An Improved Content Delivery Technology Based on Genetic Algorithm in Cloud Storage

Pu Zhao, Jingtao Shang, Qian li, Mingyuan Ma, Xin Sun

1Network and Information Center China North Vehicle Research Institute Beijing, China, 100072
2School of Computer science & technology Beijing Institute of technology Beijing, China, 100081

azhaojialipu@126.com, bsunxin@bit.edu.cn

Abstract. Compared with the traditional content delivery network (CDN), content delivery based on cloud storage has the advantage of reducing storage costs while providing superior data security over common storage solutions. There has been a lot of research on content delivery technology, but existing research on content delivery technology is rarely used in cloud storage based CDN. In this paper, we analyze the charging and content delivery mechanism of cloud storage, then propose a reasonable network topology and cost model. Based on the analysis of existing heuristic algorithms such as user greedy and server greedy, an improved content delivery technology based on genetic algorithm in cloud storage is proposed, which effectively reduces the content delivery cost.

1. Introduction (Heading 1)

Compared with the traditional CDN, cloud storage-based CDN has a great price advantage, it can provide network content providers with low-cost virtual host to achieve content delivery services, instead of taking a lot of money to buy equipment. However, the content delivery technology research in the existing CDN does not consider the network topology and charging mechanism of the cloud storage, so it can rarely be applied to the cloud storage based CDN.

The cloud storage system consists of thousands of servers and the geographical location is different. How to realize the effective organization of a large number of servers is one of the key issues for the efficient and stable operation of cloud storage systems. Therefore, the network topology is an important component of the cloud storage platform, a reasonable network topology and resource management model can not only improve the performance of the network, but also ensure the effectiveness of resource allocation and load balancing.

Based on this, an improved genetic algorithm for the Cloud-CDN (IGA-CCDN) is proposed, it abstract the content delivery problem in cloud storage into a mathematical model of the set solution problem. The improved genetic algorithm solves the problem and finally restores the obtained optimal solution to the content delivery scheme. Experiments show that the algorithm can effectively find the optimal solution and reduce the delivery cost.
2. P2P Tree resource management model

Some typical cloud storage systems use a central index to manage data and resources. Centralized management is easy to design and manage, but too many frequent managements can easily cause bottlenecks. These problems have always existed in the resource management of distributed systems, and it is difficult to manage large-scale, distributed, heterogeneous storage resources. At present, the topology structure adopted by the node's resource distribution model is not completely the same due to different application scenarios. There are several types of summing up: tree, layered, centralized, and irregular structure.

Based on [2], we add a combination of existing proxy mechanisms to build a tree resource management model based on P2P. In a plurality of data centers across regions, the nodes are first divided into several domains according to the physical distance between the nodes. A tree structure managed by the super nodes is adopted in the domain, and a P2P connection structure is adopted between domains. A node is selected in the domain to deploy the node agent.

In this model, the topology of the cloud computing is optimized by the scalability and low cost of P2P, the system bottleneck is eliminated, and the scalability of the cloud is provided. Based on the proposed network topology, the following paper proposes how to apply CDN technology to the cloud storage network topology to realize content delivery and improve the efficiency of cloud storage services.

2.1. P2P Tree structure description

The nodes in the network are classified into a super node (Super Peer, SP) and a normal node (Normal Peer, NP) according to the difference in network bandwidth and processing capability of the node. Normal nodes can share their own resources and access other resources in the system. A super node is assumed by a node with long online time, high bandwidth, and good performance. In addition to accessing other resources in the system and sharing its own resources, the super node also assumes the normal operation and management functions of the system maintenance. According to the performance of the node, the super node is further divided into a first level super node (Level-1-SP) and a second level super node (Level-2-SP). The super node is responsible for collecting

![Figure 1. P2P tree resource management model](image)

And storing the resource load of the nodes in the domain, and performing resource allocation and management according to the current node resource load status.

The basic idea of the P2P tree resource management model is introduced below. First, the node is divided into several domains according to the physical distance between the nodes. The physical
distances of all nodes in the domain are similar. If the queried resources can be found in this domain, network delay and bandwidth consumption can be minimized. Secondly, a node is selected as a Level-2-SP in each domain, and a mode of centralized is adopted. Other common nodes are directly connected with the Level-2-SP, and a centralized management method makes resource positioning more efficient. Then, the domain is managed by a Level-1-SP, which is connected to the Level-2-SP in the domain. The Level-1-SP has the responsibility of the domain's management, forwards the resource location message, and responds to the node request in the domain. Finally, the Level-1-SPs that manage each domain are connected, and the P2P tree resource management model is implemented.

The P2P tree resource management model combines the characteristics of P2P and C/S structures. The management of resource location messages adopts a combination of centralized and distributed methods. On the one hand, according to the forwarding of the parent-child relationship positioning message, the Level-2-SP centrally manages the message transmission in the domain; on the other hand, the propagation positioning request between the Level-1-SP and Level-1-SP, and between the Level-1-SPs are required when the current domain can’t locate the resource.

2.2. Resource distribution model

P2P emphasizes the full sharing of resources in a peer-to-peer manner. All participating nodes have the same responsibilities and capabilities. However, the processing ability and behavior of nodes on the actual network are quite different. In the construction of P2P tree resource management model, the heterogeneity of nodes needs to be considered, and the more effective network structure can be obtained by rationally utilizing the different capabilities of different nodes.

How to choose a super node is the first problem to be considered in the model building process. The selection of the super node needs to meet the requirements of high bandwidth, high performance, and long online time. Because the super node acts as a node in the P2P network, it not only completes the purpose of sharing its own resources and accessing other resources of the system, but also needs to be responsible for the normal operation of the system management. In one domain, according to the difference in node performance, the node with the best performance is selected as the Level-1-SP, which is connected to the Level-1-SP in other domains through network interconnection. The node with the second performance is selected as the Level-2-SP, and the C/S mode is used to connect with other common nodes. In this way, the Level-1-SP and the Level-2-SP are connected, and the resource management model is completed. The resource management of the model is mainly considered from two levels. The underlying C/S mode and the top-level P2P mode. In the underlying C/S mode, the Level-2-SP is connected to each normal node, and the Level-2-SP has the node information directory (including the node's IP address, basic configuration information, history, etc.) in the entire domain, and replicate in every normal node. The nodes use the timed mutual communication test, when the normal node sends a message and the Level-2-SP responds, it means the Level-2-SP is under normal conditions. Otherwise, the Level-2-SP loses effectiveness. At this time, it is necessary to replace the Level-2-SP with the normal node of the test, complete the connection with the Level-1-SP and the normal node, and update the information directory of all the nodes. Similarly, the Level-2-SP also needs to periodically detect the operation of the normal node. Compared with the failure of the Level-2-SP, the normal node only needs to transfer resources and update the information directory of the nodes in the domain.

The P2P mode at the top level not only needs to manage the node information of the domain in which it is located, but also manages the topology structure between the domains, which is more complicated in resource management. The Level-1-SP needs to maintain two node directory information, one is directory information of all node information in the domain connected to the Level-2-SP, and the other is directory information between the Level-1-SP, which includes the distribution and connection of all Level-1-SPs in the network. The validity test of the Level-1-SP interconnected is periodically. If there is an invalid node, the Level-2-SP is detected as a substitute for the invalid node. If the Level-2-SP invalid, the message is automatically sent to update the node information directory of Level-1-SP.
Therefore, in the P2P tree resource management model, node joining and deleting will not have a great impact on the entire network topology. For a normal node, it only needs to connect to the Level-2-SP of the domain in which it resides. The joining and exiting of the node are only performed in the domain, and does not affect the topology of other domains. Only the node information addition or deletion of the Level-2-SP and Level-1-SP. For the addition of a domain, only the Level-1-SP of the connection domain needs to be connected, and the positioning information of the Level-1-SP needs to be modified.

2.3. Content delivery process in cloud storage
The following takes a scenario of dynamic content delivery in a cloud storage network model as an example to illustrate the content delivery process, as shown in Fig. 2.

Assume that the normal node N2 is the source server, and the so-called source server refers to the original storage location of the resource, that is, the server where the original replica of the resource is located. The user is assigned to the nearest domain through load balancing DNS communication, and obtains the address of the Level-1-SP at the domain, and queries the resource metadata list in the Level-1-SP to check whether there is a replica of the requested resource, if there is, then obtain the address of the node that stores the replica from the list, and get the resource content through it. If there is no replica of the requested resource in the domain, the Level-1-SP of the domain sends a request to the Level-1-SP of the domain where the resource's source server is located by parsing the URL of the user, obtains the address of the source server, and selects one node of the request domains to send a resource request to the source server, and the source server delivers the content to the node, and the node provides the resource service to the user.

The specific distribution steps are:
1) User U1 issues a resource request to the load balancing DNS.
2) Load Balancing DNS assigns the appropriate domain to the user according to the proximity principle and the load balancing algorithm, and returns the address of the domain's Level-1-SP S1.
3) The user sends a resource request to S1, and S1 checks his own metadata list to determine whether there is a resource cache replica locally.
4) If the local cache replica exists, the address of local storage node N1 where the replica located in is returned to the user, and the resource service is directly provided by the local node N1.
5) If there is no cache replica locally, S1 parses the user URL, sends a request to the source server N0, and establishes a distribution path between the local server and the source server.
6) Specify the local node N1 as the cache server of the resource replica, deliver the resources from the source server to the local cache server, determine the cache policy of the deliver through the cache technology, and provide the resource service to the user.

3. Content delivery cost model in cloud storage

The cost model is an important indicator of content delivery technology in cloud storage. Based on the research of CDN delivery process and the characteristics of cloud storage charging mechanism, we propose a content delivery cost model in cloud storage. Take the content delivery model in Fig.3 and Fig.4 as an example, assume that each node has a path to any user. The user communicates with the load balancing DNS server to obtain a Level-1-SP that can respond to the service request, and the affiliation is indicated by a broken line as shown in Fig.3 and Fig.4 illustrate an example scenario of a content delivery path, where S1 is responsible for processing service requests sent from U1 and U2, and S2 is responsible for processing service requests sent from U3. S4 is responsible for processing the service request sent from U4, and is responsible for forwarding the content of the resource replica to the server of the domain where S1 and S5 are located. The total cost of the example includes: uploading, downloading, and storage charges for the normal replica nodes of the Level-1-SPs S1, S2, S3, S4, and S5. For a normal replica node (for example, N1) that provides a resource download service for users, the upload cost is generated by the uploading traffic generated when the node requests resources from the source server resource, and the download cost is generated by providing resources for the user to download. We regard the source server N0 as a normal node in the cloud storage, and the upload cost is generated by the uploading fee when the content upload to the cloud storage, and the download cost is generated from the outflow of content deliver from N0 to N1. For a Level-1-SP that forwards a replica to other nodes, such as S2, the download cost is also generated by the outgoing traffic generated by forwarding the replica to other super nodes. Storage charges are generated from the storage of the replica, including the storage of replicas on the source server N0 and all replica nodes (N1, etc.). The cost associated with the user request is ignored here, mainly because the cost of the user request is negligible relative to the amount of data delivery by the content. As video traffic begins.
To dominate the distribution of content, the proportion of user requests will become less and less. Therefore, the content delivery cost model in cloud storage is given as: the Level-1-SP in cloud storage is represented as $S = \{S_1, S_2, ..., S_n\}$, and the terminal user is expressed as $U = \{U_1, U_2, ..., U_m\}$, the size of the replica is expressed as $W$, the fee charged per GB on the cloud storage node is expressed as $C_j$, the fee charged by the node $j$ per GB of uploading traffic is expressed as $D_j$, and the fee charged by the node $j$ per GB of downloading traffic is expressed as $P_j$, the cost of forwarding content between nodes $u$ and $v$ is expressed as $V_{uv}$.

If $U$ is the source server and $V$ is the Level-1-SP in the cloud storage, the calculation method of the content of forwarding content between $U$ and $V$ is as shown in (1).
If $U$ and $V$ are both the Level-1-SPs in the cloud storage, the calculation method of the cost of forwarding the content between $U$ and $V$ is as shown in (2).

$$V_{uv} = (P_u + P_v) \cdot W$$

(2)

If $V$ is a cloud storage node, $U$ is a user terminal, the method for calculating the cost of forwarding content between $U$ and $V$ is as shown in (3).

$$V_{uv} = (N_u + B_u) \cdot W$$

(3)

After defining the above variables, an optimized objective function can be obtained, as shown in (4).

$$\min \sum_{(u,v) \in S} y_{uv} V_{uv}$$

(4)

Where $y_{uv}$ indicates whether a delivery path is established between $u$ and $v$.

### 4. Improved content delivery technology based on genetic algorithm in cloud storage

#### 4.1. Mathematical model establishment

First, the mathematical model of content delivery in cross-domain cloud storage is abstracted. Let $A= (a_{ij})$ be a 0-1 matrix of $m$ row and $n$ columns. The row corresponds to the set of users in the content delivery network topology in cloud storage, denoted as $i \in M, M=\{1,2,\ldots,m\}$; the column corresponds to the set of edged Level-1-SPs in the network topology, denoted as $j \in N, N=\{1,2,\ldots,n\}$. The cost of the column is $C=(c_j), j \in N$, where $c_j$ represents the cost of column $j$, that is, the cost of delivering content to the edged Level-1-SP represented by column $j$, with $c_j>0, j \in N$. If $a_{ij}=1$, it indicates that row $i$ is covered by column $j$, that is, the user represented by row $i$ can be served by the edged Level-1-SP represented by column $j$; if $a_{ij}=0$, it indicates that row $i$ is not covered by column $j$, that is, The user represented by row $i$ is not served by the edged Level-1-SP represented by column $j$. Since the purpose of content delivery in cloud storage is to deliver content to the appropriate edged Level-1-SPs, all users can be provided of resource services by edged Level-1-SPs directly while minimizing delivery costs. So, the set covering problem (SCP) abstracted in this paper is to find an optimal solution $X (X \subseteq N)$, which is a set of columns, represents a content delivery solution. So that each row in $M$ can be covered by at least one of the columns in $X$, that means, each user can be served by one of the edged Level-1-SPs in the delivery solution, and the sum of the costs of all the columns in the solution $X$ is required to be the smallest, that is, the delivery cost is the smallest. The abstracted SCP mathematical model is as follows:

$$\min \sum_{j=1}^{n} c_j x_j$$

(5)

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \quad i = 1,2,\ldots,m$$

$$x_j \in \{0,1\}, \quad j = 1,2,\ldots,n$$
The objective function (5) indicates that the total delivery cost is the smallest; the constraint \( \sum_{t=1}^{n} a_{tj}x_{j} \geq L \) means that each user can overlaid by at least one edged Level-1-SP; in the constraint \( x_{j} \in \{0,1\}, \quad j = 1,2,\ldots,n \), \( x_{j}=1 \) indicates that column \( j \) is included in solution \( X \), and the content will be delivered to the edged Level-1-SP represented by column \( j \), and \( x_{j}=0 \) means that the column \( j \) is not in the solution \( X \), that is, the content will not be delivered to the edged Level-1-SP represented by the column \( j \). If all \( c_{j} (j \in N) \) are the same, this is the set covering problem without weights. If \( c_{j} (j \in N) \) is not the same, this is the set covering problem with weights. The weight \( c_{j} \) corresponds to the content delivery model in cloud storage, which is the cost of delivery content to the edged Level-1-SP represented by column \( j \).

4.2. Improved genetic algorithm for the Cloud-CDN

Content delivery is essentially to solve the NP-hard set covering problem. The genetic algorithm has good adaptability to the NP-hard problem, and the requirements of the optimization function are not high, and it has a good global optimization result. However, the traditional genetic algorithm has a slow convergence speed and easy to generate local optimization in solving large-scale set covering problem such as cloud storage content delivery. Therefore, based on the mathematical model abstracted from content delivery in cloud storage, we propose an improved genetic algorithm for the Cloud-CDN (IGA-CCDN), this algorithm proposes a heuristic initial population generation method, using the idea of linear transformation to increase the difference of fitness function results and optimize the maternal selection, it introduces the repair operation and operation of transforming the duplicate individual to increase the diversity of the population and avoid local optimum. The heuristic multi-point crossover selection method is proposed to preserve the high-quality genes as much as possible. A new mutation operator is proposed by determining the mutation probability according to the individual fitness. The algorithm idea is explained in detail next.

1) Individual coding

First, the encoding method of the individual in the population is given. Here, the binary coding method is used to represent an individual as the form of the vector \( S \). For example: \( S=[1,1,0,0,1,1,0,0] \) represents an individual. The individual has 7 genetic loci, and the corresponding solution contains columns 1, 2, 5, and 6. The cloud storage content delivery model represents a delivery scheme of there is a total of 7 potential edged Level-1-SP needs to be delivered, and finally only the first, second, fifth, and sixth edged Level-1-SPs are selected for delivery.

2) Generating an initial population

The initial population is generated by a heuristic algorithm and a random generation algorithm. The advantages of this are: first, using a heuristic algorithm to generate half of the initial population, which can speed up the convergence of the algorithm and reduce the running time of the algorithm; secondly, generating another half of the initial population by means of random generation, which can well guarantee the population diversity.

The size of the initial population is recorded as \( Rs \) and the size of the population is constant. Each individual \( S \) in the initial population is an initial solution, assuming that \( B(i) \) refers to a set of all columns covering the row \( i (i \in M) \), which means all potential edged Level-1-SP that can provide service direct to user \( i \). \( D(j) \) refers to the set of all rows covered by column \( j (j \in N) \), that is, the set of all potential users that the edged Level-1-SP can serve; \( X \) refers to the set of columns in the current solution, which means all the edged Level-1-SP in the current solution that needs to deliver content; \( U \) refers to a set of uncovered rows in the current solution, that is, users in the current solution that are not covered by the edged Level-1-SP; \( W(i) \) refers to the number of columns in \( X \) that can cover row \( i (i \in M) \) in the current solution, that is, the number of edged Level-1-SPs in the current solution that can cover user \( i \). The random generation method can use the existing random function. The specific process of generating an individual by the heuristic algorithm is as follows:

The objective function (5) indicates that the total delivery cost is the smallest; the constraint \( \sum_{t=1}^{n} a_{tj}x_{j} \geq L \) means that each user can overlaid by at least one edged Level-1-SP; in the constraint \( x_{j} \in \{0,1\}, \quad j = 1,2,\ldots,n \), \( x_{j}=1 \) indicates that column \( j \) is included in solution \( X \), and the content will be delivered to the edged Level-1-SP represented by column \( j \), and \( x_{j}=0 \) means that the column \( j \) is not in the solution \( X \), that is, the content will not be delivered to the edged Level-1-SP represented by the column \( j \). If all \( c_{j} (j \in N) \) are the same, this is the set covering problem without weights. If \( c_{j} (j \in N) \) is not the same, this is the set covering problem with weights. The weight \( c_{j} \) corresponds to the content delivery model in cloud storage, which is the cost of delivery content to the edged Level-1-SP represented by column \( j \).
a) Preparation, the current solution $X$ is set to null, $W (i)=0, \forall i \in M$. Let an individual $S$ be a zero vector of 1 row and $n$ columns.

For each row $i$ in $M$, do the following operation: arbitrarily select a column $j$ in $B(i)$ and add it to the current solution $X$, Expressed as $X=X+\{j\}$; then for $\forall i \in D(j)$ Let $W(i)=W(i)+1$.

b) $\forall j \in X$: If $\forall i \in D(j), W(i) \geq 2$, that is, each row $i$ is covered at least two columns, then let $X=X-\{j\}$; then for $\forall i \in D(j)$, let $W(i)=W(i)-1$.

c) Repeat random times for steps 2, 3, and finally for each column $j$ of $X$, set the value of the $j$th bit of the individual vector $S$ as 1.

3) Calculating the fitness of each individual in the group

About the selection of the fitness function, the objective function is generally used to evaluate the fitness of the individual, but the objective function of the algorithm takes the minimum value, so we first define the fitness $P_1$ based on the objective function:

$$P_1 = \begin{cases} p_{\text{max}} - p, & \text{when meets the constraint of (1)} \\ 0, & \text{when does not meet the constraint of (1)} \end{cases}$$  \hspace{1cm} (6)

Where $p = \sum_{p_{ij}} p_{ij}$ is the body part of the objective function, which refers to the current individual cost, representing the delivery cost of this delivery solution, $p_{\text{max}}$ is any value larger than maximum value of $p$. The greater $P_1$, the better fitness of the individual, the fitness function $P_1$ can be used to evaluate the individual's fitness well. In the selection section, the algorithm will adopt the fitness proportional selection method, which is also known as the roulette method. When there are many individuals with low fitness in the group, the deception may occur and the better individuals may not be well selected. In order to amplify the differences between individuals and highlight dominant individuals, the fitness function $P_2$ for the selection section is defined using a linear transformation, as shown in (7), with reference to the idea in [11].

$$P_2 = \alpha \ast P_1 + \beta$$  \hspace{1cm} (7)

Here, $\alpha$ and $\beta$ are adjustment parameters, and $\alpha = 0.5 \ast \frac{p_{\text{avg}}}{p_{\text{max}} - p_{\text{avg}}}$, $\beta = \frac{p_{\text{avg}}}{p_{\text{max}} - 1.5 \ast p_{\text{avg}}} (p_{\text{max}} - p_{\text{avg}})/p_{\text{avg}}$, $p_{\text{max}}, p_{\text{avg}}, p_{\text{avg}}$ and $p_{\text{avg}}$ respectively represent the maximum, minimum, and average values of the fitness $P_1$ of the individual in the current population. After linear transformation, when $p_{\text{avg}} = p_{\text{max}}$, $P_1 = 1.5 \ast p_{\text{avg}}$; when $P_1 = p_{\text{avg}}$, $P_2 = p_{\text{avg}}$; when $P_1 = p_{\text{min}}, P_1 = p_{\text{avg}} - 0.5 \ast p_{\text{avg}} (p_{\text{avg}} - p_{\text{avg}})/p_{\text{avg}}, p_{\text{avg}} < p_{\text{avg}}$. It can be seen that the fitness function after linear transformation increases the difference of fitness between individuals, and is more conducive to the selection of better individuals.

4) Repair operation

Individuals with zero fitness will not produce the next generation, which leads to its partial gene loss, which is not conducive to the preservation of population diversity. Therefore, Ref. [4] is referenced to use the repair operation to turn the infeasible solution $S$ into a feasible solution. The detailed steps of the repair process are given below:

a) First, the coverage of all rows is counted, and for $i$ rows, it is represented as $W (i)=|X \cap B(i)|, \forall i \in M$.

b) Count the set of rows not covered by any column, expressed as $U= \{i|W(i)=0, \forall i \in M\}$.

c) For all the rows in $U$, perform the following three steps in the order of the row number: First, find the column $j$ with the smallest $c_j/|U \cap D (j)|$ in $B(i)$, that means, find the column $j$ in a set of columns which row $i$ can be covered, and the column $j$ can cover the uncovered rows as many as possible, while at the same time cost a relatively small price. The purpose is to find the most cost-effective potential column. Secondly, add the most cost-effective column $j$ to the current solution $X$, that is, let $X=X+\{j\}$, and let $w(i)=W(i)+1(\forall i \in D(j))$, $U=UD(j)$, it means, remove the previously
uncovered rows which now covered by column \( j \); in the end, the gene of the individual \( S \) is adjusted according to the change of \( X \), that is, let \( S(j) = 1 \). So, \( S \) becomes a feasible solution from the infeasible solution.

5) Rebuilding duplicate individuals

In the iterative process of genetic algorithm, we hope to have as diverse individuals as possible, which is beneficial to the optimal solution. The generation of duplicate individuals not only occupies the population resources, but also reduces the diversity of species. If we directly eliminate the duplicate individuals, the population size is reduced, which affects the convergence result of the algorithm. Therefore, the algorithm proposes duplicate individual rebuilding steps, and heuristically transforms the duplicate individual, so that both the population number and the local search for it can be strengthened. The detailed steps of the transformation process are given below:

a) For all the columns in the current solution \( X \) corresponding to the duplicate individual \( S \), if each row \( i \) covered by the column \( j \) has \( W(i) \geq 2 \), that is, rows are all covered by more than one column, then let \( X = X - \{j\} \), and let \( w(i) = W(i) - 1(\forall i \in D(j)) \), and finally let \( S(j) = 0 \), that means, delete redundant columns, without reducing the population size.

b) If the first step does not have the effect of transformation, a certain number of columns are randomly deleted from the current solution \( X \) corresponding to the duplicate individual \( S \). The processing method of each column refers to step a), and we use the tunable parameter \( W_u \) to decide the number of columns to be delete.

6) Select operation

We adopt the fitness proportional selection method, which is the so-called roulette method. The probability that an individual is selected is equal to the ratio of its fitness value to the total fitness value of the group. Here we use the fitness function \( P_2 \) to represent the individual's fitness value, the selection probability is expressed as \( P_i = \frac{P_i}{\sum P_i} \). Obviously, the higher the individual fitness value, the more likely it is to be selected.

7) Cross operation

Based on the existing multi-point crossover method, we further propose a heuristic multi-point crossover method. The method can well preserve high-quality gene fragments, effectively reduce the occurrence of duplicate individuals, and optimize the search for solution space. Let the two parent individuals who need to perform the cross operation be recorded as \( E \) and \( F \), and calculate their fitness points \( p_E \) and \( p_F \) respectively using the fitness function \( P_1 \). The gene length of the individual chromosome is recorded as \( L \), and the number of intersection points is recorded as \( N_u \). The resulting offspring are recorded as \( G \). The detailed steps for the cross operation are given below:

a) Generate a set of distinct random integers \( Y \) values from 2 to \( L/k \) with size of \( V_u \), \( k \) is an adjustment parameter, each number of \( Y \) represents the intersection position on the chromosome, and there are a total of \( V_u \) intersections. Thus, the chromosome of the parent individual is divided into \( V_u+1 \) gene blocks, and the \( i \)-th gene block of the parental individual \( E \) is represented by \( E(i) \), where \( 1 \leq i \leq V_u+1 \). \( E(1) \) represents the gene string fragment between the head and first intersections of the chromosome of the parental individual \( E \), and \( F(V_u+1) \) represents the gene string between the \( N_u \) intersection and the end of the chromosome fragment of the parental individual \( F \).

b) For each random number \( i \) in \( Y \), if \( E(i) = F(i) \), that is, the two gene blocks are the same, then the gene block at the corresponding position of the chromosome of the offspring individual \( G \) may be any parent, represent as \( G(i) = E(i) = F(i) \). If \( E(i) \neq F(i) \), a number \( r \) between 0 and 1 is randomly generated, If \( r \geq Q \), the gene block at the corresponding position of the chromosome of the offspring \( G \) is \( G(i) = F(i) \), and if \( r < Q \), the gene block at the corresponding position of the chromosome of the offspring \( G \) is \( G(i) = E(i) \). Here we define the variable \( Q \) to adjust the probability that the parental individual gene is inherited to the next generation: \( Q = 0.5 + \frac{(p_F - p_E)}{(p_E + p_F)} \), it can be seen that the higher the fitness, the more likely the parental individual pass the gene to the offspring.

8) Mutation operation

A commonly used mutation operator randomly reverses the gene locus on a chromosome according to a fixed mutation probability. In addition, the probability of mutation is generally small in practical
applications, resulting in mutation operations play a minor role in ensuring population diversity and expanding search space. Therefore, we propose an adaptive mutation operator with variable mutation probability according to the individual's fitness in the population. The mutation probability $f_m$ for individual $E$ is defined as shown in (8).

$$f_m = mg \times (1 - p_E)$$ (8)

Where $p_E$ is calculated from the fitness function $P_2$ and $mg$ is the adjustment parameter. It can be seen that the lower the fitness, the greater probability of individual mutation, which is conducive to the diversity of the population and the overall optimization. The specific operation steps are:

a) Calculate the mutation probability for the individual $E$ by (8).

b) Producing a uniformly distributed random number in the interval $[0, 1]$ for each gene locus on the individual chromosome. If the random number corresponding to a gene locus is less than the mutation probability, a reversal occurs, that means, 0 becomes 1, and 1 becomes 0.

9) Optimal preservation strategy

In order to ensure that high-quality genes can be inherited to the offspring individuals as much as possible without being mutated or discarded, we adopt the optimal preservation strategy, that is, the individuals with the lowest fitness in the offspring population are replaced with the individuals with the highest fitness in the parent population. It avoids the situation that the crossover and mutation operations inadvertently damage the high-quality genes. It effectively improves the convergence speed of the algorithm, and optimizes the structure of the solution.

The specific algorithm is shown in Algorithm 1.

Algorithm 1: Improved genetic algorithm for the Cloud-CDN

Input: content delivery instance
Output: an optimal delivery solution

Initialization all the parameters;
Generate the initial population through Random and Heuristic algorithm;
While (termination condition is not satisfied) {
    Calculate the individual fitness $P_1$ and $P_2$ using equation (6) and (7);
    Repair the infeasible solution;
    Rebuilding duplicate individuals heuristically;
    Select parent individuals through Roulette algorithm;
    Cross gene by the heuristic multi point crossover algorithm;
    Mutation offspring individuals using adaptive variation;
    Preserve the most fitness Parent individual;
}
Output the most fitness individual with the maximum value of $P_1$ as the optimal solution.

5. Simulation experiments and analysis

The algorithm is simulated on the CloudSim simulator, the system environment is Intel(R) Core (TM) i5, 2.80GHZ, and the memory is 8GB. Initialization of parameters involved in the algorithm is: the initial size $Rs$ of the population is set to 100; the tunable parameter $W_6$ indicating the number of deleted columns in the duplicate individual is set to 10; The cross-rate $Q_c$ is set to 0.99, which indicates the probability that the parental individual gene is inherited to the next generation in the crossover operation; the parameter $k$ is generally set to 2, and is set to 1 in Scpcyc06, 07; the value of the parameter $V_6$ indicating the number of intersections is differently set in different instances, in Scpb1-4 set to 300, in Scp41-45 set to100, and in Scpcyc06, 07 set to 40; in the mutation operation, the parameter $mg$ to adjust the mutation probability is set to 0.02.
Table 1. Algorithm experiment results comparison

| Instance | Size      | User-Greedy | GA-heuristic | IGA-CCDN |
|----------|-----------|-------------|--------------|----------|
| Scp39    | 1000×200  | 838.9       | 584.5        | 567.5    |
| Scp40    | 1000×200  | 854.2       | 575.8        | 566.0    |
| Scp41    | 1000×200  | 854.2       | 575.6        | 575.6    |
| Scp42    | 1000×200  | 854.2       | 566.8        | 566.8    |
| Scp43    | 1000×200  | 854.2       | 566.0        | 566.0    |
| Scp45    | 3000×300  | 963.9       | 624.5        | 594.6    |
| Scpb1    | 3000×300  | 424.2       | 296.0        | 280.8    |
| Scpb2    | 3000×300  | 428.1       | 298.8        | 298.8    |
| Scpb3    | 3000×300  | 454.2       | 324.0        | 324.0    |
| Scpb4    | 3000×300  | 443.9       | 354.5        | 341.6    |

Figure 5. Delivery cost comparison

First, we use several instances on OR-Library to simulate content delivery in cross-domain cloud storage. The columns in the instance correspond to edged Level-1-SPs in the content delivery network topology in cross-domain cloud storage, and the rows correspond to users. The weights correspond to the cost of the delivery, and the examples have given the optimal values for reference. The instances used are all SCPs with weights. When the number of iterations exceeds 100, it is judged whether the structure of the solution changes in the nearly 50 consecutive iterations, and if there is no change, the algorithm terminates. Finally, the experimental results are compared with the c and the GA-heuristic algorithm proposed by Beasley. The results are shown in Table I.

Next, we apply the Improved genetic algorithm for the Cloud-CDN (IGA-CCDN) and the genetic algorithm GA-WSCP proposed by Chen Liang to solve the four large instances of cross-regional cloud storage content delivery problem in reference [6] separately, and simulate it on the Cloudsim simulator. The results are shown in Fig.5. It can be seen that the IGA-CCDN algorithm proposed in this paper has a great advantage over the GA-WSCP algorithm in reducing the delivery cost. The reason is that for the four large instances, the IGA-CCDN algorithm can solve the optimal solution well, while the GA-WSCP algorithm can’t get the optimal solution when dealing with large instances, it only produces the approximate optimal solution. Therefore, the cost generated by the IGA-CCDN algorithm is far lower than the GA-WSCP algorithm. It can be seen that the IGA-CCDN algorithm is more suitable for solving the problem of content delivery in cross-domain cloud storage.
6. Summary
In this paper, we first propose the P2P Tree resource management model to provide a network topology for content delivery in cloud storage. At the same time, the cost model is an important indicator to determine the content delivery technology in cloud storage, we refer to the CDN distribution process, combined with the cloud storage charging mechanism, to give the content delivery cost model in cloud storage. Based on the above network topology and cost model, we propose an improved content delivery technology based on genetic algorithm in cloud storage. Finally, using CloudSim to carry out simulation, experiments show that the content delivery algorithm based on genetic algorithm can better solve the problem of content delivery in cloud storage.

References
[1] Zhou Ke, Wang Hua, and Li Chunhua, Cloud Storage Technology and Its Application, ZTE Corporation, 2010, 16(4): 24-27.
[2] Han Xingye, Li Xinning, and Liu Yinpeng, Research on Resource Management for Cloud Computing Based Information System, 2010 International Conference on Computational and Information Sciences, 2010, 491-495.
[3] C.E. Lemke, H.M. Salkin, and K. Spielberg. Set covering by single-branch enumeration with linear-programming subproblems, Operations Research, 1971, 19(4): 998-1022.
[4] J.E. Beasley, and P.C. Chu. A genetic algorithm for the set covering problem, European Journal of Operational Research, 1996, 94(2): 392-404.
[5] Zhang Yufen, Qi Hongran, and Liu Shipe, Model and Algorithm for Location Problem of a Class of Emergency Service Facilities, Mathematics in Practice and Theory, 2009 (014): 37-41.
[6] Chen Liang, and Ren Shijun, Application of a genetic algorithm in set coverage problem, Journal of Harbin University of Commerce (Natural Science Edition), 2006, 22(2): 67-70.
[7] Liu Shuan, and Tang Fei, Research on system identification method based on genetic algorithm, Systems Engineering Theory and Practice, 2007, 27(3): 134-139.
[8] Wang Jiyi, and Wu Yanxian. Implementation of adaptive multi-bit mutation genetic algorithm, Computer Science, 2003, 30 (8): 141-143.
[9] Hu Dawei, and Chen Cheng, Application of genetic algorithm (GA) and tabu search algorithm (Ts) in distribution center location and route problem, Systems Engineering Theory and Practice, 2007, 27(9): 171-176.
[10] Li Yanfeng, Xia Guoping, Yang Yuexiang, and Gong Zheng, Multinational supply chain tactical planning model based on fuzzy random expectation planning, Systems Engineering Theory and Practice, 2005, 25(8): 1-9.
[11] Wang Xiaoping, and Cao Liming, Genetic Algorithm Theory, Application and Software Implementation, Xi'an Jiaotong University Press, 2002