2009 International Conference on Web Information Systems and Mining

Common Caching Replacement Algorithm for Video-On-Demand System

Fengbin Li, Jun Li, Zhong Hu, Jun Zhou
Key Laboratory of Network Communication System and Control,CAS
University of Science and Technology of China (USTC)
Hefei, Anhui, 230027, P.R.China
lfb0132@mail.ustc.edu.cn, Ljun@ustc.edu.cn, huzhong@mail.ustc.edu.cn , zuju@mail.ustc.edu.cn

***Abstract*—As the streaming media files growing larger and larger in size, it inevitably aggravates the network congestion and user perceive latency, to settle problem lots of caching algorithms have been applied in video-on-demand (VOD) system. However, different algorithm is correspondence to an unique caching replacement policy, which limits its applications and the effects are not very satisfied, so in VOD aspect it is necessary to find a common replacement algorithm. In this paper, we propose a Common Caching Replacement Algorithm (CCRA). As a unification replacement algorithm, it sufficiently considers the recent visit and segment size, effectively solves the disk I/O bandwidth bottleneck limitation and improves byte hit rate. We respectively replace the previous caching replacement policy in Uniform segmentation algorithm and Exponential segmentation algorithm with CCRA. From the experimental comparison, Uniform segmentation with CCRA improves almost 5% in Disk Reduce Ratio, and Exponential segmentation with CCRA increases near 8% in Byte Hit Ratio.**

***Keywords*-caching replacement; video-on-demand; segment-based cache;**

I. INTRODUCTION

In the recent years, the demand of video services over the network have been increasing fast due to high development of network and digital video processing technologies. The more and more large streaming media files have caused significant impacts on user perceive latency and network congestion, a popular approach to reduce the response time and backbone bandwidth consumption is to deploy proxy caches at the edge of the Internet. Although proxy caching has been successful in delivering static text-based content, it encounters difficulties in delivering streaming media content for two factors. One is that a media object is larger than a text-based object, thus, caching entire media objects cannot as feasible as static objects, which can quickly exhaust entire cache. The other is that the popularity of different parts of a video can be different, like the prefix is generally more popular.

Due to the large size of streaming media, many caching algorithms for videos have been employed. Prefix caching[1] and segment-based caching[2][3][4] are two frequently used algorithms. Segment-based caching is the recent research hot, especially the exponential segmentation algorithm[2] and the adaptive and lazy segmentation algorithm[3][4], which are the two typical segment caching algorithm. Prefix caching is mostly targeted to reduce the start delay and jitter of the streaming media, especially benefit to a small cache. Exponential segmentation strategy can quickly replace a segment in order to adapt the change of the movie's popularity. Adaptive and lazy segmentation can determine the segment length based on the client access behaviors in real time, admission and eviction of segments are carried out adaptively based on an accurate utility function. All these research caching algorithms almost aim at reduce the backbone bandwidth, the most difference is the segmentation strategy and the performance influence factors.

At the same time, the streaming media applications facing a change in the principal contradiction. The network transmission bandwidth has been greatly improved, while disk I / O bandwidth only has a few dozen megabytes compared to Gigabit Ethernet, So disk I / O bottleneck is the key issues to be urgently settled in VOD system . In addition, how to effectively support the VCR(Jump, pause, fast-forward, rewind, etc.) operation has become an inevitably caching research questions in streaming media. Under this situation, lots of Caching algorithms are been applied in vod system, but different algorithms with different replacement implementation, it brings lots of troubles to programmers to determine which area should use that replacement. So it is urgent for us to propose a new method to resolve this.

This paper proposes a Common Caching Replacement Algorithm called CCRA, first using CCRA to replace LRU in Uniform segmentation algorithm and the utility function of Exponential segmentation algorithm, respectively form the new algorithm Uniform segmentation(CCRA) and Exponential segmentation(CCRA), then compare with the previous formation. CCRA neither like LRU just considers the last time of using, nor like the replacement policy of exponential algorithm considers the different size of segment, it comprehensively takes into account of the recent visit and the segment size. From the experimental comparison, we can see that Uniform segmentation(CCRA) almost improves 5% in Disk Reduce Ratio and Exponential segmentation(CCRA) increases near 8%.

The rest of the paper is organized as follows. Related Works is presented in Section II. Algorithm Design is presented in Section III and Section IV presents Simulation And Evaluations. We make the Conclusion in Section V.

978-0-7695-3817-4/09 $26.00 © 2009 IEEE
DOI 10.1109/WISM.2009.154

II. Related works

Some earlier studies on caching focused more on web-caching[6,7], Cao and Irani [6] presents a relative comparison of various cache replacement policies that have been proposed for web-caching. Their work discusses some of the merits and concerns of replacement policies such as Least Frequently Used (LFU), Least Recently Used (LRU), etc. However LFU exits the problem of caching pollution, means that it has no ways to distinguish recent versus past reference frequency of a page and LRU exits long-loop model problems, that is, it caches the segment which just had been replaced out of cache. Simultaneously both of these algorithms prone to continuing to replace the same media object, this acting definitely increase the probability of release, also decrease request hit rate and increase response latency.

Recent research in the area of caching streaming media also exists a lot[8,9,10]. In [8], the authors put forward a method through caching the file's 'hotspots' clips to support the user's VCR operation. In [9], they through the boundary detection, key-frame selection and face detection and tracking technique to obtain the 'video summary(a number of key-frame images)' document to support the user's VCR operation. Both of these methods to provide users with a video preview function and support the user's fast-forward and back jump, but both of these are required extra files to record the 'hot' and 'video summary', increasing the complexity of caching management. When it comes to the access of 'video summary', it is also required analysis of coding techniques and graphic technology, which limits the application of these two algorithms. Dynamic Interleaved Segment Caching [10] also supports the VCR operation, in which the algorithm according to client access patterns, dynamically cache the segments of a media object, which greatly reduces the start-up delay of the user-demand. However, the algorithm at the expense of caching performance to support the user's VCR operations, also in byte hit rate is inferior to the consecutive caching algorithm.

III. Algorithm design

The essence of the caching problem is at the constraint of limited resources to decide which data should be cached and how to manage the cache data. In resource utilization aspect, how to make unit caching space gain the largest caching value is one of main purpose of the study. In order to reach the maximum using value, the unit caching space should cache the most valuable data which is the most frequently used. The definition of data's utility function is expressed as:

$$\phi_i(t) = \frac{P_i(t) \times C_i(t)}{size_i} \tag{1}$$

In this formula, for each data i, $size_i$ is its size, $C_i(t)$ is the cost of accessing data i, $P_i(t)$ is the probability of future use. Assuming the maximum caching capacity is B, the collection of streaming media in cache is Data(t)={1,2, $\cdots$,n}, the way of caching recorded as $\delta \in \{1, 0\}$, 1 express cache, 0 indicate no cache. In order to maximize space utilization and byte hit rate, t time to implement the caching strategy should be made:

$$Max \sum_{i \in Data(t)} P_i(t) \times C_i(t) \times \delta_i(t)$$

$$\sum_{i \in Data(t)} size_i \times \delta_i(t) \approx B$$

Up to now, the parameter $size_i$ can be easily calculated, the key step is to estimate the $P_i(t)$ parameter and the $C_i(t)$. As $P_i(t)$, because of the uses' random visit, it is effectively to assume that parameters are independent and obey the Poisson distribution with parameter $\lambda_i(t)$. So the probability can be replaced by the approximation[5]:

$$P_i(t) = \frac{\lambda_i(t)}{\sum_{j=1}^{n} \lambda_j(t)} \tag{2}$$

Since the replacement is determined by the ranking of the files' utility function value, we brought (2) into (1) and can be simplified as:

$$\phi_i(t) = \lambda_i(t) \times \frac{C_i(t)}{size_i} \tag{3}$$

To estimate the value's parameter $\lambda_i(t)$, according literature[5] method, we define $k_i(t)$ as: the number of the most recent references retained between[t_{ki}, t], in which t is the current time, t_{ki} is the time of the $k_i(t)$ backward reference, in other words, t_{ki} is the first time appear of file i, so

$$\lambda_i(t) = \frac{k_i(t)}{t - t_{ki}} \tag{4}$$

Bring (4) to (3), we can get the utility function:

$$\phi_i(t) = \frac{k_i(t)}{t - t_{ki}} \times \frac{C_i(t)}{size_i} \tag{5}$$

In order to obtain the $C_i(t)$, we should select the smallest cost in cache, which equals to the smallest cost which obtained from the servers. Assume the servers which we get the data are { ser_1,ser_2,...ser_m }, the cost of accessing data i from ser_k ($1 \le k \le m$) is $NC(k)$, and all the link which the data pass as L, each link l the cost is $lk(l)$, $NC(k)$ can be calculated as:

$$NC(k) = \sum_{l \in L} lk(l)$$

so $C_i(t)$ can be expressed as:

$$C_i(t) = \min_{1 \le k \le m} NC(k) \tag{6}$$

Bring (6) to (5), the utility function can be written as:

$$\phi_i(t) = \frac{k_i(t)}{t - t_{ki}} \times \frac{\min\limits_{1 \le k \le m} NC(k)}{size_i}$$

In the implementation of the algorithm, first of all it should calculate each file media's utility function value, secondly replace the smallest value of the media file. In the utility function, it fully considers the media files' recent visit, the segment size and the network cost. It also benefits to decreasing the start-up latency and increasing the caching space utilization.

IV. Simulation and Evaluations

Generally speaking, the Byte Hit Rate (BHR) and Disk Reduce Ratio (DRR) are the two main important indicators to measure the performance of traditional caching system. In all video-on-demand operation, in order to know how much disk bandwidth we need, we can through test how many numbers of on-demand which are not delayed, in other words, what we demand have inside the cache and the user can see directly without waiting.

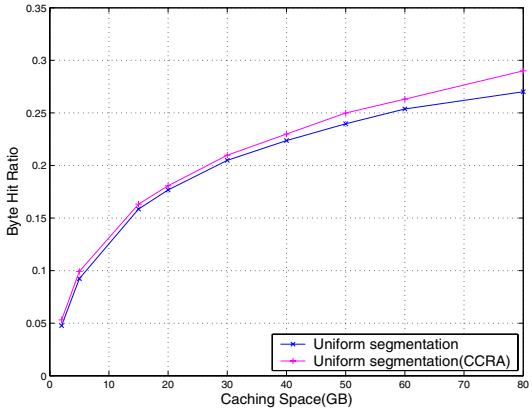
The Byte Hit Rate can be defined as the size of data which we order has been in cache divided the whole data size. In order to give the definition of DRR, we assume that each jump as one demand and the total number times of on-demand operation is R. Define T_i as the mask in section on the i-th times demand, if i-th demand has been cached, then T_i=1, else T_i=0. So DRR is that all on-demand number $\sum_{i=1}^{R} T_i$ divide the total number R, it can be written as:

$$DRR = \frac{\sum_{i=1}^{R} T_i}{R}$$

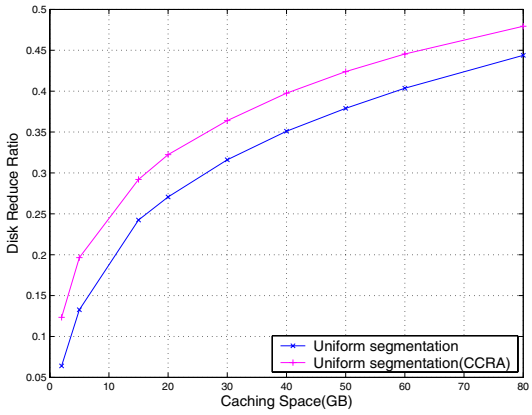
We choose the uniform segmentation algorithm and the exponential segmentation algorithm as comparison. Uniform segmentation algorithm uses uniform length as its segmentation strategy, incremental caching as its access control strategy, that is, when the on-demand data is not in cache, it will only cache the next piece of data, and uses a simple LRU as replacement strategy. And we will use CCRA to replace LRU in uniform segmentation algorithm, so it will form a new caching algorithm, we will use Uniform segmentation(CCRA) in the following graph.

Exponential segmentation algorithm uses multiplication segmentation strategy, with the same access control strategy as uniform segmentation algorithm and uses formula (7) as the replacement criteria, in which T_c is the current time, T_i is the last time to access the data, i is the number of the paragraph. Still we use CCRA to replace the formula (7), the new form we write it Exponential segmentation(CCRA).

Because of adaptive and lazy algorithm would cache the full text on the first demand, making the algorithm in a poor performance in a small capacity cache, even making out the cost of the replacement greater than the benefit of caching, adaptive and lazy algorithm probably will increase the disk load and does not apply to the local cache, thus we do not have this algorithm as a comparison. Exponential



(a)



(b)

Figure 1. Uniform segmentation comparison

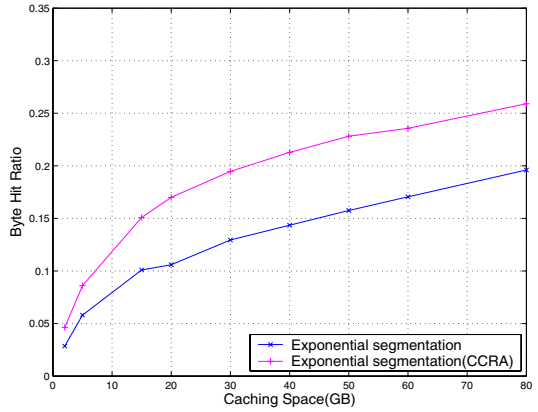
segmentation algorithm's utility function can be expressed as:

$$\frac{1}{(T_c - T_i) \times i} \tag{7}$$

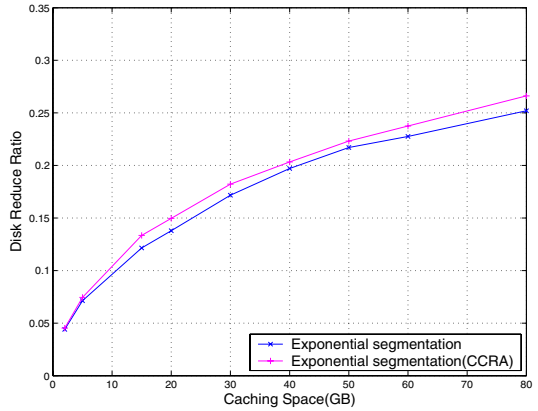
Our simulation will use 10 days' USTC VOD log, it detailed recorded all the users' various operating. In our segmentation, we set the segmentation size of uniform segmentation algorithm is 2M, the base segment of exponential segmentation algorithm is 2M, the first incremental caching size is 4M.

Uniform segmentation compares with Uniform segmentation(CCRA) in byte hit ratio in Fig.1 graph (a), and in disk reduce ratio in Fig.1 (b). From Fig. 1 we know that the new form just improves a little compared with uniform segmentation algorithm in Byte Hit Ratio, but it really improves in DRR, almost improves by 5%, so the new one can effectively alleviate the disk bandwidth requirements. The reason is that it comprehensively considers the media files' recent visit, so it can guarantee that the caching contents are all the popular media.

Exponential segmentation compares with Exponential



(a)



(b)

Figure 2. Exponential segmentation comparison

segmentation(CCRA) in byte hit ratio in Fig.2 graph (a), and in disk reduce ratio in Fig.2 (b). From Fig. 2 we see that the new form improves almost by 8% compared with exponential segmentation in Byte Hit Ratio, in DRR though it only increases a little. Because the exponential algorithm only considers the last time and the file size, so it always replaces the last segment of file. But in VCR operation, we have known that the probability which the media we want is not in cache are much bigger, that is mean, we have to replace frequently, so exponential segmentation will inevitably replace the popular media, and finally reduce the Byte Hit Ratio.

In Fig.1 and Fig.2, we can conclude that CCRA can effectively improve the DRR when it is used in uniform segmentation algorithm, and used in exponential segmentation algorithm it can apparently increase the Byte Hit Ratio. So either uniform segmentation algorithm or exponential segmentation algorithm uses CCRA as its replacement algorithm can improve its performance. We know both of these algorithms are popular used in VOD, and through experiment comparison we see that CCRA has a better performance.

V. Conclusion

This paper analyzes recent exiting problems in popular segment-based caching replacement algorithm, and proposes a Common Caching Replacement Algorithm (CCRA). Through experimental comparison, CCRA has better performance than the traditional caching replacement algorithm. As alleviating the disk I / O bandwidth bottleneck limitation and improving the byte hit rate, CCRA plays a better role.

Acknowledgment

This paper is sponsored by the National High Technology Research and Development Program of China (No. 2008AA01A317) and the AnHui Colleges & Universities Provincial Science Research Project (No. KJ2008A106).

References

[1] Subhabrata Sen , Jennifer Rexford, Don Towsley. Proxy Prefix Caching for Multimedia Streams[A]. In Proceedings of IEEE INFOCOM'99. New York: 1999, 1310-1319

[2] Kun-lung Wu, Philip S. Yu, Joel L. Wolf. Segment Based Proxy Caching of Multimedia Streams[A]. ACM WWW10. Hong Kong: 2001, 36-44

[3] Song-qing Chen, Bo Shen, Susie Wee et al. Adaptive and lazy segmentation based proxy caching for streaming media delivery[A].Proc. ACM NOSSDAV. Monterey, CA: 2003, 22-31

[4] Song-qing Chen, Bo Shen, Susie Wee et al. Segment-Based Streaming Media Proxy Modeling and Optimization[J]. IEEE TRANSACTIONS ON MULTIMEDIA: 2006,8(2), 243-256

[5] OTOO E, OLKEN F, SHOSHANI A. Disk cache replacement algorithm for storage resource managers in data grids[A]. Proceedings of IEEE/ ACM Conference on Supercomputing [C] . Baltimore , Maryland, USA: 2002, 1-15.

[6] Pei Cao and Sandy Irani. Cost-aware WWW proxy caching algorithms. In USENIX Symposium on Internet Technologies and Systems: 1997, 193-206.

[7] C. C. Aggarwal and P. S. Yu. On disk caching of web objects in proxy servers. In Proc. Int'l. Conf.Info and Knowledge Management, CIKM'97, Las Vegas, Nevada: 1997, 238-245.

[8] Fahmi H,Latif M,Sedigh-Ali S,Ghafoor A,Liu P,Hsu L.Proxy servers for scalable interactive video support[J].IEEE Computer:2001,43(9), 54-60

[9] Sung-Ju Lee, Wei-Ying Ma, Bo Shen. An Interactive Video Delivery and Caching System Using Video Summarization[J]. Computer Communications: 2002,25(4), 424-435.

[10] Lei Guo, Songqing Chen, Zhen Xiao, Xiaodong Zhang. DISC: Dynamic Interleaved Segment Caching for Interactive Streaming[C]. ICDCS: 2005, 763-772