TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.4, April 2014, pp. 2850 ~ 2859
DOI: http://dx.doi.org/10.11591/telkomnika.v12i4.4731

Performance Evaluation of Dynamic Load Balancing Algorithms

Tianshu You, Wenhui Li, Zhiyi Fang*, Hongbin Wang, Guannan Qu
College of Computer Science and Technology, Jilin University, Changchun, China
*Corresponding author, e-mail: fangzy@jlu.edu.cn

Abstract

Efficient task scheduling mechanism can better meet the users' QoS requirement, and achieve the load balancing in physical hosts, so the cluster system with high scalability and reliability can effectively improve the information utilization of the system, thereby enhance the overall performance of the Web server cluster system. In order to build a network service system with better scalability and reliability, this paper describes the round-robin scheduling algorithm of LVS cluster system, least-connection scheduling algorithm, weighted least-connection scheduling algorithm and a prior proposed new weighted value assigned scheduling algorithm, dynamic adaptive feedback load balancing strategy. Meanwhile, it takes simulation experiment for the round-robin scheduling algorithm of LVS, least-connection scheduling algorithm, weighted least-connection scheduling algorithm and the new weighted value assigned scheduling algorithm, dynamic adaptive feedback load balancing strategy and take comparative analysis for the experimental data, through the analysis and assessment, effectively point out the advantages and disadvantages of the existing load balancing strategy. It is conducive to better improve the existing equalization algorithm performance deficiencies and propose optimum load balancing strategy.

Keywords: *QoS, load balancing, LVS cluster system, performance analysis, performance evaluation*

1. Introduction

With the rapid development of network technology, the application of the network has been widely popular. This trend makes high-speed Internet service become very popular in the world of public network services. For example: Amazon, Dangdang, Jingdong Mall and Taobao e-commerce systems, online train tickets booking system, online panic buying system and other network services.

These network services sometimes encounter instant and rapid increased visits, causing the entire system slow down or even website crash. To provide a large number of client requests and better performance, operators use the Web server cluster to solve the server performance issues. Expand the server's bandwidth through the load balancing technology, increase the throughput of the system, strengthen the network data processing capability, and increase network flexibility and availability.

Load balancing technology make server load achieve a relatively balance, thereby reducing the response time of client requests task, increasing the utilization of system resources, so that the performance of the system can be improved. A good scheduling strategy can effectively solve the problem of network congestion, the nearest provided when service needed to achieve location independence; provides the users with better access quality; improve the speed of response from the server; improve the utilization efficiency of the server and other resources; avoid network's key parts having a single point failure [1].

Efficient task scheduling mechanisms can better satisfy the user's QoS requirements, realizing the load balance between each physical host, improving the overall performance of the Web server cluster system. Task scheduling is an important part of the cluster system, it executes through mapping the task or tasks submitted by users to appropriate resources, and its efficiency will directly affect the performance of the entire Web server cluster system [2-4]. By using load balancing technology, it effectively improves the resource utilization of each server, improve the service availability of the system [5].

The next generation of Internet services might need more high-scalability and high-availability systems on the high-speed Internet aspect. In order to provide a more general and a

Received September 5, 2013; Revised November 3, 2013; Accepted November 21, 2013

variety of next-generation Internet services, we need to develop a highly scalable and reliable cluster system [6].

In order to analyze and evaluate the dynamic load balancing algorithm performance of Web server cluster system, we perform several experiments and measure various load balancing algorithm performance of the cluster system from load balancing indicators. We compared the laboratory proposed weighted least connection scheduling algorithm with the general LVS (Linux Virtual Server) clusters using the cycle, the least connection and weighted least connection scheduling algorithms.

The paper is organized as follows: Section two describes the domestic and foreign researchers' work on the load balancing algorithm performance analysis; Section three of this paper will be used for performance analysis of load balancing strategy; Section four takes performance analysis for the algorithms decribed in section three; Section five is the summary.

2. Related Work

In the application environment of Web server cluster system, for the client's request task, it is an important factor affecting the overall system's performance on how the load balancer optimum schedules the load. Among the solutions of high-performance distributed load, some take hardware scheme, others take software scheme. No matter which one to take, we have to consider the following issues [1]:

(1) After taking the load balancing scheme, the speed of server to receive and transmit datagram and the overall detection capability of load balancing are the primary considerations.

(2) The load balancing scheme should be able to meet the growing demand of network traffic, balance the load of different operating systems and hardware platforms, as well as different load flow.

(3) Load balancing equipment should have good redundancy solution when failure happens, ensure the availability, avoid the system suffer a huge loss.

(4) A flexible, intuitive and safe management measure is easy to install, configure, maintain and monitor, improve work efficiency and avoid errors.

At present, a large number of domestic and foreign documents describe the performance evaluation and analysis of load balancing strategy. In these documents, they take performance analysis and evaluation of the load balancing algorithms and existing algorithms, measure the advantages and disadvantages of the algorithm on performance, help people design better algorithm to meet the actual application requirements.

Surdeanu M and other people [7] proposed a kind of distributed Q/A system, using interquestion parallelism and dynamic load balancing to improve the system's throughput, and reduce the personal problems' response time.

Modern parallel adaptive grid computing simulation has various sizes, Iqbal S and Carey GF [8] studied and compared the characteristics which performed by the dynamic change of the processors' number when the four load-balancing algorithms are having parallel computation.

In the documents of Koyama K, Shimizu K, Ashihara H [9], the authors used simulation method to compare and evaluate various adaptive load balancing strategy in actual environment, the adaptive load balancing strategy has a better load balance, and better to improve the overall performance of the cluster system, and can be realized in practical applications.

Shan Z, Lin C, Marinescu DC's documents [10] introduced the load balancing strategy about the web server cluster and HTTP request content and priority process scheduling mechanism. This method can ensure the quality of load balancing and network service (QoS).

Yang J, Jin D, Li Y 's documents [11] take modeling and simulation for cluster-based real web server, build system performance analysis model, for data sent to server, each part of the system simulate and calculate the packet delay. Introduce the network address translation, ip tunnel and direct routing three kinds of load balancing. Delay as part of the input model of the running system and the system model to measure the transmission data packet in accordance with the system and the analog system performance. Through the performance assessment and the analysis of possible performance bottlenecks to adjust the solution to the problem, in order to find the maximum processing capacity of the system.

Teo YM, Ayani R's documents [12] use 16 PC and a drive track simulator two experiment platforms to take cluster-based web server performance and scalability experiment analysis. Through the performance analysis and evaluation it can be seen that round-robin scheduling algorithm is not as good as the other two algorithms, the least-connection algorithm is easy to implement, and is suitable for high load situation. However, with the decrease of load, the least-connection scheduling algorithm's waiting time is more than 2-6 times of the least-load scheduling algorithm. The performance of least-load scheduling algorithm is the best, but it needs the time information of each request service.

Distributed server cluster system is a cost-effective solution to provide scalable and reliable Internet service. In order to achieve high-quality service, it is necessary to adjust the system configuration parameters and use different algorithms to improve system performance.

3. Load Balancing Strategies

After reading and studying a large number of performance analysis articles, this article elaborated the round-robin scheduling algorithm, the least-connection scheduling algorithm, weighted least-connection scheduling algorithm and a new laboratory proposed weighted distribution scheduling algorithm and dynamic adaptive feedback load balancing scheduling algorithm of LVS cluster system.

3.1. Scheduling Algorithm of LVS Cluster System

(1) Round-robin scheduling algorithm (Round-Robin Scheduling, rr).

Round-robin scheduling algorithm is also known as 1:1 scheduling algorithm, when the load balancer receives a service request from the client, it will send the request to the back-end real server to process in accordance with 1:1 proportion. In the realization of algorithm, we consider the state of the real back-end server to ensure that the task execute properly.

Assume that a cluster system has n server nodes S_0, S_1, S_2, …,Sn_{-1}, k means the ID of the last task assigned to the service node, the algorithm is described as follows:

Input: the last select task assigned server node and the total server data;
Output: select the server to be assigned task.
Round_Robin_Algorithm(node k, Servers n).
1. Settings and initialize variables j=k；
do：
2. j=（j+1）mod n；
3. if（S_j is alive）
4. k=j；
5. return S_k；
6. else nothing to handle；
7. while（j !=k）//if agree return to execute do, not return NULL
8. return NULL；
Code Description：

```
Round_Robin_Algorihtm(node k, Servers n)
{
    int j=k;
do {
            j=(j+1)mod n;
            if(Sj is alive)
           {
                 k=j;
                 return Sk;
            }
            else{
                  }

     }while(j!=k)
     return NULL;
}
```

Round-robin algorithm is a simple scheduling algorithm, load balancer select a server to process customers' requests in turn, so that only when the back-end server configuration has the same condition and each user request almost has the same system resources cost, it can provide the best load balancing efficiency.

(2) Least-Connection Scheduling Algorithm (Least-Connection Scheduling, lc).

Least-Connection Scheduling Algorithm assess the real-time load status of back-end server nodes through the connection number recorded on the load balancer, delivering the new request to the server node which has the least number of connections to process.

Assume that a cluster system has n server nodes S_0, S_1, S_2, ..., S_{n-1}, C (S_i) means the number of connections of the i server node now, S_m means the server assigned for the new request, and the least-connection scheduling algorithm can be described as follows.

Input: the last select task assigned server node and the total server data;

Output: select the server to be assigned task.

Least_Connection_Algorithm (node m, Servers n)

1. for (traverse server cluster);
2. Determine whether the m server works properly; if not, return to step 1 and move to the next server, if so, continue to step 3;
3. for (traverse the server cluster from the m+1 server);
4. find the server with the least connections;
5. find the corresponding server, returns S_m;
6. returns NULL if not find the corresponding server.

Code Description：

```
Least_Connection_Algorithm(node m, Servers n)

{
   for(m=0;m<n;m++)
  {
     if(Sm is alive)
    {
       for(i=m+1;i<n;i++)
           {
             if(C(Si)< C(Sm))
             m=i;
             }
       return Sm;
      }
    }
   return NULL;
}
```

Least-Connection algorithm is a simple dynamic load balancing algorithm, it can take connection as unit and dynamically transfer the load request with different lengths to the back-end server for processing, when all the servers' processing performance are similar, it can be obtained good load balancing efficiency.

(3) Weighted Least-Connection Scheduling Algorithm (Weighted Least-Connection Scheduling, wlc)

The weighted least-connection scheduling algorithm is the improvement for the least-connection algorithm, not only consider the real-time number of connections for each server, but also consider the process capacity of each server, take division operation for connection number of each server and their processing performance which means the load of each server, and thus to determine the real-time status of each server, take the least ratio of servers to handle the connection request sent by clients.

Assume that a cluster system has n server nodes S_0, S_1, S_2, ..., S_{n-1}, W (S_i) means the weight value of the i server node, C (S_i) means the number of connections of the i server node now, Sm means the server assigned for the new request.

S_m server meets the following conditions:

$$(C(S_m) / CSUM) / W(S_m) = \min \{(C(S_i) / CSUM) / W(S_i)\} \quad (i = 0,1, ..., n-1)$$

Where W (S_i) is not zero, because in this round lookup CSUM is a constant, so the judgment conditions can be simplified as:

$C(S_m) / W(S_m) = \min\{C(S_i) / W(S_i)\}$ (i = 0,1, ..., n-1)

Where W (S_i) is not zero, the judgment condition can be changed inequality operation to compare.

$C(S_m) / W(S_m) > C(S_i) / W(S_i)$

As division comparison operation consume more CPU resources than multiplication comparison operation, and the float point division is not allowed in the Linux kernel, connection number and weighted division comparison operation can be converted to the connection number and weighted multiplication comparison operation to compare, server weights are all greater than zero, so the judgment condition can be further optimized as follow:

$C(S_m)\times W(S_i) > C(S_i)\times W(S_m)$

At the same time ensure that the server has a weight of zero, the server will not be scheduled, and the weighted least-connection scheduling algorithm can be described as follows:

Input: the last select task assigned server node and the total server data;
Output: select the server to be assigned task.
Weighted_Least_Connection_Algorithm(node m, Servers n).

1. for (traverse server cluster);
2. determine whether the weight value of m server is above 0,in another way whether it works properly; if not, return to step 1 and move to the next server, if so, continue to step 3;
3. for (traverse the server cluster from the m+1 server);
4. find the server with the least ratio between connection number and weight value;
5. find the corresponding server, returns S_m;
6. returns NULL if not find the corresponding server;

Code Description:

```
Weighted_Least_Connection_Algorithm(node m, Servers n)
{
   for(m=0;m<n;m++)
   {
     if(W(Sm)>0)
     {
        for(i=m+1;i<n;i++)
            {
             if(C(Sm)*W(Si)>C(Si)*W(Sm))
                m=i;
             }
       return Sm;
       }
   }
return NULL;
}
```

Weighted least-connection algorithm is the most efficient algorithm on load balancing of LVS cluster system, it not only consider the number of real-time connection of the server, but also the difference between the back-end server processing performance, truly realize a dynamic load balancing.

3.2. A New Weighted Value Assigned Scheduling Algorithm

In the new weighted value assigned scheduling algorithm [13], we use the CPU idle rate and memory idle rate of back-end sub-server which periodically acquired by load balancing to calculate the weight of the back-end sub-server. The weighted value function expression can be expressed as:

$$F(S_i)=0.6\times(1-C(S_i))+0.4\times M(S_i) \quad (1)$$

$C(S_i)$, $M(S_i)$,$W(S_i)$ denote the CPU utilization, memory idle rate and the server node weight of the each back-end server, $F(S_i)$ denotes the new weight of back-end server node. After the new weight calculation we determine whether the new weight should be written to the IPVS scheduling according to the following formula:

$$\begin{cases} |W(S_i)-F(S_i)|\le B \quad W(S_i); \\ |W(S_i)-F(S_i)|>B \quad W(S_i)=F(S_i). \end{cases} \quad (2)$$

B is a pre-given boundary value, the value of B indicates the algorithm effective assess value after the improvement of weighted least-connection scheduling algorithm. From the formula above it is not difficult to see that the smaller the value of B, the higher frequency of the weight value to write to IPVS. If the B value is 0, then every load information result collected can satisfy the boundary value condition. The load results collected will also be written to the IPVS scheduling. If B = 1, then the load information result collected cannot satisfy the boundary value condition, with IPVS without any change, still maintain the original scheduling mode.

3.3. Dynamic Adaptive Feedback Load Balancing Strategy

A new weighted distribution scheduling algorithm is periodically making a request by the load balancer to collect the load information of back-end sub-server node, this approach increases the communication overhead of the whole cluster system, also increases the load balancer's burden. Feedback dynamic adaptive load balancing strategy [14] periodically and adaptively collect their own load by the back-end server node, and send the load information to the load balancer. This approach reduces the communication overhead of the whole cluster system, and also reduces the burden of the load balancer. In order to avoid the instant overload of the load balancer and a single server node, also introduce the server node load redundancy value this parameter to further optimize the performance of the cluster system.

About the weight calculation of the back-end sub-server, we use server performance and dynamic load value these two important parameters to make more accurate assessment of the current load capacity of the server nodes, we assume that the weight value is W, the weight W about the server performance and dynamic load values these two parameters can be calculated as follows:

$$W(S_i)=L(S_i)/P(S_i) \quad (3)$$

$P(S_i)$ is the performance indicator of the server, decided by server's CPU utilization, available disk space and memory space. $L(S_i)$ is the dynamic load value of the server node, decided by CPU utilization, disk I/O read and write speed, memory utilization, network bandwidth utilization and request response time.

4. Performance Analysis and Evaluation

In the web server cluster system, it mainly use effective measure indicators of server to analyze and evaluate whether it is good or bad the cluster system's load balancing performance. These effective measure indicators [1] mainly include CPU utilization, memory usage, bandwidth utilization, disk IO throughput and network IO throughput, the number of completed service per unit time, the number of connected clients per unit time, and the response time to complete a task request.

This article focuses on a new weighted value assigned scheduling algorithm, dynamic adaptive feedback strategy and round-robin scheduling algorithm of LVS cluster system, the least-connection scheduling algorithm and weighted least-connection scheduling algorithm for performance analysis and evaluation. Compare and analyze the advantages and disadvantages of existing algorithms in order to better improve and optimize the algorithm. In Choi E's paper, in order to analyze the dynamics of server performance due to the workload, the authors perform several experiments. With some knowledge of how to select the proper performance counters and their proper threshold value, they compare the performance results of the ALBM cluster with the general LVS cluster using RR, LC and WLC scheduling algorithms. The ALBM cluster achieves the better performance than the LVS by balancing the loads among the servers. It also has verified and analyzed that among round-robin scheduling algorithm of LVS cluster system, least-connection scheduling algorithm and weighted least-connection scheduling algorithm, weighted least-connection scheduling algorithm has better performance in the three algorithms [6].

Here we take simulation experiment for the new weighted value assigned scheduling algorithm ,dynamic adaptive feedback load balancing strategy and weighted least-connection scheduling algorithm on the request response time, and take comparative analysis for the experimental data. Laboratory experiments were done under local environment, user access client uses WAS (Microsoft Web Application Stress Tool) to simulate the user pressure. Divided to 7 experimental groups, set the first group access number of users to 100, the second group access count is 200, and set 300,400,500,600,700 respectively in order. Each group take simulation tests with weighted least-connection scheduling algorithm , a new weighted value assigned scheduling algorithm and dynamic adaptive feedback load balancing strategy [13].

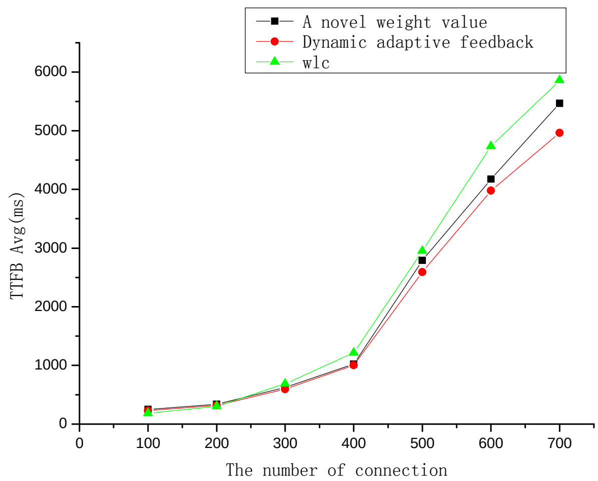

Figure 1. Response Time Comparison of the Two Algorithms

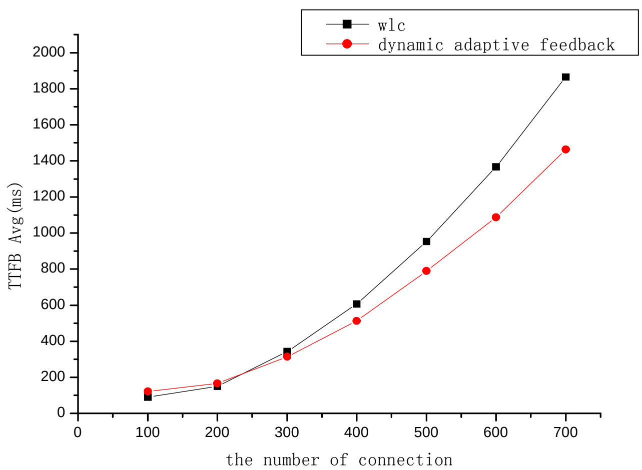

Figure 2. Response Time Comparison of the Two Algorithms

We can clearly see that from Figure 1, when the number of user request task connections is less than 230, the average wait response time of the improved algorithm is slightly longer than which of the wlc algorithm. The main reason for this phenomenon is that the improved algorithm has a new weight calculation and IPVS write process, that consumes a certain amount of time. When the number of task connection is greater than 230, with the increase number of the require task connection, the average wait response time of the improved algorithm significantly less than that of wlc algorithm. No matter the wlc algorithm, or the improved algorithm, with the increase number of task request connection, when it reaches a certain number, the whole cluster system will reach the saturation. Because the resource utilization of each server cluster system is already saturated. When this occurs, we should consider the cluster system to be expanded, adding a certain amount of back-end real servers to increase available server resources.

From the average wait response request time comparison of user request in Figure 2, we can see that when the number of user requests is less than 240, the average wait response time of the improved algorithm is slightly longer than which of the wlc algorithm. The main reason for this phenomenon is that the improved algorithm has weight value calculation and IPVS write process, this process consumes a certain amount of time. When the user request number is greater than 240, with the increase number of the user requests, dynamic adaptive feedback load balancing strategy responds faster than the weighted least-connection scheduling algorithm, dynamic adaptive feedback load balancing strategy is more effectively enhance the throughput of the system, and better balance the system load, can effectively improve the performance of the whole cluster system.

In the following we mainly take comparison analysis for a new weighted value assignment scheduling algorithm, dynamic adaptive feedback strategy and weighted least-connection scheduling algorithm from the response time and system throughput aspects.

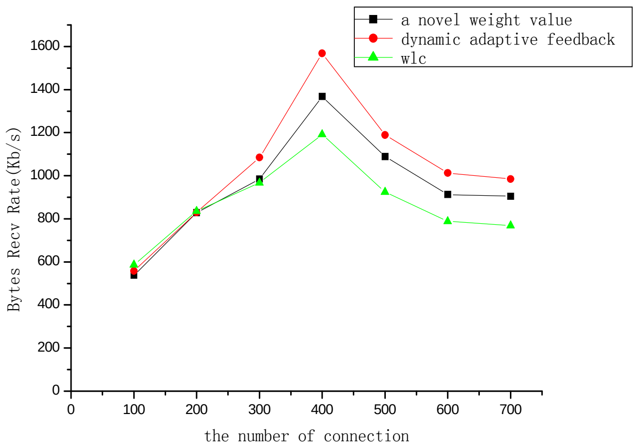

Figure 3. Note How the Caption is Centered in the Column

As can be seen from Figure 3, when the number of user' request connection is not big, a new weighted allocation scheduling algorithm and dynamic adaptive feedback load balancing strategy have longer response time for users' request than the weighted least-connection scheduling algorithm, the main reason is the former two algorithms increase the burden of the load scheduler when calculate the weights value, write and read the weight values, and load balancing assign tasks, causing additional communication overhead, cost some resources. With the increasing number of access requests connections of the current users, the new weighted assigned scheduling algorithm and dynamic adaptive feedback load balancing strategy are superior to the weighted least-connection scheduling algorithm on performance. In addition, it can also be seen from the diagram that the dynamic adaptive feedback load balancing strategy has least response time, it means that dynamic adaptive feedback strategy is superior to the new weighted distribution scheduling algorithm on performance.

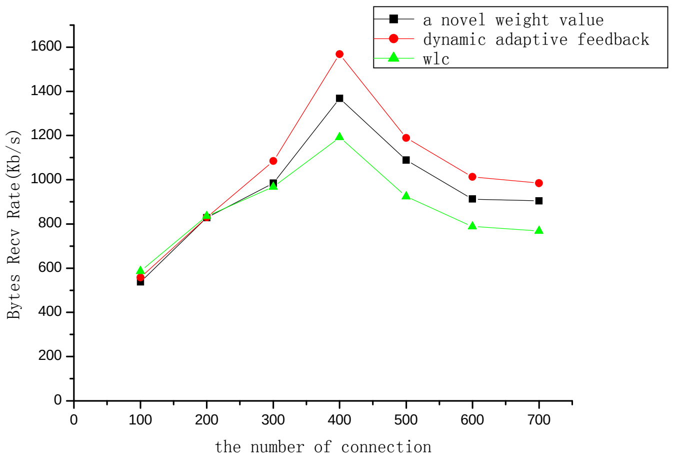

Figure 4. Note How the Caption is Centered in the Column

For dynamic pages, we can see from the above diagram, when the number of user access request connections is less than 200, the throughput of a new weighted assigned scheduling algorithm and dynamic adaptive feedback load balancing policy is slightly less than which of the weighted least-connection scheduling algorithm, mainly due to the first two algorithms increase the burden of the load scheduler in the process of weight calculation, the weight write, weight read and the task assignment of load balancer, causing additional communication overhead, spent some resources.When user access request connection reaches about 400, the throughput of the system reaches the maximum. Overall the dynamic adaptive feedback load balancing strategy has the maximum throughput. It means the dynamic adaptive feedback strategy is superior to the new weighted scheduling algorithm on performance.

In summary, through the comparative analysis of the three algorithms on performance, dynamic adaptive feedback load balancing strategy has better performance than the new weighted assigned scheduling algorithm and round-robin scheduling algorithm of the LVS cluster system, the least-connection scheduling algorithm and weighted least-connection scheduling algorithm.

5. Conclusion

This paper analyzed the performance analysis of the domestic and foreign document on the cluster system load balancing algorithm, its main purpose is to research and design a better performance of the load balancing algorithm, which makes the cluster system with high scalability and reliability, and can effectively improve the utilization and performance of the system information. Through the description, analysis and evaluation of a new weighted distribution scheduling algorithm, dynamic adaptive feedback load balancing strategy and round-robin scheduling algorithm of LVS cluster system, least-connection scheduling algorithm and weighted least-connection scheduling algorithm on performance. Finally the conclusion is that dynamic adaptive feedback load balancing strategy has better performance. Through the analysis and evaluation of the performance of the load balancing strategy of the cluster system, which can effectively point out the advantages and disadvantages of the existing load balancing strategy, is conducive to better improve the deficiencies of the existing equalization algorithm and propose optimum performance load balancing strategy.

Acknowledgements

This research is supported by China Jilin Province Natural Science Funds under program of parallelization and dynamic scheduling model of CPU/GPU cooperative high performance computing cluster (201215189).

References

[1] Xu Lan. Load balancing technology research. *Agriculture&Technology.* 2009; 29: 155-157.
[2] Sun H, Chunguang T. Optimizing Multi-agent MicroGrid Resource Scheduling by Co-Evolutionary with Preference. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(12): 7222-7229.
[3] Jiang W, Zhang J, Li J, et al. A Resource Scheduling Strategy in Cloud Computing Based on Multi-agent Genetic Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(11): 6563-6569.
[4] Quan B, Li H, Le Z. Dynamic Routing and Resource Assignment Algorithm In sloted optical Networks. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(4): 1813-1821.
[5] ZHU Jiayu, XIAO, Dan, WANG Fei. Multi-dimensional QoS constrained scheduling mechanism based on load balancing for Cloud computing. *Computer Engineering and Applications*. 2012.
[6] Choi E. Performance test and analysis for an adaptive load balancing mechanism on distributed server cluster systems. *Future Generation Computer Systems*. 2004; 20: 237-247.
[7] Surdeanu M, Moldovan DI, Harabagiu SM. Performance analysis of a distributed question/answering system. *IEEE Transactions on Parallel and Distributed Systems*. 2002; 13: 579-596.
[8] Iqbal S, Carey GF. Performance analysis of dynamic load balancing algorithms with variable number of processors. *Journal of parallel and distributed computing*. 2005; 65: 934-948.
[9] Koyama K, Shimizu K, Ashihara H, et al. *Performance evaluation of adaptive load balancing policies in distributed systems.* Proceedings of IEEE Singapore International Conference on. IEEE. 1993; 2: 606-611.
[10] Shan Z, Lin C, Marinescu DC. Modeling and performance analysis of QoS-aware load balancing of web-server clusters. *Computer Networks*. 2002; 40: 235-256.
[11] Yang J, Jin D, Li Y. Modeling and simulation of performance analysis for a cluster-based web server. *Simulation Modelling Practice and Theory*. 2006; 14: 188-200.
[12] Teo YM, Ayani R. Comparison of load balancing strategies on cluster-based web servers. *Simulation*. 2001; 77: 185-195.
[13] Wang Hongbin, Fang Zhiyi, Qu Guannan, Zhang Xi Zhang, Yunchun. A novel weight distribution scheduling algorithm. *Journal of Information and Computational Science*. 2011; 8: 1235-1244.
[14] Wang Hongbin, Fang Zhiyi, Cui Shuang. Dynamic adaptive feedback of load balancing strategy. *Journal of Information and Computational Science*. 2011; 8: 1901-1908.
[15] Wang Hongbin. Research on adaptive load balancing scheduling strategy in Web server cluster system. Jilin University. 2013.