

1 はじめに

Mahadev らが提案したクラウドレット [1] は、ユーザと同一の LAN 内に設置されたコンピュータ群であり、シングルホップで接続可能である。しかし、クラウドレットは規模が小さく、従来のクラウドコンピューティングやエッジコンピューティングと比較して計算容量が少ないため、サービスの維持が課題となっている。そのため、クラウドレットの提供者が持続的なサービスを提供するためには、限られた資源を効率的に利用しつつ、提供者自身が満足できるリターンを得られるような処理手法を考える必要がある。

従来の研究では、平均遅延時間に許容遅延を設けたうえで、クラウドレット間のオフロードを行うことで、資源の効率化だけでなく、システムの QoS を意味する遅延をクラウドレットの満足度に組み込み、これをゲーム理論で解く手法を提案している [2]。しかし、許容遅延を各クラウドレットの利用率によって重み付けしているため、オフロード割合の低いジョブの遅延が許容遅延を上回る結果になっている。そこで、遅延を重み付けせず、公平に評価したオフロードに着目した既存研究 [3] では、クラウドレット間の利用率の差を最小化する値がナッシュ均衡解となるため、従来の許容遅延内負荷分散法より高い平均利得が得られている。一方で、これらの手法では、ジョブの処理をすべて到着順である FCFS でモデル化している。FCFS では、任意の処理の長いジョブをオフロードするかどうかによって全体の処理遅延が大きく変わるため、オフロード選択に限界がある。

本稿では、オフロード後の到着流に着目し、非割込み優先規律に基づいて導出した処理遅延を用いてオフロードしたジョブを優先する処理法 (OP)、オフロードしていないジョブを優先する処理法 (ONP) の 2 つを提案する。これらを既存手法 [3] における FCFS による処理法 (FCFS) の処理遅延と比較するため、異なる遅延による 3 種類の非協力ゲームにおける利得関数を新たに定義し、ナッシュ均衡の平均利得によって、提案モデルの有効性を示す。

2 提案手法

2.1 システムモデル

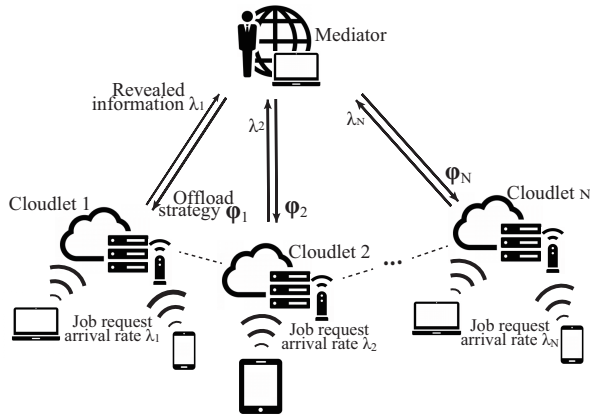


図 1: システムモデル
Fig1. System Model.

図 1 に、本手法におけるシステムのオフロードモデルを示す。本稿では、ユーザから無線環境下で送られる様々な計算処理をジョブと定義し、これらを処理するために LAN 内に設置されたクラウドレット $i \in \{1, 2, \dots, N\} = C$, $N \geq 2$ について考える。すべてのクラウドレットは有線で互いに通信可能であり、管理者 (Mediator) を通して共有される互いの平均到着率の情報 (Revealed Information) をもとに、各クラウドレットが自身のジョブのオフロード割合を決定する。

クラウドレット i に到着するユーザから送信されるジョブの到着率が、平均 λ_i ポアソン分布に従うと仮定し、各ジョブの処理率は平均処理時間 b を用いて平均 $\mu = \frac{1}{b}$ の指数分布に従うと仮定する。クラウドレット i に到着したジョブのオフロード先を決めるため、割合ベクトル $\varphi_i = (\varphi_{i1}, \varphi_{i2}, \dots, \varphi_{iN}) \in \mathbb{R}^N$, $\varphi_{ij} = [0, 1] \in \mathbb{R}$, $\sum_{j=1}^N \varphi_{ij} =$

$1, i, j \in C$ を定義する。この時、 φ_{ij} は i 番目のクラウドレットが j 番目のクラウドレットにオフロードするジョブの割合を示す。したがって、 $\varphi_{ii}\lambda_i$ のジョブがオフロードされずにクラウドレット i で処理され、 $\varphi_{ij}\lambda_i$ のジョブがクラウドレット j へとオフロードされる。

本稿におけるクラウドレットの満足度の指標は、オフロードによって変わるクラウドレットの利用率の変化量と、オフロード後の平均稼働率を示す利用率の和で表現する。また、システムの QoS は遅延が許容時間以下であれば十分であることを考慮し、クラウドレット i に到着するジョブの平均遅延時間が許容遅延 D_{rq} を満たせばその分だけクラウドレットの満足度が増加し、満たさなければその分に応じて満足度が減少するような値 E_i を、既存手法 [3] に基づいて利用率で重み付けせず導入する。これにより、クラウドレットはできる限り許容遅延を満たすようなオフロードを行うため、QoS の維持とクラウドレットの満足度を直結させることができる。

これらを用いて、FCFS を従来の M/M/1 待ち行列モデル、提案手法である OP と ONP を複数到着流を持つ非割込み優先規律を用いた M/M/1 待ち行列モデルでモデル化し、非協力ゲーム理論を用いてナッシュ均衡を導出することで、処理順序の違いによるオフロード選択への影響を明らかにする。

2.2 待ち行列モデル

個々のクラウドレット間の平均往復遅延を t_{ij} , $i \in C, j \in C \setminus \{i\}$ と定義する。各クラウドレットが FCFS で処理を行うとき、パラメータ $\lambda_i, \mu, \varphi_i$ よりリトルの公式を用いて各ジョブの割合に対応する平均処理遅延 T_{ii}^{FCFS} , $i, j \in C$ を導出できる。したがって、オフロードされずに処理されるジョブ $\varphi_{ii}\lambda_i$ がクラウドレット i に到着してからユーザへと返されるまでの平均遅延 T_{ii}^{FCFS} は、次式で定義される。

$$T_{ii}^{FCFS} = \frac{1}{\mu - \sum_{j=1}^N \varphi_{ji}\lambda_j}. \quad (1)$$

また、オフロードされたジョブ ($j \neq i$ の場合) の平均遅延 T_{ij}^{FCFS} は次のように表される。

$$T_{ij}^{FCFS} = t_{ij} + \frac{1}{\mu - \sum_{k=1}^N \varphi_{kj}\lambda_k}. \quad (2)$$

次に、各クラウドレットが到着するジョブをその到着流のクラス $m \in \{1, 2, \dots, N\}$ に応じて $1 < 2 < \dots < m < \dots < N$ の順に処理を行うとしたとき、それぞれのクラスの処理遅延を考える。クラウドレット i からクラウドレット j にオフロードされる任意のジョブの到着流をクラス m としたとき、そのクラスのジョブの利用率を ρ_m と表す。処理率が指数分布に従うことを考慮すると、クラウドレット i からクラウドレット j に到着するクラス 1 の到着流のジョブの平均処理時間 $T_{ij,1}$ は、サービス時間の 2 次積率 $b^{(2)}$ を用いて以下のように求められる [4]。

$$\begin{aligned} T_{ij,1} &= \frac{\sum_{m=1}^N b^{(2)} \rho_m}{2b(1 - \rho_1)} + b \\ &= \frac{\sum_{m=1}^N \rho_m b}{(1 - \rho_1)} + b. \end{aligned} \quad (3)$$

クラス 2 以上の到着流の平均処理時間は、到着時点でのクラス m より先に処理されるジョブの平均待ち客数と、待ち中に新たに到着するクラス m より先に処理されるジョブの平均到着数に依存する。したがって、任意のクラス M ($2 \leq M \leq N$) の平均処理遅延 $T_{ij,M}$ は以下のよう

められる。

$$T_{ij,M} = \frac{\sum_{m=1}^N b^{(2)} \rho_m}{2b(1 - \sum_{m=1}^{M-1} \rho_m)(1 - \sum_{m=1}^M \rho_m)} + b$$

$$= \frac{\sum_{m=1}^N \rho_m b}{(1 - \sum_{m=1}^{M-1} \rho_m)(1 - \sum_{m=1}^M \rho_m)} + b. \quad (4)$$

2.3 ゲーム理論

本稿では、異なる処理順序を考慮した3種類の非協力ゲームにおいて、クラウドレット i が選択する戦略となるオフロード割合 φ_i と、それ以外のクラウドレットの戦略となるオフロード割合 $\varphi_{-i} = (\varphi_1, \dots, \varphi_{i-1}, \varphi_{i+1}, \dots, \varphi_N)$ によって変化する利得関数 $U_i(\varphi_i, \varphi_{-i})$, $i \in C$ を定義し、ナッシュ均衡となる最適オフロード割合 φ_i^* を決定する。

クラウドレット i の利得関数 U_i は、 E_i とクラウドレット i に到着するジョブを処理する上で得られる $\sum_{j=1, j \neq i}^N \varphi_{ji} \lambda_j b + \sum_{j=1}^N \varphi_{ji} \lambda_j b$ から、 $\sum_{j=1, j \neq i}^N \varphi_{ji} \lambda_j b$ を差し引いた満足度の和を示す。また、各ジョブの平均遅延時間は処理順によって異なるため、許容遅延に基づく満足度の値 E_i は式(1)~(4)を用いて次の3通りで表される。

- FCFS: $E_i = D_{rq} - T_{ii}^{FCFS} + \sum_{j=1, j \neq i}^N D_{rq} - T_{ji}^{FCFS}$.
- OP: $E_i = D_{rq} - T_{ii,N} + \sum_{j=1, j \neq i}^N (D_{rq} - T_{ij,m} - t_{ij})$.
($1 \leq m \leq N-1$)
- ONP: $E_i = D_{rq} - T_{ii,1} + \sum_{j=1, j \neq i}^N (D_{rq} - T_{ij,M} - t_{ij})$.

以上を用いて、クラウドレット i の利得関数 U_i は以下のように与えられる。ただし、 α, β, γ は各項の重みを表す。

$$U_i(\varphi_i, \varphi_{-i}) = -\alpha \sum_{j=1, j \neq i}^N \varphi_{ij} \lambda_j b + \alpha \sum_{j=1, j \neq i}^N \varphi_{ji} \lambda_j b$$

$$+ \beta \sum_{j=1}^N \varphi_{ji} \lambda_j b - \gamma E_i. \quad (5)$$

本稿では定常状態のシステムのみを考えるため、オフロード後の利用率が1を超えてはならない。よって、利得関数が $[\mu - (1 - \sum_{j=1, j \neq i}^N \varphi_{ij}) \lambda_i - \sum_{j=1, j \neq i}^N \varphi_{ji} \lambda_j] \geq 0$, $i, j \neq i \in C$ を満たす中で、ナッシュ均衡解 φ^* を導出する。

3 数値解析

本解析では、 $N=2$ のクラウドレット間でのオフローディングについて評価を行った。

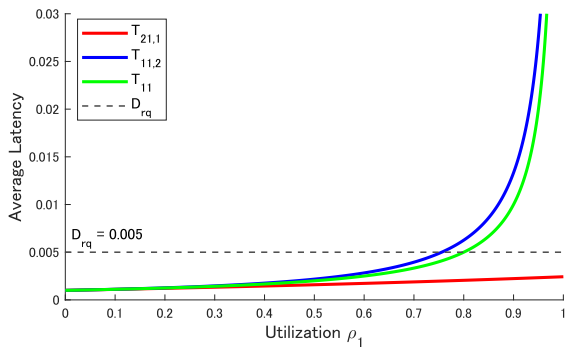


図2: 理論的な遅延 T_{11}^{FCFS} , $T_{21,1}$, $T_{11,2}$ の比較
Fig2. Theoretical Latency Comparison of T_{11}^{FCFS} , $T_{21,1}$, $T_{11,2}$.

まず、オフロードが必要となるトラフィック負荷の調査を行うため、FCFS と OP, NOP の遅延 T_{11}^{FCFS} , $T_{21,1}$, $T_{11,2}$ を比較する。ただし、 T_{11}^{FCFS} は、他クラウドレットとオフロードせず FCFS で処理を行った場合における遅延、 $T_{21,1}$, $T_{11,2}$ は、到着するジョブのうち3割がクラウドレット2からのジョブで、クラス1となる場合のクラウドレット1における各クラスの遅延である。平均処理時間 $b = 1.0 \times 10^{-3}$ のクラウドレット1の利用率 $\rho_1 = \lambda_1 b$ を0から1の間で変化させ、これら3つの遅延を図2に示す。

許容遅延を $D_{rq} = 5 \times 10^{-3}$ [s] とすると、クラス1の遅延は利用率に関わらず許容遅延以下でほぼ一定である。また、利用率が0.75以上でクラス2のジョブとFCFSで処理されるジョブの両方が許容遅延を超えていることが確認できる。したがって、本環境では利用率0.75より高い領域でオフロードが必要になることがわかる。

上記内容を踏まえて、本解析では、初期利用率が0.75以上となるようなクラウドレットが存在する間でのオフローディングについて、評価を行った。基本的なパラメータは前述のとおりとし、 $\alpha = 1, \beta = 1, \gamma = 100$ と設定することで、利得関数の各項が同等の価値となるようにする。そのもとで、クラウドレット間の往復遅延 $t_{ij} = 2 \times 10^{-3}$ [s], クラウドレット1の到着率を $\lambda_1 = 0.5$ [jobs/s] で固定したとき、クラウドレット2のオフロード前の利用率 ρ_2 を変化させ φ^* を導出した。このとき、処理順の違いにおけるクラウドレットの平均利得 $\bar{U} = \frac{1}{N} \sum_{i=1}^N U_i$ を、図3に示す。

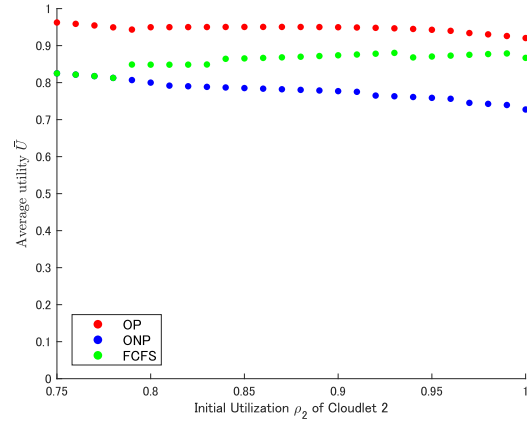


図3: クラウドレットの平均利得
Fig3. Average Utility Comparison.

結果から、提案手法であるOPは他処理順と比較するとつねに上回っていることが確認できる。反面、ONPは利用率の増加に伴い緩やかに減少しており、オフロードジョブを優先しない処理順は有効ではないことが示された。本ゲームの利得は、QoSを維持したクラウドレットの満足度を表しており、平均的に高い利得を得られているOPは、オフロードが必然である本解析環境において他手法より優位であり、提案手法の有効性を示すことができた。

4 おわりに

本稿では、クラウドレットのオフロード後の到着流に着目し、異なる処理順序を考慮した場合の最適なオフロード割合を導出した。結果として、オフロードしたジョブを優先する処理法によって、各クラウドレットの平均的な満足度が最も高くなることが示された。

今後の展望として、シミュレーションを用いた動的な処理順変更を考慮したオフローディングが考えられる。

参考文献

- [1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol.8, no.4, pp.14–23, 2009.
- [2] S. Mondal, G. Das, and E. Wong, "A game-theoretic approach for non-cooperative load balancing among competing cloudlets," *IEEE Open Journal of the Communications Society*, vol.1, pp.226–241, 2020.
- [3] Y. Yokota, and S. Miyata, "Game Theoretic Approach for Non-Cooperative Load Balancing between Local Cloudlets," *Nonlinear Theory and Its Applications, IE-ICE*, vol.15, no.2, pp.473-484, 2024.
- [4] T. Takine, "Lecture paper on Communication Network Engineering (Ver. 3.15)," Osaka University, 2024. [Online]. Available: <http://www2b.comm.eng.osaka-u.ac.jp/takine/tmp/v3-15.pdf>

代表的な研究実績

Y. Yokota, and S. Miyata, "Game Theoretic Approach for Non-Cooperative Load Balancing between Local Cloudlets" *Proc. of NOLTA'23*, paper ID (6207), September 2023.