A Distributed ADMM Approach for Data Offloading in Fog Computing

Tri Nguyen Dang, Choong Seon Hong
Department of Computer Science and Engineering, Kyung Hee University, 446-701, Korea
Email: {trind, cshong}@khu.ac.kr

Abstract

Cloud Computing is a technological revolution that provides Storage, memory, computational and many applications for the end user. Internet of Things(IoT) has been introduced to raise the number of users connected to the network, which means increased data traffic and envisioned the performance of Cloud server(CP) significantly. The Cloud cannot satisfy the huge demand from network's user. Data offloading is one of the best solutions can be used to avoid this changeling. In this paper, we proposed a distributed Alternating Multiplier Methods(ADMM) to optimizing the performance of offloading algorithm also guarantee the Quality of Service(QoS) for end-user. We have numerically analyzed our system using Julia, the results show that the proposed mechanism has performance and fast convergence to the optimal solution.

1. INTRODUCTION

Cloud computing has been introduced in [1], [2]. It is a technological revolution that provided storage, memory, computational for end-user(CUE). CUE can connect to CP and use services, run applications, storage data. It was raised the data traffic in the network significantly then makes the congestion in anytime. The frequent congestion occurs will be taken down the network quality and increasing the delay [4]. A number of efforts have been performed in the literature for solving this problem [5], [6]. The approach that using distributed server has been proposed in [5] named as Fog Computing. The Fog provide storage, memory, computational and network but cannot replace the CP [6]. The Fog extends the Cloud to be closer to the CUE. The Fog's goal is to make better efficiency and reduce the amount of data that needs to be transported to the cloud for data analysis, processing and storage [7]. In the case of CUE, the frequently using CP service is upload data then data can be temporary saving in somewhere before uploading to the CP that can reduce a big amount of traffic in the Core.

2. NETWORK MODEL

In this section, we introduce the system model that data from Cloud user can be offloaded via Fog nodes. In the Fig. 1, There is a number of Fog nodes available amount of storage and a number of users want to transfer data to Cloud Server. In the other hand, Cloud Server(CP) is far away from CUEs, data should be transferred via the core. It means the data traffic will be increasing.

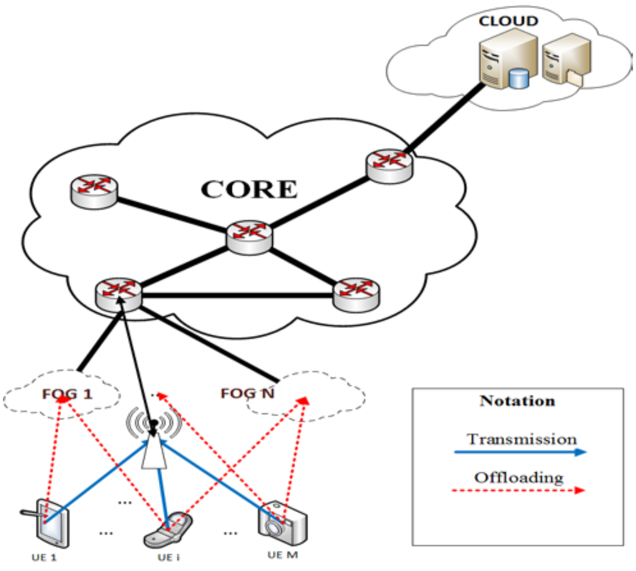

Figure 1: System model

We assuming that CUE only uploads some media file to storage on the CP then that data can be storage temporary at some nearby Fog nodes at the daytime and transfer to CP at the nighttime. It can help the core reduce the traffic and congestion. In the network, the traffic is not always dense, almost traffic distributed around daytime and free at nighttime. Let D is the total amount of data uploading of CUEs. On the other side, we consider there is exists a set of $\mathcal{N}$ virtual Fog nodes $\mathcal{N} = \{1, 2 \ldots, N\}$ which are available for storage amount of data. For each $n \in \mathcal{N}$, it has a limited capacity Γ_n (GB). The CP can make a decision with the Fog to allocated the data upload from CUE. When a request comes from CUE, based on the information about the Fog nodes, the size that CUE upload, and the network condition between the CUEs and the Fogs, the CP will calculate where to allocated the data such that minimize the transmission time, the cost and the association times between CUE and the Fog. Let x_n is the amount of data that will be storage in the Fog node n, then the cost function is $f_n(x_n)$, assuming that the transmission time depends on the size of data upload and the bandwidth allocated for CUE denoted as $\Gamma_n = h_n(x_n)$.

Therefore, the problem can be formulated at

$$\begin{aligned}&\underset{x}{\text{minimize}}: \sum_{n=1}^{N} f_n(x_n)\\ &\text{subject to}:\\ &\qquad \sum_{n=1}^{N} x_n = D \qquad (1)\\ &\qquad x_n \le \Gamma_n, \forall n \in N \quad (2)\\ &\qquad x_n \ge 0, \forall n \in N \quad (3)\end{aligned}$$

Where the objective function is minimizing the cost that CP will use the storage of the Fogs. The first constraint makes sure that all of the requesting from CUE will be offloaded. The second constraint represents that the transmission time to during offloading is not greater than the uploading time. Denote $f(x) = \sum_{n=1}^{N} f_n(x_n)$ and a set $\mathbb{C} = \{x : x \ge 0, 1^T x = D\}$, where $1 \in \mathbb{R}^N$ is a vector of all 1 values. Then the problem can be written in ADMM as:

$$\begin{aligned}&\underset{x}{\text{minimize}}: f(x) + g(z)\\ &\text{subject to}:\\ &\qquad x - z = 0 \quad (1)\\ &\qquad x - \Gamma \le 0 \quad (2)\end{aligned}$$

Where $g(z) = \Pi_C(z) = \begin{cases} 0, & z \in C \\ \infty, & otherwise \end{cases}$ and $z = \{z_1, ..., z_N\} \in \mathbb{R}^N$.

Following the ADMM [3] methods, the Lagrangian function can be defined at

$$\begin{aligned}&\mathcal{L}_\rho(x, \lambda, \nu) = f(x) + g(z) + \nu^T (z - x)\\ &+ \lambda(x - \Gamma) + \frac{\rho}{2} \| z - x \|_2^2 \ (4)\end{aligned}$$

Then the unscaled ADMM update is as follows

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^N} \{ f(x) - x^T \nu^k + \lambda^T (x - \Gamma) + \frac{\rho}{2} \| x - z^k \|_2^2 \} (5)$$

$$z^{k+1} = \arg\min_{z \in \mathbb{R}^N} \{ g(z) - z^T \nu^k + \lambda^T (x^{k+1} - \Gamma) + \frac{\rho}{2} \| z - x^{k+1} \|_2^2 \} (6)$$

$$\nu^{k+1} = \nu^k + \rho(x^{k+1} - z^{k+1}) (7)$$

$$\lambda^{k+1} = \lambda^k + \rho(x^{k+1} - z^{k+1}) (8)$$

3. EVALUATION

In this section, we present simulation results from our proposed Algorithm 1. In this algorithm, we first randomly generate $x^0, z^0, \nu^0, \lambda^0, \rho = 0.5$. We then loop at most T iteration. For each iteration, we update the objective function and all of the variable. The algorithm will be stopped if the time is greater than predefined time or the objective value has converged.

Algorithm 1: Minimize offloading cost(MOC)

1. Initial information for any $k \in \mathcal{K}$ and $c \in C$
2. Generate value for all variable $x^0, z^0, v^0, \lambda^0, \rho = 0.5$
3. For each time slot $t \in \{1, 2, \ldots, T\}$
   a. Calculate (5) - (8)
4. Update the objective value.
5. **If** objective satisfied, **then** algorithm stopped
6. Break and return results.

In this simulation, we have chosen N = 100, D = 1000(GB), random $f_n(\bullet)$ as well as $h_n(\bullet)$.In the simulation, we analyze the algorithm by Julia and had shown the numerical results. We have used two different algorithms for performance comparison:

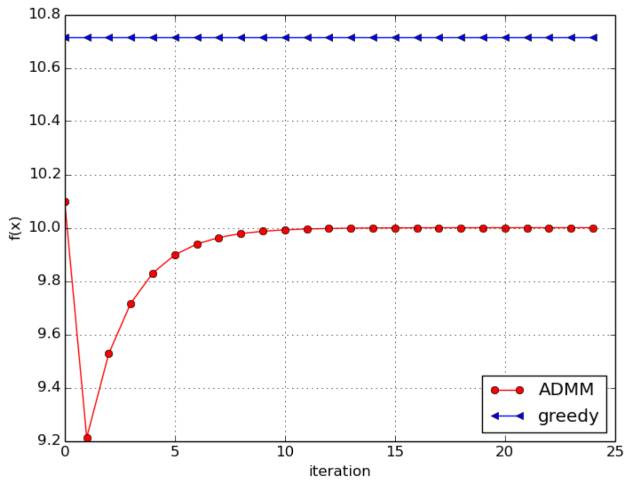

Figure 2: Simulation result

The first one is a Greedy algorithm and the second one is our proposed dynamic offloading.

The simulation result showed that the proposed algorithm has achieved solution after few iteration and the solution is better than greedy algorithm. The system information when we ran this simulation, CPU is *Intel Core i5-4150* 3.5(*GHz)*, and the capacity of RAM is 12.0*(GB).*

4. CONCLUSION

Data offloading can help CP improve its performance, saving energy, and reduce the data traffic in the network. This study shows us that by applying the ADMM method to determine whether the data will be allocated or not depending upon the network condition can significantly enhance the performance. Our numerical result has shown that proposed scheme significantly enhances the performance and achieves very fast convergence. Finally, a detailed study of the actual operating environment is required, as a future course, to validate our proposal and its performance.

5. ACKNOWLEDGEMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0190-16-2017, Resilient/Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices) *Dr. CS Hong is the corresponding author

REFERENCES

[1] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011)

[2] Armbrust, Michael, et al. "A view of cloud computing." *Communications of the ACM* 53.4 (2010): 50-58.

[3] Wahlberg, B., Boyd, S., Annergren, M., & Wang, Y. (2012). An ADMM algorithm for a class of total variation regularized estimation problems. *IFAC Proceedings Volumes*, *45*(16), 83-88.

[4] Jacobson, Van. "Congestion avoidance and control." *ACM SIGCOMM*. Vol. 18. No. 4. ACM, 1988.

[5] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. Fog computing and its role in the internet of things. In *Proceedings of MCC* (pp. 13-16). ACM 2012.

[6] Cisco: Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are [white paper]

[7] http://fognetworks.org/ [October 26, 2016]