

個別構築された IoT システムのサービス連携における課題検討

平山秀昭^{†1} 鄭顕志^{†2} 清雄一^{†3} 大須賀昭彦^{†3}

近年, M2M (Machine to Machine) あるいは IoT (Internet of Things) と呼ばれる技術が脚光を浴びている. M2M や IoT は, 現実世界のありとあらゆる物を, 人の手を介することなくネットワークに繋げる. M2M や IoT がセンサーの小型化, ネットワーク化により身近なものとなり, それがビッグデータを生み, 更にはスマートシティ, スマートコミュニティ実現の原動力となりつつある. その際, 複数の M2M, IoT システムを連携させることで, より大きな価値を生むシステムの構築が可能になると考えられる. しかし, 実際には個別に構築された M2M, IoT システムを連携させるには様々な課題がある. 本稿では, この連携システムの適用領域を選定した上で, その課題と解決策についての検討結果について報告する.

Study about Mashup of the Services which are Provided by IoT Systems Built Individually

HIDEAKI HIRAYAMA^{†1} KENJI TEI^{†2} YUICHI SEI^{†3} AKIHIKO OHSUGA^{†3}

Recently, M2M (Machine to Machine) or IoT (Internet of Things) is attracting attention. It connects everything in the real world to the Internet without operator's assistance. M2M or IoT has become familiar, because of miniaturization and connectivity of sensor devices. Then it is generating Big Data and becoming the driving force of realization of Smart City or Smart Community. If we can mashup the services of several M2M or IoT systems, we will be able to develop the systems which are producing greater value. But there are various problems to mashup the M2M or IoT systems. In this paper, we select an application domain of this mashup system and study its problems and solutions.

1. はじめに

近年, M2M (Machine to Machine) あるいは IoT (Internet of Things) と呼ばれる技術が脚光を浴びている. M2M や IoT は, 現実世界のありとあらゆる物を人の手を介することなくネットワークに繋げる. M2M と IoT の違いは明確ではなく, 同じ意味で使われることが多い. ただし, M2M は比較的大型の機器の監視・制御用途のシステム等を示す場合に使われる傾向が強い. 一方, IoT は多数のセンサー機器から得られる情報に基づくシステム等を示す場合に使われる傾向が強い. 本稿では, 以降, M2M, IoT という言葉を前述の意味で使い分けていく. 前者の例としては, 建設機械, エレベータ, 医療機器等の遠隔監視システム等があげられる. このようなシステムは以前から実用化されている. 後者の例としては, 人, 車, 道路, ビル等, 様々な場所や物に埋め込まれる, あるいは付帯される様々なセンサーからデータを収集し可視化するシステム等があげられる. これらはセンサーネットワークの研究からの発展形と位置付けられる.

ここ数年, スマートシティ, スマートコミュニティという言葉よりも, ビッグデータという言葉がより頻繁に聞か

れるようになってきている. M2M や IoT がセンサーの小型化, ネットワーク化により身近なものとなり, それがビッグデータを生み, スマートシティ, スマートコミュニティ実現の原動力となりつつある. この際, 複数の M2M, IoT システムを連携させることで, より大きな価値を生むシステムの構築が可能になると考えられる. しかし実際には, 個別に構築された M2M, IoT システムを連携させるには様々な課題がある. この課題について検討するには, 連携させる M2M, IoT システムが多岐に渡るため議論が発散しかねない. そこで本稿では, まず適用領域を選定する. その上で, 適用領域における M2M, IoT の連携システムについて, その課題と解決策を検討した.

2. M2M システム

2.1 現状の M2M システム

エレベータ遠隔監視, 医療機器遠隔監視, 自動販売機在庫管理等, 現状の M2M システムは個別構築された垂直統合型のシステムとなっている. そのため, 複数サービス間での相互連携が困難だという課題がある[1]. 図 1 は, 個別構築された垂直統合型のシステムを示す. センサー等の M2M デバイスは, ゲートウェイを経由してデータをクラウドに送る. ただし, ゲートウェイを介さず, M2M デバイスからクラウドに直接データを送ることもある. クラウドはデータを蓄積すると共に, データを用いたサービスを提供する. A から C のシステムは個別構築されインタフェースが共通化されていないため, それらが提供するサービスの連携は

^{†1} 東芝ソリューション株式会社ソリューションセンター
Solution Center, Toshiba Solutions Corporation.

^{†2} 国立情報学研究所先端ソフトウェア工学・国際研究センター
Center for Global Research in Advanced Software Science and Engineering,
National Institute of Informatics.

^{†3} 電気通信大学大学院情報システム学研究所
Graduate School of Information Systems, The University of
Electro-Communications.

困難になっている。このこと自体は正しいと考えるが、M2Mシステムは多岐に渡るため、課題を整理していく上では焦点が定まり難い。そのため、本稿では研究対象とするサービスの領域を定めた上で課題検討を行っていく。

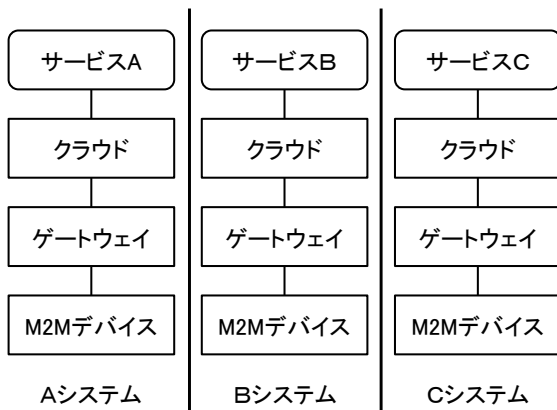


図1 個別構築された垂直統合型のM2Mシステム

Figure 1 Vertically integrated M2M systems developed individually

2.2 研究対象とするサービス

本稿では、M2Mシステムが対象とする比較的大型の機器に対するサービスではなく、IoTシステムが対象とするセンサーを活用したスマートシティを実現するサービスを対象として選んだ。理由は地球上の様々な場所や物にセンサーを埋め込むIoTシステムの連携は、防災や環境対策を拡張する安心で安全な未来社会に繋がると考えたからである。この様なサービスを目指す例として情報流[2]の研究やClouT Project[3]がある。

情報流とは多数の実時間データストリームの集合である。この情報流を実時間で融合・編纂・配信する実時間キューレーションの実現を目指している。実時間データストリームに対し、キューレーションで価値ある情報を付加するには、実時間データストリームに加え、関連データの参照が必要になると考える。

ClouT Projectでは、実世界の情報をビジネスプロセス・都市のプロセスに統合すること、クラウドモデルによりスマートシティの革新的なソリューションを構築するコスト・労力を削減し、より多くの中小企業を含めたエコシステムを加速することを目指している。ClouT Projectでは研究チャレンジとして以下の点をあげている。

- 異種混在なIoTとクラウドシステムのエコシステムにおける互換性・相互運用性
- 実時間でのIoTデータに対するクラウド上での保存、処理、アクセス
- IoTデータをオープンデータとしてクラウドに移行
- ディペンダブルなデータマッシュアップ・IoTサービス

ス合成

- データ・イベントのストリーミングにおける自己修復
- 開発者・起業家によるデータの再利用を通じた新たなサービス構築の促進
- 市民自身が都市のエコシステムに参加しDIYを通して共有し恩恵を受けることの促進

これらの課題のうち最初の4つは、「複数サービス間での相互連携が困難」という個別構築された垂直統合型のM2Mシステムの課題に関するものである。「実世界の情報をビジネスプロセス・都市のプロセスに統合する」というClouT Projectが目指すシステムにおいて、課題が具体化されていると捉えることができる。

3. IoTサービス連携システム

前節で対象と定めたIoTサービスの連携システムについて検討する。図2は個別構築された垂直統合型のIoTシステムのサービス連携システムを示す。サービス連携システムは、AからCのシステムが提供するサービスを連携させるものである。そしてAからCのシステムが提供するIoTシステムのデータを組み合わせることで集計・分析する連携サービスを提供する。

このサービス連携システムは、前述の情報流の研究やClouT Projectが目指すようなシステムと考える。それを実現する上での具体的な課題を検討すると以下があげられる。

- データのディペンダビリティ
- 多様なデータの扱い
- サービスのスケラビリティ

データのディペンダビリティとは、ここでは必要なデータが確実に利用できるということの意味している。個別構築されたIoTシステムは、それぞれ目的が異なっている。そのためデータの保存期間も、そのシステムの運用方針に従っている。また、個別構築されたIoTシステム本来の目的があるため、そのシステムの運用に大きな影響を与えてはならない。

多様なデータの扱いとは、ここではセンサーデータは多様で共通化しきれないということの意味している。勿論、W3C Semantic Sensor Network Incubator Group[4]やoneM2M[5]等による標準化は進められプロトコル等の標準化は進むものの、多様なセンサーデータの形式等までは共通化しきれない。特定のIoTシステムを固定的に連携させるのではなく、全体をコントロールできないまま増加するIoTシステムの連携を考える必要がある。

サービスのスケラビリティでは、例えば災害時のアクセス集中時こそが、本稿で扱う連携システムの活用時だと

いうことである。アクセス集中時の対応を考えるとこの
 ではなく、使用時は常にアクセス集中時だと考える必要が
 ある。

次節以降で上記3点の課題について検討するが、その前
 に本稿で対象とするサービス連携システムが扱うデータの
 特性について考える。理由は、検討する課題が主にデータ
 に関するものだからである。

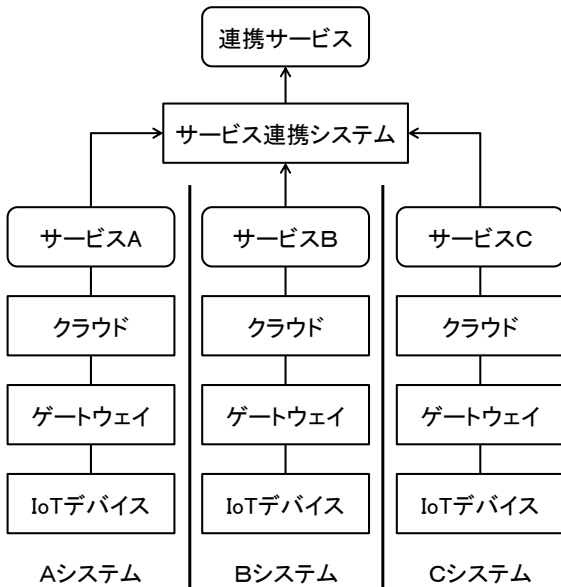


図2 IoT サービス連携システム

Figure 2 IoT service integration systems

3.1 サービス連携システムにおけるデータの特性

本稿で扱う IoT サービスの連携システムが扱うデータの
 特性は以下の様に捉えられる。

- データは時間軸に乗っている
- データは地域（位置）属性を持つ
- 個別構築されたシステムのデータ保存期間は保証され
ない
- データは実時間で参照される一方、関連データは時
間・地域指定で参照される
- 時間・地域指定で参照されたデータは再利用される可
能性が高い

データは時間軸に乗っているとは、本稿で扱うシステム
 はセンサーから収集したデータであるため、必ずセンサー
 からのデータ収集時間、あるいはイベント発生時間のよう
 な時間軸を持ったデータだということである。

データは地域（位置）属性を持つとは、データを収集す
 るセンサーは配備された地域あるいは位置に関する情報を
 持っているということである。時間と地域は、IoT システ
 ムにとって主要なキーとなるデータであり、他のデータと

は性質が異なる。

個別構築されたシステムのデータ保存期間は、個別構築
 されたシステムの個別方針次第であり、指定期間保存させ
 る等のコントロールは困難だということである。そのため、
 連携システムでデータを使用する必要性が生じた際に、そ
 のデータが保存されていることは保証されていない。

データは基本的には実時間で参照される。これは災害対
 策等を考えれば明らかである。一方で、1時間前のデータ
 と比較する、過去の災害のデータと比較する等、現在のデ
 ータと過去のデータを比較するケースが容易に考えられる。
 そのような関連データは時間・地域指定で参照することが
 多いと考えられる。

そして、このように時間・地域指定で参照されたデータ
 は比較対象となる事例データであるため、再利用される可
 能性が高い。よって、この時間、地域で参照されたデータ
 は長期保存されるべきである。以上が本稿で扱う IoT サ
 ービスの連携システムが扱うデータの特性である。

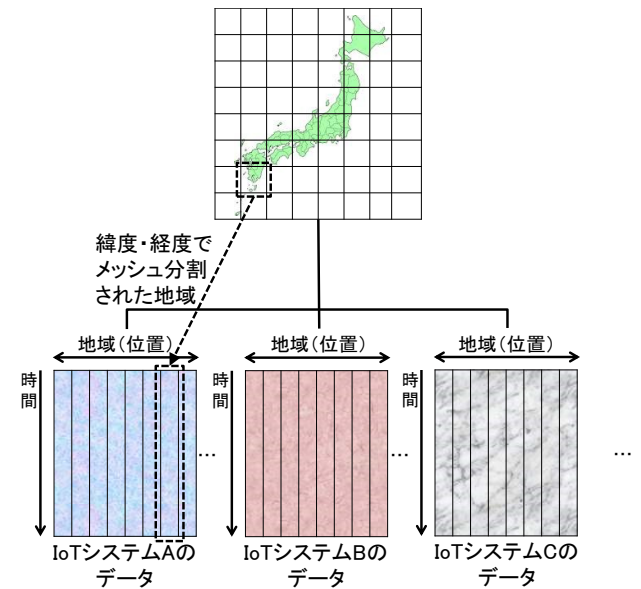


図3 地域指定方法

Figure 3 Way of region designation

3.2 データのディペンダビリティ

IoT サービスの連携システムでは、時間指定でアクセス
 されたデータの長期保存と個別構築システムへの影響の低
 減を考える必要がある。そのため、IoT サービスの連携シ
 ステムでは、時間指定でアクセスしたデータをキャッシュ
 し、キャッシュしたデータを用いた実時間での応答、そし
 て再利用されることを想定した長期保存を行う。それ以外
 のデータは連携元の IoT システムから取得する。キャッシ
 ュを有効に利用するため、本稿ではキャッシュの結合・一
 括化・拡張方式を提案する。

防災や環境対策等を対象サービスとして選定したので、

地域は緯度・経度でメッシュ分割した領域を基本に考える。理由は、過去の災害場所として選定した市町村等の地域からのマッピングが容易で、かつ、厳密な比較が可能になるからである。図3は地域指定を行うために地域を緯度・経度を基準にメッシュ分割したイメージを示す図である。このような地域指定方法を用いることで、異なるIoTシステムA, B, Cの同一地域のデータを指定し易くなる。

最初に、一つのIoTシステム内のメッシュ分割された一地域内のデータをキャッシュする場合について考える。従来、キャッシュはブロック単位の固定サイズあるいは全体一括で管理を行っていた。例えば、メモリはラインを単位に、ディスクはセクタを単位に、データベースはブロックを単位に、ウェブコンテンツはファイルを単位に管理を行っていた。

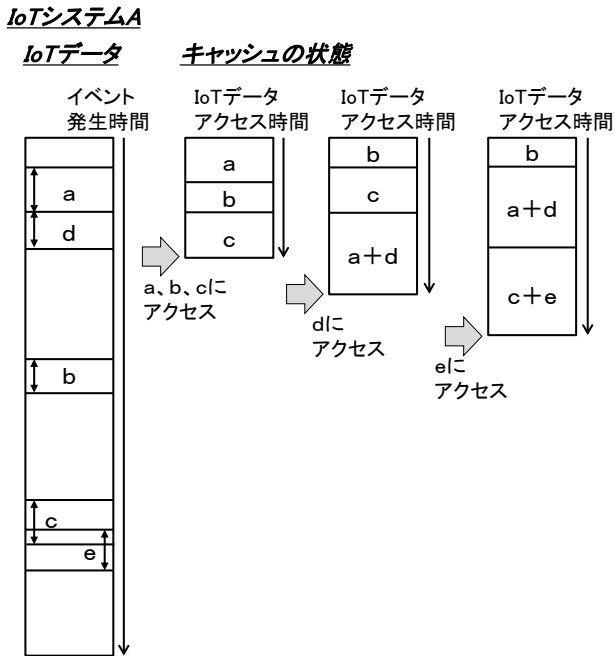


図4 キャッシュの統合・一括化方式
 Figure 4 Integration and gathering method of cache

本方式では、キャッシュを指定時間単位毎に可変サイズで管理する。更に連続・重複する時間帯は結合して一括化する。図4はこの方式を示す図である。最初に、IoTシステムAのIoTデータの時間帯a, b, cのデータがアクセスされ、キャッシュに記録される。次に時間帯dのデータがアクセスされる。時間帯aとdは連続していて、aとdは一括化したデータとしてキャッシュで管理される。その後、時間帯eのデータがアクセスされる。時間帯cとeは重複しているため、重複部分は既にキャッシュに記録されているデータが利用され、cとeは一括化したデータとしてキャッシュで管理される。

キャッシュされたIoTデータは、一括化されて管理され

る可変サイズのブロック単位で、LRUやLFU等のアルゴリズムにより、再利用の可能性を判断する。可能性が低いと判断した場合には、キャッシュから削除する。また、一つのIoTシステム内のメッシュ分割された一地域内でキャッシュされたデータを、他の地域のデータと比較し、時間帯に重なりがある場合は双方の地域のキャッシュを互いに双方の時間帯を含むように拡張する。そして、他のIoTシステムにおいても、同一時間帯、同一地域のデータが参照されることを予測して同一時間帯、同一地域のデータを自動的にキャッシュする。

3.3 多様なデータの扱い

IoTサービスの連携システムでは、センサーデータは多様で共通化しきれないことを考える必要がある。一方で、対象となるシステムの領域を選定すると、時間や地域(位置)のような特定のデータは共通化が容易だという面もある。すなわちIoTサービスの連携システムでは、時間・地域を絞った後、多様なデータ検索が行える必要がある。そのため、本稿ではメタデータ型と非メタデータ型のハイブリッドなデータ処理方式を提案する。

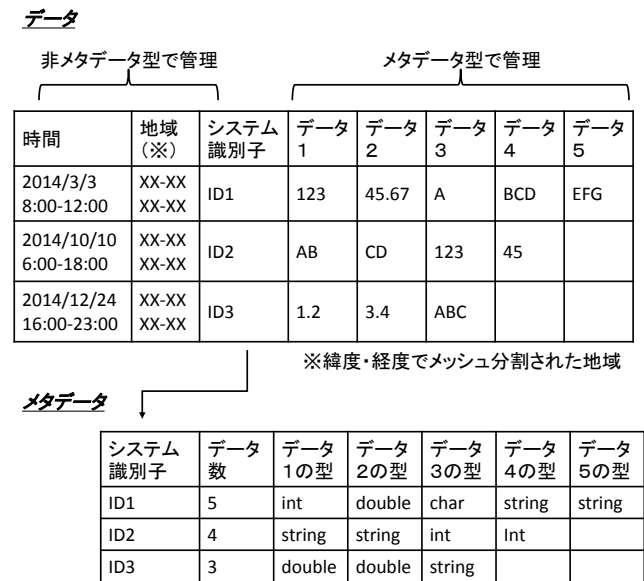


図5 メタデータ型と非メタデータ型のハイブリッドデータ管理
 Figure 5 Hybrid data management with meta-data structure and non meta-data structure

従来、データベースでは、データを非メタデータ型で管理していた。あるいは一部のマルチテナントシステムのデータベースでは、データをメタデータ型で管理していた。データをメタデータ型で管理している例としてはForce.comのシステムがあげられる[6]。

本方式では、一部の共通データ(時間, 地域)は非メタデータ型で管理し、それ以外のデータはメタデータ型で管

理する。図5は、この方式を示す図である。共通データである時間、地域は非メタデータ型で管理していて、各データの型は固定的に決まっている。一方、データ1からデータ5はメタデータ型で管理していて、各データの型はシステム識別子毎に異なる。各データの型はメタデータとして別テーブルで管理している。そのため、検索のキーとなる共通データ（時間、地域）には索引を用いた効率的な検索が行える。共通データによる検索後は、メタデータを用いて型を解釈しながら柔軟なデータ処理が行える。

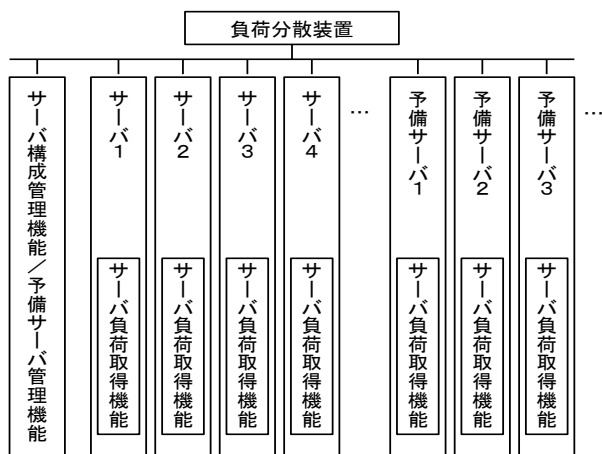


図6 スケーラブルなシステム構成

Figure 6 Scalable systems configuration

3.4 サービスのスケーラビリティ

IoT サービスの連携システムでは、例えば災害時のアクセス集中時こそが活用時だということを基本に、負荷に応じたスケールアウトが可能ないように予備サーバを配備した図6の様な構成をとる。図6ではサービス実行を行うサーバが1から4の4台ある。各サーバはサーバ負荷取得機能を備えていて自サーバの負荷を常時取得している。サーバ構成管理機能は各サーバの負荷を事前に設定された閾値と比較する。閾値を超える場合には予備サーバ管理機能が1から3の予備サーバを順次起動する。そしてサーバ構成管理機能は起動した予備サーバに負荷が分配されるよう負荷分散装置を設定する。逆にサーバの負荷が閾値を下回る場合には予備サーバ管理機能が1から3の予備サーバを順次停止する。予備サーバの起動数の上限・下限は予備サーバ管理機能が管理している。

前述のシステム構成は、予備サーバを起動しようが、しまいが予備サーバを準備していればコストがかかってしまうため、実際にはこのような構成を採ることは少なかった。しかし、クラウドでは予備サーバが物理サーバではなく、仮想サーバ (VM) となり、必要な時に予備サーバを確保できるため、前述の構成が採れるようになってきている。例えば、AWS (Amazon Web Services) のクラウドデザインパターン [7]には、Scale Out パターンがある。Scale Out パターン

は、ロードバランサ (ELB)、モニタリングツール (CloudWatch)、自動スケールアウト (Auto Scalability) の3つのサービスを組み合わせることで、負荷に応じたスケールアウトを実現できるようにしている。

4. 評価

前節で述べた連携システムにおけるキャッシュの必要性について考えてみる。日本の面積は 377,960 km² [8], そのうち平野は約 25%にあたる。平野では 100m四方あたりに 10個 (複数種類)、平野以外では 1km四方あたりに 10個 (複数種類) のセンサーを配備した場合、配備されるセンサーの数は以下のように 97,324,700 個となる。

$$(377,960 \times 0.25 \div (0.1 \times 0.1) + 377,960 \times 0.75 \div 1) \times 10 = 97,324,700$$

1個のセンサーが1分あたりに1イベントを発生した場合、1年間で発生するイベントの数は以下のように 525,600 イベントとなる。

$$60 \times 24 \times 365 = 525,600$$

以上から、日本中のセンサーが1年間に発生するイベントの数は以下の様に約 51T イベントとなる。

$$97,324,700 \times 525,600 = 51,153,862,320,000$$

1イベントのデータサイズを、主要な無線通信プロトコルの中で1回あたりの転送データ量が大きい ZigBee [9] にあわせて 127 バイトとすると、1年間で蓄積されるデータ量は以下のように約 6.5PB となる。

$$51,153,862,320,000 \times 127 = 6,496,540,514,640,000$$

1年間で約 6.5PB のデータが蓄積されるので、連携システムがキャッシュを持って再利用される可能性の高いデータを長期保存する必要がある。

また、連携システムが1回に使用するデータ量について考えてみる。ある地域の1日分のデータ、それと比較するための同じ地域あるいは他の地域の1日分のデータ、あわせて10件分を使用するとする。ある地域として、例えば川崎市の面積は、142.70 km² [10]である。川崎市で 100m四方あたりに 10個のセンサーを配備した場合、配備されるセンサーの数は以下のように 142,700 個となる。

$$(142.70 \div (0.1 \times 0.1)) \times 10 = 142,700$$

1個のセンサーが1分あたりに1イベントを発生した場合、1日で発生するイベントの数は以下のように1,440イベントとなる。

$$60 \times 24 = 1,440$$

1イベントのデータサイズを先程同様127バイトとすると、川崎市の1日分のデータ、それと比較するための同じ地域あるいは他の地域の1日分のデータ、あわせて10件分のデータ量は以下のように約260GBとなる。

$$142,700 \times 1,440 \times 127 \times 10 = 260,969,760,000$$

1年間で蓄積されるデータ約6.5PBの1%にあたる65TBを連携システムにキャッシュとする。65TBのデータベースをメタデータ型と非メタデータ型のハイブリッドデータとして管理する。65TBのデータは地域毎にパーティション分割すれば、地域をキーとして索引検索することでデータベースへの1回のアクセスはパーティション単位となり高速になる。10回のアクセスで得られたデータは合計で約260GBなので、メタデータ型のデータではあるが、メモリ上で処理可能なサイズなので高速に処理できる。よって、メタデータ型と非メタデータ型のハイブリッド型のデータ管理が有効である。

5. 関連研究

PRESTO[11]は、センサーネットワーク(IoTシステム)において、プロキシがインタラクティブクエリのためのキャッシュを提供する。キャッシュは予測技術を使って自律的な取得も行う。ただし、本提案方式のように、あるシステムでのアクセスに応じて、他システムでのアクセスを予測する方式は含まない。COLR-Tree[12]は、センサーネットワーク(IoTシステム)において、時空間クエリのためのインデックスとキャッシュを提供する。インデックスはRツリーの拡張である。本提案方式は空間情報をメッシュ分割することで検索性を高めていて、Rツリーに基づくものではない。SENSORID[13]は、時空間情報と共に保持したIoTデータ(センサーデータ)をルールに基づいて処理するものであり、本提案方式とは異なる。時空間センシングデータ集約送信技術[14]は、IoTデータ(センサーデータ)を時空間的に集約・圧縮するものであり、本提案方式とは異なる。

6. おわりに

現実世界のありとあらゆる物を人の手を介することなく

ネットワークに繋げるIoTシステムが脚光を浴びている。センサーの小型化、ネットワーク化によりIoTが普及し、それがビッグデータを生み、スマートシティ、スマートコミュニティ実現の原動力となりつつある。しかし、個別構築された垂直統合型のIoTシステムは、複数サービス間の相互連携が困難だという課題がある。本稿では、地球上の様々な場所や物にセンサーを埋め込み、防災や環境対策を拡張するようなサービスを想定した。その想定の下、個別構築された垂直統合型システムを相互連携させる上での課題は、データのディペンダビリティ、多様なデータの扱い、サービスのスケラビリティだと捉えた。そして、それらの課題を解決するために、サービス連携システムのキャッシュの結合・一括化・拡張方式、メタデータ型と非メタデータ型のハイブリッドなデータ処理方式を提案した。今後、関連研究と本提案方式を詳細に比較すると共に、方式の詳細化、有効性の評価を進めていく。

参考文献

- 1) 藤田隆史, 後藤良則, 小池新: M2Mアーキテクチャと技術課題, 電子情報通信学会誌, Vol.96, No.5, pp. 305-312 (2013).
- 2) 安本慶一, 山口弘純: 多数のデータストリームを実時間で融合・編纂し利活用するための次世代「情報流」技術-情報流キューレーション基盤実現に向けた課題抽出と取り組み-, 情報処理, Vol.55, No.11, pp. 1281-1287 (2014).
- 3) 石川冬樹: クラウドとIoTの結合によるスマートシティプロジェクトClouTの紹介, http://www.jpgrid.org/ws40-ishikawa_r.pdf.
- 4) W3C Semantic Sensor Network Incubator Group, <http://www.w3.org/2005/Incubator/ssn/>.
- 5) oneM2M, <http://www.onem2m.org/>.
- 6) Craig D Weissman, Steve Bobrowski: The Design of the Force.com Multitenant Internet Application Development Platform, Proceedings of the 2009 ACM SIGMOD International Conference (SIGMOD'09), pp. 889-896, 2009.
- 7) AWS Cloud Design Pattern, <http://aws.clouddesignpattern.org/>.
- 8) 日本の統計, 総務省統計局, <http://www.stat.go.jp/data/nihon/index1.htm>.
- 9) ZigBee, <http://zigbee.org/>.
- 10) 平成25年全国都道府県市区町村別面積調査, <http://www.gsi.go.jp/KOKUJYOHO/MENCHO/201310/opening.htm>.
- 11) Peter Desnoyers, Deepak Ganesan, Huan Li, Ming Li, Prashant Shenoy: PRESTO: A Predictive Storage Architecture for Sensor Networks, Proceedings of the 10th Workshop on Hot Topics in Operating Systems (HOTOS'05), 2005.
- 12) Yanif Ahmad, Suman Nath: COLR-Tree: Communication-Efficient Spatio-Temporal Indexing for a Sensor Data Web Portal, Proceedings of the 24th International Conference on Data Engineering (ICDE'08), pp. 784-793, 2008.
- 13) 幸島明男, 池田剛, 井上豊, 車谷浩一: センサイイベント指向のサービス連携ミドルウェア: SENSORID, 情報処理学会研究報告, 2006-UBI-12, pp. 37-44, 2006.
- 14) 宵憲治, 山口弘純, 廣森聡仁, 内山彰, 東野輝夫, 柳谷尚寿, 中谷俊和, 立花篤男, 長谷川輝之: 多数のセンサーによる時空間センシングデータの効率的なセンシング技術, マルチメディア通信, 分散, 協調とモバイルシンポジウム (DICOM02014), pp. 903-913, 2014.