

Proxy Caching Strategies for Internet Media Streaming

Manuela Pereira

University of Beira Interior, Portugal

Mário M. Freire

University of Beira Interior, Portugal

INTRODUCTION

Media streaming consists in the viewing of dynamic media information while being downloaded by clients. With the explosive growth of the Web and the mature of digital video technology, *media streaming* has received a great deal of interest as a promising solution for multimedia delivery services. This approach allows that media objects can be accessed in a similar way to conventional text and images using a download-and-play mode. However, unlike static text-based content, proxy caching has difficulty in delivering streaming media content because media objects are usually very large and its transmission consumes a great amount of network resources, prolongs startup latency, and threatens the playback continuity. The size of a conventional *Web object* is typically on the order of 1–100 kbytes and, therefore, a decision regarding either caching or not an object in its totality is an easy task (Liu & Xu, 2004). However, the size of *media objects* is very large, reaching a size on the order of several hundreds of Mbytes or even Gbytes. Therefore, caching a whole media object at a Web proxy optimized for delivering conventional small-size Web objects is not feasible, since large streams would quickly exhaust the capacity of the proxy cache. Besides, the streaming of *media objects* requires a significant amount of resources such as disk space and network bandwidth, which need to be maintained during a long period of time. Moreover, the long playback duration of a streaming may allow several client-server interactions. Therefore, access rates might be different for different parts of a stream, which makes cache management potentially more complex, as pointed out by Liu and Xu (2004). On the other hand, a download-before-playing solution provides continuous playback, but it also introduces a large startup delay.

An effective solution to reduce client-perceived latencies and network congestion is to cache data at proxies widely deployed across the Internet. This solution, besides inexpensive, also leads to an improvement of both availability of objects and packet losses since redundant network transmission decreases while transmission efficiency increases. However, proxies are generally optimized for delivering

conventional small-size Web objects, which may not satisfy the requirements of streaming applications. Due to these particular features of media objects, novel caching strategies have been proposed.

With the evolution of the Internet as the dominant architecture for applications, contents, and services, these are gradually migrating from the *client-server paradigm* to the edge services paradigm and to the peer-to-peer (P2P) computing paradigm. Recently, P2P system has received a great amount of interest as a promising scalable and cost-effective solution for next-generation multimedia content distribution. This kind of systems have advantages regarding systems based on the client-server paradigm, namely improved scalability and reliability, cheaper infrastructures due to direct communication among peers, and easiness of resource aggregation in order to provide, for instance, massive processing power (Ye, Makedon, & Ford, 2004). However, P2P systems also have some drawbacks, namely the considerably more complex searching and node organization and security issues (Aberer, Puceva, Hauswirth, & Schmidt., 2002). Therefore, this article limits the discussion to low-cost proxy caching strategies for *media streaming* over Internet.

BACKGROUND

As discussed earlier, media caching has different requisites regarding conventional Web caching due to the special features of media streaming. Since the content of a media object is rarely updated, management issues like cache consistency and coherence are less critical in media caching. However, it requires an effective management of proxy cache resources due to the resource requirements of media objects (Liu & Xu, 2004).

Roughly, there are two main types of caching strategies: the strategies focused on homogeneous clients and the strategies focused on heterogeneous clients. Most of the proposed strategies are focused on homogeneous clients, which have identical or similar configurations and capabilities behind a proxy. Figure 1 presents an overview of caching strategies

for media streaming. A brief description of these strategies is provided in the next sections.

CACHING STRATEGIES FOR HOMOGENEOUS CLIENTS

Strategies for homogeneous clients can be classified, regarding the parts of media objects to cache, as *prefix caching*, *sliding-interval caching*, *segment-based caching*, and *rate-split caching*. A brief description of these strategies follows.

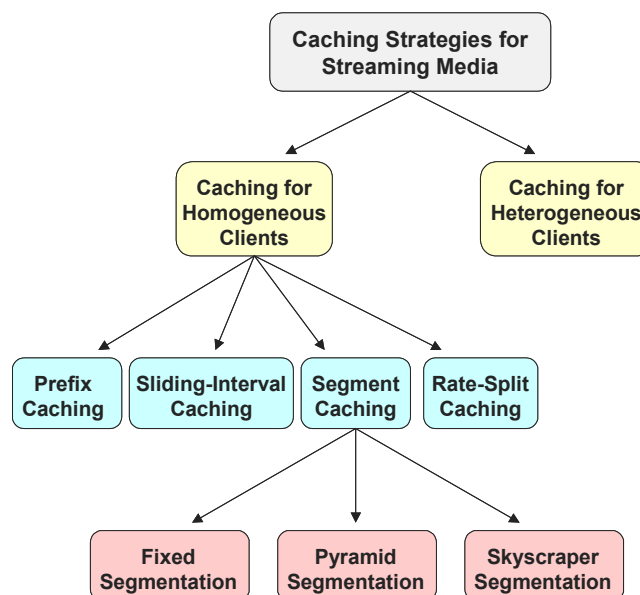
Prefix Caching

According to this strategy, the media object is divided into two parts: the prefix and the suffix. The prefix is cached at a proxy. After the reception of a client request, the proxy immediately delivers the prefix to the client and fetches the suffix from the source server to be further delivered to the client. This strategy leads to a significant reduction of the startup delay for a playback since the proxy is generally closer to the clients than the source server (Liu & Xu, 2004; Miao & Ortega, 1999; Sen, Rexford, & Towsley, 1999). In this strategy, the prefix size is a key issue for the system performance. In general, this strategy leads to a moderate bandwidth reduction but to a high startup latency reduction.

Sliding-Interval Caching

According to this strategy, a sliding interval of a media object is cached in order to exploit the sequential access of a streaming media. For instance, if two consecutive requests for the same object are received, the first request may access the object from the server and incrementally store it into the proxy cache, while the other request only needs to access the cached portion and release it after the access. In general, if multiple requests for a given object are received in a short period of time, a set of adjacent intervals may be grouped to form a run, of which the cached portion will be released only after the satisfaction of the last request. This strategy can substantially reduce the consumption of network bandwidth and the startup delay for subsequent accesses. However, it involves high disk bandwidth utilization because the cached portion is dynamically updated with the playback. Besides, the effectiveness of the sliding-interval caching strategy diminishes with increased access intervals. Moreover, in the case of the access interval of a given object be longer than the duration of the playback, it degenerates to the case of full-object caching (Liu & Xu, 2004; Tewari, 1998). These limitations may be mitigated if the cached content is retained over a relatively long period of time. Nevertheless, for a good cache design, this strategy may lead to a high bandwidth reduction and to a high startup latency reduction.

Figure 1. Overview of caching strategies for media streaming



3 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/proxy-caching-strategies-internet-media/14043

Related Content

Integrating OLAP/SOLAP in E-Business Domains: An Empirical Study

Oualid (Walid) Ben Ali and Samar Mouakket (2013). *Managing Information Resources and Technology: Emerging Applications and Theories* (pp. 196-211).

www.irma-international.org/chapter/integrating-olap-solap-business-domains/74509

ADPD and SW-CMM

Fabrizio Fioravanti (2006). *Skills for Managing Rapidly Changing IT Projects* (pp. 158-175).

www.irma-international.org/chapter/adpd-cmm/29007

Successful HIT Requires Inter-Team Communication

Charles H. Andrus and Mark Gaynor (2013). *Journal of Cases on Information Technology* (pp. 1-6).

www.irma-international.org/article/successful-hit-requires-inter-team-communication/102714

Hybrid Framework for a Robust Face Recognition System Using EVB_CNN

Tamilselvi M. and S. Karthikeyan (2021). *Journal of Cases on Information Technology* (pp. 43-57).

www.irma-international.org/article/hybrid-framework-for-a-robust-face-recognition-system-using-evbcnn/277656

Optimizing the Configuration of Development Teams Using EVA: The Case of Ongoing Project Adjustments Facing Personnel Restrictions

Alexander Baumeister and Alexander Floren (2013). *Perspectives and Techniques for Improving Information Technology Project Management* (pp. 268-283).

www.irma-international.org/chapter/optimizing-configuration-development-teams-using/73240