A PSO-Based Overload Control Algorithm to Application Servers in Next Generation Electric Communication Networks

Honghao Zhao[1], Fanbo Meng[1,4], Qingqi Zhao[1], Weizhe Ma[2], Rimin Jiang[3], Xiaoguang Bian[3]

[1]Liaoning Electric Power Company Limited, Shenyang 110006, China
[2]Benxi Power Supply Company, Benxi 117000, China
[3]Liaoning Planning and Designing Institute of Post and Telecommunication Company Limited, Shenyang 110011, China
[4]College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
amengfb@163.com

Abstract - This paper analysis overload control characteristics of the application server in the next generation electric communication network, as well as the application server overload detection requirements, propose an energy efficiency overload control algorithm based on particle swarm. First of all, we consider the problem of overload control and network efficiency at the same time, based on rate allocation algorithm, construct the control input rate and energy efficiency model, on which the energy efficiency optimization overload control model is built. After that, the particle swarm optimization is used to solve the model, where the optimal solution is obtained by the iterative process. Finally, the simulation results show that the proposed algorithm is feasible and effective.

Index Terms - NGN, electric communication, application server, energy efficiency, overload control.

I . Introduction

Next generation electric communications is a business-driven. Two core entities are softswitch and application server, the emergence of the next generation electric communication network, and all its multimedia value-added services provide operators with fresh flood to enhance the operational capacity and the competitiveness of enterprises [1-3]. The application server provides various types of enhancing services to users in the next generation electric communication network. Current electric communication service network has features like large business diversity and high business burst, and likely causes system overload [4-6]. Ss the carrier-class equipment of the next generation electric communication network, overload control which is an important technique to guarantee normal operation of application server, has caused extensive attention.

Currently, many researches about overload control of the next generation electric communication network have been launched. Deng et al. [4] analyzed the relation between service quantity and CPU utilization ratio in parlay gateway, proposed an adaptive overload control algorithm based on ticket bank. Abdelal et al. [7] introduced Signal-Based Overload control, which was implemented at the sending servers and did not impose processing burden on overload servers. Hwang et al. [8] proposed a method to control SIP server overload based on the window-based overload control method, considering not only the maximum window size but also the number of confirmation messages. Chentouf et al. [9] monitored a set of SIP servers' features and used Support Vector Machines to classify traffic behavior. The above method can control the overload operations of the application server, but these methods are not energy efficiency.

This paper puts forward a new overload control algorithm for the next generation electric communication network application server. We construct the relationship between the control input rate and energy efficiency [10-11] using rate allocation. Regarding this relationship as objective function and overload control requirement as constraint condition, we construct the optimization model. Particle swarm algorithm [12], also known as Particle Swarm Optimization (PSO) is a new evolutionary algorithm developed in recent years. It uses fitness to evaluate the quality of the solution by following the current searched optimal value to find the global optimum. Using PSO to solve the model, we can get the optimal solution. Simulation results show that our algorithm is effective.

II . Problem Statement

According to the overload characteristics of the application server, in order to achieve the overload control target of effectiveness, fairness etc., our paper will follow the following principles: overload detection and control are based on load balancing. overload control mechanism must be easy to implement and the additional overhead brought to the application server is to be as low as possible; not only to ensure the different quality of service, at the same time normal business cannot be restricted; for different service requester, the overload mechanism is applicable. Figure 1 shows the overload control block diagram of the next generation electric communication network application server. Application server provides supports for the operation of the business, when it receives service requests from softswitch, webserver or other entities, the request is sent to the appropriate business logic to handle, and the processing results of business logic are sent back to the corresponding entity.

Assume that there are two types of business in the system: high priority and low priority, and the number is n_h and n_l, $n_h + n_l = n$. And assume that when the system is overload, the CPU utilization rates are φ_h and φ_l respectively, and meet $\varphi_h + \varphi_l = \varphi$. When application server is overload, high and low priority business meets

$$\sum_{i=1}^{n_h} n_{hi} / (T\mu_{hi}) + \sum_{i=1}^{n_l} n_{li} / (T\mu_{li}) \le \varphi \qquad (1)$$

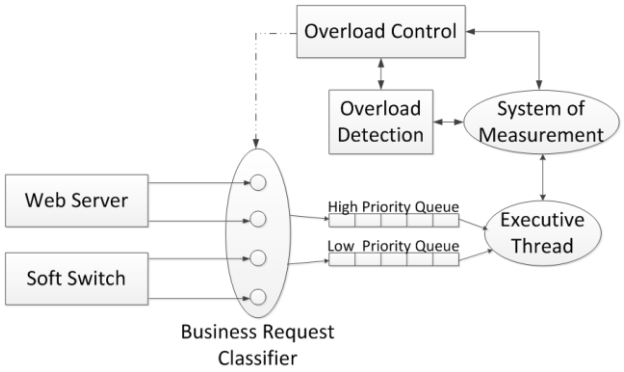

Fig. 1 Overload control model for application servers.

where n_{hi} and n_{li} refers to the amount of high and low priority business, T is the control time, μ_{hi} and μ_{li} denotes application server processing rate of high and low priority business. In addition, according to the experience of the operator, when the application server is overload, the arrival rate of the business meets a certain proportion. Set the ratio of jth high priority business is α_j ,where $0 \le \alpha_j \le 1$, $\sum \alpha_j = 1$, $1 \le j \le n_h$; the ratio of jth low priority business is β_j, where $0 \le \beta_j \le 1$, $\sum \beta_j = 1$, $1 \le j \le n_l$.

The overload essence of application server is the arrival rates of some business are too large, and the resources of the application server cannot handle the incoming business, we see the CPU resource as the bottleneck resource of system and use CPU utilization rate as a mark of system resources, when the rate exceeds φ, the system is overload. In order to control overload business powerfully, when the application server is overload, the system must determine what kind of business causes overload. In this paper, overload detection method is as follows: when $\varphi_h' \le \varphi_h$, low priority business is overload, where φ_h' means current CPU utilization; when $\varphi_h' > \varphi_h$, high priority business is overload, because of the priority enforcement of high priority business, we cannot determine whether the low priority business is overload, therefore, we also need to calculate the CPU utilization rate of low priority, if larger, then the low priority business is overload, otherwise, the low priority business is not overload.

III. Particle Swarm Optimization Algorithm

PSO algorithm is a new evolutionary algorithm developed in recent years. Mathematical representation of PSO algorithm is as follows: set dimension of search space is D, the total number of particles is m. The ith particle position is represented as vector $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$; the past optimal position of the ith particle in the history of "flying" (the position corresponding to the optimal solution) is $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$, where the gth particle is the best of all the best position $P_i (i = 1, ..., m)$; the position change rate(speed) of the ith particle is vector $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$. The position of each particle changes according to the following formula:

$$\begin{aligned} v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand \times [p_{id}(t) - x_{id}(t)] \\ + c_2 \times rand \times [p_{gd}(t) - x_{id}(t)] \end{aligned} \qquad (2)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \qquad (3)$$

where, $1 \le i \le n, 1 \le d \le D$; c_1 is the weight coefficient usually set to 2; c_2 is a weighting coefficient of the particle tracking groups optimal value, typically set to 2; $rand$ is a random number in $[0,1]$, usually set to 1; w is a factor to maintain the original speed. w in velocity update formula decreases linearly to the minimum weighted factor $w_{\min}$ from the maximum weighted factor $w_{\max}$ and it can denotes as:

$$w = w_{\max} - iter \times \frac{w_{\max} - w_{\min}}{iter_{\max}} \qquad (4)$$

IV . Energy Efficiency Overload Control Model

In order to improve the efficiency of overload control, enhance the stability of overload control, reduce resource requirement, make the application server more energy, we put forward a kind of overload control model with energy efficiency. After studying the characteristics of application server business, we propose a practical energy consumption-arrival rate model:

$$f_e(V) = \begin{cases} 0 & N = 0 \\ \lambda c_e + (1-\lambda)V & 0 < V < c_e \end{cases} \qquad (5)$$

where c_e denotes server capacity, V denotes the arrival rate, λ is used to describe independent condition of the energy consumption and the server capacity. According to Equation (5), we can draw the energy efficiency – arrival rate model:

$$E_e(V_i) = \frac{f_e(V_i)}{V_i} = E_e(V_i) = \mathrm{C}^{-1} \sum_{i \in S} \lambda c_e + (1-\lambda) V_i \qquad (6)$$

where S is a collection of overload business, C denotes the total arrival rate of overload business. By Equation (5)-(6), we obtain overload control model of energy efficiency, whose objective function can denoted as follows:

$$U = \frac{\sum_{i \in S} \lambda c_e + (1-\lambda) V_i}{C} \qquad (7)$$

Overload set S has the following three kinds of circumstances: the overload of high priority business, the overload of low priority business, and the overload of both businesses. Energy efficiency rate allocation constraints are various to different overload conditions. When high-priority is overload, CPU utilization must meet the following conditions:

$$\sum_{x \in S} \frac{V_p}{T\mu_p} \le \varphi - \sum\nolimits_y V_{ly} / \mu_{ly} \qquad (8)$$

Considering the fairness of business, all kinds of business need to meet the following conditions:

$$(1-\theta)\frac{\alpha_i}{\alpha_j} < \frac{V_i}{V_j} < (1+\theta)\frac{\alpha_i}{\alpha_j}, \theta \in (0,1) \quad (9)$$

where θ is a random number in $(0,1)$. By inequations (8)-(9), overload constraints of high priority business is as follows:

$$\begin{cases} \sum_{x \in S} \frac{V_p}{T\mu_p} \le \varphi - \sum_y V_{ly} / \mu_{ly} \\ (1-\theta)\frac{\alpha_i}{\alpha_j} < \frac{V_i}{V_j} < (1+\theta)\frac{\alpha_i}{\alpha_j}, \theta \in (0,1), V_x > 0 \end{cases} \quad (10)$$

Likewise, overload constraints of low priority business is:

$$\begin{cases} \sum_{x \in S} \frac{V_p}{T\mu_p} \le \varphi - \varphi_h^{'} \\ (1-\theta)\frac{\beta_i}{\beta_j} < \frac{V_i}{V_j} < (1+\theta)\frac{\beta_i}{\beta_j}, \theta \in (0,1), V_x > 0 \end{cases} \quad (11)$$

When both high and low priorities are overload at the same time, the constraint conditions is:

$$\begin{cases} \sum_{x \in S_L} \frac{V_{lp}}{T\mu_{lp}} \le \varphi_l, \sum_{y \in S_H} \frac{V_{hp}}{T\varphi_{hp}} \le \varphi_h \\ (1-\theta)\frac{\alpha_i}{\alpha_j} < \frac{V_{hi}}{V_{hj}} < (1+\theta)\frac{\alpha_i}{\alpha_j}, \theta \in (0,1) \\ (1-\theta)\frac{\beta_i}{\beta_j} < \frac{V_{li}}{V_{lj}} < (1+\theta)\frac{\beta_i}{\beta_j}, \theta \in (0,1), V > 0 \end{cases} \quad (12)$$

where S_H and S_L is a overload set of the high and the low.

Here, according to Equations (7)-(12), with overload control as constraints, we can establish the optimal energy efficiency model. To solve the optimal model, we propose an energy efficiency overload control algorithm based on PSO. The concrete steps of our algorithm are as follows:

Step1: Initialize time window T, and let $k = 1$.

Step2: Measure the rates of high and low priority business that enters application server in T, $\{V_{h1}, V_{h2}, ...\}$ and $\{V_{l1}, V_{l2}, ...\}$.

Step3: Ddetect whether the application server is overload. If not, let $k = k + 1$, and go back to Step 2.

Step4: If high priority business is overloaded, set high priority control target as $\varphi - \sum_i V_{li} / \mu_{li}$. If low priority business is overloaded, set the low priority control target as $\varphi - \varphi_h^{'}(k)$. If both are overloaded, set the target as φ_h and φ_l. Then, the application server limits new service requests, and handle all business in the queue in the next time window T.

Step5: Initialize the high and low priority business in accordance with the requirements of the particle swarm algorithm. Set the size of the group is N, randomly generate the matrices $x_h = \{x_{h_i j}\}$ and $x_l = \{x_{l_i j}\}$, where $i = 1, ..., N$ and $j = 1, ..., n$, and v_{ij} is the corresponding rate;

Step6: Calculate the objective function value of each particle;

Step7: For high and low priority business, calculate the best position $p_i(t) = (x_{i1}, ...x_{im})$ that particles have experienced and have the best fitness, determined by the following formula:

$$p_i(t+1) = \begin{cases} p_i(t) & f(x_1(t+1), ..., x_n(t+1)) < p_i(t)) \\ x_i(t+1) & f(x_1(t+1), ..., x_n(t+1)) \ge p_i(t)) \end{cases}$$

Calculate the best position of all the particles, namely:

$$G_i(t) = \max\{f(p_1(t)), ..., f(p_N(t))\}$$

Step8: According to the following formula, evaluate the speed and position of the particles for high and low priority services;

$$\begin{cases} v_{ij}(t+1) = w \times v_{ij}(t) + c_1 r_1 (p_i(t) - x_{ij}(t)) \\ \qquad\qquad + c_2 r_2 (G_i(t) - x_{ij}(t)) \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \end{cases}$$

where $c_{1,2}$ is acceleration constant (learning rate), $r_{1,2}$ is uniform distributed random numbers, w is inertia factor

Step9: Determine end conditions, achieve the fitness of the objective function, obtain the current solution V_{hi}^c and V_{li}^c, otherwise, return to Step 6;

Step10: Get the maximum rate of high and low priority business in the next Ts $V_{hi} = V_{hi}^c$ and $V_{li} = V_{li}^c$.

Step11: Overload control module sets high and low priority business tokens V_{hi} and V_{li} in token bank.

Step12: In the next Ts, when business arrives, if there is a corresponding business token, then the overload control module receives the service request, otherwise, refuse.

Step13: After Ts, application server determines if the business is overloaded. If overloaded, continue overload control, go back to Step 3. If not, algorithm is over and exit.

V . Simulation Result and Analysis

In this part, we verify the energy efficiency overload control algorithm proposed in this paper through simulation. In the process of simulation, the application server has high and row two queues, can deal with four kinds of business, business 1 and 3 are the high priority business of webserver and softswitch respectively, the proportion of business 1 and 3 is 1:2; business 2 and 4 are the low priority business of webserver and softswitch respectively, the proportion is 1:2, too. The treatment rate of application server for business 1,2,3,4 is 50/s, 100/s, 100/s, 50/s. When the server is overload, the CPU utilization of high and low priority is $\rho_h = 0.6$, $\rho_l = 0.2$. In the following simulation, we compare our method(abbreviated as PSO) with the existing overload control algorithm RA[13].

In the simulation, assuming business 1-4 are all overload, including 0~30s and 131~200s, the arrival rate of business 1-4 is 10/s; 31 -130s, the rate of business 1-4 is 100/s. Figure 2 shows the above four kinds of business. In Figure 2, we can see, the application server overload occurs in 31-130s. Our algorithm controls the arrival rate of business within a reasonable range, and meets certain fairness.

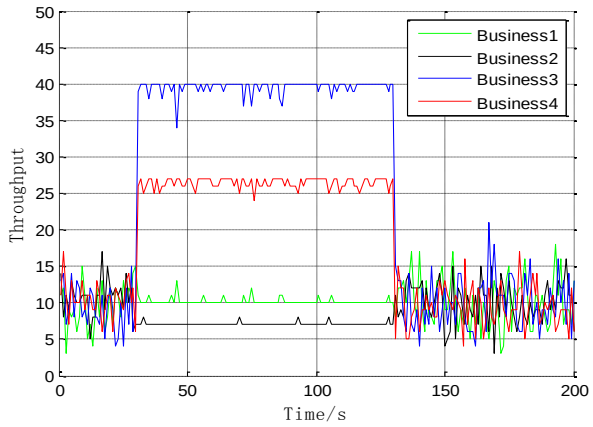

Fig. 2 Four different business

Fig. 3 shows the CPU utilization of high and low priority business under our proposed algorithm. Fig. 3 tells us, by our algorithm, high and low priority business is successfully controlled under the overload threshold of CPU utilization. As shown in Figs. 2 and 3, the total CPU utilization is maintained at about 0.8, and the high and low priority CPU utilization meet the requirements of the default. This shows that, our algorithm can effectively operate the overload control of application server, and meet certain fairness and effectiveness.

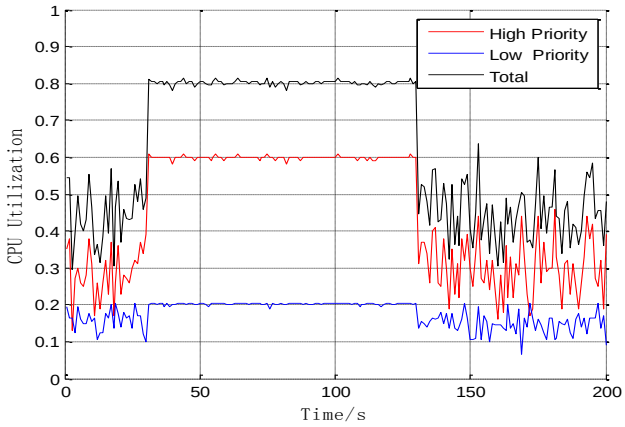

Fig. 3 CPU utilization.

Figure 4 shows the energy efficiency of network between our PSO algorithm and the existing RA algorithm. From Figure 4, we find that, compared with RA, our PSO algorithm has better energy efficiency. This indicates that our SA algorithm has better energy efficiency. More importantly, at the overload starting time(31s), RA energy efficiency has greater volatility, and our energy efficiency is relatively stable, which further indicates the robustness of our SA algorithm is better. Figure 3 and 4 show that, our PSO algorithm has succeeded in reducing the demand of application server for resources. Moreover, our energy efficiency control algorithm PSO can effectively control the overload service rate, make business rate remain at a steady state, and various business satisfy the fairness requirement basically.

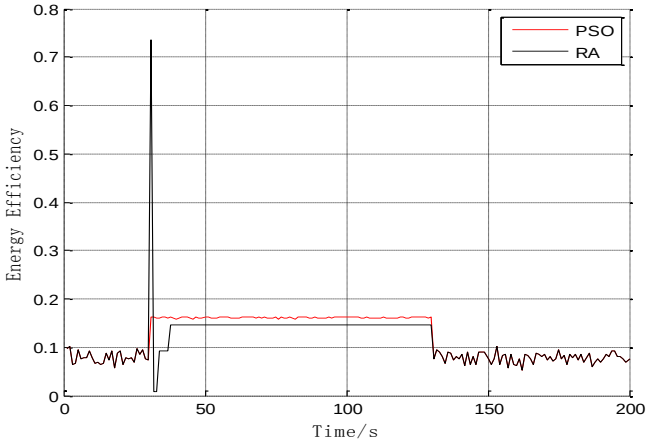

Fig. 4 Energy efficiency of network

VI . Conclusions

This paper studies the overload control method of the next generation electric communication network application server. By establishing the relationship function between the control input rate and energy efficiency, we construct the optimization model with overload control as constraint conditions. To solve the optimal model, we propose an energy efficiency overload control algorithm based on PSO. The simulation results show that our algorithm is feasible and effective.

References

[1] D. Jiang., Z. Xu, L. Nie, et al. An approximate approach to end-to-end traffic in communication networks, Chinese Journal of Electronics, 2012, 21(4): 705-710.
[2] D. Jiang, Z. Xu, et al. Joint time-frequency sparse estimation of large-scale network traffic, Computer Networks, 2011, 55(10): 3533-3547.
[3] S. Mohapatra. Integrated planning for Next Generation Networks. In Proc. of ISINMW'09, New York, Jun. 1-5, 2009, pp. 205-210.
[4] Z. Deng, X. Tan. Overload control for parlay gateway in the next generation network. In Proc. of ICITA'10, Aug. 20-22, 2010, pp. 1-4.
[5] D. Jiang, Z. Xu, H. Xu, et al. An approximation method of origin-destination flow traffic from link load counts, Computers and Electrical Engineering, Nov. 2011, 37(6): 1106-1121.
[6] D. Jiang, G. Hu. GARCH model-based large-scale IP traffic matrix estimation. IEEE Communications Letters. 2009, 13(1): 52-54.
[7] A. Abdelal, W. Matragi, et al. Signal-Based Overload Control for SIP Servers. In Proc. of CCNC'10, Las Vegas, Jan. 9-12, 2010, pp. 1-7.
[8] D. Hwang, J. Park, et al. A window-based overload control considering the number of confirmation Massages for SIP server. In Proc. of ICUFN'12, Phuket, Jul. 4-6, 2012, pp. 180-185.
[9] Z. Chentouf. SIP overload control using automatic classification. In Proc. of SIECPC'11, Riyadh, Apr. 24-26, 2011, pp. 1-6..
[10] S. Lee. A Distributed Link Management Algorithm for Energy Efficient Networks. In Proc. of (Globecom'11), Houston, Dec. 5-9, 2011, pp. 1-5.
[11] A. Cianfrani. Introducing routing standby in network nodes to improve energy savings techniques. In Proc. of ICFES'12 Madrid, May 9-11, 2012, pp. 1-7.
[12] L. Lu. An improved particle swarm optimization algorithm. In Proc. of ICGC'08, Hangzhou, Aug. 26-28, 2008, pp. 486-490.
[13] W. Wu, F. Yang, et al. The study on overload control of application server in next-generation networks. In Proc. of ICCT'03, Beijing, Apr. 9-11, 2003, pp: 1429-1432.