Prediction-Based Elastic Load Balancing Mechanism in Cloud Environment

Xin Yang [a], Xiuquan Qiao [b]

State Key Laboratory of Networking and Switching, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[a]1105842102@qq.com, [b]qiaoxq@bupt.edu.cn

Abstract. An elastic load balancing mechanism in cloud computing environment is studied in this paper. The mechanism that uses kNN (k-Nearest Neighbors) and Naive Bayes classification algorithms in machine learning can effectively solve the problem of resource allocation lag by predicting the future load trend on the basis of analysis and study of historical data. And taking into account the cross regional nature of the cloud computing environment, applications will be deployed to the computing nodes closer to the user to reduce user access time. Finally, we verify the feasibility and effectiveness of the proposed mechanism through some experiments.

Keywords: Elastic load balancing; Cloud computing; Machine learning.

1. Introduction

Elastic load balancing is a kind of load balancing mechanism based on monitoring feedback to achieve dynamic load change [1]. The mechanism can adjust the resources dynamically according to the load change situation, and the cost of the user is greatly saved according to the requirement. Different from the traditional load balancing mechanism in the physical cluster, it is more convenient to create/recycle resources for load balancing in cloud environment because of its establishment in the virtual machine cluster. This feature makes the elastic load balancing mechanism become more realistic and operational. However, the application of the load machine need to spend some time in the process of elastic load balancing, resulting in the lag of resource allocation and a bad experience to costumers. Research [2, 3] shows that the user access frequency has a more stable distribution in the long term, so it can effectively solve the problem of resource allocation lag by predicting the trend before the arrival of the peak according to the historical data in advance to prepare the load machine [4, 5]. Different from a single data center, cloud computing is cross regional and its computer room may be distributed in a few very distant place apart [6]. Network requests are very time consuming for remote users so that the system return time will be greatly prolonged if the load machine is all placed in the same place. This situation is not only a waste of network bandwidth resources, but also is not conducive to the user experience. The load balancing in cloud computing need distribute the request of load machine to different nodes according to the status of the network request of different regions and the overall application of the overall load condition. Therefore, this paper proposes a new intelligent elastic load balancing mechanism based on historical trend prediction in cloud computing environment.

2. Trend prediction based on kNN classification algorithm and naive Bayes classification algorithm

2.1 Related concept definition

The basic data is divided into two parts, one is the network request packet, and the other is the cluster load.

Definition 1: network request feature. The network request packet section records the feature of user's request, including access time T, source IP address A. Access time record the time the user requests achieve the load machine, and source IP address records the user's geographical attribution.

Definition 2: workload feature. Each load characteristic data includes CPU usage C1, memory usage rate C2, disk usage C3 and network IO status C4 in a single load machine. The data will be accompanied by a time stamp to identify the data acquisition time.

Definition 3: regional division characteristics. The data is a multidimensional array, which records the distance between the national various places and each computer room. This distance will be updated regularly, and it is longer if the transmission time of two place is longer. This article takes the province as the unit to carry on the region division.

2.2 Geographic Classification of Request based on kNN Classification Algorithm

kNN (k-Nearest Neighbors) algorithm attribute an unknown sample X to the majority of categories that the most of the nearest K samples from X belong to [7].

In the kNN algorithm, an instance of the nearest neighbor is defined according to the standard Euclidean distance. An arbitrary instance of X can be represented as the following feature vector:

< a 1 (x) , a 2 (x),..., a n (x)>

An r (x) represents the first R attribute value of the instance x. The distance between the two instances of I X and j x is defined as d (x i, x j). d (x i , x j) can be calculated by:

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}$$

We can get the category that the most of the nearest K instances from the instance X belong to. The probability of X belong to this category is the largest from the point of view of probability theory so that the instance X can be divided into this category.

The distance between each province can be calculated with the latitude and longitude of the province as its coordinate attribute value. Each province will be divided into different categories by the distance between this province and each computing room through kNN based on some training data. These classified data will be saved to regional division array.

The above analysis is based on the situation of the same congestion status of networks and the same network transmission speed in different parts of the network. However, this situation does not exist exactly in reality. In order to better simulate the reality of the network, we can use the distance weighted nearest neighbor algorithm to divide by adding a weight to the distance between two places. The more congestion and the lower transmission speed in network, the greater the value of the weight is.

We can classify each network request according to regional division array with the province which is gotten by the IP of the request. Then we can count on the number of requests to access the application on each small time period (10 minutes etc) and get the application access features during the entire time period.

2.3 Load Machine Condition Classification Based On Naive Bayes Classification Algorithm

The classification of the load status of the machine is divided into three kinds in this paper: overload state T0, normal state T1, idle state T2. The division of these states can not be divided simply according to whether a single indicator reaches a threshold value. The classification should be carried out synthetically according to the load machine CPU usage, memory usage, disk load status and the status of network IO.

Item to be classified can be classified to the category whose probability is the biggest among the probability of occurrence of each category under the condition of the occurrence of the item in Naive Bayes. This classification method is not only simple and effective, but also can reduce the extra load because the information collected from each node can be processed in parallel using Map/Reduce mechanism in cloud computing environment [8].

2.4 Trend Forecast

Data of access request distribution and load data of the load machine for any period of the week can be obtained through the analysis and calculation of 1.2 and 1.3. These data can help us to forecast the trend of the same period of time in the next week and create/recycle resource in advance in the corresponding resource area unit.

3. Elastic load balancing

3.1 Framework

The elastic load balancing mechanism in this paper is mainly composed of the virtual machine image template library, the virtual machine generator, the resource pool, the configuration analyzer, the resource pool manager and so on. Each module is defined as follows:

1) The virtual machine image template library: Store the user virtual machine image template which is used to generate a mirror image of a user virtual machine.

2) The virtual machine generator: it creates virtual machines with the corresponding template gotten from the virtual machine image template library if the resource pool does not have the corresponding mirror virtual machine.

3) The resource pool: store some virtual machines not in the job queue temporarily. Query the resource pool when the load balancing mechanism needs to create a virtual machine.

4) The resource pool manager: Responsible for the creation / recovery of virtual machine resources.

5) The configuration analyzer: Be responsible for monitoring and analyzing periodic data of the virtual cluster and load balancing and predict the trend according to the historical data.

3.2 Elastic Load Process

A typical elastic load process is shown in Figure 1.

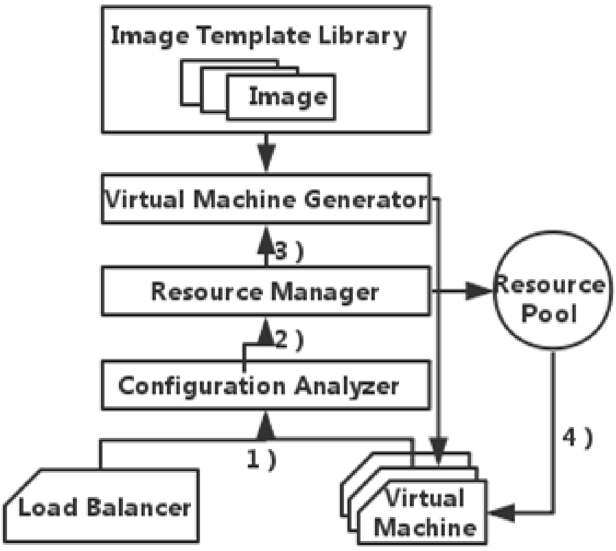


Figure 1. Typical elastic load process

The specific process is as follows:

The monitoring program on virtual cluster and load balancer will send CPU usage, memory usage, disk usage and network of IO monitoring data and some user data such as IP and access time to the virtual machine configuration analyzer regularly.

The configuration analyzer collect, store and analyze data. When the collection of the virtual machine monitoring data in a single indicator is greater than the maximum threshold or the combination of various indicators through the naive Bayes classification is in accordance with the state of overload, it will send a request to the resource pool manager to create a virtual machine resource. When load machines are in a low load condition, it will send resource recovery instruction. Another job of the configuration analyzer is to be responsible for the analysis of trends in the past week's historical data. It will issue a resource operation request to the manager at a given time point and update related DNS server or load equalizer according to the trend prediction results.

The resource pool manager sends out the query operation to the resource pool at first when the new virtual machine is needed to join the load balancing cluster if the application is overloaded. If there is a corresponding mirror template virtual machine in the resource pool, take out and add it to the load cluster. Virtual machine will be removed from the cluster and added to the resource pool when the application is in low load.

The virtual machine generator will create virtual machine with the template from the virtual machine image template library when there is not a corresponding virtual machine in the resource pool.

4. Experimental results and analysis

We use the two computing nodes A and B in OpenStack environment to simulate two computer room that are far apart from each other and simulate the user's access request with 30 IP in this experiment. We need to manually set the coordinates of these IP points because these 30 IP belong to the same geographical environment. These IP are classified into class A and class B by taking 12 of them as the training data. Then we can modify load equalizer and make specific IP can only access the virtual machines created by specific computing node. 50 groups of virtual machine monitoring data on the virtual machine load conditions will be token and divided into three types of low load, normal and overload. These labeled data will be used as the training data for the future of the virtual machine load status of the classification. Three hours of data a day before the experiment will be collected with 10 minutes as a unit as training data that are used for trend prediction. Figure 2 shows the three hours system response time variation on the experimental day.

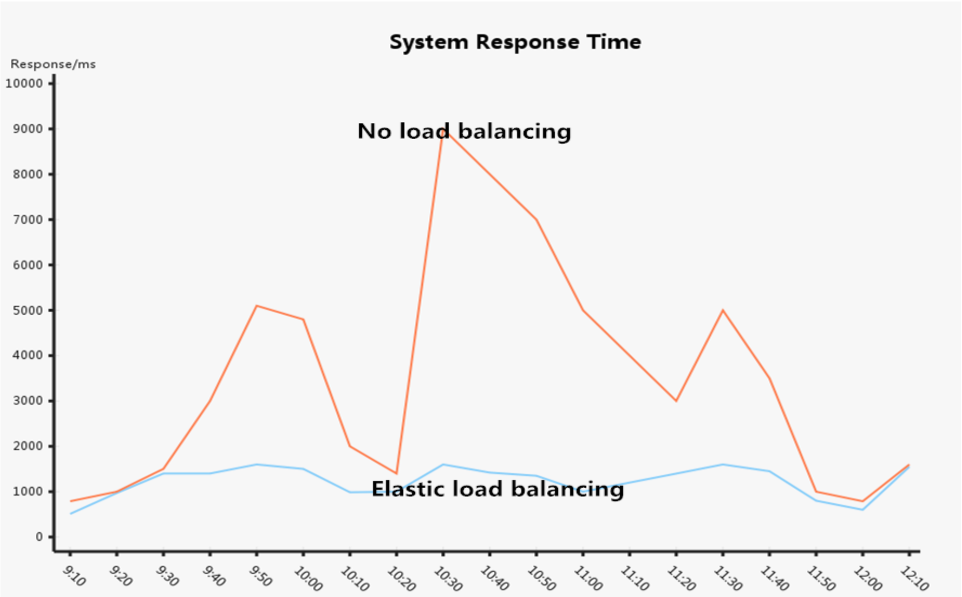
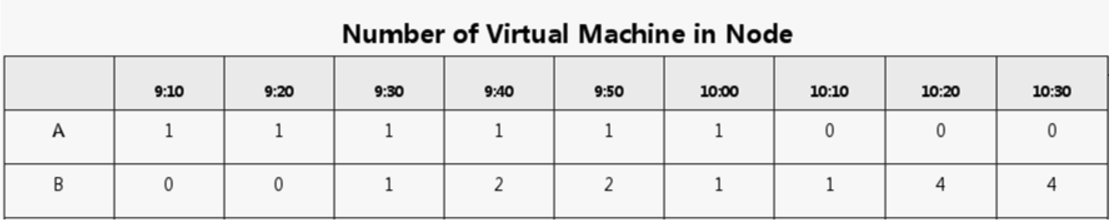


Figure 2. Three hours system response time variation on the experimental day

As shown in Figure 2, the response time in the system without elastic load balancing has a significant upward trend when three visit peaks come, but system with elastic load balancing has only a slight upward trend. The response time in the system without elastic load balancing drop significantly when low visit peak come, but system with elastic load balancing has only a slight down trend and will be essentially flat with an elastic load balancing system. It can be seen that the elastic load balancing mechanism can effectively reduce the system response time.

The number of virtual machines created by the A and B two computing nodes in each time period is shown in table 1 and table 2.

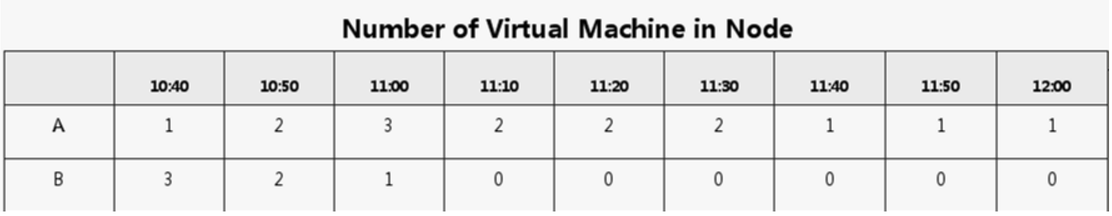
Table 1. Number of virtual machines created by the A and B two computing nodes (9:10~10:30)



Number of Virtual Machine in Node

	9:10	9:20	9:30	9:40	9:50	10:00	10:10	10:20	10:30
A	1	1	1	1	1	1	0	0	0
B	0	0	1	2	2	1	1	4	4

Table 2. Number of virtual machines created by the A and B two computing nodes (10:40~12:00)



Number of Virtual Machine in Node

	10:40	10:50	11:00	11:10	11:20	11:30	11:40	11:50	12:00
A	1	2	3	2	2	2	1	1	1
B	3	2	1	0	0	0	0	0	0

As shown in Table 1, the proportion of the virtual machine belonging to the B computing nodes is getting higher and higher at 9:10 to 10:30. This is consistent with the results expected of increasing access from IP class B during this time. As shown in Table 2, the proportion of the virtual machine belonging to the A computing nodes is getting higher and higher at 10:30 to 12:00. This is consistent with the results expected of increasing access from IP class A during this time.

5. Conclusion

This paper proposes a new cross region elastic load balancing mechanism based on historical data in the cloud computing environment, which can effectively solve the response lag and not flexible allocation in traditional load balance mechanism. Considering the cross regional and network transmission complexity of computing nodes in cloud computing environment, it will greatly reduce the user's access time if the application is automatically deployed to the least user access time consuming nodes.

Acknowledgements

In this paper, the research was sponsored by the Service Identification and Migration mechanism of Future Internet (No. 2012CB315802), National Key Basic Research Program of China (973 Program).

The research was also sponsored by Web-based Dynamic Service Provisioning Mechanism and Method Research in Information-Centric Networking (No. 61671081), Project of National Natural Science Foundation of China (NSFC).

References

[1] Yao D U, Guo T, Chen J. Fleet elastic load balancing mechanism in cloud environment [J]. Journal of Computer Applications, 2013, 33(3):830-833.

[2] Barabási A L. The Origins of Bursts and Heavy Tails in Human Dynamics [J]. Nature, 2005, 435(7039): 207-11.

[3] Gelman A. Bursts: The Hidden Pattern Behind Everything We Do [J]. Physics Today, 2010, 63(63): 46-46.

[4] Gao X W, Wang Q, Ouyang Y M. Algorithm research of load balancing based on blending prediction model [J]. Computer Engineering & Design, 2010, 31 (16): 3557-3561.

[5] Wang Q, Xin-Hua H E, Zhao Y K, et al. Load Balancing Algorithm Based on Access Characters-load Prediction[J]. Journal of Academy of Armored Force Engineering, 2009.

[6] He-Sheng W U, Wang C J, Xie J Y. TeraPELB-an Algorithm of Prediction-based Elastic Load Balancing in Cloud Computing[J]. Journal of System Simulation, 2013, 25(8):1751-1750.

[7] Soucy P, Mineau G W. A simple KNN algorithm for text categorization[C]// IEEE International Conference on Data Mining. IEEE, 2001:647.

[8] CAI Song, ZHANG Jianming, CHEN Jiming. Load balancing technology based on naive Bayes algorithm in cloud computing environment [J].Journal of Computer Applications, 2014, 34(2):360-364.