SRConfig: An Empirical Method of Interdependent Soft Configurations for Improving Performance in n-Tier Application

Yuliang Shi, Xudong Zhao, Shanqing Guo, Shijun Liu, Lizhen Cui*
School of Computer Science and Technology, Shandong University, Jinan 250101, China
Email: shiyuliang@sdu.edu.cn, sdu_zxd@163.com, {guoshanqing, lsj, clz}@sdu.edu.cn

***Abstract*—Efficient resources utilization and better system performance are always two important objectives that service providers pursue to enjoy a maximum profit. In this paper, through analyzing experimental measurements, we study the performance impact of interdependent soft resources on an n-tier application benchmark - the RUBiS system. Soft resources are vital factors that influence hardware resources usage and overall application performance. Improper soft configurations can result in correlated bottlenecks and make performance degradation, so tuning the configuration of soft resources is imperative. Based on the experimental measurements, SRConfig method is applied to predict the soft configurations through adopting the back propagation neural network in n-tier application. Experimental results validate the accuracy and efficacy of our method.**

Keywords- performance, interdependent, soft resources, n-tier, neural network, tune

I. INTRODUCTION

Web service-based systems, especially n-tier applications, have been widely used in the daily life. Since the explosive growth of the data, there is an increasing requirements of computing capacity and system performance, which is an urgent problem to solve. The rapid development of electronic technology have promoted the continuous improvement of hardware capability, for example, the CPU processing speed is greatly improved, and the memory capacity is significantly increased. The emergence of cloud computing technology [1] reduces the cost of hardware resources. Besides, N-tier architecture [2] has been employed in the design of web-based applications, it can greatly improve the system flexibility and scalability. Although deploying more servers to realize load balancing also can alleviate this problem, it requires high cost, and it is even not necessary due to the fact that none servers are fully utilized. Therefore, another choice is to efficiently use hardware resources.

Soft resources are important software components that use hardware resources to complete requests in the web systems [3], [4]. One of the critical soft resources is the number of threads in the servers, especially in n-tier application. It can be configured to limit the amount of concurrent requests the server can handle. Meanwhile, it also limits the utilization of hardware resources, for instance, CPU and memory. Under the condition of unlimited hardware resources, creating and running more threads can share more hardware resources and process more requests in parallel. However, it is a complex and important issue to allocate appropriate soft resources at a fixed hardware configuration for improving performance.

In this paper, we show that improving system performance and achieving efficient utilization of hardware resources require to tune soft resources of different servers. The reason is that their interdependency as running an n-tier application. Concretely, through measuring system throughput, response time and hardware resources utilization of each server, the impact of interdependent soft resources based on the RUBiS system [5] is analyzed. Using the experimental data and back propagation neural network, soft configurations can be predicted accurately.

The main contributions of this paper are twofold as follows.

- We study the performance impact with regard to different soft configurations at a fixed hardware. Using a representative n-tier application benchmark - the RUBiS system, which is composed of web server tier, application server tier and database server tier, the experiments of single soft resource of each server and multiple soft resources are conducted, and then the performance impact is analyzed.
- We adopt the back propagation neural network to predict soft configurations. Giving the input data including workload, throughput, response time, hardware usage of each server, the predicted output data, i.e., soft configurations, can be obtained. And finally the accuracy of our method is validated by comparing the real configurations and predictions.

The remainder of the paper is organized as follows. Section II introduces the related work. The experimental environment and the interdependency of soft resources are described in Section III. Section IV introduces the method of predicting soft configurations in detail through adopting the back propagation neural network. Section V presents the experimental measurements and we validate the accuracy and efficacy of our method by experimental results in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK

In software engineering, N-tier architecture is applied to the development and deployment of web applications. Through using this architecture, the functionalities of an application can be separated into different tiers, it makes the application more flexible and reusable. Therefore, many previous works have focused on the performance improvement in n-tier application.

*Lizhen Cui is the corresponding author. Email: clz@sdu.edu.cn.

978-1-5090-2628-9/16 $31.00 © 2016 IEEE
DOI 10.1109/SCC.2016.84

Soft resources, such as threads, are non-trivial factors to influence the overall system performance. Some previous works [6]-[8] showed that threads play a vital role on the internet services and have a great impact on the control of system concurrency in the request processing. However, these works mainly focus on the study of configuration parameters in single web server, and then give the approach to make the server perform well.

Wang et al. [4] showed that the similarities and differences between soft resources and hardware resources by various allocation strategies when the system approaches saturation. Our work studies the impact of soft resources from another more comprehensive aspect, which mainly focus on the performance impact analysis of soft resources through increasing single soft resource and tuning multiple soft resources of three different tiers on an n-tier application.

Many previous works [9]-[11] used queueing theory to model and study n-tier applications. In [9], the authors modeled the request process in the n-tier web applications as the closed queueing network, and then gave the algorithm of performance prediction. However, this analytical model is formulated at the foundation of many assumptions, and it is difficult to generalize since considering various conditions, such as different workload types. In this paper, we use the back propagation neural network to predict the soft resource configurations. This method is based on historical data, so it is easier to create than the queueing model.

Artificial neural network, especially back propagation neural network, has been used in various fields for prediction. In [12], the authors analyzed principles of stock prediction and gave a model to make efficient short-term prediction based on BP neural network. In [13], the authors use the BP neural network to optimize the web service selection in smart distribution grid. However, to the best of our knowledge, using the neural network to predict soft configurations in n-tier application has not been studied.

In this paper, we analyze the performance impact of interdependent soft resources with different configurations at a fixed hardware. Through adopting the back propagation neural network, an empirical method is provided to predict soft configurations in n-tier application.

III. SOFT RESOURCES IN N-TIER APPLICATION

In this section, the experimental environment is introduced, and then we describe the interdependency of soft resources in brief.

A. Experimental Environment

An n-tier application benchmark – the RUBiS system is used as the testing cluster in our experiments. The RUBiS system is implemented as three-tier architecture, which mainly consists of web server tier, application server tier and database server tier. Fig. 1 outlines the system network topology used in our experiments.

In this paper, the RUBiS system is set up by allocating dedicated physical node to each server, which has one Apache server, two Tomcat servers and one MySQL-Cluster, with respect to MySQL-Cluster, the management node and SQL node are on one physical node, the cluster has two data

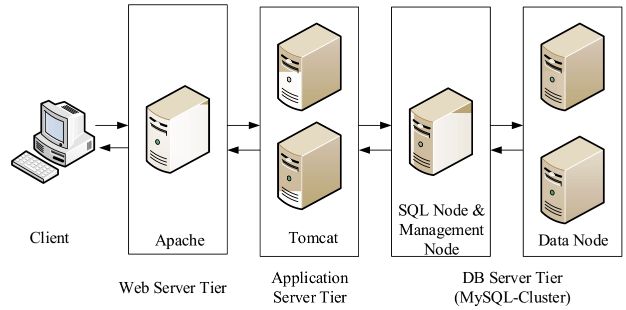


Fig. 1 System network topology

nodes, and each node is on one physical server. All the servers have the same hardware configuration. The processor of each server is Intel E7500 3GHz 64-bit and the memory is 4GB. As to the software setup, we use the Apache 2.4.3, Tomcat 8.0.9 and MySQL-Cluster 7.3.6 in the web server tier, application server tier and DB server tier, respectively.

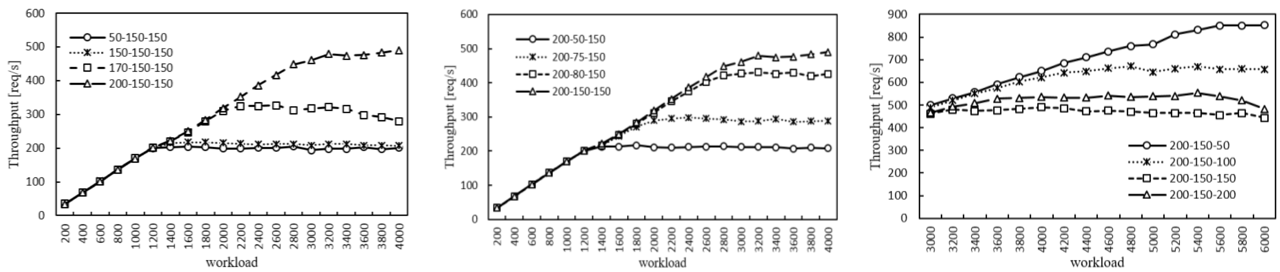
The workload of the RUBiS system consists of 26 different types of request pages, such as ViewItem, BuyNow. In our experiments, we use the workload mode of read/write interaction mixes. There are three phases in each simulation, which are a 1-minute up ramp, a 10-minute runtime and a 1-minute down ramp. Especially, we mainly analyze various experimental data of the runtime phase.

Different servers of an n-tier application play a distinct role to ensure the efficient operations and services. Web server accepts the requests originated from clients, and then distributes them to application servers, finally, carries out queries on the database servers and returns the execution results to the clients. Soft resources, especially threads of each server, play an essential role in the whole request process. So we select the configured soft resources of each tier that can control the concurrency of each server.

Because the prefork MPM [14] is used as the multi-processing module in Apache server, the MaxClients parameter is selected as soft resource we study, which can control the server concurrency by limiting processes created in the server. With respect to Tomcat server, we select the maxThreads parameter to determine the maximum requests that the server can simultaneously handle. With the similar functionality, we select the max_connections parameter in the MySQL-Cluster. We use N_A, N_T and N_D, respectively to represent the size of three parameters. However, N_T only represents one of Tomcat server's maxThreads size, and N_D only represents SQL node's max_connections size in our experiments.

B. Interdependency of Soft Resources

Soft resources in different tiers may have different impact on the performance of the application. Fig. 2 shows the throughput impact of changing three soft resources, respectively. As we can see from Fig. 2(a), when the configuration of MaxClients is too low, the system reaches bottleneck at a low workload even if the other two soft resources are enough high, increasing the MaxClients can increase the throughput, it shows that the low MaxClients can limit the requests processed in the other two tiers. Fig. 2(b) shows the maxThreads has similar functions to the



(a) Throughput of changing MaxClients (b) Throughput of changing maxThreads (c) Throughput of changing max_connections

Fig. 2 Throughput of changing three soft resources respectively

MaxClients. Fig. 2(c) shows the throughput is gradually reduced as the max_connections increase from 50 to 150. However, comparing the throughput of the *200-150-150* configuration and the *200-150-200* configuration, the *200-150-200* configuration have higher throughput. This phenomenon is related to not only its configuration, but also the dependency of the other two soft resources, which is the buffer effect and will be described in Section V.

Therefore, the system performance may be complex with the variations of soft configurations because of the interdependent soft resources. High configuration of soft resources in backend servers may not improve the system performance due to the limitation of the front tiers. However, relying on the high configuration of soft resources in front tiers, the system can have better performance even the low soft configurations in backend servers.

IV. SRConfig Method

In this section, we firstly introduce the back propagation neural network model, and then the details of the SRConfig method are described.

A. BP Neural Network Model

The back propagation neural network (BPNN) is one of the most popular neural networks. It allows the multi-layer feed forward neural network to learn and store mapping relationships between the inputs and outputs [15]. It mainly includes two phases, which are information forward propagation process and error back propagation process. The schematic diagram of BPNN model is shown in Fig. 3

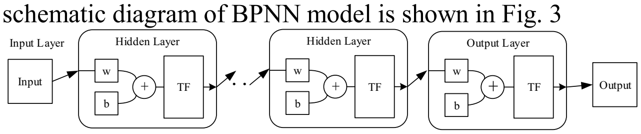


Fig. 3 The schematic diagram of BPNN model

In the information forward propagation process, the model can learn the relationship between the input data and output data.

$$t_i = f_i(\sum w_i * p_i + b_i) \quad (1)$$

In (1), p_i and t_i represent the input and output vector of the layer i , respectively. f_i is the transfer function, w_i and b_i represent the weight vector and bias.

The total error of the model E can be calculated by (2).

$$E = \frac{1}{2}\sum_j \left(T_j - O_j\right)^2 \quad (2)$$

T_j represents the jth variables of desired output value, and O_j is the jth value of output vector in output layer, which can be obtained by (1).

The weight and bias of the network layers can be changed in the error back propagation process. The expression is as follows:

$$\Delta w_{mn} = -\eta \frac{\partial E}{\partial w_{mn}} \quad (3)$$

$$\Delta b_m = -\eta \frac{\partial E}{\partial b_m} \quad (4)$$

w_{mn} represents the weight between the mth node in the current layer and the nth node in the previous layer, b_m represents the bias of the mth node in the current layer and η is the learning rate. Note that the current layer excludes the input layer, because the input layer is only the entry of the neural network.

Through adjusting the weight and bias of hidden layers and output layer, the BPNN can better learn the relationship of the training data and have the advantages in prediction.

B. Details of SRConfig Method

Due to the complex relationship between the soft resources and various measurements, the SRConfig method adopts the back propagation neural network.

In the SRConfig method, the layers number of BPNN and the neurons number of hidden layers depend on the real system architecture, i.e., the number of tiers in system determines the layers number, and the server's number in different tiers determines the proportion of the neurons.

It has been theoretically proved that the network with at least one S-type hidden layer and one linear output layer can approximate any rational function. In the method, the transfer function of hidden layers and output layer are sigmoid function and linear function, respectively.

The procedure of SRConfig method is shown as follows:

1) Defining the input data and output data: The purpose is to predict soft configurations, so the input data includes workload, maximum throughput, response time, hardware resources usage of each server, respectively. The input vector can be defined as follows.

$$input(WL, TP, RT, HR) \tag{5}$$

The output data is soft configurations SR .

2) Creating and Training neural network: The procedure is to obtain the trained neural network and the corresponding performance of network by changing the transfer function and neurons number in hidden layers.

Due to the fact that the unit and range of inputs and outputs are different, so the normalized process is necessary before creating neural network.

Assuming that X has finite values, so the elements x can be normalized to the interval $[y_{\min}, y_{\max}]$ through using (6), where y is the normalized data.

$$y = (y_{\max} - y_{\min}) \frac{x - x_{\min}}{x_{\max} - x_{\min}} + y_{\min} \tag{6}$$

Then the normalized data can be used to create and train the network based on the BPNN model. In order to evaluate the accuracy of the trained network, we directly calculate the mean square error of the predicted soft configurations rather than normalized data after the simulation.

3) Selecting suitable neural network: Through changing the transfer function and neurons number in hidden layer, we can create different neural networks, and then determine a proper BPNN by comparing the mean square error of multiple neural networks.

There are two reasons to illustrate the importance of selecting proper BPNN. On one hand, it can learn enough mapping relationships between inputs and outputs, and then accurately predict outputs. On the other hand, it can prevent over-fitting phenomenon as much as possible.

After determining the neural network with better accuracy, the soft configurations can be predicted using the trained network.

V. Experimental Measurements

In this section, we present the analysis of soft resources through changing single soft resource and tuning multiple soft resources in terms of experimental measurements, including throughput, response time and hardware resources utilization.

A. Analysis of Single Soft Resource

In this subsection, we analyze the impact of each single soft resource through changing one soft resource size while the other two soft configurations are fixed.

1) Impact of MaxClients in Apache server

Fig. 4(a) shows that the trend of response time as increasing the MaxClients configuration in Apache server. Combined with Fig. 2(a), the performance becomes better and better, i.e., the throughput increases and the response time decreases, as the MaxClients increases from 50 to 200. For example, at the *50-150-150* configuration, the system reaches the bottleneck when the workload is around 1200, i.e., the throughput starts to become stable and the response time begins to increase dramatically. However, at the *200-150-150* configuration, the system can handle larger workload and the bottleneck point is at the workload of 3200.

Fig. 4(b) shows the average CPU utilization of the Apache server. This trend shows a correspondence with throughput. The MaxClients limits the maximum processes created in the Apache server. Low configuration means that the small number of concurrent requests that the server can handle. So the server cannot use more hardware resources and improve system throughput even if most of the hardware resources are idle, i.e., it causes soft resource bottleneck. Increasing the configuration can make Apache server accept and handle more requests from clients, and then increase the hardware resources utilization and improve the performance.

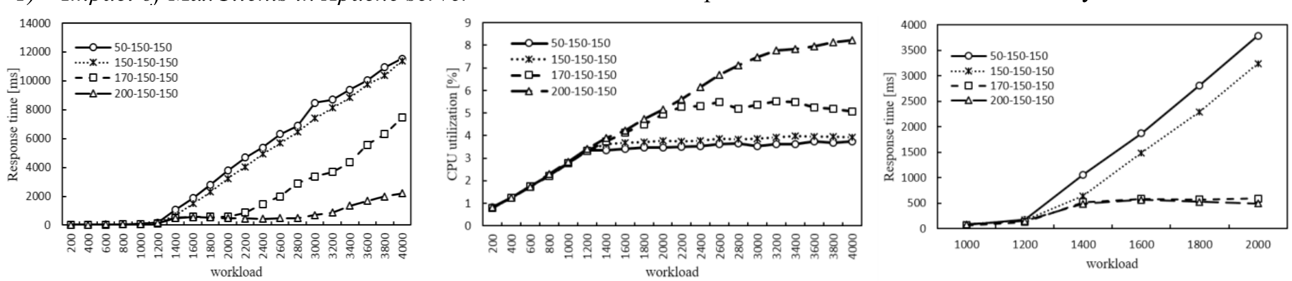
2) Impact of maxThreads in Tomcat server

The impact of increasing the maxThreads configuration in Tomcat server is described in Fig. 5. Combined with Fig. 2(b), it can be found that the tendency is similar to the impact of MaxClients in Apache. At workload 3200, the throughput of the *200-150-150* configuration is about twice of *200-50-150* configuration from the Fig. 2(b), however, the response time of the *200-150-150* configuration is only around one tenth of the *200-50-150* configuration from the Fig. 5(a). It means that low maxTheads also can result in performance degradation.

Because of the optimal load balancing among Apache and Tomcat servers by mod_jk module, the two Tomcat servers have similar trend when the maxThreads is configured equally, and it is sufficient to show the CPU utilization of one Tomcat server. The CPU utilization increases as the maxthreads ranges from 50 to 150 from Fig. 5(b). The small allowed concurrent threads make the server handle few requests, and thus cause low CPU utilization. Therefore, the maxThreads should also be increased to prevent soft resource bottleneck and improve the system performance.

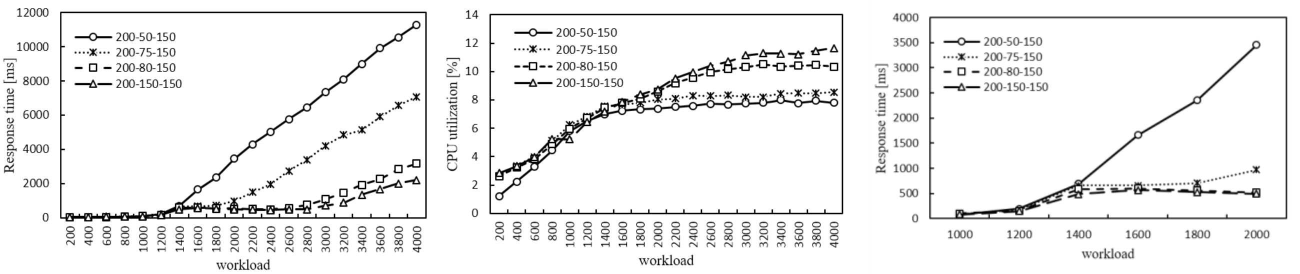
3) Impact of buffer effect in front two tiers

From Fig. 4(a) and 5(a), the response time appears a seemingly strange phenomenon. At some configurations, the response time trend remains relatively flat or even lower



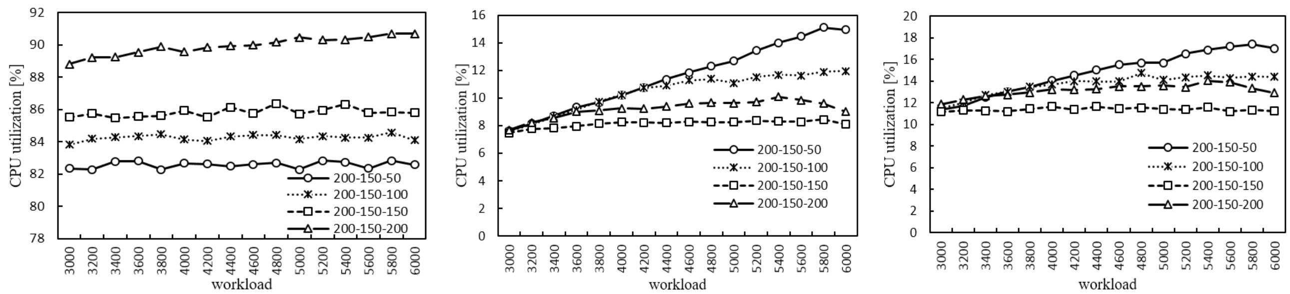
(a) Response time (b) CPU utilization of Apache server (c) Extracted Response time

Fig. 4 Impact of MaxClients in Apache Server



(a) Response time (b) CPU utilization of Tomcat server (c) Extracted Response time

Fig. 5 Impact of maxThreads in Tomcat Server



(a) CPU utilization of DB server (b) CPU utilization of Apache server (c) CPU utilization of Tomcat server

Fig. 6 Impact of max_connections in DB server

instead of continuously increasing within a certain range of workload, that is, the response time is non-monotonic growth as the workload increases. Through a careful analysis of soft configurations and the system topology, this phenomenon may depend on the buffer effect in front tiers. This effect can stabilize the process of sending requests to DB server under high workload, and then it will mitigate the rapid growth of response time and improve the performance.

Fig. 4(c) and 5(c) show the trend of response time as the workload ranges from 1000 to 2000, which is extracted from Fig. 4(a) and 5(a). At the *200-75-150* configuration, the two Tomcat can just send enough requests to DB server, however, the MaxClients size is a little high, which can accept and buffer some more requests from clients, so the response time remains stable as the workload from 1400 to 1800. Both high MaxClients and maxThreads configurations also can appear this phenomenon and achieve better overall performance, which can be explained by the *170-150-150*, the *200-150-150* and the *200-80-150* configuration. However, at the *150-150-150* configuration, the response time presents the continuously upward trend, the soft resource of Tomcat server are more than needed, but Apache can only handle 150 requests and distribute them to Tomcat. It shows only the high maxThreads configuration cannot have a buffer effect due to the limit of MaxClients.

4) Impact of max_connections in DB server

From Fig. 2(c) in Section IIIB, it can be seen that the trend of throughput as increasing the max_connections is complex.

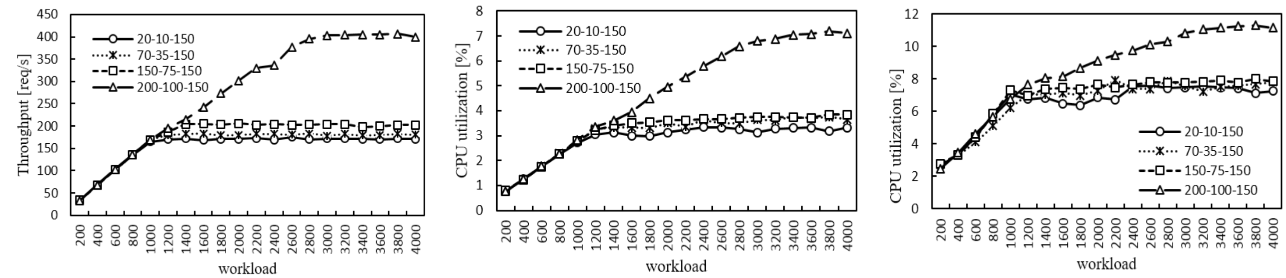
As shown in Fig. 6(a), the CPU utilization of DB server becomes nearly stable as the workload ranges from 3000 to 6000, it means the DB server has created maximum threads to execute queries. The CPU utilization increases due to the more running threads in DB server as increasing max_connections. The performance should be improved because of the higher CPU utilization, however, the buffer effect of the front two tiers should be considered. When the max_connections increases from 50 to 150, the size of MaxClients and maxThreads are enough high to have a buffer effect, which can greatly improve the performance and can be seen from Fig. 2(c). Even if the max_connections is lower, the relatively fewer requests can be quickly executed in DB server, and then the buffer of the front tiers can timely send requests to DB server, this process greatly enhance the system capacity, which can be illustrated by the increase of the CPU utilization from Fig. 6(b) and 6(c). Comparing with the *200-150-150* configuration, although the system has no buffer effect at the *200-150-200* configuration due to the same size of MaxClients and max_connections, however, high configuration allows high concurrency, it also can improve the overall throughput, besides, the CPU utilization of every server is higher. Hence, at this time, high concurrency plays a main role in the system performance.

B. Analysis of Multiple Soft Resources

In this subsection, the impact of tuning multiple soft resources is analyzed. All soft resource configurations are designed to follow a certain proportion 2:1:2, since there is no buffer effect in principle. When analyzing the impact of tuning two soft resources, the two soft resources will be changed according to the above proportion while the last soft resource is fixed. The three soft resources should be simultaneously tuned when analyzing the impact of tuning three soft resources.

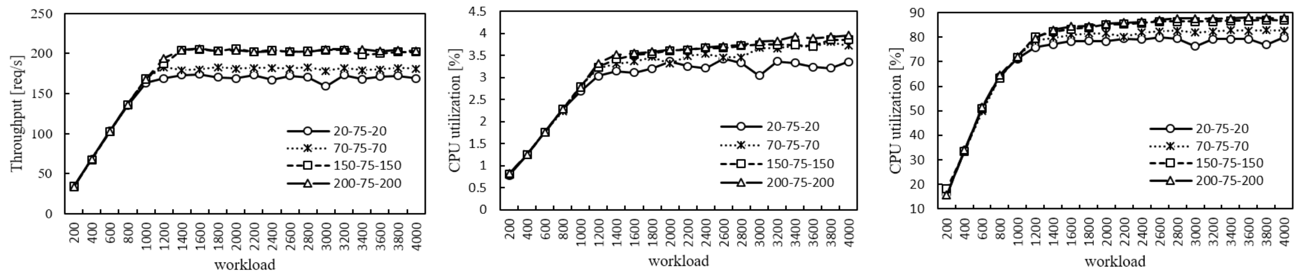
1) Impact of tuning Apache and Tomcat soft resources

Fig. 7 shows the impact of tuning MaxClients and maxThreads as the max_connections is fixed at 150. The throughput increases as increasing the two soft resource configurations from Fig. 7(a), the average CPU utilization of



(a) Throughput (b) CPU utilization of Apache server (c) CPU utilization of Tomcat server

Fig. 7 Impact of tuning MaxClients and maxThreads



(a) Throughput (b) CPU utilization of Apache server (c) CPU utilization of DBserver

Fig. 8 Impact of tuning MaxClients and max_connections

Apache and Tomcat server also increases, which can be seen from Fig. 7(b) and 7(c). Based on the previous analysis of MaxClients and maxThreads, this trend is expected. At the *200-100-150* configuration, both the system throughput and CPU utilization of Apache and Tomcat server are greatly increased, which illustrates the buffer effect plays a critical role in improving the overall system performance.

2) Impact of tuning Apache and DB soft resources

Fig. 8(a) shows the throughput of tuning the MaxClients and max_connections. As increasing the soft configurations from *20-75-20* to *150-75-150*, the throughput is an upward trend. However, the system at *200-75-200* configuration has almost the same throughput of the *150-75-150* configuration. Fig. 8(b) and 8(c) show that the average CPU utilization of Apache and DB server has a similar trend on the whole.

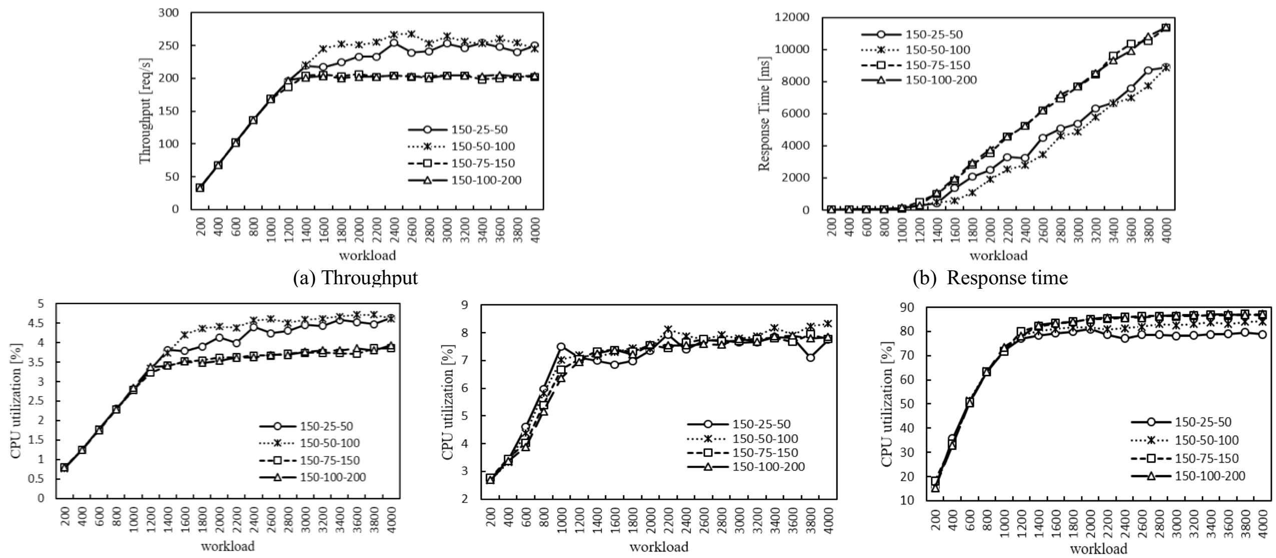
On one hand, increasing the MaxClients size allows Apache to create more threads to accept and distribute more requests to backend servers to process. On the other hand, the growth of max_connections can increase the concurrency of DB server. Therefore, the overall system concurrency is also increased, and then the CPU utilization and the system performance can be improved. Through a further analysis, at the *150-75-150* configuration, requests dispatched from Apache reach the maximum number that Tomcat server can accept, so the CPU utilization and system throughput will not be greatly increased at the *200-75-200* configuration.

3) Impact of tuning Tomcat and DB soft resources

Fig. 9(a) shows that the throughput of tuning maxThreads and max_connections configuration. The throughput rises at first and goes down latter, finally remains almost the same in our measurements as increasing the two soft configurations. At the *150-25-50* configuration and the *150-50-100* configuration, as increasing the configurations, the system concurrency is improved, and then the system has a better performance. Meanwhile, the CPU utilization of Apache server and DB server also increases, which can be seen from Fig. 9(c) and 9(e). Comparing with the *150-50-100* configuration, although Tomcat server and DB server of the 150-75-150 configuration can create more threads to handle requests, the Apache server can buffer some requests at the 150-50-100 configuration, which makes better impact on performance. It can be explained by the trend of response time in Fig. 9(b). The *150-100-200* configuration has almost the same performance and CPU utilization of the *150-75-150* configuration due to the limitation of the MaxCliens. From Fig. 9(d), the CPU utilization of Tomcat is almost the same among the four configurations on the whole. It shows that the buffer effect of Apache server can push more requests to Tomcat server to handle and increase the CPU utilization of Tomcat server.

4) Impact of tuning three soft resources

Fig. 10(a) shows that the throughput of tuning three soft resources. The throughput increases from the *400-200-400* configuration to the *800-400-800* configuration, while the system throughput at the *1000-500-1000* configuration is lower than at the *800-400-800* configuration, it means the system performance will not continuously increase as adding the configuration of three soft resources. High concurrency can improve system performance at the expense of the hardware resources. From Fig. 10(b), all configurations have almost the same and high CPU utilization of DB server in our measurements. This phenomenon may be caused by our machine's process capacity, which achieves the system hardware bottleneck at that CPU usage. However, the trend of Apache CPU utilization is similar to the throughput trend, the Apache CPU utilization of the *1000-500-1000* configuration is lower than the *800-400-800* configuration from the Fig. 10(c). It shows that the Apache can handle more requests within the same time at the *800-400-800* configuration. At the *1000-500-1000* configuration, the large



(a) Throughput (b) Response time

(c) CPU utilization of Apache server (d) CPU utilization of Tomcat server (e) CPU utilization of DB server

Fig. 9 Impact of tuning maxThreads and max_connections

(a) Throughput (b) CPU utilization of DB server (c) CPU utilization of Apache server

Fig. 10 Impact of tuning three soft resources

amount of parallel processing requests in DB server may cause the long average processing time per request, which can decrease the whole requests processing speed and degrade the overall throughput. Therefore, too high soft configurations may also cause the performance degradation.

VI. VALIDATION

In this section, the accuracy and effectiveness of SRConfig method are validated by the experimental results based on the RUBiS system and experimental data.

Based on the system topology of the RUBiS system and the details of SRConfig method, the BPNN is designed with three hidden layers, and all three transfer function are same, the neurons number proportion is 1:2:1.

According to the (5), the vector HR of input data in the method can be defined as (7).

$$\mathrm{HR}(\mathrm{HR_A},\mathrm{HR_{T1}},\mathrm{HR_{T2}},\mathrm{HR_S},\mathrm{HR_{D1}},\mathrm{HR_{D2}}) \quad (7)$$

Besides, CPU and memory are considered in hardware resources, using the Apache server as an example.

$$\mathrm{HR_A}(\mathrm{CPU_A},\mathrm{Mem_A}) \quad (8)$$

The outputs are soft configurations and described as (9).

$$\mathrm{SR}(N_A, N_T, N_D) \quad (9)$$

Thus, the inputs and outputs of the BPNN are determined, and then we can create and train the neural network.

Fig. 11 shows the mean square error (MSE) of the created BPNNs with different sigmoid transfer function and neurons number in hidden layers. Note that the MSE in that figure is calculated by the output of soft configurations. It shows that the BPNN with the tan-sigmoid function in all three hidden layers have better accuracy.

The neurons number of three hidden layers are set based on the proportion 1:2:1, so we can use the number of neurons in the first hidden layer as the x-axis. The MSE is high when the neurons number is low, because the network cannot learn and store enough mapping relationships among input data and output data, and then the MSE decreases as increasing the neurons number, finally the trend becomes stable. However, too many neurons may cause over-fitting problem.

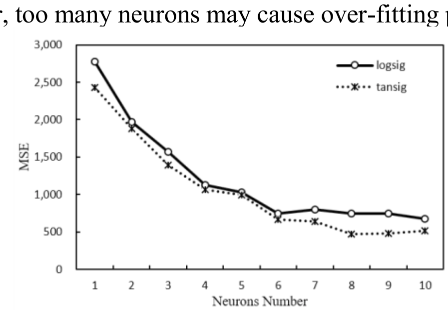


Fig. 11 MSE of different BPNNs

TABLE I. INPUT DATA OF NEURAL NETWORK

Group	**WL**	**TP (req/s)**	**RT (ms)**	**CPU_A (GHZ)**	**Mem_A (GB)**	**CPU_{T1} (GHZ)**	**Mem_{T1} (GB)**	**CPU_{T2} (GHZ)**
1	1400	204	1006	0.1023	0.3172	0.2208	0.7192	0.2199
2	4600	662	1010	0.3381	0.4232	0.4197	0.9564	0.4245
	Mem_{T2}(GB)	**CPU_S (GHZ)**	**Mem_S (GB)**	**CPU_{D1} (GHZ)**	**Mem_{D1} (GB)**	**CPU_{D2} (GHZ)**	**Mem_{D2} (GB)**	
1	0.7128	2.4648	0.7656	1.4481	3.8292	1.4313	3.7988	
2	0.9588	2.5320	0.7416	1.5219	3.8704	1.5093	3.8684	

TABLE II. RESULT COMPARISON

Group	**Real Configurations**	**Predicted output**
1	*150-75-150*	*149-71-169*
2	*200-150-100*	*197-138-102*

In order to guarantee the accuracy of the neural network, we use the tan-sigmoid as transfer function, and the neurons number of three hidden layers is set 8,16,8 in our final neural network, and the MSE is around 500.

Due to the limitation of the pages, we only select two groups of experiment to compare the real soft configurations and the predicted results obtained from the trained network. Table I is the input data of the two groups, i.e., the experimental measurements of the *150-75-150* configuration and the *200-150-100* configuration, respectively. Table II shows the comparison between predicted results of the trained network and real soft configurations. The error of the two soft configurations is so small that can be acceptable. It also validates the accuracy and effectiveness of our method.

VII. CONCLUSION

The impact of interdependent soft resources is analyzed through changing single soft resource and tuning multiple soft resources based on the n-tier application RUBiS. Improper soft configurations may greatly degrade the overall system performance. Therefore, tuning soft resources is important to improve performance and increase hardware resource usage. Based on the experimental measurements, the SRConfig method is used to predict soft configurations through adopting the BPNN model in n-tier application. This method can be potentially applied to dynamically configure soft resources according to the prediction results. Finally, the accuracy and efficacy of our method are validated by the experimental results.

ACKNOWLEDGMENT

The research work was supported by the National Natural Science Foundation of China under Grant No.61572295, 91546203, the Innovation Method Fund of China No.2015IM010200, the TaiShan Industrial Experts Programme of Shandong Province, the Natural Science Foundation of Shandong Province of China under Grant No. ZR2014FM031, ZR2013FQ014, ZR2014FM020; the Shandong Province Science and Technology Major Special Project No.2015ZDXX0201B03, 2015ZDXX0201A04, 2015ZDJQ01002, 2014CGZH1106, 2014ZZCX08102, 2014ZZCX03409; the Shandong Province Key Research and Development Plan No.2015GGX101015, 2015GGX101007, 2014GGD01063, 2015GGE27033, 2014GGX101028; the Fundamental Research Funds of Shandong University No.2015JC031.

REFERENCES

[1] Kanakala, V. RaviTeja, V. Krishna Reddy, and K. Karthik. "Performance analysis of load balancing techniques in cloud computing environment." In *International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, 2015.

[2] M. Kostoska, A. Donevski, M. Gusev, & S. Ristov. "Porting an N-tier application on cloud using P-TOSCA: A case study." in *38th International Convention on. Information and Communication Technology, Electronics and Microelectronics (MIPRO),* IEEE, 2015.

[3] J. Li, Q. Wang, C. A. Lai, J. Park, D. Yokoyama, & C. Pu. "The Impact of Software Resource Allocation on Consolidated n-Tier Applications." in *7th International Conference on Cloud Computing (CLOUD),* IEEE, 2014.

[4] Q. Wang, S. Malkowski, Y. Kanemasa, D. Jayasinghe, P. Xiong, C. Pu, M. Kawaba, and L. Harada, "The impact of soft resource allocationon n-tier application scalability," in *International Parallel Distributed Processing Symposium (IPDPS)*, IEEE, 2011.

[5] RUBiS: Rice University Bidding System http://rubis.ow2.org/.

[6] V. Beltran, J. Torres, and E. Ayguade, "Understanding tuning complexity in multithreaded and hybrid web servers," in *IEEE International Parallel and Distributed Processing Symposium*, 2008.

[7] R. von Behren, J. Condit, F. Zhou, G. C. Necula, and E. Brewer, "Capriccio: scalable threads for internet services," in SOSP '03, NewYork, USA, 2003.

[8] H. Jamjoom, C.-T. Chou, and K. Shin, "The impact of concurrency gains on the analysis and control of multi-threaded internet services," in INFOCOM '04, 2004.

[9] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An Analytical Model for Multi-Tier Internet Services and Its Applications," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, pp. 291–302, 2005.

[10] X. Liu, J. Heo, L. Sha, and X. Zhu, "Adaptive Control of MultiTiered Web Applications Using Queueing Predictor," in *10th IEEE Netw. Operations and Management Symp.*, pp. 106–114, 2006.

[11] P. Xiong, Z. Wang, S. Malkowski, Q. Wang, D. Jayasinghe, and C. Pu, "Economical and robust provisioning of n-tier cloud workloads: A multi-level control approach," in *31st Int. Conf. on Distributed Computing Systems*, pp. 571–580, IEEE, 2011.

[12] F. Li, and C. Liu. "Application study of BP neural network on stock market prediction." In *Ninth International Conference on Hybrid Intelligent Systems*, 2009.

[13] L. Rui, Y. Xiong, K. Xiao, & X. Qiu, "BP neural network-based web service selection algorithm in the smart distribution grid". In *16th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1-4, IEEE, 2014.

[14] Apache MPM prefork: http://httpd.apache.org/docs/2.4/mod/prefork.html.

[15] P. Sibi, S. A. Jones, & P. Siddarth, Analysis of different activation functions using back propagation neural networks. *Journal of Theoretical and Applied Information Technology*, 2013.