

フォグコンピューティングにおける RDF 推論処理の動的な負荷分散

小久保柚真[†] 天笠 俊之^{††}

[†] 筑波大学大学院 理工情報生命学術院 システム情報工学研究群 〒305-8577 茨城県つくば市天王台 1 丁目 1-1

^{††} 筑波大学 計算科学研究センター 〒305-8577 茨城県つくば市天王台 1 丁目 1-1

E-mail: [†]kokubo@kde.cs.tsukuba.ac.jp, ^{††}amagasa@cs.tsukuba.ac.jp

あらまし IoT デバイスの急速な増加に伴い, IoT に適した分散処理環境としてフォグコンピューティングが注目を集めている. フォグコンピューティングを採用することにより, クラウドサーバへの一極集中と通信レイテンシを改善できるが, 一方で各フォグノードの計算能力は限られているという制約がある. 既存のフォグコンピューティングにおける RDF 推論処理では, 推論を行うノードの負荷を考慮していないため, 特定のノードに負荷が集中し却って遅延が大きくなってしまふことがある. そこで提案手法では, 各ノードにおいて CPU 使用率を用いて負荷状況を把握し, 高負荷なノードがある場合にはサーバと処理を分散することで負荷の軽減を図る. 予備実験により, 負荷を考慮することの必要性及び分散機構の導入による効果を示した.

キーワード フォグコンピューティング, 負荷分散, RDF

1 序 論

通信技術の発展やセンサの性能向上により, 交通, 農業, 医療などをはじめとする幅広い分野で Internet of Things (IoT) の活用が始まっている [1]. IoT の一般的なアーキテクチャでは, 各 IoT デバイスから得られたデータはクラウドサーバに集約され処理される. しかしながらデバイス数の増加によって, 通信帯域の圧迫やクラウドへの負荷が集中することによる処理の遅延などが問題になっている.

そこで IoT に適した分散処理環境として, フォグコンピューティングが提案された [2]. フォグコンピューティングは処理能力を IoT デバイスのある LAN 内に配置し一次処理を行うもので, これにより上述したクラウドコンピューティングの課題を解決することができる.

近年, 特に推論や機械学習などといったタスクが IoT デバイスを用いて行われるようになってきている. このようなタスクを行う上で, 多種多様なデバイスから得られるデータを相互に扱う必要がある. そのため各デバイスから得られた生データに対してメタデータを付与することで, 機械で解釈可能な形へ統一されるが, この際に標準的に用いられるのが RDF というデータ形式である. RDF を用いることで, 背景知識を活用した推論を容易に行えるため, RDF 形式のストリームを対象としたシステムが注目を集めている [3] [4] [5] [6].

既存のフォグコンピューティングにおける RDF 推論処理では, 推論を行う各フォグノードの負荷を考慮していない. そのため通勤などにより情報源が移動する場合, 特定のノードに負荷が集中し, 却って遅延が増加してしまうことがある.

そこで本稿では, 各フォグノードにおいて CPU 使用率を用いて負荷状況を把握し, 高負荷なノードがある場合にはクラウ

ドサーバと推論処理を分散することで負荷を軽減する手法を提案する.

本稿の構成は, 以下の通りである. 2 章で本研究の前提となる知識について説明し, 3 章で関連研究について述べる. 4 章で分散機構を導入した提案手法について述べ, 5 章で小規模なトポロジーで実施した予備実験の結果を示す. 最後に 6 章で本稿のまとめを述べる.

2 前提知識

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は, Cisco Systems 社が提唱した分散処理アーキテクチャである. IoT デバイスのあるローカルなネットワーク内にデータを一次的に処理するフォグノードを配置し, 処理したデータを必要とするデバイスへ送信する. 図 1 にフォグコンピューティングの概念図を示す.

フォグノードで処理することができればクラウドを経由することなく結果を送信できるため通信の遅延が少なく, 同時にクラウド・フォグ間の帯域コストを削減することもできる. またデータをローカルで処理できるため, セキュリティレベルの向上やプライバシー保護も期待できる. 一方フォグノードはクラウドサーバと比較して処理能力が限られている点や, 一時処理を行うために追加のデバイスが必要となるといった点も考慮しなければならない.

2.2 RDF

RDF (Resource Description Framework) は, 主語, 述語, 目的語の 3 要素で構成されるトリプルによってリソースの関係を表現する枠組みであり, W3C によって勧告されている [7]. トリプルの各要素に, URI (Uniform Resource Identifier) と

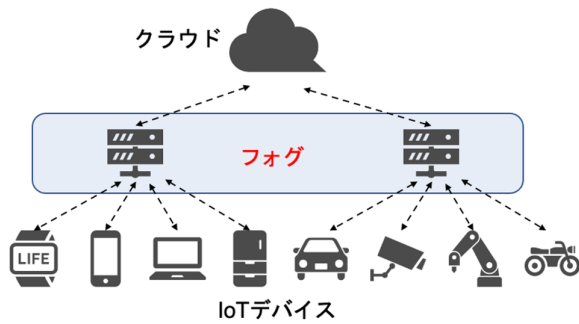


図1 フォグコンピューティング

いう Web 上の特定のリソースを一意に識別する識別子を用いることで、データを統一的に扱うことが可能となる。RDF の表記には RDF/XML や JSON-LD をはじめとするさまざまなものがあるが、提案手法では人間にとっても理解しやすく広く利用されている Turtle を利用する。

本研究においては、RDF で表現されたデータを用いてルールベースの推論を行う。受信したトリプルに対して、条件を列挙した推論ルールを適用し、条件を満たす場合には推論によって導出された結論を新たなトリプルとして生成する。

3 関連研究

3.1 エッジデバイスを用いた IoT システムにおけるルールベース推論

Su らの研究では、クラウドとエッジデバイス（フォグノード）に推論を分散させることで、どのようにパフォーマンスを向上させることができるかを分析している [3]。ユースケースとしてスマートシティの交通システムを想定し、タクシーの軌跡データ（位置、速度、方向など）を用いて実験を行った。また、実験では 4 つの RDF フォーマットごとに転送時間、推論時間の比較も行なっている。

分析によりエッジデバイスを用いることで、エッジで処理できる場合には非常に高速に処理できることわかった。しかしエッジに配置するルールについては静的であり、あらかじめ決められたルールを適用している。

3.2 推論ルールを動的に分散するためのアーキテクチャの提案

Seydoux らは Su らの研究において、ルールの配置が静的であったことに注目し、フォグコンピューティングにおいて推論ルールを動的に分散するためのアーキテクチャとアプローチの提案を行った [4]。ルールを適用できる最下ノードで推論を行うことにより、応答時間の高速化とスケーラビリティの向上を図った。

しかしながら、推論を行うフォグノードの処理能力とデータ量の動的な変化を考慮していないという課題がある。上述したように、フォグノードはクラウドサーバと比較して処理能力が乏しいため、処理を分配することでかえって全体の実行時間が長くなってしまうことがある。また、情報源が移動する場合や特定の時間、領域においてデータ量が一時的に増加する場合、

既存手法では各ノードの負荷を考慮していないため、特定のノードに負荷が集中し遅延が増加してしまう。

4 提案手法

本研究では Seydoux らの研究の課題であったフォグノードの処理能力とデータ量の動的な変化を考慮していないという点に対し、各フォグノードの負荷を考慮することで、より洗練された分散を行う分散推論アーキテクチャ及びアプローチの提案を行う。具体的なアプローチとして、各フォグノードにおいて CPU 使用率を用いて負荷状況を把握し、負荷が基準値を超える場合、適用する推論ルールを分散するような機構を導入する。

4.1 アーキテクチャ

本研究で設計した効率的に推論を行うためのシステムの全体図を図 2 に示す。この設計については、3.1 で説明した Su らの実験で使用されたものを参考にした。システムには IoT ノードとフォグノード、そしてクラウドサーバが含まれており、それぞれのノードの構成について図の右側に示している。IoT ノードからフォグノードへの転送にはソケット通信を、フォグノードからクラウドサーバへの転送には MQTT というプロトコルを用いている。MQTT は Publish/Subscribe モデルという Broker を介してメッセージの送受信を行うプロトコルである [9]。http と比較して軽量かつ双方向で多対多の通信ができるといった特徴があるため、IoT において一般的に用いられている。MQTT では転送の品質を表す QoS (Quality of Service) を設定する。QoS0, QoS1, QoS2 の 3 段階に分けられているが、本システムにおいては制御信号を用いてルールの分散を行うため、メッセージがロストしないことを保証する QoS1 を採用した。

各ノードの構成について、IoT ノードはデータを生成、収集するデータ収集器と RDF 化するための RDF アノテータで構成される。

フォグノードは実際に推論を行うための推論器と推論に用いる知識ベース、そして負荷を分散するための分散機構で構成される。推論器については、RDF を処理するための Java フレームワークである ApacheJena を利用した。Jena には RDF を用いた推論に加え、推論に用いる知識階層の定義や推論結果を格納するデータベースなどの API も含まれている。

クラウドサーバは同様に推論器、知識ベース、そしてとの間で処理を分散するための分散機構があり、加えて MQTT による Publish/Subscribe 通信を行う際にその仲介役となる MQTT Broker が配置されている。MQTT Broker には mosquitto を使用した。

4.2 分散機構の動作

基本的な負荷分散の方針は、あるフォグノードに負荷が集中した場合、フォグノードで推論の際に適用するルールを減らし代わりにクラウドサーバで適用するルールを増やすというものである。具体的な手順は以下になる。

- (1) IoT ノードからデータを受信する

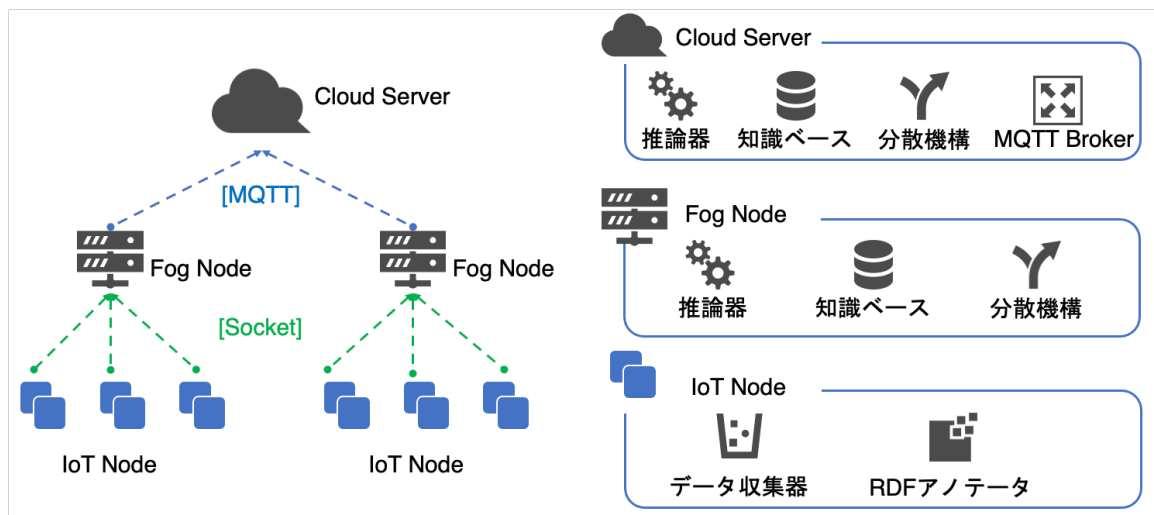


図 2 設計したシステムの全体図

- (2) CPU 使用率を取得する
- (3) 使用率に応じて適用する推論ルールを決定する
- (4) データにどのルールを適用したか示す情報を付与し、クラウドサーバに転送する
- (5) クラウドサーバではその情報を元に残りのルールを適用する

CPU 使用率は推論を行う度に取得し、使用率が変動すればその都度適用ルールの割り当てを変更する。現在は CPU 使用率が 30 %を超えたら徐々に推論で適用するルールの数を減らし、80 %を超えたら推論を行わないようにしている。IoT ノードでは 50 トリプルごと 1 つのメッセージとして送信しているため、50 トリプルごとに情報を付与することになる。

5 実 験

予備実験として 1 つのフォグノードを用いて、フォグノードに負荷をかけた時の処理時間の変化と提案した分散機構の効果を確認した。

5.1 実験環境

IoT ノードには、MacBook Pro (2.4GHz quad-core Intel Core i5, RAM16GB) を用いた。クラウドサーバは、同 MacBook Pro 上に構築した仮想マシンに実装した。仮想マシンのプロセッサ数は 2、割り当てメモリは 4096MB である。フォグノードについては、Raspberry Pi 4 Model B (1.5GHz quad-core ARM Cortex-A72, RAM 8GB) を用いた。

5.2 データセット

データセットには、City Pulse プロジェクト [8] で提供されているデータを利用した。このプロジェクトでは、スマートシティに関する交通、天気、イベントなどのデータやベンチマークなどの各種ソフトウェアを提供している。

その中でも大気汚染に関するデータを使用し、AQI (Air Quality Index) という汚染の程度を示す指標の推定を行った。データには、データを取得したセンサの緯度経度、取得日時、

AQI が含まれている。AQI の具体的な区分について以下の表 1 に示す。

提供されているデータには、あらかじめ RDF フォーマット的一种である Turtle でアノテーションが付与されている。そのため実験では IoT ノードからこのデータをそのまま転送する。使用した RDF データ数は 36000 トリプルである。

表 1 AQI の区分

AQI	Description
0-33	Very Good
34-66	Good
67-99	Fair
100-149	Poor
150-200	Very Poor
200+	Hazardous

5.3 実験結果

図 3 にクラウドサーバとフォグノードのどちらか一方で全ての推論を行った時の処理時間を示す。図 3 において、オレンジで示したその他には MQTT での転送時間やファイル IO などにかかる時間が含まれている。また、左側の「クラウド配置」が全ての推論をクラウド上で、右側の「フォグ配置」が全ての推論をフォグで行った時の結果である。

結果より、青色で示した推論時間がクラウドサーバとフォグノードの一方に集中していることが確認できる。そしてクラウドサーバに対してフォグノードの推論時間の方が長くなっていることから、両者の処理能力の差を表現できている。

次にフォグノードに負荷がかかった時の処理時間の変化を図 4 に示す。ここで負荷は stress-ng コマンドを使用し、意図的に CPU に負荷をかけている。図 4 において、左側のフォグに 4 つがフォグノードで 4 つのルールを適用した場合の結果、右側が分散機構を導入した提案手法の結果である。その上の 0, 50, 100 は意図的にかけた CPU 負荷率を、F,C はそれぞれフォグノードとクラウドサーバの処理時間を表している。

フォグノードで4つのルールを適用した場合の結果からルールを静的に配置した場合、負荷がかかると処理時間が増加することがわかる。一方提案手法では負荷がかかるとフォグノードの推論時間が短くなり、クラウドサーバの推論時間が増加していることから、提案した分散機構が正しく動作していることが確認できる。

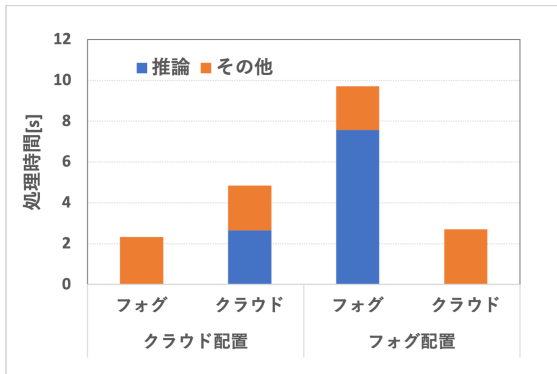


図3 クラウドサーバとフォグノードの処理能力の比較

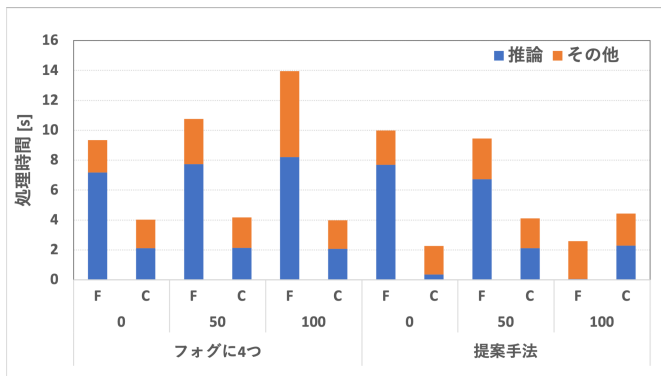


図4 フォグノードに負荷をかけた時の処理時間の比較

最後に IoT ノードがデータを送信してからフォグノードまたはクラウドサーバで推論を行い、その推論結果を受信するまでにかかった時間の比較である。図5に実験に使用したトポロジーを、図6に実験結果を示す。今回結果を受け取る app は IoT ノードおよびクラウドサーバと同じ PC 上に実装しているため、想定されるクラウドサーバとの通信時間よりも短くなると考えられる。そのため、図5中のオレンジ色で示したクラウドサーバを経由する経路に対し意図的に通信遅延を加えた。ping コマンドを用いて北海道、沖縄県のホームページがあるサーバまでの時間を計測したところ、それぞれ平均 40ms, 60ms であった。そこでクラウドサーバに日本全土のデータが集められると仮定し、オレンジの矢印に対し片道 25ms 分の通信遅延を加えた。

図6において、左側のフォグ4がフォグノードで4つのルールを適用した場合の結果、右側が提案手法の結果である。なお提案手法については、負荷が0の場合全てフォグノードで、100の場合は全てクラウドサーバで推論を行なっている。負荷が100の場合、フォグノードで推論のためのモデルを構成する

必要がないため、他に比べ時間が短くなっている。

図6より全体を通して、クラウドサーバを経由することで app までの到着時間が遅くなっていることが確認できる。現状の推論タスクは比較的軽量であるが、今後知識を活用したより洗練された推論を行う中でフォグノードの負荷は増加し、青色で示したフォグノードからの通知時間はオレンジの色のクラウドからの通知時間に近づくと考えられる。そのためできるだけクラウドを経由せずフォグノードで推論を行う方が望ましいが、一方でフォグノードに負荷がかかっている際には通常よりも通知に時間を要してしまうため、一部クラウドで推論した結果を受け取る方が良いということになる。少なくともフォグノードを用いて分散推論を行う際は、フォグノードの負荷を考慮して処理を変更した方が良いことが予備実験により示された。

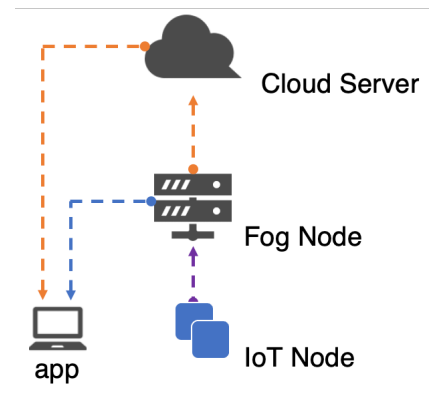


図5 通知時間の実験で使用したトポロジー

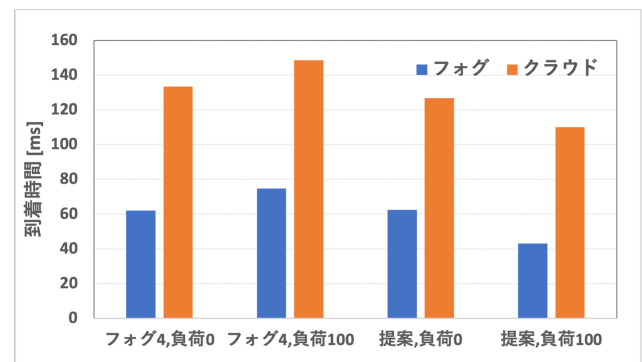


図6 推論結果の通知にかかる時間の比較

6 まとめ

本稿では、フォグコンピューティングにおいて、各フォグノードの負荷に応じて推論処理をクラウドとの間で分散する分散 RDF 推論手法を提案した。予備実験により、負荷を考慮することの必要性及び分散機構の導入による効果が確認できた。

今後の課題としては、推論内容の充実、フォグノードの数を増加させたより大きなトポロジーでの実験、推論ルールの分散を行う際に各ルールの推論コスト及び適用順序を考慮したより効果的な分散の実現などが挙げられる。また、推論に使用する知識ベースが巨大でフォグノードに配置できないことも考えら

れるため、使用する頻度などに基づいて知識の配置を上位ノードとの間で分散することも検討している。

謝 辞

この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務 (JPNP20006) の結果得られたものです。また本研究は、SKY (株) (CPI03114) による共同研究経費の助成を受けたものです。

文 献

- [1] 総務省. 令和 3 年度版情報通信白書 IoT デバイスの急速な普及. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/html/nd105220.html>.
- [2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, Sateesh Addepalli. Fog Computing and Its Role in the Internet of Things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13-16, 2012.
- [3] Xiang Su, Pingjiang Li, Jukka Rieki, Xiaoli Liu, Jussi Kiljande, Juha-Pekka Soininen, Christian Prehofer, Huber Flores, Yuhong Li. Distribution of Semantic Reasoning on the Edge of Internet of Things. 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 1-9, 2018.
- [4] Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, Thierry Monteil. EDR: A generic approach for the distribution of rule-based reasoning in a Cloud-Fog continuum. Semantic Web journal, vol. 11, no. 4, pp. 623-654, 2020.
- [5] Davide Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, Michael Grossniklaus. C-SPARQL: A Continuous Query Language for RDF Data Streams. International Journal of Semantic Computing Vol. 04, No. 01, pp. 3-25, 2010
- [6] Josiane Xavier Parreira Manfred Hauswirth Danh Le Phuoc, Minh Dao-Tran. A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. In Proceedings of the 10th International conference on The Semantic Web, pp. 370-388, 2011.
- [7] W3C. RDF 1.1 Concepts and Abstract Syntax. <https://www.w3.org/TR/rdf11-concepts/>.
- [8] Stefan Bischof, Athanasios Karapantelakis, Cosmin-Septimiu Nechifor, Amit Sheth, Alessandra Mileo and Payam Barnaghi. Semantic Modeling of Smart City Data. Position Paper in W3C Workshop on the Web of Things: Enablers and services for an open Web of Devices, 2014.
- [9] MQTT.org. MQTT: The Standard for IoT Messaging. <https://mqtt.org/>.