Game Theoretic Approach for Non-Cooperative Load Balancing between Local Cloudlets

Yuki Yokota[†], Sumiko Miyata[‡]

Shibaura Institute of Technology, 3–7–5 Toyosu, Koto-ku, Tokyo, 135–8548 Japan

Email: ma23203[†], sumiko[‡]@shibaura-it.ac.jp

Abstract— Cloudlet is a cluster of computers which exists within the same Local Area Network (LAN), and can be accessed by 1 hop by all the users connected to the network. Compared to the usual edge computing system in which the servers are located on base stations, cloudlets enable real-time communication with less network delay. On the other hand the lack of computational ability is another problem to be solved. Conventional research considers the offloading of jobs between cloudlets to reduce the variance of workload, while limiting the total latency of each cloudlet below acceptable latency. However, the total latency is calculated based on the fraction of offloading jobs which results in the offloading decision to be unfair for smaller fraction of jobs in the system in terms of latency. In this paper, we propose a game theoretic load balancing method to consider a fair latency among each fraction of offloading jobs and evaluate the effectiveness of the load balancing model by comparing the difference in utilization between cloudlets.

1. Introduction

Conventional mobile cloud computing technologies have been developed to compensate for the problems faced by mobile terminals such as limited power supply, storage capacity, and computing power. In this context, the increasing number of applications that require average latency of approximately 10-100 ms, such as virtual reality, online gaming, and tele-operation, has led to the need for solutions using edge computing techniques to reduce latency [1]. Cloudlet proposed by Mahadev et al. [2] is a cluster of computers that reside in the same LAN as the user, which can be connected via a single hop. Unlike edge computing in large scale network, where a computational server is installed at a base station, cloudlet can communicate with the user with smaller latency by installing them at a router or other location within the LAN.

One problem with cloudlet is that they are small in scale and have significantly lower processing power than conventional cloud computing or edge computing in larger scale network. Therefore, the limited resources of cloudlets must be used efficiently. Conventional research in wireless access networks has focused on the problem of cloudlet placement to distribute the resources efficiently, such as the method [3]. In distributed computing systems, it is necessary to determine which computer will process each job that arrives in order to distribute the load across the entire system [4]. For this purpose, methods such as [5] and [6] have been proposed to determine the allocation of job requests from mobile terminals to cloudlets based on the average delay time. However, considering the complex movement patterns of mobile terminal users and the randomness of arrival rates, it is inevitable to have highly concentrated cloudlet as well as empty cloudlet. Therefore, it is necessary to consider a method to distribute jobs among cloudlets to enable efficient use of resources and reduce load dispersion [7].

Sourav et al. [8] proposed a method for offloading jobs among cloudlets, noting that user satisfaction does not change as long as the delay is below a certain level. Each fraction of jobs that are offloaded is determined such that each cloudlet processes as many jobs as possible while the average delay of user jobs remains below the set acceptable latency. This is expected to maximize the use of limited cloudlet resources while satisfying user satisfaction. On the other hand, this method calculates the delay time weighted by the percentage of offloading jobs to meet the acceptable latency which results in the delay time of jobs with low percentages not affecting the results effectively. Thus, load balancing is not performed to maximize the utilization of all cloudlets.

This paper introduces the utility function that fairly evaluates the delay time for each percentage of offloading jobs by the acceptable latency, and evaluates the effectiveness of the proposed method of load balancing by the difference in utilization among cloudlets that can be derived using the optimal offloading percentage. The contribution of this paper are:

1. Optimal offloading percentage for each arriving jobs at cloudlet is proposed.

2. Building a non-cooperative load balancing game that maximizes the utilization below acceptable latency.

3. The numerical experiments show that the proposed method maximizes the utilization while meeting the acceptable latency.

ORCID iDs Yuki Yokota: 0009-0003-2691-2403, Sumiko Miyata: 0000-0001-8023-7435

2. Related Research

As the conventional methods for distributing the load of cloudlets [3], the access points where cloudlets are installed are determined according to the density of users in the area, and the allocation decisions are made so that the average delay time for each user is the smallest. This affects the cost and number of cloudlets installed due to the trade-offs. In the research [5], [6], it only determines the allocation of job requests from mobile terminals to cloudlets based on average latency. Many of the studies mentioned above, such as the cloudlet placement decisions [3], and job assignment [6], have only analyzed in limited environments because it is difficult to account for the randomness of job arrival rates due to the complex mobility pattern of real users.

Sourav et al. [8] is a research that focuses on offloading jobs between cloudlets, and it uses queuing theory to model the randomness of the users theoretically. Sourav et al. also sets a bound on the average latency for processing each job in terms of acceptable latency instead of minimizing it, and uses non-cooperative game theory to determine the offload fraction so that the utilization at each cloudlet is uniformly maximized. However, the delay time is weighted according to the offload fraction of each job, and the delay time of jobs with small fractions is not reflected in the results. This raises a problem of fairness among offloading fractions. In this paper, we derive an optimal load balancing method using an utility function that allows for more rigorous offloading, based on the method in [8].

To find the optimal solution of the problem the Mixed-Integer Programming is used in [3], and game theory based approach is used in [5], and [8]. In the assumed environment, there are multiple providers that maintain each cloudlets. Thus, cloudlets will not cooperate with each other to achieve global optimal solution but will act selfishly to maximize their own individual utility. Therefore, the use of non-cooperative game theory to find the optimal solution is suitable for this topic.

3. Proposed method

3.1. System model

Jobs arriving at cloudlet $i \in \{1, 2, ..., N\} = C, N \geq 2$ are first determined to be offloaded before being processed. Jobs that are not offloaded are processed as they arrive in cloudlet i and the results are returned to the user. If offloaded, the job is sent to another cloudlet, where it is processed and returned to the user via the original arriving cloudlet i. All cloudlets are assumed to have known information about the processing power of each other and are aware of each other's existence as competitors.

Based on the cluster data provided by Google, we can assume that the arrival rate of jobs sent by users to servers follows a Poisson distribution and the service rate of servers follows an exponential distribution, so their behavior is a Poisson process. Therefore, we model the job arrival system using the $M/M/1$ model for each cloudlet. Figure 1 shows the system model of the proposed method. Information on the arrival rate to each cloudlet is collected to the manager, and the offloading method is determined based on this information.

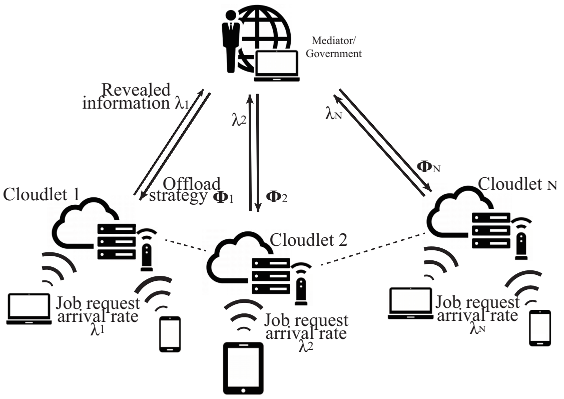

Figure 1: System model.

Let $\lambda_i,\ i \in C$ [jobs/s] be the average arrival rate of jobs in cloudlet i following a Poisson distribution, and let μ [jobs/s] be the average service rate following an exponential distribution. To determine the offload destination for jobs arriving at cloudlet i, we define the percentage vector $\boldsymbol{\varphi}_i = (\varphi_{i1}, \varphi_{i2}, ..., \varphi_{iN}) \subset \mathbb{R}^N,\ \varphi_{ij} = [0,\ 1] \subset \mathbb{R}, \sum_{j=1}^{N} \varphi_{ij} = 1, i, j \in C$ is defined. In this case, φ_{ij} indicates the fraction of jobs that the i-th cloudlet offloads to the j-th cloudlet. Thus, jobs in $\varphi_{ii}\lambda_i$ are processed in cloudlet i without being offloaded, and jobs in $\varphi_{ij}\lambda_i$ are offloaded to cloudlet j. From this, the actual arrival rate $\lambda_i^* = \varphi_{ii}\lambda_i + \sum_{j \neq i} \varphi_{ji}\lambda_j$ at cloudlet i after all the jobs have been offloaded can be defined.

The average round-trip delay between users and cloudlets is defined as t_{prop}, and the average round-trip delay between each cloudlets is defined as $t_{ij},\ i, j \neq i \in C$. Using parameters $\lambda_i,\ \mu,\ \boldsymbol{\varphi}_i$ the average processing delay at each cloudlet can be calculated using Little's formula, which derives the average delay for each fraction of jobs $T_{ij},\ i, j \in C$.

The average delay T_{ii} for jobs $\varphi_{ii}\lambda_i$ that are not offloaded is defined as

$$T_{ii} = t_{ui} + \frac{1}{\mu_{ii} - \varphi_{ii}\lambda_i - \sum_{j \neq i} \varphi_{ji}\lambda_j}. \tag{1}$$

The average delay of offloaded jobs $\varphi_{ij}, j \neq i$ can be expressed as follows.

$$T_{ij} = t_{ui} + t_{ij} + \frac{1}{\mu_{ii} - \varphi_{ij}\lambda_i - \sum_{k \neq i} \varphi_{kj}\lambda_k}. \tag{2}$$

3.2. Non-Cooperative game theory

In this paper, the offloading fraction $\boldsymbol{\varphi}_i$ of cloudlet i and the offloading fraction of other cloudlets $\boldsymbol{\varphi}_{-i}$ =

$(\boldsymbol{\varphi}_1, ..., \boldsymbol{\varphi}_{i-1}, \boldsymbol{\varphi}_{i+1}, ... \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})$, $i \in C$ are the strategy of each cloudlet that participate in the load balancing game. From the strategy set, utility function $U_i(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})$, $i \in C$ is defined and the each cloudlet maximizes their utility non-cooperatively by choosing their best strategy. The optimal offloading fraction $\boldsymbol{\varphi}^*{}_i$ that maximizes the utility is determined using game theory.

The utility of cloudlet i is considered to depend on the initial arrival rate λ_i, multiplied by the price value Ω_1. At the same time, for every $\sum_{j=1}^{N} \varphi_{ji}\lambda_j$ of jobs offloaded from other cloudlets and for each offloaded job $\sum_{j=1}^{N} \varphi_{ij}\lambda_i$ cloudlet i receives a reward of Ω_2 and give away the same amount for each offloading job $\sum_{j=1}^{N} \varphi_{ij}\lambda_i$ as a penalty. We also define ζ and η as the cost weights required to process jobs in cloudlets, respectively, to reproduce reasonable offloading of cloudlets.

If T_{ij} does not satisfy the acceptable latency D_{ij}, $i, j \in C$, penalty of Ω_3 is given accordingly, to make the computed offloading strategy satisfy the latency.

Using the above and by considering that the diagonal component can be replaced by $\varphi_{ii} = (1 - \sum_{j=1, j\neq i}^{N} \varphi_{ij})$ from the definition of the offload fraction φ_{ij}, the utility function for cloudlet i is given as follows.

$$\begin{aligned} U_i\,(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-i}) = {} & \Omega_1 \frac{\lambda_i}{\mu} + \Omega_2 \sum_{j=1, j\neq i}^{N} \varphi_{ji} \frac{\lambda_j}{\mu} \\ & - \Omega_2 \sum_{j=1, j\neq i}^{N} \varphi_{ij} \frac{\lambda_i}{\mu} - \left\{ \zeta \left[\frac{(1 - \sum_{j=1, j\neq i}^{N} \varphi_{ij})\lambda_i + \sum_{j\neq i} \varphi_{ji}\lambda_j}{\mu} \right] + \eta \right\} \\ & - \frac{\Omega_3}{N} \frac{\lambda_i}{\mu} \left\{ (t_{\text{prop}} + \frac{1}{\mu - (1 - \sum_{j=1, j\neq i}^{N} \varphi_{ij})\lambda_i - \sum_{j\neq i} \varphi_{ji}\lambda_j} - D_{ii}) \right. \\ & \left. + (\sum_{j=1, j\neq i}^{N} \left[t_{\text{prop}} + t_{ij} + \frac{1}{\mu - \varphi_{ij}\lambda_i - \sum_{k\neq i} \varphi_{kj}\lambda_k} - D_{ij}) \right] \right\}. \end{aligned} \tag{3}$$

Each cloudlet determines its offloading fraction so that the utility function $U_i\,(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})$ is larger and offloading is performed to distribute the load. Therefore, it is irrational for the utility obtained by offloading to be less than the utility obtained by not offloading at all. Therefore, if the utility with no offloading at all is U_i^0, $i \in C$, the following inequality condition can be defined.

$$\begin{aligned} U_i\,(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-i}) \geq U_i^0 = {} & \Omega_1 \frac{\lambda_i}{\mu} - \{\zeta[\frac{\lambda_i}{\mu}] + \eta\} \\ & - \Omega_3 \frac{\lambda_i}{\mu} \{t_{\text{prop}} + \frac{1}{\mu - \lambda_i} - D_i i\}. \end{aligned} \tag{4}$$

Moreover, since the expected systems are steady-state, the solution could only be derived while the utility function satisfies $[\mu - (1 - \sum_{j\neq i} \varphi_{ij})\lambda_i - \sum_{j\neq i} \varphi_{ji}\lambda_j] \geq 0$, $i,\ j \neq i \in C$.

Using game theory, the game treated in this paper can be modeled as $\Gamma = (C,\ \{\boldsymbol{\theta}_i\}_{i\in C},\ \{U_i(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-\mathbf{i}})\}_{i\in C})$. The set of cloudlet strategies is modeled by $\Theta = (\boldsymbol{\theta}_i = \{\boldsymbol{\varphi}_i : U_i(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-i}) \geq U_i^0\},\ i \in C)$, where cloudlet i chooses a strategy from the set such that it gains more than U_i^0. Therefore, the game Γ is treated as a Generalized Nash Equilibrium (GNE) problem because each player's strategy affects each other's set of strategies. GNE problems are computationally very complex and finding a solution is difficult. However, it can be proved that under certain conditions the problem can be replaced by a Variational Inequality (VI) , and if only one solution can be found for the VI problem the solution is known to match with the Nash equilibrium of the replaced GNE [8].

Algorithm 1 Projection Algorithm with Constant Step Size

```
1: Initialization: Choose any Lagrange multipliers $\boldsymbol{\alpha}^0$, $\boldsymbol{\beta}^0$, $\boldsymbol{\xi}^0 \geq 0$, step size $\omega > 0$, and tolerance limit $\epsilon > 0$. Set the index $t = 0$.
2: Output: NE of the computation offload game $\boldsymbol{\varphi}^*$.
3: If $\boldsymbol{\alpha}^t$, $\boldsymbol{\beta}^t$, and $\boldsymbol{\xi}^t$ satisfies a desirable tolerance limit: STOP
4: Given $\boldsymbol{\alpha}^t$, $\boldsymbol{\beta}^t$, and $\boldsymbol{\xi}^t$, compute $\boldsymbol{\varphi}^t(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t, \boldsymbol{\xi}^t)$ as the NE of the GNE problem (5)-(8) with fixed Lagrange multipliers $\boldsymbol{\alpha} = \boldsymbol{\alpha}^t, \boldsymbol{\beta} = \boldsymbol{\beta}^t$, and $\boldsymbol{\xi} = \boldsymbol{\xi}^t$.
5: Update all the Lagrange multipliers: for all $i,\ j \in C$, compute
       $\alpha_{ij}^{t+1} = [\alpha_{ij}^t - \omega(\varphi_{ij})]^+,\ i \neq j$
       $\beta_{ij}^{t+1} = [\beta_{ij}^t - \omega(1 - (\varphi_{ij}))]^+,\ i \neq j$
       $\xi_{ij}^{t+1} = [\xi_{ij}^t - \omega(U_i - U_i^0)]^+,$
   where $[z]^+ = \max\{0,\ z\}$.
6: Set $t \leftarrow t + 1$; go to Step 3.
```

3.3. Solving for optimal solution

To solve for the optimal solution $\boldsymbol{\varphi}^*{}_i$, using the Lagrange multipliers $\boldsymbol{\alpha}_i \in \mathbb{R}^{N-1}$, $\boldsymbol{\beta}_i \in \mathbb{R}^{N-1}$, $\boldsymbol{\xi}_i \in \mathbb{R}^N$, $i \in C$ the KKT conditions for cloudlet i can be defined as follows.

$$\nabla_{\boldsymbol{\varphi}_i} U + \nabla_{\boldsymbol{\varphi}_i} \left(\boldsymbol{\alpha}_i^{\mathrm{T}} \boldsymbol{\varphi}_i + \boldsymbol{\beta}_i^{\mathrm{T}} (1 - \boldsymbol{\varphi}_i) + \boldsymbol{\xi}_i^{\mathrm{T}} (U_i - U_i^0)_{i\in C} \right), \tag{5}$$

$$\boldsymbol{\alpha}_i^{\mathrm{T}} \boldsymbol{\varphi}_i = 0, \tag{6}$$

$$\boldsymbol{\beta}_i^{\mathrm{T}} (1 - \boldsymbol{\varphi}_i) = 0, \tag{7}$$

$$\boldsymbol{\xi}_i^{\mathrm{T}} (U_i - U_i^0) = 0. \tag{8}$$

Using the conditions above the optimal solution $\boldsymbol{\varphi}^*{}_i$ can be derived using gradient projection algorithm described in Algorithm 1 [8].

4. Numerical analysis

4.1. Parameters

To show the effectiveness of the proposed method following numerical analysis will take place. In this analysis, each weight parameter is based on conventional methods with $\Omega_1 = 5 \times 10^2$, $\Omega_2 = 1 \times 10^6$, $\Omega_3 = 5 \times 10^8$, $\zeta = 300$, $\eta = 700$ [8]. The processing rate of the cloudlet was set to $\mu = 10000$ jobs/s, and $t_{\text{prop}} = 2$ ms, $t_{12} = 0.75$ ms, and $D_{ij} = 5$ ms were defined.

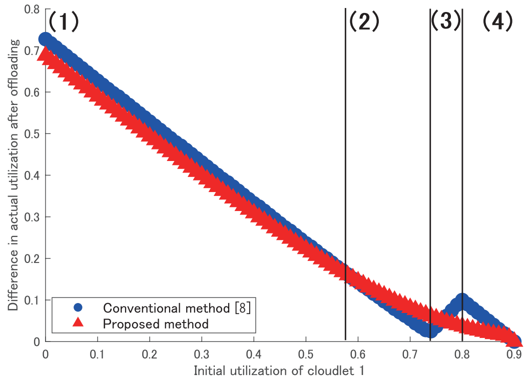

Figure 2: Difference in utilization after offloading for different arrival rate of cloudlet 1

Although the general scenario assumes multiple cloudlet environment, the numerical analysis under offloading between two cloudlets with $N = 2$ is enough to evaluate the effectiveness of our method. This is because multiple cloudlet environment is a linear extension of the environment with two cloudlets. By setting the initial arrival rate of cloudlet 2 to $\lambda_2 = 9000$ jobs/s the utilization before offloading is fixed at 0.9. The initial arrival rate λ_1 is set between 0 to 9000 jobs/s which varies the utilization of cloudlet 1 before offloading from 0 to 0.9. This allows to consider the randomness of the arriving mobile users and evaluates the performance of load balancing in various system environments. The optimal offloading fraction for each case is derived using Algorithm 1, and the results are evaluated in terms of difference in utilization after offloading, which is calculated from the actual arrival rate λ_1^*, λ_2^*.

4.2. Results

Figure 2 shows the comparison of the conventional method and the proposed method in terms of utilization difference after offloading between two cloudlets.

In the area (1) shown on figure 2 the proposed method performs better in terms of load balancing. This is because the proposed method evaluates the latency of each fraction of jobs individually, which balances the load for both cloudlets. In the area (2), the conventional method performs slightly better due to the difference in the amount of acceptable latency applied to the utility function. However, in the area (3) the results of the conventional method starts to increase. The exponential increase in the latency of offloaded jobs is directly reflected on the utility function of conventional method which decreases the offloading fraction until it reaches to a point where no offloading will occur. Consequently, in the area (4) no offloading will take place for the conventional method and the proposed method will start to outperform because it continues to offload even in highly crowded environment. Overall, the latency of all the jobs in the system are kept below acceptable latency.

5. Conclusion

This paper has contributed on the optimal load balancing of cloudlets below acceptable latency by modelling the offloading decision making system using queuing theory and game theory. Compared to the conventional research [8] we focused on evaluating the latency of each fraction of jobs individually to improve the load balancing between cloudlets. The result show that the proposed method has managed to determine the offloading strategy in various system environment and outperforms the conventional method.

Acknowledgments

These research results were obtained from the commissioned research(No.05601) by National Institute of Information and Communications Technology (NICT), Japan.

References

[1] E. Wong, M. Pubudini Imali Dias, and L. Ruan, "Predictive resource allocation for tactile internet capable passive optical lans," *Journal of Lightwave Technology*, vol.35, no.13, pp.2629–2641, 2017.

[2] 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.

[3] Q. Fan and N. Ansari, "Cost aware cloudlet placement for big data processing at the edge," *IEEE International Conference on Communications* pp.1–6, 2017.

[4] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol.98, pp.289–330, 2019.

[5] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Transactions on Vehicular Technology*, vol.67, no.1, pp.752–764, 2018.

[6] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based iot," *IEEE Internet of Things Journal*, vol.5, no.3, pp.2146–2153, 2018.

[7] Y. Jiang, "A survey of task allocation and load balancing in distributed systems," *Transactions on Parallel and Distributed Systems*, vol.27, no.2, pp.585–599, 2016.

[8] 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.