CBBS: A Content-based Bandwidth Smoothing Scheme for Clustered Video Servers [⋆]

Dafu Deng, Hai Jin, Xiaofei Liao, and Hao Chen

Cluster and Grid Computing Lab,
Huazhong University of Science and Technology, Wuhan, 430074, China,
{dfdeng,hjin,xfliao,haochen}@hust.edu.cn

Abstract. Due to the inherent bandwidth burstness, *variable-bit-rate* (VBR) encoded videos are very difficult to be effectively transmitted over clustered video servers for achieving high server bandwidth utilization and efficiency while guaranteeing QoS. Previous bandwidth smoothing schemes tend to cause long initial delay, data loss and playback jitters. In this paper, we propose a content-based bandwidth smoothing scheme, called CBBS, which first splits video objects into lots of small segments based on the complexity of picture content in different visual scenes so that one segment is exactly including one *visual scene*, and then, for each segment, a constant bit rate is allocated to transfer it. Performance evaluation based on real-life MPEG-IV traces shows that CBBS scheme can significantly improve the server bandwidth utilization and efficiency, while the initial delay and client buffer occupancy are also significantly reduced.

1 Introduction

Due to high scalability and low cost, clustered video servers [6] become inevitable to provide large capacity to serve thousands of concurrent clients. It is comprised of two parts: one RTSP server node and several RTP server nodes. The RTSP server is responsible for exchanging control messages with clients, while RTP servers are responsible for transferring video data to clients. Video objects are often divided into lots of fixed-length segments that uniformly distributed on RTP server nodes.

Usually, in order to guarantee QoS, for each stream, a bandwidth of b equaling to the peak bit rate of requested video object must be reserved in the corresponding RTP server nodes. Nevertheless, for VBR encoded video objects, the peak bit rate is far larger than the mean bit rate [8]. It indicates that the most reserved bandwidth is not used for most of the time. It tends to cause low server bandwidth utilization and better solutions are necessary.

To improve the network bandwidth utilization, previous works have proposed lots of schemes. Among them, a constant rate transmission and transporting (CRTT) scheme [3] employs *constant bit rate* (CBR) transmission of VBR video objects. It works by calculating the minimum bandwidth to prevent the client buffer underflow. Further, considering both the bandwidth and the client buffer size, it determines the amount of data

[⋆] This paper is supported by National 863 Hi-Tech R&D Project under grant No.2002AA1Z2102.

to be transmitted to the client in advance. Other schemes, such as the e-PCRTT [2], MCBA [1], DBA [8] and MVBA [4] [5], first prefetch video data until half of client buffer be filled, and then dynamically select the transmission bit rate in the ”river” constructed between the maximum transmission rate that guarantees no buffer overflow and the transmission rate that guarantees no buffer underflow. Since the transmission bandwidth is dynamically changed several times, the peak bit rate is smoothed somewhat. However, for the popular client buffer configurations, it tends to result in long initial delay or large initial bandwidth requirement that prefetch video data until half of client buffer be filled.

In this paper, we focus on the pre-recorded VBR video objects and propose a content-based bandwidth smoothing scheme, called CBBS, which can significantly improve the server bandwidth utilization and efficiency while guaranteeing QoS. The following sections are organized as follows. In section 2, we describe the content-based bandwidth smoothing scheme. Section 3 estimates the performance of CBBS via real-life MPEG-IV traces. Finally, section 4 ends with conclusions and future works.

2 Content-based Bandwidth Smoothing Scheme

The bit rate variety of VBR encoded videos is resulted from two issues. One is the picture content variety of different visual scenes, called *inter-scene* variable-bit-rate. It results in the size fluctuation of different *I*-frames. The other is the frame size variety in the visual scene, called *intra-scene* variable-bit-rate. Usually, in the same visual scene with same picture content, the size of *I*-frames are larger than that of *P*-frames and *B*-frames have smallest frame size. Tab.1 shows the quantitative analysis for different kinds of variable bit rate based on real-life MPEG-IV video traces[1]. Since *I*-frames are encoded by the basic content of visual scenes. In Table 1, we assume that one visual

Table 1. The quantitative analysis for different kinds of variable bit rate.

Movie	**Length**	**Std. dev. of**	**Std. dev. of**
	(frames)	***inter-scene*** **VBR**(Kb/s)	***intra-scene*** **VBR**(Kb/s)
Silence of the Lambs	89998	510.001	177.665
Mr. Bean	89998	364.201	243.740
Star Wars IV	89998	192.513	133.620
Jurassic Park	89998	490.650	241.544
Aladdin	89998	361.410	207.373
Robin Hood	89998	390.707	290.764
Sports-Soccer	89998	421.061	230.432
Sports-Formular-1	30334	373.620	223.432
News-ARD News	22498	375.941	309.414
News-ARD Talk	89998	339.640	235.097

[1] The traces can be obtained from the web site: http://www-tkn.ee.tu-berlin.de/ fitzek/TRACE/pics.

scene is comprised of one *group of pictures* (GOP) and use the bit rate variety among I-frames to represent the variable bit rate caused by the *inter-scene*. As one can see, the bit rate variety caused by the *inter-scene* is far larger than that caused by the *intra-scene*.

Based on the above analysis, allocating a *constant bit rate* (CBR) for transferring each scene would not result in large client buffer occupancy and long initial delay since the variable quantity of the *intra-scene* VBR is relative small. Thus, we can use a splitting scheme to divide video objects according to the picture content of visual scenes and allocate constant bandwidth for transferring each segment. After bandwidth allocation, the maximum client buffer requirement to guarantee no buffer overflowing, and the segment information, such as the start playback time, the allocated bandwidth, and the IP address of the storage RTP node, are available and can be maintained on the RTSP server node. Whenever the RTSP server admits a request, for each segment of requested video object, it just needs to query the RTP node whether they have enough bandwidth to transfer in the corresponding time interval. If so, it notifies RTP server nodes to reserve the corresponding constant bandwidth. Since the reserved bandwidth is the allocated constant bandwidth not the peak bit rate, the server bandwidth utilization is improved significantly.

Let d and $p(p \geq 0)$ be the initial delay and the threshold for splitting video objects, respectively. Sequence $\{\chi_1, \chi_2, \ldots, \chi_N\}$ represents the frame size sequence of the requested video object, where χ_i represents the size of the i-th frame. We define S to be the size of first I-frame in the segment starting from time point t_s and ending at time point t_e, and define b_{min} and B_{max} to be the minimum bandwidth at which clustered video servers may transmit and the maximum buffer occupancy at client side over a given interval $[t_s, t_e]$, respectively, where the client buffer must be guaranteed no underflowing and transmission is started from initial buffer level Q.

The video splitting procedure starts a new segment if and only if the processing frame is an I-frame, and the size of this I-frame χ_k satisfies the following equation.

$$|\chi_k - S| < p \times S \tag{1}$$

For the first segment, the allocated bandwidth is set to be the mean bit rate of the first segment. i.e.

$$b_1 = \frac{\sum_{t_s^1}^{t_e^1}}{t_e^1 - t_s^1} \tag{2}$$

In order to guarantee no buffer underflow during the first segment being in playback, at the time point of each frame for the first segment, the amount of data sent must larger than or equal to the amount of data consumed. Thus, we obtain

$$d = \max\{\frac{\sum_{j=1}^{t} \chi_j}{b_1} - t_e^1\} \quad t \in \{1, 2, \ldots, t_e^1\} \tag{3}$$

For other segments, we use following equation to calculate the allocated bandwidth.

$$b_{min} = max\{\frac{\sum_{j=1}^{t} \chi_j - (\sum_{k=1}^{t_s} \chi_k + Q)}{t - t_s}\} \quad t \in \{t_s + 1, t_s + 2, \ldots, t_e\} \tag{4}$$

Once the bandwidth b_{min} has been allocated, the maximum client buffer occupancy during the processing segment being played back and the initial buffer level Q for transferring next segment can be derived as follows.

$$B_{max} = max\{t \times b_{min} + Q - \sum_{k=t_s}^{t} \chi_k\} \quad t \in \{t_s+1, t_s+2, \dots, t_e\} \tag{5}$$

$$Q = b_{min} \times (t_e - t_s) - \sum_{k=ts}^{te} \chi_k \tag{6}$$

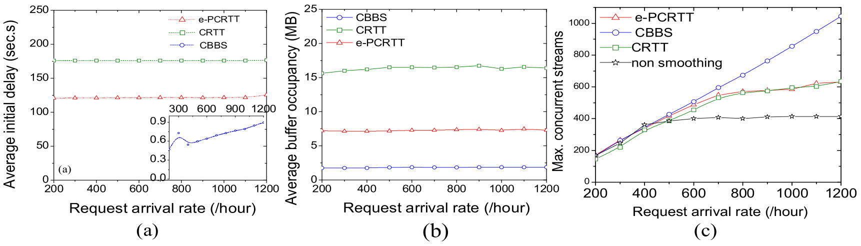
PROCEDURE FOR VIDEO SPLITTING AND BANDWIDTH ALLOCATION

INPUT: Video frame sequence $\{X_1, X_2, ..., X_N\}$, where N is the number of frames included in inputting video object.
OUTPUT: Initial delay, maximum client buffer occupancy and video segments with allocated bandwidth.

```
$S=X_1$, $Q=0$, $ts=1$, $te=1$, $b_1=0$, $d=0$, $b_m=0$, $B=0$, $B_{max}=0$;   //$X_1$ is the first $I$-frame in frame sequence.
FOR ($k=1;k<=N;k++$){
    IF ($S==X_1$){                      // processing the first segment.
      IF (($X_k$ is not an I-frame)||($X_k$ is an $I$-frame)&&(|$X_k$-$S$|<$pS$)){
          $b_1 += X_k$;
      }
      ELSE {
        $b_1=b_1/k, te=k$;
        calculate $d$, using equation (3).
        ts=-d;
        calculate $Q$ and $B_{max}$, using equation (6) and (5), respectively.
        IF ($B<B_{max}$) $B=B_{max}$;
        output the first segment stating from $ts$ and ending at $te$, with allocated bandwidth $b_1$, and
        initial delay $d$;
        $ts=k$, $S=X_k$;
      }
    }
    ELSE {                          //processing other segments.
      IF(($X_k$ is not an $I$-frame)||($X_k$ is an $I$-frame)&&(|$X_k$-$S$|<$pS$)){
          calculate $b_m$, using equation (4);
      }
      ELSE {
          $te=k$;
          calculate $Q$ and $B_{max}$, using equation (6) and (5), respectively;
          IF ($B<B_{max}$) $B=B_{max}$;
          output a video segment stating from $ts$ and ending at $te$, with allocated bandwidth $b_m$.
          $ts=k$, $S=X_k$;
      }
    }
}
output the maximum client buffer occupancy $B$.
```

Fig. 1. Pseudo-code of video splitting and bandwidth allocation algorithm.

The algorithm of video splitting scheme and bandwidth allocation scheme is presented formally in Fig.1. Notations used in this figure have been defined above.



(a) (b) (c)

Fig. 2. Comparison of (a) the average initial delay, (b) the average buffer occupancy, and (c) the maximum concurrent streams among e-PCRTT,CRTT,and CBBS with splitting threshold $p = 0.4$.

3 Performance Evaluation

For the illustrative purpose, we evaluate the performance of the CBBS scheme by the experiment and compared it with that of the e-PCRTT and CRTT scheme. In the experiment, the clustered video servers are the prototype of Turbogrid streaming servers [2] with one RTSP server node and 8 RTP sever nodes. Each node uses 1.4GHz CPU and 100Mb/s NIC. Real-life MPEG-IV traces with different contents including *Movies*, *Sports*, *News*, *Talk show*, *several Episodes*, and *Cartoon* are splintered into lots of small segments based on the proposed video splitting scheme with threshold $p = 40\%$. The length of each video trace is 89,998 frames. All video traces are played back at a frame rate of $F = 25$ frames/s.

There are three kinds of popular clients used in our experiment-the multimedia PDA with buffer capacity 8 Mbytes, the set-top box with buffer capacity 32 Mbytes and the PC with buffer capacity 64 Mbytes. Client requests are generated using the Poisson arrival process with an interval time $1/\lambda$. The arrival rate λ is varied from 200 to 1200 per hour. Once generated, client selects a video object and sends the request to clustered video servers. If the request is admitted, the client simply playbacks the received stream until the transmission is completed.

Fig.2 plots the performance comparison among e-PCRTT, CRTT and CBBS schemes with splitting threshold $p = 0.4$, where the inner figure of part (a) is the magnification of the initial delay for CBBS with $p = 0.4$. From this figure, we can easily find that CBBS scheme significantly outperforms other two schemes. For example, the average initial delay and the average maximum buffer requirement of CBBS scheme is less than 1 second and 2MB, respectively, whereas those of the e-PCRTT scheme and the CRTT scheme needs approximately 120 seconds, 8MB and 160 seconds, 16MB, respectively. For the bandwidth utilization which can be indicated via the maximum concurrent streams supporting by the clustered video servers, CBBS scheme can support 1044

[2] Turbogrid streaming servers are developed by Cluster and Grid Computing Lab of Huazhong University of Science and Technology.

concurrent streams, while e-PCRTT and CRTT schemes can just support approximately 600 concurrent streams.

4 Conclusions and Future Works

In this paper, we propose a content-based bandwidth smoothing scheme, called CBBS, which can significantly improve server bandwidth utilization and efficiency while guaranteeing QoS. Unlike previous schemes, CBBS scheme first splits video objects into small segments based on the complexity of picture content in different scenes. Then, it uses constant bit rate to transfer each segment so that the *intra-scene* variable bit rate can be effectively smoothed. When admitting a request, CBBS scheme accurately judges whether the remaining bandwidth of RTP nodes is enough to transmit the stored segments in the corresponding time interval. It significantly reduces the effect of the *inter-scene* variable bit rate on the server bandwidth utilization.

On going researches include:

1. Evaluating the effect of splitting threshold p on the performance of clustered video servers and deriving the optimal p based on statistically analysis of large amount of real-life video traces;
2. Developing optimal disk retrieving models and strategies to work with the scene-based video striping scheme;
3. Designing a time-scaled resource reserving protocol to reduce the impacts of traffic burstness and improve network utilization over the Internet.

References

1. H. Chao, C. L. Hung, Y. C. Chang, and J. L. Chen, ”Efficient changes and variability bandwidth allocation for VBR media streams”, *International Journal of Network Management,* Vol. 12, 2002, pp. 179-185.
2. O. Hadar and R. Cohen, ”PCRTT Enhancement for off-line video smoothing”, *Real-Time Imaging,* Vol. 7, 2001, pp.1-14.
3. J. M. McManus and K. W. Ross, ”Video on demand over ATM: Constant-rate transmission and transport”, *IEEE Journal of Selected Areas in Communication,* Vol. 14, 1996, pp. 1087-1098.
4. A. R. Reibman and A. W. Berger, ”Traffic descriptors for VBR video teleconferencing over ATM networks”, *IEEE/ACM Transactions on Networking,* June, 1995.
5. J. D. Salehi, Z. L. Zhang, J. F. Kurose, and D. Towsley, ”Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing”, *Proc. of ACM SIGMETRICS,* 1996.
6. C. Shahabi, R. Zimmermann, K. Fu, S. Yuen, and D. Yao, ”Yima: a Second-Generation Continues Media Server”, *Computer,* pp.56-64, Jun. 2002.
7. H. Zhang and E.W. Knightly, ”Red-vbr: a renegotiation-based approach to support delay-sensitive vbr video”, *ACM Multimedia Systems,* Vol. 5, 1997, pp. 164-176.
8. L. Zhang and H. Fu, ”A novel scheme of transporting pre-stored MPEG video to support video-on-demand (VoD) services”, *Computer Communications,* Vol. 23, 2000, pp.133-148.