデジタル化してデータを転送している．そのため，論理温度センサは，直接の測定やディジタル化はするわけでなく，その代わりに温度センサからのデータをセンサ管理（ソフトウェア）経由で受け取って，温度データや管理データをインターフェイス経由でユーザに提供する．そのため，この論理温度センサは，ネットワークでアクセス可能な共通 API を持つ．この API を使うことによって，アプリケーションやソリューションからセンサデータやセンサのハードウェア特性情報を取得できる．このことから，論理温度センサは，他のソフトウェア・コンポーネントからは，温度センサ（物理センサ）と同等の機能を持っているようにみえる．

図 4 は，温度センサ (物理センサ）と論理温度センサのデータフローである．この図において，センサデータと管理データの 2 つデータが，それぞれ転送されてアプリケーション・ソリューションに提供されている．センサデータは，測定データのため，その都度データの値が変わるが，一方で管理データのうち特にデバイスの制度や特性データは，デバイスが使用されている限り常に同じデータ値になる．そのため管理データは，頻繁に送られることがなく，一定間隔で同じデータが転送して論理センサに記録される．論理センサでは，それをキャッシュ・データとして記録しておき，API 経由でリクエストがあったときに返答と

して送る．キャッシュの期間が切れた時点で管理データを温度センサから取得する．この機能によって，管理データも更新することを可能にしている．

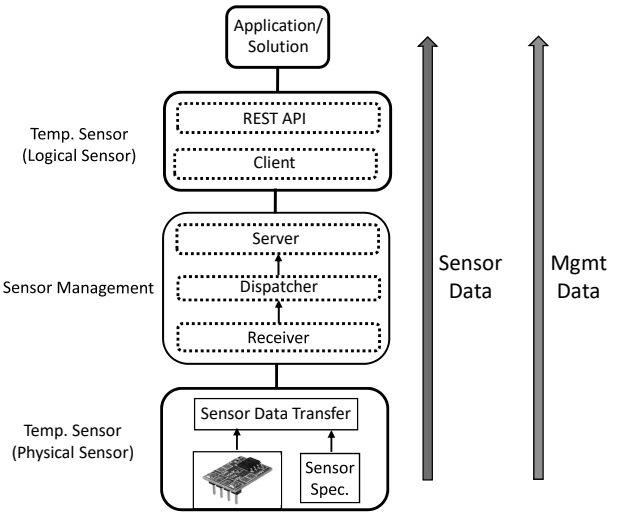

図 4　温度センサと論理温度センサ

3.2　センサ管理

物理センサの仕様：論理センサは，物理センサが持っているデバイス情報やプロパティを提供することができる．従来まで物理センサに対する仕様は，センサデータと別に入手する必要があった．例えば，A 社の a センサから出てきているデータと，B 社の b センサを使って出てきているデータは，それぞれの物理センサによって仕様が異なっているために，データの精度や特性が異なる．そのため，これらの物理センサに対する仕様の情報は，センサデータとは別に入手してデータを解析するときに，仕様やセンサの所有者についての情報を取得する必要があった．

例えば，温度センサであれば，温度測定に対する特性がある．この特性は，デバイスやその測定温度によって異なり，約数パーセント程度の誤差や違いがある．この特性や誤差は，温度センサを製造した会社によっても異なる．また，温度がディジタル化されて提供できるデータでも，小数点以下の桁数が異なる．温度の測定範囲もセンサによって異なる．このようなセンサのハードウェア仕様は，それぞれのハードウェアによって異なるにもかかわらず，センサデータと同じように，システムにアクセスして簡単に入手できるようにはなっていない．

物理センサと論理センサの監視：物理センサは，物理的な測定装置で通常，リモートに設置されている．そのため，物理センサが正しく動作しているかを監視することが必要であり，この監視はシステム管理のうちの重要な機能の一つである [9]．監視のアプローチとして，クラウドの CPU, メモリ，ディスク，ネットワーク，ソフトウェア，サービスの管理と同様のアプローチで行う．また，物理センサが，

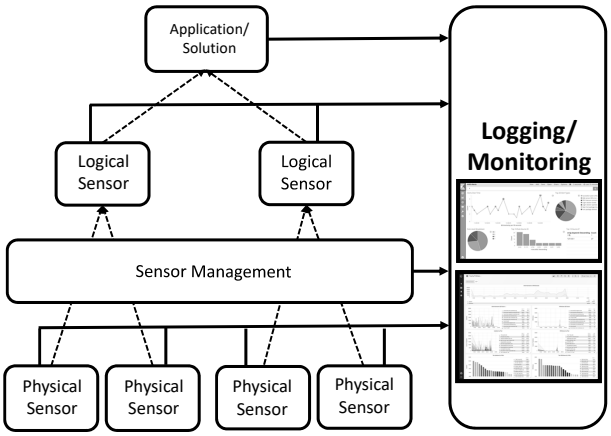

図 5　ロギングとモニタリング

物理量を測定して，そのデータが正常に生成されていることを監視することも必要である．さらにセンサデータの値が正しいことを監視することによって，物理センサのデータに対する信頼性をユーザに提供することができる．

論理センサは，ソフトウェアのコンポーネントとして実装され，常に物理センサと同じ役割を果たす必要があるため，機能が遅延なく動作していることと，物理センサの値が正しいかを判定する必要がある．例えば，物理センサが何らかの理由で故障してデータを出さない場合，物理センサが故障であることを論理センサは，ステータスとして提供する必要がある．論理センサが，遅延なく動作していることを監視するため，この研究では，クラウドで使用されているモニタリングとロギングサービスを利用する．クラウドのこれらのサービスは，クラウドのリソースやサービスの監視に利用されている．

図 5 は，物理センサと論理センサの監視機能である．図 5 において物理センサ，センサ管理，論理センサ，アプリケーション・ソリューションのからのログメッセージとモニタリング・データが，ログ・モニタリングサービス (Logging/Monitoring) に送られる．ログ・モニタリングサービスで記録されて，ダッシュボードに表示され，また異常があればアラート生成して管理者に送る．

高可用性 (High Availability)：プロダクションシステムのときに，重要になる非機能要件は，高可用性である．論理センサは，まったく同じ機能をもつ複数の物理センサを統合して，センサ機能をアプリケーションやソリューションに提供することによって，高可用性を実現する．

図 2 は，1 つの論理センサーに 2 つの仮想センサがあり，それぞれの仮想センサは，物理センサとのリンクがある．高可用性を実現するため，これら 2 つの物理センサは，同じ測定を行なうこととする．通常の運用の場合，論理センサは，1. 2 つのデータを同時に提供するか，2. どちらか一方を提供する，3. 2 つのデータの平均値を提供するという方法を選択することができる．

例えば，故障したときの運用を考える．2 つの物理セン

サのうち，1つが故障して動作しなくなってセンサデータが転送されなくなった場合，もう一方の物理センサが動作していれば，論理センサは，センサデータを動作している一方だけに切り替えて，データを途切させることなく，アプリケーション・ソリューションに要求に応じて送ることができる．この方法によって，論理センサにおいて高可用性を実現することができる．

プロビジョニング機能：クラウド・サービスとしてライフサイクル管理するとき，論理センサを生成や消滅させるためのプロビジョニングとデプロビジョニング機能が重要になる．図6は，新しい論理センサがプロビジョンをされているところである．エンド・ユーザは，必要となったときにクラウドに対して新規に論理センサをリクエストする．リクエストを受けたクラウドは，新規に論理センサをプロビジョングして，その論理センサが所属する新規の論理センサグループを作成することができる．また，ユーザが，既存の論理センサグループに追加することもできる．どちらの場合でもクラウド管理に対して，論理センサを加入させることをリクエストする．クラウド管理は，論理センサと論理センサグループを適切に操作することによって，アプリケーション・ソリューションから利用できるようにして，ユーザに引き渡す．同様に論理センサが必要なくなったときは，デプロビジョニングとして論理センサグループから削除することも可能である．

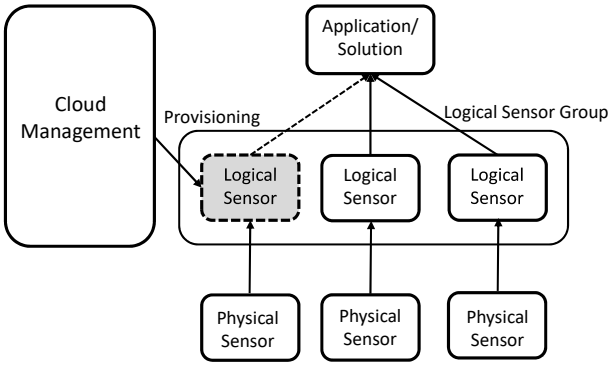

図 6　論理センサのプロビジョニング

4. 実装

この章では，クラウド・センサ管理の実装について述べる．図7は，論理センサグループ，センサ管理，アプリケーション・ソリューションを実装するときの構成である．それらすべては，クラウドのコンテナとして実装される [10]．コンテナによる実装によって，OSやライブラリによるメモリのフットプリントを小さくすることができ，また，コンテナの数を増やすことによってスケーラビリティーを確保することができる．

また，モニタリングやロギングの監視機能は，クラウドのPaaS(Platform as a Service) を使って実装される．

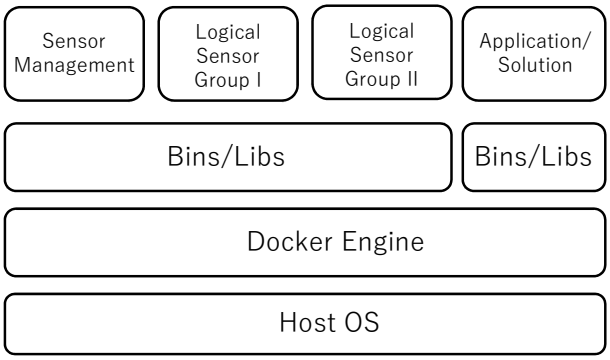

図 7　論理センサをコンテナとして実装

```
{
  "Sensor" : "Temperature Sensor",
  "ID" : "002",
  "Device" : "ADT7410",
  "Manufacturer" : "Analog Device",
  "Specifications" : {
    "ADC Resolution" : 16,
    "Precision" : 0.0078,
    "Conversion Time" : 240,
    "Fast Converstion Time" : 6,
    "Hysteresis" : 0.002,
    "Repeatability" : 0.015,
    "Accuracy" : "plus/minus 0.5",
    "Interface" : "I2C",
  },
  "URL" : "https://www.analog.com/media/en/
      technical-documentation/data-sheets/
      ADT7410.pdf",
  "TTL" : 3600
}
```

図 8　温度センサの仕様 (JSON)

4.1 コンテナによる実装

図7は，論理センサとアプリケーション・ソリューションをコンテナとして実装したときのアーキテクチャである．コンテナ技術は，独立したソフトウェア実行環境をライブラリで提供するため，仮想マシン (Virtual Machine) よりも少ないコンピューター資源で実現することができる．図7は，Host OSの上にDocker Engine(コンテナのためのエンジン) あり，Bin/Libsを使うことで，それぞれの機能を実装している．

4.2 REST API 経由でのアクセス

アプリケーショ・ソリューションから論理センサに対するアクセスには，REST APIを使用する．論理センサは，コンテナを使ってREST APIによるアクセスをすることで，独立したコンポーネントとして定義され実装することができる．このアイソレーションによって論理センサは，クラウドのプロビジョング，モニタリングやロギング機能から個別に識別して使用することができる．

表2は，REST APIによる永続性のCRUD操作を表し

ている．登録は，特定の論理センサとの接続する．この登録操作で接続 ID を取得する．取得は，接続 ID のセンサデータや管理データの取得を行う．更新は，接続 ID の論理センサとの接続を更新する．削除は，接続 ID の論理センサとの接続を解除する．また，論理センサをコンテナに

処理	HTTP メソッド	CRUD 操作	論理センサ	アクション
登録	POST	CREATE	接続の生成	/api/v1/sensor
取得	GET	READ	データ取得	/api/v1/sensor/sensorId
更新	PUT	UPDATE	接続の更新	/api/v1/sensor/sensorId
削除	DELETE	DELETE	接続の解除	/api/v1/sensor/sensorId

表 2 論理センサへの CRUD 操作

実装するとき，1 つのコンテナに複数の論理センサを同時に動作させることが可能である．また，論理センサは，ソフトウェアによって実装されているので，論理センサの処理は，アクセスリクエストによって API 経由でデータを提供することである．そのため，コンテナに十分なメモリが与えられ，データ転送処理が十分であれば，多数の論理センサを一つのコンテナに配置することが可能である．この構成よって少数のコンテナで，多数の論理センサを利用することができる．

4.3 センサの例

ここでは，アナログデバイス社の温度センサ ADT7410 を例にして提案方式について述べる [11]．温度センサ ADT7410 は，電源電圧が 2.7V から 3.6V の範囲で，測定温度が-40°C から+105°C の範囲あれば，±0.5°C の精度で測定することができる．またセンサの測定レンジは，-55°C から+150°C である．このデバイスは，最大 16 ビットの温度分解能があるので，0.0078°C の分解能で測定結果の温度を出すことができる．電源を入れてから 6msec 以上経過していれば，温度変換することができる．このデバイスは，ユーザによる温度の校正や線形性の修正を必要としないため使いやすくできている．また，ホストコンピュータからは，I^2C インターフェイス経由でデバイスの設定したり，センサデータの取得を行うことができる．

この温度センサは，16bit の分解能であるの，例えば，23.123°C の温度データを測定したとき，±0.5°C の精度で，実際のデータは，23.123°C (±0.5°C の精度) のように処理することが必要になってくる．そのため，アプリケーションやソリューションが温度データを処理するとき，温度データに加えて，これらの測定デバイス仕様の情報が必要となってくる．さらに温度センサの測定場所の情報（ロケーションデータ）や測定時間（タイムデータ）やその精度に関するデータも同様に必要となる．

さらに，これらの仕様やそれに関連するデータは，TTL(Time To Live) を付加して物理センサから転送して論理センサに一時的に記録する．論理センサでは，TTL が 0 になる前に，新しいデータを入手して最新のデータにすることで常に最新のデータを保持する．TTL は，物理センサに付随する情報なので，物理センサのところで設定される．センサ管理では，このデータを取得して論理センサが生成されたときに論理センサに転送してアプリケーションやソリューションからアクセスできるようにする．図 4 の管理データ (Mgmt Data) で示されている流れは，このようなセンサの仕様を転送するためのデータ・パスである．このデータは，センサデータとは，別な独立したパスで転送することになる．

図 8 は，温度センサの仕様データを JSON で表現したものである．論理温度センサ，仕様のデータをストアして，アプリケーションやソリューションに提供することによって，センサの所有者でもなく，また遠隔であっても，物理センサを使って結果を得たことと，同等の処理を行うことができる．

4.4 課題

物理センサの信頼性：従来の技術は，センサを設置したオーナが正確に設置して，設置された物理センサの精度が高く正確なデータを出すことを暗黙の前提条件として，センサデータを処理していた．また，センサ管理は，全て人手で行われていて物理センサに対する前提条件として，取得したデータを暗黙に信頼することによってデータを処理し，その解析結果も信じる必要があった．

一般的に設置されている物理センサは，リモートロケーションに設置されていることになる．また，論理センサを導入してサービス化すると，物理センサの所有者と利用者は，異なる組織でお互いに知らないことが普通のことになる．そのため，センサとそこから出てくるセンサデータをどの程度信用して良いのかが，明確ではない．センサとセンサデータを信頼するため，センサの所有者あるいは設置者を信用するためのなんらかの方法を導入する必要がある．例えば，信用できるセンサには，信頼していることを示すための Certification を付加する．また，過去に物理センサにアクセスして利用した人の結果やコメントを公開する．過去に物理センサが，いくつかのアプリケーションにデータをどの程度の期間提供したかを信頼できるかどうかの情報として提供することも考えらえる．

データの遅延：提案方式では，センサ管理や論理センサを入れることによって，センサデータを転送して処理する．そのため，データの提供に遅延が生じる．温度センサ（物理センサ）で取得した温度データが，アプリケーション・ソリューションで共通 API で取得するときに途中のソフトウェアコンポーネントがあるため処理遅延がある．しかし，論理センサの数が増えたり，途中のサーバでのデータの処理が増えたとしてもデータ提供するときの遅延時間は，一定にする必要がある．また，センサのハードウェアが持っている変換処理の遅延があったとしても，そこから

共通 API までの遅延を最小限にすることによって，現実に測定されたデータの取得を短時間でできる．測定結果を短時間で得られるため，提案方式をさらに広い範囲のアプリケーションやソリューションで利用することができる．このよう実時間での処理するためには，リアルタイムの技術を導入する必要がある．

5. おわりに

今後，課題に上げた物理センサの信頼性やセンサデータの遅延に加えて，提案方式に対するユーザやソフトウェアからのアクセス管理，多数のデバイスを統合管理するためのスケーラビリティ，災害が起こったときのディスアスタ・リカバリについても研究を進めていく必要がある．

本論文は，論理センサと論理センサグループと呼ばれる概念をクラウド上に定義して，クラウドとセンサの統合管理の方式について提案して，その実装について述べた．また，プロダクション・システムを構築するときに必要となる監視機能，高可用性，プロビジョニング機能について論理センサと論理センサグループへの適用方法について述べた．

参考文献

[1] E. Jonas, J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Menezes Carreira, K. Krauth, N. Yadwadkar, J. Gonzalez, R. A. Popa, I. Stoica, and D. A. Patterson, "Cloud programming simplified: A berkeley view on serverless computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2019-3, Feb 2019.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[4] H. Takahashi and T. Kushida, "Innovative iaas management system for sensor devices and it resources," *International Journal of Distributed Sensor Networks*, vol. 10, no. 6, p. 931968, 2014.

[5] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure-physical sensor management with virtualized sensors on cloud computing," in *2010 13th International Conference on Network-Based Information Systems*. IEEE, 2010, pp. 1–8.

[6] S. Nastic, S. Sehic, D.-H. Le, H.-L. Truong, and S. Dustdar, "Provisioning software-defined iot cloud systems," in *2014 2nd International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2014, pp. 288–295.

[7] H. Truong and S. Dustdar, "Principles for engineering iot cloud systems," *IEEE Cloud Computing*, vol. 2, no. 2, pp. 68–76, Mar 2015.

[8] M. Yuriyama, T. Kushida, and M. Itakura, "A new model of accelerating service innovation with sensor-cloud infrastructure," in *2011 Annual SRII Global Conference*. IEEE, 2011, pp. 308–314.

[9] B. T. Sloss, S. Nukala, and V. Rau, "Metrics that matter," *Commun. ACM*, vol. 62, no. 4, pp. 88–88, Mar. 2019. [Online]. Available: http://doi.acm.org/10.1145/3303874

[10] C. Boettiger, "An introduction to docker for reproducible research," *SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 71–79, Jan. 2015. [Online]. Available: http://doi.acm.org/10.1145/2723872.2723882

[11] *Data Sheet: ADT7410 Temperature Sensor (Revision C)*, Analog Devices, 2017. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADT7410.pdf