Journal of Theoretical and Applied Information Technology
31st January 2014. Vol. 59 No.3

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

VIDEO REPLICA PLACEMENT STRATEGY FOR STORAGE CLOUD-BASED CDN

[1] SHIJIA. YAO,[2]WENLE ZHOU ,[3]HAOMIN CUI AND, [4]MING. ZHU
[1,2,3,4]Department of Automatic of University of Science and Technology of China, Hefei, China
Email: [1,2,3]{ yaosj ,chhmm,wlzhou}@mail.ustc.edu.cn, [4]mzhu@ustc.edu.cn

ABSTRACT

The online video service need the support of CDN(Content Delivery Networks). Compared with traditional CDNs, it can save a lot of cost by using cloud-based storage nodes to deliver the video content. To guarantee end users' QoS, CDN should pre-deploy the content files of online video service to the edge nodes which are close to the users. Existed researches have shown that the cost of building CDN by cloud storage nodes is much less than that of using traditional CDNs. The existed off-line replica placement algorithm named GS(Greedy Site) can meet the QoS requirement with relatively small cost when the information of users' requests is provided. However GS will result in bad load balance and it need the information of users' requests. In this paper, two classes of offline algorithms are proposed. One named GUCP(Greedy User Core Preallocation) effectively solved the load imbalanced problem caused by GS ,and the other one named PBP(Popularity Based Placement) which is based on the popularity of content effectively placed replicas while there is no users' requests information. Numerical experiments have demonstrated the effectiveness of the algorithms above.

Keywords: *Cloud storage, CDN, Replica Placement, Load balance，QoS.*

1. INTRODUCTION

With the rapid development of online video service, there has been a large number of small companies of online video service. The video online service needs the support of CDN(Content Delivery Networks). Traditional CDNs such as Akamai and Mirror Image have deployed tens of thousands of data centers and edge servers to deliver content across the globe. Unfortunately, the price of traditional CDNs(such as Akamai) is so much higher than small organizations such as medium-sized enterprises, government agencies, universities, and charities[1].The price of building CDNs or hiring existing CDNs is much higher than the ability of finance of medium-sized video service provider. As a result, the idea of utilizing storage clouds as a poor man's CDN is very enticing. The cloud storage providers promise the ability of rapid, cheap reading and writing and are easy to be expanded to meet flash crowds of web sites. Economies of scale, in terms of cost effectiveness and performance for both providers and end users, can be achieved by leveraging existing "storage cloud" infrastructure, instead of investing large amounts of money in their own content delivery platform or utilizing one of the incumbent operators like Akama[2].

The recent emergence of storage cloud providers such as Amazon S3, Nirvanix and Rackspace has opened up new opportunities to provide cost-effective CDNs. Storage cloud providers operate data centers that can offer Internet-based content storage and delivery capabilities with the assurance of service uptime and end user perceived service quality. Service quality is typically in the form of bandwidth and response time guarantees [3].

Utilizing storage cloud building CDNs can effectively reduce the cost of content storage and delivery. In rest of this paper, the based-cloud storage CDN is called cloud CDN for short. .

It's difficult to use multiple cloud storage to provide service of CDN, because each cloud storage providers offers different Web services or programmer APIs and each service is best utilized via unique Web services or programmer APIs and has their own unique quirks. Many Web sites have utilized individual storage clouds to deliver some or all of their content [4], most notably the New York Times [5]and SmugMug [6], however, there is no general-purpose, reusable framework to interact with multiple storage cloud providers and leverage

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

their services as a content delivery network. Most "storage cloud" providers just provide basic file storage and delivery services. But do not offer the capabilities of a typically CDN such as automatic replication, fail-over, geographical load redirection, and load balance. Furthermore, a customer may need coverage in more locations than offering by a single provider. The MetaCDN is a system that utilizes numerous cloud storage providers in order to create an overlay network that can be used as a high-performance, reliable, and redundant geographically distributed CDN to solve these problem[7].Storage cloud providers charge their customers by their storage and bandwidth usage following the utility computing model [8]. Storage cost is measured per GB per unit time and bandwidth cost is measured per GB transferred. Bandwidth cost consists of upload cost for incoming data and download cost for outgoing data.

The costumers of storage cloud, cloud CDN are accustomed to utilize different cloud storage providers in order to reduce the cost. Because cloud storage can be scaled on-demand, cloud CDN can be easily adjusted according to demand. Cloud CDN can offer multiple resources to multiple customers as traditional CDN. In other words, cloud CDN can provide service such as traditional CDN, but without maintaining or owning any infrastructure.

The file of online video service is very large, the response time should be as short as possible as possible. The replicas should be placed on the edge nodes, which are nearest to the users.

In this paper, two kinds of off-line replica placement algorithms for cloud CDN which provides service for online video service were proposed. One algorithm named GUCP could be used to place replicas when cloud CDN has users and requests from them. This algorithm could solve the load imbalance problem which is caused by GS. The other algorithm named PBP could used to place replicas while there is no information of users' requests in cloud CDN.

2. RELATED WORKS

The replica placement problem in traditional CDNs refers to finding the best set of servers to place content replicas. Replica placement belongs to the NP-complete class of problems [9]. The CDN performance can be affected by decisions such as:

1. The number of surrogates required.
2. Their location.
3. The cost model adopted including storage cost, content retrieval cost, and content updating cost.
4. QoS considerations.

A considerable amount of research has been done for replica placement in CDNs. The cost model has evolved to include one or more of the three types of costs: retrieval (or download), storage and update (or upload) cost.

In terms of minimizing content retrieval cost only, Li et al. [10] and Krishnan et al. [11] showed that replica placement in general network topologies is NP-complete and provided optimal solutions for tree topologies. Qiu et al. [12] evaluated a number of heuristics and found a greedy algorithm offering the best performance.

In addition to retrieval cost, Xu et al. [13] and Jia et al.[14] further added update cost, whereas Cidon et al. [15] added storage cost into consideration. Furthermore, Kalpakis et al. [16] comprehensively considered all three costs (retrieval, update and storage) and offered solutions for a tree topology only. However, none of the work studied the case in which provisioning cost between.

MetaCDN [7] system is a commercially available cloud based CDN that provides an interface for standard cloud providers to be used for content delivery. The system, via an appropriate web portal, grants end users with a number of different options related to cost and QoS. Specifically, it enables a set of replica deployment options that consequently define the request redirection policies [17].

However, the details of the replica placement strategies are not provided. Chen et al. [18] are the first ones to investigate the problem of placing server replicas in storage cloud CDNs along with building content distribution paths among them. Their goal is to minimize the cost incurred on CDN providers while satisfying QoS requirements for end users.

In the proposed work, we go one step further by considering the optimized replica placement problem and distribution path construction for networked cloud environment. Two kinds of off-line replica placement algorithms for cloud CDN either have or not the information of users' request.

3. CLOUD CDN AND ITS ERFORMANCE

3.1 Cloud CDN

The cost of cloud CDN results from the bandwidth and storage, which is changing with users load. Intelligent replica placement and users redirection strategies are required. The problem of replica placement is investigated widely in the traditional CDN. However, the existing results on

Journal of Theoretical and Applied Information Technology
31st January 2014. Vol. 59 No.3

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

replica placement cannot be used on cloud CDN setting for the flowing reasons:

Much research on traditional CDNs assumes that the network topology is given, for example, the tree structure which root is the origin server. However, in the cloud environment, the CDN builders have the freedom to build any topology. Thus, the problem of replica placement in cloud CDN is a joint problem of building distribution paths and replication.

The cost of replicas copy from site u to v is $d(u;v)$. For traditional CDN, the edges are usually undirected, i.e. $d(u;v) = d(v;u)$. However, the prices for uploading and download in storage cloud are different, which demonstrates that the edge is directed. This implies that only choosing a set of replica nodes is not enough; replication directions should be presented.

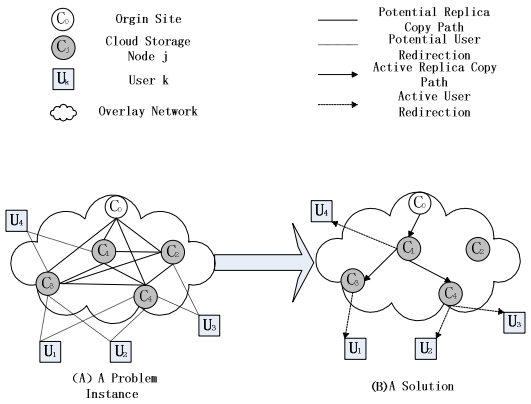

Figure 1. The Problem of Replica Placement in Cloud CDN

Replica placement problem in cloud CDN is given in figure 1.The potential replicas nodes are C_1 -- C_4. Some potential paths of distribution from C_0 are demonstrated by bold line. It is assumed that each nodes have the path to every users U_k .However, only a subset of these paths can satisfy the requirements of QoS for users request which are drawn as dashed line in figure 1(A). A kind of solution of replica placement is shown in Figure 1(B), where C_3 is chosen to provide service for to the request from U_1 and C_4 is chosen to provide service for the request from U_2 and U_3 . C_1 is chosen to provide service from U_4 as well as forward the replica from C_0 to C_3 and C_4 [18].

In Ref. [18], a off-line replica placement algorithm is proposed. Via trace-based study, it is shown that cloud CDN significantly reduces the cost of providing CDN service to small Web sites to as low as 2.62 US Dollar compared to the 99 US Dollar minimum charge by a traditional CDN. However the GS algorithm does not consider the load balance problem, which is assumed that the cloud CDN managers have both the information and the requests from the users. In this paper, two classes of off-line replica placement algorithms for cloud CDN were proposed, which provide service for online video service .One of the algorithms named GUCP could be used to place replicas when cloud CDN. It could solve the load imbalance problem caused by GS. The other named PBP could be used to place replicas while there is no information of users' requests in cloud CDN.

3.2 The Cost of Cloud CDN

Multiple storage cloud nodes copy replicas from the origin server C_0 , and then respond to all the edge users' requests. The cloud CDN uses multiple storage nodes or data centers, however, each storage node belongs to a certain provider. The cloud nodes from different providers may be co-local, but they may offer different service prices as well. It is assumed that there are m edge users $U = \{U_1, U_2, \cdots, U_m\}$ index by k , $(k = 1,2,\cdots m)$,and n cloud storage nodes $C = \{C_1, C_2, \cdots, C_n\}$ index by i or j , $(i, j = 1,2,\cdots n)$.

In this paper, it is assumed that the size of replica is T .The cloud storage nodes will change with the data storage, input and output traffic. Node C_j will charge unit storage cost of S_j per GB for storing the replica, P_j per GB for replica uploading traffic and D_j per GB for replica downloading traffic.

Let V_{uv} denote the cost of replica copied from node u to node v . V_{uv} has two different meanings which are depended on the nature of node v .

Case I: node v is the node of storage cloud node. V_{uv} is the cost of opening node, that is to say, the cost of node $v = C_j$ downloading replica from any node which has it. The origin service will update the replica in real time. In this paper, it is assumed that $F \cdot T$ is the content needs to be updated per unit time, where F represents the frequency of content update, $V_{uv} = V_{ij} = (S_j + P_j F + D_i F)T$. The node C_0 which has the original replica is assumed to be the origin server. The path of replica distribution will be a tree structure, the root node of which is C_0 , as is shown in figure 1(B).

Journal of Theoretical and Applied Information Technology
31st January 2014. Vol. 59 No.3

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

Case II: v in V_{uv} indicates a user, $v = U_k$, V_{uv} is the access cost of user who is assigned to node $u = C_j$ .The size of content which edge user U_k request is T_k and $V_{uv} = V_{ik} = T_k D_j$. T_k may be smaller than T or larger than or equal T , because of users did not request is not the whole replica or using web cache, or repeated request the without caching.

In summary, the cost of cloud CDN is the sum of the cost of replicas placed on the storage cloud node(i.e. open node) and the cost of service for the requests from users, $Cost = OpenCost + UserCost$ In this paper, the cost of uploading, downloading and storage of each node is defined randomly. The cost is measured by abstract monetary values (not a specific currency such as the U.S. dollar). The value of cost is used to compare the performance of different algorithms, but is not the real money in this paper.

3.3 The Definition of QoS in Cloud CDN

There is significant correlation between network delay and route distance. Because the actual hop count information between users and cloud sites is difficult to achieve, the geographic distance is considered as an indicator of delay. In this paper, the route distance is defined as the geographic distance between user and storage cloud node. Moreover, the Euclid distance is chosen as the route distance in our simulation experiments. In fact, the choice of distance metric does not impact the performance of our algorithms; any distance metric that is capable of describing the QoS requirement is applicable.

Let R_{jk} denotes the route distance between user U_k and node C_j . R_{jk} represents the communication quality between two nodes. The smaller R_{jk} (the distance from node to user) is, the better the quality of communication will be obtained. In this paper, the storage cloud nodes are utilized as replica servers. The abilities of bandwidth, reliability and concurrency of replica server nodes are consistent. The QoS distance could be represented by route distance. To satisfy the requirements of QoS from the users, it should be ensured that the route distance is less than QoS threshold Q , i.e. $R_{jk} \le Q$. The GS algorithm assigns the users to the nearest node.

The number of users is very large, and the total route distance will be increasing with the users. Hence, the average route distance is used to measure the QoS performance. The average route distance R is obtained by (1).

$$R = \sum_{j=1}^{n}\sum_{k=1}^{m}\sqrt{\left(X_{node}^{j} - X_{user}^{k}\right)^2 + \left(Y_{node}^{j} - Y_{user}^{k}\right)^2} \Big/ m \quad (1)$$

Where X_{node}^{j} and Y_{node}^{j} respectively denote horizontal and vertical coordinates of the node, which is assigned to users k , m is the total number of users.

3.4 The Load Balance of Cloud CDN

The load balance is an important performance in distributed system. To achieve the minimum of the running time, the distributed systems assign jobs according to the servers' performance.

In the process of online video service, the time of the connection between users and service nodes is very long. In order to serve more users with limited nodes, the improvement of the load balance performance of multiple video server nodes is a very important problem.

In this paper, it is assumed that the resource of load is the number of users served by cloud storage nodes, while the other resources (storage, bandwidth or CPU and so on） is not taken into account. The number of users which node provides service to is used to scale the performance of load. In this paper, we used the load weight [19] to describe the performance of load in cloud CDN.

It is assumed that there are n cloud storage nodes in cloud CDN. The ability of load of cloud storage node C_i , $i = 1,\cdots,n$ is ω_i , $i = 1,\cdots,n$. Based on the principle that the node load should match with its load ability, We assume that the ability of cloud storage nodes is the same, i.e.

$$\omega_1 = \omega_2 = \cdots = \omega_n = \omega \quad (2)$$

It is assumed that the load of C_i is L_i , load weight is defined as the ratio of current load and ability of load.

$$W_L^i = \frac{L_i}{\omega_i} = \frac{L_i}{\omega} \quad (3)$$

When the load balance is achieved,

$$W_L^i = W_L^j , \quad i \ne j; i, j = 1,2,\cdots,n \quad (4)$$

The average load weight is defined as formula (5).

$$\bar{W}_L = \frac{1}{n}\sum_{i=1}^{n} W_L^i = \frac{1}{n\omega}\sum_{i=1}^{n} L_i \quad (5)$$

Load weight could be represented by formula (6) when the cluster is balancing.

$$W_L^1 = W_L^2 = \cdots = \bar{W}_L = \frac{1}{n}\sum_{i=1}^{n} W_L^i = \frac{1}{n\omega}\sum_{i=1}^{n} L_i \quad (6)$$

For both n and ω are fixed, the load weight

Journal of Theoretical and Applied Information Technology
31st January 2014. Vol. 59 No.3

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

could be represented approximately as formula (7) while the cluster is balancing.

$$L_1 = L_2 = \cdots = L_n = \overline{W_L}' \quad (7)$$

The load of a node is associated with its uses in formula (7). The load coefficient is defined to evaluate the load performance. In this paper, the mean square deviation of load is used to describe the load coefficient of cloud CDN. The load coefficient η is defined as formula (8).The smaller η is, the load is better.

$$\eta = \frac{\sum_{i=1}^{n}(L_i - \overline{W_L}')^2}{n} \quad (8)$$

3.5 The Other Performance of Cloud CDN

Performances mentioned above can be used to measure cloud CDN with information of users' requests. Unfortunately, it fails when there is no information of users' request to use.

After replicas are placed on the nodes randomly, requests from users will be redirected to the nearest nodes. If the redirected node has no replica which is needed, the node will download it from the other nodes. The process of copying replicas from the other nodes will result in additional cost and time delay.

Let $Push\_Cost$ denote the additional cost, and it can be represented as:

$$Push\_Cost = \sum_{i=1, j=1, i\neq j}^{n} \sum_{l=1}^{L} T_l(S_j + P_j + D_i) \quad (9)$$

In (9), j denotes the node which downloads and stores the replica from the other nodes, and i denotes the replica provider. T_l is the size of the replica.

Let $DELAY_{ij}$ denote the time delay and Avg_Push_Delay denote the average time delay. It is considered that we should download replicas from node j to i when $DELAY_{ij}$ is nonzero, and M is the time of downloading. It is assumed that the value of $DELAY_{ij}$ is the route distance between node i and j. Avg_Push_Delay can be obtained by (10).

$$Avg\_Push\_Delay = \sum_{i=1, j=1, i\neq j}^{n} DELAY_{ij} / M \quad (10)$$

4. THE VIDEO REPLICA PLACEMENT ALGORITHM WITH REQUESTS INFORMATION

In this paper, cloud CDN investigated is for online video. The content of the online video has large file size and low updating frequency. The users must be responded to as soon as possible, hence, the video replica should be placed on the edge nodes.

The cloud CDN may either have or not the information of users' requests. On the one hand, if the cloud CDN has the information of users' requests, it could be used to push the content to the edge nodes; On the other hand, if cloud CDN has no requests information, PBP (popular based placement) algorithm could is used to place the replica based on the content popular.

4.1 GS Algorithm

The GS(Greedy Site) algorithm is proposed in [18],GS is adopted from an approximation algorithm for the Set Covering Problem[20]. GS iteratively decides to open a closed site which has the maximum utility and assign all its potential users to it. A potential user of a node has minimum route distance and it has not been assigned to any node. The Cloud CDN opens a storage cloud node, and then finds the next one to open, until all of the users are assigned to a certain node. Let O_j denote the cost to open C_j.

$$O_j = \begin{cases} 0 & if C_j is OPEN \\ \min_{i\in\{i|C_i isopen\}} V_{ij} & otherwise \end{cases} \quad (11)$$

The algorithm 1 is the pseudo-code of GS.

Algorithm 1 : Greedy Site [18]

E **is the set of user who have not been assigned**
E_j **is the current set of users who can be assigned to** C_j
while $E \neq \varnothing$ **do**
$W_j \leftarrow \sum_k w_k, U_K \in E_j$
$j^* \leftarrow \arg\max_{j\in\{C_j isCLOSED\}} \frac{W_j}{W_j D_j + O_j}$
Assing all users in E_{j^*} **to** C_{j^*}
Open C_{j^*}
$E \leftarrow E - E_{j^*}$
end while

4.2 The Description of Load Imbalance Problem

The assignment of users in GS is operated according to the route distance , which may lead to load imbalance.

It is assumed that there are 20 nodes and 600 users in the location where the replicas should be placed. Load imbalance will come from allocating users to nodes using GS. The load imbalance may occur when the GS algorithm is used. The load

Journal of Theoretical and Applied Information Technology
31st January 2014. Vol. 59 No.3

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

coefficient of cluster is $\eta = 1411.71$, the maximum and minimum load of node are $L_{Max} = 155$ and $L_{Min} = 54$ respectively.

The load threshold defined in Ref. [21] is $Th = (1+\overline{L})/2$, where $\overline{L}$ is the average load of all the nodes. Unfortunately, the definition is not suited for the cloud CDN, which is proved by experiments. We define the new dynamic load threshold of cloud CDN as (10), where α is the imbalance factor taking values in (0,1]. The value of α will be given in section VI by experiments.

$$Th = \overline{W_L}' * \alpha \quad (12)$$

The users of nodes should be adjusted to ensure the load balance, when $L_p > L_q$ and $L_p - L_q \geq Th$ and $p \neq q; p,q = 1,2,\cdots,n$. To solve the load imbalance caused by GS, A new algorithm named GUDP(Greedy User Core Preallocation) is proposed.

4.3 GUCP Algorithm

The GUCP（Greedy User Core Preallocation）will adjust the assignment when load imbalance appears, after assigning users to the nearest nodes. Some users of node p should be assigned to q, while $L_p - L_q \geq Th$, $p \neq q;$ $p,q = 1,2,\cdots,n$. It is assumed that the coordinate of node is uniformly distributed.

K-means algorithm is a normal cluster algorithm. It is a clustering method based on statistics. K-means clustering is a method of vector quantization originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering (MacQueen, 1967) is a method commonly used to automatically partition a data set into k groups. It precedes by selecting k initial cluster centers and then iteratively refining them as follows [22]:

1. Each instance U_i is assigned to its closest cluster center.
2. Each cluster center $CORE_j$ is updated to be the mean of its constituent instances.

Firstly, K-means algorithm is adopted to accomplish the clustering of the users. Then, about $(L_p - L_q)/2$ users should be picked from node p and added to q. K-means algorithm could be used to cluster the users of node p, and move users belonging to p and nearest to q in batches.

But it is difficult to get a suitable value of k for different data sets. In this paper, we want to move about $(L_p - L_q)/2$ users because the users are uniform distribution at the location need to place replicas. The number of cores in k-means can be given by (13):

$$N_k = 2L_i/(L_p - L_q) \quad (13)$$

A core $CORE_k$ is selected from N_k cores which are clustered by k-means， and it is nearest to q. Then, we move the users that correspond to $CORE_k$ to q. Then we check whether the system is load imbalance again. If load imbalance, we will repeat the operation of users assigning adjustment until load balance. The core idea of GUCP is using k-means to cluster the users which are assigned to the over load node, and then adjusting the users by the cores to load balance of cloud CDN. The system achieves balance quickly because of the frequency of adjustment is reduced by this algorithm. The algorithm 2 is the pseudo-code of GUCP:

Algorithm 2 : Greedy User Core Preallocation

E **is the set of user who have not been assigned**
E_j **is the current set of users who can be assigned to** C_j
while $L_p - L_q \geq Th$（$p \neq q; p,q = 1,2,\cdots,n$） **do**
  N_k **cores** $\leftarrow$ **Kmeans (**U_p**)**
  Find min distance $Core\_num$ **between** C_q **and cores**
  Assigned users U_{core_num} **to** C_q
end while
while $E \neq \varnothing$
$W_j \leftarrow \sum_k w_k, U_K \in E_j \quad j^* \leftarrow \underset{j \in \{C_j isCLOSED\}}{\arg\max} \frac{W_j}{W_j D_j + O_j}$
Assing all users in E_{j^*} **to** C_{j^*}
  Open C_{j^*}
  $E \leftarrow E - E_{j^*}$
end while

5. THE VIDEO REPLICA PLACEMENT ALGORITHM WITHOUT REQUEST INFORMATION

5.1 The Popularity of Video content

In the last section, a video replica placement algorithm for cloud CDN is proposed for cloud CDN with information of users' requests. However, if cloud CDN has no requests information, the presented algorithm in the last section fails. Hence， in this section，a new algorithm based on popularity of video replica is proposed for cloud CDN without requests information. Popularity of video contents

The quality of service of CDN is affected by the placement strategy of the copies [23].on the

Journal of Theoretical and Applied Information Technology
31st January 2014. Vol. 59 No.3

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

one hand, If the replicas are placed more than they are needed, the storage space is wasted. On the other hand, when the copies are placed too few, then it will result in low service quality. Although the CDN can adjust the number of copies itself, it will cost a lot of time and much operation of I/O.

The popularity of replica is an important basis for placement. Methods of decision tree and neural network model to predict the popularity of movies were proposed in Ref. [24] and Ref. [25]. These methods can be used in cloud CDN when it has no requests information.

The distribution of popularity of replica is assumed as Zipf distribution in Ref. [26] and Mandelbrot-Zipf distribution in Ref. [27] , but the distribution of video popularity obeys neither of them. s. The extensive law model is suitable for the distribution of popularity of movie [28]. The 17220 pieces of movies' request information are got from http://movie.youku.com in 2013. It is shown that the use of stretched exponential model on income popularity data fitting and found more in line with the stretched exponential model in figure 2. It proves that the distribution of video popularity is a stretched exponential distribution. The probability density function (PDF) of stretched exponential is shown in (12). The parameters of the video popularity distribution function can be got , which is stretched exponential $c = 0.33$ and $x_0 = 27$ 。

$$p(i) = c\left(\frac{i^{c-1}}{x_0^{\,c}}\right)\exp[-(\frac{i}{x_0})^c]\,,\quad i = 1,\cdots,N \tag{12}$$

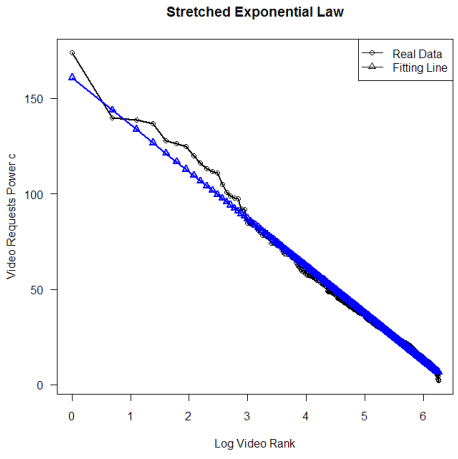

Figure 2. Fitting Of Data And Stretched Exponential Model

5.2 The Relationship between Number and Popularity of Replicas

It is assumed that cloud CDN has N nodes, which have been opened. Let NUM_i denote the number of replicas of content i and POP_i denote the popularity of video content i . It is known to us that the more popular the video content is, the more replicas ars needed. Hence it is assumed that the most popular content's replicas are placed on all of the storage cloud nodes, and then, the number the most popular content the replicas is N . It is assumed that the ratio of POP_i and MAX_POP is equal to the ratio of NUM_i and N .

$$NUM_i = N * POP_i / MAX\_POP \tag{13}$$

5.3 PBP Algorithm

To place the replicas suitably with requests information, a new algorithm named PBP (Popularity Based Placement) is proposed.

Let T denote the content set and T_i denote the content of k .PBP utilizes popularity of video content to place the replicas. For every content in T , The popularity of T_i is given first, then calculate the number of replicas of T_i NUM_i by (13) lastly , select the nodes are selected with minimum cost from the opened nodes to place the replicas. Algorithm 3 is the pseudo-code of PBP:

Algorithm 3 : Popularity Based Placement

T **is the set of content which have not been placed**
C **is the current set of nodes which can be placed replica**
N **is the number of cloud storage nodes**
for($i = 0\,; i < T.length()\,; i++$ **)**
  $POP_{T_i} \leftarrow$ ***Get_Popularity(*** T_i **)**
  $NUM_i \leftarrow N * POP_i / MAX\_POP$
  $C' \leftarrow C$
  for($j = 0\,; j < NUM_{T_i}\,; j++$ **)**
    $k \leftarrow \arg\min_{k \in C'}(T_k D_k)$
    Place T_i ***on*** C_k
    $C' \leftarrow C' - C_k$
  end for
end for

5.4 Random Algorithm

The PBP algorithm gives the number of replicas by contents popularity. In Ref [29], random algorithm is used to place the replicas. In this paper, random algorithm is chosen to compare

Journal of Theoretical and Applied Information Technology
31st January 2014. Vol. 59 No.3

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

with the PBP. NUM_i is given in random algorithm as (14).

$$NUM_i = rand()\%(N+1) \quad (14)$$

6. EXPERIENCE AND DISCUSSION

In this section, we present the parameters for numerical experiments in section A, and then

6.1 Parameters of Simulation

In this section, numerical experiments are performed to evaluate the performance of the proposed algorithms in Matlab. It is assumed that there are 20 nodes whose cost of uploading, downloading and storage are random numbers taking values in $(0,10]$. The cost values are assumed to be different from each other and the users is less than 1000. The bandwidth of each node is 10 Mbps; the users are randomly distributed in a ring, of which the inside diameter is 10 and the outside diameter is 20. When every user watches video，an average of 100Kb is occupied.20 nodes can serve 2000 users at most. Hence, the simulation experiment via 1000 users will not cause overload. The number 20 reflects the current status of the number of cloud storage providers is not much. The number 20 reflects the fact that there are not so many cloud storage providers. However, even if more than 20 nodes are chosen, it has little influence on the final results.

The value of imbalance threshold has influence not only on the value of load coefficient η , but also on the frequency of adjustment when load imbalance occurs. Reasonable threshold value can ensure less frequency of adjustment while the load coefficient η is smaller. To get the value of dynamic imbalance factor α ,we assume that there are 600 users using the cloud CDN. The evolution of the load coefficient and the adjustment frequency are shown in figure 3 and 3 respectively, as α varies from 5% to 99% .

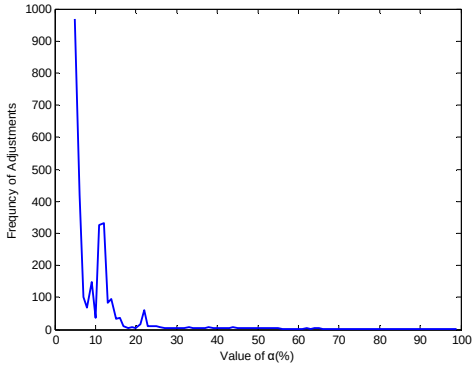

Figure 3. The Relationship of Load of α and Number of Adjustments

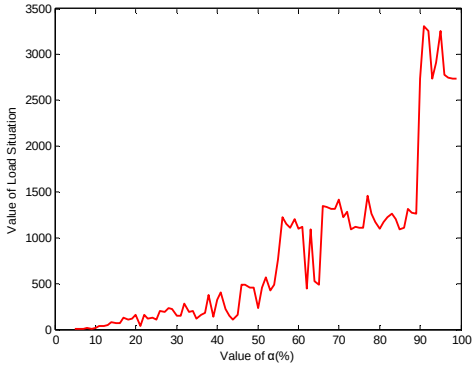

Figure 4. The Relationship of α and Load Coefficient

From the figures 3 and 4, we can see that at the point 25% , the adjustment frequency has decreased to some acceptable extent, besides, the load coefficient is not very large. Hence, α is chosen as 25% ,consequently, the load threshold $Th = \overline{W_L}' * 0.25$.

According to the distribution of nodes and users, It is assumed that the average distance between users and nodes is 10, the threshold of QoS can be chosen as 10, the upper limit of route distance is 10. The parameters of numerical experiments are listed in table I.

Table1. Parameters of Numerical Simulation

Experiment Parameter Type	Nodes	Users	Contents number	Contents size	Q	α
Value	*20*	*100 ~1000*	*100*	*0~10*	*10*	*25%*

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

6.2 Result of Numerical Experiment for Algorithm with Requests Information

Three sets of experiments are carried out, with the users being a variable factor. GUCP is compared with GS in load, cost and average route distance, respectively, as the users is varying from 100 to 1000.

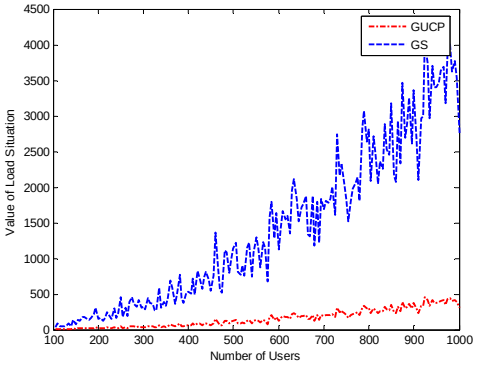

Figure 5. The Comparison of Load Coefficient

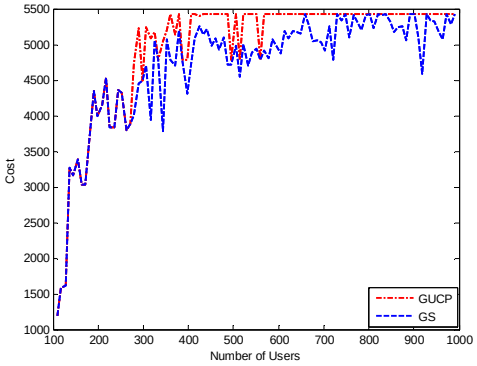

Figure 6. The Comparison of Cost

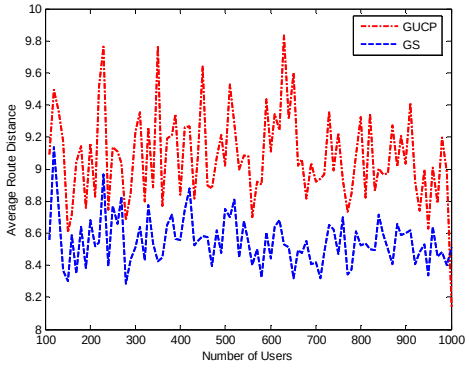

Figure 7. The Comparison of Average Route Distance

The comparison of load coefficient between GS and GUCP is shown in figure 5, which implies that the latter one has much advantage over the former in load balancing abilities.

In figure 6, we see that more cost should be pain in the algorithm GUCP compared with GS, which is also acceptable for us. In figure 7, we see that the average route distance is little lager in GUCP than in GS, but it is obviously still acceptable.

6.3 Result of Numerical Experiment for Algorithm without Requests Information

In the case that there is no users' request information to utilize, the PBP algorithm is to be chosen. Two sets of experiments are carried out, with the users being a variable factor.PBP is compared with Random in push cost and average delay.

In figure 8 and 9, we see that less average delay and push cost is smaller in PBP than in Random. Hence, from the comparison, it is conclude that the PBP is more effective.

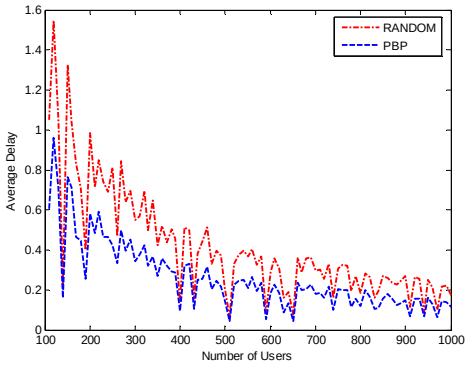

Figure 8. The Comparison of Average Delay

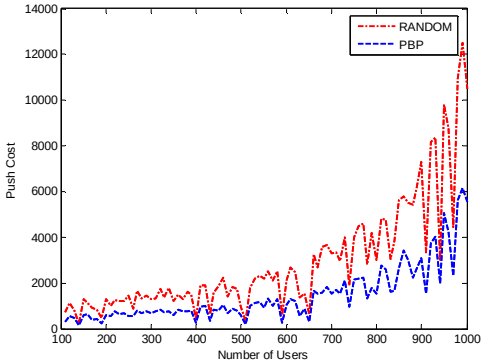

Figure 9. The Comparison of Push Cost

7. CONCLUSIONS

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

In this paper, two offline replica placement algorithms are proposed for cloud-based storage CDNs. The GUCP effectively solved the load imbalance problems in replica placement compared with the existed GS algorithm. When there is no information of users' requests, a new algorithm called PBP is proposed based on the popularity of video content. It is shown that the PBP has much advantage in average delay and push cost compare with the Random algorithm. Numerical experiments have demonstrated the effectiveness of the method above.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper. This work was supported in part by National Key Technologies R&D Program of China (Grant No. 2012BAH73F02) and the "Strategic Priority Research Program" of the Chinese Academy of Sciences(Grant No. XDA06030900).

REFERENCES

[1] Rayburn D. CDN pricing: Costs for outsourced video delivery[J]. Streaming Media West, 2008.

[2] Broberg J. Building Content Delivery Networks Using Clouds[J]. Cloud Computing: Principles and Paradigms, 511-531.

[3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski,G. Lee, D. Patterson, A. Rabkin, I. Stoica et al., "Above the clouds:A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.

[4] Elson J, Howell J. Handling flash crowds from your garage[C]//USENIX ATC. 2008, 4.

[5] D.Gottfrid.Self-service,prorated superc omputing fun! OPEN： All the code tha is ft to printf().http://open.mytime.com/2007/11/01/self-service-prorated-super-com-putting-fun/,1/11/2007

[6] MacAskill D. Scalability: Set Amazon's servers on fire, not yours[C]//ETech 2007: O'Reilly Emerging Technology Conference. 2007.

[7] 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.

[8] J. Broberg, S. Venugopal, and R. Buyya, "Market-oriented grids and utility computing: The state-of-the-art and future directions," Journal of Grid Computing, vol. 6, no. 3, pp. 255–276, 2008.

[9] Neves T A, Drummond L, Ochi L S, et al. Solving replica placement and request distribution in content distribution networks[J]. Electronic Notes in Discrete Mathematics, 2010, 36: 89-96.

[10] Li B, Golin M J, Italiano G F, et al. On the optimal placement of web proxies in the internet[C]//INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. IEEE, 1999, 3: 1282-1290.

[11] Krishnan P, Raz D, Shavitt Y. The cache location problem[J]. IEEE/ACM Transactions on Networking (TON), 2000, 8(5): 568-582.

[12] Qiu L, Padmanabhan V N, Voelker G M. On the placement of web server replicas[C]//INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. IEEE, 2001, 3: 1587-1596.

[13] Jia X, Li D, Hu X, et al. Placement of web-server proxies with consideration of read and update operations on the internet[J]. The Computer Journal, 2003, 46(4): 378-390.

[14] Xu J, Li B, Lee D L. Placement problems for transparent data replication proxy services[J]. Selected Areas in Communications, IEEE Journal on, 2002, 20(7): 1383-1398.

[15] Cidon I, Kutten S, Soffer R. Optimal allocation of electronic content[J]. Computer Networks, 2002, 40(2): 205-218.

[16] Kalpakis K, Dasgupta K, Wolfson O. Optimal placement of replicas in trees with read, write, and storage costs[J]. Parallel and Distributed Systems, IEEE Transactions on, 2001, 12(6): 628-637.

[17] Papagianni C, Leivadeas A, Papavassiliou S. A Cloud-oriented Content Delivery Network Paradigm: Modeling and Assessment[J]. 2013.

[18] Chen F, Guo K, Lin J, et al. Intra-cloud lightning: Building CDNs in the cloud[C]//INFOCOM, 2012 Proceedings IEEE. IEEE, 2012: 433-441.

[19] Guo Cheng-cheng Yan Pu-liu.A Dynamic Load-bal anci ng Al gorit hmf or Heterogeneous Web Server Cluster[J].Chinese Journal of Computers, 2005,28(2):179-184

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

[20] Chvatal V. A greedy heuristic for the set-covering problem[J]. Mathematics of operations research, 1979, 4(3): 233-235.
[21] Godfrey B, Lakshminarayanan K, Surana S, et al. Load balancing in dynamic structured P2P systems[C]//INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. IEEE, 2004, 4: 2253-2262.
[22] Wagstaff K, Cardie C, Rogers S, et al. Constrained k-means clustering with background knowledge[C]//ICML. 2001, 1: 577-584.
[23] V.S.RAMASUBRAMANIAN,"Cost-Aware Resource Management for Decentralized tralized Internet Services," PhD thesis, Cornell University, January 2007.
[24] Shijia Y, Liuzhang Z, Ming Z. The popularity of movies predict system based on data mining technology for CDN[C]//Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. IEEE, 2010, 7: 604-607.
[25] Li J, Hong S, Xia S, et al. Neural Network Based Popularity Prediction For IPTV System[J]. Journal of Networks, 2012, 7(12): 2051-2056.
[26] Zhou X, Xu C Z. Optimal video replication and placement on a cluster of video-on-demand servers[C]//Parallel Processing, 2002. Proceedings. International Conference on. IEEE, 2002: 547-555.
[27] Hefeeda M, Noorizadeh B. On the benefits of cooperative proxy caching for peer-to-peer traffic[J]. Parallel and Distributed Systems, IEEE Transactions on, 2010, 21(7): 998-1010.
[28] Zhuo You,"VOD system access behavior research: measurement, analysis and modeling," PhD thesis, University of Science and Technology of China, January 2007.
[29] Borst S, Gupta V, Walid A. Distributed caching algorithms for content distribution networks[C]//INFOCOM, 2010 Proceedings IEEE. IEEE, 2010: 1-9.