

Towards a Content Delivery Load Balance Algorithm Based on Probability Matching in Cloud Storage

Chen Xiang^{1,*}, Tang Junyong² and Zhang Yong³

¹School of Civil & Architecture Engineering, Xi'an Technological University, Xi'an, 710021, P.R. China; ²School of Computer Science & Engineering, Xi'an Technological University, Xi'an, 710021, P.R. China; ³Wuhan University of Science and Technology, Wuhan, 430081, P.R. China

Abstract: Load balance just reasonably allocate request from users to each edge cloud storage servers. It can avoid overloading or congestion to ensure data transmission efficiency. Referring to idea of ant colony algorithm, the paper brought out Content Delivery Load Balance Algorithm Based on Probability Matching (CDLBA-PM) algorithm. The method calculated access probability of each edge cloud storage peer and selected edge cloud storage peer with probability matching algorithm comprehensively considering about history access costs and real-time access cost of edge cloud storage peers. It fully takes into account current network bandwidth, edge cloud storage peer performance as well as history access cost. Random probability matching was also introduced for cache copy selection in the algorithm. Simulation experiment results with CloudSim show that CDLBA-PM can not only reduce response time of user request, but also effectively balance load of edge cloud storage peer.

Keywords: Ant colony algorithm, cloud storage, load balance, probability matching.

1. INTRODUCTION

Cloud storage is an Internet service itself, which emphasizes centrally providing resource by cloud data center but weaken software and hardware capability of terminals. After cloud storage service been used, all content needed is kept in distance data center and users access to it *via* network [1-5]. So, the cloud storage service may also use Content Delivery Network (CDN) as accelerate manner similar to other Internet services, so as to achieve higher access efficiency and better user experience. The cloud CDN reduces own cost using competitive price providing by different cloud. Combined with on-demand service of cloud, the CDN can easily adjust own storage and bandwidth usage according to the requirement. It may also reduce cost by reducing the quality of service. Meanwhile, cloud CDN can reuse various clients or sites among users. In another word, it provides functions similar with traditional CDN without any infrastructure actually.

Scholars have made preliminary studies on programs of CDN based on cloud storage. The most representative one is MetaCDN proposed by Broberg *et al.* in [6-8], which is a low-cost CDN using storage cloud resources. The system provides a mechanism to deploy content in the network provided by different storage cloud service providers and relocate to appropriate copy corresponding to user requests. However, the system does not put forward new cache strategy and load balance algorithm. The F_{cache} content

acceleration cache technique brought out by FastWeb just cache and find optimal match on web acceleration content according to strategy. Within a period of time, it does not access file entity from source website for repeated access, but directly content copy from cache to users, thus effectively improving response and saving bandwidth [9].

The cost of cloud CDN includes bandwidth and storage cost. However, existing content delivery techniques have not provided reasonable solution to charging mechanism of cloud CDN. At the same time, the network topology of cloud storage is quite different from traditional CDN networks. In order to operate cloud CDN effectively, it is necessary to study on effective content delivery and reasonable load balance strategy in cloud storage. In the content delivery process, after user issued resource request, the load balance server will find most suitable edge server to deal with requests, namely the user request redirected technology. The target is to effectively and correctly delivery user requests and related traffic to servers working in normal status so that each background server keeps load balance as possible. Referring to idea of ant colony, the paper proposed CDLBA-PM algorithm. It not only considers current network load, but also takes edge cloud storage peer performance and history access information into account so that the selection of user on edge cloud storage peer is globally. The specific arrangement of the paper is as follows: section 2 analyses target and algorithms of load balance; section 3 gives load balance algorithm based on probability matching; section 4 performs simulation experiments on the proposed algorithm and section 5 concludes our work.

2. LOAD BALANCE SCHEDULING AND ALGORITHMS

2.1. Load Balance Scheduling

A user in the content delivery system can send access requests to several edge cloud storage peers, which needs load balance algorithm to select a suitable one to response user's request from multiple peers. It also balances load at edge cloud storage peers. The selection of optimal edge cloud storage peer is an important technique in the content delivery system. The selection of optimal peer should be determined in accordance with specific status and needs from requesters, which is dynamical. Appropriate edge peer selection strategy is affected by many factors, including network bandwidth, delay, storage space of each peer, read and write speed of server disk, reliability as well as access cost. The access cost of edge cloud storage peer is obtained by function and the peer with least cost response to connection request. Appropriate load balance strategy of content deliver in the cloud storage can improve system delivery performance, reduce network congestion and improve user experience.

In order to develop selection strategy of edge cloud storage peer, we firstly discuss indexes that can affect system performance. These indexes are important factors that determine whether the edge cloud storage peers can be selected, including read speed, network bandwidth and peer load. Considering about factors that affect system performance, the final goal is to construct access cost function of edge cloud storage peer.

The calculation of access cost is that should be paid to access cache copy on some edge cloud storage node. The cost is related with network bandwidth between user that sends service request and edge peers as well as the performance of edge peer itself. The key to implement edge cloud storage peer selection is to compute and predict access cost at the edge peer and to select the peer with least prediction access cost. The access cost of some peer mainly includes two parts, communication cost and peer response time. The communication cost is related to current network bandwidth between user and edge storage peer. The peer response time is related to the peer performance. Here only the data read speed of copy from edge cloud storage unit is considered.

2.2. Scheduling Algorithm Analysis

Currently, there are many load balance scheduling algorithms in commercial systems. Generally speaking, the scheduling algorithms can be divided into two major categories, namely static algorithm and dynamic algorithm [10-13]. The static algorithms redirect delivery in accordance with set strategies, not considering about real-time change of load at edge servers. It is relatively simple to implement.

Typical static load balance algorithms include round robin, weighted round robin, random, weighted random, source IP Hashing, source IP and source port hashing, destination IP hashing and UDP packet load hashing. The Round Robin (RR) algorithm allocates requests to different edge servers. Each edge server peer undertakes user requests equally. The algorithm is suitable to scene that performance of each edge server in the cluster has not distinguished difference. The difference between weighted round robin and

RR is the users undertaken by each edge server not same. The server with larger weight is allocated more requests and vice versa. Random algorithm distributes requests to different edge servers randomly. View from aspect of statistics, the result of scheduling is each edge server equally undertake users' connection request. In the weighted random algorithm, each edge server undertakes connection requests in proportion to weight. The source IP hashing algorithm maps request from same source IP to an edge server with a Hash function, which is usually applied to scene that it needs to ensure requests from same user should be delivered to same edge server. The difference between source IP and source port hashing and the former one is that it maps requests from same source IP and port number to same edge server. The destination IP hashing just maps request from same destination IP to an edge server with a Hash function, which is suitable to scene that it needs to ensure requests to same destination should be delivered to same edge server. The UDP packet load hashing method maps requests of same content in specific fields of UDP load packets to a same edge server. The algorithms is adapt to scene that it needs to ensure same request in specific field of UDP packet load to same server.

Dynamical load balance algorithm can analyze real-time load of each edge peers, adjust contention distribution scheme, so as to reduce response time and improve QoS. Typical dynamic load balance algorithms include least connect, weighted least connection and least response time. These dynamic algorithms all use accurate compute model. The performance and connection status of redirected edge server may lead to local hot spot phenomenon. The least connection algorithm allocates new connection to edge server with least connection number in accordance with the load of server estimated from current connection number. The method can reasonably allocate user request that keeps long time to each server, which is suitable for scene that different users have different delivery time difference requirements. The weighted least connection algorithms schedule the upper edge server activities already established connections as much as possible so that new connections right edge servers is proportional to the value. The method is usually used in the scene that each server in the cluster has distinguished difference and the connection time from different user is quite different. The least response time method selects edge server with least response time as possible. The method is suitable to situation that user request has high response time needs on edge server.

In actual application scene, we can select different scheduling algorithm according to the load balance effect to be achieved. For example, Web edge server cluster can use round robin algorithm to gain higher access response. FTP edge server cluster adapts least connection algorithm to reduce pressure of single server. Special application cluster as tax can use source IP and port hashing algorithm to ensure continuity of user access. Except for above traditional delivery algorithms, many delivery algorithms aiming at content analysis also attracted more and more emphasis by load balance device providers. However, all these algorithms have not considered about fitting for load balance requirements in specific scene. As to load balance of content delivery in cloud storage, the physical distribution of cloud data center, physical distance and network bandwidth among user and

edge cloud storage peers, history access and response information of edge cloud storage peers as well as charging mechanism of cloud storage have not been taken into account, so they are not suitable for implement of content delivery load balance scheduling in the cloud storage.

3. CDLBA-PM ALGORITHM

3.1. Algorithm Description

There are currently two kinds of peer selection methods of edge cloud storage peers. One is based on performance of edge cloud storage peer. It just establish performance model for content delivery system to select edge peers by predict peer performance. The performance model prediction mainly includes establishing performance model of content delivery system as well as obtaining physical parameters in the performance model.

Ant Colony Optimization (ACO) is an important artificial intelligent algorithm. It is firstly put forward by M. Dorigo *et al.* in Italy. It is inspired from ants foraging in nature. Ant individual communicates to each other with pheromones and effectively find food through concentration of pheromone. The ACO algorithm is derived from this process. The principle of ACO is actually a positive feedback mechanism. By continuous accumulation of pheromones, the result converges to optimal path. It has natural self-convergence, load balancing and distribution features, which has been widely used for parallel computation and solving NP-hard problems.

Referring to idea of ACO, the paper proposes load balance algorithm based on probability selection. It not only takes into account current network load, but also consider about edge storage peer performance as well as history access information so that the selection of users of edge cloud storage peer is more global. The evaluation of cloud storage peer selection mainly determines whether the selected peer can decrease the access cost of users on edge cloud storage peers as a whole. The algorithm mainly considers two factors, namely history access value of edge cloud storage peer and real-time cost of current request resource.

The access value of edge cloud storage peer is that it been accessed. It just decreases with the increment of peer access cost.

Edge cloud storage node selection algorithm is good or bad, the key is to look at the selected edge cloud storage node can reduce the user's access edge cloud storage node cost a whole. Algorithms mainly consider the cost of two factors: the edge of cloud storage node to access the value of historical and real-time cost of the current request resources. The access cost has large relationship to peer performance in that it requires edge cloud storage peer firstly takes cache copy out from storage unit in case of response request on some resource of a user. After process, it initiates transmission. So the response time of edge cloud storage peer is closely related to storage speed of unit where cache copy located. Therefore, the initialization of edge cloud storage peer history access value is

$$HP_{Nv}(0) = \frac{1}{t} \quad (1)$$

where, t is response time. The network bandwidth between users request for service and edge cloud storage peer is dynamically changing. The prediction on real-time network bandwidth is

$$BW_{Nv(t)} = \frac{MaxBandwidth_{Nv}}{Connections_{Nv}(t)} \quad (2)$$

where, $MaxBandwidth_{Nv}$ is the maximize real-time bandwidth between user N and edge cloud peer v ; $Connections_{Nv}$ is the connection number between user N and peer v . Before connection established between user N and peer v , add 1 to connection number between user N and peer v . After transmission, decreases 1 to connection number among peers so that each network connects equally allocate network bandwidth.

The prediction on real-time access cost is

$$\omega_{Nv}(t) = \frac{filesize}{BW_{Nv}(t)} \quad (3)$$

The selection on edge cloud storage peer is based on prediction of two cost functions. Larger is history access cost and real-time access cost, greater is the cost to be paid for select this edge cloud storage peer. In other words, the probability to select this edge cloud storage node should be small as possible.

The selection price and file access cost are two different concepts. The access cost indicates that should be paid by this edge cloud storage peer, while selection price means whether this peer is valuable to be selected. Under normal circumstances, larger access cost of edge cloud storage peer means smaller access cost. The access price function of edge cloud storage peer is

$$Price_{Nv} = \frac{\alpha}{\omega_{Nv}(t)} + \beta HP_{Nv} \quad (4)$$

By adjusting on parameters α and β , the proportion of communicate cost and history access cost in the whole access cost can be determined. The formula of selection probability function can be arrived in the order of its selection price in accordance with selection price of all edge storage peers.

$$P_{Nv}(t) = \frac{Price_{Nv}}{\sum_{k \in list(LN)} Price_{Nk}} \quad (5)$$

In the probability convergence process, in case of user N access to resource R_i , if edge cloud storage peer v is selected to transmit cache copy of resource R_i on it, the probability that the cache copy be accessed at next request at this edge cloud storage peer is large based on file access local principle. The updated history access price function of edge cloud storage peer v is

$$HP_{Nv}(t + \Delta t) = HP_{Nv}(t) + \frac{1}{\omega_{Nv}(t)} * \epsilon \quad (6)$$

Thus, the access probability of edge cloud storage peer with larger access price will increase. Finally, the access probability of each peer will automatically tend to a stable

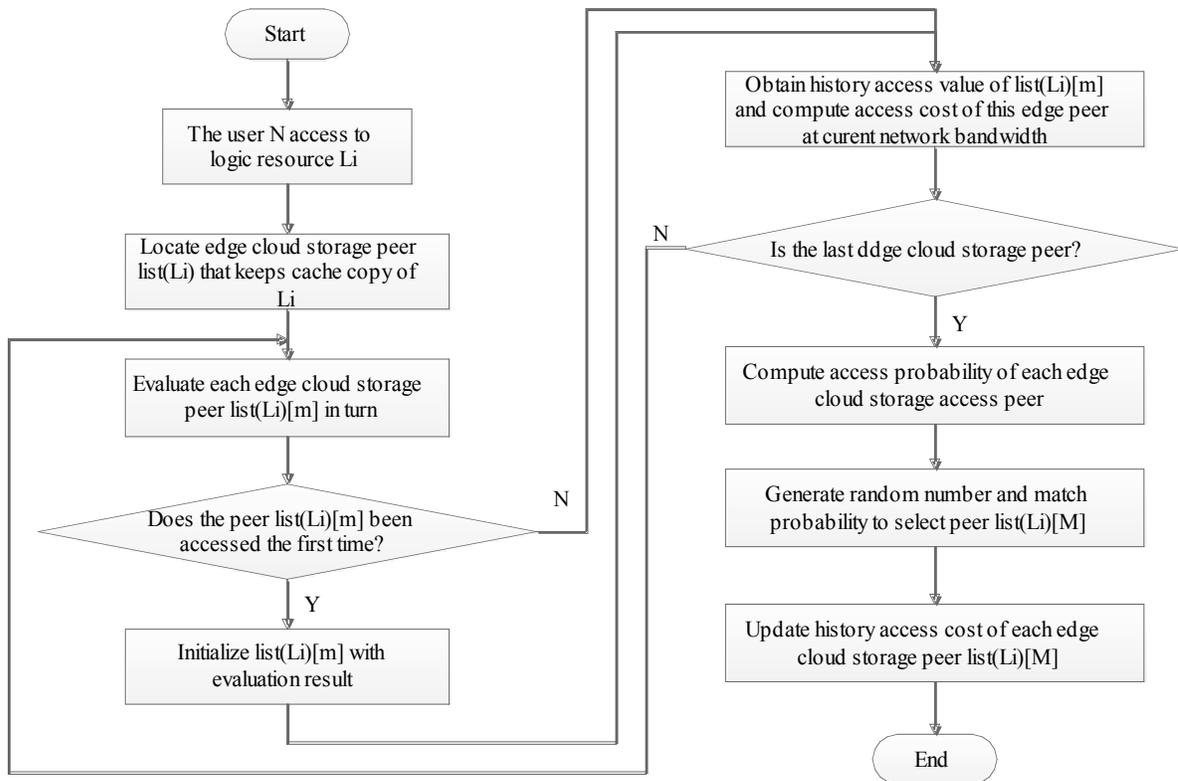


Fig. (1). CDLBA-PM load balance algorithm flow.

and efficient status, called self-convergence of probability. Where, the parameter ε is used to adjust convergence speed.

According to above probability formula, compute access probability of all edge cloud storage peers where there is user request file copy in the cache. If the peer selects one with maximum access probability each time, it results to edge cloud storage peer with larger bandwidth will be frequently accessed, while some peers may not be accessed, so as to waste resources. So the probability matching algorithm was applied here.

Assume there are m edge cloud storage peers that meet above requirements. Firstly obtain the history access price of each one. If an edge cloud storage peer is accessed the first time, initialize its history access price in accordance the read speed of this edge peer. Subsequently, calculate real-time access cost of edge peer according to (4). Then, the access price of this peer can also be obtained. After access price of each edge cloud storage peer been obtained, compute the selection probability $q(v) (v \in list(LN))$ of each peer. Set $q(0)=0$ and $q(j) = q(0) + q(1) + \dots + q(j) (j \in list(LN), 0 \leq j \leq m)$. Generate a random number $rand$ in $[0, 1]$ with random function. Compare $rand$ with edge peers in the $list(LN)$. If $q(j-1) \leq rand \leq q(j)$, the edge cloud storage peer represented by j is selected. Finally, in accordance with file local access principle, appropriately add the history access price of the selected edge cloud storage peer. Thus, it has larger probability to be selected at next round, so that the probability distribution of edge peer been selected automatically converge to optimal status.

3.2. Algorithm Implementation

The flowchart of CDLBA-PM is shown in Fig. (1). The specific implementation steps of CDLBA-PM are shown in Fig. (2).

4. SIMULATION EXPERIMENT AND RESULT ANALYSIS

4.1. Experiment Condition

The simulation was carried out with CloudSim simulator. The network topology for simulation is shown in Fig. (3). The network transmission speed of peers among regions in the network topology is set 1Mbps; data transmission speed between peers in region as 10Mbps. The file read speed of each edge cloud storage peer is randomly set. The CDLBA-PM algorithm was implemented on the CloudSim platform to compare with content delivery load balance methods as RR (Round Robin), Random and SIPH (source IP hashing). The function that it plays in aspect of reducing average job time and balancing peer load was verified.

4.2. Result Analysis

At the circumstance of number of job at 500, 1000 and 2000, compare performance of CDLBA-PM, RR, random and SIPH from the aspect of reducing average job time and load balance status. The mean job time and load balance status are shown in Figs. (4 and 5).

The node load status is evaluated by variance of total number that each edge cloud storage peer been accessed. Smaller is the variance, the user request distribution is more

```

Input: User N that initiate access and Logic resource LFN to be accessed
Output: The edge cloud storage peer where optimal cache copy located

Get all the edge peers that replica LFN;
For (i=0; i<number of edge cloud peers; i++)
{
    if(!edge cloud peers(i) is initialed)
    {
        calculate the value of  $HP_{Nv}(0)$  using equation (1);
    }
    calculate the value of  $HP_{Nv}(t)$  using equation (3);
    calculate the value of PriceNv(t) using equation (4);
    add sum by PriceNv(t) for the calculate of PNv(t);
    if (i=number of edge cloud nodes-1)
    {
        For(j=1; j<number of edge cloud nodes; j++)
        {
            calculate the value of PNv(t) using equation (5) according to sum;
        }
        generate a random value ran between 0 and 1;
        For (j=0; j>number of edge cloud peers; j++)
        {
            If (ps(j)<ran<ps(j+1))
            {
                choose edge cloud peers(j) as the best choice;
                update the history load of edge cloud peers(j);
                return edge cloud peers(j);
            }
        }
    }
}
    
```

Fig. (2). CDLBA-PM algorithm implementation.

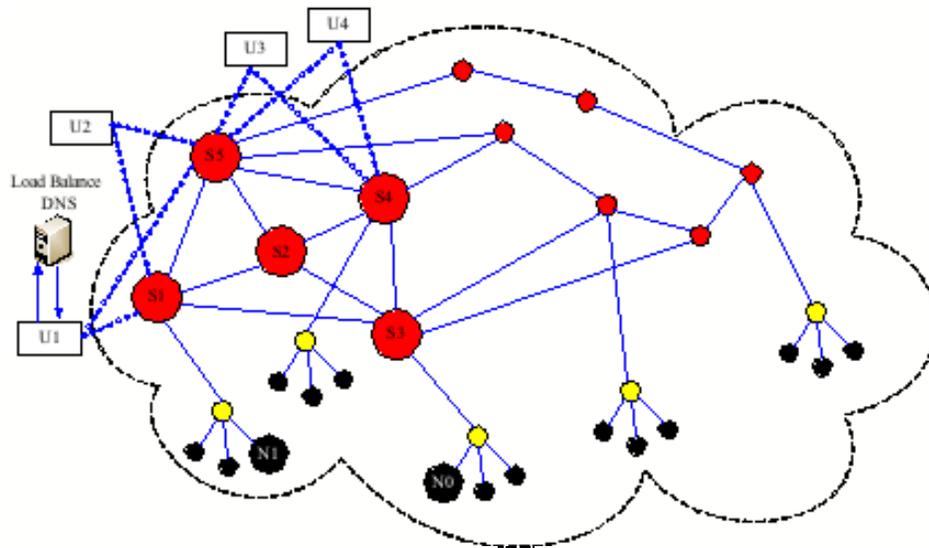


Fig. (3). Experiment network topology.

evenly among peers, namely the load on all edge cloud storage nodes are balanced.

The purpose of load balance is to maintain effective distribution of edge servers by balancing load of servers and to reduce resource request time of users so that they get better services. From Fig. (4) we can see that the CDLBA-PM algorithm proposed in the paper can better converge distribution time of overall user request file resource to low level compared with existing RR, random, SIPH algorithms. It achieves that fundamental purpose to optimal user experiences of load balance. At the same time, we can see from the

Fig. (5) that the CDLBA-PM algorithm can effectively balance load among peers while reducing resource file request distribution time. Thus, the load of edge cloud storage peers on content delivery network is equalized, which take full advantage of cloud storage resource to achieve content delivery to great extent.

CONCLUSION

Based on introduction of typical content delivery load balance algorithms, the paper proposed self-convergence load balance algorithm of content delivery in cloud storage

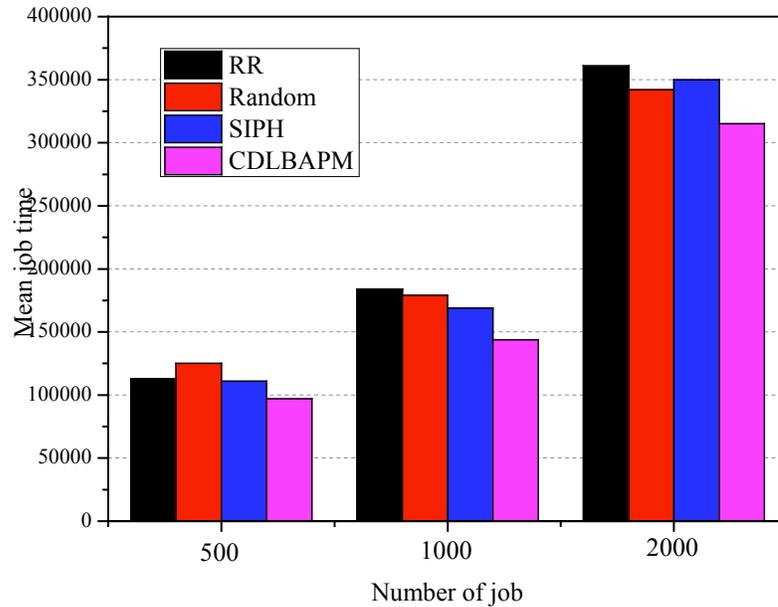


Fig. (4). Mean job time comparison.

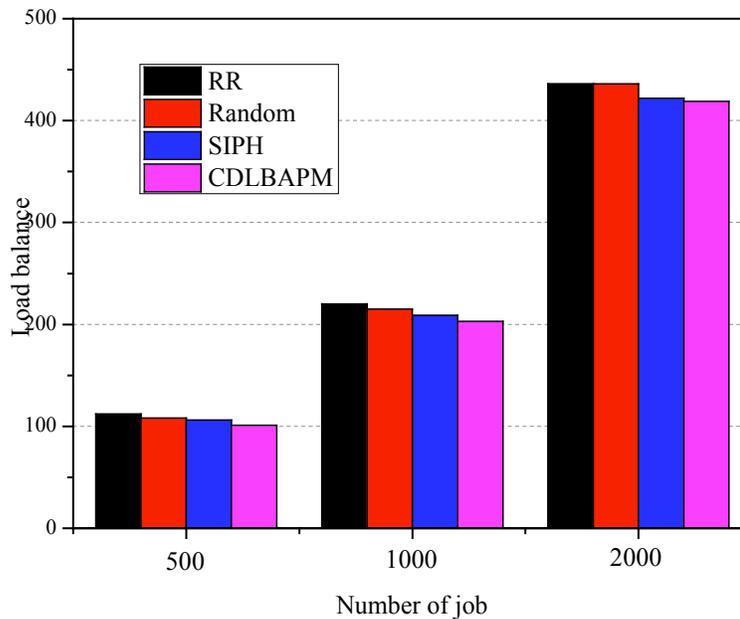


Fig. (5). Peer load balance comparison.

on the basis of probability by studying on edge cloud peer selection target and objectives. The algorithm comprehensively considers history access cost and real-time access cost of edge cloud storage peer. After obtained access probability of each edge peers, the probability matching algorithm was used to select edge cloud storage node. Finally, the performance of CDLBA-PM algorithm was compared with RR, random, SIPH methods from aspects of mean job time and load balance circumstance. In case of content delivery and load balance researches, we usually recognized all the cache copy information of resource been obtained, which is also related to consistency of the cached copy. It is the focus that we will pay more attention in the next step researches.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] F. Chen, K. Guo, J. Lin, and T. La Porta, "Intra-cloud lightning: Building CDNs in the cloud", In: *Proceedings of IEEE INFOCOM*, Florida, 2012, pp. 433-441.

- [2] M. Hu, J. Luo, Y. Wang, and B. Veeravalli, "Practical resource provisioning and caching with dynamic resilience for cloud-based content distribution networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2169-2179, 2014.
- [3] Z. Zhuang, and C. Guo, "Building cloud-ready video transcoding system for Content Delivery Networks (CDNs)", In: *Proceedings of 2012 IEEE Global Communications Conference (GLOBECOM)*, California, 2012, pp. 2048-2053.
- [4] F. P. Tso, and D. P. Pazaros, "Improving data center network utilization using near-optimal traffic engineering", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1139-1148, 2013.
- [5] G. Keller, M. Tighe, H. Lutfiyya, and M. Bauer, "A hierarchical, topology-aware approach to dynamic data center management", In: *Proceedings of 2014 IEEE Network Operations and Management Symposium (NOMS)*, Krakow, 2014, pp. 1-7.
- [6] J. Broberg, R. Buyya, and Z. Tari, "MetaCDN: 'Harnessing 'Storage Clouds' for high performance content delivery", *Journal of Network and Computer Applications*, vol. 32, no. 5, pp. 1012-1022, 2009.
- [7] A. Saad, and A. El-Mahdy, "Network topology identification for cloud instances", In: *Proceedings of 2013 Third International Conference on Cloud and Green Computing (CGC)*, Karlsruhe, 2013, pp. 92-98.
- [8] C. Inzinger, S. Nastic, S. Sehic, and M. Voßler, "MADCAT: A methodology for architecture and deployment of cloud application topologies", In: *Proceedings of 2014 IEEE 8th International Symposium on Service Oriented System Engineering (SOSE)*, Oxford, 2014, pp. 13-22.
- [9] FastWeb Corp., Website solution, [Online]. Available: <http://www.fastweb.com.cn/a/solution/>.
- [10] D. Battre, N. Frejnik, S. Goel, and O. Kao, "Evaluation of network topology inference in opaque compute clouds through end-to-end measurements", In: *Proceedings of 2011 IEEE International Conference on Cloud Computing (CLOUD)*, Beijing, 2011, pp. 17-24.
- [11] P. Talebifard, and V. C. M. Leung, "Expansion properties of topology for networking of information in cloud", *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2877-2887, 2014.
- [12] B. Li, M. J. Golin, G. F. Italiano, and X. Deng, "On the optimal placement of web proxies in the Internet", In: *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, 1999, pp. 1282-1290.
- [13] J. Broberg, R. Buyya, and Z. Tari, "MetaCDN: Harnessing 'Storage Clouds' for high performance content delivery", *Journal of Network and Computer Application*, vol. 32, no. 5, pp. 1012-1022, 2009.

Received: July 15, 2015

Revised: August 29, 2015

Accepted: September 23, 2015

© Xiang *et al.*; Licensee Bentham Open.

This is an open access article licensed under the terms of the (<https://creativecommons.org/licenses/by/4.0/legalcode>), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.