

エッジコンピューティング基盤におけるストレージ配置に関する検討

宮澤 元†

南山大学 理工学部 ソフトウェア工学科†

1. はじめに

IoT 機器の普及に伴い、クラウド上で IoT アプリケーションを動作させる際の問題点を解決するためにエッジコンピューティングやフォグコンピューティング*が注目されている。エッジコンピューティングを活用することによりアプリケーション実行時の通信レイテンシの削減やエッジの計算リソースの活用などの効果が期待できる。

近年、エッジコンピューティングにおいて、アプリケーションを実行するコンテナや仮想マシンを必要な計算性能や通信レイテンシに応じてさまざまな計算ノードに動的に配置するようなエッジコンピューティング基盤ソフトウェア (エッジ基盤) が提案されている [1, 4]。エッジ基盤を利用することにより、エッジ環境を構成するデバイスの差異などを吸収し、アプリケーションを個別にエッジ環境に対応させることなくエッジコンピューティングを活用することができる。

こういったエッジ基盤の多くは、コンテナや仮想マシンのようなアプリケーションの計算主体の配置を考えたものであるが、アプリケーションの動的な配置に合わせたストレージの効率的な配置方法については十分考慮されていないとは言えない。

本稿ではエッジコンピューティング基盤におけるストレージの配置について検討する。典型的なアプリケーション配置についてストレージ配置に応じたアプリケーション実行性能の違いを確認する。

2. エッジコンピューティング基盤ソフトウェア

エッジコンピューティング (フォグコンピューティング) の普及に伴い、データセンタなどクラウドで利用されている計算ノードとエッジの計算ノードを協調動作させるエッジ基盤が提案されている。このようなエッジ基盤では、エッジノードをアプリケーションの計算ノードとしてコンテナや仮想マシンを動作させることを目的としているものが多い。

KubeEdge[4] はクラウドと同様のアプリケーション実行環境をエッジにも拡張することを目的として開発され

ているエッジ基盤で、エッジにおいてもクラウドと同様のランタイム環境を提供している。Cloud4IoT[1] は IoT アプリケーションを構成するマイクロサービスの最適配置を行うエッジ基盤であり、クラウドの計算負荷をエッジにオフロードする他、クライアントデバイスの移動に応じてマイクロサービスをエッジからエッジへと水平にマイグレーションすることもできる。

アプリケーションの動作にはストレージが不可欠であるが、アプリケーションの配置と同時にストレージの配置についても考慮するようなエッジ基盤は少ない。

3. シミュレーション

エッジ環境を想定した iFogSim[2] シミュレータを用いて計算実験を行った。iFogSim は IoT とエッジ環境をモデル化し、レイテンシなどのさまざまな観点からリソースの管理手法についてシミュレーションを行うことができる。今回の実験では、ストレージに対するデータ配置のシミュレーションを行えるように iFogSim を拡張したシステム [3] を利用した。

3.1 シミュレーションの構成

図1のような構成でシミュレーションを行った。クラウドにあたるデータセンタ、フォグノードにあたる RPoP (Regional Point-of-Presence), LPoP (Local Point-of-Presence), エッジノードにあたるゲートウェイから構成されている。これらのデバイス上に1個ずつアプリケーションモジュールが配置される。また、ゲートウェイには経由でインターネットに接続される IoT 機器にあたるセンサとアクチュエータが接続されており、センサから流れるデータ列をアプリケーションモジュールで処理し、アクチュエータに返すような処理の流れとなっている。アプリケーションが配置されるデバイスはストレージを備えており、データ配置戦略に応じて処理したデータを格納するために利用される。シミュレーションで設定したデバイス数を表1に示す。

	DC	RPoP	LPoP	GW	SN	AC
個数	1	2	10	100	100	100

表 1: 設定したデバイス数

A Discussion on Storage Placement in Edge Computing Infrastructure

† Department of Software Engineering, Faculty of Science and Engineering, Nanzan University

*以下、両者を特に区別せずエッジコンピューティングと呼ぶ。

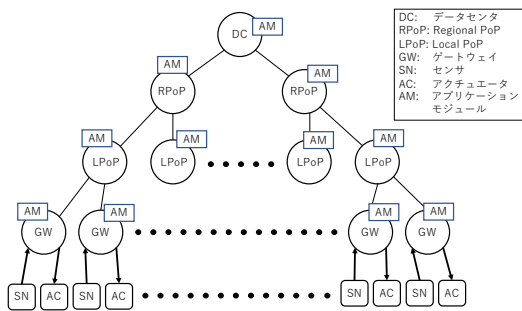


図 1: シミュレーションの構成

3.2 シミュレーション結果

前節で示した構成のネットワークを対象にシミュレーションを行った。アプリケーションモジュール全体の書込時間、読出時間を表 2 に示す。表に示すデータ配置戦略の詳細は以下の通りである。

クラウド 全てのデータをデータセンタに配置する。

ローカル 可能ならばデータを利用するアプリケーションモジュールと同じデバイスに配置する。空きがなければ最もネットワーク的に近いデバイスに配置する。

GAP 一般化割当問題に類似の手法で、全体のアクセス時間を最適化するようにデータを配置する。

	クラウド	ローカル	GAP
書込	78600	55325	25850
読出	81150	0	26900
総計	159750	55325	52750

表 2: シミュレーション結果

比較した 3 つのデータ配置戦略の中ではクラウドが最もアクセスに時間がかかる結果となっており、エッジを利用する意味があることがわかる。ローカルにおける読出時間 0 は、ストレージアクセス自体にコストがかかることを考えると不自然であり、検証の必要がある。

4. エッジ基盤におけるストレージ配置に関する検討

クラウドコンピューティングでは、大量の計算リソースを利用可能であり、またデータセンタ内では低レイテンシで広帯域の通信が可能であることから、アプリケーションを動作させる計算ノードとストレージを管理する

ストレージノードに物理的に異なる計算機を利用することも多い。一方、エッジコンピューティング環境では、利用できる計算リソースは一般に少なく、計算ノードに直接接続されたストレージを利用したデータ配置を行うことがデータアクセス時間に関しては有利であることが多いと考えられる。ただし、エッジコンピューティング環境では個々のデバイスが持つ計算リソースは質・量ともに非常に幅広いので、個々のデバイスの性質に応じたデータ配置を行う必要がある。また、モビリティを持つ IoT 機器などの利用を考慮すると、最適なデータアクセス性能を保証するには、アプリケーションの配置に応じてデータの配置も動的に変化させる必要がある。

5. まとめと今後の課題

エッジ環境における IoT アプリケーションを想定したシミュレータを用いて、各デバイス上のストレージに配置されたデータへのアクセス時間について計算実験を行った。複数のデータ配置戦略の比較により、エッジコンピューティングを利用する効果があるケースがあることが分かった。一方で、今回の結果は非常に限定された仮想的な状況での計算実験であり、実アプリケーションにおけるデータセットや一般のアプリケーション配置においてどのような結果になるかはさらに検証する必要がある。

今後は、アプリケーション配置が動的に変化するようなケースでのストレージおよびデータの効率的な配置手法について検討を進める。

参考文献

- [1] Corentin Dupont, et al., “Edge computing in IoT context: Horizontal and vertical Linux container migration,” in *2017 Global Internet of Things Summit (GIoTS)*, pp.1–4, 2017.
- [2] Harshit Gupta, et al., “iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments,” in *Software: Practice and Experience*, Vol.47, No.9, pp.1275–1296, 2017.
- [3] Mohammed Islam Naas, et al., “An Extension to iFogSim to Enable the Design of Data Placement Strategies,” in *2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*, pp.1–8, 2018.
- [4] Ying Xiong, et al., “Extend Cloud to Edge with KubeEdge,” in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pp.373–377, 2018.