

## Resource configuration algorithm based on popularity for P2P streaming media in cloud computing

Yang Ge<sup>1,2, a</sup>, Gao Bing<sup>1</sup>, Feng Yiju<sup>1</sup> and Yu Ying<sup>1</sup>

<sup>1</sup> Institute of Information Technology, Beijing Normal University (Zhuhai Campus)

Zhuhai, China, 519087

<sup>2</sup> Engineering Lab on Intelligent Perception for Internet of Things (ELIP), Shenzhen Graduate School, Peking University, China

<sup>a</sup>yangge@bnuz.edu.cn,

**Keywords:** P2P network; deficit bandwidth; popularity; resource configuration algorithm; replacement algorithm.

**Abstract**—In this paper, we proposed the new popularity which included “deficit bandwidth”; secondly, based on the new popularity the streaming media files which need to be replicated were determined. For the files, their replica number were equal to product of its deficit bandwidth divide the sum of the deficit bandwidth of all these files and multiple the available cache space of all active peers that were not watching one of these files. Replacement algorithm would be carried out if there was no enough space to cache the new file. Experimental results show that the workload of the proposed algorithm, compared with proportional replication algorithm, is earlier and smaller in steady conditions.

### I. Introduction

In P2P streaming media quality of service is often hard to get effective guarantee. In P2P for different nodes their calculations, storage capacity, network bandwidth and so on are different<sup>[1]</sup>.

[2] Studied a distributed replica approximate placement algorithm within a given distributed replication group. [3] studied the optimal requirements of replication algorithm: content placed in different nodes, meet the storage and constraints, expect to have a minimum of service and load balancing of bandwidth. [4] pointed out that P2P VOD system, there are two factors affecting replication strategies, upload speeds and popularity of streaming media files. In this paper deficit bandwidth is used to calculate popularity. Reasons are: The existing streaming media placement algorithms mainly consider nodes' storage performance and the access cost. The validity of the placed node is not considered in access cost.

### II. Streaming resource configuration algorithm

Streaming resource configuration algorithms include streaming media replication establishment algorithm and streaming media replication update algorithm.

#### 2.1 streaming media replication establishment algorithm

Requesting node is requested for streaming media files to the current nodes or replication nodes. The service node is the node storage origins of media files, it can only upload data to other nodes, cannot download data from other nodes there. This article assumes that all streaming media files from start to finish watching [4]. The requesting node to obtain the required streaming media files from the current node, if the current node does not meet the requirements.

(1) Streaming media file k's expected deficit bandwidth

Symbol:

$M_s$  : a collection stored in the service nodes on streaming media files.

$A_k$ : watching streaming k current set of nodes,  $A_k''$ : Request streaming k collection of nodes  $A_k'$ :

to provide services  $A_k^n$  replication nodes for collection.

$d_i^c$  : download speeds of current node  $c$  that serve requesting node  $i$   $c \in A_k$

$d_i^r$  : download speed of replication nodes  $r$  that serve requesting node  $i$ . Replication nodes  $r$  uses part of or all cache to cache streaming media file,  $r \in A_k$

$R$  : means the streaming media file playback rate.

$n_k$  : Indicates the number of nodes to watch streaming media file  $k$ .

$s$  : Indicates the service node.

In P2P networks, due to the downloading speed of the request node  $i$  determining on current nodes, replication nodes and service nodes. If  $d_i^r + d_i^c$  less than the playback speed of streaming media file. It is needed to add  $R - d_i^r - d_i^c$  to the upload speed of service node to meet the quality of service. Using  $d_i^s = R - d_i^r - d_i^c$  ( $\forall i \in A_k$ ) to indicate the upload speed from the service node  $s$  to request node  $i$ . The speed and bandwidth which are mentioned in this article represents the same meaning.

Using  $U$  to indicate the expectation function of the service node  $S$ 's consumption of uploading bandwidth, the mathematical expression of it is formula(1)

$$U = E\left(\sum_{k \in M_s} \sum_{i \in A_k^n} d_i^s\right) \quad (1)$$

In large scale P2P system, if given the number of the current node, the expectation of the service node upload bandwidth  $U$  [4] is the formula (2):

$$U \approx \sum_{k \in M_s} [E\left(\sum_{i \in A_k^n} (R - d_i^c) - R_k\right)^+] \quad (2)$$

$R_k$  is from all the replication nodes, but can contribute to the summation of streaming media file  $k$ 's maximum upload bandwidth. While for the streaming media file  $k$ , defined as the bandwidth deficit formula (3)

$$\tilde{D}_k(n_k) = \sum_{i \in A_k^n} (R - d_i^c) = \sum_{i \in A_k^n} (d_i^r + d_i^s) \quad (3)$$

According to this definition, the current node's bandwidth is the difference between the speed of request playback and the total speed provided by the current node.

Further, streaming media file  $k$ 's expectation is defined as formula (4)

$$D_k(n_k) = E(\tilde{D}_k(n_k)) \quad (4)$$

Put the formula (3), (4) into formula (2), we can get

$$U = \sum_{k \in M_s} [D_k(n_k) - R_k]^+ \quad (5)$$

Formula(5) is the relationship between the consumption of current node bandwidth and the service nodes bandwidth. Based on this relationship, the purpose of this paper is to find a replication algorithm to determine  $R_k$  and minimize  $U$ .

When node scheduling policies are succession and greedy, the deficit bandwidth [9] as shown in formula (6) as follows:

$$D_k(n_k) = n_k R - \sum_{i \in A_k^n} \tilde{d}_i^c \quad (6)$$

Calculate  $\tilde{d}_i^c$  by iterative formula (7) [4]

$$\tilde{d}_i^c = \begin{cases} 0, & i = 1, \\ \min\left(\sum_{j=1}^{i-1} (\tilde{u}_j - \tilde{d}_j^c), R\right), & i \geq 2. \end{cases} \quad (7)$$

$\tilde{u}_j$  indicates the maximum upload speed of node  $j$ , if there is not a replica, which means  $d_i^r$  equal to 0, for streaming media files  $k$ ,  $D_k(n_k)$  is the consumption of service node  $s$ 's upload bandwidth. After calculate deficit bandwidth by iteration, it is needed to find its expectation  $E(D_k(n_k))$ . The

calculation of  $E(D_i(n_i))$  is: if the upload bandwidth distribution of nodes is known, we can find its expectation's deficit bandwidth.

Meanwhile,  $\frac{E(D_i(n_i))}{\sum_k E(D_k(n_k))}$  is both the ratio of streaming media file i's deficit bandwidth and all the streaming media files' bandwidth deficit, and also the best replication ratio of stream media files i.

(2) The expectation of streaming media file i's memory space is

$\sum \{ E(D_i(n_i)) \times l(s) \}$   
 $E(D_i(n_i))$  : said streaming media file I expect the deficit of bandwidth.  $l(s)$ : streaming media file's broadcast time length.

(3) Streaming media file popularity definition

In this paper, definition of streaming media file popularity for formula (8) :

$$popularity = \left[ \frac{T_{sum}}{T_{last} - T_{first}} \times \min \left\{ 1, \frac{(T_{last} - T_{first}) / N}{T_c - T_{last}} \right\} \times \frac{E(D_i(n_i))}{\sum_k E(D_k(n_k))} \right] / H_k \quad (8)$$

Among them:

$T_{first}$  : Media file is accessed for the first time;

$T_{last}$  : Media files last accessed time, if the media file is accessed for the first time, than  $T_{first} = T_{last}$  ;

$T_{sum}$  : the media file is accessed, the total time length;

$N$  : the number of media files to be accessed;

$T_c$  : the current time.

$H_k$ : the cache files in the current node k the size of the amount of data;

$\frac{E(D_i(n_i))}{\sum_k E(D_k(n_k))}$  is popular the expectations of the streaming media file i deficit bandwidth and all streaming media file expected deficit ratio of the sum of the bandwidth.

(4)Performance is higher than service node selection criteria

According to the proportion of different, can make the algorithm has good robustness. Define node comprehensive performance index for formula (9) :

$$W = \partial \times \frac{u_j}{\sum_M u_j} - \beta \times \frac{d_j}{\sum_M d_j} + \eta \times \frac{S_j}{\sum_M S_j} + \xi \times \frac{1}{\sum_M J_{jq}} \quad (9)$$

$W$ : every node in the system of comprehensive performance.

$u_j$  : the maximum upload speed of node j.

$S_j$  : the storage capacity of node j.

$d_j$  : the biggest download rate of node j.

$J_{jq}$  : indicates the current node in addition to the current node in the P2P network hop count between the service node.

$M$ : is the total number of streaming media files stored on the service node.

$\partial, \beta, \eta, \xi$  : respectively, storage capacity, maximum upload speed, maximum download rate, the current node and in addition to the current node in the system of the value of the hop count between the service node according to actual situation to adjust its network. Its constraints is:  $\partial + \beta + \eta + \xi = 1, 0 \leq \partial \leq 1, 0 \leq \beta \leq 1, 0 \leq \eta \leq 1, 0 \leq \xi \leq 1$ .

(5) Streaming media replication algorithm is described

1 to calculate the popularity of streaming media files.

2 Select high popularity before B % of streaming media files as popular, B positive integers

within 0 to 100.

3 popular each flow media files, from the trace log nodes to find "are watching other streaming media file, at the same time local cache does not cache the streaming media file" active node.

4 on the popular streaming media file, calculation of each stream media file expect deficit bandwidth and storage space.

5 to calculate active node general can make use of storage space.

6 judgment of the active node general can use storage space is greater than the sum of these popular file the sum of total expected storage space? If it is, in step 7 to 11. Otherwise, go to the step 12-19.

7 the popular each flow media file, step 8 to 11.

8 according to the expectations of the streaming media file storage space and computing the expected number of copies of streaming media file, with the expected amount of replica number as the actual number of copies.

9 Find the current active node that didn't watch the stream media file.

10 for each active node, go to step 11.

11 if the streaming media files need to be stored number greater than zero, active node and available storage space at the same time, the judge whether the active node can hold a the streaming media files. Accordingly, update the rest of streaming media files need to store the number of copies, and the active node available storage space. Otherwise, you skip the node.

12 the popular streaming media files, but also to step 13 to 19.

13 for each flow media file, use the active node of the general can make use of storage space, the sum of times each file deficit expected bandwidth and the expectations of all files deficit ratio of the sum of the bandwidth, as each product results from these files need to be placed a replica of the storage.

14 according to determine the total replica storage need to store the number of copies of streaming media files.

15 find the current are watching the stream media file all the current node.

16 randomly selected from a current node.

17 computing P2P network comprehensive performance of non-service node.

18 Sort comprehensive performance and select the higher n nodes. Here n is the number of the replica that is stored.

## 2.2 Streaming media replication update algorithm

When a node request for new streaming media, it should check this new streaming media file whether it cache in the local cache. If it is cached, then it will not be replaced. The satisfaction index is  $SI_i$  which means the ratio of amount of current replication and excepted replication. The replacement algorithm replaces the max  $SI_i$  of streaming media in the system. This algorithm main idea is to keep the ratio of replication and deficit bandwidth balance. Because the ratio of deficit bandwidth is the Best Replication Ratio<sup>[4]</sup>. If the  $SI_i > 1$ , that means the current replication is more than the excepted replication at this moment. If the  $SI_i < 1$ , that means the current replication is less than the excepted replication. If the  $SI_i = 1$ , that means that the current replication equal the excepted replication. Therefore, using this algorithm in every cycle, the local cache will replace the streaming media of max  $SI_i$ . Calculating the excepted replication in current node  $N_i^{\text{exp}}$  is the (10).

The following log node will mark every replication  $N_i^{\text{cur}}$  in current node. And then the system will calculate every streaming media's satisfaction index  $SI_i$

$N_i^{\text{exp}}$  the excepted amount of replication in P2P network in current node, exp is the index of excepted number

$N_i^{\text{cur}}$  the current amount of replication in P2P network in current node, cur is the index of current number

$SI_i$  the satisfaction index of every streaming media file

$N$  the number of node in the system

The description of the algorithm of updating streaming media

If the node has not capacity to store the streaming media fully, then go to step (2)

In this paper using MatlabR2012 as the platform of simulation to find out the effectiveness of this algorithm.

In this experiment, the parameter as following:

1. the relationship of number of router and router; every router will include service node and non-service node<sup>[4]</sup>. Assuming the router include non-service node and service node as  $peerset=[10,1;20,2;10,2;15,3;10,1]$ , among each row represent the number of router cover the non-service node and service node. Service node is 9 and non-service node is 65.

2. The number and size of streaming media in P2P network<sup>[4]</sup>. Assuming that the system has 20 streaming media, each file is 90min the playback ratio is  $R=500\text{kbps}$ .

3. The node request rate<sup>[4]</sup>; In order to play the streaming media fluency, the node request rate is equal to streaming media playback rate, therefore  $r'=500\text{kbps}$

4. Each node (including service node and non-service node) the number of streaming media cache files and available storage space. Initial moment of non-service node does not cache any streaming media files, and the non-serving Node available storage space for 600M-700M, its size [600M, 700M] obey the uniform distribution<sup>[4]</sup>.

5. In this experiment one set of data are shown in Table 1 (Non-service node maximum upload speed distribution, NSNMUSD) and Table 2(service node maximum upload speed distribution, SNMUSD)<sup>[4]</sup>. In two algorithms the higher 30% of the streaming media files are only copied, namely  $B = 30$ .

Table 1 NSNMUSD

NSNMUSD (kbps)	200	100	50	12
Probability value	0.5	0.3	0.05	0.15

Table 2 SNMUSD

SNMUSD (kbps)	768	384	256	128
Probability value	0.5	0.3	0.05	0.15

6. The actual length of time for each iteration and the iteration number corresponding [4]. Assuming iteration corresponding is 3 hours. The total number of steps is 15 iterations.

7. Popularity of a streaming media file and expectations deficit bandwidth. Initial time, the popularity of a streaming media file is the same for each, so each time the initial popularity of streaming media files assigned zero. In addition, due to the expected deficit bandwidth streaming media file with the number of users accessing it related to the initial moment yet requesting node watch streaming files, so expect the deficit will be assigned an initial bandwidth streaming media files each time zero.

8. Select comprehensive performance formula weight values. Overall performance equation, initial weight values are equal to 0.1, 0.2, 0.5 and 0.2. The subsequent resource configuration algorithm according to the weight values is still not completed. We update the weight values. Update law is: weight value  $\alpha$  of maximum upload speed of a node every time decreases 0.05. Weight value  $\beta$  of maximum download rate of a node every time decreases 0.05. Weight value  $\gamma$  of storage space of a node every time increases 0.2. Weight value  $\xi$  of hop numbers between the current node and the non-serving node every time decrease 0.1. If updating five times there is still not enough space to store, then out of the loop.

### III. PERFORMANCE EVALUATION

In the experiment the performance of our proposed resource configuration algorithm and the proportional replication algorithm<sup>[5]</sup> are evaluated by the service peers workload and satisfied peers rate.

The experimental simulation results are in Fig.1 and Fig.2. The set of experiments results ( $B=30$ , non-service peers and service peers upload rate obeys the Tab.1 and Tab.2 distribution) show that

service peer workload of the proposed streaming media resource configuration algorithm is significantly lower than the proportional replication algorithm when the service peer workload is reduced to 0.5Mbps in the fifth iteration. Experimental results show that the workload of the proposed algorithm, compared with proportional replication algorithm, is earlier and smaller in steady conditions.

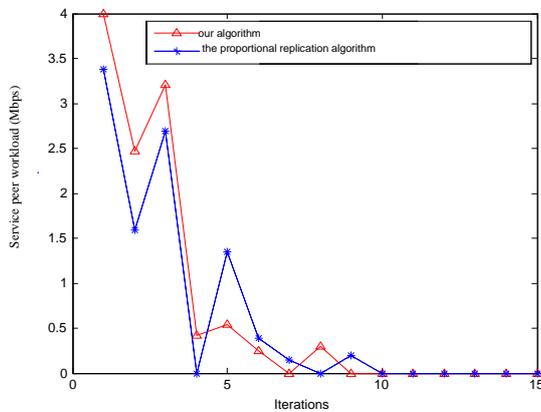


Fig.1 the service peer workload

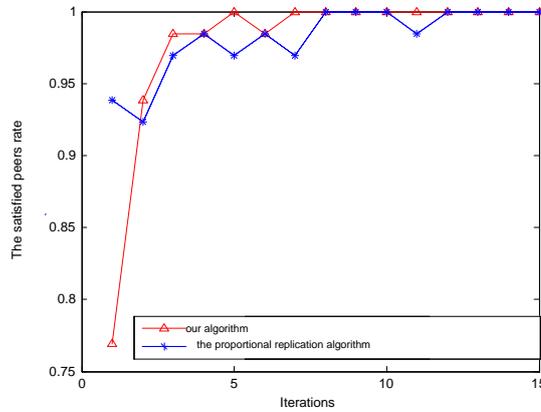


Fig.2 the satisfied peers rate

#### IV. Conclusion

In this paper the resource configuration algorithm based on the popularity of P2P streaming media in cloud computing is mainly presented. How to effectively motivate peer upload streaming media is the next problem to consider.

#### Acknowledgements

This work is financially supported by Guangdong Provincial Special Funds Project for Discipline Construction (No.2013WYXM0122); Guangdong Provincial College Innovation and Entrepreneurship Project (201413177035). The Special Development Funds Project for Strategic Emerging Industry of Shenzhen (No. JCYJ20120614144655154); Research and Innovation Team of Beijing Normal University (Zhuhai Campus) (No.201251006).

#### References

- [1] Rohmer T, Nakib A, Naffa A. Priori Knowledge Guided Approach for Optimal Peer Selection in P2P VoD Systems[J]. IEEE Transactions on Network and Service Management, 2014, 11(3): 350-362
- [2] ZAMAN S, GROSU D. A distributed algorithm for the replica placement problem[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(9): 1455-1468.
- [3] ZHOU Y, FU T Z J, CHIU D M. A unifying model and analysis of P2P VoD replication and scheduling[C]. IEEE INFOCOM 2012, March 25-30, 2012, Orlando, USA.
- [4] WU W, LUI J C S. Exploring the optimal replication strategy in P2P-VoD systems: Characterization and evaluation[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(8): 1492-1503.
- [5] TEWARI S, KLEINROCK L. Proportional replication in Peer-to-Peer networks[C]. IEEE INFOCOM 2006, April 23-29, 2006, Barcelona, Spain.