



Automated Server Operation Model to Improve the Efficiency of Domain Services

Seong-Ik Kim, Dept. of Computer Engineering, Kongju National University, 31080, Rep. of Korea

***Koo-Rack Park**, Dept. of Computer Science & Engineering, Kongju National University, 31080, Rep. of Korea, ecgrpark@kongju.ac.kr

Dong-Hyun Kim, Dept. of Energy IT Engineering, Far East University, 27601, Rep. of Korea

Han-Jin Cho, Dept. of Energy IT Engineering, Far East University, 27601, Rep. of Korea

*Corresponding author

Abstract. While using the internet services, users tend to frequently experience the inconvenience in service due to buffering caused by overloads in the domain service server. If there are some margins for availabilities for the service server, the above issue does not occur, so it is possible to remove the inconvenience. In each time period of the domain that provides the service, the peak traffic data is collected to configure the data set, and the peak traffic data in each time period within the configured data set is summed up. After that, in this newly proposed model, the cumulative hourly peak traffic data is checked in terms of whether it exceeds the availability threshold set by an operator, and if it does, it is transferred to other pool group (pool N). The test results for the proposed model indicates that in the past, it normally took 2 hours to analyze the domain traffic data and 10 to 20 minutes to apply the analysis results, but in the newly proposed model, it takes only less than 5 minutes to analyze and apply results. System load frequency is reduced from 13 times a month to 1 time per month, and also the number of times to use the reserved server is reduced from 18 times to 2 times. The CPU margin rate is increased from 27% to 71%, which indicates that the usage rate is decreased. Overall, the operator's resource is reduced, and thus it is possible to achieve server stability as well.

Keywords: Domain service; Domain traffic; Reducing latency; GSLB; Data center

Received: 09.12.2020

Accepted: 13.01.2021

Published: 06.02.2021

INTRODUCTION

Due to the recent, rapid increases in the number of cloud based services and the number of internet users, there are increasingly more companies that host services through their data centers, thus causing a huge increase in the traffic volume [1]. The cloud computing environment or the internet data center consists of many servers that process internet users' requests [2]. In addition, as the amount of data is increased due to the increasing number of social network users, service providers are faced with challenges to learn how to deploy user data on many servers [3]. Everyone must have probably experienced the inconvenience of service-use as the internet service tends to display images too long or cause buffering during the streaming play, thus failing to provide normal service. The causes can be classified into impacts of the user access environment and impacts of the service server. When using services on the internet, as there are increasingly more users accessing it, there are frequently lots of inconveniences due to server overloads such as discontinued streaming or delayed downloading. The number of servers is directly proportional to the traffic availability. Thus, if there are already enough servers in place, then users will not experience any inconveniences in service-use. In general, due to economical reasons, such as server implementation costs, most of internet service providers have been implementing and operating a single server system, and in addition, instead of direction operation, they tend to provide services through those companies specialized in server service. In the single server system, there are lots of difficulties in fulfilling the service requirements for many users. In addition, when multiple domain services are provided from a single server group, if one domain fails, then the rest of domains that the server group provides services for will also fail. Therefore, in order to minimize the impacts, generally, one server group provides service for one domain. For stable service, server operators add servers to distribute loads in case than domain traffic is increased. From the standpoints of operators, it is a very difficult task to increase or decrease the number of servers by checking traffic at each domain every time when there is an issue. In addition, although it is possible to prepare reserved servers, but from the standpoints of companies, the limited quantity and the costs are real challenges. In this paper, the author aims to

propose an automated server operation model to improve the efficiency of domain services in a limited number of servers since companies can prepare reserved servers in case of an increase in domain traffic, but they are faced with problems of limited quantity and costs.

1. Related Works

1.1. Domain Name System

The DNS (Domain Name System) is a critical part used in implementing the internet infrastructure. It is used to transform the domain name to an IP address [4]. If there exists no DNS, then it is impossible for many services and apps to map domain names to IP addresses, thus operation will fail. So, it is necessary to distribute DNS data and store them repeatedly in multiple locations [5]. In addition, servers located in other networks connected to the internet can improve availability of DNS data in case that there are some partial problems in the network [6]. Before users check the web site that they try to access, attackers sometimes use malware scripts or other malicious contents to link or redirect them to other malicious web sites [7]. Every internet that is currently in use includes various DNS interactions, and especially, efficient, expandable and safe DNS is a key factor for overall internet use and it is also a very efficient distributed system that offers redundancy and caching used in the DNS server. Besides, every domain name is comprised of a layer structure, and the DNS tree structure is used to execute and search for [8].

1.2. Reducing Latency

The CDN (Content Delivery Network) is designed to provide high performance and availability so that spatial services can be distributed to users, and short delay and fast response time can provide users with better experiences [9]. In general, the GSLB (Global Server Load Balance) is used to connect internet users to the closet CDN node, but traditionally, CDN node in most cases is located in the local network. So, mobile users can use the mobile core network [10] to access the video service and so on, but due to increased traffics, it is causing lots of burdens to the core network. To resolve this issue, the MEC (Multi-access Edge Computing) technology is used [11], a CDN node is located inside the mobile network but not through the core network, so it can reduce the standby time for end to end transmission, and also help reducing the bandwidth of core network [12].

2. Proposed System

From one pool group, multiple domain services are provided. If there is an increase in the traffic of a pool group, reserved servers are added. For efficient use of pool groups and reduced use of reserved servers, it is necessary to assess the traffic pattern of each domain and optimize the use of pool groups. First of all, assess the traffic pattern of a domain. Traffic patterns are classified into consistent one and inconsistent one on a daily basis. Inconsistent one is classified as an event domain. In case of consistent domains, check the cumulative data on an hourly basis. Set the server pool's availability to threshold, and transfer most domains at the closest time to another pool group. Repeat this process.

2.1. System Configuration

Following Fig. 1 illustrates a system diagram proposed in this paper. The GSLB (Global Server Load Balance) server that provides the DNS (Domain Name Service) manages the server IP address for the domain service in a pool group. Different domain groups are provided as services for each pool group. The domain traffic data that is provided as service from each server is transmitted to the data collection server. The web server re-estimates the domain's pool group by performing the service server pool assignment process on the domain traffic data collected. The re-estimated data is sent to the GSLB server to change the pool group of each domain.

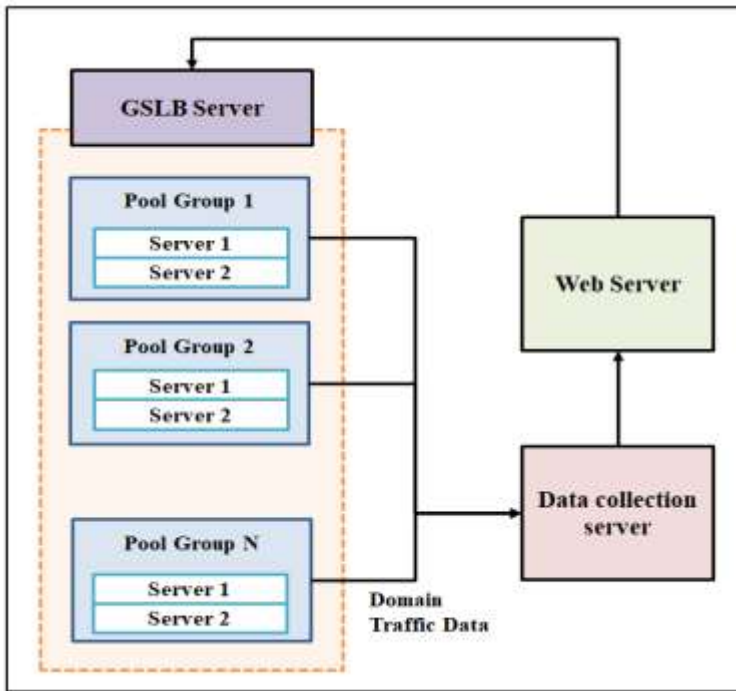


Fig. 1. System Configuration Diagrams

2.2. Service Server Pool Assignment Process

Following Fig. 2 illustrates a process to designate a service server pool. First, collect the hourly peak traffic data for each domain that provides service to construct the data set. Second, within the constructed data set, sum up the hourly peak traffic data. Third, check if the hourly cumulative peak traffic data exceeds the availability threshold set by an operator. Fourth, if the condition in the third step is met, then extract a domain with the maximum, and transfer it to other pool group (pool N). If there are some domains remaining after repeating the above steps, then it can be determined that the domain service is stable within the availability threshold, and repeat the above steps to other pool groups to acquire the availability stability for the domain of each pool group.

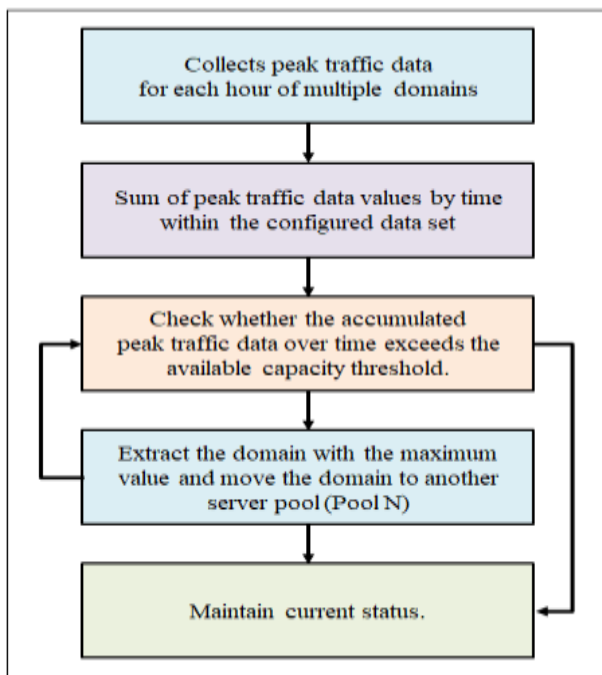


Fig. 2. Service Server Pool Assignment Process

Following Fig. 3 illustrates the main algorithm for the proposed model to re-estimate a pool group. Hourly domain peak traffic data collected from each server and saved in the DB is summed up on an hourly basis. Cumulative data is compared to the availability threshold set by an operation by considering the system stability. If the cumulative hourly data is greater than the availability threshold, then transfer the domain with the maximum traffic to other pool group. Repeat the above steps on the remaining domains after the above domain is transferred.

```

for ( time count ) {
    for ( domain count ) {
        traffic data = read DB data;
        time sum data += traffic data;
    }
}

long Available capacity = 300;

for ( time count ) {
    while ( Available capacity < time sum data ) {
        for ( domain count ) {
            if ( max data domain check ) {
                move another pool group;
            }
        }
    }
}

```

Fig. 3. Pool Group Re-estimation Algorithm

2.3. Database Configuration

Following [Table 1] illustrates the structure of the table where domain's hourly traffic data is managed. 'pool_group' is the name field of pool group that is managed, 'server_ip' is the IP address field for the server that belongs to the pool group and provides domain service. 'domain_name' is the name field for the domain where service is provided in the pool group. 'time' is expressed in 24 hour formats, and it is the time information field that represents the duration of traffic data service on a day. 'traffic_data' is the field where the hourly domain peak traffic data collected from the server is saved. Data in the table refers to the domain traffic data provided as service from each server and sent to the central collection server, so that it can be saved in the DB. The data is the foundation for the service server pool assignment process. The table shows the domain traffic data to be analyzed.

Table 1. DB Table Information

| Field | Type | Description |
|--------------|----------------|-----------------|
| pool_group | VARCHAR2 (10) | Pool group Name |
| server_ip | VARCHAR2 (10) | Server IP |
| domain_name | VARCHAR2 (100) | Domain Name |
| time | DATETIME | Time (24H) |
| traffic_data | FLOAT (10) | Traffic data |

3. Result discussions

Following Fig. 4 illustrates the main screen of the model proposed in this paper.

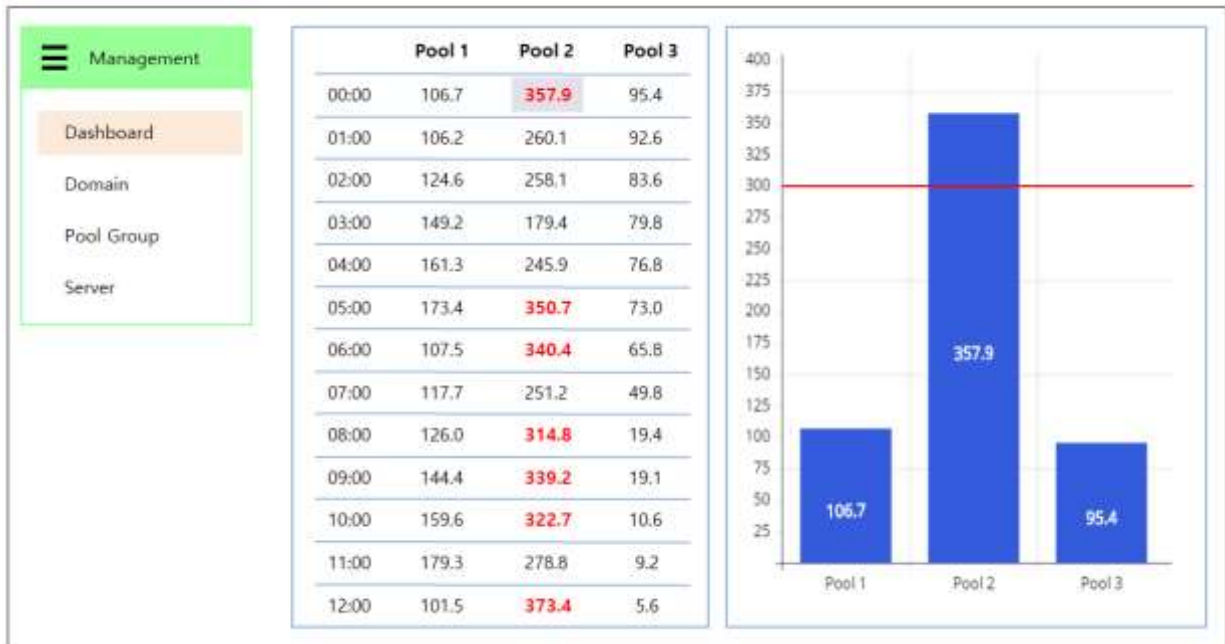


Fig. 4. Proposed Model Main Screen

It is composed of the menu tree on the left and the body on the right. The menu tree is composed of the dashboard that is used to monitor entire operation, and the menus that manage each of domain, pool group and server. The dashboard screen shows the hourly pool group traffic data in the table. In the table, the data exceeding the availability threshold is marked in red, and the pool graph data for the hour with the selected data is illustrated in a graph. The graph shows the cumulative pool group traffic data for a specific hour and the availability threshold.

Following Fig. 5 illustrates the screen to select a pool group that requires re-estimation of a domain. Once the pool group is selected, click the get button on the following screen in Fig. 6, and view the list of domains where services are provided in the pool group. Before selecting a pool group, for efficient distribution, one must first check a group with lots of traffic and a group with less traffic.

Select a Pool Group

Group 1
 Group 2
 Group 3
 Group 4
 Group 5
 Group 6

Fig. 5. Selecting a Pool Group



Fig. 6. Getting the Domain List

In the following screen of Fig. 7, click the Read button to retrieve the traffic data from the domain list. The following screen in Fig. 8 shows the table containing the hourly peak traffic data for every domain in the domain list. Hourly traffic peak data for a domain is necessary for executing the service server pool assignment process.



Fig. 7. Reading Domain Traffic Data

| Time | A | B | C | D | E | F | G | H | I | J |
|-------|------|------|------|------|------|------|------|------|------|------|
| 00:00 | 99.4 | 1.8 | 5.5 | 64.3 | 3.9 | 99.7 | 9.8 | 96 | 84.2 | 95.4 |
| 01:00 | 88 | 11.3 | 6.9 | 35.2 | 8.1 | 78.5 | 20.8 | 81 | 36.5 | 92.6 |
| 02:00 | 75 | 24.3 | 25.3 | 45.6 | 18.8 | 58.8 | 40.9 | 66.4 | 27.6 | 83.6 |
| 03:00 | 66.9 | 45.1 | 37.2 | 4.5 | 27.2 | 31.7 | 65 | 39.1 | 11.9 | 79.8 |
| 04:00 | 62.3 | 52 | 47 | 15.7 | 39.9 | 18.1 | 83.2 | 26 | 63 | 76.8 |
| 05:00 | 53.5 | 68.1 | 51.8 | 95.7 | 83.8 | 15.5 | 93.8 | 10.4 | 91.5 | 73 |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| 20:00 | 78.2 | 40.6 | 28.6 | 89.6 | 27.6 | 52.6 | 53.1 | 50.3 | 92.2 | 74.9 |
| 21:00 | 85.1 | 50.6 | 13.3 | 40.3 | 22.6 | 20.8 | 79.9 | 28.6 | 86.5 | 75 |
| 22:00 | 91.7 | 64.9 | 3 | 39.4 | 9.4 | 15.7 | 93.3 | 12.7 | 33.8 | 81.6 |
| 23:00 | 98.7 | 92.7 | 0.8 | 9.5 | 6.1 | 4.3 | 99.8 | 2.3 | 93.8 | 82.9 |

Fig. 8. Reading Domain Traffic Data Results

Click the Action button in the following screen of Fig. 9 to check the data obtained by re-estimating the domain of each pool group by using the service server pool assignment process. Newly estimated pool group’s domain is reflected in the GSLB to change the pool group that provides the domain service.



Fig. 9. Reassign Domain Result

The following Table 2 compares the model proposed in this paper before and after application, in terms of systems and operation time. The test results for the proposed model indicates that in the past, it normally took 2 hours to analyze the domain traffic data and 10 to 20 minutes to apply the analysis results, but in the newly proposed model, it takes only less than 5 minutes to analyze and apply results. System load frequency is reduced from 13 times a month to 1 time per month, and also the number of times to use the reserved server is reduced from 18 times to 2 times. The CPU margin rate is increased from 27% to 71%, which indicates that the usage rate is decreased. Overall, the operator’s resource is reduced, and thus it is possible to achieve server stability as well.

Table 2. Analysis and Setup Time Comparison

| Division | Comparison Target | AS-IS | TO-BE |
|-----------|-------------------------------|-----------------|------------------|
| Operation | Analysis time | 2 hours | Within 5 minutes |
| | Setting time | 10 ~ 30 minutes | |
| System | System load frequency (month) | 13 | 1 |
| | Redundant server usage count | 18 | 2 |
| | CPU Idle (%) | 27% | 71% |

4. Conclusions

Everyone must have probably experienced the inconvenience of service-use as the internet service tends to display images too long or cause buffering during the streaming play, thus failing to provide normal service. The causes can be classified into impacts of the user access environment and impacts of the service server. Thus, if there are already enough servers in place, then users will not experience any inconveniences in service-use. In general, due to economical reasons, such as server implementation costs, most of internet service providers have been implementing and operating a single server system, and in addition, instead of direction operation, they tend to provide services through those companies specialized in server service. This pool group is obtained by grouping a limited number of servers providing multiple domain services. Consistently checking the varying domain traffic when processing traffic data requires lots of resources from the standpoints of operators. In this method, lots of servers are deployed in a pool group, thus successfully improving the traffic availability, but the cost issues cannot be

overlooked. In this paper, the author proposes a system to change the pool group that processes the domain traffic by checking the hourly traffic of the domain that uses the pool group. The newly proposed model can be used by an operator to reduce unnecessary resources and acquire stable traffic availability, thus improving the overall efficiency of domain service.

In the future, the author suggests that efforts should be made to study how to set the traffic data's availability threshold through a domain traffic analysis and also how to reduce the automated operation time.

References

- [1] Balanici, M., & Pachnicke, S. (2019) Server traffic prediction using machine learning for optical circuit switching scheduling. In Photonic Networks; 20th ITG-Symposium, VDE, 1-3.
- [2] Tadakamalla, V., & Menascé, D. A. (2018) Model-driven elasticity control for multi-server queues under traffic surges in cloud environments. In 2018 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 157-162.
- [3] Zhou, J., Fan, J., Wang, J., Cheng, B., & Jia, J. (2017). Towards traffic minimization for data placement in online social networks. *Concurrency and Computation: Practice and Experience*, 29(6), e3869.
- [4] Messabi, K. A., Aldwairi, M., Yousif, A. A., Thoban, A., & Belqasmi, F. (2018) Malware detection using dns records and domain name features. In Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, 1-7.
- [5] Kröhnke, L., Jansen, J., & Vranken, H. (2018) Resilience of the Domain Name System: A case study of the. nl-domain. *Computer Networks* 139, 136-150.
- [6] Caesar, M., & Rexford, J. (2005). BGP routing policies in ISP networks. *IEEE network* 19 (6), 5-11.
- [7] R. Masri and M. Aldwairi. (2017) Automated malicious advertisement detection using VirusTotal, URLVoid, and TrendMicro. In 2017 8th International Conference on Information and Communication Systems (ICICS), 336-341.
- [8] Jiang, J., Zhang, J., Duan, H., Li, K., & Liu, W. (2018) Analysis and measurement of zone dependency in the domain name system. In 2018 IEEE International Conference on Communications (ICC), IEEE, 1-7.
- [9] Y. Sani, A. Mauthe, and C. Edwards, (2017) Adaptive Bitrate Selection: A Survey. *IEEE Communications Surveys & Tutorials* 19(4), 2985-3014.
- [10] Unicom, C. (2018) White Paper for China Unicom's Edge-Cloud Service Platform Architecture and Industrial Eco-system. China Unicom Network Technology Research Institute.
- [11] Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017) On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials* 19(3), 1657-1681.
- [12] Han, S., Ma, H., Chen, D., Wang, Y., Wu, Y., & Zhang, P. (2018) Streaming Video Optimization in Mobile Communications. In 2018 IEEE/CIC International Conference on Communications in China (ICCC), IEEE, 495-499.