

フォグコンピューティングにおける 実行配置管理システムの提案

鎌田 幸希¹ 稲村 浩¹ 中村 嘉隆¹

概要 : IoT などの応用において発生する大量のデータを効率良く扱うためにクラウドコンピューティングのパラダイムをネットワークの端まで広げたフォグコンピューティングが提唱されている。フォグコンピューティングでは、データと処理の最適配置について議論されている。ネットワークの分野では、従来のホスト指向のアーキテクチャではない、CCN(Contents Centric Network)の研究がなされており、ネットワーク内にキャッシュされたコンテンツを自然に扱うことができ、トラフィックや遅延時間の削減が可能であることが示されている。本研究ではフォグコンピューティングにおいて実行地点選択と資源配置のための実行配置管理システムを提案する。本システムはCCNに習い、クライアントからのアクセストラフィックそのものを最適化のヒントにすることで、通信量のオーバーヘッドを押さえつつ継続的なネットワーク内の資源監視を可能にする。セッション層にネットワーク内資源監視機能、候補選出機能、サービス移送機能を実装することで、フォグノード上のアプリケーション層で動くサーバやクライアントから透過とし、自律的にサービスの実行地点選択と資源配置を実現するものである。提案したシステムをシミュレータ上に実装し、正しいサービス移送動作の確認を行った。

キーワード : フォグコンピューティング, インネットワークキャッシュ, コンテンツ指向型ネットワーク

KAMADA KOUKI¹ INAMURA HIROSHI¹ NAKAMURA YOSHITAKA¹

1. 背景

2017年時点で274億個あるIoT機器は2020年までには約400億個にまで増えるという予想がある[1]。これらの大量のIoT機器の生成するデータに対してクラウドコンピューティングのような処理集中型のアーキテクチャでは、エッジの持つ処理能力が活用されておらず、データ発生地点から遠隔にあるデータセンターまでのレイテンシも無視できない。そこで、クラウドコンピューティングのパラダイムをネットワークの端まで広げたフォグコンピューティングが提唱されその研究が活発である[2]。

ネットワークの分野では、従来のIPアドレスによるアーキテクチャではなくNDN(Named Data Networking)を始めとするコンテンツ指向型ネットワーク(CCN:Contents Centric Network)の研究[3]がなされており、位置によらないコンテンツを識別子とすることで、ネッ

トワーク内(In-Network / インネットワーク)でキャッシュされたコンテンツを自然に扱うことができ、トラフィックや遅延時間の削減が可能であることが示されている。

2. 関連研究

2.1 フォグコンピューティング

フォグコンピューティングでは、実行処理に必要な地点を選択し、移送させることで、実行にかかる遅延時間を抑えている。たとえば、Wireless Sensor and Actuator Networkingにおいて、センサノードが収集したデータをクラウドに移す前に、単純な処理はゲートウェイノードで実行し、アクチュエーションまでの遅延時間を削減することが可能になっている。このように、フォグコンピューティングでは計算リソースの最適配置について多くの議論がなされているが、コンテンツの最適配置という面では議論がなされていない。

2.1.1 CoBOS

Bergら[4]は、異なる性能特性とコストの関係を持つ高度に分散した異機種リソースを含む n 層環境を対象とした

¹ 公立はこだて未来大学システム情報科学部
School of Systems Information Science,
Future University Hakodate

コードオフロードシステム CoBOS を提案している。この手法では、コードバブリングという概念を提案している。コードバブリングは、コードをより強力により遠い層に動的かつ適応的に移動させ、 n 層環境で効率的でスケラブルなコードオフロードを可能にしている。これにより、 n 層環境でのコードオフロードで 77 % のエネルギー消費と 83 % の実行時間を削減している。

2.2 CCN

Jacobson ら [3] は、従来の IP アドレスによるアーキテクチャではない CCN を提案している。CCN の通信では、Interest と Data という 2 種類の CCN メッセージを用いたプロトコルで行われる。メッセージの送受信には、Interest メッセージをルーティングするための FIB(Forwarding Information Base)、コンテンツをキャッシュする CS(Content Store)、Data を要求元に送り返すための PIT(Pending Interest Table) の 3 つの主要なデータ構造が用いられている。これらのデータ構造を用いて CCN では Interest と Data のメッセージのやりとりをしている。彼らは、これにより、IP のシンプルさとスケラビリティを維持したまま、セキュリティ、配信効率、および中断耐性が大幅に向上したことを証明している。このように、CCN ではコンテンツをユーザーに近い場所に置くことによるスタティックなコンテンツの配布は可能である。しかし、実行されているシステムを同様に扱うには、単純にキャッシュを増やしてユーザーに近い場所へ配置するだけでは、同一の内部状態をもとにしたシステムを継続させることができず、動的なコンテンツやサービスの提供は不可能である。

3. 研究の目的と課題

本研究の目的は、自律的な実行地点選択と資源配置のためのフォグコンピューティングとインネットワークキャッシュ機構の融合の実現である。フォグコンピューティングは、クラウドコンピューティングでは解決できないレイテンシの問題を分散した計算資源の有効活用によって解決する新しいパラダイムであったが、フォグネットワーク上でデータ配置に自由度がある場合について拡張する余地がある。コンテンツ指向型ネットワークは、インネットワークキャッシュを用いて従来のネットワークにおけるホスト指向ではなくコンテンツ指向を実現することで、低レイテンシ、トラフィックの削減、耐障害性向上などを図ろうという考え方である。そのフォグコンピューティングにコンテンツ指向型ネットワークのメリットを融合することでフォグネットワーク上の容量を有効に活用でき、より分散させたリソースを効率よく使えるようになる。

目的の実現のためには、実行処理とデータ配置の選択性を実現可能なアーキテクチャを設計し有効性を確認する必要がある。まず実行処理の配置を自律的に最適化する手法

を検討する。

4. アプローチ

実行位置の配置を管理する手法として本研究では CCN に習い、クライアントからのアクセストラフィックそのものを最適化のヒントにすることで、通信量のオーバーヘッドを押しえつつ継続的なネットワーク内の資源監視を可能にする。CCN ではクライアントからのアクセスに応じてコンテンツ配置が最適化されるように、本システムではサーバの実行地点も同様に最適化したい。従って、クライアントからのアクセスメッセージで必要な資源情報を入手できるようにする必要がある。本システムでは位置が固定されたクライアントに対し、サーバの実行地点を調整することで利用資源と提供するサービスの品質の最適化を図る。

5. 実行配置管理システムの提案

提案するシステムを実行配置管理システムと呼ぶ。本システムではクライアントはセンサやユーザーなどと直接のやりとりがあるためネットワークにおける動作位置は固定されているものと想定する。それに対してサービスを提供しているサーバはフォグ/クラウド内での配置に任意性があるため、処理の実行にあたり必要な資源を求めてサービス実行の状態を移送することで実行地点を変更可能であり望ましいと仮定する。

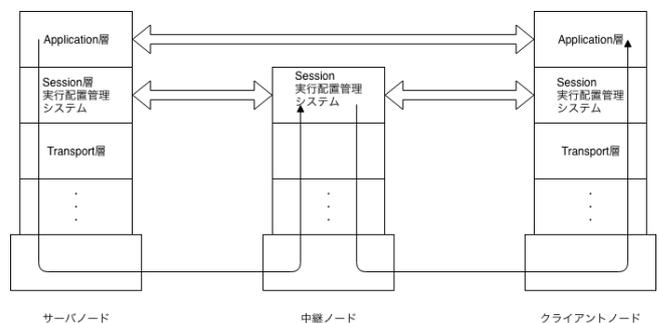


図 1 実行配置管理システム

実行地点管理システムの動きを図 1 で示した。本システムは全参加フォグノードのセッション層で動作しているものとする。サービスの実行地点の変更はクライアントやサーバに透過的に行われることが望ましい。従って 7 層モデルで言うところのセッション層で動作するものとする。サービスの移送などの管理動作はトランスポート層以下の資源が操作対象であるため、それらが可視である上位層に位置する必要がある。かつクライアントやサーバが動作しているアプリケーション層よりも下位であるのはセッション層である。サーバノードとクライアントノードとのアプリケーション層での通信を利用して、セッション層の実行配置管理システムで通信処理を中継し、サーバノード、中継ノード、クライアントノードの実行配置管理システム間

で通信し、サービスの実行地点選択と資源配置を実現する。

本提案システムは3種類のノードで構成される。オリジナルのデータを持ちネットワーク遅延はあるが処理能力の高いクラウドノード、次に中処理能力でエンドデバイスと比較的低レイテンシで通信ができる中間のフォグノード、スマートフォンのようなサーバに参加するエンドデバイスであるクライアントノードをそれぞれ複数接続しているネットワークを構築している。クライアント/サーバ型の通信モデルを基本にクラウドノードとフォグノードがサーバの役割を葉のノードがクライアントの役割をこなす。サーバは、参加しているクライアントの通信状況を監視し、通信状況から自立的に自分がサーバの役割をするべきか、他のフォグノードにサーバの役割を委託するべきか判断する。委託されたフォグノードは、サーバの役割を引き継ぐ。このように、サーバの役割の実行地点最適化を図る。これにより、レイテンシを抑えようとするものである。

5.1 ノードの持つ機能

各フォグノードは、ネットワーク内資源監視機能、候補選出機能とサービス移送機能を持っている。ここでは、各機能について説明する。各ノードにおける実行配置管理システムはセッション層で動作することによりサーバとクライアント間の通信メッセージに合わせて管理情報を重畳させ、ネットワーク内資源監視機能を実現する。同時にネットワーク内資源の変化を常時監視し、変化が見られた場合に候補選出機能を実行し、必要と判定されれば選択されたノードに向けてサービス移送機能を実行しサーバの実行地点の最適化を実現する。

5.1.1 ネットワーク内資源監視機能

ネットワーク内資源監視機能では、現在使われているネットワーク資源を監視して、サービスを移送するかどうか決めるための情報を収集している。具体的には、本システムを載せたフォグノードがサービスを展開している際に、この機能は、そのサービスに参加しているクライアントノードとの遅延などのネットワーク情報やフォグノード自体のCPUやメモリの資源を監視して記録している。前項で述べたように通常のクライアントとサーバの通信メッセージのやりとりに管理情報を載せ、クライアントとサーバの経路上に存在する中継ノードの情報を収集する。

5.1.2 候補選出機能

候補選出機能は、ネットワーク内資源監視機能で集めた情報に変化が見られた場合に呼び出される。本機能はサービスを移送するべきか判断する。また、移送するべきであれば、ネットワーク内資源監視機能によって収集された情報に基づき適切な候補ノードを選択する。この機能で、委託先を決定した後に、サービス移送機能によりサーバの委託を行う。

5.1.3 サービス移送機能

本機能は、候補選出機能で選出したノードに対してサービスを移送する機能である。現在のサーバであるノードは、候補選出機能で定めた新しいサーバ候補のフォグノードに対して、委託することを知らせるメッセージとともに、サービスに必要なデータの情報を送信する。サーバ候補のフォグノードは必要なデータをすべて集めサーバを担う準備ができたところで現在のサーバであるフォグノードに委託可能であることを知らせるメッセージを送信する。現在サーバであるフォグノードはそれを受け取ると、サーバに参加しているクライアントノードに対して新たな宛先を通知するメッセージを送信する。このようにして、サービスを移送する。

6. シミュレータ上での実現

提案システムの有用性の検証に向けて、ユースケースを用意し、それに適した管理対象のネットワーク内資源と候補選出基準を定義し、シミュレータ上に本システムを実装し動作の確認を行う。評価するにあたり、シミュレーションソフトウェアには、既存のネットワークシミュレーションソフトウェアである ns3[5] に CCN の 1 つである NDN の拡張モジュールを追加した ndnSIM[6] を用いる。

提案システムの有用性を示すにあたり以下のようなユースケース、及び、シミュレーションシナリオにおいて実験を行った。

6.1 ユースケース

ユースケースは、多地点でのインターネット会議を想定する。インターネット会議システムではサーバは接続する全端末から受信したメディアデータを混合して一つのストリームとして全端末に配信する動作を行う。地理的に分散した参加者が随時システムに接続する状況でメディア品質を公平に保つことを目標とする。ネットワーク上で遅延時間が異なる位置に分散したクライアントに対してサーバが最適な位置に移動する挙動をシミュレーションで確認する。

6.2 候補選出基準

候補選出基準の定義にあたり、まず本ユースケースで管理対象資源と定めたクライアントとサーバ間の遅延時間について以下の2つの基準を定義する。

遅延時間の平均

遅延時間とは端末 T_i とサーバ S までの時間距離を $delay(S, T_i)$ とし、以下の式から算出する。

$$\frac{1}{N} \sum_{i=1}^N (delay(S, T_i)) \quad (1)$$

遅延時間の分散

端末とサーバ間の通信の遅延時間の分散に注目する。参

加端末 T_i が低遅延であるのに対して、他の参加者が T_j にて高遅延を経験するのは相互通信の品質の観点から有益ではないと考えたからである。従って、なるべく参加者間での遅延時間が分散しないことが望ましいと考えられる。そこで、遅延時間の平均と同様に参加端末 T_i とサーバ S の間の遅延時間を測り、それらの分散を算出することで評価する。遅延時間の平均と同様に S と T_i の遅延時間を $delay(S, T_i)$ 、参加端末の遅延平均値 av としたときに以下の式から算出する。

$$\frac{1}{N} \sum_{i=1}^N (av - delay(S, T_i))^2 \quad (2)$$

地理的に分散したノード間での回線品質の公平さの実現のために、本ユースケースでの候補選出基準は遅延時間の分散とし、小さな分散を実現できるノードを候補として選出するものとする。

6.3 シミュレーションシナリオ

インターネット会議に参加している端末 T_i の通信参加状況によってサーバの位置が自律的に移動することを確認するために、 N 個の端末 T_i が随時参加していくシナリオとした。シミュレーションを行ったネットワークトポロジであるが、まず、単純なネットワークでサーバを切り替える事による遅延が軽減し、自律的で最適な実行地点選択が行われているかを示すために、図2のような深さ5の2分木のネットワークトポロジにて動作検証を行った。この2分木におけるルートノードを従来のサーバ・葉のノードをサーバと通信を行うクライアントのノード、その他のノードをフォグノードとした。サーバ S に対して4個の端末 T_i が随時参加していくシナリオを用意した。従来手法のシナリオとして、同一のネットワークトポロジで移動することのない単一のサーバと通信するシナリオを比較のために用意した。

6.4 シミュレーション環境

シミュレーションの環境について述べる。シミュレーションの際に用いたネットワークトポロジは図2のような2分木のトポロジをもちいた。表1にトポロジに用いたパラメータを記す。

6.5 結果と考察

第6章で定義したシミュレーションシナリオを実行し、参加する4端末が表2のタイミングでインターネット会議に参加するとしたとき、図2においてクライアントノード1がインターネット会議に参加すると、サーバがフォグノード1 → 2 → 4 とサービス移送を行う。その次に、クライアントノード2がインターネット会議に参加を開始するが、クライアントノード1と2ともに遅延は2msでそれらの間

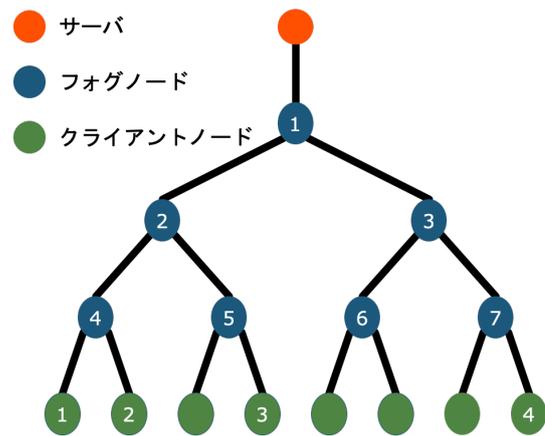


図2 シミュレーショントポロジ

表1 トポロジパラメータ

パラメータ	設定値
Server 数	1
FogNode 数	7
Client 数	8
回線容量	2Gbps
伝播遅延	2ms
シミュレーション時間	20s

表2 参加端末の情報

ノード番号	参加秒数
1	1-20 秒
2	6-20 秒
3	11-20 秒
4	16-20 秒

の分散は0となりそのままサーバはフォグノード4で安定する。その後、クライアントノード3がインターネット会議に参加すると、フォグノード4がそのままサーバを担った場合では、分散は3.56、フォグノード2にサーバを移送した場合、分散は0となり、本システムはより遅延の分散の小さいフォグノード2にサービス移送を行い安定する。最後にクライアントノード4が参加すると、現在のサーバであるフォグノード2における分散は3、隣接するフォグノードに移送した場合の分散は、それぞれフォグノード4では14.25、フォグノード5では10.19、フォグノード1では0となり、一番クライアントノードの遅延の分散が低くなるフォグノード1にサービス移送を行い、フォグノード1にてサーバが安定する。現在、実装中の実行配置管理システムを利用したシナリオと利用しないシナリオのシミュレーション結果から得られた、必要に応じたサービス移送の様子を各ノードから見た遅延時間 [ms] の変化として、横軸の経過時間 [s] に対してプロットしたものが、それぞれ図3と図4である。どちらのシナリオにおいても表2のタ

イミングで参加している。この結果から、サービス移送の動きを方式に則り説明できた。現状の実装では候補選出機能に問題があり(図3の縦方向の意図しないプロット)、正しい実装とはなっていない。

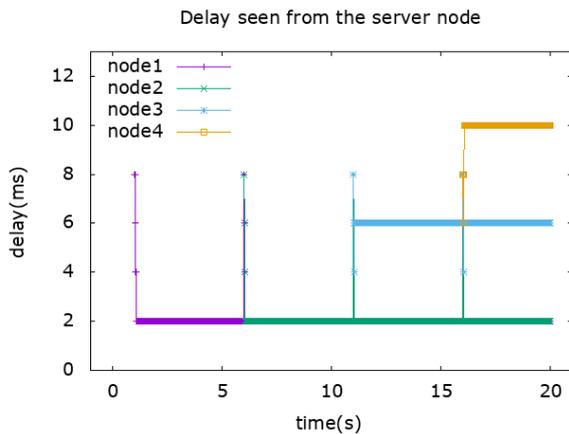


図3 提案システムを利用したシミュレーション結果

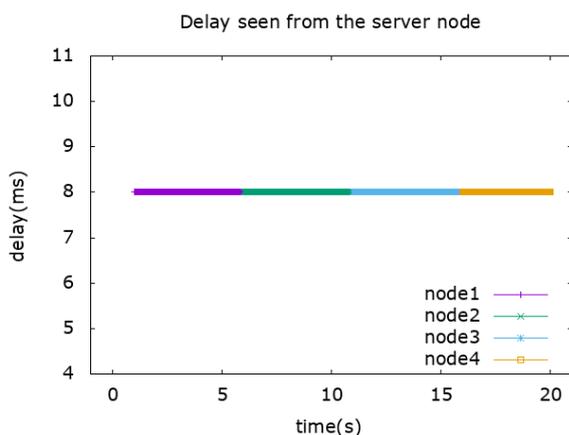


図4 既存システムを利用しないシミュレーション結果

7. 終わりに

フォグコンピューティングにおいて実行地点選択と資源配置のための実行配置管理システムを提案した。本システムは、フォグネットワークにおける、フォグノードのセッション層にネットワーク内資源監視機能、候補選出機能、サービス移送機能を実装することで、アプリケーション層で動くサーバやクライアントを意識することなく、自律的にサービスの実行地点選択と資源配置を実現するものである。提案したシステムをシミュレーション上に実装し、正しいサービス移送を行えているか確認を行った。今後、開発中のシミュレーションを完成させ、更に検証と評価を進める。今後は、バックグラウンドトラフィックを生成し競合させるなど、より現実的なシミュレーションシナリオを検討していく。今回は、サービス移送対象の候補を選出す

るためのパラメータを遅延のみに絞って実験をおこなったが、CPU使用率・メモリ・電力・ノードを利用する金銭的成本など様々なパラメータを元に移送先の候補を選出することも検討していく。

参考文献

- [1] Ministry of Internal Affairs and Communications, Japan: 平成30年版情報通信白書, Japanese Government (2018).
- [2] Bonomi, F., Milito, R., Zhu, J. and Addepalli, S.: Fog Computing and Its Role in the Internet of Things, *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, ACM, p. 1316 (2012).
- [3] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H. and Braynard, R. L.: Networking Named Content, *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, ACM, pp. 1–12 (2009).
- [4] Berg, F., Drr, F. and Rothermel, K.: Increasing the Efficiency of Code Offloading in N-tier Environments with Code Bubbling, *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MOBIQUITOUS 2016, ACM, pp. 170–179 (2016).
- [5] Henderson, T. R., Lacage, M., Riley, G. F., Dowell, C. and Kopena, J.: Network simulations with the ns-3 simulator, *SIGCOMM demonstration*, Vol. 14, No. 14, p. 527 (2008).
- [6] Mastorakis, S., Afanasyev, A. and Zhang, L.: On the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation, *ACM Computer Communication Review* (2017).