Journal of Web Systems and Applications (2017) Vol. 1, Numuber 1
Clausius Scientific Press, Canada

Load Balancing Algorithm for Web Server Based on Weighted Minimal Connections

Pan Zhu[1, a], Jiangxing Zhang[2,b]

College of Information Engineering, Zhejiang University of Technology, Hangzhou, China

[a]578798852@qq.com, [b] zjx@zjut.edu.cn

Keywords: Web server cluster, Dynamic feedback, Load balancing, Weighted minimal connection number algorithm, OPNET.

Abstract. We have analyzed the clustering technology of Web server and studid the load balancing algorithm, due the lack of the existing load balancing algorithm can not accurately reflecting the size of the server load and manually setting the weight and other aspects, in this paper ,we propose an improved weighted minimum join Improved Load Balancing Algorithm for Web server cluster. On the one hand,The algorithm takes into account server utilization, memory utilization and network bandwidth mutilation and other factors, combined with web server's own characteristics,Accouding to the real-time load of the server to dynamically change the size of the weight, and on the other hand, assigns the new connection request according to the weighted minimum connection number algorithm. Through the use of simulation software OPNET, the algorithm not only reduces server latency, reduces HTTP response time,and can effectively improve the load balancing efficiency.

1. Introduction

In recent years, as the rapid development in Internet, the demand of the network service is also growing, which leads to the page loading is too slow and the website is not responding in the case of high concurrent data request.Thus It has become an urgent problem for website managers.When a single server load is too large, the usual method is upgrate the host. However, the ability of a single host is limited and high performance host is expensive. In addition, the server cluster is low-cost and has good scalability, which has become an effective method to solve this problem. A cluster consisted a group of computers that are independent of each other, interconnected by high-speed networks and managed in a single system mode[1]. Load-balancing system is the core part of server cluster[2], and the load-balancing algorithm is the key to deliver user's requests [3].

The existing load-balancing algorithms can be divided into two categories: static and dynamic. The static load-balancing algorithm assigns tasks with the fixed probability, which do not consider the state information of the server; Dynamic balancing-algorithm assigns tasks based on real-time load status information of the server. Dynamic balancing-algorithm consisted minimum connection method, weighted least connection method, minimum connection method based on location .

In practical application, there are many factors affecting client access frequency, node network-load , which are difficult to predict; If the system is long time running, the load can't be timely corrected, hence the effects of the load-balancing is not good [4]. Many Researchers have

studied the load-balancing algorithms and improved them. Liu proposed a load-balancing algorithm based on dynamic feedback[5], which considers the node performance, real-time load of the server and assigns the task according to the dynamic feedback. This algorithm does not make use of the node performance and real-time load when assigning the task thus its performance is not good. Geng proposed a load-balancing algorithm based on adaptive weight. This algorithm reflects real-time by adaptive weight [6];Wang.el proposed improved active load-balancing algorithm by adjust parameter in receive station serverb by WEB clustered system which could waste too much compute cost.Besides l[7].

In summary, by compare the advantage and disavantage both static load-balancing and active load-balancing algorithm,this paper proposed an impoved algorithm with weighted minimum connect number, by adjust balance the sever quality and active weight,then combine sever node as active connection number to assign tasks.

2. Traditional load balancing algorithm

The main task of the load balancing module is to receive network requests, and calculate the server's load . Load balancing policy is adjusted by load balancing algorithm during server operation, And then distribute the request to the appropriate server, finish the load balancing process finally.

In the existing load balancing algorithm, There are some load balancing algorithms based on real-time load conditions, The existing weighted minimum connection (WLC) algorithm can effectively balance the load of the server cluster and distribute user requests.

2.1 Weighted Minimum Connection Number Algorithm (WLC)

Weighted least connection algorithm is based on the minimum connection (LC) algorithm. When the load balancer receives a new task request, it responds to the task request by selecting the one has the smallest ratio of Server connection number and weight in the current server group. Assuming that the server set is S = (S1, S2 ..... Sn), the initial weight of the i-th server is W (Si) (1≤i≤n), the current connection number C (Si). When the load balancer receives a new connection request, it selects the server (Sm)for service according to the following rules:

$$C(S_m) / W(S_m) = \min\{C(S_i) / W(S_i)\} \; i \in [1,2,\cdots n], W(S_i) \neq 0 \quad (1)$$

among them, (1) the calculation of division is more complex than multiplication, so the division calculation spends more CPU cycle. Then the Judgment rule $C(S_m) / W(S_m) > C(S_i) / W(S_i)$ can be effectively expressed by $C(S_m) \cdot W(S_m) > C(S_i) \cdot W(S_i)$,In the conditions above, the server's weight can not be zero (when the value is zero, the server is not scheduled).

The current WLC algorithm take the differences in performance of each server and the effect of connection number into consideration. it greatly improves the efficiency of the system. But reflecting the server's load status only by the current number of connections is not reasonable enough. It is also impossible to dynamically adjust the server weights according to the current response of the server for the unbalanced load between the servers.

2.2 Weighted Round-Robin algorithm (WRR)

Weighted Round-Robin algorithm is an improvement of the classical Round-Robin algorithm. This algorithm according to the performance of each server to assign different weights for each server performance is inconsistent. Its weight is relatively high for high performance server and it can accept the request is more; It's weight is relatively low for low performance and it can receive fewer requests.

In the process of scheduling algorithm, the request is carried out by the cyclic scheduling of the queue. It is a good solution to a single request pattern problem.

Weighted Round-Robin algorithm have some characteristics: It added the concept of weight on the basis of classical polling algorithm. However, the algorithm does not consider the effect of the number of connections and the state of the server and it cannot respond to server status in real time. It belongs to the static algorithm , and it has the limitation.

3. Design and Implementation of Improved Algorithm

In order to compensate for the shortcomings of the weighted minimum connection number algorithm, this paper proposes a load balancing improvement algorithm for weighted minimum connection number for Web server. In the load balancing process, we consider the processing capability of the background server and the current load of the background server. According to the size of each server's current load scale value of the reasonable allocation of the request which is real-time collection of server data information to change the traditional load calculation method, and these metrics and WLC algorithm to achieve improved algorithms to improve the user request scheduling efficiency purpose.

3.1 Node performance evaluation and real-time quantization of load

The load index is the standard for the load evaluation quantification, and the choice of different load indicators will lead to different load levels of the load at the same time. If all the resources of the host as a load indicator is unrealistic, one is a wide variety, the second is the collection and calculation are very complex, so a good load indicators should have the following conditions: (1) easy to obtain the index data to facilitate Multiple measurements; (2) the index data can objectively reflect the load situation.

3.2 Evaluation of node performance

Due to different types of services, the impact of various indicators on the load is also different. Acc ording to the situation to adjust the position of each index in the load, only need to change the corre sponding weights can be. The results show that the efficiency of cluster is different when using diff erent factors. The character of Web server based on HTTP request, for performance of the node in t he cluster system mainly from the number of CPU n, CPU frequency, memory capacity, disk rate, n etwork throughput and other indicators to consider, using equation (1) to calculate.

$$C(S_i) = k_1 \times C(C_i) + k_2 \times C(D_i) + k_3 \times C(N_i) \quad i = 0,1,\cdots,n-1, \sum k = 1 \tag{2}$$

3.3 Calculation of the system load

In the paper, the server load is defined as L, C is the CPU usage , the memory occupancy rate is M, the usage of the connection is P, the network bandwidth occupancy rate is B, and D represent the disk usage and $C, M, P, B, D \in [0,1]$.For a formal formula, we usually give a formula for weighting the sum of the data, as shown in Eq. 3:

$$L = \lambda_1 C + \lambda_2 M + \lambda_3 P + \lambda_4 B + \lambda_5 D \quad \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 = 1 \tag{3}$$

In (Eq. 3) each utilization factor represents the importance of each resource in the overall load. After the normalization of the coefficients, the finally results is $L \in [0,1]$.When the usage of some resources is high, the system will crash. The cumulative sum of each coefficient is 1, in Eq.3 , so it

can only represent the importance of a specify resource, but can not equally reflect other resources. In the fuzzy set theory, there are three kinds of weighted integrated method: weighted average method, product mean method, mixed method. When utilization rate of the resource exceeds a certain value,and the ability of the system receiving new requests significantly decline , We should choose the product mean method[8]. In this paper, we use the product mean method to describe the server load, as shown in Equation 4:

$$L = 1-(1-\lambda_c C)(1-\lambda_m M)(1-\lambda pP)(1-\lambda_b B)(1-\lambda_d D) \quad \lambda_i \geq 0, \text{且} 0 \leq (1-\lambda_i X_i) \leq 1 \tag{4}$$

In (Eq. 4), if an application has a greater impact on one or more of these indicators, the corresponding coefficient can be increased, and the sum of the coefficients is no longer equal to 1, thus it is more objective description of the load generated by the server.

4. The periodic adjustment of weights

4.1 Load condition determination

The load-balancer periodically receives the current load of the server and changes the value of weig ht according to the comparison with the value of the load and the threshold . Where the threshold h ere changes with the value of the incoming load. As the threshold is represented by the ritio of the p erformance of the current server versus all servers:

$$\delta = \frac{C_i}{\sum C_i} \sum L_i \tag{5}$$

When $L(S_i) \leq \delta$,we stimate the current load of the server node is a low load, indicating that the load is relatively light at this time, then its weight should be increased; when $L(S_i) > \delta$, we stimate the current load of the server node is a high load, indicating that the load is relatively heavy at this time, then its weight should be reduced.and in this paper,we have introduced a modified variable ∂ ,it will calculate in the following fomula (6):

$$\partial = \left| 1 - \frac{C(S_i)/W(S_i)}{(1/n)\sum_{1}^{n} C(S_i)/W(S_i)} \right| \tag{6}$$

$W(S_i)$ indicates the initial weight of the server i , $C(S_i)$ indicates the current number of connections for the server,and $W(S_i)$ can not be zero. The new weight can be expressed as:

$$W(S_i) = \begin{cases} W(S_i) + \partial , & L(S_i) \leq \delta \\ W(S_i) - \partial, & L(S_i) > \delta \end{cases} \tag{7}$$

The new weight of the node $W(S_i)$ is periodically acquired, Select the server with the smallest ratio between the current connection and the updated weight to accept the new connection request,namely the server S_m accept the new request,at the same time ,it would satisfy:

$$C(S_m)/W(S_m)' = \min\left(C(S_i)/W(S_i)'\right) \tag{8}$$

4.2 Algorithm flow of the module

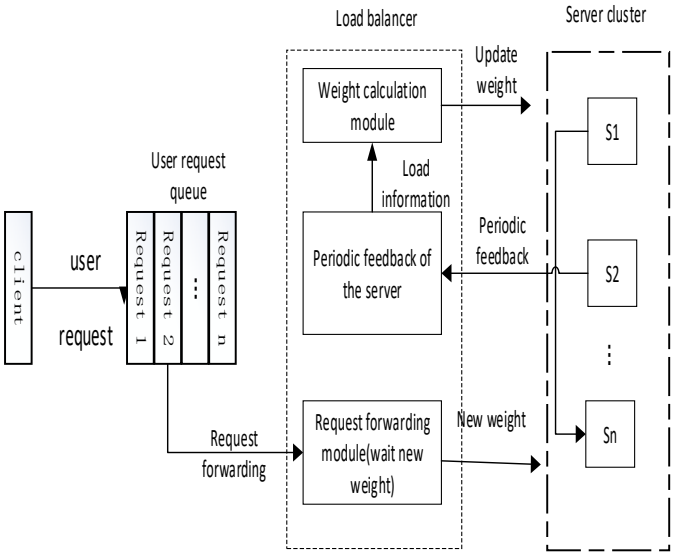

Fig.1 Module algorithm flow

4.3 Algorithm to achieve the work flow chart

Each incoming user request must go through the following process in order to select the appropriate server,as shown in fig.2

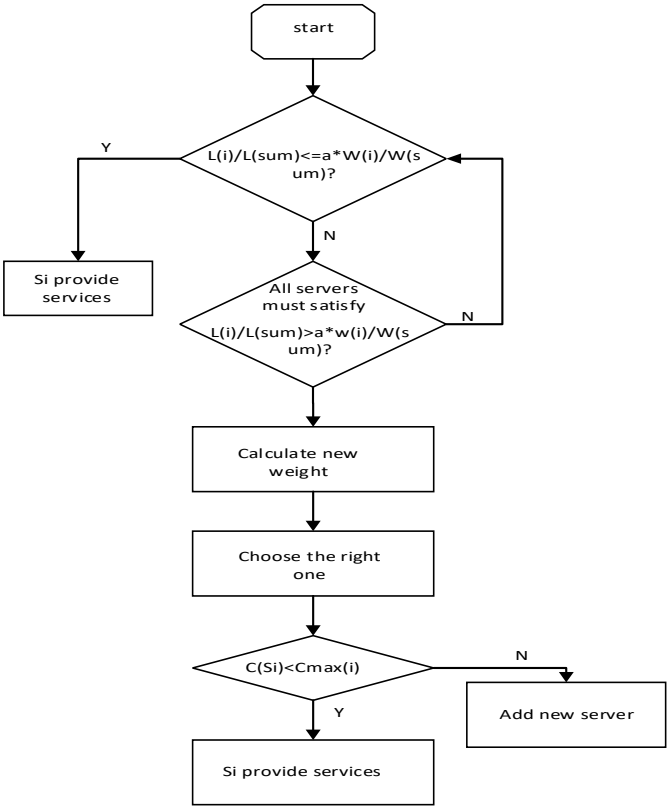

Fig.2 Algorithm work flow chart

4.4 Simulation Experiment and Result Analysis

In order to verify the basic performance of the improved algorithm, this experiment uses the networ k simulation software OPNET Modeler 14.5 to simulate and test[9]. and with the standard WLC algo rithm comparison. OPNET provides a three-tier modeling mechanism for bottom-up modeling at th e process layer, node layer, and network layer[10], while in the simulation process it uses a discrete e vent-driven simulation mechanism that accurately analyze the performance and behavior of comple x networks for data collection and statistics[11]. The network topology is shown in Figure 3.

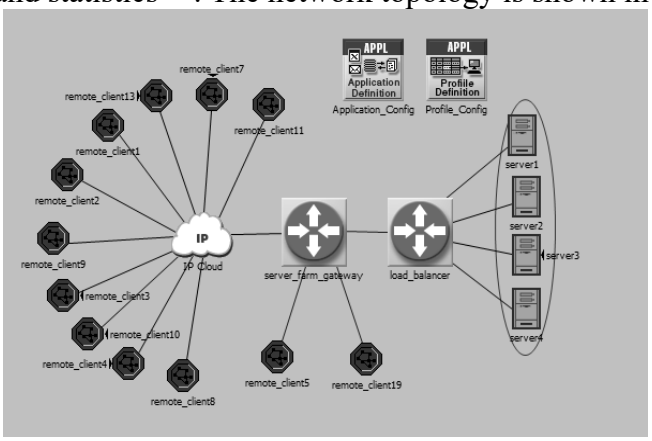

Fig.3 Network topology diagram

The server cluster consists of four Web servers, using 100M line, respectively server1, server2, server3, server4. The client consists of 12 subnets, each subnets containing 45 user terminals. To verify the algorithm in the cluster system , we use four different of servers to make up the server cluster. Select Emulation Set the HTTP application in Application_Config to simulate an HTTP picture and data load request scene. In order to simplify the operation, the initial parameters k1, k2, k3 values were 0.5,0.3,0.2, the simulation time is 30min, the update time is set to 10s. In this experiment, we mainly observe the average response time,and the CPU utilization ratio of the three algorithms.

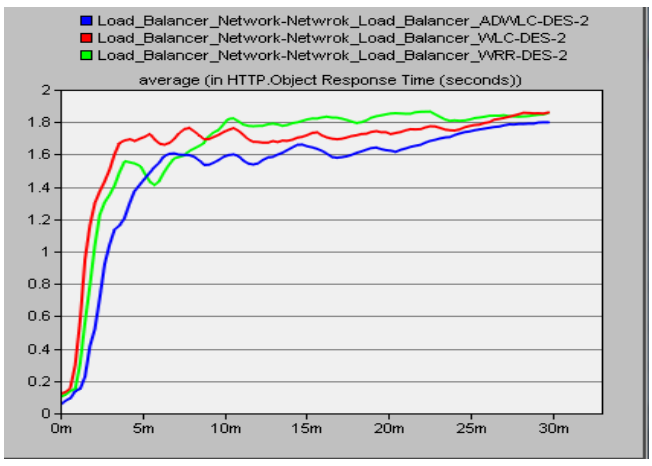

Fig.4 The http time comparison

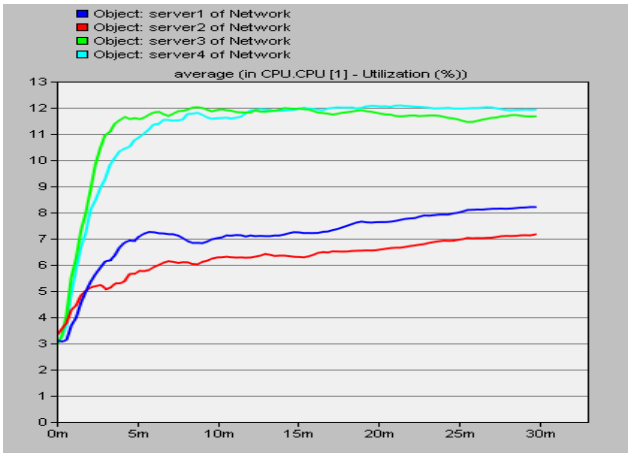

Fig.5 WRR algorithm for cpu utilization

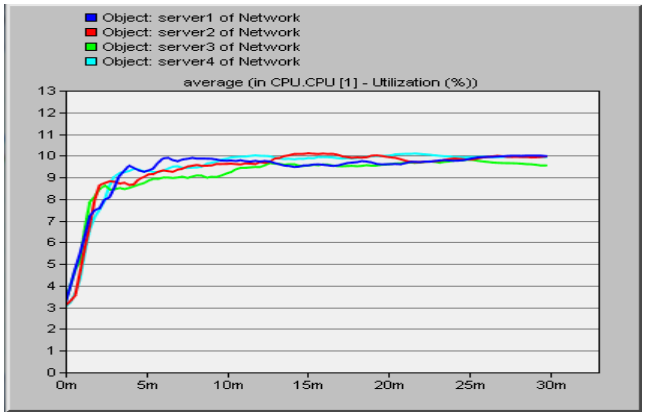

Fig.6 ADWLC algorithm for cpu utilization

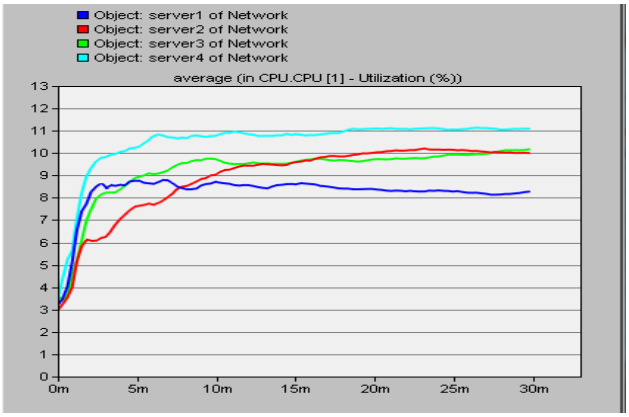

Fig.7 WLC algorithm for cpu utilization

It can be seen from Figure 4 in the HTTP request, ADWLC algorithm average response time of 1.8s or so, than the WLC algorithm and WRR effect is good.

Figure 5 shows that the CPU utilization of WRR algorithm is relatively scattered, the difference is large; and through Figure 6, we can see that the improved weighted minimum number of connections improved algorithm CPU utilization is more balanced, maintained at about 10%; 7 in the WLC algorithm, each server cpu utilization while maintaining a small gap, but still not balanced, and the utilization rate is higher;

5. Conclusions

The core of the Web clustering server is the load balancing algorithm. Compare to the Weighted Minimum Connection Number Algorithm, the algorithm put forward in this paper consider the impact of real-time load on the load balance, and quote the periodic feedback mechanism to dynamically change the weights, combined with the original Weighted Minimum Connection Number Algorithm mechanism to determine which server to accept a new connection request. According to the simulation results we can see, this algorithm can sufficiently reduce the HTTP responding time, reduce the delay of the server, balance the CPU utilization of each server.

Of course, the algorithm proposed in this paper also has its weakness, the ADWLS algorithm based on the improved minimum number of connections can only be achieved in the experimental environment, but not put into use in the real network environment yet. So, the next step is to put this algorithm on the actual network environment to run the test, to verify and improve its performance; Second, when the number of connections between the server is large, how to re-adjust the newly added server node is the next step to consider the issue.

Acknowledgements

National Science Fund subsidized project (61372087)

References

[1] Zhang Xi. Research on Dynamic workload Scheduling Strategy of LVS.University of Jilin 2010.

[2] Zhou Songquan. An Improved dynamic load-balancing Algorithm for Cluster[J].Computer and modernization 2012(1):135-139

[3] HWANG S T; JUNG N S Dynamic scheduling of web server cluster 2002.

[4] Hung-Chang Hsiao; Hao Liao; Ssu-Ta Chen; Kuo-Chan Huang. Load Balance with Imperfect Information in Strutured Peer-to-Peer System[J]. IEEE Transactions on Parallel and Distributed Systems.Volume22 .Issue4,2011, Page(s):634-649.

[5] Shen Wei Research and Realization of an improved load Balancing Algorithm Based on LVS Cluster. Journal of China university of geology, 2010

[6] Mikael Andersson, Jianhua Cao, Maria Kihl, and Christian Nyberg. Admission Control with service level agreement s for a web server.In proc.of European Internet and Multimedia Systems and Applications,2005.

[7] Nakagawa, Y., Suda, H., Ukigai, M., & Miida, Y. (2003). An innovative hands-on laboratory for teaching a networkingcourse. 33rd ASEE/IEEE Frontier in Education Conference, Boulder, CO, November 2003.

[8] Jussara Almeida, Mihaela Dabu, and Pei Cao. Providing differentiated levels of service in web content hosting. In proc. of the First Workshop on Internet Server Performance, 2008.

[9] Fitzhugh, S. (2012). Portable network laboratory. 32nd ASEE/IEEE Frontier in Education Conference, Boston, MA,November 2002.

[10] Sandeep S. W. Classification of dynamic load balancing strategies in a Network of Workstations [J]. ITNG, 2008:1005-1014.

[11] Wu Wendan .The Rearch on Load Balancing Technology for Video Transmission in IP Network [D]. Journal of Wuhan University of Science and Technology 2010.