Nash Bargaining Solution-based Datacenter Selection under Cloud Content Delivery Network Environments

Heejae Kim, Yun-Gi Ha, and Chan-Hyun Youn

Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Korea

Email: {*kim881019, milmgas, chyoun*}*@kaist.ac.kr*

1. Introduction

Integrating Content Delivery Networks (CDNs) with cloud computing is an emerging issue in CDN technology recently. Amazon and Rackspace provides CloudFront [1] and Rackspace CDN [2] as the solutions for it using their clouds, and many CDN providers such as Akamai and Limelight have adopted cloud computing to improve performance of their CDNs [3,4]. Netflix started to provide their services using Amazon Web Service [5], and Google have boosted speed of YouTube using its cloud [6]. We refer to this kind of CDN combined with cloud computing as a cloud CDN [7]. Because there are usually many short busty periods in time series of end user demand of CDN [8], the use of cloud computing can give huge elasticity to CDN. Therefore, the cloud CDN can handle the dynamic demand adaptively as well as acquire enough resource even if the amount of resource which they have is relatively small. The cloud CDN providers provide their services using resources in cloud datacenter. Because quality of service (QoS) in the cloud CDN depends on the locations of the resources, datacenter selection is one of the important issues in the cloud CDN. Many researches have been studied with focusing on caching server placement [9, 10], data placement [11], domain name system (DNS) server placement [12], end user request placement [13,14,15].

In this paper, we present a novel datacenter selection algorithm to place caching server for the cloud CDN. The algorithm is based on Nash bargaining solution (NBS) [16,17] which is an attractive method for this problem guaranteeing pareto efficiency, symmetry, invariance to equivalent payoff representations, and independence of irrelevant alternatives. Also, the algorithm considers predicted end user demand and virtual machine (VM) reservation based on autoregressive integrated moving average (ARIMA) model and our previous work, C-VMR [18] respectively. In evaluation, we compare the algorithm with uniform and proportional fair sharing.

2. Datacenter Selection Algorithm

Figure 1 shows the cloud CDN environment considered in this paper. In the figure, a cloud CDN provider has several available geo-distributed datacenters. The cloud CDN provider leases VMs from the datacenters to build caching servers to handle caching server requests of the content providers. When the content providers request the use of caching servers to the cloud CDN provider, the cloud CDN provider places the caching servers to available datacenters by the datacenter selection algorithm. We suppose that it is predetermined that which datacenter handles content requests of end users in each region. Because the algorithm considers end user demand, the end user demand in every region of each content provider should be monitored and managed consistently. When end users send content requests to the DNS servers, the DNS servers determine a caching server and forward the packets to the caching server. Because QoS in content services such as online game and video streaming is highly sensitive to the factors such as network latency and throughput, the following datacenter selection algorithm is designed to handle caching server requests of the content providers effectively.

The datacenter selection algorithm is based on the NBS in consideration of end user demand prediction and VM reservation.

End User Demand Prediction

End user demand prediction is performed based on the ARIMA model similarly with [19, 20]. Before applying the model, time series of the demand should be preprocessed to make it stationary, and the order of the model should be determined.

VM Reservation

On-demand VMs (OVMs) and reserved VMs (RVMs) are two major VM types in the current cloud industry. These refer to VMs whose leasing time is relatively short such as an hour and long such as a month and respectively. Obviously, the price of RVMs in the unit time is set to be `lower than that of OVMs. Therefore, using the appropriate number of RVMs gives a benefit to the cloud CDN provider. For VM reservation in the datacenter selection algorithm, we use the C-VMR [18] to determine the appropriate number of RVMs to be leased. The mechanism of the C-VMR is as follows. The C-VMR is performed every the fixed period. At each epoch t, demand in period $[t,t + T_p]$ is predicted, and it is determined that how many RVMs are to be leased as depicted in (1) where $r_l(t)$ is the number of RVMs to be leased at epoch t, $r_e(t)$ is the number of

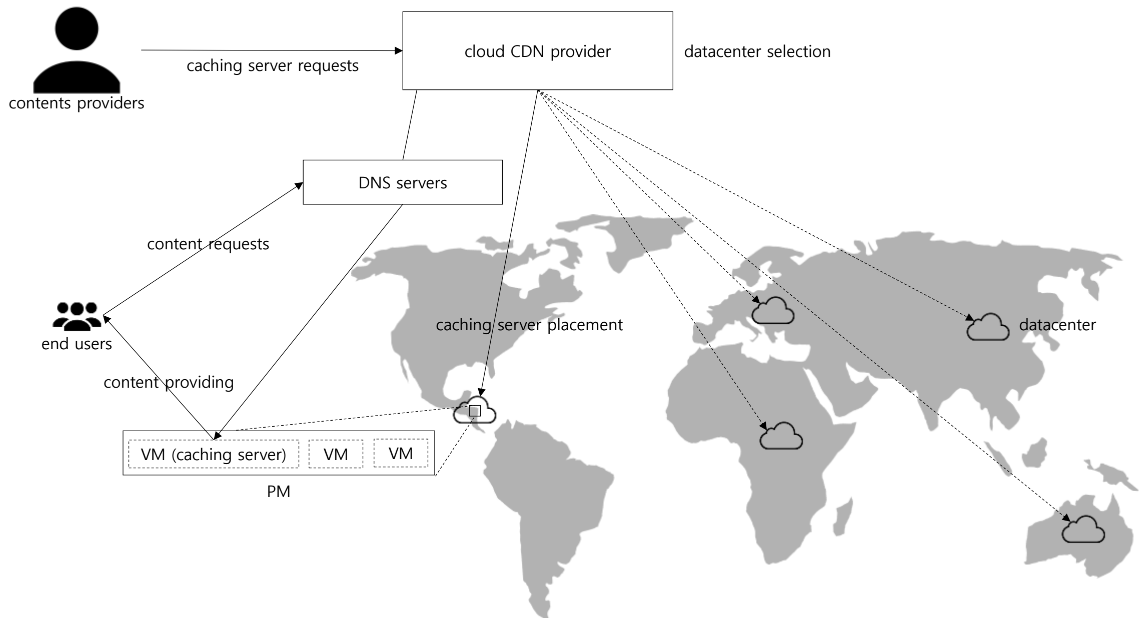


Figure 1 Cloud CDN environment.

RVMs which is available to the cloud CDN provider at the epoch t, and $d_p(t)$ is the predicted capable VM demand in period [t,t + 1]. We note that the demand refers to the number of VMs which are capable to provide the caching server requests of the content providers.

$$r_l(t) = \left\lfloor \frac{1}{T_p} \sum_{k=t}^{t+T_p} d_p(k) - r_e(k) \right\rfloor . \qquad (1)$$

Algorithm 1. C-VMR

Input	time series of historical capable VM demand in period [t - T_h,t - 1] where T_h is a period to be used as the historical demand in demand prediction
1:	predict capable VM demand in period [t,t + T_p]
2:	Obtain $r_l(t)$
3:	lease additional RVMs as much as $r_l(t)$

NBS-based Datacenter Selection

We present a scheme for the NBS-based data selection in this section. The goal of the scheme is to determine the number of VMs which is to be created newly for each datacenter.

Based on the NBS, we formulate an optimization problem in consideration of end user demand and VM reservation as depicted in (2), (3), and (4).

$$\text{maximize} \sum_i \log\left(x_i - \frac{c \cdot \delta_i}{n} \right) \qquad (2)$$

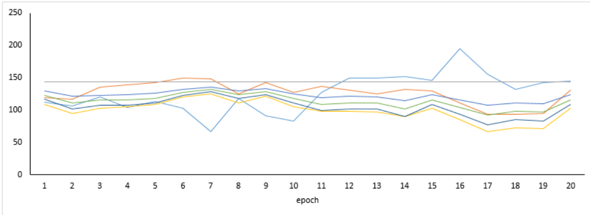
$$\text{subject to} \sum_i x_i - \left(s + \sum_i \rho_i \right) \qquad (3)$$

$$x_i \geq \rho_i, \forall i. \qquad (4)$$

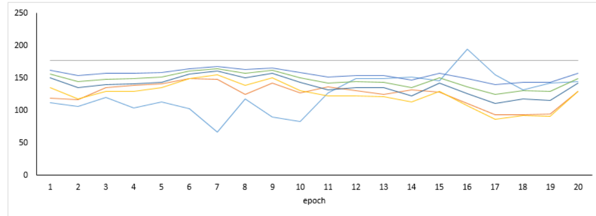
$u(X)$ denotes an utility function where x_i in vector X is a decision variable and represents the number of VMs to be operated in datacenter $i \in I$. We note that I is set of available datacenters. Also, n is the number of capable end user requests in a VM, δ_i is the predicted end user demand of datacenter i, c is a control parameter. (3) is the constraint to guarantee that the sum of x_i is equal to the sum of the number of caching server requests s and the number of available RVMs where ρ_i denotes the number of available RVMs in datacenter i. (4) is the constraint to limit the minimal number of xi because x_i cannot be less than ρ_i. Finally x_i – ρ_i represents the number of VMs which is to be created newly for datacenter i.

3. Evaluation

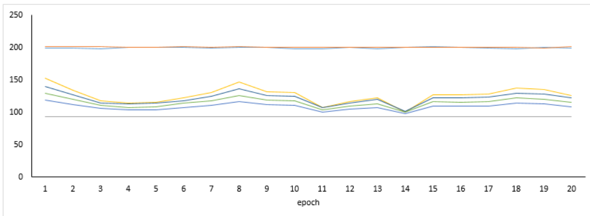
In this section, we evaluate the scheme for the NBS-based datacenter selection. For the evaluation, we generated a different end user demand for each datacenter arbitrarily and the demand is unit of the number of end user requests at an epoch. ASTSA package in R [21] is used for the demand prediction using the ARIMA model. The window size is set to 30. In the evaluation, we suppose that there are 3 available datacenters (Datacenter 1, Datacenter 2, and Datacenter 3), and the number of RVMs of each datacenter is 30, 80, and 60 respectively. Also, n is set to 10.



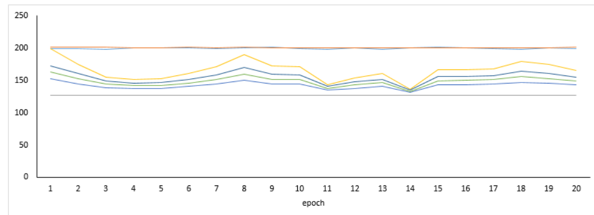
(a)



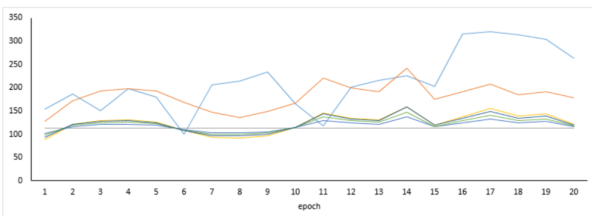
(b)



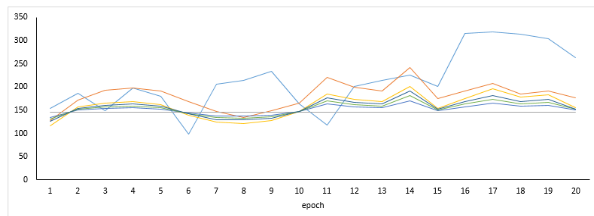
(c)



(d)



(e)



(f)

Figure 2 Evaluation results: Sky blue and orange represent actual end user demand / 10 and predicted end user demand / 10 respectively. Gray, yellow, blue, yellow green, and dark blue represent the datacenter selection result by uniform fair sharing, proportional fair sharing, NBS (c = 0.5), NBS (c = 0.7), and NBS (c = 0.9) respectively. (a) Datacenter 1, the number of caching server requests = 350. (b) Datacenter 1, the number of caching server requests = 450. (c) Datacenter 2, the number of caching server requests = 350. (d) Datacenter 2, the number of caching server requests = 450. (e) Datacenter 3, the number of caching server requests = 350. (f) Datacenter 3, the number of caching server requests = 450.

We compare the algorithm with uniform and proportional fair sharing. The uniform and proportional fair sharing are the approaches which divide the caching server requests to each datacenter uniformly and proportionally to end user demand respectively.
Figure 2 shows the result. As depicted in the figure, the trend is similar between Fig. 2(a), (c), (e) and (b), (d), (f) respectively. Demand prediction of Figure 2(c) and (d) are more accurate than Figure 2(a), (e) and (b), (f) respectively. It is because end user demands of Figure 2(c) and (d) are rarely fluctuated. Also, in all figures in Figure 2, the datacenter selection results of NBS are in between those of uniform and proportional fair sharing. Therefore, we can control *c* adaptively by system status, service level agreement (SLA), and so on. Therefore, cloud CDN providers can provide flexible services to contents providers.

4. Conclusions

In this paper, we presented the NBS-based datacenter selection for cloud CDN in geo-distributed clouds. To achieve effective datacenter selection for cloud CDN, we proposed the NBS-based datacenter selection algorithm in consideration of demand prediction and VM reservation. In evaluation, we compared the algorithm with uniform and proportional fair sharing. As on-going and future work, we are extending the algorithm to consider SLA such as latency, response time, throughput, etc.

Acknowledgement

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2014, and the MSIP under the ITRC (Information Technology Research Center) support program (NIPA-2014(H0301-14-1020)) supervised by the NIPA (National IT Industry Promotion Agency).

References

[1] CloudFront, available: http://aws.amazon.com/ko/cloudfront/.

[2] Rackspace CDN, available: http://www.rackspace.com/cloud/cdn-content-delivery-network/.

[3] Akamai, available: http://www.akamai.co.kr/.

[4] Limelight, available: http://www.limelight.com/.

[5] Google Cloud Boosting YouTube Upload Speeds, available: http://blogs.wsj.com/digits/2011/02/14/google-cloud-boosting-youtube-upload-speeds/.

[6] AWS Case Study: Netflix, http://aws.amazon.com/ko/solutions/case-studies/netflix/.

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

[8] V. Aggarwal, V. Gopalakrishman, R. Jana, K. K. Ramakrishnan, and V. A. Vaishampayan, "Optimizing Cloud Resources for Delivering IPTV Services through Virtualization," in COMSNETS, 2012.

[9] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the Electric Bill for Internet-scale Systems," in SIGCOMM, 2009.

[10] Q. Zhang, Q. Zhu, M. F. Zhani, R. Boutaba, and J. L. Hellerstein, "Dynamic Service Placement in Geographically Distributed Clouds" in JSAC, vol. 31, no. 12, 2013.

[11] S. Agarwal, J. Dunagan, N. Jain, S. Saroju, A. Wolman, and H. Bhogan, "Volley: Automated Data Placement for Geo-distributed Cloud Services," in NSDI, 2010.

[12] H. Xu and B. Li, "A General and Practical Datacenter Selection Framework for Cloud Services," in CLOUD, 2012.

[13] H. Qian and Q. Wang, "Towards Proximity-aware Application Deployment in Geo-distributed Clouds," in ACSA, vol. 2, no. 3, 2013.

[14] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "DONAR: Decentralized Server Selection for Cloud Service," in SIGCOMM, 2010.

[15] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, F. C. M. Lau, "Scaling Social Media Applications into Geo-distributed Clouds," in INFOCOM, 2012.

[16] J. F. Nash, Jr., "The Bargaining Problem," in Econometrica, vol. 18, no. 2, 1950.

[17] M. J. Osborne, An Introduction to Game Theory, Oxford University Press, 2009.

[18] H. Kim, Y. Ha, Y. Kim, and C. H. Youn, "A VM Reservation-based Cloud Service Broker and Its Performance Evaluation," in CloudComp, 2014.

[19] P. J. Brockwell and R. A. Davis, Introduction to Time Serires and Forecasting, Springer, 2002.

[20] W. Fang, Z. Lu, J. Wu, Z. Cao, "RPPS: A Novel Resource Prediction and Provisioning Scheme in Cloud Data Center," in SCC, 2012.

[21] "R", available: http://r-project.org/

Heejae Kim received the B.S. and M.S. degree in electrical engineering from KAIST, Daejeon, Korea, in 2011 and 2013, respectively. Currently, he is a Ph.D. candidate in electrical engineering at KAIST. His research interests include cloud-based network, cloud service broker, and resource management in cloud datacenters.

Yun-Gi Ha received the B.S. degree in electronic and electrical engineering from Sungkyunkwan University, Suwon, Korea, in 2013 and the M.S. degree from KAIST, Daejeon, Korea, in 2014. Currently, he is a Ph.D. candidate in electrical engineering at KAIST. His research interests include scientific workflow system management and efficient resource management in cloud computing environment.

Chan-Hyun Youn received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Korea, in 1981 and 1985, respectively, and the Ph.D. degree in electrical and communications engineering from Tohoku University, Japan, in 1994. From 1986 to 1997, he worked for KT Telecom Network Research Laboratories, where he was a team leader of high-speed networking projects. Since 2009, he has been a professor in the department of electrical engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. Where, he is a dean of office of planning and budgets and is also a director of Grid Middleware Research Center, KAIST. His research areas are cloud, computing workflow system, distributed system, communication middleware, and e-Health application services. Dr.Youn was the Editor-in-Chief in the Korea Information Processing Society, an editor of the Journal of Healthcare Engineering, U.K., and was the head of Korea branch (computer section) of the IEICE, Japan, during 2009–2010. He is a member of the IEEE.