Vol.2018-GN-103 No.1
Vol.2018-CDS-21 No.1
Vol.2018-DCC-18 No.1
2018/1/26

フレキシブル IoT エッジコンピューティングモデルと応用

荻野正[†1]　北上眞二[†2]　白鳥則郎[†3]

概要：クラウドコンピューティングを中心とする大規模な IoT システムにおいて，ネットワーク負荷の増大，応答遅延やプライバシー侵害などの問題が懸念されている．解決策として，エッジコンピューティングが提案されている．しかしながら，単にクラウドの処理をエッジに移すだけでは，ローカルの IoT システム間のデータの共有ができない等から利用者の利便性を損ねることになる．我々は，この問題を解決するために，マルチエージェントベースの柔軟な IoT エッジコンピューティングアーキテクチャを提案している．提案アーキテクチャでは，クラウドとエッジの機能分担を動的に変更することで，クラウドでの全体最適化とエッジでの個別最適化を適切に実現する．本稿では，そのアーキテクチャモデルの説明と，エネルギー管理システムと ITS への応用例について示す．

キーワード：IoT，エッジコンピューティング，クラウドコンピューティング，フレキシブル，マルチエージェント

Flexible IoT Edge Computing Model and Its Applications

TADASHI OGINO[†1]　SHINJI KITAGAMI[†2]
NORIO SHIRATORI[†3]

Abstract: On a large-scale IoT system based on cloud computing, problems such as increase of network load, delay in response, invasion of privacy, are concerned in recent years. As a solution to this problem, edge computing has introduced to the IoT systems. However, if you migrate the cloud function to the edge too much, the collected data cannot be shared between IoT systems and this decreases the usefulness of the IoT system. In this paper, we propose a multi-agent based flexible IoT edge computing architecture to balance global optimization by a cloud and local optimization by edges and to optimize the role of the cloud server and the edge servers dynamically. Also, as its application examples, we introduce an energy management system and a parking management system based on proposed edge computing system architecture to show the effectiveness of our proposal.

Keywords: IoT, edge computing, cloud computing, flexible, multi-agent

1. はじめに

近年，多くのセンサや装置がインターネットに直接接続され，人間の介在なしに様々なサービスを提供する IoT システムが注目されている[1]． IoT システムは，産業部門，家計部門，社会部門などでの利用が拡大しているクラウド集中型アーキテクチャである． 大規模なクラウド集中型 IoT アーキテクチャのシステムでは，ネットワーク負荷の増大，フィードバック応答の遅延，プライバシー侵害などの問題があることが指摘されている[2].

クラウド集中型 IoT アーキテクチャの問題を解決するために, IoT アーキテクチャにエッジコンピューティング（EC）の概念が導入されている[2,3]．EC では，データ収集機能，フィルタリング機能，およびフィードバック制御機能が，クラウドではなく，例えば携帯電話基地局のエッジサーバあるいはセンサとアクチュエータに近い IoT ゲートウェイなどに実装される．その結果，通信トラフィックの不足やフィードバック制御機能の遅延を解決することが可能となる．しかしながら，エッジサーバへの過度のクラウド機能の移行は，ローカル IoT システム間で収集されたデータを共有することが困難となる等から，IoT システムの利便性を低下させる[4]．また，EC はエッジドメインにおけるローカル最適化に有効であるが，複数ドメインのグローバル最適化を実現することはできない.

これらの問題を解決するために，我々はいくつかの研究を行ってきており，クラウドとエッジの役割を最適化する環境適応型 IoT アーキテクチャを提案した[5-7].
本論文では，これらの研究を更に拡張し，従来の EC の問題点を解決するための柔軟な IoT エッジコンピューティングアーキテクチャを提案する．提案する IoT アーキテクチャでは，クラウドによるグローバルな最適化とエッジによるローカル最適化のバランスをとり，マルチエージェント技術によってクラウドサーバとエッジサーバの役割を動的に最適化する.

2. 背景

現在，さまざまな IoT アーキテクチャが標準化団体や研究者によって提案されている[1]．しかしながら，以下のような問題点があることが分かっている[2].

1）センサ数が多い大規模な IoT システムでは，膨大な量のデータを収集することでネットワークの通信帯域が枯渇し，サービスレベルが低下する.
2）インターネットとクラウドにおける通信の集中が制御遅延の原因となる．制御遅延は，データ収集の頻度にも依存する.
3）クラウド上にすべてのデータを収集するため，セキュ

†1 明星大学
Meisei University
†2 福井工業大学
Fukui University of Technology
†3 中央大学
Chuo University

Vol.2018-GN-103 No.1
Vol.2018-CDS-21 No.1
Vol.2018-DCC-18 No.1
2018/1/26

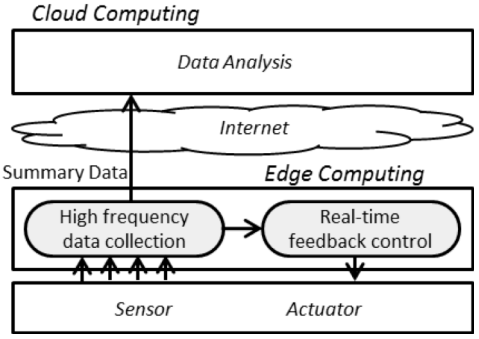


Fig.1. IoT Edge Computing

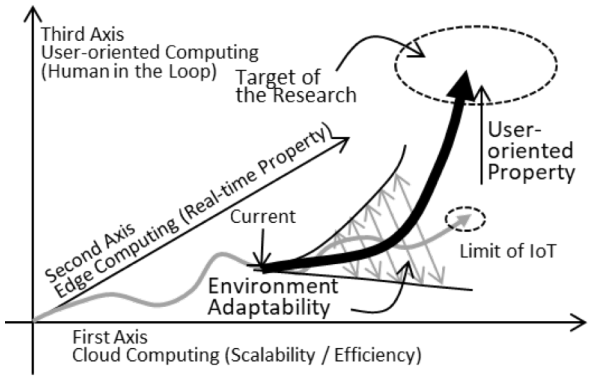


Fig.2. Concept of Environmental Adaptive IoT Architecture

リティ問題が懸念される．

これらの問題点を解決するために，エッジコンピューティング(EC)の概念がIoTアーキテクチャに導入された[3,4]. EC は，データの発生源または制御対象の近くの場所で処理を実行する．Fig.1 に，EC システムを使用した IoT アーキテクチャ（IoT-EC）の構成を示す．このアーキテクチャでは，データ収集機能，フィルタリング機能およびフィードバック制御機能が，IoT デバイスに近い携帯電話基地局のエッジサーバや IoT ゲートウェイに実装される．

IoT-EC は，ネットワークトラフィックの不足やフィードバック制御の遅延を解消するのに有効である．しかし，以下に示す課題がある[4,7].

課題 1）大規模な IoT システムでは，すべてのエッジサーバに同じリソースを展開することは困難であり，リソースの異なるエッジサーバが混在することになる．この時，エッジサーバのリソースとネットワーク環境の変化に応じて，クラウドとエッジ部の役割を動的に変更し，IoT システム全体を最適化する必要がある．

課題 2）すべての IoT 機能がエッジサーバに配置されると，垂直統合システムになり，全体的な最適化が困難になる．逆に，クラウドでの全体最適化を優先すると，機器の制御などの個別最適化が妨げられる．クラウドによる全体最適化とエッジによる個別最適化のバランスをとるための仕組みが必要になる．

課題 3）ある領域での個別最適化が，他の領域の個別最適化を妨げる可能性がある．その場合，エッジサーバ間で個別最適化を行う仕組みが必要になる．

3. マルチエージェントベースのフレキシブル IoT エッジコンピューティングモデル

3.1 基本コンセプト

我々は第 2 章で述べた IoT-EC の問題を解決するために，やわらかい IoT エッジコンピューティングの研究を行ってきている[4-9]. IoT-EC の課題 1 に対しては，環境適応型 IoT アーキテクチャを提案した[4,5]. Fig.2 に環境適応型 IoT アーキテクチャの概念を示す．

次に，この研究を更に発展させ，IoT-EC の課題 2 を解決し，クラウドによるグローバルな最適化とエッジでのローカル最適化のバランスをとる柔軟な IoT エッジコンピューティングアーキテクチャについて提案した[8,9]. Fig.3 に示すように，一般に，クラウドコンピューティングによる全体最適化とエッジコンピューティングによる個別最適化はトレードオフの関係にある．すなわち，クラウドコンピューティングは，システムの全体最適化には効果的であるが，エッジ側のリアルタイム処理などに係わる個別最適化には不向きである．逆に，エッジコンピューティングは，広域にわたるシステムの全体最適化には不向きである．本稿で提案する IoT アーキテクチャは，クラウドとエッジ間で最

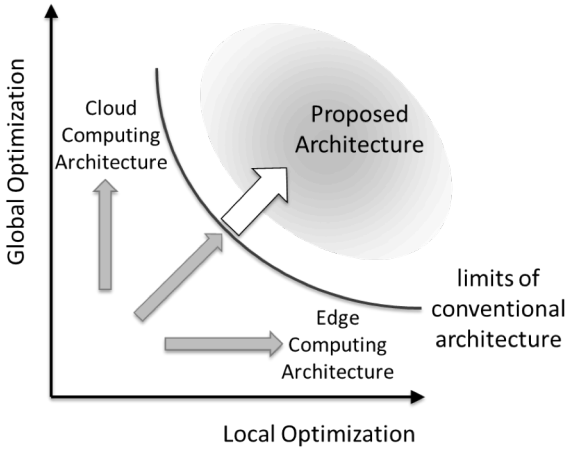


Fig.3. Balancing Cloud Performance and Edge Performance

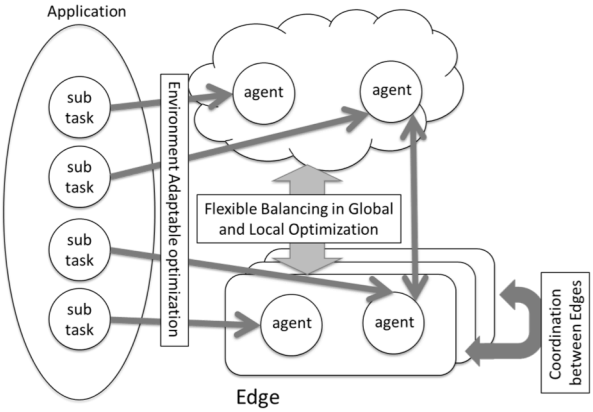


Fig.4. Multi-agent based Architecture of Flexible IoT Edge Computing

Vol.2018-GN-103 No.1
Vol.2018-CDS-21 No.1
Vol.2018-DCC-18 No.1
2018/1/26

小限の情報を交換することで，システム全体の最適化を実現することを目的とする．

3.2 アーキテクチャ

Fig.4 に，提案する IoT アーキテクチャを示す．アプリケーションは複数のサブタスクから構成され，クラウドまたはエッジにエージェントとして割り当てられる．たとえば，全体最適化サブタスクは，クラウドで収集されたすべてのデータにアクセスする必要があるため，クラウドに配置される．リアルタイム性が要求される機器制御サブタスクは，低遅延通信を必要とするため，制御対象に近いエッジサーバに割り当てられる．

アプリケーションをサブタスクに分割してクラウドとエッジに配置する場合，システム全体を適切に動作させるための仕組みが必要になる．システムの最適化を制御するエージェントがクラウドとエッジの両方に存在する場合，両方のエージェントが連携して，クラウドとエッジの両方の視点からシステム全体を適切にバランスさせる必要がある．場合によっては，エッジエージェントが近傍の異なるエッジエージェントと通信する必要がある．提案アーキテクチャでは，これらの機能はエッジ間の連携として実現される．クラウド間についても同様である．

3.3 定式化

アプリケーション全体の最適化のバランスをとるためには，クラウドの全体最適化機能とエッジの個別最適機能は，最適化プロセスの詳細情報を交換しながら，システム全体の最適化プロセスの調整をする必要がある．

アプリケーションがサブタスクに分割され，クラウドとエッジに配置されると，各サブタスクは，システム全体を最適化するために必要なシステム情報へのアクセスが制限されるようになる．その結果，クラウドのサブタスクはクラウドシステムのみを最適化でき，エッジのサブタスクはエッジシステムのみを最適化できることになる．

この状態を以下のように定式化する．

variables:

$$\boldsymbol{v_c} = [v_{c0}, v_{c1}, \cdots v_{cK-1}] \quad \text{; from cloud}$$
$$\boldsymbol{v_e} = [v_{e0}, v_{e1}, \cdots v_{eL-1}] \quad \text{; from edges}$$
$$\boldsymbol{v_s} = [v_{s0}, v_{s1}, \cdots v_{sM-1}] \quad \text{; from both}$$
..... (1)

cost function:

$$cost_c(\boldsymbol{v_s}, \boldsymbol{v_c}) \quad \text{; for cloud}$$
$$cost_e(\boldsymbol{v_s}, \boldsymbol{v_e}) \quad \text{; for edge}$$
$$cost_t(\boldsymbol{v_s}, \boldsymbol{v_c}, \boldsymbol{v_e}) = cost_c(\boldsymbol{v_s}, \boldsymbol{v_c}) + k * cost_e(\boldsymbol{v_s}, \boldsymbol{v_e}) \quad \text{; for both cloud and edge}$$
..... (2)

global optimization (cloud):

$$min\big(cost_c(\boldsymbol{v_s}, \boldsymbol{v_c})\big) \text{ under } constraints_c(\boldsymbol{v_s}, \boldsymbol{v_c})$$

local optimization (edge):

$$min\big(cost_e(\boldsymbol{v_s}, \boldsymbol{v_e})\big) \text{ under } constraints_e(\boldsymbol{v_s}, \boldsymbol{v_e})$$

total optimization:

$$min\big(cost_t(\boldsymbol{v_s}, \boldsymbol{v_c}, \boldsymbol{v_e})\big) \text{ under } constraints_t(\boldsymbol{v_s}, \boldsymbol{v_c}, \boldsymbol{v_e})$$
..... (3)

変数$\boldsymbol{v}$ は，アクセス可能なノードの配置によって 3 つのタイプに分類される．$\boldsymbol{v_s}$は共有変数で，クラウドとエッジの両方に影響を与える情報である．$\boldsymbol{v_c}$は，クラウドの動作のみに影響を与える変数である．$\boldsymbol{v_e}$は，エッジの振る舞いにのみ影響する変数である．

$cost_c(\,)$および $cost_e(\,)$は，それぞれクラウドまたはエッジでのみ計算されるコスト関数である．$cost_t(\,)$は総コスト関数であり，システム全体の最適化は，すべての制約の下でこのコスト関数を最小にする事であると定義する． kは，クラウドの処理とエッジの処理のバランスを調整するためのパラメータである． ここでは総コストは $cost_c(\,)$ と $cost_e(\,)$の一次式と仮定する．

クラウドとエッジは，Fig.5.に示すように共有変数を交互に交換することで，最適化を実現する．

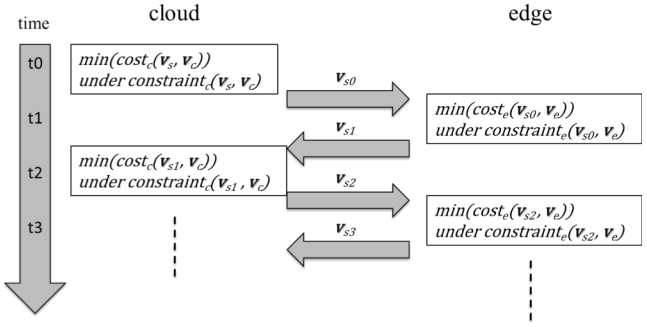


Fig.5. Protocol for Optimizing Between Cloud and Edges

4. 全体最適化と個別最適化の例

以下の章で，提案アーキテクチャのアプリケーションとして，次世代 EMS(Energy Management System)と次世代 ITS(Intelligent Transport System)を取り上げる．Fig.6, Fig.7 に，各システムの全体最適化と個別最適化の概念図を示す．

次世代 EMS については，電力供給が逼迫した時に需要家の電力消費を削減することにより電力需給を安定化させる DR(Demand Response) の導入が進んでいる．DR では，RA(Resource Aggregator)が電力供給と電力需要が一致するように，需要家の電力消費を抑制する．需要家は，あらかじめ契約した削減量に対する実削減量の割合に応じて報酬金を得る．しかしながら，需要家への太陽光発電や蓄電池の導入が拡大すると，需要家の消費電力の予測が困難になる．また，現在の DR は電力需給バランスの維持や需要家に対するユニバーサルサービスを目的としており，それぞ

Vol.2018-GN-103 No.1
Vol.2018-CDS-21 No.1
Vol.2018-DCC-18 No.1
2018/1/26

れの需要家における快適性や省エネ性は考慮されない．つまり，EMS においては，RA がクラウドサービス，需要家 EMS がエッジサービスの位置づけとなる，また，RA による電力需給バランスとユニバーサルサービスが EMS の全体最適化，需要家 EMS による快適性や省エネ性の維持が EMS の部分最適化となる．

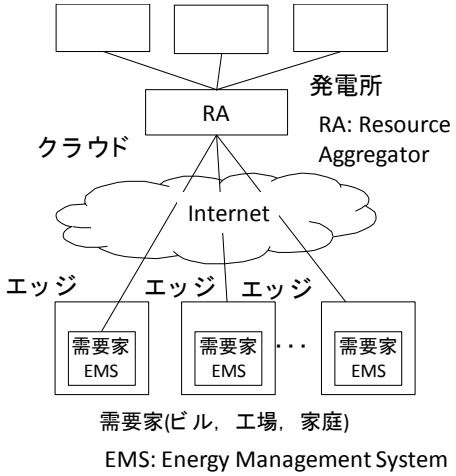


Fig.6. 次世代 EMS の構成

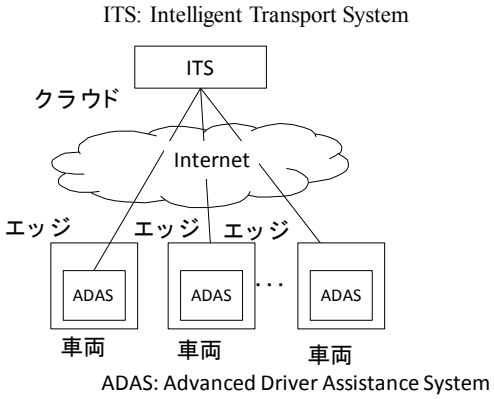


Fig.7. 次世代 ITS の構成

次世代 ITS は，交通事故の減少と交通渋滞の緩和のための管理と制御を行う．一方で，自動車単体の自動運転／運転支援システムは，目的地への早期到着と運転者の快適性を確保するための制御を行う．すなわち，安心・安全で快適な交通社会を実現するためには，次世代 ITS と自動運転／運転支援システムの連携が欠かせない．ここで，次世代 ITS がクラウド，各車両の自動運転／運転支援システムがエッジの位置づけとなり，次世代 ITS による交通事故減少と交通渋滞の緩和が全体最適化，自動運転／運転支援システムによる目的地への早期到着と運転者の快適性確保が個別最適化となる．

以下の章で,それぞれのアプリケーションへの本アーキテクチャの適用について詳細を述べる．

5. エネルギー管理への適用

本章では，提案アーキテクチャのエネルギー管理への適用について述べる．

5.1 電力デマンドレスポンス(DR)

近年，電力供給が逼迫したり電力料金が急騰したりした時に需要家の電力消費を削減することにより電力需給を安定化させる電力デマンドレスポンス(DR)システムの導入が進んでいる[12,13]．DR システムの基本構成を Fig.8 に示す．DR では，リソースアグリゲータ(RA)が，翌日の天気予報や過去の電力消費データに基づいて電力需要を予測し，電力供給と電力需要が一致するように，需要家の電力消費を削減する．一方，需要家は，あらかじめ契約した削減量に対する実削減量の割合に応じて報酬金を得る．

しかしながら，需要家への太陽光発電や蓄電池(Fig.8 の SB)の導入が拡大すると，需要家の消費電力の予測が困難になる[11]．また，現在の DR は電力需給バランスの維持（すなわち，電力システムの全体最適化）を目的としており，それぞれの需要家における快適性や省エネ性（エネルギー消費の個別最適化）は考慮されない[7]．今後，DR を普及させるためには，電力システムの全体最適化とエネルギー消費の個別最適化を両立させる高度 DR システムの開発が欠かせない．

5.2 高度 DR の定式化

高度 DR において，RA によるクラウドサービスが実現する全体最適化と，各需要家のエネルギー管理システム(EMS)が実現させる個別最適化の要件と制約は以下の通りとなる．

（1）RA による全体最適化

電力需要が電力供給を上回る場合は，RA はインターネットを介して需要家に対して電力消費の削減を要請する．Fig.9 (a)に，RA による電力需給の全体最適化のイメージを示す．ここで，各需要家に要請する電力削減量は平等に配分するものとする．

RA による全体最適化の目的関数を式(4)に示す．

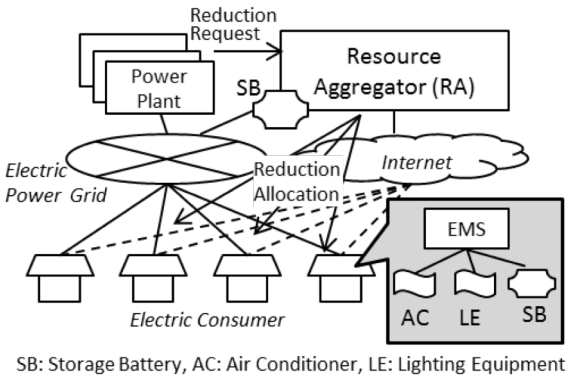
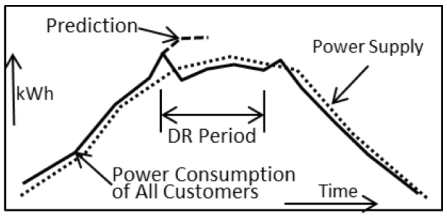
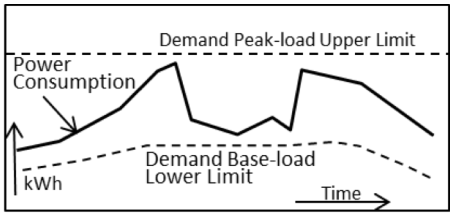


Fig.8. Electric Power Demand Response System

Vol.2018-GN-103 No.1
Vol.2018-CDS-21 No.1
Vol.2018-DCC-18 No.1
2018/1/26



(a)



(b)

Fig.9. Optimization of Advanced DR System
(a) Global Optimization by RA
(b) Local Optimization by Consumer's EMS

$$\boldsymbol{v}_s = \left[\left(D_1(t), C_1(t)\right), \left(D_2(t), C_2(t)\right), \cdots \left(D_n(t), C_n(t)\right)\right]$$

$$\boldsymbol{v}_c = [S(t)]$$

$$Cost_c(\boldsymbol{v}_s, \boldsymbol{v}_c) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(\frac{C_k(t)}{R_k} - \overline{CR(t)}\right)^2} \quad \ldots (4)$$

$$\overline{CR(t)} = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{C_k(t)}{R_k}\right)$$

$$\sum_{k=1}^{n} C_k(t) = \sum_{k=1}^{n} D_k(t) - S(t)$$

ここで，$S(t)$は時刻 t における電力供給量，$D_k(t)$は時刻 t における需要家 k の電力需要量，$C_k(t)$は時刻 t における需要家 k に対する削減要請量，R_kは需要家 k の契約削減量，n は需要家数である．

式(4)は，それぞれの需要家の契約削減量に対する実削減量の割合の標準偏差を意味する．RA は，この目的関数が最小になるように，各需要家への削減量を配分する．

RA による全体最適化の制約条件を式(5)に示す．

$$C_k(t) \geq 0,\ D_k(t) \geq 0,\ R_k \geq 0,\ \mathrm{S(t)} \geq 0 \quad \ldots (5)$$

電力需給をバランスさせるためには，RA や各需要家に導入される蓄電池の活用が有効である[13]．この場合，電力需給の状況に合わせて蓄電池を効果的に蓄放電制御する必要となる．

（2）需要家 EMS による個別最適化

需要家 EMS は，需要家内の設備を制御し，快適性と省エネ性を維持する．特に，快適性を維持するために，RA から電力消費を削減する要請がきた場合であっても，最小電力消費（需要ベースロード下限）を下回らないように制御する． また，省エネ性を維持するために，所定の消費電力上限値（需要ピークロード上限）を超えないように制御する． Fig.9（b）は，需要家 EMS による快適性と省エネ性を維持するための個別最適化のイメージを示す．

需要家 EMS による個別最適化の目的関数を式(6)に示す．

$$\boldsymbol{v}_s = \left[\left(D_1(t), C_1(t)\right), \left(D_2(t), C_2(t)\right), \cdots \left(D_n(t), C_n(t)\right)\right]$$

$$\boldsymbol{v}_e = [B_1(t), B_2(t), \cdots B_n(t)]$$

$$Cost_e(\boldsymbol{v}_s, \boldsymbol{v}_e) = \sum_{k=1}^{n}\left(\left(1 - \frac{C_k(t)}{R_k}\right) + S\left(\frac{B_k(t)+C_k(t)-D_k(t))}{R_k}\right) + S\left(\frac{U_k(t)+C_k(t)-D_k(t))}{R_k}\right)\right) \quad \ldots (6)$$

$$S(x) = \begin{Bmatrix} x, when\ x > 0 \\ 0, when\ x \leq 0 \end{Bmatrix}$$

ここで，$B_k(t)$は需要家の快適性を維持するための時刻 t における需要ベースロード下限，$U_k(t)$は需要家の省エネ性を維持するための時刻 t における需要ピークロード上限である．

式(6)の第 1 項は，契約削減量に対する実削減量の割合を意味する．式(6)の第 2 項は，快適性についてのコストを意味する．電力消費が需要ベースロード下限を下回った場合は，目的関数に快適性コストが加わる．式(6)の第 3 項は，省エネ性についてのコストを意味する．電力消費が需要ピークロード上限を上回った場合は，目的関数に省エネ性コストが加算される．

需要家 EMS による個別最適化の制約条件を式(7)に示す．

$$R_k - C_k(t) \geq 0, B_k(t) \geq 0,\ U_k(t) \geq 0 \quad \ldots (7)$$

$$D_k(t) - B_k(t) - C_k(t) \geq 0$$

$$D_k(t) - U_k(t) - C_k(t) \geq 0$$

電力需要が電力供給を大きく下回った場合は，需要家に設置した蓄電池への充電により，需要家の電力需要を増加させるように要請する（いわゆる上げ DR）．この場合，需要家における省エネ性を維持するために，電力需要が需要ピークロード上限を上回らないように，蓄電池への充電量を制御する．

式(4)と式(6)に示した通り，高度 DR の全体最適化と個別最適化は，時系列最適化問題となる．また，RA（クラウド）と各需要家 EMS（エッジ）の目的関数をバランスよく最小化するためには，RA と需要家 EMS の間で調整が必要となる．RA と各需要家 EMS の目的関数の重みが等しいものと仮定すると，高度 DR 全体の目的関数は式(8)の通りとなる．

$$Cost_c(\boldsymbol{v}_s, \boldsymbol{v}_c, \boldsymbol{v}_e) = Cost_c(\boldsymbol{v}_s, \boldsymbol{v}_c) + Cost_e(\boldsymbol{v}_s, \boldsymbol{v}_e) \quad \ldots (8)$$

Vol.2018-GN-103 No.1
Vol.2018-CDS-21 No.1
Vol.2018-DCC-18 No.1
2018/1/26

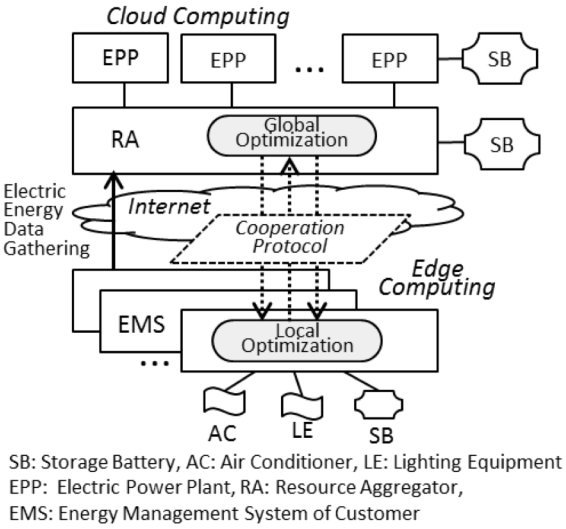


Fig.10. System Configuration of Advanced DR System

5.3 システム構成

以上に示した定式化に基づく高度 DR システムのシステム構成図を Fig.10 に示す．電源供給が逼迫する場合，RA は，式（6）に基づいて電力需給の均衡をとるために各顧客に割り当てるべき削減要求の電力量を計算する． 一方，顧客の EMS は，快適性と省エネルギーを維持した状態で，式（7）に基づいて電力需要を調整する． ここで，RA と需要家 EMS 間で式(4)および式(6)の $\boldsymbol{v_s}$ のデータを交換することにより，RA による全体最適化と需要家 EMS の個別最適化のバランスを取る．

Table 1. Simulation Condition

	C1	C2	C3	Sum
R_k [kWh]	150	290	50	490
$D_k(t_j)$ [kWh]	300	300	300	900
$B_k(t_j)$ [kWh]	100	200	250	550
margin [kWh]	200	100	50	350

R_k : contract reduction amount of customer k
$D_k(t_j)$: power consumption of customer k at time t_j
$B_k(t_j)$: demand base-load lower limit of customer k at time t_j
margin:margin for reducing power consumption of customer k

Table 2. Results of Optimization

		C1	C2	C3	RA	$Cost_t$
Step 1	$C_k(t_j)$ [kWh]	92	177	31	300	-
	Cost	0.39	0.65	0.39	0.00	1.43
		1.43				
Step 2	$C_k(t_j)$ [kWh]	150	100	50	300	-
	Cost	0	0.65	0	0.31	0.96
		0.65				

$C_k(t_j)$: reduction allocation amount to customer k at time t_j

5.4 シミュレーション評価

ここで，需要家が 3 件（C1，C2 および C3）の場合について提案アーキテクチャに基づく高度 DR システムのシミュレーション結果を示す．Table1 に，時刻 t_j における各需要家 k の契約削減容量 R_k，電力需要量 $D_k(t_j)$，需要ベースロード下限 $B_k(t_j)$を示す．

電力供給が 600kWh（すなわち，電力供給が 300kWh 不足）の場合，RA による全体最適化の結果を Table2 の Step1 に示す．ここで，$C_k(t_j)$ は，時刻 t_j における各需要家 k 対する削減要請量である．Step1 では，RA の最適化コストは 0 となるが，需要家 2 に対する削減要請量 177kWh が需要ベースロード下限を下回り快適性が劣化するため，需要家 2 の最適化コストが高くなった．これは，需要家 EMS による個別最適化は反映されておらず，従来型の DR システムの制御結果と一致する．

Table2 の Step2 は，各需要家 EMS による個別最適化を反映した結果である．需要家 2 については，最適化コストが 0 になるが，その分削減量が減少するためコストは変化しない．しかし，需要家 1 と 3 の削減量が増えるため，Step1 に比べて最適化コストが改善した．

このように，提案アーキテクチャに基づく高度 DR システムでは，RA の全体最適化（電力需給のバランス）と需要家 EMS の個別最適化（快適性の維持）を両立させることが可能となり，全体の最適化コストを 1.43 から 0.96 に約 33%改善することができた．

6. ITS への適用

本章では，提案アーキテクチャの ITS システムの駐車場管理システムへの適用について述べる．

6.1 ITS と駐車場検索システム

ITS(Intelligent Transport Systems，高度道路交通システム)とは，人と道路と自動車の間で情報の受発信を行い，様々な課題を解決するシステムである．自動車自体は多数のセンサを持つ情報処理装置であり，IoT システムのエッジであると捉えることができる．その周りに存在する人や駐車場等の設備も同じくエッジであり，これらのエッジからの情報を集約して処理するクラウドが存在する大規模なクラウド–エッジシステムである[12,13]．本論文では，ITS の中で比較的単純な駐車場検索システムに対して我々の提案するアーキテクチャを適用して評価を行う．

6.2 駐車場検索システムの定式化

ここでは，車が駐車できる駐車場を探すシステムを駐車場検索システムとする．課金システム，駐車支援システム，周遊システム等は考慮せず，1 台の車が 1 回だけ駐車できる駐車場を見つけるシステムとする．駐車場検索システムでは，各自動車が，駐車したい駐車場を指定して駐車可能かどうかを確認し，可能であれば駐車する．

自動車は，駐車できるまでの待ち時間を最小にすること

Vol.2018-GN-103 No.1
Vol.2018-CDS-21 No.1
Vol.2018-DCC-18 No.1
2018/1/26

を目標とする．駐車場側は，できるだけ空きスペースがないようにする．

$$Cost_{e,carN} = T_{w,carN}$$
$$Cost_{e,parkingM} = 1 - U_{parkingM} \quad \ldots (7)$$

ここで，$T_{w,carN}$ は，車 N の待ち時間，$U_{parkingM}$ は駐車場 M の利用率とする．エッジ全体のコスト関数は以下の通り．

$$Cost_e = \Sigma\, T_{w,carN} + k \times \Sigma\, Cost_{e,parkingM} \quad \ldots (8)$$

k は，全体のコスト関数を計算する時の，車の待ち時間と駐車場の空き率の重みを調整するパラメータである．

車，駐車場いずれも複数が存在し，どちらもエッジであるとしている．通常は，各車と各駐車場が個別に 1 対 1 で情報交換するため，全体最適を司る機能は存在せず，エッジのコスト関数を計算することができない．そこで，提案システムでは，車および駐車場の調停を行うアグリゲータ（aggregator）をクラウドに導入することで，全体最適化を図る．

$$Cost_c = Cost_e = \Sigma\, T_{w,carN} + \mathrm{k} \times \Sigma\, Cost_{e,parkingM} \quad \ldots (9)$$

6.3 システム構成

全体最適の機能が存在しない場合には，Fig.11 のように車と駐車場が個別にエッジ間通信を行うことで，駐車場所の検索を行っている．提案システムでは，Fig.12 のようにクラウドに配置される全体最適を司るアグリゲータを介して車と駐車場がエッジ-クラウド間通信を行う．

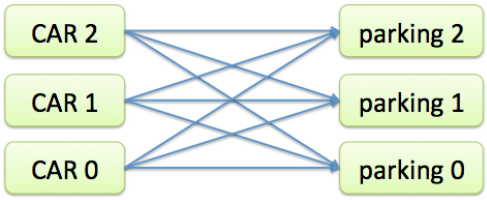


Fig.11. Traditional Parking System

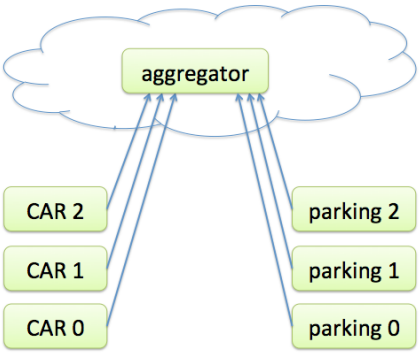


Fig.12. Cloud Edge Parking System

6.4 通信プロトコル

駐車場を検索する場合は，車からアグリゲータに，希望駐車場のリクエストを送る．アグリゲータは駐車場に問い合わせし，その駐車場に空きがあればその情報が，空きがない場合には代替駐車場の情報を返信する．

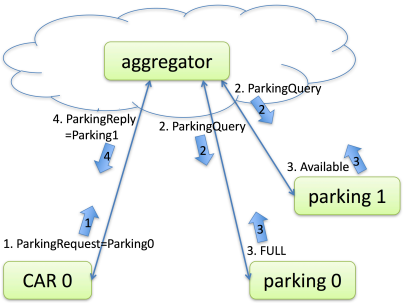


Fig.13. Protocol for Cloud Edge Parking System

6.5 シミュレーション評価

駐車場が 2 個所のシンプルなケースでのシミュレーションを行う．車は指数分布に従って目標の駐車場に到達する．駐車スペースがない場合には，従来システムではその駐車場で空きができるまで待つものとする．新システムでは，もう一方の駐車場に問い合わせをし，空いている場合には移動して駐車する．この時，移動には移動時間がかかり，待ち時間にはこの移動時間も含むものとする．

駐車場 0 の駐車可能台数を 10 台，駐車場 1 は 10 台，出発間隔は平均 10 分，駐車時間は平均 50 分～100 分，移動時間 5 分，シミュレーション台数 5,000 台の場合の待ち時間結果を以下に示す．

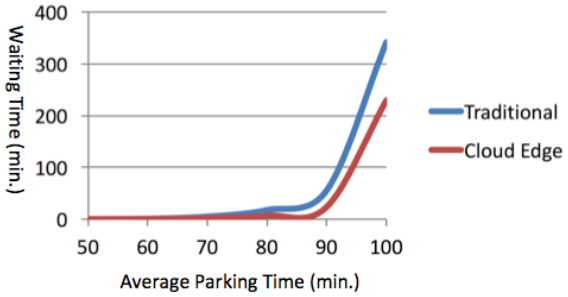


Fig.14. Waiting Time for Case0

待ち時間は移動時間を含めても 30%以上削減している．また，駐車場の利用率結果を以下に示す．

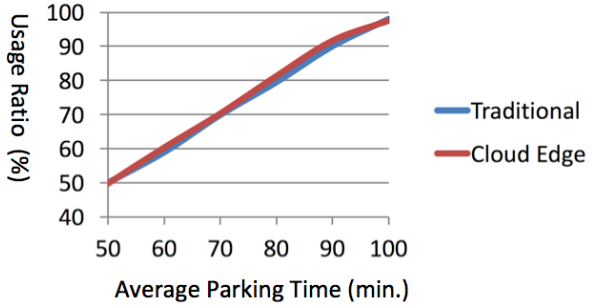


Fig.15. Usage Ratio for Case0

Vol.2018-GN-103 No.1
Vol.2018-CDS-21 No.1
Vol.2018-DCC-18 No.1
2018/1/26

駐車場の利用率は，従来システムと提案システムではほとんど変わらない．

駐車場１の駐車可能台数を 15 台に増やした場合のシミュレーション結果を以下に示す．

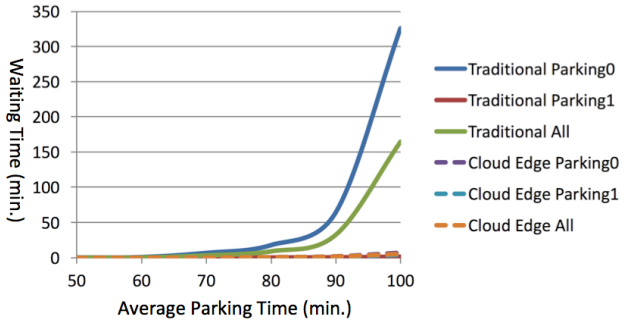


Fig.16. Waiting Time for Case1

従来システムでは，駐車場１の待ち時間は大幅に減少しているが，全体としては駐車場０の待ち時間が先程と同様のため平均値としては約半分になっている．提案システムでは，どちらの待ち時間も大幅に減少していることが分かる．

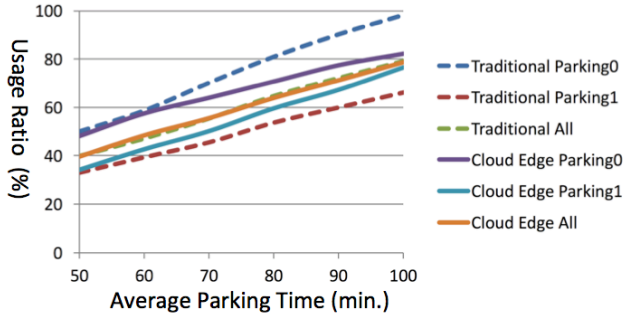


Fig.17. Usage Ratio for Case1

駐車場の全体での利用率自体はほとんど変わらない．

以上の２ケースの場合には，駐車場全体の利用率自体は変わらないが，待ち時間が 30%以上削減されている．実際の応用ケースでは，待ち時間が減少することで車の収容可能台数が増え，トータルとしての利用率は向上していくものと予想される．

7. まとめ

本論文では，クラウドでの全体最適化とエッジでの個別最適化のバランスをとるため，柔軟な IoT エッジコンピューティングアーキテクチャを提案し，その定式化を提案した．また，提案したアーキテクチャを電力デマンドレスポンスシステムと駐車場検索システムに適用し，その有効性を示した．今後の研究として，クラウドとエッジによる分散最適化問題の一般的な定式化とエッジ間の協調制御について検討する予定である．

謝辞　本研究を推進するに当たり，早稲田大学松山泰男名誉教授，同浦野義頼名誉教授，千葉工業大学藤田茂教授，株式会社アイエスイーエム宮西洋太郎殿には，議論に参加いただき貴重なご意見を頂きました．謹んで感謝の意を表します．本稿の一部は，東北大学電気通信研究所における共同プロジェクト研究による支援を受けたものである．

文　献

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communication Surveys & Tutorials, vol.17, no.4, pp. 2347-2376, Jun. 2015.

[2] M.Abdelshkour,"IoT, from Cloud to Fog Computing, "http://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing, Mar. 2015. (accessed 18 Jun. 2017)

[3] P.G. Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric Computing: Vision and Challenges," ACM SIGCOMM Computer Communication Review, vol. 45 issue 5, pp. 37-42, Oct. 2015.

[4] 白鳥則郎，北上眞二，菅沼拓夫，菅原研次，嶋本　薫，“IoT アーキテクチャの最新動向，” 電子情報通信学会誌, vol.100, no.3, pp.214-221, Mar. 2017.

[5] 菅沼拓夫，内林俊洋，北上眞二，菅原研次，白鳥則郎，“やわらかい IoT のための環境適応型アーキテクチャの提案，” 電子情報通信学会技術研究, Vol.116, No.231, pp.13-18, Sep. 2016.

[6] S. Kitagami, V. T. Thanh, D. H. Bac, Y. Urano, Y.Miyanishi, and N.Shiratori, "Proposal of a Distributed Cooperative IoT System for Flood Disaster Prevention and its Field Trial Evaluation," International Journal of Internet of Things, vol. 5, no.1, pp. 9-16, Apl. 2016.

[7] 北上眞二，山本森樹，今村誠，神戸英利，小泉寿男，菅沼拓夫，“オープンソースソフトウェア環境を基盤とした M2M データ分析サービスシステムの開発，” 電気学会論文誌 C, Vol.133, No.8, pp.1521-1528, Aug. 2013.

[8] 荻野正,北上眞二,菅沼拓夫,白鳥則郎, “柔軟な IoT エッジコンピューティングアーキテクチャの提案とエネルギー管理への応用，” 電子情報通信学会技術報告, vol. 117, no. 233, IN2017-35, pp. 1-6, 2017 年 10 月.

[9] S. Kitagami, T. Ogino, T. Suganuma and N. Shiratori, " Proposal of A Multi-agent Based Flexible IoT Edge Computing Architecture Harmonizing Its Control with Cloud," 10th International Workshop on Autonomous Self-Organizing Networks, pp.223-pp.229, Nov. 2017.

[10] P. Palensky and D. Dietrich,"Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads," IEEE Transactions on Industrial Informatics, vol. 7, Iss. 3, pp.381 - 388, Aug. 2011.

[11] 浅野浩志，“電力システム運用における需要側資源の活用，” 電気学会誌, vol. 135, no. 11, pp. 766-771, Nov. 2015.

[12] Y.Usha Devi and M.S.S. Rukmini, "IoT in connected vehicles: Challenges and issues — A review," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Oct. 2016.

[13] D. Peraković, S. Husnjak and I. Cvitić, " IoT infrastructure as a basis for new information services in the ITS environment," 2014 22nd Telecommunications Forum Telfor (TELFOR), Nov. 2014.