Efficient Support of Streaming Videos through Patching Proxies in the Cloud

Kuei-Chung Chang, Feng Chia University, Taichung, Taiwan
Kuan-Hsiung Wang, Feng Chia University, Taichung, Taiwan

ABSTRACT

Multimedia streaming applications such as mobile conferencing, e-learning and mobile cinema are becoming popular for the Internet and mobile devices. Patching can enable a client to receive multicast streaming by listening to ongoing transmission of the same video clip in order to reduce retransmission from the media server. However, multimedia streams multicasting to clients must be stored in the client buffer in advance; most fractions of ongoing streaming frames are lost due to limited space. The authors first propose a patching cache scheduling algorithm to patch and cache the shared frames of the ongoing streams on cloud-based patching proxy servers. The patching proxy cloud is responsible for patching and caching streaming data and mobile devices in the same cloud share bandwidth and cooperatively stream video frames to distribute the load. The simulation results show that the proposed patching proxy cloud can efficiently improve the cache utility, decrease the user-perceived latency, and reduce the transmission overhead between the proxy and the original media server when a video clip is very popular.

Keywords: Cache, Cloud System, Mobile, Multimedia Streaming, Patching, Proxy Server

INTRODUCTION

Due to the growing popularity of powerful mobile devices, mobile applications are a significant new application form for the next generation Internet. Especially, multimedia streaming applications (Chang & Chen, 2007; Hua & Xie, 2007; Eichhorn, Schmid, & Steinbach, 2008) are becoming more and more popular for the Internet and mobile devices. They consume a significant amount of server and network resources due to the high bandwidth and long duration of audio and video clips.

If a video is popular on the media server, then a large number of requests will overload the server and cause network congestion. In order to mitigate this problem, researchers have recently investigated the use of network multicast for video streaming, in which a multicast video stream can be shared by more than one client. There are two approaches for multicast streaming algorithms: patching and caching (Wong, Lee, Li, & Chan, 2007). A client can cache one or more of the ongoing multicast...
video streams in the client buffer to share the transmitted data but will be unable to begin playback immediately because the initial portion of the video stream has missed. To tackle this problem, the client can request another video stream to transmit the missing initial video portion in order to enable playback to begin; this is called patching. We can dynamically expand the patching (Cai, Hua, & Vu, 1999; White & Crowcroft, 2000) to serve more clients by offering multicast-based services for streaming multimedia in a mobile environment (Dutta, Chennikara, Chen, Altintas, & Schulzrinne, 2003; Mancuso & Bianchi, 2004). This can reduce the server and the network overhead by allowing a client to receive multimedia streaming from ongoing transmission of the same video clip.

Proxy-based patching is also useful if multicasting capability is not available on an end-to-end basis from the content server to clients. For example, in a heterogeneous inter-networking environment, a proxy server in a domain close to clients may receive the video on a unicast connection from the media server and may multicast the stream to downstream clients. Patching at the proxy reduces the bandwidth consumed on the path from the multimedia source to the proxy and from the proxy to clients. The streaming proxy can cache streaming frames in memory and quickly forward those frames to clients. This reduces the server workload and the network traffic if the streaming proxy can efficiently cache frames (Bellavista, Corradi, & Giannelli, 2005; Jiang, Ge, & Li, 2005; Mancuso & Bianchi, 2004).

The Greedy Buffer Reuse (GBR) algorithm (Sen, Gao, Rexford, & Towsley, 1999a) minimizes the required transmission bandwidth by allowing clients to patch from multiple ongoing transmissions. It maximizes the number of frames that a new client can retrieve from the most recently initiated ongoing complete transmission by scheduling the transmission channel. However, the limited client buffer can only cache few fractions of streaming frames. Therefore, the client needs to retransmit other un-cached frames from the server again and again.

In this paper, we first propose a patching cache scheduling algorithm for the patching proxy to efficiently cache frames from the ongoing streams. The patching proxy is able to generate a proxy cache schedule to dynamically cache frames which can be requested again later. It caches frames that are lost due to limited client buffer space; thus, the proxy cache can be regarded as the extension buffer of clients. The objective is to improve cache effectiveness, decrease the user-perceived latency, and reduce the traffic between the proxy and the content original server by properly scheduling the patch frames in the fixed-capacity proxy. From the viewpoint of a cloud consumer, the proposed patching proxy can dramatically reduce the server and network overheads. In large-scale systems like mobile networks, handheld devices can retrieve video streams from the patching proxy, so that devices can play video streams smoothly.

In addition, the proposed patching proxy cloud can dynamically deploy a new patching proxy to support overloaded requests in order to provide more storage and computing power for mobile users. Cloud computing has recently emerged as a new computing trend providing a shared pool of specific servers in datacenters to a cloud infrastructure that can provide on-demand computing and storage resources for users (Wu, Wu, Li, Qiu, & Lau, 2011; Jin & Kwok, 2010; Huang, Mei, Li, & Woo, 2011).

The remainder of this paper is organized as follows. First, we review related researches in the cloud-based computing and multimedia streaming patching mechanism. Then, we describe the proposed cloud-based patching proxy architecture and the patching scheduling algorithm. In the section after that, we show the experiment environment and simulation results. Finally, we conclude the paper.
Credential Management Enforcement and Secure Data Storage in gLite
[www.igi-global.com/article/credential-management-enforcement-secure-data/39043?camid=4v1a](www.igi-global.com/article/credential-management-enforcement-secure-data/39043?camid=4v1a)