Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN EFFICIENT REPLICA CREATION SCHEME IN P2P NETWORK

[1]ZHITAO GUAN, [2]YUE XU
[1, 2] School of Control and Computer Engineering, North China Electric Power University, Beijing
E-mail: [1]guan@ncepu.edu.cn, [2]xuyue12@ncepu.edu.cn

ABSTRACT

There's one major problem in the unstructured p2p file sharing systems, which is their heavy network traffic. A potential solution is to use replication technology. The objective of replication is to create enough replicas to let nodes in p2p systems get required files with only a few hops. Replication is widely used in the distributed database systems and gains great success. However, processing replication in unstructured p2p network is very challenging because a p2p system is a dynamic and decentralized system. In this paper we present an efficient hierarchical replica creation scheme based on the track and popularity of files, which includes two strategies, one is super node layer replica creation and the other is bottom layer replica balancing. Instead of passively accepting replicas, each node determines file replication by dynamically adapting to popularity and query track of the files, which can not only supply enough replicas to decrease the query messages, but also avoid unnecessary file replications. The evaluation results indicate that our method shows good performance.

Keywords: *Peer-to-peer (P2P); popularity; replication; replica creation*

1. INTRODUCTION

According to the topology structure, P2P system can be grouped as structured P2P system and unstructured P2P system. Both have their own advantages and disadvantages, and the unstructured P2P system spreads more widely because of its low maintenance cost and high availability. However, there're two major problems of unstructured p2p systems, one is their heavy network traffic which is caused by flooding query mechanism, and the system scalability were decreased; the other problem is high data access latency due to replicas located in distributed nodes. To solve the above problems, data replication techniques were proposed. When one replica is requested by a user, the request will be sent to nearest node that has the requested replica. By this method, the access latency can be decreased and data access efficiency will be increased. Meanwhile, load imbalance can also be avoided, and the reliability and scalability of the P2P systems can be promoted.

Section 2 presents the related work. Section 3 gives an introduciotn on unstructured p2p network. In section 4, we propose our top-k query model and algorithms. Section 5 shows the simulation results. Section 6 gives a conclusion to the whole paper.

2. RELATED WORK

Reference[1] proposed OR(Owner Replication) algorithm and PR(Path Replication) algorithm to cope with the problem of data replication. OR algorithm focused on the factor of the requester, when the query was finished successfully, the replica would be replicated on request node. While PR algorithm would make replicas on all nodes between the request node and response node. The main shortcoming of the two algorithms is that they don't take the replica popularity into consideration. While, the frequency of the data access in network is in accordance with power-law distribution. That is, a few files were very popular and were accessed very frequently, while most files were rarely accessed. In OR and PR schemes, for the popular files, the bandwidth consumption and query response time would be increased due to the insufficient number of replicas.

Reference[2] use a simple statistical model to derive this relationship between the storage capacity (at each peer), the number of videos, the number of peers and the resultant off-loading of video server bandwidth. The authors propose and analyze a generic replication algorithm RLB which balances the service to all movies, for both deterministic and random demand models, and both homogeneous and heterogeneous peers.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Reference[3] studis some data replication techniques for P2P collaborative systems. The authors identify several contexts and use cases where data replication can greatly support collaboration. And then consider as a case study replication techniques for dynamic documents in the context of a peer-group based P2P system of super-peer architecture. To meet the requirements for high availability and system reliability for P2P collaborative systems, they propose a replication system for documents structured as XML files to address the dynamics of the documents at peers and use the super-peer to ensure a satisfactory level of document consistency among peers.

SADDSR replica creation strategy was proposed in reference[4]. Firstly, the concept of storage alliance was defined, and then the double layer dynamic replica creation strategy based on storage alliance was proposed. SADDSR strategy could get good feedback facing to the data grid application in enterprise. However, it's difficult to apply it to loosely structured and highly dynamic P2P network.

In addition, there were also some research work [5-10] on replica creation strategy to increase the file availability and the system efficiency.

3. UNSTRUCTURED P2P NETWORK

Gnutella is one of the most popular applications of the unstructured P2P file sharing and in this subsection we will take gnutella for instance to illustrate the topology and routing strategy of the unstructured P2P network at present. The Gnutella topology has two layers as shown in Figure 1. According to the network's connection condition, nodes gather into a domain and the capable and stable node is selected from the leaf nodes as the center of the domain (the super node) and it will be the server in the domain. The super nodes interconnect to form the backbone network of the system. In Figure 1 the image on the left shows the overall logical structure of the system as a hierarchical structure and the super nodes corresponded by each domain form the super node layer. The image on the right represents the structure of the backbone layer as a fully distributed structure and its topology can be represented by the random graph.

According to their roles Node in the system can be divided into two categories:

(1)Super Node

According to the bandwidth capacity, processing ability and on-line frequency, the super node is selected from the leaf nodes. Super node has the function of child area server and is responsible for the maintenance of index information and search request of leaf nodes which are connected with it and is in charge of processing the queries of the leaf nodes. So the area which the super node is responsible for is named as super node domain and also it is referred to as the domain.

(2)Leaf Node

It is the weak node which has poor stability in the scope of super-node. Leaf node can connect several super nodes together and can belong to different domains, preventing the super-nodes being single points to lose efficacy.

The routing-query is operating only in the super node layer. When leaf node is querying the routing, firstly the query is submitted to the super node. Super node is querying the routing in the super node layer according to certain strategies. In the end, the super node collects the results and returns to the leaf nodes. According to their own preferences leaf nodes choose the connection in the results and download the document.

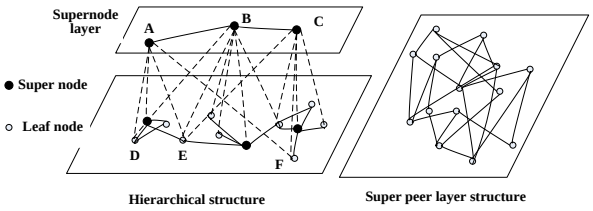

Fig 1 Two-layer topology of gnutella

Stutzbach's research [8] indicates the average connectivity of gnutella's nodes is about 30. Because of the substantial increase of the nodes' connectivity, either flooding query or dynamic query [9] can lead to more redundant messages, reducing the efficiency and scalability of the entire P2P network. Creating replica for documents is an effective way to improve the network's performance. It is necessary to propose a new and effective replica creation scheme to adapt to today's P2P network with dense network characteristics.

4. HIERARCHICAL REPLICA CREATION STRATEGY PPSR

Replica creation scheme is how to determine the time and location to create a replica. The rules of the policy's cost evaluation must take into account the physical characteristics, such as network load, the efficiency of storage nodes, network conditions,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and the size of the data replica and so on and combine the features of user access to decide the time and location of replica creation. In order to control replica creation better, PPSR is divided into two sub-strategies: super node layer replica creation strategy and super node intra-domain balancing strategy. Super node layer replica creation strategy mainly considers the replica replication among the super nodes. The second layer intra-domain balancing strategy mainly considers the reasonable distribution of the data in the domain which the super nodes belong to.

PPSR has two main characteristics as follows.

1. To reduce network traffic and query response time, and to improve query satisfaction. In the network file access frequency shows the power-law behavior. When the replica is created for the file, the file's popularity is given full consideration and create more replicas for the file with high popularity to reduce network traffic and prevent the waste of storage. At the same time, P2P network has small world characteristics [8], and the nodes with similar interests often cluster clusters. PPSR can ensure that a replica tries to be created in or closer to the range of the node cluster interested in the file so as to further reduce network traffic and improve query response time.

2. To guarantee the effective use of the node's storage. The available storage of nodes is limited, and obviously it is not feasible to create a sufficient number of replicas for each file, so the right strategy is needed to guarantee a balance between the efficiency of the system and the storage cost. The PPSR can effectively ensure this balance. In the super node layer, using the index on the way unnecessary replica creation can be reduced; In the bottom layer balancing strategy can ensure the rationalization of the number of replicas, and then to make the storage be effectively utilized.

4.1 Popularity of the Resources

This section gives several definitions and calculation methods related to the popularity of resources' replicas.

Defination 1. single-point popularity.

File resources F_j ' access frequency on the node p_i is a_{ij} and we call a_{ij} as the single-point popularity of file resource F_j on the leaf nodes p_i .If there is no resource F_j on the node p_i , $a_{ij} = 0$.

Defination 2. Domain popularity matrix.

Let L is the node collection in the domain of the super node P.

$L = \{ p_i \mid 1 \le i \le m, m$ is the number of the nodes in the P's domin$\}$

. $DP = \begin{pmatrix} a_{11}, a_{12}, ...a_{1r} \\ ..... \\ a_{m1}, a_{m2}, ...a_{mr} \end{pmatrix}$ is super node p's domain popularity matrix, where r is the total number of domain resources. In order to fully utilize the storage, if DP is a sparse matrix that 0 elements is more, then it is compressed to store. Triples table processing method is used in this article. that is only storing the non-zero elements of the sparse matrix, and at the same time taking note of its location (x, y) , stored as (a_{xy}, x, y) .

Defination 3. Domain popularity A.

Let A_{ij} represents total access frequency of file resource F_j which is under the jurisdiction of super node P_i .We call A_{ij} as F_j 's domain popularity in P_i 's domain. $A_{ij} = \sum_{i=1}^{m} a_{ij}$ is the sum of single-point popularity of F_j on each leaf node in the P. That is, the sum of J-th column element in the corresponding domain popularity Matrix.

Defination 4. Local popularity M.

Let M_{ij} represents the estimated value of the popularity of super node P_i for file resources F_j in the whole P2P network. That is the estimated value obtained by the perspective of the node P_i . Given

$M_{ij} = \alpha A_{ij} + (1-\alpha) \sum_{peer_k \subset neighbors(i)} A_{kj}$, Among it α is a constant and $0 \le \alpha \le 1$.

For a wayward node, $\alpha = 1$ can be set. At this time this node only believes its own interaction history. At the same time due to its limited contacts, thus less choice can be taken. For a node that is non-assertive or do not have the resources, $\alpha = 0$ can be given, At this time the node completely relies on the opinions of its neighbor nodes to determine the value. In General , $\alpha = 0.5$ can be set up.

Based on the above definitions and calculation methods of the file popularity, the specific policy of PPSR's replica creation will be given below.

Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2 Super Node Layer Replica Creation Strategy SLR

SLR includes the following steps.

Step 1. The super nodes p initiates queries to the domain node x and requests file F, as described in section 2.1.Super node p is responsible for super node layer for queries and after a successful query, node q is made become one of the nodes in response to the query (that is, it has a target file F);

Step 2. The replica of F is created on super node p. We can get the local popularity M_{qF} of F on the node q and assign $M_{pF} = M_{qF}$ .Super node P will regularly exchange of the information such as the new replica and its popularity with the neighbors, and in accordance with the formula (1) update the value of M_{pF} .

Step 3. Get the local popularity M_{qF} of F on the node q. IF MQF is greater than the threshold value σ , then we can create a replica of the index of F in node m which is in the path between node p and node q. and the selected scale of the node m is the distance from the query node p to node m named $dist(p,m)$. The calculation is given by the formula (2).

$$dist(p,m) = \begin{cases} dist(p,q) \times M_p \,/\, (M_p + M_q) & , M_p \neq 0 \\ dist(p,q) \,/\, 2 & , M_p = 0 \end{cases} \quad (2)$$

$dist(p,q)$ is the distance between the node p and node q, and that is hops between two nodes. M_p and M_q are the local popularities of F on the node p and node q.

Step 4. The node m is access to local query routing record. If the node F does not have a corresponding record, Then add a record in the local query routing record (F, counter) and set counter is 1 and the "counter" is a counter. Record numbers of inquiries of the node F through the node m ; If the node m has a query routing record of F, then its "counter" plus 1. If the counter value is higher than the threshold value δ (for example $\delta = 10$), then on the node m create a replica of F. The value δ can be the experience given or regulation in the simulation.

Figure 2 shows the replica creation process in the super node layer.

SLR strategy combines the strengths of the OR and PR strategy. Firstly, after the success of query, similar with OR, create a replica on the super-node in the query node where domain. Then, similar with PR strategy, create a replica of the query path. Differently, PR ,on each node along the path ,create a replica of a document, but SLR strategy only choose a reasonable node along the way to create a replica of the index. when the number of queries in the replica which through the node is over a certain threshold value, then create a replica of a document. In this way, both to increase the effective number of replicas of, and also to avoid wasting a lot of bandwidth and storage.

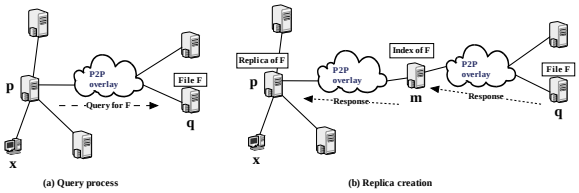

Fig 2 Illustration of replica creation in super node layer

4.3 Domain balance strategy IDB

IDB strategy has two major functions:

1.Make the data in the domain get reasonable distribution where super node in it. However, the response time of the entire P2P system can be influenced to some extent because of the data within the domain frequently copied generated by the data distribution unreasonable in the same domain.

2.Reducing the load of access to a replica of the existing domain data and improving the load balancing all need to be derived a domain replica to achieve the goals that build the enough replicas. The super node is used to be a domain task scheduler, which can distribute the access request of other domain node Replica to the related the replica of the host.

IDB strategy includes the following sub-strategies.

Strategy1. Domain replica placement mechanism.

After query successfully, create a replica of the target file F in the super node which is in the domain of queried node(leaf node or super node). If the queried node is a leaf node , you need to create a replica of F in this leaf nodes. Delete some data if there is not enough free space in the leaf node when you replica. To minimize the impact of delete operation on the system performance when ,an algorithm of deleting files based on the gain factor BF (Benefit Factor) is proposed.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The BF calculation method such as the formula (3):

$$BF_i = \omega_1 \cdot M_i + \omega_2 \cdot CT_i^{-1} + \omega_3 \cdot (size_i - size_F)^{-1} \quad (3)$$

M_i is local popularity of file I; CT_i isthe file's creation time; $size_i$ and $size_F$ Respectively are the size of the document i and the file F. $\omega_i (i = 1,2,3)$ is weight adjustment factor. The core idea of the GetRemovedFiles algorithm is,when there is not enough space to create a replica ,deleting files with low BF value to provide sufficient space for the replica creation. The pseudo-code of the algorithm is as follows:

Algorithm GetRemovedFiles(Node p, File newReplica)

Input: files[] //Files set in node p

Output: removedFiles[] // Files to be removed

```
1.  Compute UF of each object in files [] according to formula (3);
2.  Sort files[] by UF value in ascending order;
3.  int freeSpace = 0, counter=0;
4.  while(freeSpace < size of newReplica) {
5.      Add files[i] to removedFiles[];
6.      freeSpace = freeSpace + files[counter].size;
7.      counter++;
8.  }
9.  return removedFiles;
```

Figure 3. Algorithm GetRemovedFiles

Strategy 2. Replica management mechanism in the domain

The definition is the management of super node to a leaf node information. Through the replica, the load balance can be improved and the consistency of replica can be maintained.

1. Node information management mechanism. Leaf nodes need submit their own information of storage space to the super node periodically, so that the super node can put this as the reference while scheduling. In order to reduce the communication traffic, only the 3-tuple will be sent, including leaf node ID, shared space size and the used space size.

2. Load balancing mechanism. If a replica file is accessed frequently, and the access frequency exceeds a preset threshold, then the replication operation will be triggered. According to the leaf nodes information, the replicas will be created in other nodes by the super node. If the space is not enough for creating the replica, the algorithm GetRemovedFiles will be invoked.

3. Consistency maintenance mechanism of replicas. If the file is read-only, then the consistency problem is not exist. If the file is writable, the other replicas must be updated to maintain the consistency of replicas when a file is updated. Here the version vector is used to ensure the consistency of all replicas. For the page reason, here the other content is not be mentioned.

5. PERFORMANCE EVALUATION

5.1 Environment setup

P2P file sharing system is selected as simulation scenario and uses Gnutella's dynamic query method [9] as the underlying query protocol. Super node layer topology containing 10,240 super nodes is organized according to the structure of today's Gnute lla [8]and the distribution of connectivity between super nodes is in line with a power-law distribution, As shown in Figure 4. The super node layer topology generates by BRITE [11]. Leaf nodes' number of each super node is a random number in the range of [20,30]. System contains certain categories of documents and each type of documents is identified by a keyword. When the initialization, each type of documents has the same number of replicas which are randomly distributed evenly on each node.The simulation program is written by java and query simulation carries on by run (run).In each run a node is randomly selected to initiate query.

The parameters and initial settings of the simulation platform are shown in Table 1. The description of the parameters and symbolic representation of the simulation are included.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

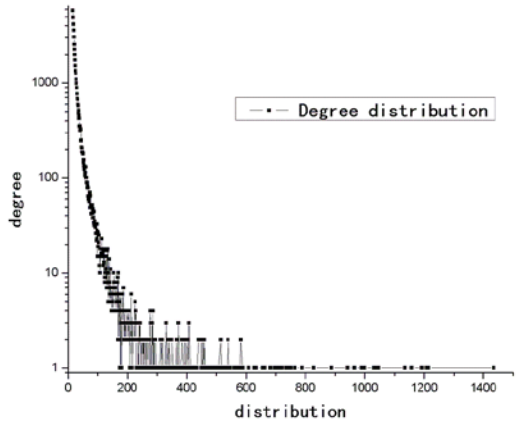

Fig 4 Distribution of node degree in simulation

Table 1 Simulation parameters

Parameter	Symbol indicates	Values (range)
total number of super nodes	N	10240
the number of leaf nodes affiliated with the super node	LN	[20,30]
The total number of document categories	DN	$0.8\times N$
the initial number of copies	R	$[0.02, 0.1]\times N$
the document number stored by Node i	ST_i	$[0.1, 0.5]\times DN$
the initial document number of the node i	SI_i ,	$[0.1, 0.5]\times ST_i$
query TTL value and forwarding neighbors	TTL_p, Nei	2, 3
average connection degree	d_{avg}	29.991

5.2 Results and analysis

In this experiment, on the premise of submitting the same query , compare the performance of OR,PR and PPSR from the following three aspects:

1. the time required to return the specified number of results (set to 10);

2. bandwidth consumption caused by query;

3. the storage costs. Storage cost is calculated as: storage cost = the storage that system has consumed /total storage of the system.

Experiment runs 25,000 rounds into 50 data collection points and each data collection point contains 500 rounds. The data from each data collection point is the mean of data from the 500 rounds. To note is, in simulation experiments, different number of nodes (test node, such as 2048, 4096, 10240 and 20480) get the same experimental conclusion. So this article lists only the situation of 10240 nodes.

What is shown in Figure 5 is the query bandwidth consumption comparison of OR, PR and PPSR.As can be seen, as the number of experimental rounds to increase, PPSR has greater advantages than the other two strategies and continues low bandwidth consumption. Bandwidth consumption caused by the query in OR and PR is always high. This is because, although OR and PR create a replica of the file for those with the successful query, their creation has a "blindness" that on all replicas of the same treatment, this is bound to affect the validity of the replica creation operation. PPSR guides replica creating operation by a reasonable estimate to the popularity of the replica. The replica with higher access probability in a region would be created in this region and reasonably allocated the number. So queries are allowed under a smaller TTL to be access to a sufficient number to the destination file and the bandwidth consumption is reduced.

What is shown in Figure 6 is storage cost comparison of three policies. Compared with the OR, PR, storage cost of PPSR is small. Similar to the previous analyses to bandwidth consumption, due to PPSR using popularity of files to direct replicas creation, and combining with its indexing mechanism on the way, replica creation operation can maximum avoid "blindness", efficiently using the storage.

What is shown in Figure 7 is the average response time of the query under three policies. In the early experimental runs, the query of PR strategy has a shorter response time. As the number of experimental rounds to increase, query response time of the PPSR is declining and gradually approaching the PR strategy. When nearly 20,000 rounds, response time of PPSR gradually flat with PR. This showing early in the system running, query response time for PR strategy should be shortest. After the system runs stable, PPSR policy unchanged from the PR strategy on query response

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

time, and they are better than OR strategy. Caveat is that PR strategy, reducing query response time, is implemented by means of a large number of redundant replicas, leading to storage is costly (such as analysis of Figure 4);PPSR strategy reduces query response time by heuristic method to create a replica of the policy and taking a smaller storage costs and bandwidth consumption to achieve.

In conclusion, PPSR policy is better than OR and PR strategies on both bandwidth consumption and storage costs; After the system runs stable, query response time of PPSR strategy and PR strategy flats, and surpass the OR policy. Thus, taken together, compared with both OR and PR strategy, PPSR policy can better improve the performance of P2P network.

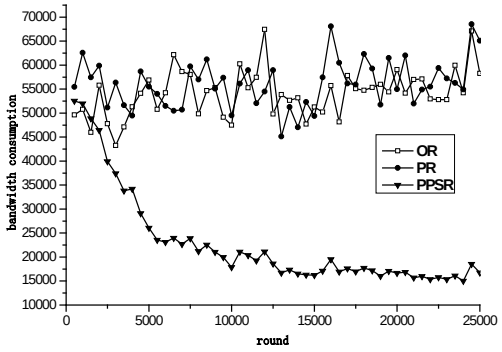

Fig 5 Bandwidth consumption by three strategies

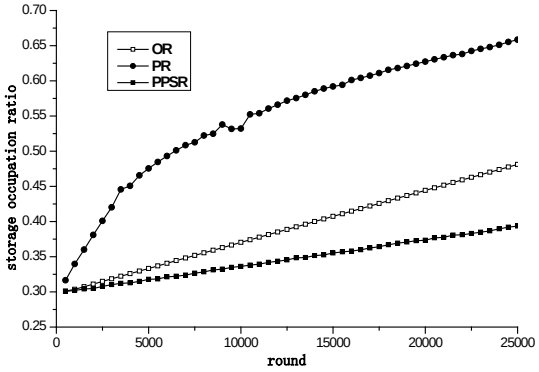

Fig 6 Comparison of the storage consumption

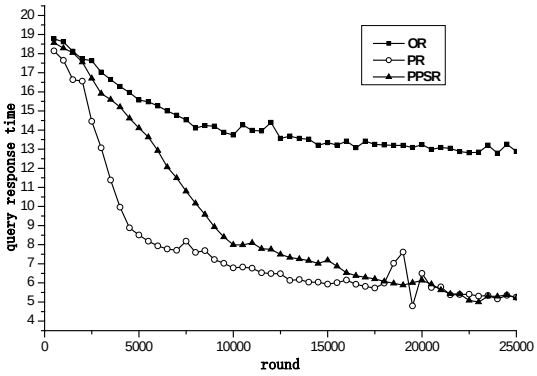

Fig 7 Comparison of query response time

6. CONCLUSION

In this paper, a novel popularity based replication scheme PPSR is proposed. In PPSR, the heterogeneity of the resources is taken into consideration, and popularity of resources is used to help the procedure of replica creation, which can make the amount and distribution of the replicas be more efficient. Super node layer replica creation strategy and the bottom layer balancing strategy are integrated based on the current two layer topology of P2P network. The simulation results indicate that PPSR shows better performance than the compared strategies. Next, we will further study replica creation and update process, and how to keep replication consistency with small costs.

ACKNOWLEDGEMENT:

This work was supported by Central Government University Foundation (Grant No. JB2012087).

REFRENCES:

[1] L. Qin, C. Pei, C. Edith. "Search and replication in unstructured peer-to-peer networks", *Proceedings of the 16th international conference on Supercomputing*, New York USA, 2002, pp. 84-95.

[2] Fu, T.Z.J, Dah Ming Chiu. "Statistical modeling and analysis of P2P replication to support VoD service", Proceedings of the INFOCOM 2011, HongKong China, 2011, pp. 945-953.

[3] Kolici, V., Potlog, A.-D., Spaho, E., et.al. "Data Replication in P2P Collaborative Systems", Proceedings of the 7th international conference on P2P, Parallel, Grid, Cloud and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Internet Computing, Barcelonia Spain, 2012, pp. 49-57.

[4] SUN Hai-yan, WANG Xiao-dong and ZHOU Bin. "The Storage Alliance Based Double-Layer Dynamic Replica Creation Strategy-SADDRES". *Acta Electronica Sinica*, Vol. 7, 2005, pp. 1222-1226.

[5] Yipeng Zhou, Tom Z. J. Fu and Dah Ming Chiu. "Statistical modeling and analysis of P2P replication to support VoD service". *Proceedings of IEEE Infocom 2011*.Shanghai China, 2011, pp. 945-953.

[6] G. Xie, Z. Li, and Z. Li. "Efficient and Scalable Consistency Maintenance for Heterogeneous Peer-to-Peer Systems", *IEEE Trans. Parallel and Distributed Systems*, Vol. 19, No. 12, 2008, pp. 1695-1708.

[7] H. Shen. "An Efficient and Adaptive Decentralized File Replication Algorithm in P2P File Sharing Systems", *IEEE Trans. Parallel and Distributed Systems*, Vol. 21, No. 6, 2010, pp.827-840.

[8] K. Yohei , M. Noriko and Y. Norihiko. "Popularity-Based Content Replication in Peer-to-Peer Networks". *Lecture Notes in Computer Science*, Vol. 3994, 2006, pp.436-443.

[9] Guo, Liangmin, Yang, Shoubao, Wang, Shuling . "Replica Deletion Strategy Based on Gray Prediction Theory and Cost in P2P Network", Proceedings of the international conference on Computer Science & Service System, Barcelonia Spain, 2012, pp. 2243-2246.

[10] ZHOU Xu, LU Xian-Liang, HOU Meng-Shu et al. "Research on Distributed Dynamic Replication Management Policy"]. *Journal of Electronic Science and Technology of China*, Vol 3. , 2005, pp.97-102.

[11] D. Stutzbach, R. Rejaie and S. Sen. "Characterizing unstructured overlay topologies in modern p2p file-sharing systems". *Proceedings of Internet Measurement Conference*. Berkeley USA, 2005, pp. 49-62.

[12] Gnutella. http://www.gnutella.com. 2012.

[13] R. Schwarz, F. Mattern, "Detecting causal relationships in distributed computations: In search of the Holy Grail". *Distributed Computing*, Vol. 7, 1994, pp.149-174.

[14] Medina A., Lakhina A. and Matta I. "BRITE: an approach to universal topology generation". *Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. Cincinnati, OH 2001, pp.346-353.