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

Huang Yongsheng, Tian Xiaoyu and Liu Yazhi
Tangshan Key Laboratory of Informationization Technologies and Engineering Control,
School of Management, Hebei United University, China
School 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 (Marins 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 larger 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 and 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 (Berst 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
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 user 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 users who 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, \( F = \{f_1, f_2, \ldots, f_k\} \) is put forward to denote the set of files the CDN can provide and \( T = \{t_1, t_2, \ldots, t_k\} \) is used to denote the set of categories the files can be divided into. As the set of users of CDN is expressed by \( U = \{u_1, u_2, \ldots, u_n\} \), then the relation between a user \( u_i \) and a file category \( t_j \) can be defined by the following formula:

\[
\tau_{u_i, t_j} = \frac{n_{u_i, t_j}}{n_{u_i}} 
\]

(1)

In the formula above, \( \tau_{u_i, t_j} \) denote the quantitative relation between \( u_i \) and \( t_j \). \( n_{u_i, t_j} \) expresses the number of files which has been required by \( u_i \) before. \( N_i \) 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:

\[
\tau_{t_i, t_j} = \frac{\sum_{u_i} \sum_{t_j} \tau_{u_i, t_j} n_{u_i, t_j}}{\sqrt{\sum_{u_i} \sum_{t_j} \tau_{u_i, t_j}^2 n_{u_i}}} \sqrt{\sum_{u_i} \sum_{t_j} \tau_{u_i, t_j}^2 n_{u_i}} 
\]

(2)

If a user \( u_i \) has required the file \( f_j \), 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:

\[
p(u_{n+1}, f_j) = \frac{\sum_{u_i \in U} \tau_{u_i, f_j} + \sum_{f_i \in F} \tau_{f_i, f_j} \sum_{u_i \in U} \tau_{u_i, f_i}}{|U|} 
\]

(3)

The relation between the user and the file category is combined with the relation between categories in the prediction above. In the formula, \( \lambda_1 \) and \( \lambda_2 \) is the weight factors of the relations and \( \lambda_1 + \lambda_2 = 1 \). \( T(f) \) is the set of categories which \( f \) belongs to. \( \tau(f) \) is the set of categories which \( f \) 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 \( f \) belongs to and which \( f \) 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 \( S = \{s_1, s_2, \ldots, s_m\} \) which contains in surrogate servers. Correspondingly, the set
of the files stored in the surrogate can be denoted as \( F = \{f_1, f_2, ..., f_n\} \) 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:

\[
c_i = \text{cor}(f_i, s_j) = \begin{cases} 
0, & \text{the file } f_i \text{ is not stored in the surrogate server } s_j \\
1, & \text{the file } f_i \text{ is stored in the surrogate server } s_j
\end{cases}
\]  

(4)

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

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

In the CDN system, the storage capacity of a surrogate server is usually restricted. To conform 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:

\[
\sum_{i=1}^{m} (\text{SQ}(f_i) \times c_i) \leq SC(s_i) \quad 1 \leq i \leq m
\]

(5)

In the formula above, \( \text{SQ}(f_i) \) is the storage capacity requirement of \( f_i \), \( SC(s_i) \) 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 \( c_{xy} \) is 0, the bidirectional route are built between the elements \( c_{xy} \) and \( c_{yx} \). The value of \( c_{xy} \) is defined according to the rules in the following formula:

\[
c_{xy} = \begin{cases} 
1, & |i - x| = 1 \text{ or } |j - y| = 1 \\
0, & \text{otherwise}
\end{cases}
\]  

(6)

As described above, the value of \( c_{xy} \) is equal to that of \( c_{yx} \) which means there is a bidirectional route between \( c_{xy} \) and \( c_{yx} \). 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:

\[
n_i = \frac{1}{\sum_{s \in U(s_i)} \rho(u_i, s)}
\]

(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:

\[
\text{Allowed}_{j} = \{c_{ij} | c_{ij} \text{ is permitted to walk to from } c_{ij}\}
\]

In the procedure of ACO, a group of ants is used to implement the optimization. The set \( A = \{a_1, a_2, ..., a_p\} \) 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:

\[
\text{Allowed}_{j} \setminus \text{Allowed}_{j} \cap C(a_k)
\]

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 \( \rho_{ij} \) is put forward to the number of ants which has walked to \( c_{ij} \). The \( \rho_{ij} \) will be varied continuously from the start time to the end time and \( \rho_{ij}(pt) \) is the \( \rho_{ij} \) at the time \( pt \) (0 \( \leq pt \leq D \)). The probability an ant staying at the element \( c_{ij} \) can be expressed by the formula 8:

\[
r_{ij}(pt) = \begin{cases} 
\frac{\rho_{ij}(pt) \cdot n_{ij}^{\alpha}}{\sum_{c \in \text{Allowed}_{j} \setminus \text{Allowed}_{j} \cap C(a_k)} \rho_{c}(pt) \cdot n_{c}^{\alpha}}, & c_{ij} \in \text{Allowed}_{j} \setminus \text{Allowed}_{j} \cap C(a_k) \\
0, & \text{otherwise}
\end{cases}
\]

(8)

In the formula above, \( \alpha \) is heuristic factor about pheromones. The value of \( \alpha \) is bigger comparatively and the \( \rho_{ij}(pt) \) 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 \( \alpha \) is bigger means the ants are better to collaborate with each other. \( \beta \) is the heuristic factor about the heuristic function and represents the comparative importance of heuristic information (Sadeghizadeh 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, \( p_h(t) \) is adjusted according to the rule in the formula 9:

\[
 p_h(t + \Delta t) = (1 - \theta) p_h(t) + \theta p_h(0)p(t)
\]  

(9)

\( \Delta p_h(\Delta t) \) is the increase value of \( p_h \) during the period from the time \( t \) to the time \( t + \Delta t \). Accompanying with the variation of the \( p_h(0)p(t) \), the value of the element \( c_i \) should be adjusted to according to the updated information. The following is the rule for the element adjustment:

\[
 c_i = \begin{cases} 
 1, & p_h > H_\epsilon \\ 0, & p_h < H_\epsilon 
\end{cases} 
\]  

(10)

Corresponding to the description above, the number of the pheromones for every element of the matrix \( C_{w,m} \) 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 \( C_{w,m} \) (which is the optimized file storage distribution in the surrogate server) and the corresponding set of active users are recorded by a pair denoted as \((US, OC)\). In the pair, \( OC \) is the optimized matrix \( C_{w,m} \) 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 \( US = \{US_1, US_2, \ldots, US_n\} \) and the elements are corresponding to the users by order. The value of the \( US_i \) is defined by the formula 11:

\[
 US_i = \begin{cases} 
 1, & \text{the user}_i \text{ is active} \\ 0, & \text{the user}_i \text{ is not active} 
\end{cases}
\]  

(11)

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

\[
 \text{sim}(US, US') = \frac{US \cdot (US')^T}{||US|| ||US'||}
\]  

(12)

The more similar two vectors used to represent the active users is, the corresponding optimized matrix \( C_{w,m} \) will be approximate. Consequently, as US is taken as the information of the active users for achieving the optimized matrix \( C_{w,m} \), the optimization procedure will be simplified by finding a pair \((US_i, OC)\) in which US, is the most similar with US 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](image-url)
Procedure 1 Initialization
Create the matrix IM_{0,W};
Select the S row vectors from IM_{0,W};
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 C_{US};
Adjust the file storage according to the updated C_{US};
Update the pair with US and the updated C_{US};
Update the row vector with order number above in W by US;
Else
Run the smaller scale ACO and achieving the updated C_{US};
Adjust the file storage according to the updated C_{US};
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 C_{US};
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_{0,W} (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:

\[
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\]

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
of per optimization procedure reduces and will be comparatively stable at a smaller time cost value. The simulation 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 achieved, 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 simulation results shown in the Fig. 5.

The simulation 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.

<table>
<thead>
<tr>
<th>No. of users</th>
<th>No. of surrogate servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>5</td>
</tr>
<tr>
<td>350</td>
<td>5</td>
</tr>
<tr>
<td>500</td>
<td>5</td>
</tr>
<tr>
<td>750</td>
<td>5</td>
</tr>
<tr>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>1500</td>
<td>10</td>
</tr>
<tr>
<td>2000</td>
<td>10</td>
</tr>
</tbody>
</table>

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.
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