Optimally Designing Virtualized CDN Maximizing Profit of Content Providers

Noriaki Kamiyama and Yutaro Hosokawa
Fukuoka University, Fukuoka, 814-0180 Japan
E-mail: kamiyama@fukuoka-u.ac.jp

Abstract—**Recently, the virtualized CDN such as Amazon CloudFront has been widely used. In the virtualized CDN, cache servers are provided on virtual machines of cloud datacenters. Using the virtualized CDN, content providers can flexibly select the regions of using cache servers according to the geographical demand pattern of their users, so content providers can expect to reduce the cost of using CDN. However, to maximize the profit of content providers, they need to carefully select the locations of cache servers which affect the quality perceived by users as well as the total cost. In this paper, we propose a method of optimally selecting the geographical regions to use cache servers in the virtualized CDN maximizing the profit of the content providers.**

I. Introduction

Traffic generated by delivering content including web, user generated content (UGC), e.g., YouTube, and rich content produced by content providers, e.g., movie and dramas, has dominated more than 90% of traffic in the Internet [5]. Content delivery networks (CDNs) that use a number of cache servers deployed in multiple networks have been widely used as a method to efficiently transmit content items [23][24][26]. We can expect various effects of using CDNs, e.g., reducing the user-perceived latency, reducing the amount of traffic transmitted within the networks, and decreasing the processing load of origin servers. In traditional CDNs, CDN providers collocated many cache servers in many ISP networks [26], or CDNs providers deployed cache servers at datacenters located at a limited number of positions [16].

Content providers (CPs), e.g., Netflix, which create or license content items try to provide good quality of service to their end users by using CDNs. CPs generate revenue through rich streaming services in which users directly pay to CPs or advertising, e.g., YouTube. Each CP has its special requirement for the design of CDN depending on the geographical demand pattern against the content items provided by the CP. For example, if a large part of demand for content of CP A is generated from users in North and South America, it is desirable for CP A to use cache servers in North and South America. On the other hand, if a large part of demand for content of CP B is generated from users in Europe and Japan, it is desirable for CP B to use cache servers in Europe and South America. However, many existing CDN providers charge CPs based on the amount of traffic delivered from cache servers, and CDN providers generally do not price their services to reflex costs at individual server locations which may vary considerably [21]. Moreover, CDN providers do not distinguish CPs when inserting and removing content items in cache servers, i.e., applying the identical cache-insertion and cache-replacement policy to content items of all CPs, and CPs cannot limit the locations where content items are cached to reduce the cost paid to the CDN providers. Therefore, it is difficult for CDN providers to flexibly provide the CDN service customized for each CP with satisfying its special requirement.

Public cloud services, e.g., Amazon EC2, in which users can use computation resources over networks are widely used. In cloud systems, infrastructure providers (InPs) construct and manage datacenters hosting many physical machines (PMs) on which virtual machines (VMs) are set up as well as networks connecting multiple datacenters and users. By purchasing the right to use VMs from InPs, service providers (SPs) offer various services to end users [1][2][15]. Therefore, the virtualized CDN or cloud-based CDN in which cache servers are provided by VMs on public cloud have gathered wide attention and investigated [11][14], and commercial virtualized CDN services, e.g., Amazon CloudFront [6], have been already provided by major cloud providers. In the virtualized CDN, CPs directly contract with the cloud providers, and each CP can flexibly construct its own CDN by itself. In other words, each CP becomes a CDN provider serving just its content items.

Before starting to use the virtualized CDN, CPs will face the problem of optimally design its CDN, i.e., selecting the regions of using cache servers. In Amazon CloudFront, CPs are charged the usage-based fee whose unit price depends on the region [7]. Using the virtualized CDN service, CPs can flexibly select the regions where cache servers are provided according to the geographical pattern of user demand, so CPs can expect to minimize the cost of using CDN by limiting the regions for using cache servers. However, for end users of the regions without cache servers, content items need to be delivered from cache servers provided in other regions or original servers, so the user-perceived quality, e.g., delay and smoothness in playback of video, will be degraded for users in these regions without cache servers. The number of views of users with degraded service quality will decrease, and the revenue of CPs will also decrease. Therefore, to maximize the profit of CPs, CPs should carefully select the regions to use cache servers considering both the fee paid to the cloud provider and the user-perceived quality. However, the problem of optimally designing the virtualized CDN maximizing the profit of CPs has not been investigated.

In this paper, we propose a method of optimally designing the virtual CDN maximizing the profit of CPs considering both the user-perceived quality and the cost of using cache servers when geographical demand distribution is given. The contribution of this paper is summarized as follows.

- We formalize the profit of CPs which is a function of

user demand as well as the usage price of virtualized CDN, and we also formalize the optimization problem of selecting the regions of using cache servers in the virtualized CDN which maximize the profit of CPs.
- Using the actual price data of Amazon CloudFront, we evaluate the effectiveness of the proposed design method of virtualized CDN by comparing the average profit of CPs in the proposed method with the two nave approaches: *No Cache*, without using cache servers in all the regions, and *All Cache*, with using cache servers at all the regions.

In Section II, we briefly summarize the related works, and we describe the conditions and models assumed in Section III. In Section IV, we describe the proposed design method of virtualized CDN, and we show numerical results in Section V. Finally, we conclude this paper in Section VI.

II. Related Works

Several works have investigated methods of allocating the location of caching each content minimizing the cost of CDN providers. For example, Marchetta et al. proposed to optimally allocate the cached location of each content minimizing the cost paid to cloud providers by CDN providers [19]. Dodout et al. proposed an architecture of designing the cached locations of content items for each content provider on the virtualized CDN [9]. Moreover, Hu et al. proposed a method of periodically optimizing the storage capacity of virtual cache servers and the location of cached copies with the target of minimizing the total cost of CDN providers with the constraint of the maximum distance between delivery servers and users [12]. Chen et al. proposed a method of optimally designing both the location of cached copies and distribution paths, i.e., pairs of source and destination of content deliveries, with the target of minimizing the cost of the virtualized CDN providers with the constraint that the requirement of user QoS was satisfied [3]. Llorca et al. proposed a method of jointly allocation the content location and routing between delivery servers and users minimizing the total cost of CDN providers [17]. However, optimally allocating the location of caching each content requires a huge amount of computational time, so it seems difficult for CPs to obtain the solution when a large number of content items are provided. Moreover, commercial virtualized CDN services, e.g., Amazon CloudFront, do not provide CPs a function of freely controlling the cached location with the granularity of content.

Several authors have proposed to differentiate CPs in treating their content items on cache servers. For example, Gourdin et al. evaluated the impact of allocation method of storage space of cache servers to each CP on the profit of CDN providers [10]. Chu et al. proposed to allocate the cache storage space to each CP depending on the utility which is a function of cache hit ratio maximizing the total utility of CPs [4]. Dehghan et al. and Neglia et al. also proposed to apply different cache replacement policy to content items of different CPs with the target of maximizing the total utility [8][22]. Moreover, Yala et al. proposed a method of allocating virtual cache servers on physical machines with jointly optimizing the cost of CDN provider and the service availability [27]. However, these methods did not consider the issue how to select the regions of using cache servers on the virtualized CDN.

III. Assumptions

A. Virtualized CDN Service

We assume one virtualized CDN service is provided to CPs by one cloud provider, and CPs can use virtualized cache servers provided at N regions. As the virtualized CDN service, we assume Amazon CloudFront which provides cache servers at seven regions: North America, Europe, South Africa, Asia, India, Australia, and South America [7]. We consider just a single CP. Let R_n denote region n, and we define C_n as the cache servers provided in R_n. In Amazon CloudFront, CPs are charged for delivering content from cache servers to users, uploading content to origin servers, and using the name resolution service of the cloud provider [7]. However, to simplify the analysis, we consider just the fee for delivering content to users from cache servers because this seems to the dominant factor of the total fee.

The unit price of delivering content from cache servers to users depends on both the region of cache servers and the total amount of data transmitted from the cache servers within one month. We define $p_n(v)$ as the unit price when delivering content from C_n to users when the total amount of data transmitted from C_n is v. Origin servers which store originals of content items of CPs are also managed and operated by the cloud provider, and CPs can use the CDN service by just uploading their content items to the origin servers of the cloud provider.

Let x_n denote a binary variable which takes unity if CP s decided to use C_n or takes zero if CP s decided not to use C_n. Moreover, we define $\boldsymbol{X}$ as the set of R_n with $x_n = 1$. In other words, $\boldsymbol{X}$ is the set of regions where CP s decided to use cache servers. The origin servers are provided in just a single region R_o, and CP s does not use C_o, i.e., $x_o = 0$, because content can be delivered to users of R_o from origin servers. We assume that sufficient amount of storage capacity of virtual cache servers is provided, and content is always delivered from C_n to the requesting users if C_n is selected as delivering servers. When users of R_n with $x_n = 0$ request content, content is delivered from the nearest cache servers of $\boldsymbol{X}$ or R_o giving the minimum average RTT to the requesting users.

B. User Demand

We assume that each user of CP s views content W times on average within a month if he or she totally satisfies the quality of video streaming service of CP s. It is anticipated that the number of views of users will decrease as the quality of video streaming is degraded, i.e., increasing the frequency of video freeze or noise occurrence. Moreover, the quality of video streaming will be degraded as the RTT (round-trip time) between users and servers delivering content increases. Therefore, we use the depression model of user demand [18] which is a monotonically decreasing function of delay, and we assume that $w(d)$, the average view count of each user when the RTT is d, is given by

$$w(d) = W\mathrm{e}^{-\alpha d}, \tag{1}$$

where α is a sensitivity parameter of users against delay. $w(d)$ agrees with the maximum value, i.e., $w(d) = W$, when d is zero, and $w(d)$ exponentially decreases as d increases. As α increases, $w(d)$ decreases more sharply with increase of d. Let $d_{i,j}$ denote the average RTT between R_i and R_j, and

w_n, the average view count of each user of R_n, is given by $w_n = W\mathrm{e}^{-\alpha d_{n,y}}$ when content items are delivered from R_y to users of R_n.

C. *Revenue of CP*

There are mainly two business models on video streaming services: SVOD (subscription video on demand), e.g., Netflix, and AVOD (advertising video on demand), e.g., YouTube. In the case of SVOD, CPs directly charge end users for video streaming services, and SVOD is further classified into two types: flat-rate charging and PPV (pay per view). In the flat-rate charging model, users can view video content as many as they want by paying a fixed monthly fee to CPs, whereas users pay fee to CPs for each view in the PPV. In the case of AVOD, on the other hand, CPs do not directly charge to end users, and CPs obtain revenue from advertisers by displaying video content as well as advertisements. In any business models, we can assume that CP s obtains r fee for each view. In the case of the SVOD of flat-rate charging, we can calculate r by dividing the monthly charge by the potential average number of views per user W.

IV. Optimum Design Method of Virtualized CDN

As mentioned in Section III-A, for requests from users of R_u, content is delivered from the region giving the minimum average RTT to R_u among the candidate regions of $\boldsymbol{X} \cup R_o$. In other words, let z_u denote the region from which content items are delivered to users of R_u, and z_u is given by

$$z_u = \arg \min_{z \in \boldsymbol{X} \cup R_o} d_{z,u}. \tag{2}$$

Let B denote the average size of content, and let q_n denote the number of subscribers of CP s in R_n. From (1), v_i, the average amount of data transmitted from C_i of $x_i = 1$ to users of CP s within one month, is obtained by

$$v_i = \sum_{u \in \boldsymbol{X} \cap \{u;\ z_u = i\}} q_u W \mathrm{e}^{-\alpha d_{i,u}} B. \tag{3}$$

The total fee paid to the cloud provider by CP s is $\sum_{i \in \boldsymbol{X}} p_i(v_i) v_i$.

As mentioned in Section III-C, CP s obtains revenue r from a user for each content delivery. Therefore, $P(\boldsymbol{X})$, the average profit of CP s within one month for given $\boldsymbol{X}$, is derived as

$$P(\boldsymbol{X}) = \sum_{u=1}^{N} q_u r W \mathrm{e}^{-\alpha d_{z_u,u}} - \sum_{i \in \boldsymbol{X}} p_i(v_i) v_i. \tag{4}$$

We define the optimization problem maximizing $P(\boldsymbol{X})$:

$$\begin{aligned} \max \quad & P(\boldsymbol{X}) & (5)\\ s.t. \quad & x_n = \{0,\ 1\},\ 1 \leq n \leq N, & (6)\\ & x_o = 0. & (7) \end{aligned}$$

V. Numerical Evaluation

A. *Evaluation Conditions*

In the numerical evaluation, we assume a CP providing video streaming service of the SVOD type with flat-rate charging, e.g., Netflix. With the assumption that the average transmission bit rate of Netflix is 4.4 Mbps [20], and the average length of content items is 100 minutes, the average size of content, B, is 3 GB. As mentioned in Section III-A, Amazon CloudFront provides cache servers at seven regions. To obtain the setting values of $d_{i,j}$ between region i and region j, we measured the average RTT using *ping* command on the virtual machines in these regions of Amazon EC2. However, we cannot setup virtual machines in Africa on Amazon EC2, we selected the seven regions as North America, Europe, North Asia, Singapore, India, Australia, and South America. We assume that delay was zero when delivering servers and users existed in the same region, i.e., $d_{n,n} = 0$ for any n.

We set $p_n(v)$ according to the price list of Amazon CloudFront [7] which provides the unit price for each of the seven volume zones: initial 10 TB, next 40 TB, next 100 TB, next 350 TB, next 524 TB, next 4 PB, and more than 5 PB. For example, when delivering content from cache servers in North America, $p_n(v)$ was set to 0.085, 0.08, 0.06, 0.04, 0.03, 0.025, and 0.02 USD per GB in each volume zone, respectively. As v increased, $p_n(v)$ decreased thanks to the volume discount. The unit price in North America and Europe was the lowest, whereas that in South America was highest, and those in the other regions were moderate. The ratio of users in Netflix was about 0.03 [25]. We set q_n to the population in each region [13] multiplied by 0.03. We set W, the average potential number of views per each user within one month, to 15, and we set r, the revenue of the CP obtained by each content delivery, to 0.6 USD which was obtained by monthly flat-fee of Netflix, 10 USD, divided by W.

B. *Optimum Region Count Using Cache Servers*

Table I summarizes X^*, the optimum number of regions in which the cache servers was used, i.e., $x_n = 1$, designed by the proposed method, for various values of α between zero and 10 [18] when placing the origin servers in one of the seven regions. As α increased, the sensitivity of users to delay increased, so X^* increased. When α was close to zero, the merit of using cache servers was small, so the optimum virtualized CDN was close to the case without using any cache servers (denoted as *No Cache*). On the other hand, when α was close to 10, the merit of using cache servers was large, so the optimum virtualized CDN was close to the case using cache servers in all the regions excluding R_o (denoted as *All Cache*).

TABLE I
Optimum region count selected to use cache servers

Region placing	α										
origin servers	0	1	2	3	4	5	6	7	8	9	10
North America	0	1	2	2	3	4	4	4	6	6	6
Europe	0	2	2	2	3	4	4	4	6	6	6
North Asia	0	1	2	3	3	4	4	4	6	6	6
Singapore	0	1	2	2	3	4	4	4	6	6	6
India	0	1	3	3	4	4	4	5	6	6	6
Australia	0	2	3	3	3	4	4	4	6	6	6
South America	0	2	3	3	4	5	5	5	6	6	6

C. *Monthly Profit of CP*

We compare the average monthly profit of the CP in the proposed design method as well as the two cases: *No Cache* and *All Cache*. Figure 1 plots the average profit of the CP in each of the three methods against α when placing the origin servers in North America. The following tendencies were also observed when placing the origin servers in other areas as well. In All Cache, content items were always delivered from the cache servers or origin servers located in the same region with the requesting user, so delay was always zero, and the profit of the CP was independent of α. In No Cache, content items were always delivered from the origin servers, so delay was the largest among the three methods, and the profit of the CP rapidly decreased as α increased. As a results, the average profit of the CP in All Cache when α was small and that in No Cache when α was large were degraded compared with the other methods.

On the other hand, the proposed method optimally selected the regions in which cache servers were used according to the user demand as well as the sensitivity of users to delay, so the average monthly profit of the CP in the proposed method was the highest among the three methods in the entire range of α. For example, when α was zero, the proposed method increased the profit of the CP about 40 % compared with All Cache, and the proposed method increased the profit of the CP about 120 % compared with No Cache when α was 10.

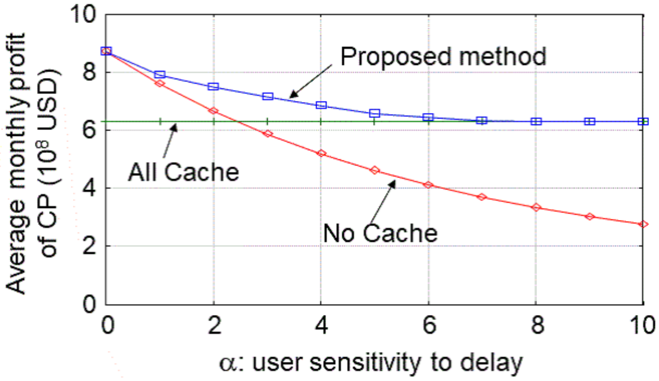

Fig. 1. Average monthly profit of CP against α when placing origin servers in North America

VI. Conclusion

Using the virtualized CDN, CPs can flexibly design CDN by selecting the regions using cache servers according to the demand pattern, and CPs can expect to decrease the cost of CDN. However, to maximize the profit, CPs are required to carefully select the cache regions considering both the user-perceived quality and the cost of using cache servers in each region. In this paper, we proposed a method of optimally selecting the cache regions when the unit price of using cache servers, the user demand, and the sensitivity function of users to delay were given. Through the numerical evaluation assuming the CP of Netflix type, we confirmed that the proposed method can increase the profit of the CP by about 40 % or 120 % compared with the case with using cache servers at all regions and the case without using cache servers at all, respectively.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 18K11283.

References

[1] D. Ardagna, B. Panicucci, and M. Passacantando, A Game Theoretic Formulation of the Service Provisioning Problem in Cloud Systems, WWW 2011.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, F. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, A View of Cloud Computing, Communications of The ACM, Vol. 53, No. 4, pp. 50-58, Apr. 2010.

[3] F. Chen, K. Guo, J. Lin, and T. K. Porta, Intra-cloud Lightning: Building CDNs in the Cloud, IEEE INFOCOM 2012.

[4] W. Chu, M. Dehghan, D. Towsley, and Z. L. Zhang, On Allocating Cache Resources to Content Providers, ACM ICN 2016.

[5] Cisco Visual Networking Index: Forecast and Methodology, 2016-2021.

[6] Amazon CloudFront, https://aws.amazon.com/cloudfront/

[7] Amazon CloudFront Pricing, On-Demand Pricing, https://aws.amazon.com/cloudfront/pricing/

[8] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, and Y. C. Tay, A Utility Optimization Approach to Network Cache Design, IEEE INFOCOM 2016.

[9] F. Dudout, P. Harsh, S. Ruiz, A. Gomes, T. M. Bohnert, A Case for CDN-as-a-Service in the Cloud: A Mobile Cloud Networking Argument, ICACCI 2014.

[10] E. Gourdin, P. Maille, G. Simon, and B. Tuffin, The Economics of CDNs and Their Impact on Service Fairness, IEEE Transactions on Network and Service Management, Vol.14, No.1, pp.22-33, Mar. 2017.

[11] N. Herbaut, D. Negru, Y. Chen, P. A. Frangoudis, and A. Ksentini, Content Delivery Networks as a Virtual Network Function: a Win-Win ISP-CDN Collaboration, GLOBECOM 2016.

[12] M. Hu, J. Luo, Y. Wang, and B. Veeravalli, Practical Resource Provisioning and Caching with Dynamic Resilience for Cloud-Based Content Distribution Networks, IEEE Transactions on Parallel and Distributed Systems, Vol.25, No.8, pp.2169-2179, Aug. 2014.

[13] Internet World Stats, https://www.internetworldstats.com/stats.htm

[14] Y. Jin, Y. Wen, G. Shi, G. Wang, and A. V. Vasilakos, CoDaaS:An Experimental Cloud-Centric Content Delivery Platform for User-Generated Contents, ICNC 2012.

[15] W. Li, P. Svard, J. Tordsson, and E. Elmroth, A General Approach to Service Deployment in Cloud Environments, CGC 2012.

[16] Z. Li, et al., In a Telco-CDN, Pushing Content Makes Sense, IEEE Transactions on Network and Service Management, Vol. 10, No. 3, pp. 300-311, Sep. 2013.

[17] J. Llorca, C. Sterle, A. M. Tulino, N. Choi, A. Sforza, and A. E. Amideo, Joint Content-Resource Allocation in Software Defined Virtual CDNs, CCSNA 2015.

[18] R. Ma, J. Wang, and D. M. Chiu, Paid Prioritization and Its Impact on Net Neutrality, IEEE Journal on Selected Areas in Communications, Vol. 35, No. 2, pp. 367-379, Feb. 2017.

[19] P. Marchetta, J. Llorca, A. M. Tulino, and A. Pescape, MC3: A Cloud Caching Strategy for Next Generation Virtual Content Distribution Networks, IFIP Networking 2016.

[20] J. Martin, Y. Fu, N. Wourms, and T. Shaw, Characterizing Netflix bandwidth consumption, IEEE CCNC 2013.

[21] M. K. Mukerjee, I. N. Bozkurt, D. Ray, B. M. Maggs, S. Seshan, and H. Zhang, Redesigning CDN-Broker Interactions for Improved Content Delivery, ACM CoNEXT 2017.

[22] G. Neglia, D. Carra, and P. Michiardi, Cache Policies for Linear Utility Maximization, IEEE INFOCOM 2017.

[23] E. Nygren, R. Sitaraman, and J. Sun, The Akamai Network: A Platform for High-Performance Internet Applications, ACM SIGOPS 2010.

[24] J. Ott, M. Sanchez, J. Rula, and F. Bustamante, Content Delivery and the Natural Evolution of DNS, ACM IMC 2012.

[25] Statista, the statistics portal, https://www.statista.com/topics/842/netflix/

[26] A. Su, D. Choffnes, A. Kuzmanovic, and F. Bustamante, Drafting Behind Akamai: Inferring Network Conditions Based on CDN Redirections, ACM Trans. Networking, Vol. 17, No. 6, pp. 1752-1765, Dec. 2009.

[27] L. Yala, P. A. Frangoudis, G. Lucarelli, and A. Ksentini, Balancing between cost and availability for CDNaaS resource placement, IEEE GLOBECOM 2017.