

IoT デバイス仮想化のための サービスファンクション最適割り当て手法の検討

関根 響[†] 金井 謙治[†] 金光 永煥^{‡†} 甲藤 二郎[†] 中里 秀則[†]

[†] 早稲田大学 〒169-8555 東京都新宿区大久保 3-4-1

[‡] 東京工科大学 〒192-0982 東京都八王子市片倉町 1404-1

E-mail: [†] {h_sekine, kanai, katto}@katto.comm.waseda.ac.jp

あらまし これまで筆者らは、エッジコンピューティングを活用した IoT デバイス仮想化技術として、IoT サービスの処理機能をサービスファンクションとして分割定義し、ネットワーク上の計算資源に分散配備する仕組みを提案してきた。本稿では、この仕組みにおいて、アプリケーション要件として遅延を考慮したサービスファンクションの最適割り当て手法を検討する。シミュレーション評価によって、検討する最適割り当て手法の有効性を確認する。

キーワード IoT, エッジコンピューティング, サービスファンクション, IoT デバイス仮想化

Optimal Service Function Allocation Method for IoT Device Virtualization

Hibiki SEKINE[†] Kenji KANAI[†] Hidehiro KANEMITSU^{‡†}

Jiro KATTO[†] and Hidenori NAKAZATO[†]

[†] Waseda University 3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan

[‡] Tokyo University of Technology 1404-1, Katakuramachi, Hachioji City, Tokyo, 192-0982, Japan

E-mail: [†] {h_sekine, kanai, katto}@katto.comm.waseda.ac.jp

Abstract Authors proposed IoT device virtualization technology using edge computing. In IoT device virtualization, IoT service is divided into multiple micro service functions, and these functions are allocated to computing resources that are located in the networks. In this paper, we study an optimal service function allocation method that minimizes application end-to-end latency. Through computer simulations, we confirm an effectiveness of the optimal allocation method.

Keywords IoT, Edge computing, Service function, IoT device virtualization

1. はじめに

近年、Internet of Things (IoT)が広く普及しており、スマートシティやスマートホームといった様々な分野において研究開発が行われている[1]。IoTは非常に多くの技術の融合と言え、デバイス、通信、処理においてヘテロな環境を想定する必要がある。例えば、IoTデバイスでは、温度や湿度センサといった少量のデータを生成するものから、カメラやマイクのように画像や動画、音声といった大量のデータを生み出すものまで様々なものが存在する。また、IoTネットワークに注目すると、狭帯域ではあるが広範囲の通信が可能である Low Power Wide Area (LPWA)や、広帯域かつ中範囲をカバーできる IEEE 802.11 に代表される無線 LAN、さらに比較的広帯域でかつ広範囲の通信が可能な Long Term Evolution (LTE)や 5G に代表される移動体通信などが存在する。さらに、IoT アプリケーションに注目すると、温度などの時系列データの統計処理とい

った処理負荷が低いものから、深層学習による画像処理といった処理負荷が高いものまで様々ある。このような状況下で、多種多様な IoT デバイスを効率的に収容し、様々な IoT アプリケーションを処理、配信する IoT プラットフォームが求められている。

このような背景のもと、著者らはこれまでスマートシティに対する IoT プラットフォームの研究開発を行い、その主要技術の一つとして、エッジコンピューティングや仮想化技術を活用した「IoT デバイス仮想化技術」を提案してきた[2]。これは、IoT サービスの処理機能をサービスファンクション(マイクロサービス)として機能単位で分割定義し、ネットワーク内に存在する計算資源に分散配備するという仕組みになる。近年、サービスファンクションの最適な分散配備手法として、ネットワーク資源、計算資源、サービスファンクションの必要な計算資源がわかっている元、最適化問題に定式化し遅延最小を達成する手法が多数提案さ

れている[3, 4, 7-9 など]. そこで, 筆者らも, 多様なサービス要件を満足するようなサービスファンクションの分散配備配置アルゴリズムについて検討する. 本稿では, それに向けて, まずは単純なシナリオを想定し, アプリケーション要件の1つである遅延を考慮したサービスファンクションの最適割り当て手法を最適化問題の定式化により検討する. 性能評価として, 提案手法によって得られた最適配置における実行遅延時間とランダムに配置した場合の実行遅延時間を比較し, 提案手法の有効性を確認する.

2. 関連研究

近年, IoT を支える重要なパラダイムの一つとしてエッジ/フォグコンピューティングの研究が進められている. [3]では, クラウドコンピューティング, モバイルエッジコンピューティング, エンドデバイスからなる3層を統合的に利用するためのEdgeFlowと呼ばれるシステムが提案されている. EdgeFlowにおいて効率的にデータ処理を行うために, タスクの最適割り当て手法を考案している. また, EdgeMeshと呼ばれるネットワーク内に存在するエッジデバイス(エッジコンピューティング)間で計算タスクを共有するシステムが提案されている[4].

また, ネットワークの効率的な利用のため, Network Function Virtualization (NFV)[5]といったネットワーク仮想化技術に関する研究が進められている. NFVでは, ロードバランサやファイアウォールといった従来では各専用のハードウェア上で実行していたネットワーク機能を, 論理的に統合, 分割された汎用サーバ上で実現することが可能になる. Service Function Chaining (SFC)[6]という技術によって, NFVにおけるサービス機能であるVirtualized Network Function (VNF)を適切な順序で連結しサービスを提供する. このようなNFVの研究において, VNFの最適配置問題が数多く議論されている. [7]では, 物理マシンの負荷などの状態や使用率, 容量などを考慮して, 消費エネルギーの最適化を目的とした動的VNF配置手法を提案している. また, クラウドデータセンターでのリソース最適化[8]やネットワークユーティリティ最大化[9]を目的としたVNF配置手法が提案されている.

IoTアプリケーションの実現には, 仮想化技術の利用例が多い. 特にDockerなどのコンテナ型仮想化技術は, ハイパーバイザ型と比べて, アプリケーションの起動にかかるオーバーヘッドが小さく, 近年注目されており, コンテナを利用したアーキテクチャが提案されている[10, 11, 12].

3. IoT デバイス仮想化技術

本章では, 筆者らが[2]にて提案しているIoTデバイス仮想化技術において, 分割定義されたサービスファンクションに最適な処理ノードを割り当てる手法について検討する.

ンクションに最適な処理ノードを割り当てる手法について検討する.

3.1. システム概要

まずシステム概要として, 筆者らが[2]にて提案しているIoTデバイス仮想化技術について, サービスファンクションの展開, 共有およびスケールリングに関して説明する.

IoT デバイス仮想化の主な目的は, IoT デバイスとIoT サービスの処理機能をネットワークレベルで分離し, 異なるアプリケーション間で共有することにある. また, IoT デバイス仮想化技術によって, アプリケーション開発者はIoT デバイスやセンサデバイス自身が生み出すRaw データそのものを気にする必要もなくなり, アプリケーション開発の簡便化にも繋がるのが期待される. 図1にサービスファンクションの展開, 共有およびスケールリングの概要図を示す. まず, IoT サービスの処理機能をサービスファンクション(マイクロサービス)に分割定義する. 各サービスファンクションをIoT デバイス, エッジ/フォグ, クラウドサーバなどのネットワーク内の各種計算資源にアルゴリズムに従い最適配備する. また, これらサービスファンクションを複数のアプリケーション間で効率的に共有するため, 動的にサービスファンクションの展開先を移動(マイグレーション)させることも想定している. さらに, 動的に変化するIoT デバイス数やIoT アプリケーション数に対してサービス品質を維持するために, 各サービスファンクションへ割り当てられている計算資源を適応スケールリングすることも想定している. 以上のように, サービスファンクションをコンテナ型仮想化により実装することで, サービスファンクションの容易な展開, スケールリングおよび共有を可能とするIoT デバイス仮想化技術を[2]では提案している.

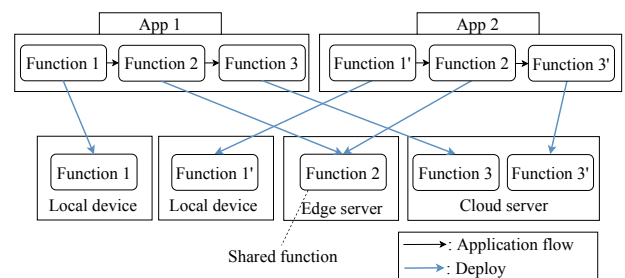


図1. サービスファンクションの展開, 共有およびスケールリングの概要図[2].

3.2. サービスファンクション最適割り当て手法

次に, サービスファンクションをネットワークノード内に最適割り当てする手法について, 最適化問題への定式化として扱い検討する. 検討手法では, センサデータを取得し始めてから, アプリケーションの処理

結果がデータを必要とする場所に届くまでの実行遅延時間が最小になることを目的に、サービスファンクションを最適に割り当てるノードを決定する。

ここで取り扱うサービスとして、センサデータを取得し、複数の処理を実行後、最終的に処理結果を必要としているユーザへデータを提供することを想定する。この想定のもと、センサデータを取得するノードをスタートノード、処理するノードを処理ノード、データを受け取るノードをエンドノードとして定義する。図2にノードの種類とデータの流れを示す。また、単純化のため、スタートノードで処理するための計算資源はあらかじめ決定しているものとする。一般的に、スタートノードはセンサデバイスに該当するが、センサデバイスでの処理を考える場合は、スタートノードと処理ノードが同一である場合も想定する。また、エンドノードでは、データ受信にかかる処理時間は0とみなし、処理ノードがエンドノードと同一である場合も想定する。

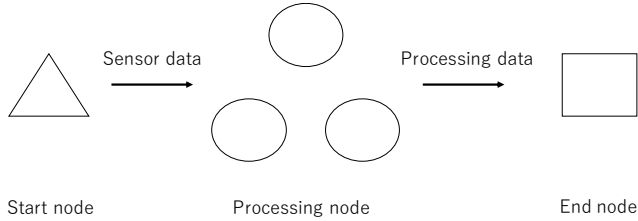


図2. ノードの種類とデータの流れ。

あるアプリケーションを形成するためのサービスファンクションの集合を $F=\{1,2,3,\dots,m\}$ 、ネットワーク内に存在し、それらのサービスファンクションを処理可能な処理ノードの集合を $N=\{1,2,3,\dots,n\}$ と定義する。また、 i 番目のサービスファンクションを処理ノード j で実行する場合の処理時間を $t_{i,j}$ 、 i 番目のサービスファンクションへの入力データサイズを D_i 、処理ノード j から処理ノード k への通信帯域(スループット)を $bw_{j,k}$ と定義する。単純化のため、これらのパラメータは事前に与えられるものとする。これらのパラメータのもと、サービスファンクションの最適割り当て問題を、次の目的関数および制約条件を設定し、0-1 整数計画問題として、最適化問題へ定式化する。

目的関数：

$$\min \sum_{i \in F} \sum_{k \in N} \left(t_{i,k} \alpha_{i,k} + \sum_{j \in N} \left(\frac{D_i}{bw_{j,k}} \beta_{i,j,k} \right) \right) \quad (1)$$

制約条件：

$$\alpha_{i,k}, \beta_{i,j,k} \in \{0,1\}, (i,j,k \in N) \quad (2)$$

$$\sum_{k \in N} \alpha_{i,k} = 1, (i \in N) \quad (3)$$

$$\sum_{i \in F} \beta_{i,j,k} = 1, (j,k \in N) \quad (4)$$

$$\sum_{j \in N} \beta_{i,j,k} - \alpha_{i,k} = 0 \quad (i \in F, k \in N) \quad (5)$$

$$\sum_{i \in N} \beta_{i+1,k,l} - \alpha_{i,k} = 0 \quad (i \in F, k \in N) \quad (6)$$

$$\beta_{1,j,k} = 0 \quad (j,k \in N) \quad (7)$$

$$\alpha_{1,1} = 1 \quad (8)$$

$$\alpha_{m,n} = 1 \quad (9)$$

目的関数である式(1)の第1項は処理時間を表し、第2項は通信時間を表している。すなわち、制約条件(2)~(9)のもと、処理時間と通信時間の組み合わせを最適化することとなる。式(2)の制約条件は、処理ノード j で処理あるいは通信を行うかどうかを表し、処理 (α)、通信 (β) を行う場合は1、それ以外は0を示す。式(3)は、 i 番目のサービスファンクションを必ずある処理ノードで実行することを表す。式(4)は、 i 番目のサービスファンクションには1回だけ通信が発生すること(処理ノード j から k への転送)を表す。式(5)は、あるサービスファンクションを、ある処理ノードで処理する場合は、前の処理を行った処理ノードから1回だけ通信を行うことを表す。式(6)は、次のサービスファンクションは現在の処理ノードから通信できる処理ノードで実行することを表す。式(5)と式(6)によって、サービスファンクションのチェイニングを保証することとなる。また、ここでは1つ前の処理を行ったノードからの通信を考えているため、1番目のセンサデータ取得のサービスファンクションでは通信を考えない。よって式(7)の制約条件が成り立つ。そして、センサデータ取得のサービスファンクションは1番目のノード(スタートノード)で行うと仮定すると、(8)の制約条件、また、最後のサービスファンクションはノード n (エンドノード)で行うと仮定すると、(9)の制約条件が加わる。

4. シミュレーション評価

本章では、3章で述べた最適割り当て手法を用いて、サービスファンクションを割り当てるべき最適なノードを算出し、その最適解における実行遅延時間を評価する。比較対象として、サービスファンクションをネットワーク内のノードにランダムに配置する場合を想定する。

シミュレーションは Python を用いて実装し、最適化問題の実装には PuLP[13]を用いた。今回想定したネットワークのトポロジーを図3に示す。スタートノードとしてセンサデバイスが1つ、エンドノードとしてクラウド内のノードを1つ想定し、その間に処理ノードと考えるエッジコンピューティングとしての計算資源

が存在することとする。この時、エッジノード（処理ノード）数はパラメータとして扱う（図 3 は、3 つの場合の例）。今回はセンサデバイスとエッジノード間にはすべて通信可能であるメッシュ型のトポロジーを想定する。また、センサデバイスとクラウドでの処理は考えないものとする。

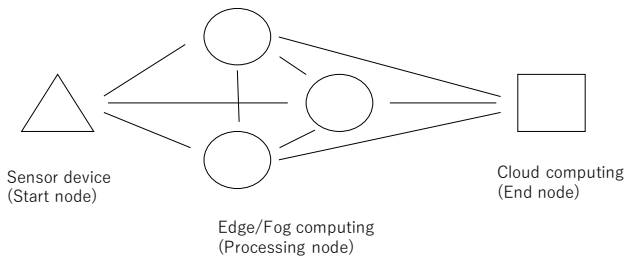


図 3. ネットワークトポロジー(処理ノード数が 3 つの場合)

シミュレーションを行う上で、今回設定したパラメータを表 1 に示す。各ノードにおける各サービスファンクションの処理時間は 0.1~3.0[s]の範囲で一様分布に従う乱数で発生させ、各ノード間のスループットは 50~100[Mbps]の範囲で 5[Mbps]ごとに一様分布に従う乱数で発生させるものとする。なお、処理ノードとクラウド間のスループットは共通で 100Mbps とする。データサイズは同一アプリケーションのサービスファンクションチェーン中では変化しないと仮定し、今回は 2K 解像度の PNG 画像の平均的なデータサイズの 2.5MB として設定した。また、処理ノードにおいて、特定のサービスファンクションが実行できないことがわかっている場合、処理時間のパラメータを十分大きな値に設定して対応する。また、ある 2 つのノード間の通信が許可されていない場合は、通信帯域のパラメータを十分に小さな値に設定し、同一ノードの通信に該当するパラメータには十分大きな値を設定し通信時間はかからないものとして考える。

表 1 シミュレーションパラメータ

サービスファンクション数	3, 5, 10
ノード数	3, 5, 10
処理時間	0.1~3.0 [s]
ノード間スループット (クラウド-ノード間は 100Mbps)	50~100 [Mbps] (5Mbps ごと)
入力データサイズ	20 [Mbit]

以上の条件のもと、ネットワークトポロジー、サービスファンクションを生成し、最適割り当て手法とランダム配置手法との実行遅延時間（end-to-end 時間）を比較評価する。ランダム配置の場合、10 回シミュレーションを実行し、その平均値と分散を示す。図 4 に、

処理ノード数を変化させたときの実行遅延時間の比較結果を、図 5 にその際のランダム配置に対する最適化による実行時間の削減割合をそれぞれ示す。

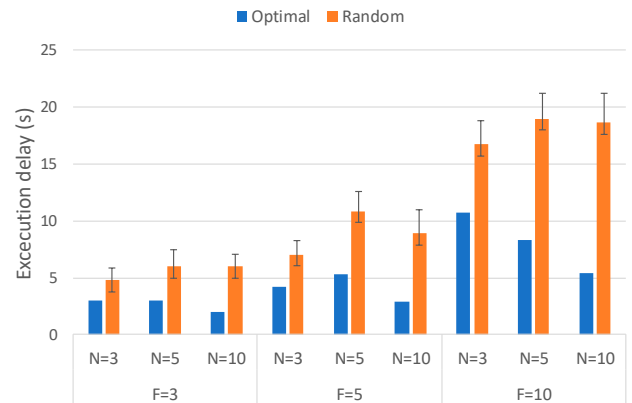


図 4. 処理ノード数を変化させた場合の実行遅延時間の比較

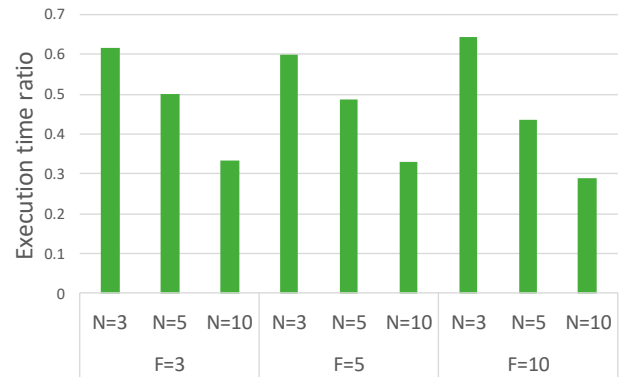


図 5. 処理ノード数を変化させた場合のランダム配置に対する最適解の実行遅延時間の削減割合。

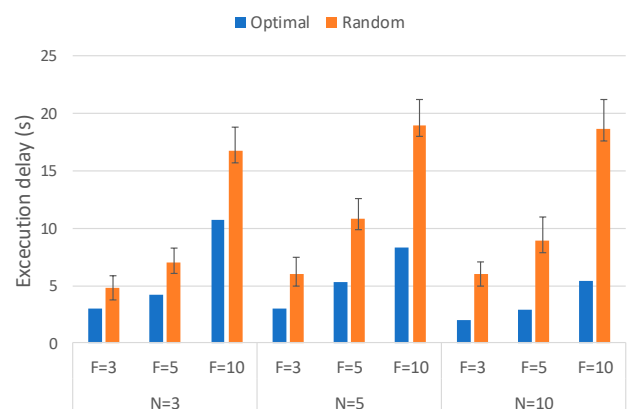


図 6. サービスファンクション数を変化させた場合の実行遅延時間の比較。

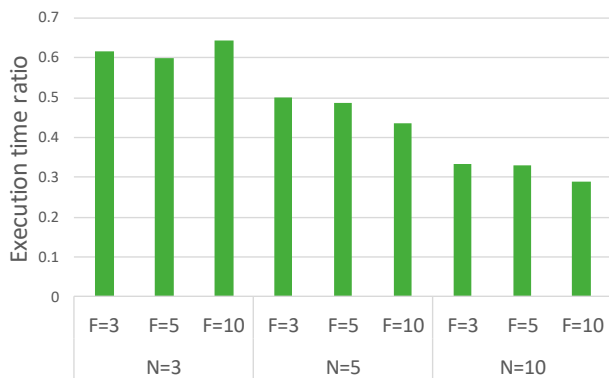


図 7. サービスファンクション数を変化させた場合のランダム配置に対する最適解の実行遅延時間の削減割合。

次に、図 6 にサービスファンクション数を変化させたときの比較結果を、図 7 にその際のランダム配置に対する最適化による実行時間の削減割合をそれぞれ示す。以上の結果より、サービスファンクション数および処理ノード数がどの条件でも最適解の実行遅延時間が大きく減少していることがわかる。したがって提案手法は有効であると言える。特に処理ノード数が増加したときの効果が大きいことがわかる。

5. まとめ

本稿では、IoT デバイス仮想化技術に適用するための、サービスファンクション最適割り当て手法を検討した。検討手法に関してシミュレーション評価により、ランダム配置と比較して、実行遅延時間が大きく減少することを確認した。

今後は、今回の検討手法のさらなる拡張として、複数アプリケーションや処理の共有などの条件を考慮していく予定である。また、処理時間だけでなく処理精度といった、アプリケーション要件も考慮していくことで、より現実的な最適割り当て手法を検討する。さらに、実機環境によって評価を行い、実システムへの適用を図る。

謝 辞

本研究成果は、戦略的情報通信研究開発推進事業（国際標準獲得型）「スマートシティアプリケーションに拡張性と相互運用性をもたらす仮想 IoT-クラウド連携基盤の研究開発（Fed4IoT）」および総務省委託研究 研究課題 VI 「IoT 機器増大に対応した有無線最適制御型電波有効利用基盤技術の研究開発」技術課題ア「有無線ネットワーク仮想化の自動制御技術」の支援を受けている。

文 献

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, et al., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", IEEE Communications Surveys & Tutorials, vol.17, no.4, 2015.
- [2] K. Ogawa, K. Kanai, J. Katto, et al., "IoT Device Virtualization for Efficient Resource Utilization in Smart City IoT Platform", IEEE Percom 2019, Mar. 2019.
- [3] P. Wang, C. Yao, Z. Zheng, et al., "Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems", IEEE Internet of Things Journal, vol.6, no.2, April. 2019.
- [4] Y. Sahni, J. Cao, L. Yang, "Data-Aware Task Allocation for Achieving Low Latency in Collaborative Edge Computing", IEEE Internet of Things Journal, vol.6, no.2, April. 2019.
- [5] ETSI GS NFV 001: "Network Functions Virtualization (NFV); Use Cases", [online]: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf
- [6] A. M. Medhat, T. Taleb, A. Elmangoush, et al., "Service Function Chaining in Next Generation Networks: State of the Art and Research Challenges", IEEE Communications Magazine, vol.55, no.2, Feb. 2017.
- [7] B. Kar, E. H. K. Wu, Y. D. Lin, "Energy Cost Optimization in Dynamic Placement of Virtualized Network Function Chains", IEEE Transactions on Network and Service Management, March. 2018.
- [8] D. Li, P. Hong, "Virtual Network Function Placement Considering Resource Optimization and SFC Requests in Cloud Datacenter", IEEE Transactions on Parallel and Distributed Systems, July. 2018.
- [9] L. Gu, D. Zeng, S. Tao, "Fairness-Aware Dynamic Rate Control and Flow Scheduling for Network Utility Maximization in Network Service Chain", IEEE Journal on Selected Areas in Communications, May. 2019.
- [10] K. Kaur, T. Dhand, N. Kumar, and S. Zeadally, "Container-as-a-Service at the Edge: Trade-off between Energy Efficiency and Service Availability at Fog Nano Data Centers", IEEE Wireless Communications, vol.24, no.3, June. 2017.
- [11] M. Amaral, J. Polo, D. Carrera, et al., "Performance of Microservices Architectures Using Containers", IEEE 14th International Symposium on Network Computing and Applications, Sep. 2015.
- [12] J. Rufino, M. Alam, J. Ferreira, et al., "Orchestration of containerized microservices for IIoT using Docker", IEEE International Conference on Industrial Technology, Mar. 2017.
- [13] Optimization with PuLP, [online]: <https://pythonhosted.org/PuLP/>