

## fogcached-ros : ハイブリッドメインメモリ KVS サーバミドルウェアの提案

東 晃希† 石綿 陽一‡ 大川 猛§ 菅谷 みどり†

芝浦工業大学工学部† Ales 株式会社‡ 東海大学情報通信学部§

## 1. 研究背景と課題

近年, 大量の IoT デバイスから収集されるセンサーデータのクラウドサービスへの保存が想定されるようになった. クラウドへのデータ保存は, 処理の集中, 過負荷の課題がある. また, 無線等のネットワークの遅延が課題となる. そこで, 課題解決を目的とし, クラウドより近距離にエッジサーバを配置する, エッジコンピューティングが期待されている. 近距離で動作するエッジサーバには高い即応性と信頼性が求められる. 応答性向上技術の一つとして, エッジにキャッシュサーバを配備し, クラウドまでのデータキャッシュをエッジサーバ上で効果的に行う仕組みが提案されている[1].

小沢らはさらに, キャッシュサーバに, DRAM に加え, 大容量の NVMM を利用可能とするハイブリッド型メインメモリ KVS の仕組みを fogcached として提案した[2]. fogcached は, In-Memory Key-Value Store である memcached の LRU を拡張し, Dual-LRU とアクセス数により, 二種類のメモリを自動で切り替える仕組みを実装した. さらに, 評価にて Intel が提供する DCPM(Intel Optane Data Center Persistent Memory)を用いて, エッジサーバの応答性が従来の仕組みよりも高い応答性を得られることをシミュレーションで示した. DCPM は NVM で, DRAM の約 4 倍のレイテンシがあるが, アクセス数が少ないデータを DCPM に移動させることで, 応答性は高く保ちつつ信頼性を達成することを示した.

しかし, fogcached はシミュレーションでしか評価されておらず, 実際のアプリケーションに対しての有効性が明らかではない問題がある.

## 2. 提案

## 2.1. 目的

我々は, fogcached を実際のエッジサーバとして, クラウドとの中間におき, 実用的に利用するために, 課題を次の 2 つにまとめた. (1) 実際のアプリケーションでの評価が不十分である, (2) 信頼性を考慮したクラウドとの連携が実施されていない. 本研究では, エッジを用いた高い応答性と信頼性を要するロボットアプリケーションである SLAM (Simultaneous Localization and Mapping) を用いる. SLAM は, ロボット等の移動体において, センサから環境情報を収集し, そのデータ解析により自己位置推定, 環境地図作成を同時に行う技術である. SLAM を活用する

fogcached-ros : Hybrid Main Memory KVS server Middleware  
Koki Higashi† Yoichi Ishiwata‡ Takeshi Ohkawa§ Midori Sugaya †  
Shibaura Institute of Technology† Ales Corp.‡ Tokai University§

ことで, ロボットのみならず, 自動運転など広く移動体の自律移動を実現できる. これら SLAM の多くは, ミドルウェアである ROS (Robot Operating Systems) 上に実装されている. SLAM を用いた fogcached を利用するためには, 現状では SLAM のそれぞれのアプリケーションを改変する必要がある.

本課題を解決するため本研究の目的を, fogcached-ros という新しいミドルウェアの設計と実装とした. 本提案, エッジサーバの応答性と信頼性の有効性を評価する.

## 2.2. 提案システム

提案システムの全体図を図 1 に示す. 提案システムにはクライアント, エッジサーバ, クラウドが存在する. クライアントとして既存のオープンソース ROS アプリケーションである turtlebot3 と SLAM (gmapping) を用いた. またエッジサーバには fogcached を用いた. エッジサーバの ROS アプリケーション fogcached-ros はクライアントとの通信, fogcached との通信を行う. クラウドはエッジサーバとのみ通信し, データを保存する. クラウドとの通信はエッジサーバからデータを取得する際に fogcached にデータがない場合, または, クラウドからデータをとりだす場合の 2 つのタイミングで行われる. また, エッジサーバへの通信とクラウドへの通信を含めたすべての通信を ROS コンポーネントで統一する.

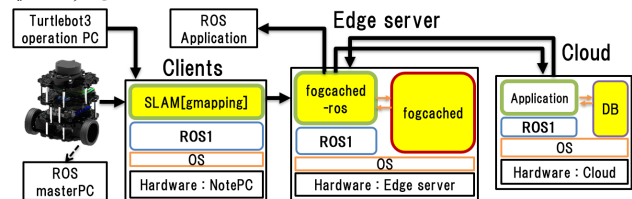


図 1 提案システム

## 2.3. SLAM とサーバの連携の設計と実装

図 2 のように, 従来の ROS の通信モデルでは memcached と通信する手段はない.

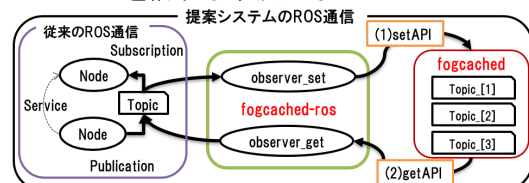


図 2 サーバ連携図

また, topic は基本的に Node により通信時に上書きされ最新の情報のみ残る. 本研究では ROS を拡張し, fogcached-ros を設計した. fogcached-ros は fogcached と通信を行うための 2 種類の API によって, 過去のデータを自動で fogcached 上の NVM に保存で

きる。DRAMにデータを保存することで応答性を向上させると同時にNVMに保存することで永続的にデータを保存することができるため信頼性を実現できると考えた。

### 3. API 評価

本実験では提案システムで使用されるAPIのオーバーヘッドが高くないかどうかを確認する。使用したクライアントはSLAM (gmapping)とシミュレーションソフトウェア gazebo 上の turtlebot3 である。また、エッジサーバ上で fogcached と ROS アプリケーション fogcached-ros を実行する。

libmemcached は、データの送受信にかかる時間である。serialization は ROS の serialize と deserialize のオーバーヘッドを指す。すべての topic で serialization はともにオーバーヘッドは 1ms 未満となったが、データサイズが大きくなるにつれ serialization にかかる時間の割合が増える(図 3)。送信周期は topic によって大きく異なる。map は 1000 秒単位で考慮すると、map のオーバーヘッドは極僅かになる(図 4)。Message サイズが大きい場合、オーバーヘッドは serialization の影響を受けるが、送信頻度が低い topic のオーバーヘッドは無視できる。したがって、本提案の API のオーバーヘッドは低い。また、送信頻度が高い topic のオーバーヘッドは大きくなるが、API 側では送信頻度を変えられないため本論文では考慮しない。

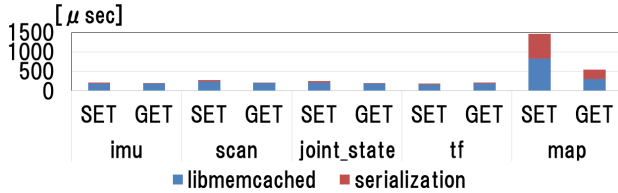


図 3 1 回あたりの API オーバーヘッド

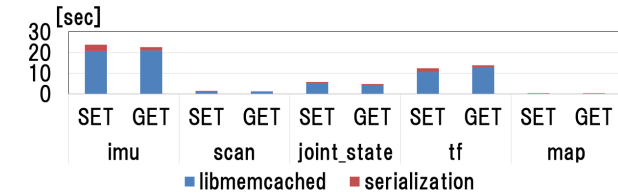


図 4 1000 秒あたりの API オーバーヘッド

### 4. クラウド評価

本実験ではレイテンシ分布と DRAM と DCPM のヒット数の割合を調査するために、複数の条件を設けて下記の構成で評価実験を行った(図 5)。実験構成として memcached-1.5.16 を疑似的なクラウドとした。また、クラウドを想定しているため、通信遅延を入れて実験を行った。基準パターンとして ROS topic の ID の数を 1,000,000、Set:Get の呼び出し割合を 1:9、DRAM 4GB、DCPM 16GB という条件を用意した。このうち、実験(1)DRAM/DCPM メモリ比率、実験(2)ID の数について条件を変えて 2 つの比較を行った。実験(1)では、メモリデバイスの応答性の差異による ROS システム全体の応答性への影響の調査と NVM による信頼性の調査、実験(2)での ID の最大数

の比較は fogcached に残るデータ数の増減による応答性への影響の調査が目的である。

実験(1)では、メモリ量での応答性の差は大きく存在しないことから、メモリ量に関係せずに利用できることがわかった。また、DRAM と DCPM 間の移動が行われたことにより、データの信頼性が確保されていることがわかった(図 6)。実験(2)は ID の数が減ると基準に比べてエッジでのアクセス数がクラウドでのアクセスに比べて増加した(図 7)。このことから、ID の数を増減することで応答性を調整できる。

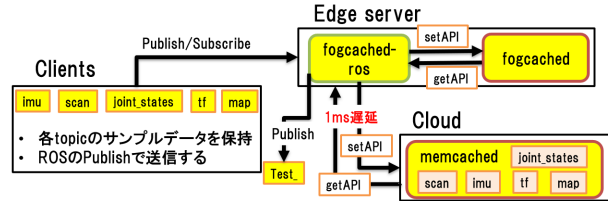


図 5 クラウド評価実験構成

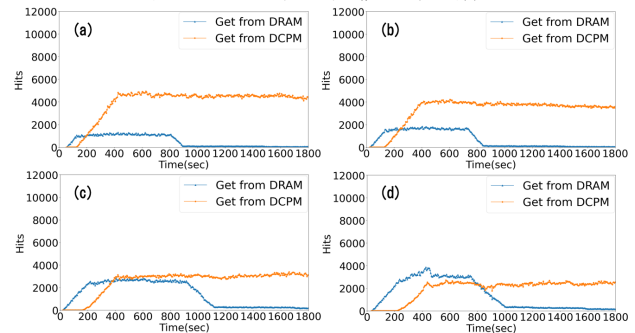


図 6 DRAM と DCPM からの各 hit 数推移

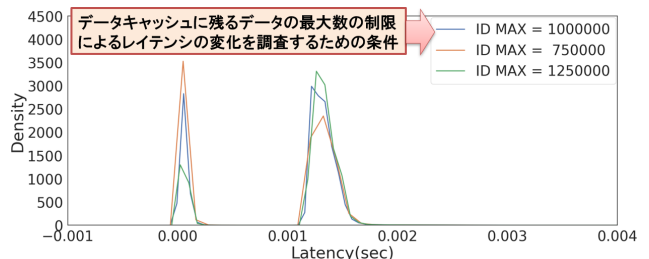


図 7 ID の最大数の比較(imu)

### 5. 結論

本論文では fogcached-ros を提案し、実装、評価した。その結果、提案した fogcached-ros を用いることで、SLAM アプリケーションという実用的なアプリケーションを応答性と信頼性を両立させて利用することがわかった。

### 謝辞

本研究は、JST,CREST,JPMJCR19K1 の支援を受けたものです。

### 参考文献

[1] Michele de donno, Koen Tange, Nicola Dragoni, "Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog", IEEE Access, Vol 7, pp. 150936-150948, Oct. 2019.  
 [2] Kouki Ozawa, Takahiro Hirofuchi, Ryousei Takano and Midori Sugaya, "DRAM-NVM Hybrid Memory - Based KVS server for Edge Computing", International Conference on Edge Computing, pp. 50-62, Sep. 2020.