

ヘテロジニアス環境における FaaS を考慮した MEC-RM ベースのスケジューリング手法の提案

蓮原裕太^{†1} 李彦志^{†1} 菅谷みどり^{†1}
芝浦工業大学^{†1}

1. 背景

近年、内閣府が推進する Society5.0 では IoT 機器やロボットなどが人工知能を用いたイノベーションによって新たな価値が生まれる社会が期待されている[1]. 実現には、IoT 機器やロボットがセンシングした大量のデータを、人工知能を用いて処理することが期待されており、クラウドを用いたスケーラブルなシステムが活用されている。

しかし、クラウドとの通信は遅延時間の問題[2]や世界の IoT デバイス数の推移および予測[3]から、クラウドへのトラフィック量やデータプロセッシング量は増大する。これを解決するために、GPGPU や FPGA 等の高性能なアクセラレータを統合したヘテロジニアス環境におけるマルチアクセスエッジコンピューティング (MEC) システムが提案されている。MEC において応答性の向上及び複数の異なるアクセラレータを透過的かつ効率的に利用可能にする資源管理システムとして MEC-RM が提案されている[4].

MEC-RM は図 1 に示すように、計算資源利用者 (リクエスター) と、計算資源提供者であるワーカーの間で最適な資源管理を行う機能を提供する。リクエスターからの計算処理リクエストをもとに RM はジョブを作成、ワーカーの割り当てを行う。

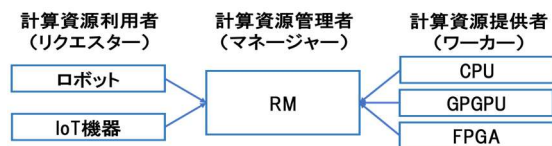


図 1 システム概要

2. 課題

MEC-RM は、応答性の向上のために基地局ごとに MEC サーバーを設置することを想定している。常にリクエスターは同じ基地局と通信するとは限らず、アプリケーションはリクエスターからのリクエストを受け取った MEC サーバーで実行することが求められる。そのため、実行場所を変化させることができる FaaS (Function as a Service) の活用が重要である。

FaaS においては関数 (アプリケーション開発者が作成したプログラム) を実行するランタイムはリソース状況や関数の実行から一定時間経過するとシャットダウンし、リソースを開放する。そのため新たにランタイムを起動しよう

としたときにランタイムの初期化 (コールドスタート) オーバーヘッドにより応答性が低下する[5]. リクエストごとに実行される関数は逐次的にコールドスタートすることを防ぐ必要があり、動作中のランタイムを持つワーカーにジョブを割り当てるための局所性を考慮したスケジューリングにより応答性は改善される[6].

しかし、MEC-RM においてはジョブに対してタグが一致したワーカーをキューの先頭から割り当てるスケジューラが提案されている[4]. ジョブとワーカーは各自タグ情報を持ち、ジョブに対して最初にタグ情報が一致したワーカーを割り当てるが、先に述べた局所性の考慮を欠いたスケジューリングを行うため、過度にコールドスタートを起こし、応答性が低下する課題がある。

3. 提案

3.1 局所性を考慮したスケジューリング手法 (提案 1)

先に述べた課題を解決するために、ジョブを初期化が不要となるワーカーに優先的に割り当てる手法を提案する。これにより、コールドスタートによるオーバーヘッドを削減することができるため、応答性が向上すると考えた。本提案を局所性を考慮したスケジューリング手法とする。

3.2 シミュレーション手法の設計と実装

ワーカーは複数のランタイムを起動し、RM は複数のワーカーとジョブをスケジュールする。使用するシミュレータではリクエスターを排除し、後述する関数のジョブを生成する機能を追加した。

応答性に影響するコールドスタート率を評価指標とした。コールドスタート率とは、ジョブをワーカーに割り当てた際にコールドスタートを発生させた割合とする。ワーカーのジョブ要求時に RM に送るタグ情報に最後に実行したランタイムの情報を付与する。既存のタグに加えてランタイムも一致するワーカーにジョブを割り当てることで、ランタイムの初期化を回避する。該当するワーカーがない場合はキューの先頭にあるワーカーを割り当てる。

3.3 評価

5G の利用シナリオとして最大上り通信速度は 10Gbps 程度が求められている[7]. アプリケーションとして 1 枚当たり 125KB 程度の画像を用いると想定し、評価では 1 秒あたり 1 万個のジョブを用いる。呼び出される関数は Azure

^{†1} 芝浦工業大学
Shibaura Institute of Technology, 3-7-5 Toyosu, Koto-ku, Tokyo 135-8548, Japan

Functions の公開データセットから作成した確率分布をもとに 10 個作成し、呼び出し確率の低い順に 1~10 の番号を付けた。図 2 表 1 のように環境を構築した。

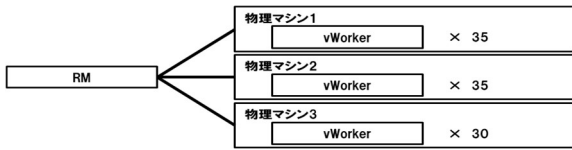


図 2 実験環境模式図

表 1 実験環境

	CPU	メモリ
RM	Intel Corei7 12700kf	DDR4-3600 32GB
物理マシン 1	AMD EPYC 7543	DDR4-3200 256GB
物理マシン 2	AMD EPYC 7543	DDR4-3200 256GB
物理マシン 3	AMD Ryzen9 3950x	DDR4-3600 32GB

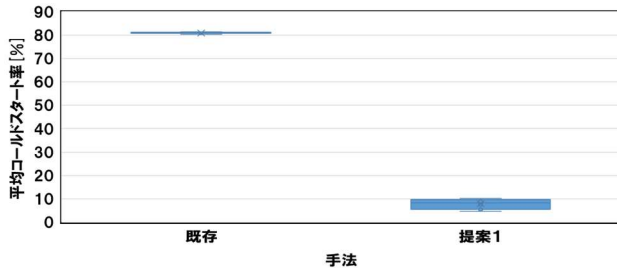


図 3 既存手法と提案手法 1 の平均ランタイム初期化率

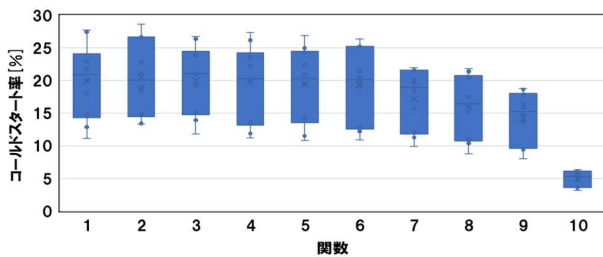


図 4 関数ごとのコールドスタート率 (提案 1)

既存手法と比較し、全関数の平均コールドスタート率をおよそ 70%削減した (図 3)。しかし、図 4 に示したように、呼び出し確率の低い関数のコールドスタート率が大きくなる結果となった。

3.4 待機率を考慮したスケジューリング手法 (提案 2)

関数の呼び出し頻度は関数ごとに異なり、それにより待機中のランタイム数にも反映される。

キュー内に存在するワーカーで待機中の各ランタイムの総和を r_1, r_2, \dots, r_n とし各関数に重み w_i (式 1) を定義する。

提案 1 でジョブに対するランタイムが存在しなかった場合に、重み w_i を用いてランダムに割り当てるワーカーを決定する。これにより、関数の呼び出し頻度を反映し、確率的にコールドスタートさせる関数を決定することで、呼び出し頻度の低い関数のコールドスタートを回避しつつ、頻度の高い関数ばかり選択されることを回避すると考える。

$$w_i = \frac{r_i}{\sum_{j=1}^n r_j} \quad (1)$$

3.5 評価

図 5 に提案 2 の関数ごとのコールドスタート率の結果を

示した。さらに提案 1 (図 4) と提案 2 の結果について、各関数のコールドスタート率の最大値と最小値の差を算出し、その値を比較したところ、提案 2 は提案 1 に比べ、その差を約 90%削減できた。また、図 6 に示すように平均応答時間を約 29%削減し、応答性が向上した。

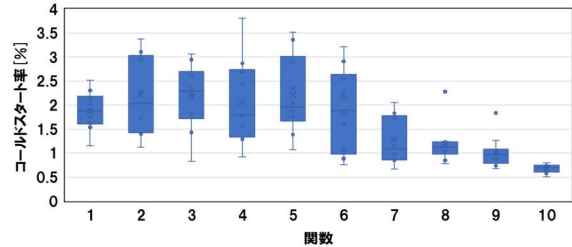


図 5 関数ごとのコールドスタート率 (提案 2)

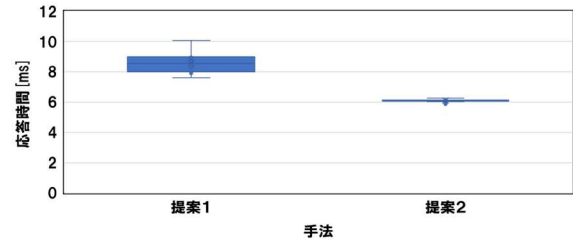


図 6 提案手法の応答時間の比較

4. 結論

本研究ではヘテロジニアス環境における FaaS を考慮した MEC システムの応答性向上を目的として、2 つのスケジューリング手法を提案した。提案 1 では局所性を考慮し、既存に比べ 70%の改善、提案 2 にてコールドスタート率の差を提案 1 に比べて 90%削減、平均応答時間を約 29%削減し、有効性を示した。ただし、待機中のワーカー数に大きな影響を受けるため、今後は予測されるジョブ数に対するワーカーの要求数の決定方法について検討する。

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] 内閣府. Society 5.0- 科学技術政策 - 内閣府. https://www.8.cao.go.jp/cstp/society5_0/. (2023 年 12 月 26 日参照).
- [2] 並木 美太郎, 天野 英晴. MEC FPGA クラスタにおける TCP/IP の実現. 情報処理学会研究報告 Vol.2021-OS-151, No.1, 2021
- [3] 総務省. IoT デバイスの急速な普及. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r02/html/nd114120.html>. (2023 年 12 月 26 日参照).
- [4] LI YANZHI, 菅谷みどり. マルチ FPGA・GPGPU 混在環境向け 資源管理システムの提案. 情報処理学会研究報告 IPSJ SIG Technical Report, Vol.2022-HPC-185 No.2, 2022/7/27
- [5] AWS. Lambda 拡張機能 API. https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/runtimes-extensions-api.html. (2023 年 12 月 26 日参照).
- [6] Alexander Fuerst, Prateek Sharma. Locality-aware Load-Balancing For Serverless Clusters. HPDC '22: The 31st International Symposium on High-Performance Parallel and Distributed Computing. pp. 227-239, 27 June 2022.
- [7] 総務省. 5G の利用シナリオと主な要求条件. <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r02/html/nd111310.html>. (2023 年 12 月 26 日参照).