Optimum Content Pollution Attack Using Genetic Algorithm in CDN

Liu Jiaqi[1] Noriaki Kamiyama[2]
Graduate Shool of Information Science and Engineering, Ritsumeikan University[1]
College of Information Science and Engineering, Ritsumeikan University[2]

1. Introduction

As the use of Content Delivery Network (CDN) becomes increasingly widespread, they also face growing security threats. Compared to typical network attacks, CDN is particularly vulnerable to cache pollution attacks (CPA) because CDN heavily relies on caching mechanisms. To ensure the efficient operation of CDN and prevent destructive attacks, extensive research focuses on optimizing defense strategies. This paper approaches the issue from an cache pollution attacker's perspective, aiming to optimize attack strategies to investigate attack characteristics. We have analyzed the characteristics of attacks in previous studies [1].

In our another previous paper [2], we studied how to optimize the attack strategy in multi-layer CDN. In this paper, we improve the multi-layer CDN model and introduce the concept of region. We divide the target users of CDN into regions, and the requests from users will be directed by the DNS server to the edge servers, so that the model is closer to the real situation.

2. Analytical Model

For most CDN, user requests follow the Poisson arrivals. In this paper, we use the M/M/1 queue model to derive the average response time of contents to measure the CDN performance. Let M denote the number of contents provided by CDN, λ_i denote the Poisson arrival rate of request for content i, and $1/\mu$ denote the mean of the exponentially distributed service time of server. 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 server, there are two cases based on whether the requested content is cached or not. The origin server stores all the provided contents, and the other servers stores a part of contents. When the content requested by a user does not exist in the cache server (CS), which is called *cache miss*, the server 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 exists in the CS, which is called *cache hit*, the CS will update the cache storage 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 CSs adopt the LRU.

Network latency constitutes a significant portion of the average response time. However, since each user' s network environment varies, this paper focuses solely on the latency between servers. Basically, we define the network latency between the edge server and the origin server as T. Based on equation 1, we further define the average processing time of the edge server as W_e and that of the origin server as W_o. Thus, the average response time is W_e in the case of a cache hit, and $W_e + W_o + T$ in the case of a cache miss.

Since the cache hit ratio varies depending on the popularity of the content, we introduce h_i as the cache hit ratio for content i. Consequently, we can express the average response time for content i as W_i :

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

We use the Che approximation to predict the hit ratio h_i of each content i on the CS [3]. Let C represent the cache capacity of the edge server, meaning the maximum number of content it can store. According to the Che approximation, the cache hit ratio h_i can be expressed as:

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

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 consists of an origin server and multiple layers of CSs; this study focuses specifically on a two-layer CS model. We refer to the CS layer closer to the users as the edge servers and the layer closer to the origin server as the central servers. Typically, a region includes a central server with a high cache capacity, which connects to the origin server and multiple edge servers. The edge servers, with smaller cache capacities, connect only to the central server. When a user request is directed by the DNS server, it will be sent to an edge server. If the cache hit occurs, it is delivered directly to the user. If the cache miss occurs, the request is forwarded by the edge server to the region's central server. The central server generally handles the request by delivering cached content to the requesting edge server. However, if a cache miss occurs on the central server, it forwards the request to the origin server.

In this study, the CDN model consists of five regions—labeled A, B, C, D, and E—each configured with a different number of edge servers (5, 10, 15, 20, and 25, respectively) based on varying user scales. While each region operates independently, they share similar content popularity trends and are all connected to the same origin server. Fig.1 shows two regions in the model.

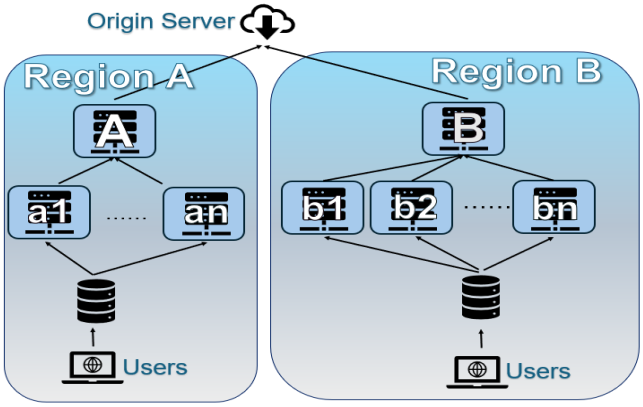

Fig. 1: Two regions of multi-layer CDN model

In this model, we assume that the DNS server uses a random algorithm to direct user requests to one of the edge servers within the region. Additionally, we define the network latency between the edge and central servers as T_1, and the latency between the central and origin servers as T_2.

Within this model, three different request scenarios can occur. Suppose a user in region A makes a request. When a cache hit occurs on the edge server, the average response time is denoted as r_e^A, where $r_e^A = W_{an}$, with W_{an} representing the average response time for any given edge server. If the cache hit occurs on the central server, the average response time is r_c^A. When neither the edge nor central server

has cached the content, the average response time is r_o^A. We can derive r_c^A and r_o^A using the following expressions:

$$r_c^A = W_{an} + W_A + T_1, \quad (5)$$
$$r_o^A = W_{an} + W_A + Wo + T_1 + T_2. \quad (6)$$

Let h_i^{an} represent the cache hit ratio of content i on the edge servers in region A, and h_i^A represent the cache hit ratio of content i on the central server in region A. We define R as the overall average response time, R^A as the weighted average response time for all users in region A, and R_i^A as the average response time for users in region A requesting content i. We can calculate R_i^A using the following expression:

$$R_i^A = h_i^{an} r_e^A + (1 - h_i^{an}) h_i^A r_c^A + (1 - h_i^{an})(1 - h_i^A) r_o^A. \quad (7)$$

Furthermore, based on the request ratio for different content by users, we can derive R^A:

$$R_A = \frac{\lambda_i^A R_i^A}{\sum_{i=1}^{M} \lambda_i^A}. \quad (8)$$

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 CSs 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.

Given that average response times vary widely across different scenarios and are generally small in value, even slight changes in response time can significantly impact user experience. Therefore, we define the average response time growth rate (GR) as an evaluation metric.

In this study, we assume the attacker targets only a single region A, which have 5 edge servers. We assume that R_A^n represents the average response time in region A when there is no attack, and R_A denotes the average response time during an attack. When attacking region A, we can calculate the GR using the following expression:

$$GR = \frac{R_A - R_A^n}{R_A^n}. \quad (10)$$

Tab. 1: Simulation parameter setting in basic case

Paramater	Value
Number of contents provided, M	100
Edge servers' cache size	20
Central servers' cache size	50
Zipf law parameter	4
Edge server number in target area	5
Requests rate to each edge server, λ	30 /second
Av. service time of Edge servers, $1/\mu_1$	10 ms
Av. service time of Central servers, $1/\mu_2$	2 ms
Av. service time of origin server, $1/\mu_o$	1 ms
Latency, T_1	500 ms
Latency, T_2	300 ms

We vary the number of edge servers in the attacked region to observe changes in GR under different attack capacities. According to Fig.2, regions with fewer edge servers are more vulnerable to CPA of the same capacity. Moreover, GR increases at the fastest rate in these regions, resulting in a higher GR when the attack capacity is high.

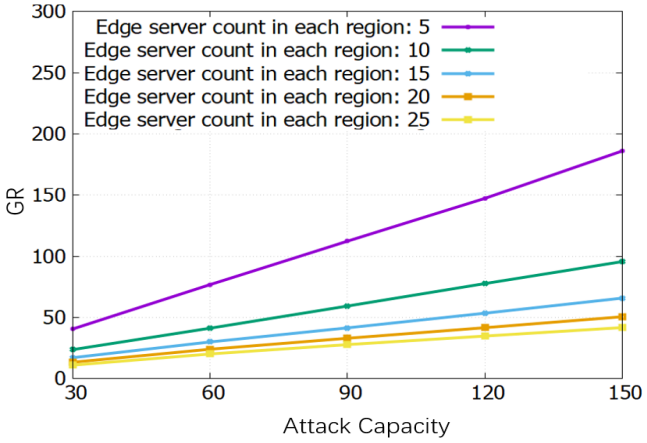

Fig. 2: Growth rate of average response time in various edge server counts in each region

The graph reveals that, under the same attack capacity, regions with more regular users experience a more pronounced dilution of the attack impact. However, it is noteworthy that even when the attack capacity is very small relative to the request rate of regular users, a certain degree of GR is still observed.

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, 23K21665, and 23K28078.

References

[1] 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
[2] Jiaqi, Liu, and Noriaki Kamiyama. "Optimum Content Pollution Attack Using Genetic Algorithm in Multi-Layer CDN." IEICE Society Conference Sep. 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