Optimum Content Pollution Attack Using Genetic Algorithm in Multi-Layer CDN

Liu Jiaqi Noriaki Kamiyama
College of Information Science and Engineering, Ritsumeikan University

1. Introduction

In recent years, as the content delivery network (CDN) is more widely used, the attacks against CDN are also increasing [1]. Although there has been a lot of research on cache attacks against CDN, it is rare to analyze them from the attacker's perspective. Therefore, we aims to calculate the optimal attack strategy of attackers through genetic algorithm to help CDN providers find vulnerable points in the system and find out how to defend more efficiently. We have analyzed the characteristics of attacks in previous studies [2]. This paper takes uses genetic algorithm to analyze the weakness of the CDN and the method of strengthening defense.

2. Analytical Model

In this paper, we use response time to measure the CDN performance. We use the M/M/1 queue model to derive the average response time of contents. We assume that server requests follow zipf law, and let M denote the number of contents provided by CDN. Let λ_i denote the Poisson arrival rate of request for content i, and we define $1/\mu$ as the mean of the exponentially distributed service time of CS. Using the M/M/1 queue model, we can obtain W, the average response time, by

$$W = \frac{1}{\mu - \sum_{i=1}^{M} \lambda_i}. \tag{1}$$

In a network with cache server (CS), there are two cases based on whether the requested content is cached or not. The origin server stores all the provided contents, and the CS stores a part of contents. When the content requested by a user does not exist in the CS, which is called *cache miss*, the CS will obtain the content from the origin server, store the content according to the cache replacement policy, and send the content to the user. On the other hand, when the content requested by the user exist in the CS, which is called *cache hit*, the CS will update the cache storage according to the cache replacement policy and deliver the content to the user. Since the LRU (least recently used) is a common cache replacement policy in CDN, this paper assumes that all CSes adopt the LRU.

Since the latency between users and CDN CSes depends on the networks between them, and it will not be affected by attacks against CSes, we do not consider the latency between users and CSes. However, we consider the latency between the CSes and the origin server because this latency will affect the impact of the attack strategy on the effect of attack. We define T as the latency between a CS and the origin server which is the latency between the time instance that the CS sends a request to the origin server and the time instance that the CS receives the requested content from the origin server. The average response time of the source server and the CS will be different according to the request rate, so we define the average response time of the origin server as W_o and the average response time of the CS as W_c. When the requested content exists in the CS.

Because each content has a different cache hit ratio, let h_i denote the cache hit ratio of content i. We can obtain the average response time of content i, W_i, by

$$W_i = h_i W_c + (1 - h_i)(W_c + T + W_o). \tag{2}$$

We use the Che-approximation to predict the hit ratio h_i of each content i on the CS [3]. Let C denote the capacity of the CS, and the maximum number of contents that can be stored in the CS is C. We assume that the average request arrival rate of content i is λ_i, and from the Che-approximation, we can obtain the cache hit ratio of content i, h_i by

$$h_i \approx 1 - e^{-\lambda_i t_c}, \tag{3}$$

where t_c is the characteristic time of the CS, and it is obtained by solving

$$\sum_{i=1}^{M} h_i = C. \tag{4}$$

Multilayer CDN are designed to provide faster service and to defend against DDoS attacks to some extent, and we focus on this model to evaluate and analyze the CPA against CSes in this paper.

The multilayer CDN model is composed of multiple independent CSes with multiple layers, and we assume CSes of two layers, L1 and L2, as shown in Fig.1. When a user requests a content, the CS accommodating the requesting user at L1 checks whether the requested content exists or not in its cache storage. If the requested content exists in the cache storage, the CS of L1 sends the requested content to the user. Otherwise, the request is forwarded to the CS of L2 connecting to the CS of L1. If the requested content does not exist in the CS of L2, the origin server which stores all M contents sends the requested content to the user.

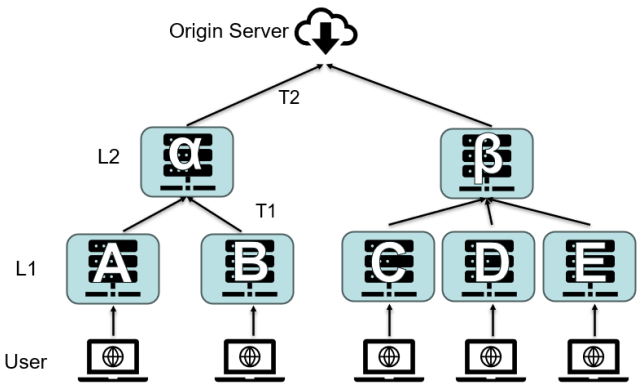

Fig. 1: Multilayer CDN model

We further assume that there are two CSes, α and β, at L2, two CSes, A and B, at layer 1 connecting CS α, and three CSes, C, D, and E, at layer 1 connecting CS β. Let T_1 denote the latency between L1 CSes and L2 CSes, and T_2 denote the latency between L2 CSes and the origin server. Moreover, let W_A, W_α, and W_o denote the average response time of contents at CS A, CS α, and the origin server, respectively.

When the requested content exists in the CS A at L1, the average response time in the model, r_A, is W_A. We define r_α as the average response time when the request is cache miss in CS A and forwarded to the CS α at L2. Moreover, we also define r_o as the average response time when the request is forwarded to the origin server. We can obtain r_α and r_o

by

$$r_\alpha = W_A + W_\alpha + T_1, \quad (5)$$
$$r_o = W_A + W_\alpha + Wo + T_1 + T_2. \quad (6)$$

Let h_i^A denote the cache hit ratio of content i in CS A and h_i^α denote the cache hit ratio of content i in CS α. The average response time of content i when request arrives at CS A, $R_A(i)$, is obtained by

$$R_A(i) = h_i^A r_A + (1-h_i^A)h_i^\alpha r_\alpha + (1-h_i^A)(1-h_i^\alpha)r_o \quad (7)$$

By summing $R_A(i)$ among all i in M, we can obtain the average response time of request when accessing CS A by

$$R_A = \frac{\sum_{i=1}^{M} R_{a_i}}{M}. \quad (8)$$

In the same way, we can also obtain the average response time when accessing each of other CSes.

3. Genetic Algorithm

To investigate the potential threat of the CPA, we need to find the optimal strategy of attackers to maximize the average response time over all CSes in the system. The Genetic algorithms (GA) are commonly used to generate high-quality solutions to optimization by relying on biologically inspired operators such as mutation, crossover and selection.

The optimal attack strategy can reflect how an attacker distributes requests to different servers in different content. To achieve GA, we randomly set s_i^n as the initialized chromosome firstly, which $0 < s_i^n < 1$. Then ,to achieve crossover, set any two chromosomes as a group and randomly exchange of s_i^n in groups. Furthermore, to achieve mutation, randomly change a s_i^n, which $0 < s_i^n < 1$. To achieve selection, add attacker's request to the normal request and calculate the average of R of each CS as fitness. Then select the best chromosomes from the parents and children. Finally, repeats until N generations.

Let λ_T denote the maximum requests sent by the attacker, and s_i^n represents the proportion of requests that the attacker allocates to server n from λ_T. We can obtain the request λ_i^n by

$$\lambda_i^n = \frac{s_i^n}{\sum_{n=A}^{E}\sum_{i=1}^{M} s_i^n}. \quad (9)$$

4. Numerical Evaluation

The setting parameters of the experiments are shown in Table 1. We assume contents provided by CDN occupy the same amount of space in the cache and have different request arrival rate depending on zipf law.

表 1: Simulation parameter setting in basic case

Paramater	Value
Number of contents provided, M	5
Cache size	2
Zipf law parameter	3
Total requests rate to CS, λ	100 /second
Attacker total requests rate, λ_T	150 /second
Av. service time of L1 CSes, $1/\mu_1$	3.3 ms
Av. service time of L2 CSes, $1/\mu_2$	3.3 ms
Av. service time of origin server, $1/\mu_o$	2.5 ms
Latency between L1 and L2, T_1	80 ms
Latency between L2 and origin server, T_2	40 ms

Service rate is often considered a measure of a server's performance, and it also affects the cost of a CDN. In order to save costs, many CDN provider do not set up very large performance redundancies, but according to our research, it can have a positive impact when faced with the optimal attack strategy. However, after our analysis in GA result, we found the reason of this result.

We evaluate the potential threat of the optimum attack strategy against CDN by P, the increase ratio of average response time (IRAR). Figure 2 shows the IRAR under different service rate. Remarkably, as the service rate increased, the IRAR also increased. This seems very unreasonable, because in general, the performance of the server can be better defense against attacks.

Due to the very fast service rate, the response time W of each server was very small. If there was no attacks, this can significantly reduce the average response time. However, in the face of the attack, due to the low cache hit ratio, many requests had to be sent to the origin server, which will spend time on latency T_1 and T_2. We find that as the service rate increased, the average response time without attack decreased significantly, while the average response time under attack decreased limitedly, so the IRAR increased.

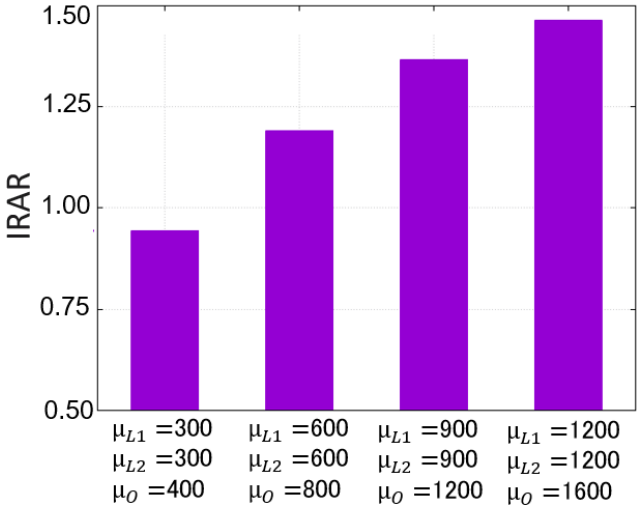

Fig. 2: Effect of service rate

While increasing service rates generally improves CDN performance, it is important to be aware of the buckets effect in the face of attacks. Especially in time-sensitive CDN, a significant change in response time when attacked can lead to a bad experience.

5. Conclusion

This paper focused on a multi-layer CDN model and used the M/M/1 queuing model to derive average response time for content requests. We concluded that understanding the attacker optimal strategy through GA can help CDN providers identify weak points and enhance their defense mechanisms.

Acknowledgements:

This work was supported by JSPS KAKENHI Grant Number 23K21664 and 23K21665.

References

[1] M. Ghaznavi, E. Jalalpour, M. A. Salahuddin, R. Boutaba, D. Migault and S. Preda, "Content Delivery Network Security: A Survey," in IEEE Communications Surveys & Tutorials, vol. 23, no. 4, pp. 2166-2190, Fourthquarter 2021

[2] J. Liu and N. Kamiyama, "Investigating Impact of DDoS Attack and CPA Targeting CDN Caches," 2024 IEEE/IFIP International Wworkshop on Analytics for Nerwork and Service Management (AnNet), Seoul, Korea, May 2024

[3] H. Che, et al., "Hierarchical Web Caching Systems: Modeling, Design and Experimental Results," IEEE J. Selected Areas of Commun., vol.20, no.7, Sep. 2002