

オープンIoT時代に向けたヘテロジニアスリソースの最適化

Optimizing Heterogeneous Resources for Open IoT Era

山 登 庸 次[†], 干 川 尚 人[†], 野 口 博 史[†],
出 水 達 也[†], 片 岡 操[†]

Yoji Yamato[†], Naoto Hoshikawa[†], Hirofumi Noguchi[†], Tatsuya Demizu[†] and Misao Kataoka[†]

あらまし 私達は、オープンIoTに向け、ユーザが必要なデータを持つデバイスをオンデマンドに発見し利用する、Tacit Computing 技術を提案し、その要素技術を実装してきた。本稿では、Tacit Computing で発見した、ユーザに必要なデータを持つデバイスを継続的に利用するサービスとするため、運用コスト低減や性能向上を行うための3レイヤーの最適化を提案する。最適化では、デバイス、ネットワーク、クラウドレイヤーで、適切な機能配置等を、本格運用前に行う。

キーワード: IoT, Tacit Computing, オープンIoT, マルチレイヤー, 最適化, ユーザコンテキスト

1. ま え が き

近年、IoT (Internet of Things) 技術が発展しており(例えば、¹⁾⁻⁴⁾)、数多くのIoT 機器がネットワークに繋がってきている。しかし、現状、IoT の適用は、特定のターゲットに合わせた垂直統合的なワンオフソリューションになりがちである。この結果、個々のIoT サービスのコストが高くなり、多彩なサービスが十分出てきていない現状がある。

多彩なIoT サービスをコスト低く開発、運用するためには、センサやアクチュエータ等のデバイスとサービスを分離し、水平分離的にデバイスとサービスを相互に利用可能にすることが必要となる。このような考えはオープンIoT⁵⁾と呼ばれ、今後のIoT サービス活性化に重視されている。

私達は、このようなオープンIoT に向けて、サービスからデバイスを自由に利用するための仕組みとして、Tacit Computing 技術を提案している⁶⁾。Tacit Computing は、デバイスが今持つライブデータに基づき、ユーザが必要なデータを持つデバイスをオンデマンドに発見し、利用することを可能にする技術である。

しかし、その時点の状況に合わせてデバイスを発見し、利用するだけでは、ユーザのニーズには一時しか答えていない。ユーザのニーズに応え続けるサービスとするためには、ユーザに必要なデータを持つデバイスを継続的に利用することをリーズナブルな価格で提供することが必要である。そこで、本稿では、Tacit Computing で発見利用するデバイスを継続的に利用したサービスとするため、運用コスト低減や性能向上を行う3レイヤーの最適化を提案する。

最適化では、デバイス、ネットワーク、クラウドレイヤーで、適切な機能配置やオフロードを、本格運用前に行う。

2. Tacit Computing の概要

Tacit Computing (図1) では、クラウドレイヤー、ネットワークレイヤー、デバイスレイヤーの3層から、ユーザに適切なリソースを発見、連携することで、ユーザのリクエストに応える。3層の構成であるが、時々刻々変化する状況に対応するため、現場に近いデバイスレイヤーでできるだけ処理を行うのがコンセプトである。低レイヤーで処理を終えることで、ネットワークトラフィックの削減や、プライバシー性が高いデータの流出を抑えることができる。

本節では、特にオープンIoT 時代のキーとなる、ライブデータ検索技術とデバイス仮想化技術について述べる。

ライブデータ検索技術は、ユーザにとって必要なデータを提供するデバイスを検索するための技術である。定点のカメラに映った人物に対して、情報案内や警告アラート等を行うような例では、数秒程度しかカメラに人物は映らないし、また、そのカメラの映像しかその人物には意味がないデータと言える。このように、デバイスレイヤーで発生し、時々刻々変わるようなデータをライブデータと呼ぶ。そこで、ユーザにとって必要なライブデータを検索するため、クラウドレイヤーにデータが上がって来るのを待つのではなく、下位レイヤーに分析する機能を配信することを、ライブデータ検索技術では行う。例えば、箱根駅伝予選会に友人が出ており、友人が映ったカメラの映像を自動で繋いで欲しいとする。この場合、友人のゼッケン番号を検索キーに、リクエストをすると、Tacit Computing では、カメラを収容するゲートウェイやネットワークエッジに、OpenCV等の画像を分析する機能を配信して、カメラに近い場所で映

2018年7月11日受付, 2019年3月8日再受付, 2019年5月24日採録

[†] 日本電信電話株式会社 NTT ネットワークサービスシステム研究所
(〒180-8585 東京都武蔵野市緑町 3-9-11, TEL 0422-59-5744)

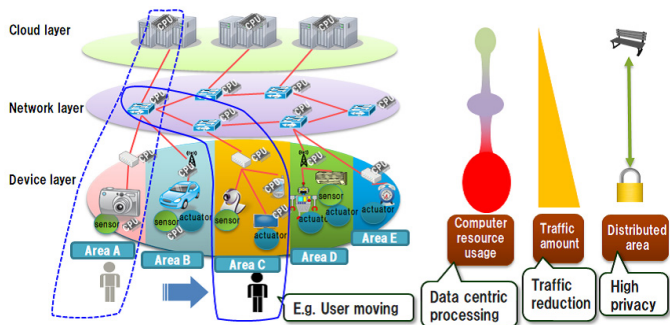


図 1 Outline of Tacit Computing

像を分析することで、友人のゼッケン番号が画像分析で抽出され、友人が映っているカメラを特定する。

次に、利用したいデバイスが特定された場合に、そのデバイスを利用する必要がある。IoT デバイスは、多数のメーカーが開発しており、利用時のプロトコルやインターフェース、アドレス等が、デバイス毎に異なる。そこで、デバイス仮想化技術によって、個々のデバイスのインターフェース等の違いを吸収している。例えば、上記例であれば、利用方法はカメラ毎に異なるが、Tacit Computing では、カメラ映像の取得のような共通的なリクエストを元に、カメラを収容するゲートウェイ等で、デバイス毎のアダプタにて変換を行い、個々のカメラに応じたリクエストを行う。このようにすることで、ユーザはデバイス個々の違いを意識せず、デバイスの利用が可能である。また、デバイス利用時は Semantic を用いて抽象化する Semantic Web Services 等の既存技術も利用できる（例えば、⁷⁾⁻¹¹⁾）。

3. 本稿で取り組む課題

Tacit Computing の利用例として、追跡カメラを考える。追跡カメラは、小学生等の子供の、学内や登下校中の映像を、子供近傍のカメラから取得し、親の携帯端末で見ると利用形態である。追跡カメラで、親が子供の安全を映像で確認することはその時点の親のニーズは満足できるが、それを 1 回幾らという課金とした場合に受け入れづらいと考える。むしろ、子供近傍のカメラから映像を取得し続けるが、親は携帯端末で見たい時は子供の映像を見られ、見ていない時にも機械学習等で見守り、登下校中に不審者が近づいた等の異常時は親にアラートを送る等の、継続的なサービスとすることで、月額課金でも受け入れられると考える。

追跡カメラの処理としては、Tacit Computing で、子供が映っているカメラを発見し、親のリクエスト時はそのカメラの映像を親の携帯端末に届けるとともに、見守りのため、カメラを収容する GW (GateWay) またはネットワークのエッジの SSE (Subscriber Service Edge) 等に画像処理をする機能 (OpenCV ライブラリ¹²⁾ 等) を配置し、子供が映っているカメラの映像から画像を切り出し分析を行う。画像分析した結果は、特徴ベクトルにサマライズされた後集約され、クラウド技術を用いて（例えば、¹³⁾⁻²⁰⁾）、不審者が

近づいている等の異常値分析を機械学習等の手法 (LOF²¹⁾ 等) を用いて行い、異常がある場合は親にアラートする。

追跡カメラ自体は、カメラと機械学習等を使った SI サービスで作りこむこともできるため、「Tacit Computing で発見利用するデバイスを継続的に利用したサービスをリーズナブルに提供すること」を、課題と設定する。

4. 3 レイヤーでの最適化提案

最適化は、各レイヤーでの処理と共に、レイヤー間で機能配置を変更するレイヤーを跨ぐ処理がある（図 2）。

デバイスレイヤーでは、ユーザの要望に合うデバイスへの切替を随時行っていくことが必要である。追跡カメラであれば、子供の場所に応じて子供の映像が映っているカメラを選択していく。この時、ネットワークに繋がる全カメラで画像解析をし、子供が映っているカメラだけ使うのでは、非常にコストが高くなってしまいうため、子供に何らかの識別子を振り、カメラメタデータを用いて、その識別子の子供近傍のカメラを絞り込んでいく処理が必要となる。

更に、デバイスを収容するゲートウェイ群の中からどのゲートウェイに分析機能を配置するかが、コストに大きく影響する。追跡カメラであれば、OpenCV 等の画像分析機能が使われるが、よく子供が通るエリアのゲートウェイには画像分析機能を配置し、そのエリアを超えて移動する場合は、画像分析機能を配信して処理する等を検討している。

ネットワークレイヤーでは、SSE 等のネットワークエッジへの機能配置を効率化する。分析の様々な機能を全エッジに置くことは、設備コスト上問題となるため、収容エリアの利用頻度が高い分析機能を、エッジに配置し、あまり使われない機能はリクエストがあった際に配信する。また、迷子検知と不審者検知のように、人物が映っている映像だけ分析すればよいような場合は、前処理として人物検知を集約して行い、後処理は人物映像だけ分析する等の処理分割により、分析用の計算量を低減する。ネットワークでのデータや機能のキャッシュや配置については、情報指向ネットワーク²²⁾ 等で検討がされており、それらの適用も含めて検討する。また、サービスによっては、ネットワークで、帯域確保や優先制御等が必要となる場合もある。

クラウドレイヤーでは、OpenStack²³⁾ 等で構築したクラウドのどこで処理を行うかが、コストや性能に大きく影響する。利用頻度が多いデバイスを収容するネットワークエッジとの遅延が小さい DC (Data Center) のクラウドに処理機能をデプロイする。更に、クラウドリソースのサイズも運用コストに影響があるため、そのサイズも最適化対象である。例えば、追跡カメラであれば、サマライズされた特徴ベクトルを、機械学習などの手法で分析し、不審者のアラート等を行うことがクラウドでの処理で考えられるが、Storm²⁴⁾、Spark²⁵⁾ 等のストリーム処理機能で処理するための適切なリソースサイズを確保する。

また、クラウドでは、近年、GPU や FPGA 等のヘテロ

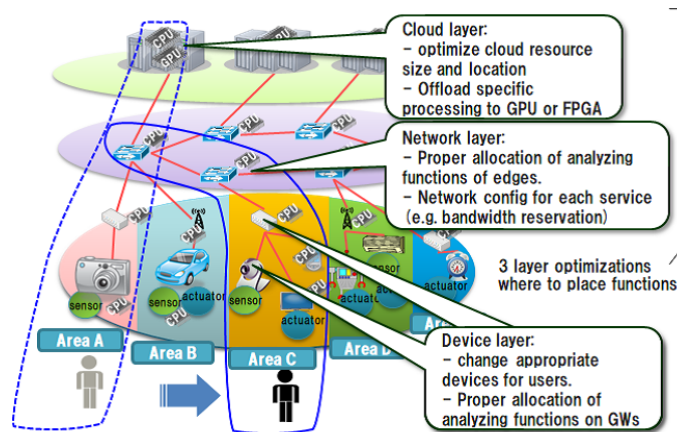


図 2 Three layer optimizations

ニアスな HW を備えたサーバが増えている。それらを活用し、行列計算等を GPU にオフロードしたり、検索等特定処理を FPGA にオフロードしたりする²⁶⁾ことで、高性能化が可能である。ヘテロニアス HW に処理オフロードする際は、2つの方式が考えられる。第一は、行列計算等、特定のロジックを、GPU や FPGA にオフロードするパターンを事前に DB 等に準備しておき、ユーザからのリクエストに当該処理が含まれる場合は DB からオフロードする処理記述 (例えば CUDA²⁷⁾, OpenCL²⁸⁾ 等) を抽出しその設定を行う形である。第二は、処理ロジックの繰り返し文を GPU にオフロードする最適化をその場で行う形である。GPU での高速化には、適切な領域の並列化が必要であり、自動で適切な領域を抽出するため、遺伝的アルゴリズム²⁹⁾等の手法を用いて、繰り返し文の並列可否を検証環境で試行反復し、抽出された最適並列領域を OpenACC³⁰⁾や PGI コンパイラ³¹⁾等を用いてオフロード設定する⁶⁾。

更に、ここまで、デバイスレイヤー、ネットワークレイヤー、クラウドレイヤー個々について述べていたが、サービスの利用形態によって、どこに機能を配置するのが適切かは変わってくる。そこで、各処理を、各レイヤーで処理する場合の、コスト (リソース確保のためのサーバコスト、発生トラフィックに基づくネットワークコスト) と性能 (スループット、遅延) をモデル化し、どこに配置するのが適切かを計算することを検討している。ユーザ向けの価格・コストを制約条件として性能の最適化をするため、組合せ最適化問題になるため、限られた最適化期間中にある程度の解は発見できるような短時間での手法で計算を行う。

5. む す び

本稿では、オープン IoT に向け Tacit Computing 技術の次ステップとして、3レイヤーでの最適化処理を、カメラ映像分析を例に検討し、提案した。Tacit Computing で発見した、ユーザに必要なデータを持つデバイスを継続的に利用するサービスとするため、運用コスト低減や性能向上によるリーズナブルな提供が必要であり、適切な機能配

置や処理オフロードを、本格運用前に行う。今後は、本稿提案の最適化の詳細評価を行い、別稿で報告する。

〔文 献〕

- 1) M. Hermann, et al., "Design Principles for Industrie 4.0 Scenarios," Working Draft, Technische Universität Dortmund, (2015)
- 2) Y. Yamato, et al., "Predictive Maintenance Platform with Sound Stream Analysis in Edges," Journal of Information Processing, **25**, (2017)
- 3) P. C. Evans, et al., "Industrial Internet: Pushing the Boundaries of Minds and Machines," Technical report of GE, (2012)
- 4) AWS IoT Platform website, https://aws.amazon.com/iot/how-it-works/?nc1=h_ls
- 5) Tron project web site, <http://www.tron.org/>
- 6) Y. Yamato, et al., "Automatic GPU Offloading Technology for Open IoT Environment," IEEE Internet of Things Journal, (2018)
- 7) Y. Yamato, et al., "Context-Aware Service Composition and Component Change-over using Semantic Web Techniques," IEEE ICWS 2007, (2007)
- 8) Y. Yamato, "Context-aware Ubiquitous Service Composition Technology," IFIP CONFENIS 2006, (2006)
- 9) Y. Yokohata, et al., "Service Composition Architecture for Programmability and Flexibility in Ubiquitous Communication Networks," IEEE SAINT Workshops 2006, (2006)
- 10) H. Sunaga, et al., "Service Delivery Platform Architecture for the Next-Generation Network," ICIN2008, Oct. (2008)
- 11) Y. Yamato, "Ubiquitous Service Composition Technology for Ubiquitous Network Environments," IPSJ Journal, **48**, 2, pp.562-577, Feb. (2007)
- 12) OpenCV web site, <http://opencv.org/>
- 13) Y. Yamato, "OpenStack Hypervisor, Container and Baremetal Servers Performance Comparison," IEICE Commun. Express, **4**, (2015)
- 14) Y. Yamato, "Optimum Application Deployment Technology for Heterogeneous IaaS Cloud," Journal of Information Processing, **25**, (2017)
- 15) Y. Yamato, "Server Selection, Configuration and Reconfiguration Technology for IaaS Cloud with Multiple Server Types," Journal of Network and Systems Management, Springer, Aug. (2017)
- 16) Y. Yamato, "Performance-Aware Server Architecture Recommendation and Automatic Performance Verification Technology on IaaS Cloud," Service Oriented Computing and Applications, Springer, (2016)
- 17) Y. Yamato, "Automatic verification technology of software patches for user virtual environments on IaaS cloud," Journal of Cloud Computing, Springer, 2015, 4:4, Feb. (2015)
- 18) Y. Yamato, "Cloud Storage Application Area of HDD-SSD Hybrid Storage, Distributed Storage and HDD Storage," IEEE Trans. on Electrical and Electronic Engineering, **11**, (2016)
- 19) Y. Yamato, "Use case study of HDD-SSD hybrid storage, distributed storage and HDD storage on OpenStack," IDEAS'15, (2015)
- 20) Y. Yamato, "Fast and Reliable Restoration Method of Virtual Resources on OpenStack," IEEE Trans. Cloud Comput., (2015)
- 21) Markus M. Breunig, et al., "LOF: Identifying Density-based Local Outliers," SIGMOD 2000, (2000)
- 22) G. Xylomenos, et al., "A survey of information-centric networking research," IEEE Communications Surveys & Tutorials, (2014)
- 23) O. Sefraoui, et al., "OpenStack: toward an open-source solution for cloud computing," International Journal of Computer Applications, **55**, (2012)
- 24) N. Marz, "STORM: Distributed and fault-tolerant realtime computation," (2013)
- 25) M. Zaharia, et al., "Spark: Cluster computing with working sets," 2nd USENIX Conf. on Hot Topics in Cloud Computing, (2010)
- 26) A. Putnam, et al., "A reconfigurable fabric for accelerating large-scale datacenter services," ISCA'14, (2014)
- 27) J. Sanders, et al., "CUDA by example : an introduction to general-purpose GPU programming," Addison-Wesley, (2011)
- 28) J. E. Stone, et al., "OpenCL: A parallel programming standard for heterogeneous computing systems," Computing in science & engineering, **12**, 3, pp.66-73, (2010)
- 29) J. H. Holland, "Genetic algorithms," Scientific american, **267**, 1, pp.66-73, (1992)
- 30) S. Wienke, et al., "OpenACC-first experiences with real-world

applications,” Euro-Par 2012 Parallel Processing, (2012)

- 31) M. Wolfe, ”Implementing the PGI accelerator model,” ACM GPGPU-3, pp.43-50, Mar. (2010)



やまと ようじ
山登 庸次 2000 年，東京大学理学部卒，2002 年，同大学院理学系研究科修了，2009 年，同大学院総合文化研究科にて博士（学術）。2002 年，日本電信電話株式会社入社。同社にて，IoT プラットフォーム，クラウドコンピューティング，サービス連携技術研究開発に従事。現在，同社ネットワークサービスシステム研究所特別研究員。2018 年，第 33 回電気通信普及財団賞（テレコムシステム技術賞）受賞，IEEE，電子情報通信学会シニア会員，情報処理学会会員。



ほしかわ なおと
干川 尚人 2009 年，名古屋大学大学院情報科学研究科博士課程中退。同年，東日本電信電話株式会社入社。2013 年，日本電信電話株式会社ネットワークサービスシステム研究所にて，災害用伝言システムの開発や IoT ネットワークサービス基盤の研究に従事。2018 年から小山高専 電気電子創造工学科助教。2019 年より講師。「安心・安全に利用できる IoT ネットワークシステム」をテーマとして研究に取り組んでいる。電子情報通信学会，情報処理学会会員。博士（情報科学）。



のぐち ひろふみ
野口 博史 2010 年，早稲田大学理工学部卒，2012 年，同大学院創造理工学研究科修了，同年，日本電信電話株式会社入社。同社にて，分散処理ミドルウェア，仮想化技術，IoT プラットフォームの研究開発に従事。電子情報通信学会員，修士（工学）



でみず たつや
出水 達也 2011 年，大阪大学基礎工学部卒。2013 年，東京大学大学院工学系研究科修了。同年，日本電信電話株式会社入社。同社ネットワークサービスシステム研究所にてキャリアネットワークのアーキテクチャ，IoT デバイスの広域通信手法の研究に従事。現在，仮想化技術をキャリア網へ適用したサービス提供基盤の研究開発に取り組んでいる。電子情報通信学会員。修士（工学）。



かたおか みさお
片岡 操 2012 年，京都大学工学部卒，2014 年，同大学院情報学研究科修了。2014 年東日本電信電話株式会社入社。同社にてネットワークシンプル化に従事。2016 年から現在にかけて，日本電信電話株式会社にて分散処理基盤および IoT 基盤の研究開発に従事。