Journal of Network and Computer Applications 37 (2014) 293–314

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

An ID-based approach to the caching and distribution of peer-to-peer, proxy-based video content

Conor Cameron *, Ibrahim Khalil, Zahir Tari

School of Computer Science and Information Technology (CS&IT), RMIT University, Melbourne, Victoria, Australia

ARTICLE INFO

Article history:
Received 7 September 2012
Received in revised form
31 January 2013
Accepted 6 March 2013
Available online 13 March 2013

Keywords:
Peer-to-peer
Proxy
Caching
YouTube
Video
ID

ABSTRACT

The viewing of streamed video content has become second nature these days for many Internet users. With such copious amounts of data being transferred between hosts involved in this streaming, it is prudent to establish some method of effectively minimising the resultant outbound network traffic, whilst providing the client with a good Quality of Service (QoS). Although many approaches have attempted to address the aforementioned issues, most tend to focus on directly aiding the content provider, rather than the client and typically require a lookup mechanism, such as a distributed hash table (DHT), which engenders a storage and complexity overhead. In this paper we propose a novel infrastructure to the aim of enhancing QoS, whilst reducing outbound network traffic, for clients downloading YouTube video content. We achieve this without a reliance upon request flooding or large hash table overheads inherent to DHT systems. Augmenting this, we proffer two additional mechanisms, the first balances localised storage-peer (client) load and the second preemptively balances super-peer (proxy) load. Lastly, we provide a peer failure mechanism, which allows the proposed system to quickly and efficiently recover from peer failures. Results garnered from emulations indicate that the proposed architecture offers respectable cached video retrieval speeds, even under dynamic and high traffic conditions, thus precluding typical client video lag and/or buffering.

1. Introduction

Increasingly over the past few years, Internet users have been flocking to video on-demand and streaming sites such as YouTube, Hulu and DailyMotion. At present, YouTube is the most popular and widely-known of these sites and according to the *Alexa Ranking System* is also the third most popular website globally (Inc. Alexa Internet, 2012). Moreover, studies suggest that YouTube traffic accounts for approximately 20% of all HTTP traffic (or nearly 10% of all traffic on the Internet) (Cheng et al., 2007). Lastly, businesses specialising in providing Internet usage, such as Internet cafes, would likely receive a large quantity of requests for YouTube videos and thus could benefit greatly from the proposed system. Such organisations would require a system that is reliable, yet that optimally does not engender the potential complexities and communication overhead of implementing and maintaining a DHT-based system. Intuitively, such a system would also need to provide both: bandwidth savings and respectable video download speeds.

The existing research has tended to focus more directly on the alleviation of the workload at the provider, rather than on the issues and workload of the client (Guo et al., 2007a; Cheng et al., 2008). Moreover, very few proposed solutions are directly aimed at YouTube or similar sites themselves. Of late, there has been significant research into the application of the P2P paradigm in the congruent field of Video on Demand (VoD) systems (Do et al., 2004; Fujimoto et al., 2011; Alhaisoni et al., 2009; Yiu et al., 2007), further advocating the usefulness of the P2P architecture for the proposed research. A portion of the existing research is also aimed at solutions to on-demand movie content and thus tends to stripe video content across multiple peers (Acharya and Smith, 2000; Ip et al., 2007), which could increase lookup latency and the potential for video loss. Furthermore, due to their relatively small size, segmenting is not necessary for the great majority of YouTube videos and thus is not appropriate for the proposed system.

* Corresponding author.
E-mail address: conor_cameron@student.rmit.edu.au (C. Cameron).

1084-8045/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2013.03.005

We address the aforementioned issues at the client-side via the provision of an economically feasible peer-to-peer architecture, which can significantly reduce the prolific number of external YouTube requests for the local network while offering fast, reliable video lookups for all clients involved. The proposed architecture comprises the follows contributions:

- Successfully balancing storage-peer load: Via extensive experiments, we demonstrate that the proposed architecture is capable of successfully transferring popular video content, and thus request load, from inundated storage-peers to neighbouring storage-peers. This is essential in ensuring that a minimal number of video requests are dropped and, more importantly, that clients are able to consistently achieve a good quality video retrieval speed. We also demonstrate that the proposed architecture is able to maintain an efficient video retrieval speed as new peers continue to enter the system. This is crucial in ensuring that the proposed system is able to cope with networks and topologies of differing sizes effectively. This is accomplished by the two proposed load-balancing mechanisms, namely global load-balancing and localised load-balancing. The former achieves this via a dissemination of video ID responsibility across all peers. Whereas the latter achieves this via a transferal of popular videos from a flooded storage-peer to a neighbouring storage-peer.
- Successfully providing a high degree of video retention: The proposed system provides a very high degree of video retention, notably under conditions involving very few peer departures and failures, thus ensuring the greatest proportion of outgoing bandwidth is preserved. We achieve this via the inherent definitive routing infrastructure of the proposed system. As a result of this mechanism, each downloaded video will always have a primary, definitive location, based on its inherent ID, and thus will always be discoverable, assuming said video has already been uploaded to the relevant peer.
- Successfully balancing super-peer request load: Through simulations we show that our system is able to balance super-peer request loads via our cooperative super-peer request handling (CSPRH) mechanism. This is an essential component of any hybrid P2P architecture, as a heavy influx of requests could otherwise render a given super-peer unavailable until the load ceases, unless said peer's request load is otherwise balanced. We are able achieve this as a result of the super-peer neighbourhood paradigm of CSPRH, which will be explained in subsequent sections. Probabilistically, this mechanism will achieve a uniform dispersion of video requests, for a given super-peer, across the neighbourhood of said super-peer.
- Successfully and efficiently recovering from peer failures: We demonstrate graphically that our architecture is able to handle failures at both the super-peer and storage-peer level in a highly efficient manner. This is another crucial element of any video on-demand system, as the architecture must be able to provide a strong degree of reliability and, notably in a P2P network, be able to dependably protect against unexpected peer dropouts. We achieve this via the use of recovery information inherent to the super-peer neighbourhood paradigm within CSPRH. Simply put, the information employed to evenly disperse video requests across a super-peer neighbourhood can also be used to expeditiously recover from super-peer failures within that neighbourhood.

The remainder of this paper is organised as follows. Section 2 describes the proposed architecture in detail, along with its functional components. In Section 3, we provide an explanation of how peer arrivals and departures operate within the system. The issue of peer failures in the proposed architecture is addressed in Section 4. Section 5 provides a detailed description of the two proposed load-balancing mechanisms. Section 6 discusses the proposed cooperative super-peer request handling mechanism, and Section 7 summarises the experimental results. A review of pertinent existing research is provided in Section 8 and compared with the proposed work. Section 9 summarises the benefits of the proposed architecture, its applications and the drawbacks that it entails and also mentions future research that we plan to undertake.

2. System components and architectural design

The proposed research is directed towards a privatised architecture, in which local clients gain access to each other, with each client representing a *storage peer*. These storage-peer clients will connect through peer-proxies, that is, through *super peers*, which will negotiate access to the rest of the network on their behalf. Each super-peer will be responsible for the storage-peers that reside under the same video ID domain as itself. In other words, the super-peers handle video requests on behalf of their child storage-peers, but do not directly serve videos themselves. Conversely, storage-peers store and distribute video content to other storage peers via their respective parent super-peer. This type of architecture is generically known as a Hybrid P2P architecture and constitutes the foundation of the proposed system.

The architecture itself draws upon exclusive, internal *YouTube Video IDs*, which enable us to provide a novel approach to the retrieval and distribution of video content across peers, in a manner that is beneficial to clients implementing the proposed system. Each video ID is a sixteen character, alphanumeric string, which is used internally by YouTube to identify and subsequently serve a specific video to a client. It is noteworthy that, although this specific ID structure is unique to YouTube, the notion of using unique video IDs, especially for video on-demand websites, is commonplace. This is because each video is typically stored in a manner that requires a unique key lookup, be it from a database or hash table. In essence, the proposed architecture engenders a *definitive routing infrastructure*, which allows fast and precise video lookups and good quality video retrievals, without the need for distributed hash table routing mechanisms or request flooding.

2.1. Video distribution and retrieval

Each storage-peer distributes its downloaded video content to the storage-peer in the system that is responsible for said video content, based on the video's ID. This is accomplished via the storage-peer's parent super-peer, who will receive the video request from their child storage-peer before forwarding it to either: the parent super-peer, of the storage-peer responsible for the video ID, or a super-peer proximal to the parent super-peer, based on ID. For simplicity, we will currently restrict our discussion to the parent super-peer and will leave the proximal super-peers for a subsequent section, which details cooperative super-peer request handling (CSPRH). The video request itself will contain the ID of the video to be distributed, as well as the IP address and port of the originating storage-peer, so that the