

# 大規模 IoT サービスの実証を可能とするオープンテストベッド JOSE

寺西裕一 齋藤祐貴 室野 栄 西永 望

我々は、IoT(Internet of Things) サービスに必要な機能要素を、ネットワーク上の仮想化されたサービスとして提供するテストベッド JOSE を開発した。JOSE は誰もが利用可能なオープンテストベッドであり、センサーネットワーク、ストレージ・計算リソースを多数用いた大規模実証を複数収容可能なサービス基盤である。本稿では JOSE の実現技術及びテストベッドとしての運用状況について述べる。

## 1 まえがき

スマートフォンやコンピュータのような専用端末のみならず、人々の身の回りに存在するモノ(センサー、携帯端末、家電、ロボット、クルマ等)をネットワーク接続させることで、人の介入を極力抑え、安全で効率化された ICT サービスをあらゆる場所で提供可能とする Internet of Things(IoT) サービスと呼ばれる新世代のネットワークサービスの実現に対する期待は大きい。ネットワーク接続された機器、センサー等による観測データは、定期的、あるいは間欠的に、連続して生成されるデータである。こうしたデータは、1つひとつのデータサイズは大きくなくともその数は大量であり、蓄積されるデータ量も膨大となる、いわゆるビッグデータである。そのようにして生成されるデータをネットワークサービスとして取得・伝送・蓄積・分析することで、迅速な災害対策や、地域産業の活性化等に利活用可能とすることが期待されている。

IoT サービスでは、上記データの取得・伝送・蓄積・分析といった一連のプロセスにおいて要求されるセンサーネットワーク、大規模なストレージリソース・計算リソース、それらを接続するネットワークといった機能要素を、必要な機密性を満たしつつ一体のシステムとして構成可能とする必要がある。我々は、こうした IoT サービスに必要な機能要素を、仮想化されたネットワーク上のサービスとして提供する基盤(IoT サービス基盤)として JOSE (Japan-wide Orchestrated Smart/Sensor Environment) を開発した<sup>1)</sup>。JOSE は、いわば IoT サービスに特化された IaaS(Infrastructure as a Service) であり、物理的な計算機やネットワーク設備の投資を抑えて必要な機能要素を持つ IoT サービスを構成することができる。

JOSE では、仮想マシン技術や仮想ネットワーク技術といった仮想化技術を用いたストレージリソース・

計算リソースを設定する手順を自動化・並列化するとともに、センサーデータを流通させるネットワークを P2P ネットワークにより自律分散的に構成することで、大規模なリソース管理にかかる手間と時間を低減させることが可能となっている。

本稿では、JOSE の概要、及び、JOSE のテストベッドとしての運用状況について述べる。

## 2 JOSE が提供する IoT サービス基盤

### 2.1 IoT サービスの構成要素

図1は、JOSE が扱う IoT サービスの機能要素を示している。JOSE が収容する複数の IoT サービスそれぞれに対して、各機能要素が、共通のハードウェア環境のもと、仮想化されて提供される。以下、それぞれの機能要素について述べる。

#### 1) センサーネットワーク

IoT サービスにおける特徴的な機能要素となるのが、センサーネットワーク(またはセンサー・アクチュエー

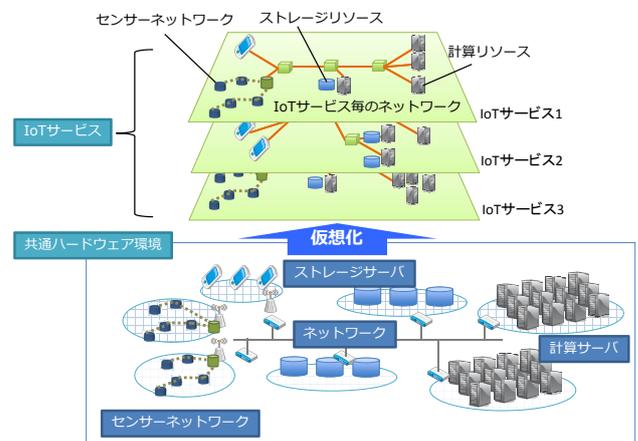


図1 IoT サービスの構成要素

タネットワーク。以下、センサーネットワーク)である。IoT サービスにおいては、観測の対象となる領域に、センサーネットワークが配備される。また、処理結果を通知、あるいは、動作を変更する等の命令を送信する対象となるアクチュエータ(端末・PC・デバイスなど)が接続されることもある。センサーや端末等の機器は、単体でインターネットプロトコルのような高度な通信機能を具備し、ネットワーク上のサーバ等に接続可能な場合もあれば、単体では近距離の、比較的単純なネットワークプロトコルを具備するのみとし、機器同士が相互接続したセンサーネットワークを構成して、ネットワーク上のサーバへの通信は、ゲートウェイと呼ばれる比較的高度な通信機能を具備する装置が中継する形態をとる場合もある。JOSEにおけるセンサーネットワークは次の2つの形態によって実現される。

- IoT サービスが、センサーネットワークを新たに配備する
- 他のIoT サービスと提携し、配備済みのセンサーネットワークを利用する

### 2) ストレージリソース

それぞれのIoT サービスは、専用のストレージリソースを持つ。ストレージリソースは、センサーから得られる観測データ等を蓄積する構成要素である。JOSEにおけるストレージリソースは、大容量のストレージを持つストレージサーバ上の仮想マシンによって提供される。

連続して生成される観測データを継続的に長期にわたって蓄積していくと、個々のデータサイズは小さくとも、膨大な容量が必要となる。また、観測データを解析処理や学習データの対象として利用可能とするには、検索可能な状態で蓄積しておく必要がある。検索を高速に行うためには、各データにインデックスなどが付与された状態でデータベース化して蓄積する必要がある。

また、IoT サービス間の提携により、ストレージリソース間の連携が行われる場合もある。この場合、ストレージリソースに対するアクセス権限を他のIoT サービスに与えてデータを提供する、あるいは、他のIoT サービスのストレージリソースへのアクセス権限を得てデータを取得するといったデータ共有が行われる。

### 3) 計算リソース

計算リソースは、センサーから得られる観測データの処理等を行うための構成要素であり、JOSEにおいては計算サーバ上の仮想マシンによって提供される。シミュレーションによる予測等、複雑な処理が行われる場合、計算量が大きくなるため、処理に必要な時間

内に終わらせるために必要な計算リソースを確保しなければならない。ビッグデータの解析処理においては、大量のデータを対象とした処理の結果を短時間で得るため、計算を実行する計算リソースの数を増やすことで性能を向上させるスケールアウト(scale out)と呼ばれる方策がとられることが多い。スケールアウト可能なアプリケーションでは、計算機の数が多いほど処理にかかる時間は短縮されるため、要求される処理完了時間と処理対象となるデータの規模に応じた数の計算リソースが利用される。

IoT サービスにおいては、必要な計算リソースが実世界で生じている事象(イベント)に応じて変動する場合も想定する必要がある。例えば、降雨があった場合に、シミュレーションによって降雨状態がどう変動するかを予測するアプリケーションでは、降雨が無い状態では何も処理が実行されず、降雨という事象の発生に応じて必要な計算リソース量が変化する。

### 4) IoT サービスごとのネットワーク

上記でも述べた通り、IoT サービスの機能要素の構成は短時間で変更可能であることが求められる。要求が頻繁に変化し得ることを考えると、IoT サービスの提供者自身が構成変更を自由に実行できることが望ましい。一方、IoT サービスでは、サービスごとに機密性を保つことに対する要求が大きい。センサー観測データは、オープンなデータとして利用可能なものばかりではなく、例えば、人の位置情報のようにパーソナルデータを対象とする場合、機密が保たれた、個別のプライバシーポリシーのもと運用を行う必要がある。気温や雨量等の環境に関する観測データであっても、観光に影響を及ぼす風評被害等を避けるためにオープンデータとすることに対して慎重となる場合もある。こうした観測データを扱う場合、盗聴等の攻撃の懸念を避けるため、データが流れるネットワークは、他のサービスとネットワークを共有しないというポリシーを設定可能としたい場合がある。また、悪意がなくとも、不具合や誤設定による想定外のデータトラヒックの発生やネットワークの誤動作などの影響が生じることは想定すべきである。JOSEはテストベッドであり、発展途上の技術の検証も行われることから、そのような事態に対する対策は重要となる。

そこで、JOSEでは、各々のIoT サービスは独立した仮想ネットワーク上に構成されるものとする。各仮想ネットワークとしては、下記を実現する。

- データトラヒック自体が、IoT サービスごとに分離される
- IoT サービスの構成変更をIoT サービスごとに独立して行える

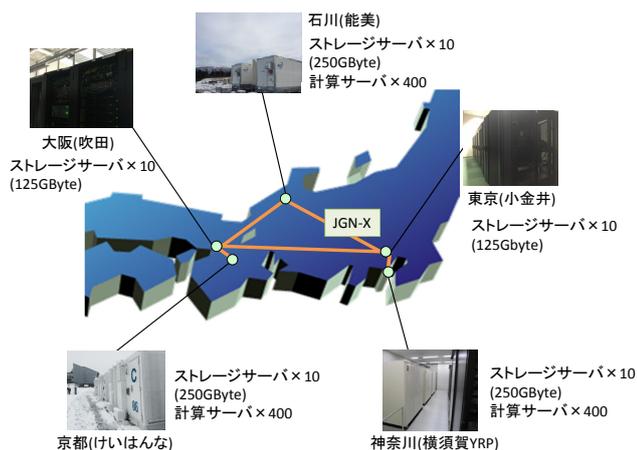


図2 JOSEのハードウェア環境

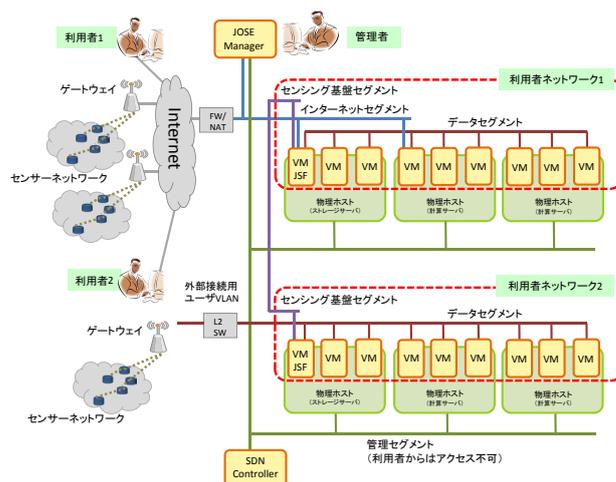


図3 JOSEが提供するIoTサービスの構成例

## 2.2 JOSEのハードウェア環境

JOSEは、前節で示した構成要素の組み合わせによるIoTサービスを同時に多数収容するIoTサービス基盤の実現を目指している。そのためのハードウェア設備として、5つの分散拠点（東京、大阪、神奈川、石川、京都）にあるデータセンターに1,250台の物理サーバを配備している。それぞれの物理サーバには、最大16個の仮想マシンを動作させることが可能であり、最大20,000個の仮想サーバを、計算リソースとして利用可能である。また、ストレージは全データセンター合計で1PByteの容量を有している。

データセンター間は、JGN-X<sup>[2]</sup>による広域レイヤ2ネットワークで接続されている。また、ネットワーク設備はハードウェアとしてSDN(Software Defined Networking)の機能を具備し、データセンター間は10Gbpsの共有ネットワーク、各物理サーバ向けには1Gbpsの占有ネットワークの容量を持つ。1台のSDNコントローラに接続可能な仮想サーバ数には上限があるため、SDNのドメインは3つに分かれている。

JOSEは、このハードウェア環境のもと、IoTサービスに必要な機能要素を仮想化して提供するIoTサービス基盤を提供する。

## 3 JOSEの実現技術

本節では、2.2で示したIoTサービス基盤としてのJOSEの実現技術について述べる。

### 3.1 JOSEが提供するIoTサービスの基本構成

図3はJOSEが提供するIoTサービスの構成例を示している。

JOSEにおける「利用者」は、IoTサービスを提供するプロバイダ(サービス提供者：テストベッドとして

はIoTサービスの実験者)である。JOSEの共通のハードウェア設備上に、IoTサービスごとの仮想ネットワーク(利用者ネットワークと呼ぶ)が構成される。

また、JOSEには、管理者が存在する。管理者は、利用者の申請を受けて利用者登録、削除、IoTサービスの作成、構成変更、監視などを行う。

JOSEの利用者は、与えられた利用者ネットワーク上のストレージリソース・計算リソースを専有し、自由にカスタマイズ可能である。また、JOSE利用者は、必要に応じてIoTサービスの構成を変更可能である。JOSEでは、利用者ネットワークに対するカスタマイズ要求レベルに応じて、以下の3種類のクラスを設けている。

- 1) Static クラス
- 2) Dynamic クラス
- 3) RISE クラス

Staticクラスは、利用者ネットワークの構成変更を、管理者の承認を経て、管理者の権限で行うクラスである。利用者によるIoTサービスの構成変更に関する要求は、即座に反映されるわけではなく、管理者による承認後に設定される。

Dynamicクラスは、利用者ネットワークの構成変更を、管理者の承認を経ることなく、利用者の権限で行うクラスである。利用者によるIoTサービスの構成変更に関する要求は、即座に反映される。

RISEクラスは、利用者がより自由に利用者ネットワーク制御を行えるよう、SDNコントローラそのものをカスタマイズし、利用者の権限で動作させる構成も可能とするものである。SDNの仮想化方法の詳細については、RISEの文献<sup>[3]</sup>を参照されたい。

図3には、2つの利用者ネットワークが存在し、それぞれStaticクラスで接続されている。各利用者は、JOSE Manager(3.2参照)にインターネット経由でロ

グインして、割り当てられた仮想マシンの起動・終了などの管理を行う。

利用者ネットワーク1では、仮想マシン (VM) がストレージサーバと計算サーバ上にそれぞれ生成されており、センサーネットワークはインターネット経由で接続されている。ストレージリソースや計算リソース間のデータ交換はデータセグメント上で行われる。また、利用者ネットワーク1では、ストレージサーバ上と計算サーバ上の仮想マシンがインターネットセグメント (NAT セグメント) に接続し、インターネットアクセス可能となっている。ストレージサーバ上の仮想マシンの1つでは、JOSE センシング基盤 (JSF: 3.3 参照) が動作しており、連携 IoT サービスとの間のセンサー情報の共有を、センシング基盤セグメントを介して行う構成となっている。

利用者ネットワーク2は、利用者がインターネット経由でセンサーネットワークや仮想マシン管理端末を接続せず、JGN-X の広域 L2 サービス等を利用し、外部接続用ユーザ VLAN によって直接データセグメントに接続する例である。このため、インターネットセグメントは、利用者ネットワーク2には接続されていない。

管理セグメントは、3.2 で述べる JOSE Manager が構成変更等を行う管理用のネットワークである。いずれの構成においても、管理セグメントは物理ホストにのみ接続しており、仮想マシンとは直接接続しない。この構成により、IoT サービスの独立性、機密性を維持する。

JOSE Manager は、仮想マシン、仮想ネットワークの設定を管理セグメント経由で行う。JOSE Manager からの仮想マシンの設定、状態監視等は、物理ホストを介して、仮想ネットワークの設定、状態監視等は、SDN コントローラを介して行われる。

Dynamic や RISE クラスも同様の構成は可能である。これらのクラスは、物理的に分離されたネットワーク接続がなされたハードウェア上に、Static クラスとは別の仮想マシン、仮想ネットワークを構成する。Dynamic クラスでは、JOSE Manager を管理者が操作することはなく、利用者が利用者の権限で管理運用を行う。RISE クラスでは、仮想ネットワーク設定を、利用者が自らカスタマイズした専用の SDN コントローラを介して行う。

### 3.2 JOSE Manager

JOSE 利用者は、利用者ネットワークに関わる次に挙げる要求を指定可能である。

#### 1) 仮想マシンに関する要求

- (ア) 仮想マシン数

- (イ) テンプレートイメージ (後述)

- (ウ) 配備場所

- (エ) メモリ量

- (オ) CPU 数

- (カ) ストレージ容量

#### 2) 仮想ネットワークに対する要求

- (ア) データネットワーク設定 (L3)

- (イ) 外部接続用インターネット (L3)

- (ウ) 外部接続用ユーザ VLAN ネットワーク (L2)

- (エ) センシング基盤ネットワーク (L2/L3)

これらの利用者の要求を受け付け、物理ホストや SDN へ反映させる、あるいは、利用状況を収集し、可視化するといった、JOSE における管理運用を担う機構を JOSE Manager と呼んでいる。仮想ネットワークの設定のうち、L3 と記載されているネットワークは、レイヤ3 ネットワークであり、IP アドレスの割当てまで JOSE Manager 経由でなされる。L2 と記載されているネットワークは、レイヤ2 ネットワークによる接続であり、IP アドレスの管理は利用者自身が行う必要がある。

JOSE 管理者は、利用者要求を受け、JOSE への反映及び利用者による利用状況の監視を JOSE Manager を介して行う。JOSE の利用者へは JOSE Manager のアカウントを割り当てられる。JOSE Manager はウェブインタフェースを持ち、インターネット経由でログインすると自身に割り当てられた仮想マシンの一覧を閲覧できる。また、起動仮想マシンの追加・削除等が可能である。

#### 3.2.1 仮想マシン・ストレージの管理

JOSE Manager では、基本機能を具備する仮想マシンイメージ (Linux OS カーネル + user land) を「テンプレートイメージ」として用意している。テンプレートイメージは、複数の候補のなかから、利用者要求に応じて選択することができる。本稿執筆時点では、Linux の Ubuntu ディストリビューションをベースとしたテンプレートイメージが存在する。

JOSE Manager は、テンプレートイメージから、利用者に割り当てられた物理ホスト上で動作させる仮想マシンイメージ (利用者イメージ) を生成する。利用者イメージは、初回仮想マシン起動時にテンプレートイメージから作成される。これにより、新規利用者は常に最新の状態の OS から利用開始することが可能である。

JOSE Manager からの利用者要求に基づく仮想マシン設定の反映は、物理ホストに対して、一括、かつ、並列で行われるため、逐一仮想マシンに対して設定を行う手間は省かれる。

利用者用イメージから生成される各仮想マシンは、

JOSE で動作するために必要な以下の機能を持つ。

(1) ネットワーク制御

SDN 設定と同期したネットワーク設定を行う。

(2) ストレージ制御

仮想マシン上のストレージ設定を行う。

(3) 状態通知

仮想マシン起動時に常駐プロセスとして起動される。仮想マシンの起動・終了状態や、使用状況を JOSE Manager に通知する。

上記 (1) によって、仮想マシンにおけるネットワーク設定は、JOSE Manager からの指示にしたがったものとなる。原則として、JOSE では各利用者に仮想マシン上のネットワーク設定を直接変更させない。

(2) により、利用者要求に応じたストレージ容量を、仮想マシンごとに確保する。JOSE Manager 上で指定されたストレージ容量に応じた「ストレージディスクイメージ」が、物理ホスト上に生成され、(2) の機能によって仮想マシンからマウントされる。JOSE のストレージサーバが物理ホストの場合、大容量のストレージディスクを指定できる。

(3) のプロセスにより、物理ホスト及び仮想マシンの生存情報、利用状況情報を、JOSE Manager 上で得ることを可能としている。マシンの状態が一定の時間間隔で JOSE Manager の状態管理プロセスに通知される。JOSE では、仮想マシン数が万単位となるため、各マシンが実用的な時間間隔で状態通知を送出するとアクセス数が膨大となり、負荷が高くなってしまう。このため、一定数のマシンごとに中継を行う物理ホスト (リレー) を設けて、木構造の収集用トポロジを形成し、リレーにおいて状態情報をまとめたのち、木構造上の上位へ通知し、最終的に木構造の根となる JOSE Manager に収集する機能を持たせた。これによ

り、JOSE Manager において受け付けねばならない一定時間あたりのアクセス数が軽減される。

### 3.2.2 仮想ネットワークの管理

JOSE Manager は複数データセンターにまたがる万単位の仮想マシンを扱うネットワークを構成するため、以下の機能を持つ。

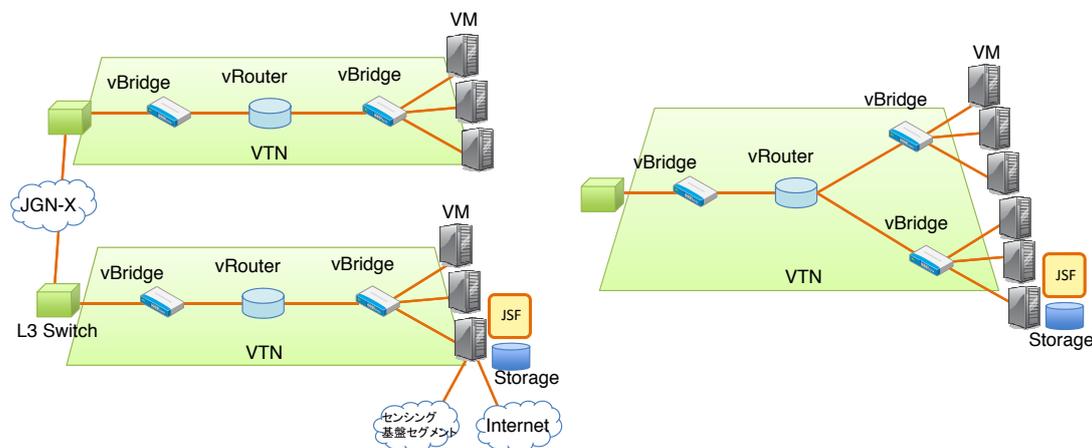
1) マルチドメイン SDN 制御

JOSE は、3つの SDN ドメインを持つため、JOSE Manager からは、各 SDN コントローラの API を並列利用する。JOSE における Static クラス、Dynamic クラスでは、利用者ネットワークごとの仮想ネットワークを、VTN (Virtual Tennant Network) と呼ばれるインタフェース<sup>[4]</sup>を用いて構成する。VTN API により、仮想化されたルータ (仮想ルータ)、仮想化されたブリッジ (仮想ブリッジ) 等を用いた仮想ネットワークを構成可能である。利用者からの計算リソース、ストレージリソースの配備場所の要求が、JOSE における複数の SDN ドメインにまたがっている場合、同時に複数の SDN コントローラに対して VTN 構成命令を送信する。

2) 仮想ネットワーク分割

図 4 は、リソース間を接続するデータネットワークの構成例を示している。各データネットワークは、仮想レイヤ 2 ネットワークを構成する仮想ブリッジ (vBridge)、仮想レイヤ 3 ネットワークを構成する仮想ルータ (vRouter) によって構成される。複数の SDN にまたがった利用者ネットワークでは、上位の物理レイヤ 3 スイッチが VTN 間のルーティングを行う (図 4 (A))。

通常、1つの L2 ネットワークに収容可能なエンドホスト数は数百程度が上限である。このため、JOSE Manager は、1つの利用者ネットワーク内に収容すべ



(A) 複数 SDN ドメインにまたがる利用者ネットワーク

(B) 仮想レイヤ 2 ネットワークの分割

図 4 利用者ネットワークの構成例

き仮想マシン数が一定以上となる場合は、仮想的なレイヤ2ネットワークを複数に分割し、それらの間でレイヤ3ルーティングを行う設定を行う(図4(B))。それぞれの仮想マシンや、vRouter に対する IP アドレスの設定は JOSE Manager によって自動的に行われる。

外部接続用インターネット (L3)、外部接続用ユーザ VLAN ネットワーク (L2)、センシング基盤ネットワーク (L3) は、それぞれ1つのVTNに対応する。各仮想マシンには、接続されたネットワークごとにインタフェースが生成される。レイヤ3ネットワークでは、インタフェースにIPアドレスが自動的に割り当てられる。

### 3.3 JOSE センシング基盤 (JSF: JOSE Sensing Foundation)

JOSE においては、既に述べた通り、センサーネットワークの利用方法として次の2つの形態がある。

- IoT サービスが、センサーネットワークを新たに配備する
- 他のIoTサービスと提携し、配備済みのセンサーネットワークを利用する

これらの利用形態でのセンサーデータアクセスのため、我々は新たなセンサーデータ管理機構 JOSE センシング基盤 (JOSE Sensing Foundation、以下、JSF) を開発した。JSF は、各IoTサービスにおけるストレージリソースとともに配備され、計算リソースからのセンサーネットワークへのアクセスを担う機能要素となる。

センサーネットワークのゲートウェイ及びJSFそれぞれにおいて、IEEE1888<sup>⑤</sup>を、JOSEのストレージの間の共通アクセス手段として採用した。IEEE1888では、観測データの名前は任意に決められるが、JOSEにおいては、利用者間でデータ共有を行えるよう独自のIEEE1888の規約(語彙)を定義している。利用者が構築したセンサーネットワークであっても、JOSEのIEEE1888に準拠していればJOSEのセンサーネットワークとして接続可能である。

IoTサービスの計算リソースは、IEEE1888を用いてJSFにアクセスすることで、過去の蓄積された観測データや最新の観測データを取得できる。また、Web Socketにより、センサーネットワークから観測データが提供された時に、イベントドリブンでリアルタイムにデータを取得できるAPIも持つ。さらに、既に配備済みの、他のIoTサービスのセンサーネットワークが収容するセンサーのうち、アクセス権があるセンサーからのセンサーデータを、検索によって得ることもできる。各IoTサービスのJSF間は、セン

シング基盤セグメントで接続され、連携動作する広域ネットワークが形成される(図5)。

利用者が設置したセンサーネットワークは、利用者が所有し、利用者の責任の範囲で運用するものである。安定したJSFによる連携動作を実現するには、利用者間の協力が必須となるが、利用者によって管理が十分に行われない場合があることを想定しなければならない。そこで、JOSEでは、JSF間の接続をPeer-to-Peer(P2P)ネットワークにより自律的に行うことで、新たなJSFの追加や削除があった場合に利用者が登録作業等を行うことなく、自動的に相互接続がなされ、検索可能となる機能を提供している。このJSFの機能はミドルウェアPIAX<sup>⑥</sup>を用いて実装した。JOSEのIEEE1888規約では、センサーの位置情報は必須項目となっており、位置座標をキーとしてセンサーを検索可能である。アプリケーションからは、必要な観測データの領域を指定してJSF上で検索とアクセスを行えば、当該領域を観測するセンサーネットワークを管理する他の利用者の手を煩わせることなく、最新の観測データへのアクセスが可能となる。

図6は、JSFにおけるセンサー検索の例である。この例では、東経130度、北緯30度から、それぞれ10度以内の矩形領域に位置するセンサーから温度の観測データを取得する例である。検索結果は、JSON また

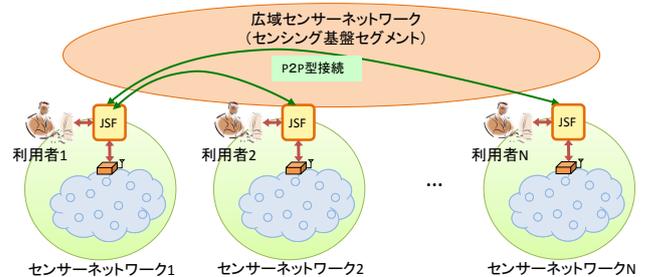


図5 JSF間のP2P接続

#### Request URL:

http://host:port/sensors/Location in rect(130,30,10,10)/values/Temperature



#### Response:

```
[{"id": "http://jose.jp/1888/00/0x0001",
  "values": [{"id": "http://jose.jp/1888/00/0x0001/Temperature",
    "value": [{"value": 27.5, "time": 1401334500000}],
    "profile": "weather"}]},
 {"id": "http://jose.jp/1888/00/0x0002",
  "values": [{"id": "http://jose.jp/1888/00/0x0002/Temperature",
    "value": [{"value": 29.5, "time": 1401334500000}],
    "profile": "weather"}]},
 {"id": "http://jose.jp/1888/00/0x0003",
  "values": [{"id": "http://jose.jp/1888/00/0x0003/Temperature",
    "value": [{"value": 29.0, "time": 1401334500000}],
    "profile": "weather"}]}]
```

図6 JSFにおけるセンサー検索の例

は XML 形式で提供される（図は JSON 形式の例）。

## 4 JOSE テストベッドとその運用

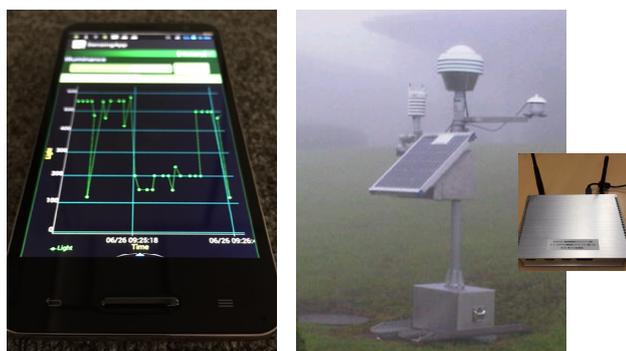
JOSE は、テストベッドとして、本稿執筆時点で、27 の IoT サービスに関連した実証実験を収容している。以下では JOSE のテストベッドとしての運用状況について述べる。

### 4.1 センサー機器例

JOSE の IEEE1888 に準拠したセンサー設備を幾つか開発し、テストベッド利用者に貸し出しを行っている。

図 7 (A) は、我々が開発したスマートフォンセンサーである。形状は通常のスマートフォンと同等であるが、位置情報、加速度の検出機能に加え、地磁気、照度、温度、湿度、気圧を観測するセンサーを内蔵している。また観測されたデータを、無線 LAN、3G/LTE のいずれかを經由して、IEEE1888 によって JSF へ転送するソフトウェア機能を組み込んでいる。

また、図 7 (B) は、太陽光パネルで自家発電し、自律動作する無線環境センサー及びそのゲートウェイ装置である。日射計、赤外放射計、騒音計、振動、積雪、CO<sub>2</sub>、風速風向、温度、湿度、気圧、降水強度等を観測するセンサーが利用可能であり、いずれのセンサーを組み込むかをカスタマイズすることができる。各無線環境センサーは、ゲートウェイ装置(図 7 (B) 右)に無線接続し、観測データを送信する。複数無線環境センサーが設置された場合、無線環境センサー同士がアドホックネットワークを形成し、マルチホップ通信で観測データを送信する。ゲートウェイ装置は、有線/無線 LAN、3G/LTE のいずれかの通信手段で、IEEE1888 によって観測データを JSF へ転送する。



(A) スマートフォンセンサー

(B) 無線環境センサーとゲートウェイ装置(右)

図 7 センサー機器の例

### 4.2 センサーネットワークの例

JOSE では、既にいくつかの場所に、センサーネットワークを設置し、運用している。以下は、センサーネットワークの例である。これらのセンサーを活用した実証実験がいくつか実施されている。

小金井市：環境センサーネットワーク

(環境センサー)

福島市：建物・橋梁センサーネットワーク

(振動センサー)

千曲市：河川センサーネットワーク

(水位・雨量センサー)

広島市：環境センサーネットワーク

(環境センサー)

東京都内：橋梁センサーネットワーク

(振動センサー)

JOSE 利用者は JSF を介してセンサーネットワークへアクセス可能である(ただし、それぞれ異なるアクセス権限が設定されている)。

### 4.3 JOSE の性能

#### 4.3.1 運用オペレーション時間

図 8 は、JOSE において、異なる拠点に 200 の仮想マシンを配備した IoT サービスを構成した場合にかかるオペレーション時間(設定開始から提供開始まで)を示している。図において、「No Manager」は、JOSE Manager を使用せずに利用者ネットワークを構築した場合のオペレーション時間、「JOSE Manager」は、JOSE Manager を用いた場合のオペレーション時間である。JOSE Manager を使用せずに構築する場合、200 の仮想マシンそれぞれに対して、アカウント設定 (user setup)、仮想マシン自体の起動設定 (VM setup)、ストレージの設定 (storage setup)、ネットワークインタフェースへのアドレス設定 (VM network setup) を行う必要がある。また、SDN に対するコマンドを入力する (SDN setup: この場合、SDN 製品の GUI インタフェースを使用)。図 8 は、仮想マシンの VM イメージをすべての物理ホス

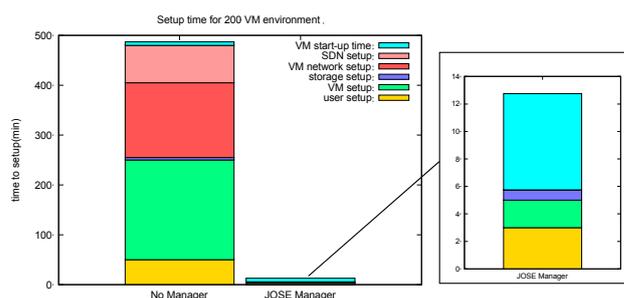


図 8 200 仮想マシンを有する利用者ネットワークの設定時間

トへ転送し、OSとしてブートさせ、稼働状態となるまでの時間 (VM start up time) も含んでいる。仮想マシンへの設定は、スクリプト処理等によりある程度自動化したものの、200の仮想マシンの設定完了までに487分の時間を要した。

一方、JOSE Managerを用いた場合、仮想マシンに対するアドレス設定やSDNのコマンド投入は、自動的に行われるため、ほぼ作業時間は0となっている。また、アカウント設定や仮想マシン設定は、一括して並列で行われるため、大幅に少ない時間での設定完了が可能となった。VM start up timeについては、JOSE Managerを使用しない場合と同等の待ち時間が生じており、これが全体のオペレーション時間の半分以上を占めている。すべての仮想マシンが稼働状態となるまでにかかった時間は、およそ13分であり、JOSE Managerを使用しない場合の487分から、作業時間はおよそ97.4%削減された。

4.3.2 JSFの検索応答時間

図9は、JSFにおける検索完了時間の計測結果である。横軸は、位置情報を持つセンサーまたはセンサーネットワークの数を示している。最大1,000の位置情報を持つ機器やゲートウェイ (node: ノード) が存在する状態での平均遅延を計測した。それぞれのセンサーは、JOSE上の3つのSDNドメインに均等に分散して配備された状態とした。また、検索条件は同一で、検索にマッチする数を10及び100となるよう変化させた。それぞれ、10回の検索試行の平均時間を示している。

図から、まず、検索マッチ数に対する規模拡張性が確認できる。マッチ数が10の場合、及び、100の場合で、検索完了までにかかる時間にほとんど差はなく、ノードの数が1,000の場合いずれも1秒以内となった。

また、ノード数の増加に対する規模拡張性が確認

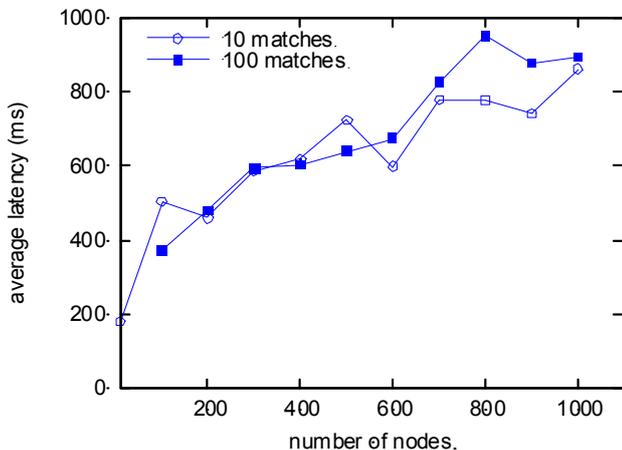


図9 JSFにおける検索応答時間

できる。ノード数が10倍となっても、遅延時間の増大は2倍程度に抑えられていることが分かる。JOSEが採用しているPIAXのオーバーレイネットワークはSkip Graph<sup>6)</sup>に基づく実装である。Skip Graphでは、ノードの数Nに対して平均O(log N)の応答時間を実現できることが知られており、本検証でもその性質を確認できた。

5 あとがき

本稿では、我々が開発したIoTサービス基盤JOSEとそのテストベッドとしての運用状況について述べた。

IoTサービスの要素技術には、研究途上にあり、実用化に至っていないものが多数ある。国内においても、それぞれの分野で様々な研究開発が行われているが、要素技術を実用化していく上では、センサー、計算機、ストレージ、ネットワークを実際に用いた実証や動作検証が必須となる。このようなアプリケーションの設計や評価をサポートできる大規模なテストベッド環境はこれまでに提供されておらず、JOSEが国内外でも初のテストベッドである。既にフィールド実証や動作検証が様々な研究開発プロジェクトによってJOSE上で行われており、また、フィールド実証を通じ、横断的なデータ利活用等も進められつつある。こうした活動を通じて、IoTやビッグデータに関わる研究開発の効率化、ひいては、国内国外でのこの研究分野の活性化を目指している。

【参考文献】

- 1 “大規模オープンテストベッド JOSE,” <http://www.nict.go.jp/nrh/nwgn/jose.html>, June 2015.
- 2 “Network Testbed JGN-X,” <http://jgn.nict.go.jp/>, June 2015.
- 3 Y. Kanaumi, S. Saito, E. Kawai, S. Ishii, K. Kobayashi, and S. Shimojo, “RISE: A wide-area hybrid OpenFlow network testbed,” *IEICE transactions on communications*, Vol.96, No.1, pp.108-118, Jan. 2013.
- 4 “OpenDaylight Virtual Tenant Network (VTN),” [https://wiki.opendaylight.org/view/OpenDaylight\\_Virtual\\_Tenant\\_Network\\_\(VTN\):\\_Main](https://wiki.opendaylight.org/view/OpenDaylight_Virtual_Tenant_Network_(VTN):_Main), June 2015.
- 5 IEEE1888-2011: standard for ubiquitous green community control network, 2011.
- 6 Yuuichi Teranishi, “PIAX: Toward a Framework for Sensor Overlay Network,” *In Proc. of CCNC 2009*, pp.1-5, Jan. 2009.
- 7 J. Aspnes and G. Shah, “Skip Graphs,” *ACM Transactions on Algorithms*, Vol.3, No.4, Article No.37, Nov. 2007.



寺西裕一 (てらにし ゆういち)

ネットワーク研究本部ネットワークシステム  
総合研究室研究マネージャー  
博士(工学)  
ユビキタスコンピューティング、オーバーレイ  
ネットワーク、マルチメディア、データベース  
モバイル



**齋藤祐貴** (さいとう ゆうき)

ネットワーク研究本部ネットワークシステム  
総合研究室技術員  
ネットワーク



**室野 栄** (むろの さかえ)

ネットワーク研究本部ネットワークシステム  
総合研究室専門研究員  
ネットワーク



**西永 望** (にしなが のぞむ)

ネットワーク研究本部ネットワークシステム  
総合研究室室長  
博士(工学)  
新世代ネットワーク