International Journal of Computer Science and Applications,
©Technomathematics Research Foundation
Vol. 13, No. 1, pp. 31 – 41, 2016

A COGNITIVE NETWORK BASED ADAPTIVE LOAD BALANCING ALGORITHM FOR EMERGING TECHNOLOGY APPLICATIONS*

Kang Lu†

Science and Technology on Space Physics Laboratory
Beijing, 100076, China‡
lu-kang@139.com§

Ting Xing

IDC(Beijing) Co,Ltd,
Beijing, 100044, China
txing@idc.com

In the era of cloud computing and big data, the demand for real-time data processing and availability poses higher requirements for network load balancing. Cognitive network has unique self-learning and re-configuration abilities that can improve the effectiveness of load balancing. Based on the existing traffic scheduling algorithms, this article will discuss the possibility of improving weighted least-connection scheduling algorithm by leveraging cognitive network. A dynamic load balancing algorithm (NNPMA,Neural Network Prediction Model Algorithm) will be developed on the basis of traffic prediction model. NNPMA can enable least load scheduling in real time for service request from node and configure available idle resources upfront to ensure compliance with QoS. This traffic scheduling algorithm will be simulated with OPNET and it will be applied to cloud computing architecture. The test results indicate that this algorithm can achieve loading performance better than the unimproved weighted least-connection scheduling algorithm without significantly increasing network overhead.

Keywords: Cloud Computing; Big Data; Cognitive Network; Load Balancing.

1. INTRODUCTION

As cloud computing embarks on the stage of real application, hybrid cloud service becomes the mainstream model provided from cloud services at present stage, and big data analysis emerges to be a hotspot of cloud computing application. Versatile and massive data is generated explosively, data is growing at an unprecedented pace around the world; network access users, traffic and access requests increase by folds, all demanding higher network and data processing quality. In such context, a data center can't deploy its servers in the traditional pattern of single point integration, but leverages highly-efficient innovative service system architecture to realize more flexible cluster deployment and more reasonable resource configuration for building a resource server

pool. However, a heterogeneous cloud computing application platform will have resource server instances in various sizes, with an excessive load may create unbalanced loading at server side, so load balancing plays a crucial part in ensuring the efficient operation of cloud computing system architecture. Working on the design of cloud computing system architecture, the cognitive network adaptive load balancing algorithm can flexibly turn on/off cloud service resources and adaptively configure request for network service to optimal server instance, ensuring the uninterrupted services.

Currently load balancing has not been integrated into cloud computing platform, but works as a rather independent module. At present factors like distribution of terminals, random mobility and various QoS requirements for user services may cause uneven distribution of network traffic, leading to local node overloading and congestion at nodes with heavy load that increase packets lost rate and service latency, while idle resources at nodes with light load are in poor utilization. Cognitive network load balancing offers an innovative solution to this scenario.

LVS cluster architecture implements load balancing by modifying IP to schedule resources, and as the virtual instance resources on cloud computing service platform all have their own ID number and IP addresses, it is very easy to apply LVS cluster technology to enable adaptive configuration of cloud computing resources. But common scheduling algorithms of LVS can't be used in a scenario in which resources are changing dynamically. The NNPMA algorithm proposed by this article enables a load balancing and scheduling scheme suitable for use in cloud computing system architecture. While ensuring the compliance with user QoS requirements, it can configure upfront available network resources in accordance with load variation to schedule network traffic in real time and dynamically so that traffic can be distributed evenly across the entire network to reduce network congestion, offering a more effective solution for today's big data application processing and the extensive use of cloud computing.

2. CURRENT STATE OF RESEARCH

Scheduling algorithm is at the center of implementing load balancing. The traffic scheduling algorithms present in use have time latency, for they schedule and distribute traffic to balance load by following the results of current network state and the analysis of captured parameters [Kong *et al*,(2007)][Luo *et al*,(2008)][Li *et al*,(2005)][2-4]. Apparently the traffic scheduling in current network state has latency, while due to its upfront learning capability, cognitive network can essentially understand the upcoming traffic and evenly distribute network traffic to each server before the arrival of network service request, so as to maintain network load in balance. Load balancing is mainly designed to complete tasks like: overcome network congestion and provide service at proximity to realize geographical-location irrelevance; provide users better access qualities through accelerating the response speed of services; and improve the utilization of server and other resources to avoid single point failure happening at mission-critical part of the network. Scheduling algorithms used in today's network mainly include [Kong *et al*,(2007)][Luo *et al*,(2008)]: round-robin scheduling, weighted round-robin

scheduling, least-connection scheduling, weighted least-connection scheduling, destination address HASH scheduling and source address HASH scheduling. These algorithms can also be divided into static traffic scheduling and dynamic traffic scheduling, with the expansion of network and users, static traffic scheduling is no longer able to meet the requirements of load balancing, so a dynamic and adaptive traffic scheduling algorithm shall be developed to more closely meet the requirements from today's network applications.

The above scheduling algorithms play a certain role in solving server load balancing. However, with the growing size of the network and increase in the access requests, deficiencies exist in the scheduling performance of these scheduling algorithms [Li *et al*,(2005)][Liang *et al*,(2009)][K *et al*,(2004)][Yan *et al*,(2005]. They do not fully consider the load status of the server, and the round-robin scheduling algorithm and weighted round-robin scheduling algorithm are the stateless scheduling algorithm, and they are static scheduling [Li *et al*,(2005)]; the destination address hashing and source address hashing algorithm is a static mapping algorithm; for the least connection scheduling and weighted least connection scheduling algorithm [Li *et al*,(2005)], although they take into account the state of number of the server's connection requests and they are the dynamic scheduling, the server load is not just the number of connection requests, and other status information also needs to be considered. On the other hand, these scheduling algorithms cannot give feedback effectively based on real-time comprehensive state of the server [Hu *et al*,(2009)][Chen *et al*,(2006)][Wang *et al*,(2006)], and schedule the load request to the appropriate server. Most of the existing studies aim at the general packet data network, and conduct traffic scheduling by use of global status information obtained at a certain moment. Since it is difficult to obtain accurate global state information, and the performed scheduling cannot be adjusted in real time, thus the load balancing effect is limited and lags significantly [Wang *et al*,(2006)] [Chen *et al*,(2008)][FU *et al*,(2006)][Zhang *et al*,(2007)]. The network state learning capability of cognitive network enables the router to have environmental learning and adaptive processing mechanism, effectively solving the scheduling lag issue. Weighted least connection scheduling algorithm is the most efficient traffic scheduling algorithm among the basic dynamic scheduling algorithms [Zhang *et al*,(2007)], but it only takes into account the number of connection requests to the server, and CPU load, network traffic and memory usage are also important parameters affecting the load balancing.

In addition, the network traffic in real world exhibits features like strong nonlinearity and multi-scale, such as fractal, long-range correlation, self-similarity and abruptness; the cognition of the cognitive network is embodied in its ability to predict network traffic. Because traditional network traffic model can only handle steady process and special unsteady process, more discrepancy happens when describing network traffic behaviors. To address the multi-scale and nonlinearity of computer network traffic system, a new combined model that integrates the multi-resolution analysis of wavelet transformation and the nonlinear approximation of neural network needs to be built to predict network traffic.

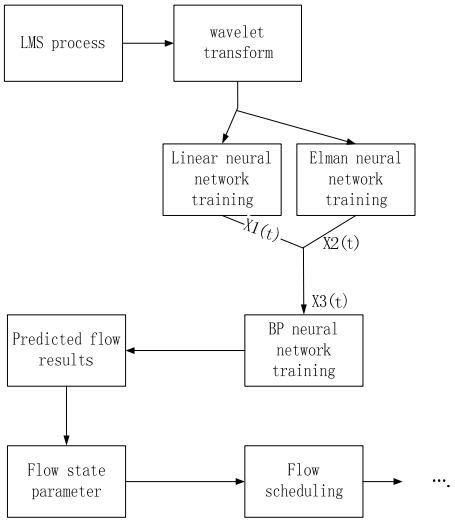

Fig. 1.Improved Wavelet Neural Network Prediction Combined Model Algorithm Scheme

The basic idea of this paper is to improve wavelet neural network prediction combined model created by integrating wavelet neural network model; improve weighted least-connection scheduling algorithm; propose NNPMA and apply it to cloud computing system architecture.

3. NNPMA ALGORITHM

3.1. *Algorithm Scheme*

The improved wavelet neural network prediction combined model algorithm scheme is illustrated in Fig 1. It can predict the upcoming network traffic from the last actual network traffic flowing through prediction model and then accordingly configure network resources.

Though weighted least-connection scheduling algorithm is a widely used load balancing algorithm with nice effects [Ron and Ariel, (2007)], it has included the factors causing imbalance of server performance into consideration and is unable to aggregate all server parameters to understand its performance. To tackle this drawback, this paper has improved this algorithm by designing an adaptive scheduling algorithm bases upon cognitive network, i.e. the NNPMA.

Cognitive network can learn network traffic status parameters by itself and understand server performance from these parameters. After fully understanding server performance, new network service requests together with server management thresholds will be brought under weight analysis, to derive new parameters about traffic flow. With these new parameters, new requests can be scheduled in such a way that new request will not be scheduled to a server with least load but low performance, thus avoiding causing new load imbalance, which is the case with least-connection scheduling algorithm.

With this thought in mind, the basic flow chart is illustrated as Fig 2:

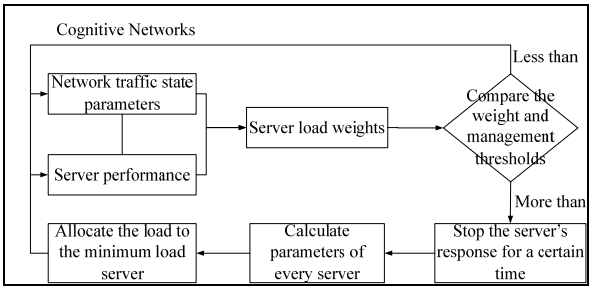

Fig. 2. Core Algorithm Basic Flow Chart

In the flow of this algorithm, the first step is to predict the upcoming network traffic according to the basic traffic; then derive the status parameters of upcoming network traffic from the traffic output predicted by traffic prediction model; after fully considering the impact patterns of various services on network traffic, a specialized network software MRTG will be used to capture the status parameters of server; calculate the weighted load value of each server, and record the largest and the least weighted value, and the traffic between paths can then be dynamically adjusted with network resource adaptive distribution approach. For service with connection established as well as newly connected, their types and resource requirements will be compared comprehensively and the comparison results will be reported to the router node at the forward path. In other words, the largest weighted value of servers will be compared with their management thresholds. Corresponding servers will stop responding for a certain period of time decided by their working state, and the latest load request will be scheduled to the server with least load. After this, these nodes will redirect those packets of existing service flows according to assessment results to enable real-time and dynamic scheduling of service flows in network.

3.2. *Algorithm Implementation*

Parameter description: There are m servers, and j ranges from 1 to 3, representing CPU load, network traffic and memory usage respectively.

$$r_i[j](1 \le i \le m, 1 \le j \le 3) \tag{1}$$

$$W_j (1 \le j \le 3) \tag{2}$$

Eq. (1) represents the CPU load, network traffic, memory usage load parameters of the ith server, while Eq. (2)represents the weighted value of server CPU load, network traffic, and memory usage. Load for each server i is Eq. (3)

$$Load[i] = \sum r_i[j] \times W_j \tag{3}$$

Proxy server, according to the load of each server, schedules the request by the scheduling algorithm to the appropriate server.

Algorithm is described as follows:

```
MaxLoadValue=0;
For (int i =1; i <= m; i++)
{
For (int j=1;j<=3;j++)
```

```
Load[i]=Load[i]+r[j]?Wj
If  (MaxLoadValue< Load[i])
{
MaxLoadValue = Load[i];
Return MinLoad[i]
}
}
If (MaxLoadValue< predetermined threshold)
Suspend server [i] for a certain time;
else
```

Obtain performance parameters of several sets of servers and calculate the weighted value.

Return the current minimum load server algorithm pseudo-code:

```
If W(Smin)>0
{for (int i=m+1;i<n;i++)
{if (W(Si)<=0)
Continue
If (Load(Si)<Load(Smin))
M=i
}
Return Smin
}
```

4. SIMULATION IMPLEMENTATION

4.1. *Create Simulation Model*

OPNET [Li and Ye, (2006)] is a universal simulation and modeling tool for communications network and facilities. Its object-oriented modeling method and graphical editor can truly reflect the actual network and structure of various network components, and the system can be mapped directly into the model. Its flexibility almost enable it to support all types of network and network technologies, and to be widely used as a decision-support tool for network to conduct detailed insight and analysis on the existing and in-design network, system or process performance and behavior.

Forming in common network structure, the network simulation model includes three parts as input, processing and output. The input generates network data using an autoregressive model based on Poisson distribution and Bernoulli distribution. The process model in router captures network parameters and server performance metrics, calculates weighted load value and outputs load scheduling instructions to egress node. As shown in Fig 3.

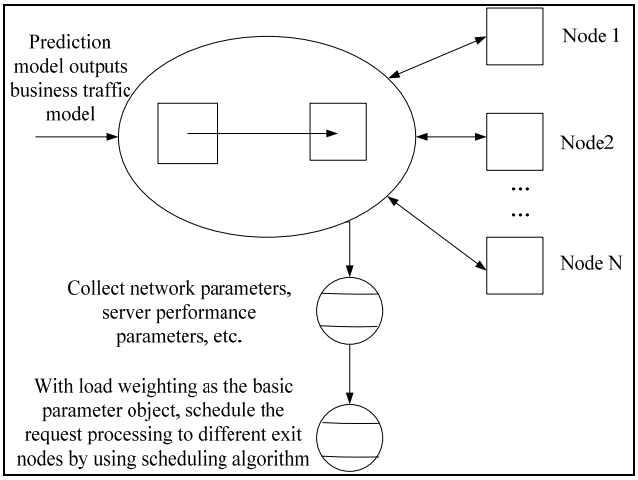

Fig. 3. Schematic Drawing of Simulation Model

The basic adaptive algorithm designed in Section 2.1 is improved and included into the simulation model to set up process model, node model and engineering scenario.

A network simulation scenario model is built, in which three clients are configured to transmit and receive data from three servers. Then the weighted least-connection scheduling algorithm is compared with improved NNPAM algorithm, where certain basic parameters are set to reflect common network data. For example, the process model is set as “Acb_fifo” (FIFO); the average interval between each arrival of packets as 1.0s; the size of data packet as 1,000 to 9,000b/p; the transmission interval as 1s, and the simulation duration as 30 min.

4.2. *Simulation Result Analysis*

When the simulation ends, results of parameters are gathered and then analyzed.

As Fig 4 Fig 5 shows, the traffic curve generated by improved prediction model well fits the actual traffic curve, enabling to predict upcoming network traffic with high accuracy. Particularly in the late phase of simulation, the traffic results of improved prediction model are even more close to actual traffic. So the network traffic generated by improved combined prediction model can be safely used to extract parameters about the upcoming network traffic.

Fig. 4. Predicting Curve Generated by Improved Prediction Model and Actual Traffic Value

Fig. 5. Predicting Curve Generated by Unimproved Prediction Model and Actual Traffic Value

The load rendering of each server is shown in Fig 5 Fig6. The horizontal axis represents 30min simulation time, and the vertical axis represents the loads of different servers.

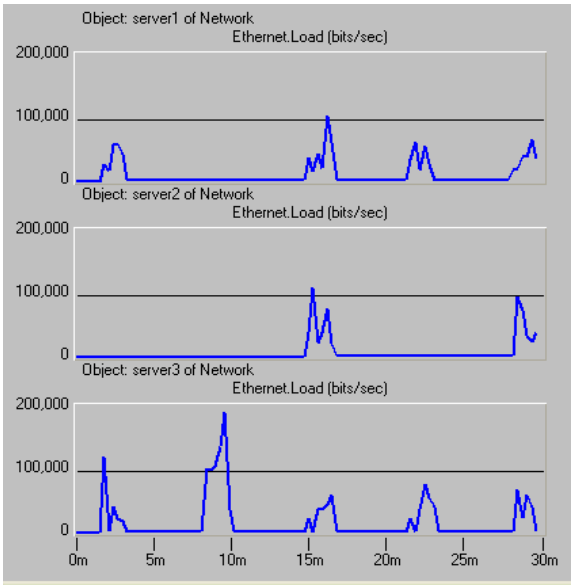

Fig. 6. Load Rendering of Minimum Traffic Scheduling Server

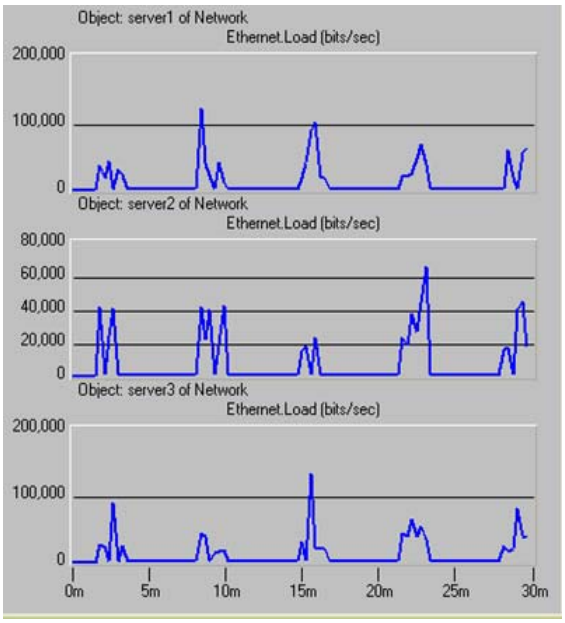

Fig. 7. Load Rendering of Adaptive Minimum Traffic Scheduling Algorithm Server

The load situations of each serve under the minimum traffic scheduling algorithm are shown in Fig 6, and Fig 7 shows the load situations of each serve under the improved adaptive minimum traffic scheduling algorithm; the loads of the three servers shown in Fig 6 are not balanced enough, and the third server takes more network requests. In Fig 7, the

loads of the three servers are relatively balanced, and they take the network requests in a relatively average manner. After using improved NNPMA algorithm, the load balancing for each server is better than using the minimum traffic scheduling algorithm.

5. Application of NNPMA Algorithm in Cloud Computing Architecture

Cloud computing platform provides flexible server resource instance for the load balancer, after significant changes of the user request access volume and adaptive adjustment of resources, conduct the adaptive adjustment to the load request scheduling, achieving the consistency between the request load adaptive scheduling and resource supply. Adaptive load balancing scheduling of the server instance resource is achieved via the appropriate deployment of LVS cluster system scheduler. Save the NNPMA algorithm in LVS scheduling algorithm module, the load status judgment and load forwarding are achieved through a variety of linked lists; the administrator calls IPVS-related functions through the configuration of management module interaction, achieving load balancing resource scheduling of LVS cluster system. The implementation of this application is shown in Fig 8.

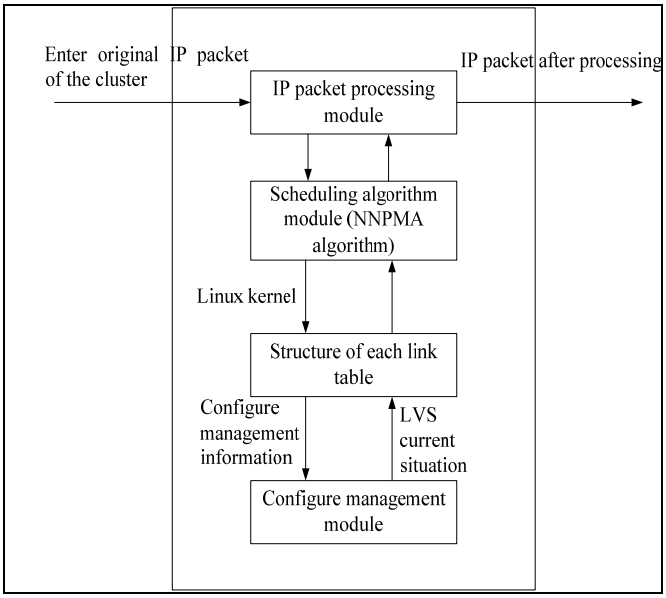

Fig. 8. Implementation of NNPMA Algorithm in Cloud Computing Application

6. Conclusions

In a context of extensive use of emerging technologies today, by applying NNPMA to cloud computing architecture and building model with simulation software OPNET to simulate the algorithm. Then compare and analyze the results of NNPMA with unimproved least-connection scheduling algorithm, it can be concluded that the dynamic and adaptive traffic scheduling algorithm can enable servers to evenly provide network services and better balance load, while minimizing impact to overall network performance, making it a more optimal solution for current cloud computing application architecture.

References

Chen Jianxun, Zhang Yong, et al. A Load Balancing Algorithm for MPLS Traffic Engineering Field. Computer and Digital Engineering, no. 6, 2006, pp. 39-41.

Chen Yijiao, Lu Xicheng, et al. A Session-Oriented Adaptive Load Balancing Algorithm. Journal of Software, no. 19, 2008, pp. 1828-1836.

FU Guowei, Liu Xinsong, et al. A Net Load Scheduling Algorithm Based on Feedback. Journal of Chengdu University of Information Technology, no. 12, 2006, pp. 779-781.

Hu Anbo, Su Jinshu, Chen Shuhui, et al. Research on Application-Level Load Balancing Technology. Computer Engineering and Applications, no. 45, 2009, pp. 84-86.

K. Gopalan, Tzi-cker Chiueh, Yow-Jian Lin, "Load balancing routing with bandwidth-delay guarantees", IEEE Communications Magazine, 2004, pp. 108-113.

Kong Dawei, Li Dandan, et al. Summarization of Load Balancing Algorithms for Network Processors. Automation Technology and Applications, no. 26, 2007, pp. 45-48.

Li Daisong, Liu Yong, et al. Study and Implementation of One Kind of Load Balancing Algorithm. Journal of Shenyang Ligong University, no. 9, 2005, pp. 47-48.

Li Xin, Ye Ming. Network modeling and simulation for OPNET Modeler. Xi'an: Xi'an University of Electronic Science and Technology Press, 2006.

Liang Benlai, Qin Yong, Chen Li, Song Fei, et al. DBCTIA Algorithm Based on Binary Object Optimization for Load Balancing of Multiple Links. Computer Applications, no. 3, 2009, pp. 655-664.

Luo Yongjun, Li xiaoyue, et al. Summarization of the Load-balancing Algorithm. Sci-Tech Information Development & Economy, no. 23, 2008, pp. 134-136.

Ron Banner and Ariel Orda, "Muttipath routing algorithms for congestion minimization," IEEE/ACM Transactions on Networking, vol. 15, no. 2, 2007, pp. 413-424.

Song Huili, et al. Research on Key Technologies of Cognitive Radio Networks. Mobile Communications, no. 4, 2008, pp. 75-77.

Wang Yue, Cai Wandong, et al. An Adaptive Dynamic Load Balancing Algorithm. Computer Engineering and Applications, no. 21, 2006, pp. 121-123.

Yan Shi, Zengji Liu, Zhiliang Qiu and Min Sheng. "Load Balance Based Network Bandwidth Allocation for Delay Sensitive Services" Proc. 19th International Conference on Advanced Information Networking and Applications (AINA'05), 2005.

Zhang Weiwen, Wu Guoxin, et al. Research on Dynamic Load Balance Algorithm Based on Structured P2P Networks. Computer Engineering and Design, no. 9, 2007, pp. 4152-4168.