Nginx cluster distribution strategy based on request load information

MA YuanLong[1,a],JIANG Yi[1,b],ZHANG ZhenWei[2,c]

[1] College of Computer Science and Technology, Chongqing University of Post and Telecommunications, Chongqing 400065, China

[2] ZTE Chongqing Research Institute, Chongqing 400065, China

[a]562497276@qq.com,[b]jiangyi@cqupt.edu.cn,[c]1172200695@qq.com

Keyword: Nginx; Request load information; Threshold value; Load balancing strategy

Abstract. According to the different characteristics of load values of the user's request,this paperpropose a Nginx cluster load distribution strategy based on the load information request. According to the relationship between the average load information of server cluster and the threshold value, combine with the threshold load information request, we select the load balancing strategy and the back-end server for different request distribution.We realize the reasonable distribution of the load and maximize utilization of server resources. Experiments and results show that in this paper the Nginx cluster distribution strategy request based on load information in two aspects of average throughput and average delay are better than the existing load balancing strategies.

Introduction

The rapid growth of Internet users to bring huge benefits to network operators but also bring them great network load[1].Now we use server cluster to solve this problem.Nginx adopts the following two load balancing categories[2,3]:one category such as the weighted round robin strategy[4],but it does not take the differences in request load information and the back-end server cluster performance into account[5];the other category such as the literature [6] proposed "the load balancing technology based on content",although this strategy classify user requests by the content, this strategy does not take the information and the back-end server cluster hardware load request consisting of different situations into account.

Nginx cluster distribution strategy based on request load information

The classification of Web requests.

This paper refers to CAP[7,8] Web for classification:(1) "N" request: W publishing service of The static and slightly dynamic;(2) "DB" request: Hard disk intensive service;(3) "CB" request: Intensive use service of CPU-based;(4) "DCB" request: The intensive use service of the hard disk and the CPU.

Server load

Server load evaluation

We use the server load evaluation standard proposed by literature [9]. The specific calculation method as shown in equation (1).

$$Load = 1-(1-\lambda_c C)*(1-\lambda_m M)*(1-\lambda_p P)*(1-\lambda_b B)*(1-\lambda_d D) \qquad (1)$$

The C issaid the utilization rate of CPU, M is said the utilization rate of memory, P is said the utilization rate of connections number, B is said the utilization rate of network bandwidth and D is said the utilization rate of disk utilization rate of I/O.

The reporting period of load

The load balancer according to the load information of the back-end server as a basis to distribute requests,but there are two problems about the load period T:one is the selection of T is too short,the other one is the selection of T is too long.We know the value of T will influence the performance of Cluster,from experiment we find thatit's best when the value of T is 10s.

Load reporting strategy

In T time not all of the load information of the server will change.In this paper we introduce formula (2) to show the server's load change rate of Δt to control the report server load.

$$\Delta t = \left| \frac{load(t_2) - load(t_1)}{t_2 - t_1} \right| (t_2 > t_1) \qquad (2)$$

Δt is said this servers' load change rate; $load(t_2)$ is said the servers' load value at the t_2 moment, $load(t_1)$ is said the servers' load value at the t_1 moment.Through the experiment ,it's the best result when the value of T is 10s.

Request load

For different types of user requests, it consumes server resources is not the same. Through experiments with different types of request load values we get the value of "N" request is 1,the value of "DB" request is 2,the value of "CB" request is 2 and the value of "DCB" request is 3.

Selection of threshold value

We compare the weighted round robin with literature [6] proposed "content based load balancing technology" in average delay.The experiment results show that the switching threshold value of load balancing strategy is one-tenth of the maximum load value of whatthe server can withstand.

List form

This paper we use a binary sort tree to manage the cluster server list .Load balancer need to maintain a binary sort tree for each request access cluster.

The distribution strategy based on request load information

Architecture

Depending on the load value of the request type request and considering the back-end heterogeneous cluster at the same time. Figure 2.2 gives the structure of the load balancer.In this structure,the load balancer contains the request classification、request distribution and load feedback acquisition module.And we add load information rate module on each server in the cluster server.

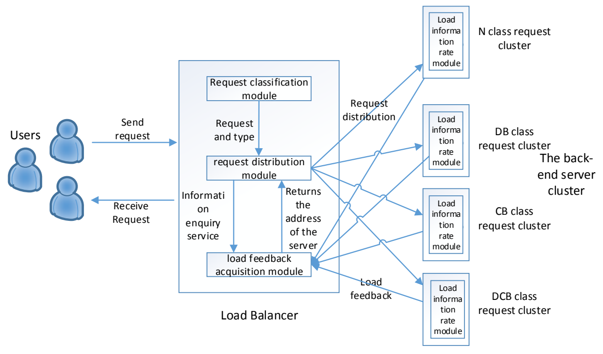

Users
Send request
Receive Request
Request classification module
Request and type
request distribution module
Information enquiry service
Returns the address of the server
load feedback acquisition module
Load Balancer
Request distribution
Load feedback
Load information rate module
N class request cluster
DB class request cluster
The back-end server cluster
CB class request cluster
DCB class request cluster

Figure 2.2 The architecture diagram

The initialization phase

(1) Initialize cluster partition

First of all, we let the back-end heterogeneous server cluster initialization for different types ofrequests cluster according to the back-end server performance metrics for each server in the cluster as well as the amount of each type of request.Such as $C_i\{S_{ij}, S_{ij}, S_{ij} \cdots\}$, the C_i is said the request type cluster and S_{ij} is said the j server in cluster C_i.

(2) Initialize the server threshold

According to the experimental method above.The average value of each server in the cluster as the cluster load average value. And take one-tenth of the average value of cluster load as threshold.

(3) Initialize the load feedback acquisition module

Load feedback acquisition module according above,establish the type of request processing cluster and the structure of the server nodes. The node is defined as $Node_{ij} = \{IP_{ij}, Load_{ij}, Max_{ij}, Type_{ij}\}$, i is said the corresponding cluster type; j is said a server in i cluster; IP_{ij} is a IP of a server in

S_{ij} ; $Load_{ij}$ isthe server's load calculated by formula (1),and initial binary sort trees according to this value; Max_{ij} is the maximum load value server can bear; $Type_{ij}$ is the type of a request of S_{ij} processing.

The load balancing module in each type of request processing maintains a node $Head_i = \left\{\overline{Load_i}, \overline{Threshold_i}\right\}$ in a cluster, which $\overline{Load_i}$ represents the average load value of the i server in the cluster and $\overline{Threshold_i}$ represents the average threshold value of the i server in the cluster.

Service request processing phase

The request distribution process consists of eight steps:

① $\mathrm{Re}quest(url)$:The user input url to the load balancer;

② $\mathrm{Re}ceive(url)$: Request classification module receives the user request url, then deliver the url and request type $type$ to the request distribution module in the form $(url, type)$.

③ $Query(url, type)$: The request distribution module receives the data $(url, type)$,,and then deliver the request type $type$ to the load feedback acquisition module to ask the back-end server ip .

④ $Inquire(type)$: The load feedback acquisition module receives the request type $type$, then asking the value of $\overline{Load}$ in corresponding request classification cluster node $Head$ is more than $\overline{Threshold}$, if not to the step ⑤ or turn to ⑥.

⑤ Use_WRR: The request will use Nginx weighted polling to select a server, then to the step ⑦.

⑥ $Minc\mathrm{os}t$: The load feedback acquisition module according to load values $load\_type$ of the request type to find a maximum server node in the corresponding binary sort tree $load\_type + Load \le Max$,then to the step ⑦.

⑦ $\mathrm{Re}turn()$: The feedback module returns the server ip to request load distribution module.

⑧The request distribution module receives the server ip , then distributes the user requests url to the server for processing.

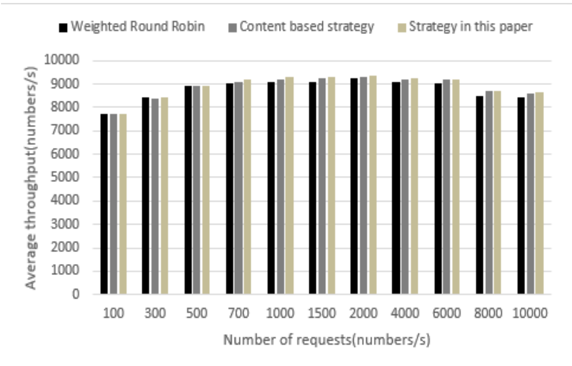
Experiment and result analysis

In this paper we compare three different load balancing strategies which are the weighted round robin, "load balancing technology based on content" is putted in literature [6] and the improved load balancing strategy we proposeed in the two aspects of the average throughput and average delay by experiments[10-12].

We select different number of requests per second,average throughput of four load balancing strategy as shown in Figure 3.1 and Figure 3.2.Experiments show that when the number of concurrent requests is less,as our load balancing strategy use weighted polling strategy and weighted polling strategy ,our strategy is superior to basedon the content strategy in average throughput and average delay. With the increase in the number of concurrent requests,our load balancing strategy is superior to the existing load balancing strategy based on content in two aspects of average throughput and average delay.

Summary

In this paper,we propose a distribution strategy based on request load information.We divide the back-end server cluster into multiple request processing cluster according to user request type.At the same time the load balancer will consider the request load value when distribute the request. The load



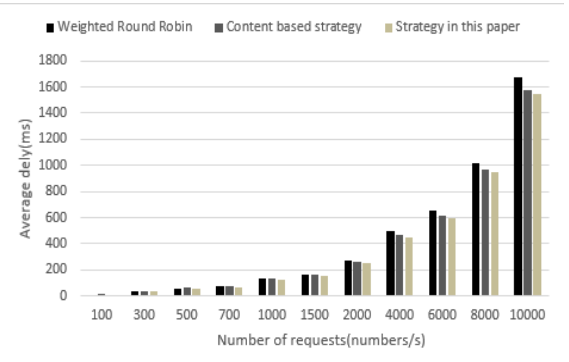


Figure 3.1 The average throughputFigure 3.2 The average dely

balancer will choose a suitable load balancing strategy for load distribution According to the server load information is whether or not more than the threshold. Next step we will solve the problems of single point of load balancer based on our study.

Acknowledgements

This work was financially supported bythe Chongqing Municipal Education Commission scientific and technological research projects(KJ1400414) and the Chongqing University of Posts and Telecommunications, Dr. startup fund, A2015-18.

References

[1] Yunfeng Li. The study of the load balancing based on web server cluster[D]. Chongqing University,2002.
[2] Huang Qian. DESIGN AND IMPLEMENTATION OF WEB CLUSTER LOAD BALANCING SYSTEM[D]. University of Electronic Science and Technology,2014.
[3] Lan Xiang. The Research and Improvement of the load balancing technology based on Nginx[D]. South China University of Technology,2012.
[4] REN Guo-qing, YANG Jin-min, ZHANG Da-fang. Content-based Dynamic Load-balancing Algorithm of Web Server[J]. Computer Engineering, 2010, 36(13):82-83.
[5] DU Zeng-Kai, ZHENG Ming-Yang, JU Jiu-Bin. A Distributed Algorithm for Content-Aware Web Server Clusters[J]. Journal of Software, 2003, 14(12):2068-2073.
[6] HuangFu Ning. Research and implement of a content-based load balancing technology[D]. South China University of Technology,2013.
[7] ZHENG Qi, ZHOU Guang-Ping. Content-Classification Load Balancing Algorithm in Cluster[J]. Computer System Application, 2011, 20(5):47-50.
[8] ZHANG Hao, LIAO Jianxin, ZHU Xiaomin. Advanced Dynamic Feedback and Random Dispatch
Load-balance Algorithm[J]. Computer Engineering, 2007, 33(4):97-99.
[9] Zhu Zhenguang. RESEARCH ON TECHNOLOGY OF DYNAMIC LOAD BALANCING FOR CDN[D]. Harbin Institute of Technology,2012.
[10] ZHANG Yu-fang,WEI Qin-lei,ZHAO Ying. Load balancing algorithm based on load weights[J]. Application Research of Computers, 2012, 29(12):4711-4713.
[11] LING Yun, ZHOU Hua-feng. Researches on dynamic load-balancing technique for heterogeneous cluster system[J]. Computer Engineering and Design, 2008, 29(12):3068-3070.
[12] LIU Xu-ri DONG Yu-min. Heterogeneous Cluster Server dynamic weighted load balancing algorithm[J]. Microcomputer Information, 2009(27):201-203.