The Adaptive Load Balancing Algorithm in Cloud Computing

Wucai Lin[1,a] , Lichen Zhang[2,b]

[1]School of Computer Science,Guangdong University of Technology Guangdong, Guangzhou 510006, China.

[2]School of Computer Science,Guangdong University of Technology Guangdong, Guangzhou 510006, China.

[a]claytale@163.com

Keywords: Cloud computing, Load balance, Resource Scheduling, Adaptive.

Abstract. Cloud computing is a calculation based on the Internet, through this way, the hardware and software resources and information sharing may be provided to a computer or other devices on demand. Load balancing algorithm plays a very important role in cloud computing environments. Through efficient load balancing algorithms, we can improve the utilization of a variety of hardware and software resources, avoiding some resources in the high load state, and some resources are idle, causing the overall system resource utilization is very low. This paper proposes different load balancing algorithms applied in different environments to analyze their strengths and weaknesses, and then propose a dynamic adaptive load balancing algorithm that we first classify user's requests and quantify them, and take physics Real-time load of resources into account to select the appropriate resources are allocated to the job.

1. Introduction

Cloud computing is a new distributed computing technology, on which more and more applications are beginning to develop and deploy based. In the cloud computing environment, we can get any configurations available computer resources (such as network, servers, storage, applications and services) through the simple network access in any place any time. Through the cloud service providers, such as Amazon, Google, Microsoft, we can rapidly configure and publish the application with minimal management. Cloud computing provides a service oriented architecture (SOA) and network services (IOS) applications, including fault tolerance and scalability, availability, flexibility, reduce the overhead of information technology, which help users reduce the purchase cost with on-demand services.

In order to avoid the cloud computing environment in some resource overload, and some resources in idle state, the load balance algorithm[1-5] is used to make all the resources of the load to be balanced, which increases system throughput, reduces the response time and improves the user experience. The load balancing algorithm is divided into content-blind algorithm and content-aware algorithm. The content-blind algorithm includes: RR (round robin), WRR (weighted round robin) and LC (least connection), WLS (weighted least connection), LL (least loaded) and random server selection. CAP (aware policy Content) belongs to the content-aware load balancing algorithm.

In this paper, a new load balancing algorithm is proposed based on the available load of each resource in the cloud environment. In the algorithm, we firstly classify the jobs according to the job execution time, and then puts forward a quantitative mathematical function load based on each type of operation in the number of jobs, so it can be adaptive of the combined work load and the cluster node availability and load on load distribution, which can guarantee the load balancing and efficient execution of the cloud system.

2. Related Work

In order to make full use of cloud computing resources, many academics and enterprises in the industry are studying and designing different load balancing architecture. Cloud computing load balancing algorithm is divided into static and dynamic. The static load balancing algorithm such as

ran (random) and RR (Round-robin) is the first generation of load balancing algorithm, in which in load distribution strategy the cloud environment resources in the basic information are needed and the real time load information of cloud nodes are not taken into account. In the second generation load balancing algorithm has been improved, such as WRR (round-robin weighted), LL (connection least) and WLC (Least Connections weighted). The second generation algorithm is mainly used to collect the information of the nodes in the cloud environment (such as CPU load, disk usage and the number of activities of the network connection) to carry out the distribution of load. The collection of statistical information in real time, at different times, the node information in the cloud environment are different to put the information from each node is transmitted to the load balancing system requires a lot of computer resources (CPU, I / O, and bandwidth), especially in large-scale cloud computing environment, is prone to network storm.

Job classification strategy of CAP algorithm based on job classification according to the load demand, combined with the available cloud load of all nodes, respectively, resource allocation for each type of work.

3. Improvement of CAP algorithm

Above we introduced to the job classification, here mainly according to the requirements of CPU classification. The same CPU for the need for a class, labeled as J class (j=1,... C). Under normal circumstances, the operation is divided into static and dynamic operation. Static job refers to the operation does not need to be calculated directly from the memory or hard drive to read, for example, a picture of the web server to get the request is a static job. Since the execution time of the static request is proportional to the rate of the return result of the request, we can classify it by the size of the flow. And the demand for CPU is very little. When a static operation request arrives, classification module from operation parameters in parsing out the job request file name and according to the type of file query operation type table work, then the the type parameter is added to the operation parameters, as behind the load distribution and scheduling with.

Dynamic job is CPU intensive, the content of the request can not be known in advance, may be required by the calculated or retrieved from the database. The classification of dynamic operation, here mainly through the operation on the performance requirements of CPU and CPU core number core requirements as classification, the performance of CPU by the number of instructions per second MIPS (millios instructions per second).

Throughput is a standard to measure the throughput performance of the system, but also in the environment of cloud computing node performance scale. Usually, the throughput of the system with a convex parabolic load increase. The beginning of the throughput increases with the increasing load, until the arrival of the parabola peak time, operation is a node that can be processed by the critical value, the throughput will decrease, this time if there job is assigned to a node, the job entered the ready queue, waiting for the next node scheduling. The number of jobs when a node exceeds the critical value, the node throughput began to decline, in order to achieve the highest throughput, need to obtain the critical value of the load of nodes.

The previous section we through the demand of CPU in the same job for a class. Here each class defines a job J weight W (J), said the average load CPU compared to J class work of other types of work. At the same time, the definition of the critical value for the Nc job type J (J), all types of operation in the smallest critical value of Nm can be expressed as follows:

$$N_m = \min\left\{N_c(1),...,N_c(j),...,N_c(C)\right\} \tag{1}$$

The minimum critical value is related to the job class with high CPU requirements, so we can define the weight of the job type J as follows:

$$w(j) = \frac{N_c(j)}{N_m}, j \in \{1,...,C\} \tag{2}$$

If the largest processing capacity of the cloud of node i is Nm. Thus, we can estimate each server load from each job type j of the number of jobs weighted. Based on the assumption that in time in each web switching an instance of vector N(i) with each network server cluster and from each level j in Web server I tracked are pending request number associated:

$$N(i) = \{n(i,j) \mid j = 1,\ldots,C\} \tag{3}$$

Thus we can estimate the load L (i) for each node in the cloud :

$$L(i) = \frac{\sum_{j=1}^{C} n(i,j)}{w(j)} \tag{4}$$

Among them, w (j) represents the weight of the job assigned to j in each node. According to the formula (2) and the formula (4), we can get the available load AC (i) of each node i:

$$AC(i) = \mathrm{Nm} - \sum_{j=1}^{C} n(i,j) \tag{5}$$

The AC (i) is used to represent the available load of each node i in the cloud. Then we assign jobs the node which has the maximum available AC load (i) values . Whenever a job assigned to node i, after a job finished, the load of i will be updated. Given the following update formula update AC (i, j):

$$AC(i) = AC(i) \pm \frac{1}{w(j)} \tag{6}$$

4. Simulation experiment and result analysis

In order to measure the efficiency of the algorithm, we use CloudSim[19] simulation software to simulate the test

The algorithm efficiency evaluation index is as follows:

(1) the average response time of a user request

(2) the throughput of system

In the test, we use the CAP algorithm and WRR algorithm to compare with our improved algorithm, the following is the results of the test data.

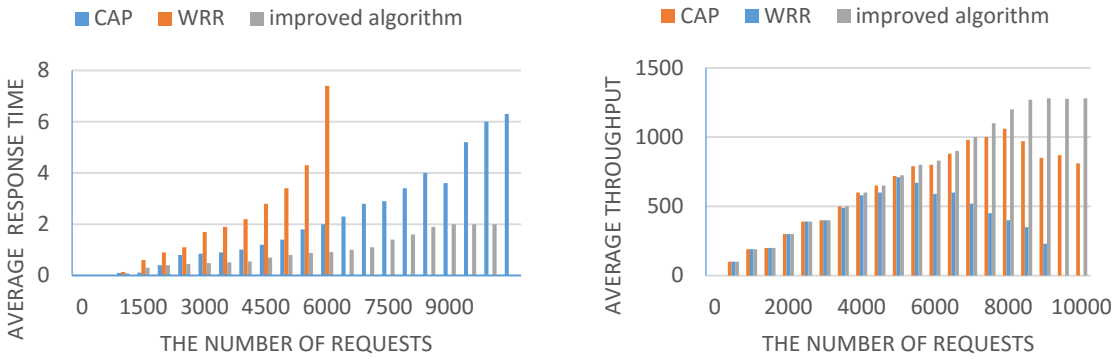

Fig.1 the results of the test data

Figure 1 reflects the relationship between the average job response time and the number of systems using different algorithms under. From the picture we can draw the number of jobs is less than 3000, three kinds of algorithm under the response time difference, when the job number exceeds 3000, the gap becomes large, obvious static load balancing algorithm of WRR because of not considering the nodes in the system, the response time is the longest, the CAP algorithm also does not consider the load information system, and the improved CAP algorithm combines the advantages of CAP algorithm combined with the load information system, make the system load

balance, improve the utilization of cloud resources, the system response time by using CAP and WRR algorithm under normal circumstances, the system response time is shorter, more jobs can be processed at the same time, from Figure 2 we can see that when the job number is greater than 6000, using the WRR algorithm system throughput Began to decline, the number of operations is greater than 7500, the system throughput of the system using CAP algorithm began to reduce, when the number of jobs more than 9000, the system throughput of the improved CAP algorithm to reach the maximum, and tend to be stable.

5. Summary

In a dynamic, multidimensional cloud computing in virtual environment, load balancing is used to improve resource utilization, shorten the system response time and improve system throughput and ensure effective guarantee system stability and availability. This paper proposes an adaptive load balancing algorithm, the load distribution, not only consider the specific requirements of job, and analysis of the load information of the nodes in the system in real time. Because the demand for different operations on the resources are different, we according to the operation requirements for CPU of job classification, and give priority to different types of work, the higher the priority, the higher the probability of resource allocation. At the same time we in unit time load assessment in real time the cloud node, finally can avoid some node overload and some nodes are idle. So it can make full use of resources, at the same time to consider Various resources of the specific load, the final load distribution, so that all the resources in the cloud environment load balance, improve resource utilization, shorten the system response time.

References

[1] Ratan Mishra and Anant Jaiswal,"Ant Colony Optimization: A solution of Load Balancing in Cloud", International Journal of Web & Semantic Technology (IJWesT), April 2012

[2] Y. Zhang, H. Franke, J. E. Moreira, A. Sivasubramaniam," A Comparative Analysis of Space- and Time-Sharing Techniques for Parallel Job Scheduling in Large Scale Parallel Systems", José E. Moreira,November,2015,pp.1-13.

[3] B. Gorda and R. Wolski. Time Sharing Massively Parallel Machines. In International Conference on Parallel Processing, volume II, pages 214–217, August 1995.

[4] D. G. Feitelson. A Survey of Scheduling in Multiprogrammed Parallel Systems. Technical Report RC 19790 (87657), IBM T. J. Watson Research Center, October 1994.

[5] A. B. Downey. Using Queue Time Predictions for Processor Allocation. In IPPS'97 Workshop on Job Scheduling Strategies for Parallel Processing, volume 1291 of Lecture Notes in Computer Science, pages 35–57. Springer-Verlag, April 1997