

エッジコンピューティング基盤上のストレージシステムにおける データ配置の評価

2017SE035 岸本佳菜子 2017SE107 横溝遥

指導教員：宮澤元

1 はじめに

これまでネットワークに接続されていなかったような様々なものをネットワークに接続する Internet of Things(IoT) 技術が普及してきている。IoT でネットワークに接続されるものを IoT デバイスと呼ぶ。IoT デバイスが生成する多種多様な大量のデータを利用したさまざまなサービスが提案されている。

IoT デバイスが生成する大量のデータを処理するために、エッジコンピューティングが注目されている [1]。エッジコンピューティングとは、利用者からネットワーク的に近い場所にあるデバイス (エッジデバイス) が持つ計算リソースを活用する技術である。エッジコンピューティングでは、ネットワークに接続することが可能で一定の計算リソースを持つデバイスであればエッジデバイスとして利用できる。エッジデバイスの地理的条件や性能が不均一であることが特徴である。

エッジコンピューティングを効率よく活用するためには、エッジデバイスが持つ計算リソースをアプリケーションに適切に割り当てるソフトウェア基盤が重要となる。本研究ではこのようなソフトウェア基盤をエッジコンピューティング基盤とよぶ。エッジコンピューティング基盤ではアプリケーションを実行するエッジデバイスの割り当てや、アプリケーションが利用するデータのストレージへの配置を適切に行う必要がある。

本研究の目的は、エッジコンピューティング基盤上で動作するアプリケーションに対する適切なデータ配置を検討することである。既にエッジコンピューティング上のデータ配置についてさまざまな手法が提案されているように [2, 3, 4]、アプリケーションにとって最適なデータ配置を考えることは、システムの運用コストと通信レイテンシの両方を最小限に抑え、スループットを最大化するために重要である。また、研究課題はシミュレーションによるデータ配置の評価と評価したデータ配置の考察の 2 点である。

2 研究の背景

2.1 エッジコンピューティング

近年、多種多様な IoT アプリケーションが普及してきている。IoT アプリケーションの中には、非常に短い応答時間を必要とするもの、プライベートなデータを含むもの、ネットワークに大きな負担をかける可能性のある大容量のデータを生成するものがある。

IoT で生成されるデータはクラウドコンピューティング (クラウド) で処理されていたが、クラウドへ負荷が集中し

たり通信レイテンシが大きいなどの問題があった。そこでクラウドの問題点を解決するためにエッジコンピューティングが提案された。エッジコンピューティングは、データソース (IoT デバイス) とクラウドデータセンタの間でコンピューティング、ストレージ、およびネットワークキングサービスを提供するネットワークリソースとして定義されている [5]。エッジコンピューティングでは、地理的集中型のクラウドとは対照的に、提供するサービスとアプリケーションは分散されており、ネットワーク接続が可能であればどこにでも展開できる。また、サービスを提供するエッジノードと IoT デバイスとのネットワーク的距離も近いので、通信レイテンシも小さくなる。エッジコンピューティングによりクラウドの問題点を解決できるので、現在の IoT ではエッジコンピューティングを用いたサービスの提供が行われている。

2.2 エッジコンピューティング基盤

エッジコンピューティング基盤は、IoT サービスを効率よく活用するためにエッジデバイスを管理するソフトウェア基盤のことである。エッジコンピューティング基盤では、大量のデータを分散させて処理するために、いくつものエッジノードやストレージノードを利用して、IoT アプリケーションや IoT データを配置して処理を行う。また、短い応答時間を必要とする IoT アプリケーションを IoT デバイスから地理的に近い場所に存在するノードに配置したり、プライバシー保護のためにユーザの近くにデータを配置したりするなど、エッジコンピューティング基盤では IoT アプリケーションや IoT データに応じて適切なリソース割り当てやエッジデバイスの管理を行う必要がある。

エッジコンピューティング基盤を用いて、あらゆる IoT アプリケーションや IoT データを活用し、社会に貢献する仕組みとしてスマートシティがある。スマートシティにはセンサなどを用いてデータを生成する機器 (IoT デバイス) が含まれる。IoT を効率よく活用するために必要なエッジノード、ストレージノード、クラウドデータセンタもスマートシティに含まれている。

2.3 エッジコンピューティング基盤におけるデータ配置

エッジコンピューティングにおける適切なデータ配置について考えるには、エッジコンピューティングを構成するノードの不均一性を考慮に入れる必要がある。あらゆる条件や場面に合わせてデータを配置することで、ネットワークトラフィックや通信レイテンシを削減することができ、ユーザは IoT サービスをストレスなく利用することができ

る。しかし、上記で示した不均一性を考慮に入れた上で、多種多様な IoT アプリケーションや IoT データを適切に配置することは困難である。

3 スマートシティシナリオ

本研究では、エッジコンピューティング基盤でのデータ配置について検討するために、スマートシティのような IoT 環境をシナリオとして想定する。スマートシティのモデル、アプリケーションのデータフロー、IoT データ配置戦略は、iFogSim の拡張版 [3] に定義されているものを利用する。

3.1 スマートシティ環境

iFogSim の拡張版で想定されているスマートシティの構成について説明する。スマートシティはクラウド (DC: データセンター), RPOP (Regional point of presence), LPOP (Local point of presence), GW (ゲートウェイ), IoT (センサ, アクチュエータ) の 5 つの要素から構成されている。ここで、RPOP, LPOP, GW の 3 層がエッジコンピューティング基盤内のエッジノードである。POP とは、スマートシティ内でいくつかのノードが接続を共有する場所を表す。RPOP, LPOP, GW は、DC よりも IoT 側に近い場所にある。

図 1 に想定したスマートシティの構成を示す。下層から IoT, エッジノード, DC で構成されている。エッジノードは、IoT に近いところから GW, LPOP, および RPOP の 3 層で構成されている。DC, RPOP, LPOP, GW のすべてで IoT アプリケーションが動作しており、IoT データを処理して送信することができる。また、それぞれの要素にストレージノードを備え付けることができる。ひとつひとつの GW には、センサとアクチュエータが 1 つずつ備え付けられている。このセンサから送信されたデータは、エッジノード (GW, LPOP, RPOP) や DC で実行されている IoT アプリケーションによって使用される。

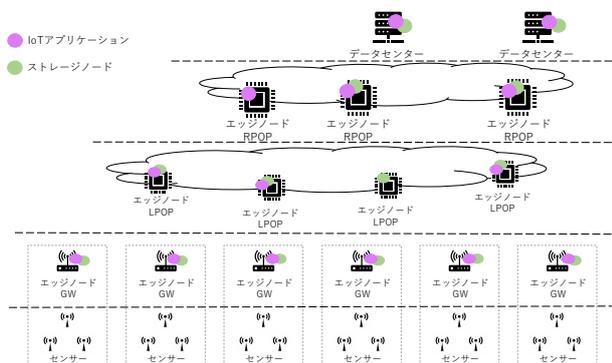


図 1 スマートシティの構成

3.2 アプリケーションのデータフロー

私たちのシナリオでは、スマートシティ全体の中から IoT アプリケーションをランダムに選択してデータを使用

する分散ワークロードを用いる。IoT データはスマートシティ全体で共有でき、使用されることが前提である。このデータフローは IoT デバイスや IoT アプリケーションから生成されるデータをスマートシティ全体で共有し、使用する目的に即したデータフローである。

スマートシティ上での分散ワークロードのデータフローを以下に示す。まず、センサは実世界からデータを収集する。センサが生成したデータは、GW で動作する IoT アプリケーションで処理される。GW の IoT アプリケーションから出力されたデータは、アクチュエータと 1 つまたは複数の IoT アプリケーションに送信される。送信先の IoT アプリケーションはスマートシティ全体のノード (GW, LPOP, RPOP, DC) で動作するものからランダムに選択される。同様に、LPOP の IoT アプリケーションから出力されたデータの送信先は、GW 以外のスマートシティのノード (LPOP, RPOP, DC) の中からランダムに選択される。最後に、RPOP の IoT アプリケーションから出力されたデータは RPOP または DC に送信される。

3.3 データ配置戦略

私たちのシナリオでは、iFogStor というデータ配置戦略を使用する [2]。iFogStor は、IoT 環境全体のレイテンシを最小限に抑えることを目的として作られたデータ配置戦略である。エッジコンピューティング基盤内のノードの場所の不均一性を考慮に入れながら、データを配置する。このデータ配置戦略を用いることで、エッジノードがあらゆる場所に分散されたエッジコンピューティング基盤内のデータ配置について考えることができる。

iFogStor では、以下の制約を考慮に入れデータ配置を行う。

- データプロデューサーとそのデータコンシューマーの間のマッピング (どのコンシューマーがどのプロデューサーからのデータを使用するかの情報)
- エッジノード間の通信レイテンシ
- エッジノードの空きストレージ容量

4 シミュレーション

エッジコンピューティング基盤上でのデータ配置について検討し、評価するために iFogSim の拡張版をシミュレータとして使用し、シミュレーションを行う。このシミュレーションでは、3 節で述べたシナリオ、ワークロード、データ配置戦略を使用する。

4.1 実験内容

シミュレーションにより以下の項目について検討する。

- エッジノードの計算リソースの違い
エッジノードの計算リソースの違いにより、IoT 環境全体にどれだけ影響を与えるかを全体処理時間 (センサがデータを生成してから、そのデータが IoT アプリケーションで処理された後、IoT 環境内のどこかに配

置されるまでの時間)の観点から比較する。

- ストレージノードの配置場所の違い
ストレージノードの配置場所の違いにより IoT 環境全体にどれだけ影響を与えるかを、全体処理時間の観点から比較する。

次に、それぞれの具体的な比較対象を示す。

- エッジノードの計算リソースの違いによる IoT 環境全体への影響の調査
 - デバイスの計算性能を表す値 (MIPS の値) がエッジノード全体で高いとき (MIPS=5000)
 - MIPS の値がエッジノード全体で一般的なとき (MIPS=3000)
 - MIPS の値がエッジノード全体で低いとき (MIPS=1000)
- ストレージノードの配置場所の違いによる IoT 環境全体への影響の調査
 - RPOP にだけストレージを配置したとき
 - LPOP にだけストレージを配置したとき
 - GW にだけストレージを配置したとき

それぞれこのような比較を行い、エッジノードの計算リソースの違いとストレージノードの配置場所の違いによる IoT 環境全体への影響を検討する。

4.2 シミュレーション設定

私たちのシナリオでは、IoT 環境としてスマートシティを想定しているので、シミュレーション内にある程度大規模な環境を構築した。具体的には、2 個の DC、5 個の RPOP、15 個の LPOP、45 個の GW を用意した。私たちのシナリオでは、1 個の GW につき、センサとアクチュエータが 1 個ずつ接続されている。それに伴い、センサとアクチュエータも 45 個ずつ用意した。

スマートシティ内に存在するエッジノードの計算リソースの違いを表すために、MIPS の値 (CPU の処理性能) に着目した。比較をやすくするために MIPS の値のみ変化させることでそれぞれのエッジノードの計算リソースの違いを表した。また、ストレージを配置する場所の違いを表すために、RPOP、LPOP、GW の空きストレージ容量は 500GB と 0 の 2 つのパターンを作成した。DC の空きストレージ容量は 1PB で固定する。IoT アプリケーションは、シミュレートしたスマートシティ内のすべてのノードに割り当てられる。

IoT アプリケーションには、ノードの MIPS の値と空きストレージ容量の範囲内で、利用できる処理能力 (MIPS の値) とストレージ容量を割り当てることができる。今回は、ノードの MIPS の値と空きストレージ容量を IoT アプリケーションの処理能力に反映させるために、各 IoT アプリケーションの MIPS の値とストレージ容量として、それらが動作しているノードの MIPS の値と空きストレージ容量をそのまま割り当てた。

4.3 結果

図 2 は、ノードの性能 (MIPS の値) を 3 通りに変化させたときのそれぞれの全体処理時間を示した図である。ノードの性能の違いによる全体処理時間の違いは比較的小さく、特徴的な傾向は見られなかった。

図 3 は、ストレージノード配置場所を 3 通りに変化させた場合のそれぞれの全体処理時間を示した図である。GW にのみストレージノードを配置したときの全体処理時間の値が大きいことがわかる。また、RPOP にのみストレージノードを配置したときよりも LPOP にのみストレージノードを配置したときの方が全体処理時間の値が大きく、LPOP にのみストレージノードを配置したときよりも GW にストレージノードを配置したときの方が全体処理時間の値が大きいという傾向も見られた。

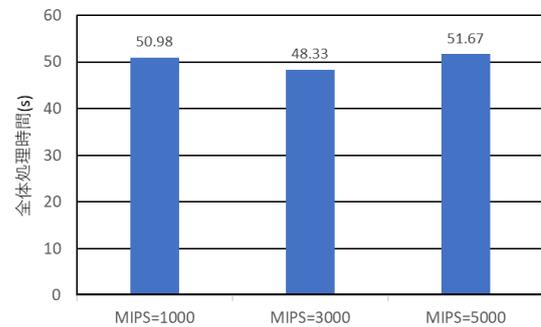


図 2 エッジノード性能を変化させた際の全体処理時間

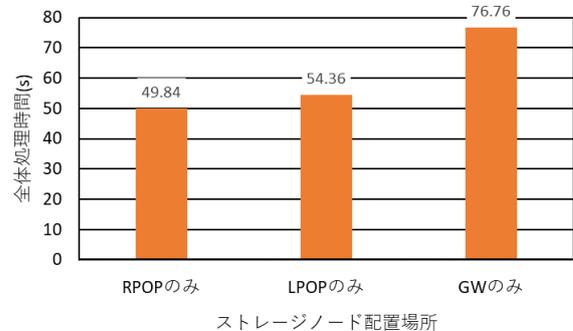


図 3 ストレージノード配置場所を変化させた際の全体処理時間

5 考察

本節では、シミュレーション結果とエッジコンピューティング基盤のストレージシステムについて考察する。

5.1 シミュレーション結果の考察

はじめに、エッジノードの計算リソース (MIPS の値) の違いがスマートシティ全体のデータ配置に与える影響について考察する。シミュレーション結果から、今回シミュレートしたスマートシティ内では、エッジノードの計算リ

ソースを変化させても全体処理時間に対する大きな影響はなかった。この理由は、IoT アプリケーションがデータを処理、生成するのにかかる時間が全体処理時間の値に直接関係がなかったからだと考える。今回シミュレートしたスマートシティ内の IoT アプリケーションが処理するデータは1つだけであり、今回のシミュレーション設定では、IoT アプリケーションの MIPS の値が、すべてのデータの CPU サイズ (そのデータの処理に必要な MIPS の値) よりも大きい。従って、データ処理に必要な処理能力が IoT アプリケーションの処理能力を超えることはないので、全体処理時間に影響がなかった。データの CPU サイズが IoT アプリケーションの MIPS の値よりも大きい場合、または IoT アプリケーションが複数のデータを同時に処理する場合は、データの処理回数や処理するまでデータを待機させる時間が増えるので、スマートシティ内の全体処理時間が大きくなると予想する。

次に、ストレージノードの配置場所の違いがスマートシティ全体のデータ配置に与える影響について考察する。シミュレーション結果から、GW にのみストレージノードを配置したときの全体処理時間の値が非常に大きい。この理由は、データの配置先がデータ生成元からネットワーク距離的に遠くなることと、各データにつき3箇所あるデータ使用先から、データ配置先にあるデータを取得する時間が全体処理時間に加算されることだと考える。今回のシミュレーションで使用したワークロードのデータフローでは、GW から他の GW にデータを送信する場合、接続を共有している LPOP や RPOP を通過させなければならない。GW にのみストレージノードが配置されている場合、データを GW に配置するしか方法がなく、データ使用先の位置によってはデータ取得時間が大きくなってしまう場合がある。従って、今回使用したワークロードやデータ配置戦略では、GW よりも RPOP や LPOP など複数のノードと接続を共有している場所にストレージノードを配置することが適切であると評価する。

5.2 エッジコンピューティング基盤内のストレージシステム

本研究で行ったシミュレーションでは、エッジコンピューティング基盤内のストレージシステムについては考えていない。今回のシミュレーションで行ったようなデータ配置を、実際にエッジコンピューティング基盤内に実装するには、エッジコンピューティング基盤のストレージシステムについて考える必要がある。

クラウドのストレージシステムでは、Ceph[6] や RADOS[7] といったオブジェクトストレージシステムが主流である。オブジェクトストレージは分散しているので、1つのファイルにアクセスすると複数のストレージノードが動作する。しかし、エッジコンピューティング基盤では複数のストレージノードを並列して使うことは難しい。そのため、エッジコンピューティング基盤ではオブ

ジェクトストレージを使用することは難しいと考える。

今後、シミュレーションで得られた知見を実際のエッジコンピューティング基盤に実装するために、エッジコンピューティング基盤内のストレージシステムについて検討する必要がある。

6 まとめ

スマートシティのエッジコンピューティング基盤における IoT アプリケーションで生成されるデータの配置場所について検討し、評価を行った。iFogSim シミュレータの拡張版を用いてシミュレーションを行い、スマートシティ内のエッジノードの計算リソースやストレージノードの配置場所の違いがデータ配置に与える影響を調べた。シミュレーション結果から、GW よりも RPOP や LPOP など複数のノードと接続を共有している場所にストレージノードを配置することが適切であると評価した。

今後は、IoT データの性質や IoT アプリケーションの処理能力についての調査を行い、エッジノードの性能も考慮に入れたデータ配置について検討する。また、シミュレーションで分かったデータ配置を実際のエッジコンピューティング基盤に実装するために、エッジコンピューティング基盤内のストレージシステムについても検討する。

参考文献

- [1] C.Dupont, et al., "Edge computing in IoT context: horizontal and vertical Linux container migration", in *Proceedings of 2017 Global Internet of Things Summit(GIoT)*, pp.1-4 (Jun, 2017).
- [2] M.Naas, et al., "iFogStor: an IoT Data Placment Strategy for Fog Infrastructure", in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pp.97-104 (2017).
- [3] Mohammed Islam Naas, et al., "An extension to ifogsim to enable the design of data placement stragies", in *2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*, pp.1-8 (2018).
- [4] D.Silva, et al., "An Analysis of Fog Computing Data Placement Algorithms", in *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pp.527-534 (2019).
- [5] W.Shi, et al., "Edge computing: Vision and challenges", in *IEEE internet of things journal 3 (5)*, pp.637-646 (2016).
- [6] S.Weil, et al., "Ceph: A scalable, high-performance distributed file system", in *Proceedings of the 7th symposium on Operating systems design and implementation*, pp.307-320 (2006).
- [7] S.Weil, et al., "Rados: a scalable, reliable storage service for petabyte-scale storage clusters" in *Proceedings of the 2nd international workshop on Petascale data storage: held in conjunction with Supercomputing'07*, pp.35-44 (2007).