スケーラブルなエッジ指向 IoT アーキテクチャの提案

濱野 真伍[†1] 青山 幹雄[†1]

概要:エッジコンピューティングと Publish/Subscribe アーキテクチャを統合し，階層化とスケールアウトによって高いスケーラビリティを実現するアーキテクチャと，エッジにおけるメッセージ配信モデルを定義する．

キーワード:IoT，エッジコンピューティング，Publish/Subscribe アーキテクチャ，ソフトウェアアーキテクチャ，スケーラビリティ

A Scalable Edge-Oriented IoT Architecture

SHINGO HAMANO[†1] MIKIO AOYAMA[†2]

1. はじめに

IoT(Internet of Things)[1]システムでは，大量かつ多種多様なデバイスがインターネットに接続される．そのため，IoT アーキテクチャには高いスケーラビリティが要求される．そこで，エッジコンピューティング(以下エッジ)や Publish/Subscribe(以下 Pub/Sub)アーキテクチャに基づく MQTT，oneM2M の標準アーキテクチャが提案されている．本稿では，複数の車両の位置情報や運行状況の収集を行う車載システムを対象とし，エッジ上で Pub/Sub を統合した階層化による機能分散とスケールアウトを可能にする IoT アーキテクチャを提案する．

2. 研究課題

本研究では，エッジと Pub/Sub を統合し，接続されるデバイス数や処理するメッセージ数に応じてスケーラブルに拡張可能な IoT アーキテクチャの提案を課題とする．

3. 関連研究

3.1 エッジコンピューティング

エッジ[4]はクラウドをネットワークのエッジへ拡張し，フィルタリングなどのデータ処理を行うアーキテクチャである．

3.2 Publish/Subscribe アーキテクチャ

ブローカを介して非同期でメッセージを配信するアーキテクチャである[5]．IoT 向けの実装にトピックベースでメッセージを配信する MQTT[2]がある．

3.3 oneM2M Reference Architecture

oneM2M 参照アーキテクチャ[3]はインフラストラクチャノード，ミドルノード，デバイスを含むアプリケーションサービスノードやアプリケーションデバイスノードからなる 3 層構造をとる．

4. アプローチ

4.1 前提条件

アーキテクチャの提案にあたり以下の前提条件をおく．

(1) クラウド，エッジ，デバイスの 3 層アーキテクチャをとる．

(2) エッジは移動しないものとする．

(3) デバイスからクラウドへのメッセージ配信のみを対象とする．

4.2 アプローチ(アーキテクチャコンセプト)

図 1 にアプローチを示す．エッジの機能にはデバイスからのメッセージを受信する受信層と，クラウドへのメッセージを配信する配信層がある．配信層ではメッセージごとに配信先を決定するため高負荷となる．そこで，本研究では以下の 3 点に着目しスケーラブルな IoT アーキテクチャを提案する．

(1) 階層化: エッジを受信層と配信層の 2 層に分離する．受信層，配信層にそれぞれブローカを導入し，メッセージのフィルタリングを行う．また，受信層と配信層間のメッセージ配信モデルを定義する．

(2) エッジ内スケールアウト: 階層化によって 2 層に分離したエッジの配信層をスケールアウトする．これにより，1 配信層内での負荷を分散する．

(3) エッジ間スケールアウト: 受信層でメッセージのフィリタリングを行い，配信層に転送しないメッセージを他のエッジの受信層に転送する．これにより，配信層は複数の受信層からメッセージを受信する必要がない．

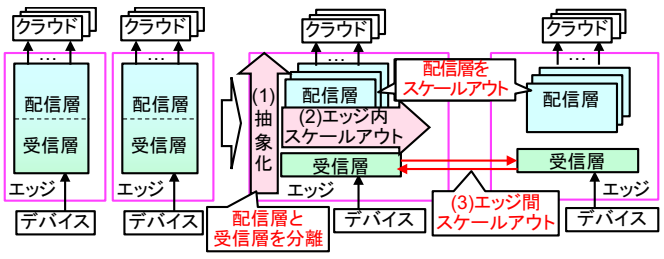

図 1 アプローチ

5. 提案アーキテクチャ

5.1 アーキテクチャの構造

アプローチに基づき Pub/Sub に MQTT を用いたアーキテクチャの構造を図 2 に示す．受信層，配信層の詳細を以下に示す．

(1) 受信層には次の 3 機能を持たせる．1)デバイスと他のエッジからのメッセージ受信，2)同じエッジの配信層にメッセージをフィルタリングして配信，3)受信層リストを参照し，エッジ間

†1 南山大学大学院 理工学研究科 ソフトウェア工学専攻
Graduate Program of Software Engineering, Nanzan University

スケールアウトによる他のエッジへのメッセージ配信．

(2) 配信層には次の 2 つの機能を持たせる．1)受信層からメッセージを受信，2)クラウドからのリクエストと Subscription に従いメッセージをフィルタリングして配信．また，配信層は非同期配信を行う Broker ノードと，メッセージの保存と同期配信とを行う KVS ノードからなる．メッセージの保存は非同期配信と比べ負荷が大きくなるため，各配信層の KVS ノードは負荷に応じて増やすことが可能である．Broker ノードと KVS ノードの詳細を以下に示す．

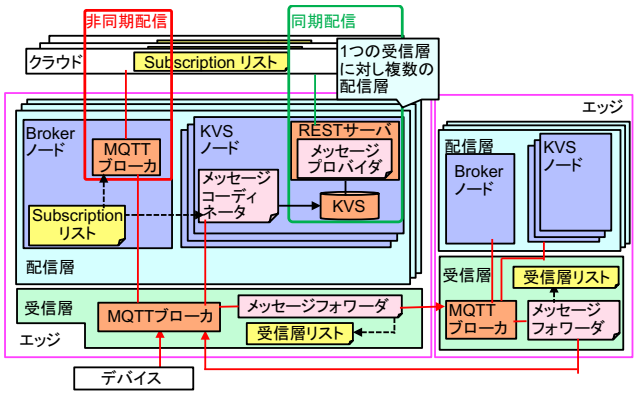

図 2 アーキテクチャの構造

1) Broker ノード: 受信層の MQTT ブローカの Subscriber として動作しクラウドに配信する MQTT ブローカと，受信するメッセージのトピックを保存する Subscription リストからなる．

2) KVS ノード: 受信層からメッセージを受信し KVS に保存するメッセージコーディネータ，メッセージを一定期間保存する KVS，クラウドにメッセージを同期配信する REST サーバとメッセージプロバイダからなる．

それぞれのエッジはエッジ内スケールアウトによって，1 つの受信層と複数の配信層からなる．クラウドからの詳細なトピックを指定した Subscription は受信層で受信する．受信層の MQTT ブローカは Subscription リストのトピックのうち共通部分を抽象化し配信層に Subscription を送信する．これにより，各 MQTT ブローカが参照する配信先の数を削減でき，エッジのスケーラビリティを向上する．

5.2 メッセージ配信モデル

エッジがデバイスからメッセージを受信してからクラウドに配信するまでのメッセージ配信モデルを示す．ただし，デバイスはエッジ B にメッセージを Publish し，クラウドはエッジ A からメッセージを受信し，クラウドと配信層からの Subscription は送信されているとする．

5.2.1 エッジ間スケールアウトのメッセージ配信モデル

図 3 にエッジ間スケールアウトのメッセージ配信モデルを示す．デバイスがエッジ A 以外のエッジ(エッジ B)に接続される場合，エッジ B の受信層は配信層からデバイスのメッセージの Subscription を受信していないため，配信層にメッセージを配信しない．エッジ B の受信層のメッセージフォワーダは同じ受信層の MQTT ブローカからの全てのメッセージを受信する．エッジ B のメッセージフォワーダは配信層リストを参照し，受信したメッセージごとにそのメッセージを Subscribe する配信層が存在するエッジの受信層の MQTT ブローカに Publish する．エッジ B が Publish したメッセージを受信したエッジ A の MQTT ブローカは配信層とエッジ A のメッセージフォワーダにメッセージを配信する．メッセージフォワーダは配信層リストを参照し，メッセージを Publish する先が導入されたエッジと一致する場合，そのメッセージを他のエッジに Publish せず破棄する．これにより，配信層は 1 つの受信層からのメッセージのみを受信可能にできる．

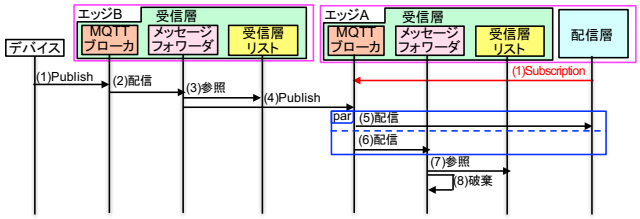

図 3 エッジ間スケールアウトのメッセージ配信モデル

5.2.2 受信層と配信層間のメッセージ配信モデル

図 4 に受信層と配信層間のメッセージ配信モデルを示す．Broker ノードはクラウドからの Subscription を受信し，Subscription リストを生成する．Broker ノードの MQTT ブローカと KVS ノードのメッセージコーディネータがそれぞれ独立して受信層の Subscriber として動作することで，非同期配信におけるスケーラビリティを向上する．

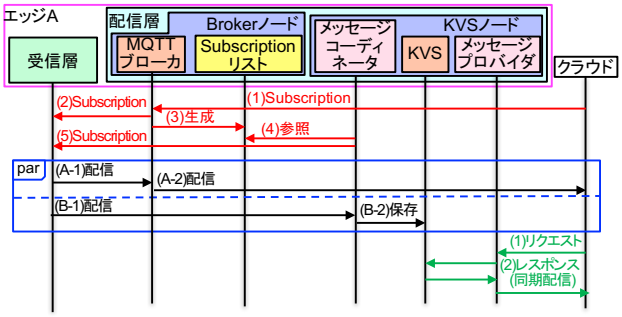

図 4 受信層と配信層間のメッセージ配信モデル

6. まとめ

本研究では Pub/Sub とエッジを統合した IoT アーキテクチャを提案した．提案アーキテクチャでは，階層化とスケールアウトによってスケーラビリティを実現した．デバイスからのメッセージをクラウドに配信するまでのメッセージ配信モデルを定義した．今後の課題として，配信するメッセージのトピックが変化する場合の Subscription リストや受信層リストの管理方法を検討する．

参考文献

[1] R. Buyya, et al. (eds.), Internet of Things, Morgan Kaufmann, 2016.
[2] ISO/IEC 20922:2016, Information Technology - Message Queuing Telemetry Transport (MQTT) V. 3.1.1, ISO, 2016.
[3] oneM2M, Functional Architecture, Aug. 2016.
[4] W. Shi and S. Dustdar, The Promise of Edge Computing, IEEE Computer, Vol. 49, No. 5, May. 2016, pp. 78-81.
[5] S. Tarkoma, Publish/Subscribe Systems, Wiley, 2012.