A Proximal Algorithm for Joint Resource Allocation and Minimizing Carbon Footprint in Geo-distributed Fog Computing

Cuong T. Do, Nguyen H. Tran, Chuan Pham, Md. Golam Rabiul Alam, Jae Hyeok Son and Choong Seon Hong

Department of Computer Enginneering, Kyung Hee University, 446-701, Korea
Email: {dtcuong, nguyenth, pchuan, robi, sonjaehyeok, cshong}@khu.ac.kr

01
Qizhao Wang

***Abstract*—Large-scale Internet applications, such as content distribution networks, are deployed in a geographically distributed manner and emit massive amounts of carbon footprint at the data center. To provide uniform low access latencies, Cisco has introduced Fog computing as a new paradigm which can transform the network edge into a distributed computing infrastructure for applications. Fog nodes are geographically distributed and the deployment size at each location reflects the regional demand for the application. Thus, we need to control the fraction of user traffic to data center to maximize the social welfare. In this paper, we consider the emerging problem of joint resource allocation and minimizing carbon footprint problem for video streaming service in Fog computing. To solve the large-scale optimization, we develop a distributed algorithm based on the proximal algorithm and alternating direction method of multipliers (ADMM). The numerical results show that our algorithm converges to near optimum within fifteen iterations, and is insensitive to step sizes.**

I. Introduction

Recently, Cisco has introduced Fog computing as a new paradigm which can transform the network edge into a distributed computing infrastructure for applications that take advantage of the billions of devices already connected to the Internet of Things (IoT) [1], [2]. In this year, Cisco continues on delivering its vision for the fog computing, being a systematic, highly virtual, secure, and network-integrated platform that provides computing, storage, and networking services between end points and traditional Cloud computing data centers [3]. The Fog is located below the Cloud in a widely distributed manner and serves as an optimized transfer medium for services and data within the Cloud. Fig. 1 presents this idealized information and computing architecture and illustrates an implementation of Fog Computing. It can be called “Fog” simply because fog is a cloud close to the ground. Since Fog has wide geographical distribution, the Fog paradigm is well positioned for big data and real time analysis and it supports mobile computing and data streaming. In Fog Computing model, data, processing and applications are concentrated in devices at the network edge, rather than existing almost entirely in the Cloud, to isolate them from the Cloud systems and place them closer to the end-user. Putting computing resource near the edge allows Fog to perform low latency processing while latency tolerant and large scope

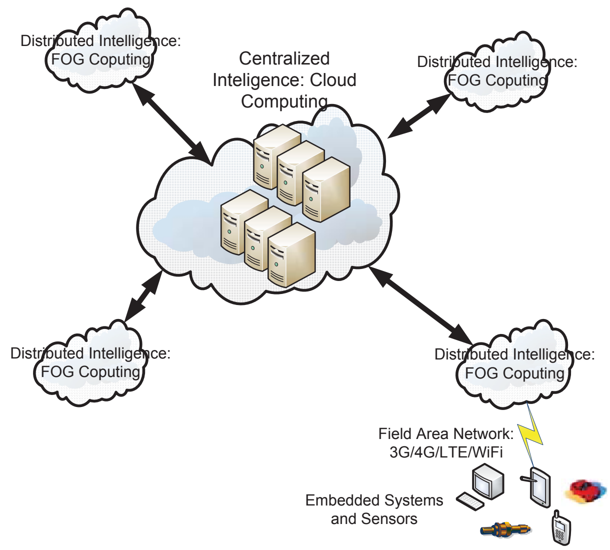

Fig. 1. Fog Computing architecture.

aggregation can still be efficiently performed on powerful resources in the core of the Cloud. Data center resources may still be used with Fog computing, but they do not dominate over the entire picture.

Fog computing is one potential approach to dealing with the demands of the ever-increasing number of Internet-connected devices sometimes referred to as the Internet of Things. Many advantages of Fog computing are listed in [1] as follows. Fog can support the applications with low latency requirements (e.g. gaming, video streaming, augmented reality). Due to low latency at the edge, Fog applications involve real-time interactions rather than batch processing. In addition, Fog can support mobility and location awareness, heterogeneity, interoperability and federation, integration with the Cloud and support for on-line analysis, predominance of wireless access and dynamic per user optimization. Potential Internet of Things (IoT) services and applications of Fog Computing are presented in [1] such as Connected Vehicle, Smart Grid, and

Wireless Sensor and Actuator Networks. The growing ubiquity of IoT devices is enabling a wide range of novel, large scale, latency sensitive applications that motivates Fog computing. Fog Computing will enable a totally new breed of applications and services that will come to life as hundreds of billion of things [4].

Fog computing is an addition which develops the concept of Cloud services and does not replace Cloud computing. In [2], the authors strive to clarify key characteristics of Cloud and Fog computing. While Fog nodes provide localization, therefore enabling low latency and context awareness, the Cloud provides global centralization. Many applications require both Fog localization, and Cloud globalization, particularly for analysis and big data [1]. In [5], Madsen and Albeanu show the reliability challenges posed by Fog computing platforms incorporating networks of smart devices communicating among them and also with Cloud. The authors in [6] present a high level programming model for future Internet applications that are geographically distributed, large scale, and latency sensitive. The proposed model is a PaaS programming model, called Mobile Fog, that provides a simplified programming abstraction and supports applications dynamically scaling at runtime. They also analyze use cases for the programming model with the camera network and connected vehicle applications to show the efficiency of Mobile Fog. In [7], the authors exploit the knowledge that is only available at the edge (Fog) nodes to improve user's webpage rendering performance.

In industry, many companies are ready for adopting Fog computing [8]. Any company that delivers content can start using Fog computing [2]. A good example is Netflix who is able to reach a large number of globally distributed customers. The delivery of video-on-demand service would not be efficient enough if it is based on the data management in one or two central data centers. Fog computing thus allows providing very large amounts of streamed data by delivering the data directly into the vicinity of the customer [9].

Almost all of these streaming servers are built on top of geographically distributed infrastructure to provide uniform reliability and performance. Consequently, an application's environmental impact can vary significantly depending on the geographical distribution of end-users. They need an effective algorithm to direct demand across the wide area to data center. Many previous works exist in this area such as [10]–[18]. The problem can be cast as an large-scale optimization that maximizes the total utility or minimize the total cost. In such a large-scale system, fast distributed algorithms for resource allocation are becoming increasingly important. A new algorithms for large-scale distributed resource allocation with coupled objectives is presented in [16]. Inspiring by Alternating Direction Method of Multipliers (ADMM) [19], the authors have introduced an unified framework for studying various cloud traffic management problems, ranging from geographical resource allocation to backbone traffic engineering.

Today, geographical resource allocation and energy cost are managed independently, leading to poor performance and high costs in many cases [12]. The objectives of the two decisions can also be misaligned and lead to sub-optimal equilibria. In this paper, we study the joint resource allocation and minimizing carbon footprint problem for streaming service in Fog computing. We assume that physical devices called Fog computing nodes (FCNs) are placed in the network infrastructure to deliver video streaming service from content providers. For example, streaming services are hosted at the network edge such as "smart" routers and switches with more application-level functionality, or even end devices such as set-top-boxes or access points. By doing so, Fog reduces service latency, and improves QoS. FCN aggregates video demands from nearby end users. As the tenants of a Fog provider, the content providers want to delivery content from the data center as much as possible to FCN to increase their utility. Because the applications and end users are heterogeneous, the utility varies significantly depending on the geographical distribution of end-users. Thus, we need to control the fraction of traffic (from huge number of end-users to data center) to maximize the utility of content providers and minimize the carbon footprint at the data center.

In a large scale systems, fast distributed resource allocation and social welfare maximization are critical problems. Traditional solutions to such problems rely on primal/dual decomposition and gradient methods [20]–[23], however these methods have slow convergence speed and sensitivity to step sizes and require strict convex assumptions. Related work has considered geo-distributed and large scale system such as work in [16] for video streaming in cloud computing. In [16], the authors propose a distributed algorithm with fast convergence speed but it requires strict convex assumptions like primal/dual decomposition and gradient methods. Here, the joint resource allocation and minimizing carbon footprint problem is a very large-scale convex optimization due to large number of FCN (dozen of thousand devices in [3]). Thus, for reasons of performance and scalability, we introduce a distributed solution for the joint resource allocation and minimizing carbon footprint problem. Our algorithm is based on proximal algorithms [24], a powerful algorithm that recently has been applied in many large scale distributed convex optimization problems. Comparing to conventional methods such as gradient methods, proximal algorithms have faster convergence speed with modest accuracy, insensitivity to step sizes, and robustness without strong assumptions such as strict convexity of the objective function [19], [24].

The remainder of this paper is organized as follows. Section II introduces the system model. In Section III, we present joint resource allocation and minimizing carbon footprint problem. Section IV shows the numerical results. We draw the conclusions in Section V.

II. System model

In this section, we introduce our model first and then formulate joint resource allocation and minimizing carbon footprint problem.

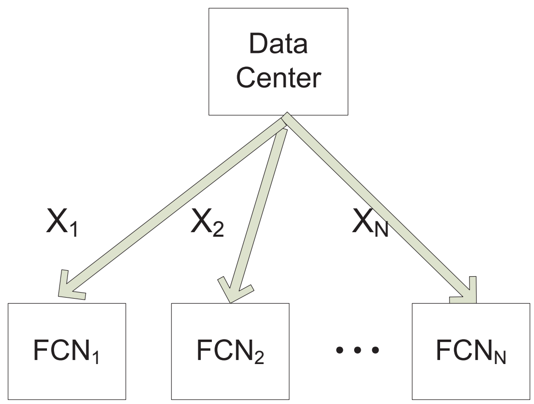

Fig. 2. Multiple Fog Computing Nodes with a Data Center.

A. System Model

We consider a content provider run Fog computing services over a data center and N Fog Computing Nodes (FCNs) are located at the edge of network in distinct geographical regions to serve video streaming for end users as illustrated in Fig. 2. We assume that FCNs send a request to the data center. After the data center have finished serving request, it sends the response video streaming back to FCNs. We use x_n to denote the amount of video streaming to FCN n from the data center. We assume that the reservation of egress network bandwidth from the data center to FCNs has already been enabled by the detailed engineering techniques proposed in [10], [25]. Thus, the data center can guarantee the bandwidth to serve video streaming of clients at FCNs. To model the social welfare, we consider both the carbon footprint cost at the data center and utility of the content provider, which are detailed below.

B. Utility and Cost Function

Utility, or usefulness, is the (perceived) ability of something to satisfy needs or wants. Utility is an important concept in economics and optimization theory, because it represents satisfaction experienced by the consumer of a goods (e.g, amount of video streaming). Some studies have applied complicated utility functions with fairness considerations [17] or a utility loss for the demand from end users that are not guaranteed [16]. However, we consider an affine utility function that is the de facto utility function widely used in the literature [18]. An affine utility function at FCN n has the following form:

$$U_n(x_n) = \alpha_n x_n, \tag{1}$$

where x_n is the amount of response video streaming at FCN n and $\alpha_n > 0$ is a conversion factor that translates user-perceived request video streaming into utility (e.g., revenue). Because the applications and end users are heterogeneous, α_n varies significantly depending on the geographical distribution of end-users.

For the environmental cost, the carbon footprint of energy at the data center can also be taken into account. The cost

TABLE I
CARBON DIOXIDE EMISSION PER KILOWATT-HOUR FOR THE MOST COMMON FUEL TYPES

	Nuclear	Coal	Gas	Oil	Hydro	Wind
CO_2 gkWh	15	968	440	890	13.5	22.5

function $C(\cdot)$ considered in many existing works [11], [12], [14] is given as follows

$$C(y) = c \cdot r \cdot PUE \cdot P(y), \tag{2}$$

where c denotes the carbon footprint cost in term of \$/g at the data center, r is the average carbon emission rate g/KWh, PUE is the power usage effectiveness and $P(y)$ represents the server power at the data center. The authors in [11] present a approximate estimation of the average carbon emission rate r of a data center. By summing the weighted contribution from each fuel type, we can calculate the average carbon emission rate of the data center as follows

$$r = \frac{\sum_j e_j \cdot r_j}{\sum_j e_j}, \tag{3}$$

where e_j is the electricity generated from fuel type j and r_j is the carbon emission rate of fuel type j which is used in the data center. The carbon emission rate of the most common fuel types was reported in [11], [26] and shown in Table I. The power usage effectiveness PUE is the ratio between total infrastructure power and server power. Since total infrastructure power mainly consists of server power and cooling power, PUE is commonly used as a measure of data center energy efficiency. We assume that the requested video streaming corresponds to the workload in the data center. $P(y)$ represents the server power at the data center, which is a function of the total of requested video streaming y and can be obtained empirically. From a measurement study by Google [27], a commonly used server power function is given by

$$P(y) = \text{Ç} \cdot P_{idle} + (P_{peak} - P_{idle}) \cdot y \cdot \kappa, \tag{4}$$

where κ is a conversion factor that translates requested video streaming into workload, Ç is workload capacity of the data center, P_{idle} is server idle power and P_{peak} peak power.

III. Joint Resource Allocation and Minimizing Carbon Footprint Problem

In this section we formulate joint resource allocation and minimizing carbon footprint problem and introduce a new distributed optimization algorithm based on proximal algorithms.

A. Problem Formulation

We now formulate joint resource allocation and minimizing carbon footprint problem. Putting the utility and cost function, the social welfare maximization problem can be written as

$$\max_{x_n \geq 0, y \geq 0} \sum_{n=1}^{N} U_n(x_n) - C(y) \tag{5}$$

$$s.t. \qquad \sum_{n=1}^{N} x_n = y \leq \text{Ç}/\kappa,$$

where the constraint is the capacity constraint at data center.

If Problem (5) is small then it would be easy to solve. However, in Fog, Problem (5) is an extremely large-scale optimization problem. For example, Akamai is the largest content delivery network in the world. It delivers $15-20$ percent of worldwide Internet traffic and it has more than $95,000$ servers in $1,900$ networks across 71 countries [28]. Then, the number of variables x_n is $\Theta(10^4)$. We need to update the resource allocation decision dynamically as demand varies on a hourly. Thus, we need a distributed algorithm to solve such a large scale problem.

Problem (5) often arises in networking systems, where each user n is associated with an utility $U_n(x_n)$ by using x_n units of some resource, while the users incur a total cost $C(y)$, which presents coupled constraint arbitrarily among users, at the resource provider. Conventional distributed manners to solve Problem (5) are dual decomposition and gradient methods [19]. We can obtain the Lagrangian dual problem of Problem (5) as follows

$$\min_u q(u) = \sum_{n=1}^{N} \max_{x_n}\{U_n(x_n) - ux_n\} + \max_y\{uy - C(y)\}, \quad (6)$$

where u is the price set by the resource provider. If Problem (5) is convex, the dual gap is zero then we only need to solve the dual problem. We have a gradient algorithm to the dual problem as follows:

$$\begin{aligned} x_n^{k+1} &:= \arg\max_{x_n} U_n(x_n) - u^k x_n, \forall n \quad (7)\\ y^{k+1} &:= \arg\max_y u^k y - C(y),\\ u^{k+1} &:= u^k - \gamma(y^{k+1} - \sum_{n=1}^{N} x_n^{k+1}), \end{aligned}$$

where γ is a sufficiently small positive step size. The conventional dual decomposition approach bases on the assumption that each U_n is strictly concave and monotonically increasing, and $C(y)$ is strictly convex. However, this assumption is not always applied to the cloud system [16], because the joint resource allocation and minimizing carbon footprint problem has non strictly convex cost function $C(y)$ in (2). Furthermore, the conventional dual decomposition approach suffers from many performance issues for solving such a large-scale problem [16], [19]. We apply proximal algorithms [24] to make new distributed optimization algorithms. We present preliminaries of proximal algorithms in the following section.

B. Preliminaries on Proximal Algorithm and ADMM

1) Proximal Algorithm: : There is a wide range of literature on applying various proximal algorithms to particular problems or problem domains, such as loss minimization in machine learning, optimal control, energy management, and signal processing [24]. The proximal minimization algorithm, also called proximal iteration or the proximal point algorithm, is

$$x^{k+1} := prox_{\lambda f}(x^k), \quad (8)$$

where $f : \Re^n \to \Re \cup \{+\infty\}$ is a closed proper convex function, k is the iteration counter, and x^k denotes the kth iteration of the algorithm. The proximal operator $prox_{\lambda f} : \Re^n \to \Re^n$ of f is defined by

$$prox_{\lambda f}(x^k) = \arg\min_x \left(f(x) + \frac{1}{2\lambda} \parallel x - x^k \parallel_2^2 \right), \quad (9)$$

where $\parallel \cdot \parallel_2^2$ is the usual Euclidean norm.

2) ADMM and Sharing problem: : ADMM was created in the 1970s and has recently received renewed interest in solving large-scale distributed convex optimization in statistics, machine learning, signal processing [19]. We first consider a problem

$$\begin{aligned} &\min f(x) + g(z) \quad (10)\\ &s.t.\ x - z = 0, \end{aligned}$$

where f, $g : \Re^n \to \Re \cup \{+\infty\}$ are closed proper convex functions. The *augmented Lagrangian* associated with the problem (10) is

$$L_\rho(x,z,y) = f(x) + g(z) + y^T(x-z) + (\rho/2) \parallel x - z \parallel_2^2, \quad (11)$$

where ρ is a penalty parameter and $y \in \Re^n$ is a dual variable associated with the consensus constraint. ADMM solves the dual problem with the iterations:

$$\begin{aligned} x^{k+1} &:= \arg\min L_\rho(x, z^k, y^k), \quad (12)\\ z^{k+1} &:= \arg\min L_\rho(x^{k+1}, z, y^k),\\ y^{k+1} &:= y^k + \rho(x^{k+1} - z^{k+1}). \end{aligned}$$

Then the ADMM is presented as the proximal algorithm as follows

$$\begin{aligned} x^{k+1} &:= prox_{\lambda f}(z^k - u^k), \quad (13)\\ z^{k+1} &:= prox_{\lambda g}(x^{k+1} + u^k),\\ u^{k+1} &:= u^k + x^{k+1} - z^{k+1}, \end{aligned}$$

with $u^k = (1/\rho)y^k$ and $\lambda = 1/\rho$.

ADMM can be used to solve the sharing problem given in the form:

$$\min \sum_{i=1}^{N} f_i(x_i) - g(\sum_{i=1}^{N} x_i) \quad (14)$$

Sharing can be written in ADMM form by copying all the variables:

$$\begin{aligned} &\min \sum_{i=1}^{N} f_i(x_i) - g(\sum_{i=1}^{N} z_i) \quad (15)\\ &s.t.\ x_i - z_i = 0, i = 1, ..., N, \end{aligned}$$

with variable $x_i \in \Re^n, i = 1, ..., N$, and $f : \Re^n \to \Re \cup \{+\infty\}$, $g : \Re^n \to \Re \cup \{+\infty\}$ are closed proper convex functions. Remark that $f(\cdot)$ and/or $g(\cdot)$ are not assumed to be strictly convex as in [20] or needs some special characteristics like those in [16].

The proximal algorithm associated with the problem (15) is given by

$$\begin{aligned} x_i^{k+1} &:= prox_{\lambda f_i}(z_i^k - u_i^k), \forall i \\ z^{k+1} &:= prox_{\lambda g}(x^{k+1} + u^k), \\ u_i^{k+1} &:= u_i^k + x^{k+1} - z_i^{k+1}, \forall i \end{aligned} \tag{16}$$

where $z = [z_1, ..., z_N] \in \Re^{Nn}$. The first and last steps can be carried out independently in parallel for each $i = 1, ..., N$ at the subsystems.

C. Our Optimal Distributed Algorithm

We rewrite Problem (5) as follows

$$\begin{aligned} \max \sum_{n=1}^{N} U_n(x_n) - C(\sum_{n=1}^{N} z_n) \\ s.t.\ \ x_n = z_n, \forall n \\ \sum_{n=1}^{N} z_n \leq Ç/\kappa. \end{aligned} \tag{17}$$

Since Problem (17) is a sharing problem, we have the iterations based on proximal algorithms as follows

$$\begin{aligned} x_n^{k+1} &:= prox_{\lambda U_n}(z_n^k - u_n^k), \forall n \\ z^{k+1} &:= prox_{\lambda C}(x^{k+1} + u^k), \\ u_n^{k+1} &:= u_n^k + x^{k+1} - z_n^{k+1}, \forall n \end{aligned} \tag{18}$$

Thus, we have a distributed algorithm (Algorithm 1) to solve the joint resource allocation and minimizing carbon footprint problem.

Algorithm 1 Optimal Distributed Solution for (17)

1: Choose randomly a initial value u^0, x^0, z^0
2: *while not convergence*:
3: Each FCN n updates request video streaming x_n^{k+1} by using x-update in (18). Send its request x_n^{k+1} and u_n^k to the data center.
4: The data center collects request video streaming x_n^{k+1}, u_n^k from all FCN and updates z^{k+1} by using z-update in (18). Then, the data center sends response video streaming x_n^{k+1} with z_n^{k+1} to FCN n for all n.
5: Each FCN n updates dual value u_n^{k+1} by using u-update in (18).
6: *end while*

IV. Numerical Results

The numerical parameters are set as follows. The servers have peak power P_{peak} = 200 W, and consume a half power at idle. The PUE is 1.5. These numbers represent state-of-the-art data center hardware. $\alpha_n > 0$, a utility-revenue conversion factor, is randomly generated by uniform distribution [0,1000] and the normal distribution 1000 x $N(0.5, 0.2)$, respectively. The initial request video streaming x_n^0 is set randomly in [50,100]. Then the capacity Ç is 1.4 x the total initial request video streaming $\sum_{n=1}^{N} x_n^0$. The upper bound of bandwidth x_n^0 is 2 x the average capacity Ç/N. The carbon emission rate at data center r is set as much as the average in the U.S. (562 g/kWh) and the cost c is equal to 19.10^{-6} \$/g.

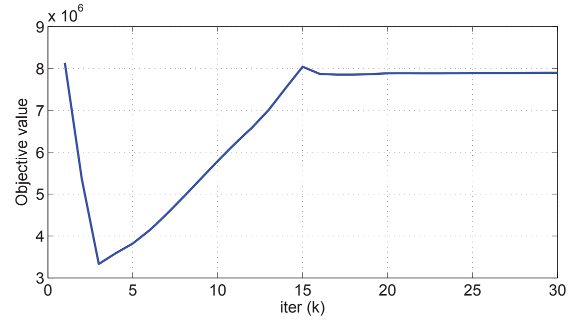

Fig. 3. Objective value (α_n is generated by uniform distribution).

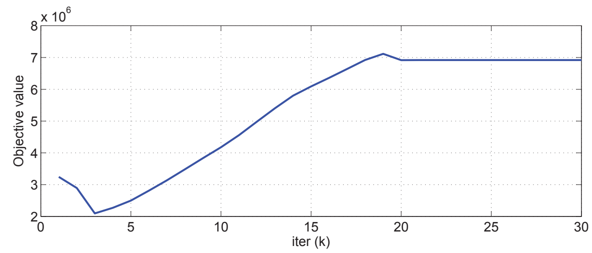

Fig. 4. Objective value (α_n is generated by normal distribution).

We evaluate the convergence of Algorithm 1 under the above setup. Fig. 3 and Fig. 4 plot the convergence of objective values. Since the number of iterations is small while the number of user N is 100, it suggests that our algorithm can solve a large-scale problem effectively.

Fig. 5 and Fig. 6 show the trajectory of the primal and dual residuals. The primary residual $\| r \|_2$ is defined as $\sum_{n=1}^{N} \| x_n - z_n \|$. The dual residual $\| s \|_2$ is defined as $\sum_{n=1}^{N} \| \rho \cdot (z_n^{k+1} - z_n^k) \|$. The residuals measure how well the iteration of Algorithm 1 satisfy the convergent conditions. The size of these residuals indicates how far the iterates are from a solution. The primary residual reflects how well the constraints $\{x_n - z_n\}$ are satisfied, and is sometimes called the primal feasibility gap. For example, if the primal residual is 10^2, which is already small enough since x_n is in the order of 10^2. Hence, we conclude that the constraints are well satisfied after 20 iterations.

V. Conclusion

We study the joint resource allocation and minimizing carbon footprint problem for streaming service in Fog computing. We formulated the problem as a general convex optimization, where the location diversity of requested video streaming utility and costs are modeled. We developed an efficient distributed algorithm based on the proximal algorithm to decompose the large scale global problem into many subproblems, each of which can be quickly solved. The numerical results are conducted to evaluate the algorithm's performance.

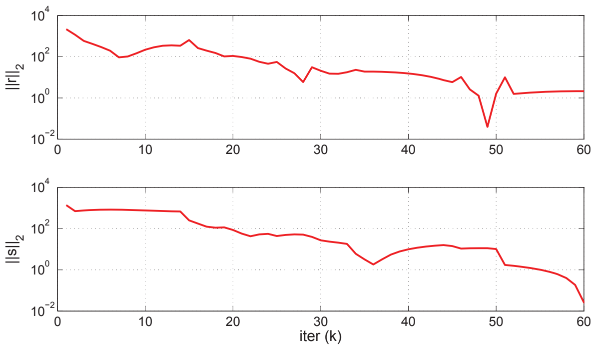

Fig. 5. Residual (α_n is generated by uniform distribution).

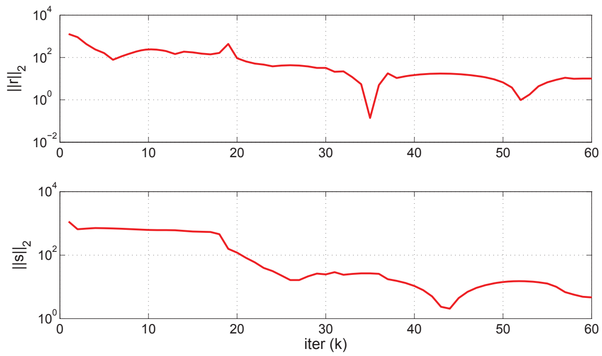

Fig. 6. Residual (α_n is generated by normal distribution).

As future work, we plan to more thoroughly study the effect of the multiple data centers with traffic engineering schemes.

VI. Acknowledgement

This work was supported by the ICT R&D program of MSIP/IITP. [14-000-05-001, Smart Networking Core Technology Development]. Dr. CS Hong is the corresponding author.

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[2] M. Hajibaba and S. Gorgin, "A review on modern distributed computing paradigms: Cloud computing, jungle computing and fog computing," *CIT. Journal of Computing and Information Technology*, vol. 22, no. 2, pp. 69–84, 2014.

[3] Cisco technology radar trends: Fog computing. [Online]. Available: http://www.cisco.com/web/solutions/trends/tech-radar/fog-computing.html

[4] Fog computing happens when big data analytics marries internet of things. [Online]. Available: http://tarrysingh.com/2014/07/fog-computing-happens-when-big-data-analytics-marries-internet-of-things/

[5] H. Madsen, B. Burtschy, G. Albeanu, and F. Popentiu-Vladicescu, "Reliability in the utility computing era: Towards reliable fog computing," in *Systems, Signals and Image Processing (IWSSIP), 2013 20th International Conference on*. IEEE, 2013, pp. 43–46.

[6] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: a programming model for large-scale applications on the internet of things," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. ACM, 2013, pp. 15–20.

[7] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*. IEEE, 2013, pp. 320–323.

[8] Fog computing is a new concept of data distribution. [Online]. Available: http://www.cloudtweaks.com/2013/12/fogcomputing-is-a-new-concept-of-datadistribution/

[9] Fog computing: Data, information, application and services need to be delivered more efficiently to the end user. [Online]. Available: http://clouduser.de/en/analysis/fogcomputing-data-information-applicationand-services-needs-to-be-deliveredmore-efficient-to-the-enduser-22362

[10] D. Xie, N. Ding, Y. C. Hu, and R. Kompella, "The only constant is change: incorporating time-varying network reservations in data centers," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 199–210, 2012.

[11] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 211–222, 2012.

[12] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM, 2011, pp. 233–244.

[13] C. Feng, H. Xu, and B. Li, "An alternating direction method approach to cloud traffic management."

[14] H. Xu, C. Feng, and B. Li, "Temperature aware workload management in geo-distributed datacenters," in *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*. ACM, 2013, pp. 373–374.

[15] H. Xu and B. Li, "Joint request mapping and response routing for geo-distributed cloud services," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 854–862.

[16] D. Niu and B. Li, "An efficient distributed algorithm for resource allocation in large-scale coupled systems," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1501–1509.

[17] H. Xu and B. Li, "A general and practical datacenter selection framework for cloud services," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 2012, pp. 9–16.

[18] S. Narayana, J. W. Jiang, J. Rexford, and M. Chiang, "To coordinate or not to coordinate? wide-area traffic management for data centers," Technical report, Princeton University, Tech. Rep., 2012.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[20] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, 2003.

[21] N. H. Tran, C. S. Hong, and S. Lee, "Joint Congestion Control and Power Control with Outage Constraint in Wireless Multihop Networks," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 2, pp. 889–894, 2012.

[22] ——, "Cross-Layer Design of Congestion Control and Power Control in Fast-Fading Wireless Networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 260–274, 2013.

[23] C. Do, N. Tran, Z. Han, L. Le, S. Lee, and C. S. Hong, "Optimal Pricing for Duopoly in Cognitive Radio Networks: Cooperate or Not Cooperate?" *Wireless Communications, IEEE Transactions on*, 2014, in press.

[24] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[25] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 242–253.

[26] J. V. Spadaro, L. Langlois, and B. Hamilton, "Greenhouse gas emissions of electricity generation chains: Assessing the difference," *IAEA bulletin*, vol. 42, no. 2, pp. 19–28, 2000.

[27] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2. ACM, 2007, pp. 13–23.

[28] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: a platform for high-performance internet applications," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing

Do, Cuong T.; Tran, Nguyen H.; Pham, Chuan; Alam, Md Golam Rabiul; Son, Jae Hyeok; Hong, Choong Seon

01 Qizhao Wang Page 1

18/7/2019 7:22