

On Design of Scalable P2P Video Caching

Alex Bikfalvi
IMDEA Networks
Av. del Mar Mediterraneo, 22
Leganés, Madrid, 28918,
Spain
alex.bikfalvi@imdea.org

Jaime García-Reinoso
University Carlos III of Madrid
Leganés, Madrid, 28911,
Spain
jgr@it.uc3m.es

Nazanin Magharei
Department of Computer and
Information Science
University of Oregon
Eugene, OR, 97403, USA
nazanin@cs.uoregon.edu

Reza Rejaie
Department of Computer and
Information Science
University of Oregon
Eugene, OR, 97403, USA
reza@cs.uoregon.edu

ABSTRACT

[write the abstract here]

1. INTRODUCTION

[write the introduction here]

2. RELATED WORK

Video streaming caching is not a new idea, and it has been studied extensively in the recent years. Recognizing the particular properties of the multimedia streams, authors have proposed different caching strategies for the purposes of reducing the load on the network and multimedia server, and maximizing the delivered video quality.

For client/server architectures, proxy caches borrow from the concepts of Web caching. One example is the work of Sen et al. [9], where they assert that user experience is improved by storing the beginning (or prefix) of the video and shielding the clients from a long delay when they start the playback. In a different approach, Rejaie et al. [6, 7] propose a caching mechanism with a congestion control mechanism. They assume the video stream is divided into segments and encoded in several layers. When possible, the entire video is cached, and cache replacements are done per-layer basis, starting from the end of the layer with the lowest popularity.

In a similar approach, Wu et al. [11], enhance the cache utilization efficiency by dividing the video object into segments of exponentially increasing sizes, effectively assigning a higher preference to beginning segments. Popularity is determined at the segment level, allowing a fine-grained control over segment replacement. The beginning segments

are cached using a strictly least-recently used (LRU) policy, while for the rest the authors apply a modified LRU depending on segment size.

In [5], the authors developed a strategy that caches video objects based on the user request pattern. As opposed to calculate objects popularity based on past requests, their explicit tracking method uses the request history to estimate future requests, and cache the video objects accordingly. Ma and Du [4] propose a segment-based caching method that stores alternating segments (chunks). In [2], the authors present a generalized caching policy that maximizes the number of cached streams by storing short intervals of successive streams.

The increasing interest and research effort dedicated to peer-to-peer video streaming architectures, as opposed to the traditional client/server, has also generated many studies in the area of peer-to-peer caching. One example is PROP [3, 1] where the authors examine the performance benefits offered by peer-to-peer caching complementing a classic segmentation-based proxy design. They show that video segment popularity is not a good metric for caching decision, and propose a new utility function that maximizes the caching probability for segments with mid-range popularity.

O. Saleh and M. Hefeeda [8] propose a generalized caching method for peer-to-peer content having a Mandelbrot-Zipf popularity. Exploring the impact of this popularity model on the cache efficiency, they conclude that objects must be cached incrementally proportional to their popularity. However, their study is focused on caching of atomic object, such shared files, rather than continuous streams.

[to do: examine related work again]

3. PROXY VS. P2P CACHING

[the rational behind our proposal - in line with Reza's items]

4. P2P VIDEO CACHING (PVC)

[describe the design concepts here]

4.1 Algorithm Design

[explain the main intuition behind the design: the difference between global and local popularity; how we estimate local popularity and its trend; design decision, i.e. what we cache depending on popularity]

4.2 Reactive vs. Proactive

[why proactive, what it is improving, how does it work]

5. PERFORMANCE EVALUATION

In this section, we present the preliminary evaluation of the PVC algorithm. For this purpose, we developed a time-domain event-based simulator that emulates the video-on-demand streaming between a population of peers. The main performance metric is the bit-hit ratio that quantifies the fraction of streams that are delivered from the peers' cache. Additionally, we measure the number of replicas (or copies) for each stream, in order to identify whether the proactive version of the PVC algorithm manages to reach a better balance between popular and unpopular streams. We ran the simulation several times for every set of input values.

5.1 Evaluation Setting

For the generation of workload we relied on Medisyn, a video-on-demand workload generator developed by HP labs [10]. Its authors used a set of traces collected from their enterprise VoD system to create a model for the users video watching behavior. Although the program is tuned by default to the values that would generate a workload consistent to their traces, the model and Medisyn are flexible enough to allow adjustments to the input parameters.

Since we realize that using only the default workload generated by Medisyn is insufficient to capture the behavior of our algorithm in a different setting, one of our goal is to measure the cache performance under various stream popularity and life span conditions.

5.2 Impact of the Cache Size

5.3 Stream Popularity

Several authors have identified the Zipf distribution (or a slightly modified version) to accurately model the stream popularity [10, 12] but with the distribution's parameters depending largely on the system where the data was collected.

5.4 Stream Lifespan (Burstiness)

5.5 Peer Uplink Bandwidth

5.6 Stream Replication

In the proactive PVC algorithm, when sender peers accept incoming requests, they already have a well defined view on the popularity of the video stream in question and they pass this information to their receivers as a *cache* indication for streams getting hot, respectively *don't cache* for streams getting cold. The rationale behind this approach is to prevent the spread of too many copies for the streams with a decreasing popularity. In all classic cache algorithms and including the reactive PVC variant, peers will cache the stream unless they already have a well established history.

This is disadvantageous in particular for very popular streams with many new viewers, leading to an over-replication of highly popular content, at the expense of less popular streams. The *cache/don't cache* indication passes this information from old to new peers so such streams are not excessively cached. In order to determine the improvement of the proactive PVC over the classic algorithms and the reactive version, in figure XXX we examine the cumulative distribution function of the number of stream replicas. The figure reveals that the stream replicas are more uniform distributed in PVC-P, whereas LRU, LFU and PVC-R allocate the largest cache slice only to very popular streams.

This results justifies why PVC-P outperforms PVC-R in certain situations

6. REFERENCES

- [1] S. Chen and X. Zhang. Design and Evaluation of a Scalable and Reliable P2P Assisted Proxy for On-Demand Streaming Media Delivery. *IEEE Transactions on Knowledge and Data Engineering*, 18(5):669–682, 2006.
- [2] A. Dan and D. Sitaram. Multimedia caching strategies for heterogeneous application and server environments. *Multimedia Tools and Applications*, 4(3):279–312, 1997.
- [3] L. Guo, S. Chen, S. Ren, X. Chen, and S. Jiang. PROP: a scalable and reliable P2P assisted proxy streaming system. In *INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS*, volume 24, pages 778–786. IEEE Computer Society; 1999, 2004.
- [4] W. Ma and D. Du. Reducing bandwidth requirement for delivering video over wide area networks with proxy server. *IEEE Transactions on Multimedia*, 4(4):539–550, 2002.
- [5] M. Reisslein, F. Hartanto, and K. Ross. Interactive video streaming with proxy servers. *Information Sciences*, 140(1-2):3–31, 2002.
- [6] R. Rejaie, M. Handley, H. Yu, and D. Estrin. Proxy caching mechanism for multimedia playback streams in the internet. In *Proc. International Web Caching Workshop*. Citeseer, 1999.
- [7] R. Rejaie, H. Yu, M. Handley, and D. Estrin. Multimedia proxy caching mechanism for quality adaptive streaming applications in the internet. In *IEEE INFOCOM*, volume 2, pages 980–989. Citeseer, 2000.
- [8] O. Saleh. Modeling and caching of peer-to-peer traffic. 2006.
- [9] S. Sen, J. Rexford, and D. Towsley. Proxy prefix caching for multimedia streams. In *IEEE INFOCOM*, volume 3, pages 1310–1319. Citeseer, 1999.
- [10] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat. Medisyn: A synthetic streaming media service workload generator. In *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, pages 12–21. ACM New York, NY, USA, 2003.
- [11] K. Wu, P. Yu, and J. Wolf. Segment-based proxy caching of multimedia streams. In *Proceedings of the 10th international conference on World Wide Web*,

pages 36–44. ACM New York, NY, USA, 2001.

- [12] H. Yu, D. Zheng, B. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. *ACM SIGOPS Operating Systems Review*, 40(4):344, 2006.