Optimization scheme based on web front-end load balancing

Gao Yuan[a], Huang Mengxing[b,*]

School of Information Science and Technology, Hainan University, Haikou, China

[a]gaoyuan001@163.com, [b]2369654285@qq.com

*Corresponding author

Keywords: Load balancing; cluster; web front-end

Abstract: Load balancing technology is an application of cluster technology. The load balancing technology is a scheduling method and strategy built on the network structure, which not only effectively expands the server bandwidth and increases the throughput, but also enhances the network data processing capability. Traditional load balancing network service system consists of several servers in a symmetrical way. Each load balancing back-end server has the same status. It can provide services independently without the assistance of other servers. In this case, there are sometimes some unusual service requests, and the symmetrical server collection facing external threats may cause all servers to crash. In this regard, we propose another load balancing scheme, which establishes a super server in multiple servers to handle receiving abnormal data requests. This paper proposes an optimization algorithm based on Web front-end load balancing. Through the initial simulation of the algorithm, the performance of the cluster server can be improved.

1. Introduction

With the development of science and technology, the progress of society, and the increase in the number of Internet users, Web services have become an indispensable part of the lives of Internet users. As a result, popular websites or services have a large number of visits. Therefore, how to make a large number of services run normally has become a hot topic. In the early days of the Internet, the business traffic is relatively small and the business logic is relatively simple, so a single server can meet the basic needs. But with the development of the Internet, the traffic is getting larger and larger and the business logic is becoming more and more complex. The performance problems and single point problems of a single machine are highlighted, so it needs more than one machine. The device performs horizontal expansion of the performance and avoids single point failures.

The early strategy was to use DNS as a load, by resolving different IP addresses to the client, so that the client's traffic directly reached the various back-end servers. However, this method has a big disadvantage that is the delay problem. After making the change of scheduling strategy, because the caching of DNS nodes at all levels will not take effect on the client in time, and the scheduling strategy of DNS load is relatively simple, which can not meet the business needs, so there is load balancing. External requests first arrive at the load balancing server. Load Balancing Servers distribute external requests to different load balancing servers through the built-in scheduling algorithm. The point is removed from the application server cluster to ensure that the load balancing background server is efficiently available. Common load balancing schemes include HTTP redirection to achieve load balancing, DNS load balancing, reverse proxy load balancing, IP load balancing, direct routing load balancing, IP tunnel load balancing. The client request server is shown in Fig 1.

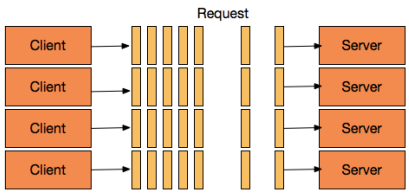

Fig 1. Client request server

Initially, a single server responds to external traffic requests. As traffic grows, when the performance of a server reaches its limit, we can use server clusters to improve the overall performance of Web sites. Then, in the server cluster, an intermediate server is needed to act as the dispatcher. All requests from outside will be received by it first. The dispatcher then assigns the requests to a background server according to the load of each server. So in this process, how to allocate tasks reasonably and ensure that all the back-end servers give full play to their performance, so as to maintain the optimal overall performance of the server cluster, this is the load balancing problem. For the load balancing technology, the load of each server can be friendly, but for a large amount of temporary data access, especially when there are aggressive or regular IP segments of abnormal access data, the load balancing server will have a large number of Data access is divided into each back-end server, which may cause all servers to crash, and the service is terminated. Therefore, load balancing has certain limitations. The load balancing server works as shown in Fig 2.

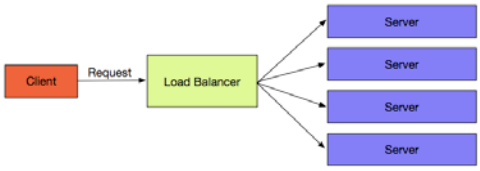

Fig 2. Load balancing server

2. Load balancing technology solutions

2.1 Load balancing algorithm

(1) Stochastic algorithm

Random randomly set random probability according to weight. The probability of collision on a cross section is high, but the larger the amount of calls, the more uniform the distribution, and the more uniform the weight is used according to the probability, which is conducive to dynamically adjusting the weight of providers.

(2) Polling and weighted polling

Round Robbin is the best algorithm to use when the processing capacity of each server in the server cluster is the same and the processing capacity of each transaction is not different. The round robin ratio is set according to the weight after the Convention. There is a problem of slow providers accumulating requests. For example, the second machine is slow but not hanging. When the request is transferred to the second machine, it is stuck there. Over time, all requests are stuck on the second machine.

Weighted Round Robbin algorithm adds a certain weight to each server in polling. For example, server 1 for weight 1, server 2 for weight 2, server 3 for weight 3, the order is 1-2-2-3-3-3-1-2-2-3-3-3...

In the actual operation process, the performance of each node in the cluster is constantly changing, and the weight is also changing dynamically, so the performance ratio of each node in the cluster is also changing. Assuming that W (T_i) represents the weight of the number of I, and i ∈ {1 ,..., n} server nodes, then the weight ratio of each node in the cluster is:

$$E_W(T_i) = W(T_i) / W(T) \quad (1)$$

Among them, Ew (Ti) denotes the weight ratio of the first node, i ∈ {1 ,..., n }.

(3) Minimum connection and weighted minimum connection (WLC)

Least Connections is an algorithm for communicating with servers with the least number of connections in multiple servers. Even if the processing capacity of each server is different and the processing capacity of each transaction is different, the load of the server can be reduced to a certain extent.

WLC achieves load balancing based on the ratio of the number of connections to the weight of the current node, assigns requests to the node with the smallest ratio, and adds 1 to the number of connections.

$$P_i = min\left(\frac{C(T_i)}{W(T_i)}\right) \quad (2)$$

And $C(T_i)$ represents the total number of links of the node, $W(T_i)$ represents the weight value of the current node, and P_i represents the server node that responds to the request.

Weighted Least Connection is an additional weight algorithm for each server in the least connection algorithm, which allocates the number of processing connections to each server in advance and transfers client requests to the server with the least number of connections.

(4) Hashi algorithm

Consistent hash consistency Hash, the same parameter requests are always sent to the same provider. When a provider hangs up, requests originally sent to that provider are spread out to other providers based on virtual nodes without causing drastic changes.

(5) IP address hashing

By managing the hash of sender IP and destination IP addresses, an algorithm for forwarding packets (or packets sent to the same destination) from the same sender to the same server uniformly. When the client has a series of business to deal with and has to communicate with a server repeatedly, the algorithm can take the flow as a unit to ensure that the communication from the same client can always be processed in the same server.

(6) URL hash

An algorithm for forwarding requests sent to the same URL to the same server by managing the hash of client request URL information.

2.2 Main technology of load balancing technology

(1) HTTP redirection to achieve load balancing

When the client requests to the server, the request is intercepted by the load balancing dispatcher first. The dispatcher chooses the load balancing back-end server according to some allocation algorithm and strategy, and encapsulates the IP address of the selected server in the Location field of the HTTP response message header, and sets the status code of the response message to 302, the most important is that the IP address of the selected server is encapsulated in the Location field of the HTTP response message head The response message is then returned to the browser. When the browser receives the response message, it parses the Location field and makes a request to the URL, then the specified server processes the user's request, and finally returns the result to the user.

In the process of using HTTP redirection to achieve load balancing in server cluster, a server is required as a request dispatcher. A user's operation needs to initiate two HTTP requests, send a request to the scheduling server once, get the IP of the back-end server, and send a request to the back-end server the second time to get the processing results. The HTTP redirection works as shown in Fig 3.

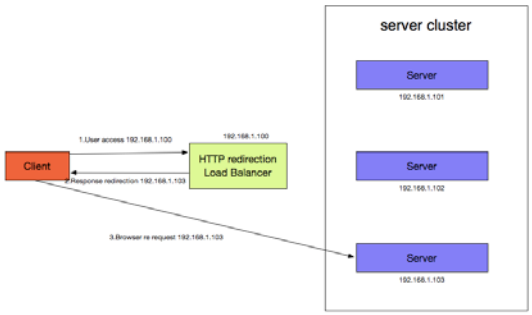

Fig 3. HTTP redirection

(2) DNS load balancing

DNS load balancing is realized by DNS server, which is mainly used to distribute external requests evenly on Nginx server. In fact, it may be a request for geographical distinction. However, the requests in a geographical area need to be evenly distributed to the nginx server.

DNS load balancing is realized by DNS server, which is mainly used to distribute external requests evenly on Nginx server. In fact, it may be used to differentiate requests according to region. But requests in a region need to be evenly distributed to the nginx server.

Load balancer collects the performance of each server, calculates the average performance of the cluster, obtains the benchmark CPU performance value Pa, memory size Ma, network bandwidth value Ba, designs the performance vector of the server $[\mathrm{pi}, \mathrm{mi}, \mathrm{bi}]$, where $q_i = \mathrm{Qi}/\mathrm{Qa}$, $m_i = \mathrm{Mi}/\mathrm{Ma}$, $t_i = T_i/T_a$, i = 1, 2. 3,..., N, Qi, Mi and Ti represent the CPU main frequency, memory size and network bandwidth of the first server, respectively. N is the number of cluster physical servers. Calculate the overall performance of the server according to the performance vector of the server:

$$S_i = k1 \times q_i + k2 \times m_i + k3 \times t_i \; i = 1,2,3,\ldots,N \quad (3)$$

And, $K1 + K2 + K3 = 1$, k1、k2、K3 represent the weight of each index.

First of all, we need to point our domain name to multiple back-end servers, and then set up a scheduling algorithm, then our preparatory work is completed, and the next load balancing is completely completed by the DNS server. When a user makes a request to our server, the DNS server will automatically select an appropriate IP according to our pre-set scheduling algorithm and send the request to the user.

(3) Reverse proxy load balancing

The reverse proxy server is a server located before the actual server. All requests sent to our website must first go through the reverse proxy server. The server either directly returns the result to the user according to the user's request, or handles the request to the back-end server, and then returns it to the user.

The common proxy method is to proxy the connection requests of the internal network users to access the servers on the internet. The client must specify the proxy server and send the connection requests which should be sent directly to the servers on the Internet to the proxy server for processing. Reverse Proxy means that the proxy server receives the connection request on the internet, then forwards the request to the server on the internal network, and returns the result from the server to the client requesting the connection on the internet. At this time, the proxy server acts as a server outside. Reverse proxy load balancing technology is to dynamically forward connection requests from the Internet to multiple servers on the internal network in the form of reverse proxy for processing, so as to achieve the purpose of load balancing. The reverse proxy works as shown in Fig 4.

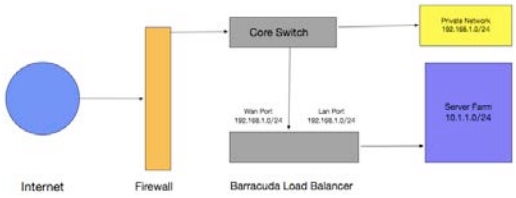

Fig 4. Reverse proxy

(4) IP load balancing

Lvs cluster adopts IP load balancing technology. It belongs to IP layer switching with good throughput. The scheduler analyses the IP header information from client to server, transfers requests to different servers to execute equally, and automatically shields the server from failures, thus making a group of servers into a high performance and high availability virtual server.

3. Load balancing scheme based on super server

Because of the uncertainty of external access, especially some large-scale access, malicious access or attack, there is a potential threat to the server. Load balancing technology can make all working servers load as average as possible. Its potential threat is that when faced with abnormal service requests, it can easily lead to partial or total server paralysis. For traditional servers, most of the exceptional access is to refuse IP access to a certain segment or region, or to handle it normally. The performance of websites is generally measured by response latency. The shorter the average response latency, the better the performance of servers. Assuming that the average response time of the first node is T (N_i), the average response time of each server node of the cluster is as follows:

$$T_j = [T(N_1), \ldots, T(N_j), \ldots, T(N_n)]^T \quad (4)$$

The average response latency of the whole cluster during the period T is:

$$\overline{T_n} = \frac{1}{n}\sum_{i=1}^{n} T(N_j) \quad (5)$$

$T(N_j)$ represents the average response time for the number of server node j, and $\overline{T_n}$ represents the average response time of the whole cluster.

The optimization strategy proposed in this scheme is to set up a well-configured server device in the load balancing background server, which is called super server in this paper. When the load balancing backend server is more than half loaded, all suspected external request access is handled by the super server in the form of request queue, or the long-time request service is handled by the super server. The response of client request server is as shown in Fig 5.

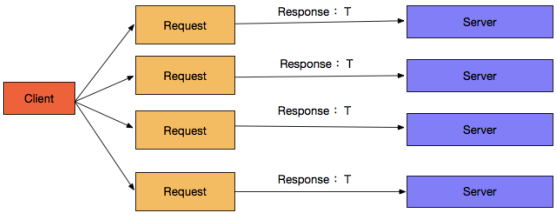

Fig 5. The response of client request server

Based on load balancing cluster model, the result is simplified. Based on the problem solved, the load balancing server is divided into N problem modules, and then allocated to N servers. Ideally, the size of each problem sub-module is the same, and the processing capacity of each server is the same. The ideal processing time of the load balancing server is the processing time of the sub-

module of the single load balancing server. Assuming that each load balancing server has the same processing capacity, but the request service sent is too large, it may lead to overload or paralysis of the load balancing server, which reduces the efficiency of load balancing. The response of big request is as shown in Fig 6.

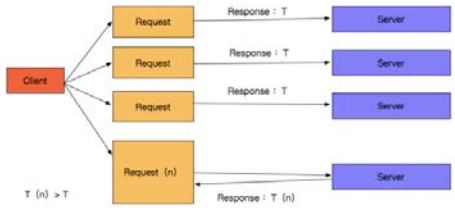

Fig 6. The response of big request

Based on the practical problems of parallel computing technology, the optimization scheme proposed in this paper is to add one or more advanced processors to the parallel computing model. The computing power of this processor is obviously higher than that of other processors. The model of this scheme is shown as follows. In the process of parallel computing, if the master computer detects that the layout of a computer is too large, the master computer gives the task of the computer to the advanced computer to continue processing. In parallel computing, if there is a problem with a single computer, the master computer will hand over the task of the computer to the advanced computer. In the process of parallel computing, the calculation method over a certain period of time is marked and handed over to the advanced computer. The calculation method after the time node is calculated by the advanced computer, which reduces the number of repeated calculations and improves the efficiency of parallel computing. The response of big request and super server is as shown in Fig 7.

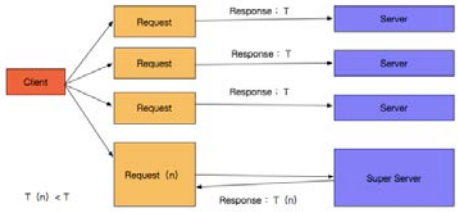

Fig 7. The response of big request and super server

The steps of the optimization model are as follows:

1) The services that are bisection requests are Q (0), Q (1), ..., Q (s), ..., Q (n-1).

2) The task is assigned to each load balancing background server, W (0), W (1), ..., W (s),..., W (n-1).

3) Load balancing server starts execution (Q)

4) The load balancing server monitors each load balancing back-end server. If a large Q (S) is found, the service request is forwarded to the super server W (n), and the service request is rejected if it cannot be processed.

5) Load balancing server monitors each load balancing back-end server. If a load balancing back-end server has abnormal response time or requests it serves, it sends the request service to the super server for processing.

6) Repeat steps 3, 4

4. Summary

In Web service system, the purpose of load balancing technology is to assign tasks according to the performance and load of each processor in Web system. The processing capacity and current load

of single load balancing background server are the main factors affecting the load change of server. Load volume is a dynamic value. To determine its parameters, a dynamic scheduling strategy is needed. The core of load balancing strategy is load balancing algorithm. How to optimize load balancing algorithm is the key to improve the efficiency of load balancing technology. For example, the algorithm complexity should be minimized when designing the algorithm. For load balancing technology, based on the original technology, this paper proposes a new solution to the problem, and improves the original technology model. This scheme is also suitable for most load balancing technologies, especially for parallel computing technology dealing with large data, through which more problems can be solved in the future.

References

[1] Collusion detector based on G-N algorithm for trust model[J]. Lin Zhang, Na Yin, Jingwen Liu, Ruchuan Wang. Journal of Systems Engineering and Electronics. 2016(04)

[2] Multi-DAGs Scheduling Integrating with Security and Availability in Cloud Environment[J]. LIU Yaqiu, SHAO Hongrun, JING Weipeng, QIU Zhaowen. Chinese Journal of Electronics. 2015(04)

[3] A Server Load Balancing Design for Peer-To-Peer Network[A]. Der-Cherng Liaw, Cheng-Hsiang Chiu, Chia-Wei Yeh,Chia-Ming Chang, Hsiao-Jen Hsieh. Proceedings of 2011 International Conference on Computers, Communications, Control and Automation Proceedings (CCCA 2011 V3) [C]. 2011

[4] A Periodic Dynamic Load Balancing Method[A]. Taotao Fan. Proceedings of 2016 3rd International Conference on Engineering Technology and Application[C]. 2016

[5] A Predictive Dynamic Load Balancing Algorithm with Service Differentiation[A]. Jiawei Jiang,Haojiang Deng,Xue Liu. Proceedings of 2013 15th IEEE International Conference on Communication Technology[C]. 2013

[6] RESEARCH AND IMPLEMENTATION OF LOAD BALANCING ALGORITHM FOR MASS MOBILE APPLICATION DETECTION TASK[A]. Yanhui Guo, Yingjie He, Xiaoming Liu. Proceedings of 2016 4th IEEE International Conference on Cloud Computing and Intelligence Systems(IEEE CCIS2016)[C]. 2016

[7] The Improvement and Implementation of the High Concurrency Web Server Based on Nginx[A]. Baiqi Wang, Jiayue Liu, Zhiyi Fang. Computing, Performance and Communication Systems (2016 Vol.1 Num.1)[C]. 2016

[8] A Webgis Load-balancing Algorithm Based on Collaborative Task Clustering[A]. Huang Ying, Guo Mingqiang, Luo Xiangang, Liu Yong Faculty of Information & Engineering, China University of Geosciences GIS software and application project research center of the educational department Wuhan, China. Environmental Science and Information Application Technology Volume 3[C]. 2009