

複数ウェアラブルデバイスにおける分散処理 環境の研究

Yokokawa, Shohei / 横川, 翔平

(出版者 / Publisher)

法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)

13

(開始ページ / Start Page)

1

(終了ページ / End Page)

4

(発行年 / Year)

2017-03-31

(URL)

<https://doi.org/10.15002/00021535>

複数ウェアラブルデバイスにおける分散処理環境の研究

Research on the Distributed Computing Environment of Multi Wearable Devices

横川 翔平

Shohei Yokokawa

法政大学大学院情報科学研究科情報科学専攻

E-mail: 16t0023@cis.k.hosei.ac.jp

Abstract

By the advent of smart phones and wearable devices, lots of the human activity data can be obtained from these devices. It is also possible for a user to carry and wear multiple devices that can be regarded as the extensions of human natural organs. And One of Distributed Computing named "Fog Computing / Edge Computing" appeared recently. These Concepts enables sensors to offload processing onto distributed computing resource. However, there are various problems in the introducing Fog/Edge computing into a multi-wearable system. It is unpredictable what wearables require and where they are. So, it is difficult to design and plan effective data flow and data management architecture. Furthermore, it cannot be ignored that there are various differences among computing environments such as processor, operating system, software language, etc. This study considers how to construct data flow and resolve the differences among various kinks of devices and environments. An architecture combining cloud and fog computing has been proposed to manage various wearable devices as well as their data communication and processing.

1. まえがき

近年、ウェアラブルデバイスのようなユビキタスデバイスが普及し始めている。iWatch, Android Wear に代表されるスマートウォッチを初めとした、これらの機器は使用者の生体情報を取得し、ヘルスケアのサービス等を提供する。しかしながら、ウェアラブル機器においては、データの活用に関して、問題を起こしている。各デバイスからセンサデータの収集を行う場合、それぞれに対して、定められた言語・SDK によってプログラミングをしなければならず、研究者、もしくはサービス提供者が、各デバイスに対しての知識を要求される。そのため、本研究室ではこれらを解決するための Wear-I というシステムの研究を行っている。

他方、ユビキタスコンピューティングの研究では、分散コンピューティングが注目を浴びている。分散コンピューティングの一種として、“Fog Computing” が Cisco を初めとした業界団体から提唱されている。これらは今ま

で取得したデータの処理が 1 カ所に集められて処理されていたものを、転送路上に設置された計算資源によって処理を行うコンセプトである。

この研究では分散コンピューティングを用いた、データの転送、活用を Wear-I にて実現するため、Wear-I や Fog Computing の発想を用い、ウェアラブルデバイス環境に分散処理の導入を目指す。

第 2 章ではこれらに関連する研究やサービス、製品について説明する。また、Fog Computing を Wear-I に導入するに当たって、考えられる問題を検討する。第 3 章では、これを解決するシステムの概要を示し、Application Management System や Calculation Node について説明する。4 章ではこれらを実装する技術について説明し、5 章にて考察を行う。

2. 研究背景

ウェアラブルデバイスを始めとした、ユビキタスコンピューティングに関する研究の振興に伴い、デバイスそのものではなく、それらを支えるインフラストラクチャにも注目が向けられるようになった。日本における光回線の輻輳が問題になり始め、通信容量の削減が求められるであろう状況を鑑み、そこで、本研究室が持つコンセプトに分散処理を導入することを考えた。

2.1. Fog/Edge Computing

工場向けの製品として、小型の温度センサなどをいくつかの地点に配置し、空調システムの改善に役立てるなど、小型機器がもたらすデータは、活用例が多い。今までもあったものの、IP 化されていない機器が、IP 化によって恩恵を受けることもできる。例えば監視カメラは、IP 化によって遠隔からの監視を実現している。

こういった流れに対して、データの処理において問題が発生している。現在の小型機器では演算処理能力が不足し、データこそ取れても、その解析が行えない点がある。この場合は、監視元の機器において解析を行うことが考えられるが、通信環境の問題において困難な場合がある。また、解析すべきデータが 1 カ所に集中し、そのコンピュータのキャパシティを超えてしまう可能性がある。

大量の IoT 機器に対して、データの処理、保管等を分散して行う分散コンピューティングの一種として、

“Fog Computing”が Cisco を初めとした業界団体から提唱されている。これらを集中しがちな計算資源を地理的、論理的に分散し、通信の負荷を低減するコンセプトである。Open Fog コンソーシアムでは、これらに関するリファレンスアーキテクチャを制定する等を行っている。

またこれに類する物として“Edge Computing”が挙げられる [2]。この研究では、特に区別せず Fog/Edge Computing と表記する。

2.2. Wear-I

ウェアラブルデバイス毎に SDK がでている環境では、1つのデバイスに対してプログラムを作成し、データを利用するアプリケーションを作成したとしても、アプリケーションが想定したデバイスが使用されていない場合には、当然このアプリケーションは使用することが出来ない（アプリケーション中心）。

そして、利用者にとっても、利用しているデバイスに依存するアプリケーションを複数使っている場合、利用者の特定デバイスが使用できない場合にすべてのアプリケーションが利用できなくなる（デバイス中心）。このように、アプリケーションとデバイスの関係が固定されている現状では、このようなシステムを利用することに制約が生じている

このように、ウェアラブルデバイスに柔軟性や拡張性に問題が存在する。これらの問題を解決するために、本研究室では Wear-I というシステムの研究を行っている [3][4]。このシステムは、スマートウォッチをはじめ、多くの人々が様々な種類のウェアラブルデバイスを身体の様々な部位に装着する社会の到来を考え、それらのデバイスのデータを個人のパーソナリティを構成するデータとして、使用者が意図する限り、自由に扱うことを可能とする。

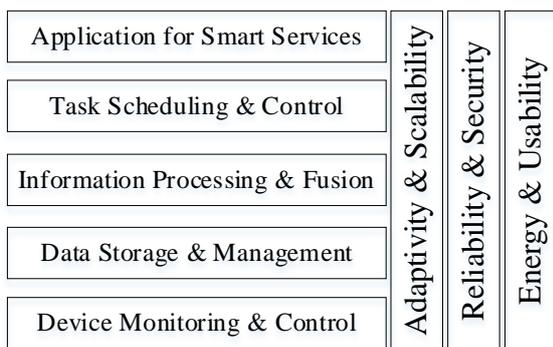


図 1. Wear-I Architecture

2.3. 関連研究・製品

Wear-I のデータ共有、活用やイベントによってウェアラブルデバイスを実現するサービスとして、“IFTTT”や“myThings”が存在する [5][6]。IFTTT/myThings は対応するデバイスにアプリケーションを導入し、特定の条件になった場合に、インターネットを通して、他のサービス・デバイスに対して通知を送信する等の機能を持つ。また、「Device Web API Manager」はスマートフォンに

サーバ機能を持ったアプリケーションを導入することで、スマートフォンをネットワークで取得可能なセンサとする機能である。これらを利用することによってデバイスとアプリケーションの関係を自由にし、様々なアプリケーションを開発することが可能になると考えられる。

今日では、Fog/Edge Computing を利用した研究や、製品が出回り始めている。Fog を提唱した Cisco Systems が “Cisco Kinetic” [7]、日本においては、インフォテリア株式会社の “Gravio” [8]等が挙げられる。

本研究室でもスマートフォン、ウェアラブルデバイス上のセンサデータをクラウド上に保存し、データを活用する研究を行っている [9]。

2.4. Wear-I における Fog/Edge Computing の導入

Fog Computing では管理用端末を通してそれぞれのノードに対して設定を送っている。工場等の施設において、固定センサやハンドヘルド端末のような、設置する機器等がすべて自己のネットワーク管理下における環境であるならば、あらかじめ処理が想定でき、大きな効果を発揮することが期待される。しかしながら、個人が所有するデバイスにおいては、あらかじめ処理を行うノードを指定することが困難であり、現状の Fog/Edge の発想では対処することができない。そのため、中央制御・事前設計によらないデータ処理の仕組みを検討する必要がある。

また、Fog/Edge Computing で想定される計算ノードは、ネットワークルータやスイッチに計算機能を付加した物から、Raspberry pi, GPU 搭載 x86 マシンなど、幅広く存在しうる。サービス提供者にとって、あらかじめ実行するノードが未確定であることは、データ処理プログラムを実行する環境がわからないことを意味し、コンピュータの差異をいかに吸収するかも課題となる。

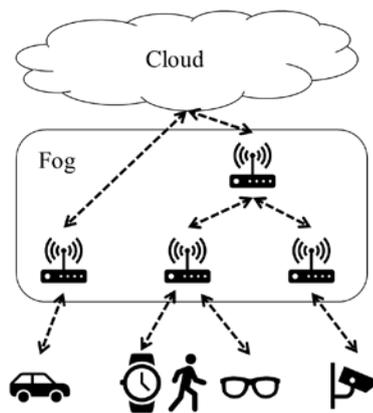


図 2. Fog/Edge Computing を用いたイメージ

3. システムの構造

本研究では、小型コンピュータや、仮想マシン、ワークステーション上に、処理を分散させる機能をもつシステムを設置し、ユーザが所持するウェアラブルデバイスやほかの機器から接続させる。デバイスは既に Wi-Fi でインターネットに接続されている場合や、Cellular による

接続のみの場合、Wi-Fi を拾えずに、システムにアクセスできない場合を想定して、Bluetooth ビーコンや GPS に基づいた情報を利用して、最寄りのシステムに接続できるようにする。

デバイスからは、取得したデータと、行うべき処理を指定してシステムに送信し、何段階(Phase)かの処理を経て、サービスを提供するサーバに送信、または、送信元の機器に処理結果を返送する機能を持つ。このような機能を実現するために、主に 2 種類のモジュール、Application Management System と Calculation Nodes で、構成される。以下の図で分散処理システムの概要を示す。

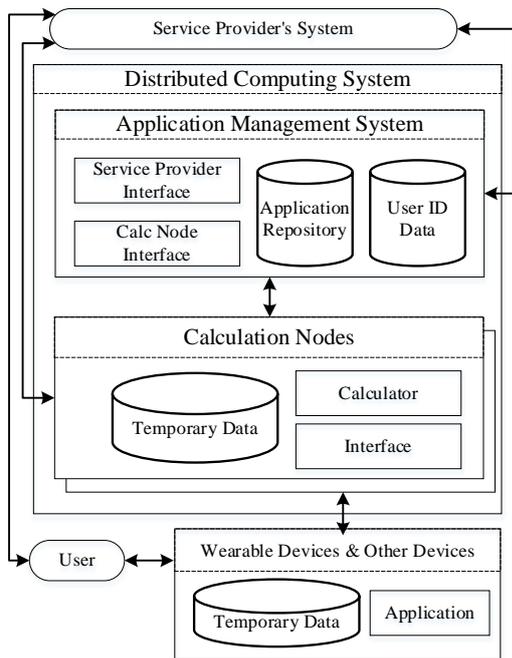


図 3. システムの構造

3. 1. Application Management System

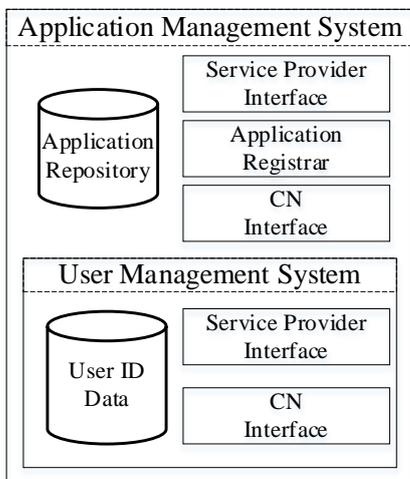


図 4. AMS の構造

Application Management System(AMS)は、このシステムを中心にあたる部分で、分散処理に必要な情報を司る。

サービス提供者は、データの加工に必要な処理を分割して記述し、それぞれに必要なプログラムを添付し、インタプリタや GPU など特定のハードウェアを必要とするかどうかを記述する。また、AMS は User を管理する User Management System をサブシステムとして持っている。UserID を格納する DB を所持していて、データの識別に使用される。

3.2. Calculation Node

Calculation Node(CN)は、分散される計算資源に当たる部分で、一般家庭に置かれるものから、データセンタに置かれるものまで、幅広く想定している。AMS において示された処理を行うモジュールである。CN は行ききれない処理が発生した場合に備えて、次に依頼すべき上位 CN のアドレスを保持している。

CN は、CN が持つ計算資源の情報(Processor のアーキテクチャ、OS、インストールされたインタプリタ)を取得、または設定し、Node Information に保存する。

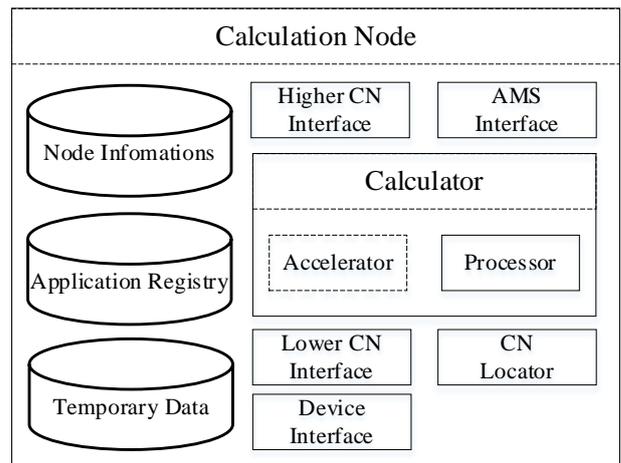


図 5. CN の構造

CN Locator は CN が持ちうる接続方法をアドバタイズするモジュールである。CN は CN Locator ユーザの機器から、TCP/IP や Bluetooth 等でアクセスされ、使用しているアプリケーションの通知を受ける。CN はその情報から処理の手順や必要なプログラム、DB のスキーマ等を AMS から取得する。そこで、自 CN で実行できる処理を実行する。すべての処理が実行できた場合には、その時点でアクセス元の機器か、サービス提供者に指定された場所に結果を送信する。HW 等の制約から、すべての処理が実行できない場合は、CN は上位 CN に対し、続きの処理を行うように依頼する。すべての CN を使用しても、処理が残っている場合は、サービス提供者に処理が依頼され、結果は来た道に戻る形でユーザの機器に提供される。

CNにはGPUのような外部 Accelerator ユニットが付くことを想定する。これによってGPUが必要な処理がある場合にGPUを持つCNまで転送することが期待される。

以下の図6で分散処理の流れを示す。ここでは仮に3段階で構成される分散処理があり、Phase1からPhase3までであるとする。サービス提供者はPhase毎に要求環境を定義しているため、Node Mでは、Node Informationに適合するPhase1のみの実行にとどめ、Phase2-3を上位CNであるNode Nに依頼している。

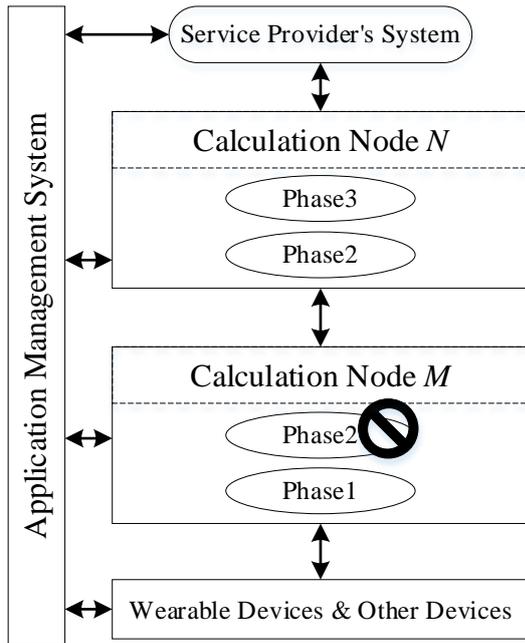


図6. 処理の流れ

4. 実装

3章において示したデータ処理分散機能のうち、主要なモジュールを実装する。CNはJavaアプリケーションとして実装し、システムコールにて外部プログラムを実行する。AMSは、Apache Webサーバを利用し、HTTPによってアプリケーションを取得できるようにする。主要なメタデータの構造は以下の通りである。これらのメタデータはMySQL及び、MySQL互換RDBMS、必要に応じてファイルシステムに直接格納される。

項目	形式
アプリケーションID	文字列
手順番号	非負整数
要求環境	文字列
プログラム本体	バイナリ
その他	

図7. アプリケーションレジストリの構造

項目	形式
CPU名	文字列
対応命令	リスト
ライブラリ	リスト
使用可インタプリタ	リスト
その他	

図8. CNのデータベースの構造

5. むすび

本研究では、ユーザが持ちうるウェアラブルデバイスや、その他のデバイスにおいて、ユーザが任意のアプリケーションを分散環境において実行するためのシステム構造の検討を行った。

今後の課題としては、不特定多数の利用者によって使用される計算資源に対して、セキュリティを確保することが挙げられる。CNにとっては、少なくない第三者のデータを同時に預かることになり、流出や悪用が無いことを保証する仕組みが必要になる。また、利用者の端末が偽のCNに接続する可能性も考えられるため、AMSに認証局機能を持たせ、証明書を発行することもセキュリティを高める手段の1つと考える。

また、ユーザが移動していることを考え、処理中にCNとユーザの機器との接続が切れた場合の処理（ハンドオーバー）を検討する必要がある。主に家庭等に設置されるルータのNAT等の問題で自由にNodeにアクセスすることができるとは限らないので、上位CNにデータを預ける等の対策が考えられる。

文 献

- [1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in Proc. 1st Edition MCC Workshop Mobile Cloud Comput., pp. 13–16, 2012.
- [2] Edge Computing https://en.wikipedia.org/wiki/Edge_computing
- [3] J. Ma and R. Huang, "Wear-I: A New Computing Paradigm in Wearable Computing", Keynote, in Proc. IEEE Conf. Ubiquitous Computing & Commun., Liverpool, U.K., pp. 1063–1068, 2015
- [4] 山田 真之, 馬 建華, "ウェアラブルシステムにおける動的に変化する複数のデバイスの管理機能の研究", 第78回全国大会講演論文集, 2016(1), pp.469–470, Mar. 2016.
- [5] IFTTT, <https://ifttt.com/>
- [6] myThings <http://mythings.yahoo.co.jp/>
- [7] Cisco Kinetic <https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-kinetic.html>
- [8] Gravio <https://www.gravio.com/>
- [9] D. Inoue and J. Ma, "Giraffe: A cloud-smartphone based hybrid database system for sensor data storage and management", Research Technical Report, Hosei University, 2016.