OpenCache: Exploring Efficient and Transparent Content Delivery Mechanisms for Video-on-Demand

Matthew Broadbent
School of Computing and Communications
Lancaster University
Lancaster, LA1 4WA, UK
m.broadbent@lancaster.ac.uk

Nicholas Race
School of Computing and Communications
Lancaster University
Lancaster, LA1 4WA, UK
n.race@lancaster.ac.uk

ABSTRACT

Watching 'on-demand' video content is a popular activity facilitated by the Internet. Existing distribution methods typically rely on unicast flows to deliver this media. This can be inefficient if identical content is delivered at a later point in time. We propose a caching platform to store content locally whilst maintaining the underlying delivery method. This paper introduces OpenCache, a prototype network-aware caching platform that utilises emerging OpenFlow technology.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications

Keywords

Caching, Video, OpenFlow, Software-Defined Networking

1. INTRODUCTION

The use of Video-on-Demand (VoD) services is increasing each year. In the UK, Internet visits to online video sites grew by 36% between September 2010 and September 2011 [8]. Current market predictions suggest that global VoD traffic will triple in volume from 2012 to 2016 [3]. VoD has become ubiquitous, with users expecting playback with minimal buffering [4], and utilising playback controls previously reserved for physical media [6]. This functionality comes at a cost: it places huge demand on the content servers to deliver this media to potentially thousands of users, sometimes simultaneously. This demand is increased by the prevalence of high-definition content, which requires a higher throughput to maintain a consistent user experience.

In order to facilitate these interactions, content producers employ Content Distribution Networks (CDNs) to meet the storage, latency and scalability requirements that must be satisfied. Yet, CDNs typically place their nodes outside of an operator's network, in which context, localised duplicate transactions can still occur. These manifest as identical requests at a later time (minutes, hours, days) by another user situated on the same network. This content, despite being an exact duplicate of what was previously delivered, still has to traverse the operator's network to the user. We aim to reduce this inefficiency.

To avoid the duplication of such traffic in a network, we present a new caching architecture. By caching the Video-on-Demand (VoD) data within the network, we can avoid this duplication of traffic, and reduce congestion on network links. This helps to satisfy throughput and latency requirements, whilst relinquishing resources for other types of traffic. This will be done transparently so that existing clients do not need updating or upgrading.

2. PLATFORM

A key design consideration in our planned platform is the ability to maintain the current delivery method of content to end-points. Content is distributed to users using unicast flows. We are preserving this delivery method to ensure that existing software and devices can be retained. To this end, the client does not need to be aware where the content is originating from. This is designed to minimise disruption to service during deployment, and significantly increases the overall chances of adoption. By leaving current delivery methods unchanged, network operators can deploy the platform incrementally. Additionally, users will not be discouraged by requiring additional monetary expenditure on new hardware.

To achieve this, we will exploit an emerging network technology: OpenFlow [9]. OpenFlow decouples control of the forwarding plane away from the switching hardware, and places it in software (such as `NOX` [7]), running on a physically different machine (as shown in Figure 1). We will exploit this separation to transparently redirect users requests for content to a local cache. This local cache will then serve the content to the user, and prevent additional traffic from traversing the network.

To facilitate this redirection, we will supplement `NOX` with the ability to communicate with a new type of controller, known as OpenCache. OpenCache will allow connections from multiple cache and switch instances. This creates a unified point of control and enables coherent collaboration of resources.

The synergies between OpenFlow and OpenCache soon become evident. Together, they facilitate the actual caching and redirection process. Furthermore, OpenFlow enables caching to be influenced by network conditions. These met-

CoNEXT Student'12, December 10, 2012, Nice, France.
Copyright 2012 ACM 978-1-4503-1779-5/12/12 ...$15.00.

rics are present on all OpenFlow switches and readable by the OpenFlow controller. Using these values, an OpenCache controller can dictate when to distribute content between caches or when to initially fetch content from a server (pre-caching). OpenFlow can also determine the traffic levels destined for a cache. Combined with hit metrics from the cache, OpenCache will be able to balance traffic accordingly.

The OpenCache platform also provides a key benefit synonymous with OpenFlow: mobility. This is the ability to place a controller in any network location desired, and even aggregate this control throughout the network using multiple controllers. Similarly, OpenCache has no requirement that both OpenFlow controllers and OpenCache controllers reside on the same machine. Each element of this architecture can be positioned in a different location on the same network. With full control of the forwarding plane, there is no longer a requirement to even have the cache placed in the network; this can be outsourced to another location and provider [5], although latency would have to be taken into consideration.

Video-on-demand platforms are moving away from the traditional Real Time Messaging Protocol (RTMP) and transitioning to HTTP chunked streaming for the delivery of content [1, 2] . Content is often delivered from CDNs by way of complex DNS resolution [1]. This process is what prevents many existing HTTP caches from functioning correctly, as CDNs do not provide a persistent link to content, nor a method to determine the quality level of video delivered from a particular URI.

This is where OpenCache differs: by negating the need for external DNS resolution, we increase simplicity and manageability in subsequent requests by responding locally. Importantly, OpenCache will also correctly store and deliver video of varying bit rates. A user should therefore receive cached content that matches his device's requested requirements and/or personal preferences.

Pertinently, MPEG-DASH [10] has been recently ratified, and widespread use of this HTTP multimedia delivery protocol seems both likely and imminent.

3. EVALUATION AND FUTURE WORK

OpenCache is still in the early-stages of development. Nonetheless, we highlight some evaluation criteria. This allows for a rapid evaluation once deployed into our existing testbed: a residential university network. The success of our platform will be measured upon the amount of cache hits we attain, and the potential bandwidth saved from such. As improving efficiency is our main goal, these metrics will clearly illustrate the level at which we have been successful.

Whilst we initially strive to deploy OpenCache as a working proof of concept, we are also looking forward to further ways to develop the caching architecture. One such avenue is to provide an interface to OpenCache, by which external parties may influence the cache content and accompanying policies. For example, a CDN may collaborate with an ISP to deliver an enhanced, feature-rich service to users. This raises additional issues, such as monetising the process and copyright concerns, which will also be considered.

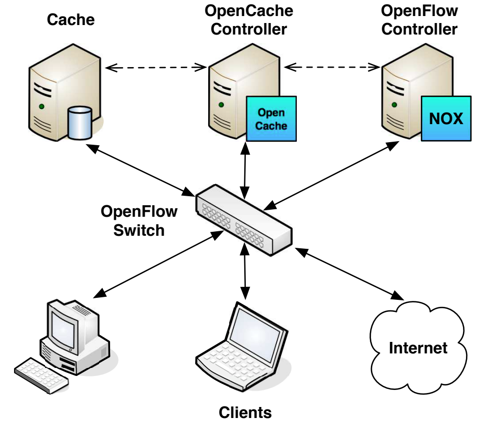

Figure 1: OpenCache Architecture

4. REFERENCES

[1] V. K. Adhikari, S. Jain, Y. Chen, and Z.-L. Zhang. How do you 'Tube'. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, SIGMETRICS '11, pages 137–138, New York, NY, USA, 2011. ACM.

[2] BBC. BBC Internet Blog: BBC iPlayer: Live Restart. `http://www.bbc.co.uk/blogs/bbcinternet/2012/06/bbc_iplayer_live_restart.html`, June 2012.

[3] Cisco Inc. Cisco Visual Networking Index: Forecast and Methodology, 2011-2016, May 2012.

[4] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *Proceedings of the ACM SIGCOMM 2011 conference*, SIGCOMM '11, pages 362–373, New York, NY, USA, 2011. ACM.

[5] G. Gibb, H. Zeng, and N. McKeown. Outsourcing network functionality. In *Proceedings of the first workshop on Hot topics in software defined networks*, HotSDN '12, pages 73–78, New York, NY, USA, 2012. ACM.

[6] V. Gopalakrishnan, R. Jana, K. K. Ramakrishnan, D. F. Swayne, and V. A. Vaishampayan. Understanding couch potatoes: measurement and modeling of interactive usage of IPTV at large scale. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 225–242, New York, NY, USA, 2011. ACM.

[7] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, 38(3):105–110, July 2008.

[8] Hitwise. Online video: bringing social media to life, October 2011.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, Mar. 2008.

[10] I. Sodagar. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia*, 18(4):62–67, Oct. 2011.