

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

A Dynamic Optimizing Adjustment Model for File Storage Distribution in Content Delivery Network

¹Huang Yongsheng, ²Tian Xiaoyu and ²Liu Yazhi

¹Tangshan Key Laboratory of Informationization Technologies and Engineering Control,
School of Management, Hebei United University, China

²School of Information Engineering, Hebei United University, China

Abstract: In Content Delivery Network (CDN), optimized the file storage distribution in the servers is a proper approach to balance the storage capacity of the servers and the cost of the file location. However, the active users and the files they require are varying with the time going. In this study, a dynamic optimizing adjustment model is proposed to adjusting the file storage distribution corresponding to the variation of the active users of the CDN. In the proposed model, the ant colony optimization algorithm with the competitive neural network algorithm involved is used to adjust the file storage distribution efficiently and effectively. The simulative experimental results testifies that the proposed model can achieve not only steadily smaller time cost of optimization procedure in consideration of the variation of the user scale but also higher hit ratio of file request.

Key words: Content delivery network, ant colony algorithm, file storage distribution, competitive neural network

INTRODUCTION

With the information sharing being widely accepted, the file storage and location in content delivery network become hot spot research fields in the computer and network sciences (Javidi, 2008; Calafate *et al.*, 2012). The strategy of the file storage distribution in the servers is determinative to the storage capacity utilization of the servers and the cost of the file location (Gao *et al.*, 2010). The real-time and time cost of file achievement is influenced by file location algorithm (Di and Wang, 2013). In content delivery network, integrating the strategy of the file storage and the approach of the file location is usually adopted (Martins and Duarte, 2010; Halinger and Hartleb, 2011). A hierarchical model is put forward in (Wu *et al.*, 2009), in which the servers are placed in two steps. It is decided which sub-cluster each server should belong to firstly by a tree-based heuristic algorithm, Load Balance Traversing (LBT). Secondly, the location of each server within a sub-cluster is determined by the Load Balance Matching (LBM) algorithm.

The files in the network are achieved by the users requiring them at lowest cost is the ultimate goal of enhancing the strategy of file storage distribution and the file location algorithm (Rossi *et al.*, 2009; Thomos *et al.*, 2011). By the requirement characteristics of the users, the CDN models with better efficiency are proposed (Peng, 2011). By predictions of individual end user behavior, a

model optimally using the network bandwidth and the storage resources is presented by Verhoeyen *et al.* (2012) with the comparison of content popularity versus new recommender. The attributes of the files are relative to the users' requirement to the files and it is an effective approach to take the attributes of the files into account in consideration of the file storage distribution and location in CDN (Costa *et al.*, 2008). To reducing the response time, organizing the CDN according to the social attributes of the users with proper file caching in the server is another efficient method (Sahagun *et al.*, 2012).

Consequently, it is a dilemma for better utilizing the storage capacity with less cost of file replicas location. Based on analysis to the characteristics of users or the requirement prediction, the past proposed solutions have decreased the cost of file location by designated file distribution strategies. However, the resource expenditure will become lager rapidly and the performances will get worse quickly with the scale of the CDN system increasing. In this study, a dynamic optimizing adjustment model is proposed to adjusting the file storage distribution corresponding to the variation of the active users of the CDN. In the proposed model, the ant colony optimization algorithm with the competitive neural network algorithm involved to enhance the performances of the CDN system with less resource expenditure especially when the scale of the CDN system becomes larger.

WORK FLOW OF THE CONTENT DELIVERY NETWORK

As shown in Fig. 1, there are a source service, an access server and a certain number of surrogate servers in the content delivery network. These servers are interconnected and construct a content service network which takes charge of the response of the file request and providing with files to the users. When a user requires a file, a connection between the user and the access server will be created and a file request will be sent from the user to the access server. The access server will then redirect the request to a surrogate server. If the request file is stored in the redirected surrogate server by cache in advance, the surrogate server can provide service directly. If the request file is not stored in the surrogate server, the surrogate server will achieve the file from the source server and then dispatch the file to the user.

ANALYSIS TO THE EFFICIENCY OF THE FILE DELIVERY IN CDN

As described earlier, a surrogate server will be selected and the service for the user will be redirected to the surrogate server when a user accesses the CDN system. Consequently, the dispatching speed from the selected surrogate server to the user and whether the selected surrogate server has store the request file in advance will influence the efficiency of the file delivery in CDN (Borst *et al.*, 2009). The distance of the physical network connection between the selected surrogate server and the user is usually determinative to the dispatching speed from the selected surrogate server to

the user with the designated access bandwidth of the user to the network. In the circumstance of congest to the selected server, the dispatching speed will decrease sharply (Adler *et al.*, 2011). As to the circumstance whether the selected surrogate server has store the request file in advance, the request file should be pushed to the surrogate server firstly and then starting the dispatching is practicable if the selected surrogate server has not stored the request file in advance.

Based on the analysis above, the efficiency optimization to the file delivery in CDN is designing feasible model by which it is more possible that the selected surrogate server for every user has stored the user's request file in advance and the distance between the selected surrogate server and the user is near as far as possible (Foo *et al.*, 2011). As a result, the optimization is related to the information about the file stored in the surrogate servers and the connections between the users and the selected surrogates for the users with the network loading in consideration.

Both insufficient entire network bandwidth and improper management can result in the appearance of network congestion. When the network congestion can not be avoided by feasible equilibration strategy, the performance of the CDN should be enhanced by the adjustment of the network bandwidth. While the entire network bandwidth satisfies the requirement, appropriate optimization will improve the efficiency of the file delivery. As the problem of insufficient entire network bandwidth can not be thoroughly solved by the optimization, the proposed model in the study is not in consideration of the circumstance.

Combining the two relative factors above, in the optimization model, it is as far as possible that the

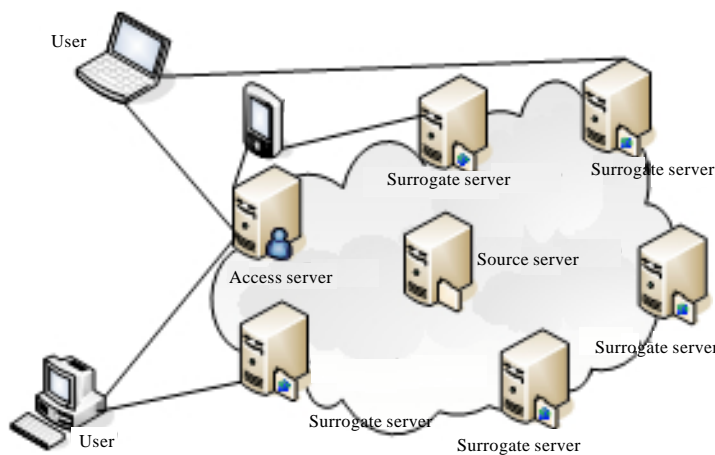


Fig. 1: Construction of the content delivery network

selected surrogate server for every user has stored the request file in advance and has available network bandwidth to dispatch the file meanwhile. In order to assure the request file having been stored in the selected surrogate server in advance, the file requirement of users should be achieved beforehand. In the following, the prediction approach to the location the user accessing the CDN and the files the use will require is addressed.

PREDICTION TO THE ACTIVITIES RELATED TO CDN OF THE USER

In reality, precise prediction to the activity about CDN is impossible. However, occurrence of the activity having been predicted occurs more probably, the prediction is more utilizable.

For the user of the CDN, the locations of the user accessing the CDN usually appear with regularity. For example, a user accesses the CDN with a fixed location or specially designated locations. In other words, the nearest surrogates being selected for a user will be also specially designated with regularity and therefore, the selected surrogate for a user is predictable.

The probability of a user's requirement to every file is usually differentiated sharply. The file request of a user is often relative to social background. For example, the file request of a college student is apparently different to that of a researcher in the same college. In the common circumstances, the categories (categorized according to their attributes) of the requiring files of a user are relative to the categories of the social attributes of the user. A file has been required by a user means the category the file belongs to is relative to the user and then it is much probable that other files of the category will be required by the user. On the other hand, two user has required a file means they has identical attribute which is relative to the category the file belongs to and then it is much probable the files one user of them has required will be required by the other. The relationship can be used to the prediction to the file request of the user. In the prediction approach, the prediction to the file request of a user is based on the historical file request records by which the relations talked about above can be extracted.

To enhance the utilizable of the relationship, the quantitative prediction should be proposed. In order to describe the relationship, $F = \{f_1, f_2, \dots, f_n\}$ is put forward to denote the set of files the CDN can provide and $T = \{t_1, t_2, \dots, t_k\}$ is used to denote the set of categories the files can be divided into. As the set of users of CDN is expressed by $U = \{u_1, u_2, \dots, u_n\}$, then the relation between a user u_i and a file category t_j can be defined by the following formula:

$$r_{u_i}(u_i, t_j) = \frac{n_{t_j}}{n_{u_i}} \tag{1}$$

In the formula above, $r_{u_i}(u_i, t_j)$ denote the quantitative relation between u_i and t_j . n_{u_i} expresses the number of files which has been required by u_i before. N_{t_j} represents the number of files which belong to t_j and has been required by u_i .

Accordingly, the relationship between the file categories t_i and t_j can be expressed by the formula as following:

$$r_{u_i}(t_x, t_y) = \frac{\sum_{i=1}^h r_{u_i}(u_i, t_x) r_{u_i}(u_i, t_y)}{\sqrt{\sum_{i=1}^h r_{u_i}^2(u_i, t_x)} \sqrt{\sum_{i=1}^h r_{u_i}^2(u_i, t_y)}} \tag{2}$$

If a user u_i has required the file f_i , it is certain that the user u_i has requirement to the file f_j . To describe the prediction uniformly, the requiring probability of the user u_i to the file f_j is designated to a fixed value 1. If a user has not required the file f_j , the probability of a user u_i requiring the file f_j in the future can be predicted by the formula below:

$$p(u_i, f_j) = \lambda_1 \frac{\sum_{t \in T(f_j)} r_{u_i}(u_i, t_x)}{|T(f_j)|} + \lambda_2 \frac{\sum_{t \in \overline{T}(f_j)} r_{u_i}(u_i, t_y)}{|\overline{T}(f_j)|} \tag{3}$$

The relation between the user and the file category is combined with the relation between categories in the prediction above. In the formula, λ_1 and λ_2 is the weight factors of the relations and $\lambda_1 + \lambda_2 = 1$. $T(f_j)$ is the set of categories which f_j belongs to. $\overline{T}(f_j)$ is the set of categories which f_j does not belong to. Consequently, the requiring prediction of a user u_i to the file f_j consists of the relation between u_i and f_j by the categories which f_j belongs to and which f_j does not belong to.

If the files a user will require are more probable to be stored in the surrogate selected to serve for the user, the user will achieve the files more quickly. In the following, the description to the file stored in the surrogate servers will be put forward.

DESCRIPTION TO THE FILE STORED IN THE SURROGATE SERVERS

In a designated period of time, the surrogate servers and the files stored in the surrogate servers are stable in the content distribution network. The set of the surrogate servers can be expressed by $S = \{s_1, s_2, \dots, s_m\}$ which contains m surrogate servers. Correspondingly, the set

of the files stored in the surrogate can be denoted as $F = \{f_1, f_2, \dots, f_n\}$ which consists of n files. In the two sets, the subscript of every element is the order number of a file/surrogate server. The order number can also be taken as the ID of the file/surrogate server. The correlation between a file and a surrogate server can be represented by the following formula:

$$c_{ij} = \text{cor}(f_i, s_j) = \begin{cases} 0, & \text{the file } f_i \text{ is not stored in the surrogate server } s_j \\ 1, & \text{the file } f_i \text{ is stored in the surrogate server } s_j \end{cases} \quad (4)$$

As a result, the correlation between all files and all surrogate servers forms a matrix $C_{n \times m}$ as follows:

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{bmatrix}$$

In the CDN system, the storage capacity of a surrogate server is usually restricted. To comfort the requirement of expression, the storage capacity of a surrogate server is defined by the indication $SC(s_i)$ which denotes the storage capacity of the surrogate server s_i . Consequently, the feasible scheme of stored files into surrogate servers should base on the following constraint:

$$\sum_{j=1}^n (SQ(f_j) \times c_{ji}) \leq SC(s_i) \quad 1 \leq i \leq m \quad (5)$$

In the formula above, $SQ(f_j)$ is the storage capacity requirement of f_j . $SC(s_i)$ is storage capacity of the surrogate server s_i .

FILE STORAGE DISTRIBUTION ADJUSTMENT MODEL

To optimize the file storage distribution, the ant colony optimization (ACO) (Yang and Zhuang, 2009) is cited to adjust the file distribution in the surrogate servers. In the adjustment model, the bidirectional routes between the elements in the matrix $C_{n \times m}$ are created. If the value of r_{xy}^i is 0, the bidirectional route are built between the elements c_{ij} and c_{xy} . The value of r_{xy}^i is defined according to the rules in the following formula:

$$r_{xy}^i = \begin{cases} 1, & |i-x|=1 \text{ or } |j-y|=1 \\ 0, & \text{elsewise} \end{cases} \quad (6)$$

As described above, the value of r_{xy}^i is equal to that of r_{yx}^i which means there is a bidirectional route between c_{ij} and c_{xy} . For sake of connecting the file distribution with

the users which serve by selected surrogate servers, the heuristic function of ACO is defined as formula 7:

$$\pi_{ij} = \frac{1}{\sum_{u_k \in U(s_i)} p(u_k, f_j)} \quad (7)$$

In the formula, $U(s_i)$ is the set of users which the surrogate server s_i is designated to serve to.

In ACO, an ant staying at a node can walk to another node if there is route between the nodes and the movement is permitted. In the adjustment model, c_{ij} ($1 \leq j \leq m$) is permitted to walk to only when the inequality in the formula 5 is tenable. For the description of the adjustment model in the following, the elements which an ant staying at c_{ij} is permitted to walk to are expressed by the set:

$$\text{Allowed}_{ij} = \{c_{xy} | c_{xy} \text{ is permitted to walk to from } c_{ij}\}$$

In the procedure of ACO, a group of ants are used to implement the optimization. The set $A = \{a_1, a_2, \dots, a_q\}$ is put forward to denote the ant group. Generally, an ant is not permitted to walk to an element which it has walked to in the past. Let the set $C(a_k)$ of elements which an ant a_k has walk to and then the ant a_k staying at c_{ij} is permitted to walk to the elements in the following set:

$$\text{Allowed}(a_k) = \text{allowed}_{ij} - \text{allowed}_{ij} \cap C(a_k)$$

Let $PT = [0, D]$ express the time period of an optimization procedure and the start time and the end time are denoted as 0 and D . When ants walk at the elements of the matrix $C_{n \times m}$, the number of the pheromones ph_{ij} is put forward to the number of ants which has walked to c_{ij} . The ph_{ij} will be varied continuously from the start time to the end time and $ph_{ij}(pt)$ is the ph_{ij} at the time pt ($0 \leq pt \leq D$). The probability an ant staying at the element c_{xy} will walk to c_{ij} can be expressed by the formula 8:

$$p_{xy}^{ij}(pt) = \begin{cases} \frac{[ph_{ij}(pt)]^\alpha [\pi_{ij}]^\beta}{\sum_{c_{xy} \in \text{Allowed}(a_k)} [ph_{wx}(pt)]^\alpha [\pi_{wx}]^\beta}, & c_{ij} \in \text{Allowed}(a_k) \\ 0, & \text{elsewise} \end{cases} \quad (8)$$

In the formula above, α is heuristic factor about pheromones. The value of α is bigger comparatively and the $ph_{ij}(pt)$ is more important and the elements which have achieved higher pheromones will be more probable to be selected to walk to. In other words, the value of α is bigger means the ants are better to collaborate with each other. β is the heuristic factor about the heuristic function and represents the comparative importance of heuristic information (Sadeghzadeh *et al.*, 2011).

In the period of the optimization procedure, the fresh information about the pheromones is more important to optimize continuously. As a result, the outdated information about the pheromones should be weakened with time going. In the adjustment model, $ph_{ij}(pt)$ is adjusted according to the rule in the formula 9:

$$ph_{ij}(pt+\Delta) = (1-\theta)ph_{ij}(pt) + \theta ph_{ij}(\theta pt) \quad (9)$$

$\Delta ph_{ij}(\Delta pt)$ is the increase value of ph_{ij} during the period from the time pt to the time $pt+\Delta pt$. Accompanying with the variation of the $ph_{ij}(pt)$, the value of the element c_{ij} should be adjusted to according to the updated information. The following is the rule for the element adjustment:

$$c_{ij} = \begin{cases} 1, & ph_{ij} > H_{ph} \text{ and } c_{ij} = 0 \\ 0, & ph_{ij} < L_{ph} \text{ and } c_{ij} = 1 \end{cases} \quad (10)$$

Corresponding to the description above, the number of the pheromones for every element of the matrix $C_{n \times m}$ will vary continuously with designated number of ants walking according to the rules designed above. With the variation of the number of the pheromones for every element, the value of the elements will be adjusted and the adjustment of the elements will reversely act on the walking of the ants.

DYNAMIC OPTIMIZATION ACCORDING TO THE VARIATION OF THE ACTIVE USERS

The CDN system is always running generally, but not all of the users are active from beginning to end. Consequently, the set of the active users in the running CDN is the subset of all users and the elements are varied continuously. The file storage distribution in surrogate server should conform to the dynamic variation of the active users. Combining with the ACO, the dynamic optimization in consideration with the variation of the active users is necessary in order to enhance the efficiency of the file delivery.

As the computation complexity of the ACO algorithm, it is inefficient that the ACO algorithm is executed directly with the variation of the active users. Using the historical optimization information in consideration of the continuous updating can enhance the efficiency of real-time optimization. In the running period of the CDN system, each optimized matrix $C_{n \times m}$ (which is the optimized file storage distribution in the surrogate

servers) and the corresponding set of active users are recorded by a pair denoted as (US, OC) . In the pair, oc is the optimized matrix $C_{n \times m}$ and correspondingly us is a vector used to represent the active users in the set consists of all users. The vector US is expressed as $US = \{us_1, us_2, \dots, us_n\}$ and the elements are corresponding to the users by order. The value of the us_i is defined by the formula 11:

$$us_i = \begin{cases} 1, & \text{the user } u_i \text{ is active} \\ 0, & \text{the user } u_i \text{ is not active} \end{cases} \quad (11)$$

The similarity between every two vectors used to represent the active users can be calculated by the formula 12:

$$\text{sim}(US_i, US_j) = \frac{|US_i \bullet (US_j)^T|}{|US_i| |US_j|} \quad (12)$$

The more similar two vectors used to represent the active users is, the corresponding optimized matrix $C_{n \times m}$ will be approximate. Consequently, as US is taken as the information of the active users for achieving the optimized matrix $C_{n \times m}$, the optimization procedure will be simplified by finding a pair (US_j, OC_j) in which US_j is the most similar with US_i in the available pairs. In order to implement the pair selection, the competitive neural network algorithm is put forward. The pair selection procedure is expressed by Fig. 2 (Nie and Cao, 2009).

Initially, there is no available pair. According to the scale of the users, a positive integer S is selected for the competitive network and S pairs are initialized. In each pair, the initial value of the oc is set to Null. With respect

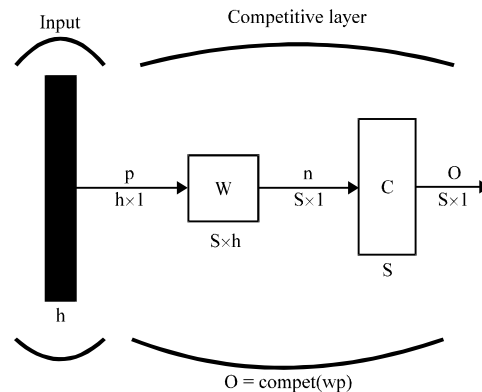


Fig. 2: Pair selection by the competitive neural network algorithm

```

Procedure. 1 Initialization
Create the matrix IMh×h;
Selecting S row vectors from IMh×h;
Initialize the S pairs;
Initialize the matrix W;
Initialize the file storage distribution in the surrogate servers;
Procedure. 2 Optimization
// it will keep waiting until the variation appears
Monitor the variation of the active users;
For
If the active users is varied
Calculate the US corresponding to the active users
Achieve the order number of the row vector most similar to US;
Achieve the pair and the prototype vector with the same order number;
Achieve the OC of the pair;
If OC is NULL
Run the larger scale ACO and achieving the updated Cn×m;
Adjust the file storage according to the updated Cn×m;
Update the pair with US and the updated Cn×m;
Update the row vector with order number above in W by US
Else
Run the smaller scale ACO and achieving the updated Cn×m;
Adjust the file storage according to the updated Cn×m;
Calculate the similarity between the prototype vector and US;
Calculate the similarity between the prototype vector and US in the pair;
If the former is smaller than the latter
Update the pair with US and the updated Cn×m;
Update the row vector with order number above in W by US
End If
End If
// it will keep waiting until the variation appears
Monitor the variation of the active users;
End For
    
```

Fig. 3: Pseudo code for the dynamic optimization procedure

to the values of the US in the pairs, S row vectors of the matrix IM_{h×h}(shown as follows) are randomly selected in consideration of the diversification of the active users. Each of the selected row vector is taken as a US of one pair. The S row vectors and the corresponding pairs are given order numbers for 1 to S by order. The S row vectors are called as prototype vectors:

$$IM_{h \times h} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

In the competitive neural network, W is a matrix which is constructed by the S row vectors selected from the matrix above initially (Meyer-Base and Thummler, 2008). A h×h vector US is taken as the input of the competitive network. By the computation of the competitive layer, A S×1 vector is outputted and only one component of the vector is 1 with the values the other being 0. If the order number of the component with the value of 1 in the outputted vector is f, then the fth row

vector of W is the most similar one in the row vectors of the matrix W. The dynamic optimization procedure can be addressed by the following Fig. 3 (Fang *et al.*, 2010).

SIMULATIVE EXPERIMENTS AND EXPERIMENTAL ANALYSIS

According to the requirement of the proposed model, simulative software for efficiency and effectiveness of the proposed model is designed and implemented. Utilizing the software, designated, the activities of the surrogate servers, the users and the source servers along with the dynamic optimization algorithm are simulated.

In order to demonstrate the efficiency of the proposed model, a CDN system with 500 user and 5 surrogate servers are simulated by the simulative software. From the time the system initializing, an optimization procedure triggered by the variation of the active users is selected every a proper period of time. The time cost of the optimization procedures is shown by the Fig. 4 in order of time. The simulative experiments indicate that the time cost of per optimization procedure is larger and unstable at beginning. With the time going, the cost



Fig. 4: Time cost of per optimization procedure by order

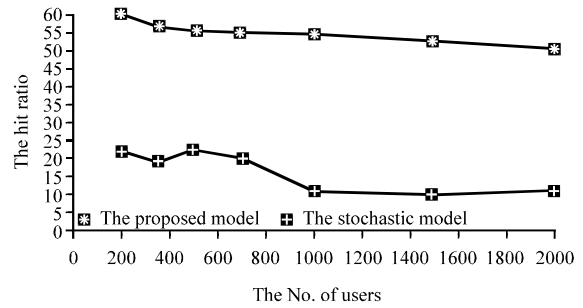


Fig. 6: Contrast of the hit ratio between the proposed model and the stochastic model

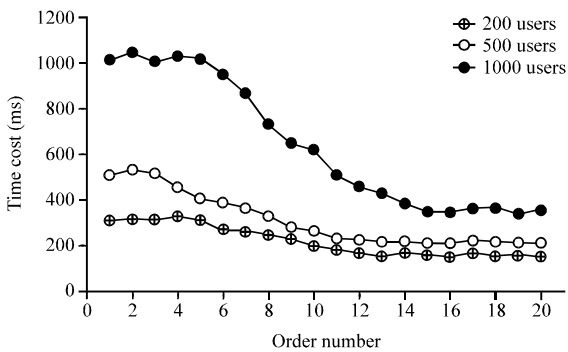


Fig. 5: Contrast of the time cost among the systems with different scale of users

Table 1: Parameters in the simulative experiments for testifying the effectiveness

No. of users	No. of surrogate servers
200	5
350	5
500	5
700	5
1000	10
1500	10
2000	10

For sake of demonstrating the effectiveness of the proposed model, the simulative CDN systems with the parameters listed by Table 1 are utilized.

When a user sends a file request to the CDN, the file is found in the surrogate servers designated to serve to the user is called a hit and on the contrary, the file is not hit if it is not found in the surrogate servers designated to serve to the user. The hit ratio is the most important parameter to the effectiveness of the proposed model. In contrast to the simulative systems about the proposed model with the configurations listed in Table 1, the simulative systems without the proposed model involved (named as the stochastic model) and with the same considerations listed in Table 1 are implemented. The simulative results shown in Fig. 6 indicate the variation of the hit ratio according to the variation of the number of users and the surrogate servers accompanying with the contrast of the hit ratio between the proposed model and the stochastic model. The hit ratio of the system with the proposed model involved is apparently higher than that of the system with the stochastic model involved as the number of users and the surrogate servers is identical. The hit ratio of the system with the proposed model involved is almost not lessened with the increasing of the number of users and the surrogate servers and contrastively, the hit ratio of the system with the stochastic model involved is sharply lessened with the number of the surrogate servers increasing. Similarly, the proposed model achieves better effectiveness in the optimization and the effectiveness is more obvious when the number of the surrogate servers is bigger.

of per optimization procedure reduces and will be comparatively stable at a smaller time cost value. The simulative results mean the optimization efficiency is unstable when the competitive neural network has not achieved proper weight matrix. With more proper weight matrix of the competitive neural network being achieving, the optimization efficiency will be enhance and the stable better efficiency will be acquired when the competitive neural network enter in stable state.

The time cost of per optimization procedure is influenced by the scale of the users in the CDN system. The contrast of the time cost among the systems with 200, 500 and 1000 users is demonstrated by the simulative results shown in the Fig. 5.

The simulative experiments indicate that the more users the CDN system has the time cost of per optimization procedure will be larger and more unstable at the starting of the system. With the time going, the time cost of per optimization procedure will be smaller and stable unless the user scale of the system is small or large. Meanwhile, the increase of the time cost of per optimization procedure is comparatively small accompanying with the raise of the user scale when the CDN system becomes stable after running a period of time.

CONCLUSION

In order to optimize the file storage distribution in the surrogate servers, a dynamic optimizing adjustment model is proposed in which the ant colony optimization algorithm is selected to implement the file storage distribution with the competitive neural network algorithm involved for enhancing the efficiency of the optimization. The simulative experimental results testifies that the proposed model can achieve not only steadily smaller time cost of optimization procedure in consideration of the variation of the user scale but also higher hit ratio of file request.

ACKNOWLEDGMENT

This study was funded by the Science and Technology Department of Tangshan City (No.12140201A-7).

REFERENCES

- Adler, M., R.K. Sitaraman and H. Venkataramani, 2011. Algorithms for optimizing the bandwidth cost of content delivery. *Computer Networks*, 55: 4007-4020.
- Borst, S., V. Gupta and A. Walid, 2009. Self-organizing algorithms for cache cooperation in content distribution networks. *Bell. Labs. Technical. J.*, 14: 113-125.
- Calafate, C.T., G. Fortino, S. Fritsch, J. Monteiro and J.C. Cano *et al.*, 2012. An efficient and robust content delivery solution for IEEE 802.11p vehicular environments. *J. Network. Comput. Applic.*, 35: 753-762.
- Costa, F., L. Silva, G. Fedak and I. Kelley, 2008. Optimizing data distribution in desktop grid platforms. *Parall. Proces. Lett.*, 18: 391-410.
- Di, S. and C.L. Wang, 2013. Dynamic optimization of multiattribute resource allocation in self-organizing clouds. *Trans. Paral. Distrib. Syst.*, 24: 464-478.
- Fang, Y., M.A. Cohen and T.G. Kincaid, 2010. Dynamic analysis of a general class of winner-take-all competitive neural networks. *Trans. Neural Networks*, 21: 771-783.
- Foo, B., D.S. Turaga, O. Verscheure, M. van der Schaar and L. Amini, 2011. Configuring trees of classifiers in distributed multimedia stream mining systems. *IEEE Trans. Circuits Syst. Video Technol.*, 21: 245-258.
- Gao, Q., L. Fei, J. Zhang and X. Peng, 2010. Performance optimisation of a medium access control protocol with multiple contention slots in multiple-input multiple-output ad hoc networks. *IET. Communic.*, 4: 562-572.
- Halinger, G. and F. Hartleb, 2011. Content delivery and caching from a network provider's perspective. *Computer Networks*, 55: 3991-4006.
- Javidi, T., 2008. Cooperative and non-cooperative resource sharing in networks: A delay perspective. *Trans. Automatic Contr.*, 53: 2134-2142.
- Martins, J.L. and S. Duarte, 2010. Routing algorithms for content-based publish/subscribe systems. *J. IEEE Communi. Surv. Tutorials*, 12: 39-58.
- Meyer-Base, A. and V. Thummler, 2008. Local and global stability analysis of an unsupervised competitive neural network. *IEEE Trans. Neural Networks*, 19: 346-251.
- Nie, X. and J. Cao, 2009. Multistability of competitive neural networks with time-varying and distributed delays. *Nonlinear Anal. Real World Applic.*, 10: 928-942.
- Peng, Z., 2011. Implementation of optimal pacing scheme in xinjiang's oil and gas pipeline leak monitoring network. *J. Networks*, 6: 54-61.
- Rossi, M., N. Bui and M. Zorzi, 2009. Cost-and collision-minimizing forwarding schemes for wireless sensor networks: Design, analysis and experimental validation. *Trans. Mobile Comput.*, 8: 322-337.
- Sadeghzadeh, M., M. Teshnehlab and K. Badie, 2011. Feature selection using combine of genetic algorithm and Ant Colony Optimization. *Adv. Intellig. Soft Comput.*, 75: 127-135.
- Sahagun, R.L., S.Q. Ren, H.S. Beng and K.M.M. Aung, 2012. Development of intelligent network storage system with adaptive decision-making. *Int. J. Adv. Comput. Technol.*, 4: 122-131.
- Thomos, N., J. Chakareski and P. Frossard, 2011. Prioritized distributed video delivery with randomized network coding. *Trans. Multimedia*, 13: 776-787.
- Verhoeyen, M., J.D. Vriendt and D.D. Vleeschauer, 2012. Optimizing for video storage networking with recommender systems. *Bell. Labs. Technic. J.*, 16: 97-113.
- Wu, J.J., Y.F. Lin, D.W. Wang and C.M. Wang, 2009. Optimizing server placement for parallel I/O in switch-based clusters. *J. Parall. Distribut. Comput.*, 69: 266-281.
- Yang, J. and Y. Zhuang, 2009. An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem. *Applied Soft Comput. J.*, 10: 653-660.