Journal of Network and Computer Applications 65 (2016) 48–71

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

The power of swarming in personal clouds under bandwidth budget

Rahma Chaabouni [a,*], Marc Sánchez-Artigas [a], Pedro García-López [a], Lluís Pàmies-Juàrez [b]

[a] Universitat Rovira i Virgili, Tarragona, Spain
[b] HGST Research, United States

ARTICLE INFO

Article history:
Received 25 May 2015
Received in revised form
27 November 2015
Accepted 7 February 2016
Available online 18 February 2016

Keywords:
BitTorrent
Peer-assisted content distribution
Personal clouds
Bandwidth allocation

ABSTRACT

Users are unceasingly relying on personal clouds (like Dropbox, Box, etc) to store, edit and retrieve their files stored in remote servers. These systems generally follow a client–server model to distribute the files to end-users. This means that they require a huge amount of bandwidth to meet the requirements of their clients. Personal clouds with limited bandwidth budget can benefit from the upload speed of the clients interested in the same content to improve the quality of service. This can be done by introducing a peer-to-peer protocol, BitTorrent for instance, when the load on a certain content becomes high. The main challenge is to decide when to switch to BitTorrent and how to allocate the cloud's available bandwidth to the different clients. In this paper, we propose an algorithm for the allocation of the cloud's bandwidth. Based on the current load and the predefined quality of service constraints, the algorithm identifies the most suitable protocol for each swarm and provides the corresponding bandwidth allocation. We validate the algorithm using a real trace of the Ubuntu One system and the results show important gains in the download times experienced by the clients.

1. Introduction

Nowadays, users are unceasingly relying on cloud storage services (like Dropbox, Google Drive or Box, etc) to store, edit and retrieve their data stored in remote servers and which can be accessed all over the Internet. Such systems are hosted by cloud-based datacenters spread all over the world and are generally equipped with a set of features that allow sharing and collaboration between the users. That is why these popular applications account for a major share of Internet traffic today (Drago, 2013).

Small and medium-sized personal clouds with limited budget constraints generally have fixed amount of bandwidth. This bandwidth is shared by all the concurrent active end-users, which might jeopardize the overall quality of service especially when the demand becomes high. As a matter of fact, these systems are based on a client–server architecture and the default content distribution protocol is usually HTTP or HTTPS. This means that all download requests are handled by a central entity which sends the requested content in a single stream. Unfortunately, such transfer is limited by the narrowest network condition along the way, or by the server being overloaded by requests from many clients.

To cope with these limitations, the cloud can benefit from the clients' upload capacities to overcome its bandwidth limits. This can be done by using the BitTorrent protocol (Cohen, 2003) to distribute the files that are shared between a set of devices. In such scenarios, it is possible to benefit from the common interest of users in the same file and use their own upload bandwidth to offload the cloud from doing all the serving. However, the use of BitTorrent may incur a longer download time compared to HTTP especially for small files (Chaabouni et al., 2014). The main challenge is to decide for each swarm which approach is more suitable for transferring the requested files (client-server or peer-assisted), and how much cloud bandwidth should be allocated to each swarm.

In this paper, we study the relationship between the cloud bandwidth allocated to a swarm of clients and the resulting download time for the end-users. We also propose a bandwidth allocation and protocol decision algorithm that evaluates for each swarm the most suitable protocol and returns the amount of bandwidth to be allocated, based on the current load on the cloud. Our key contributions are as follows:

- We analyze the relationship between the amount of cloud bandwidth allocated to a given swarm and the resulting download time. Based on a fixed quality of service constraint, we calculate the amount of seed bandwidth needed to ensure a given ratio between the download times in HTTP and BitTorrent.

* Corresponding author.
E-mail addresses: rahma.chaabouni@urv.cat (R. Chaabouni), marc.sanchez@urv.cat (M. Sánchez-Artigas), pedro.garcia@urv.cat (P. García-López), lluis.pamies-juarez@hgst.com (L. Pàmies-Juàrez).

http://dx.doi.org/10.1016/j.jnca.2016.02.006
1084-8045/

- We propose a dynamic algorithm which uses simple parameters that can be collected by the system and evaluates the efficacy of using HTTP and BitTorrent as a distribution protocol for each requested file. Based on the load on the cloud and the predefined switching constraints, the algorithms decides the most suitable protocol for each case and provides the corresponding bandwidth allocations at the swarm level. This algorithm can be applied in cloud-based content distribution systems to achieve important improvements in the overall quality of service.
- We evaluate the efficiency of our proposal using two simulators. The first simulates the default behavior of the cloud where each download operation is treated individually and the content is delivered using HTTP. The second simulator simulates the bandwidth distribution and switching algorithm where BitTorrent can be used along with HTTP to distribute content. We validate both approaches using a real trace of the Ubuntu One[1] system: we vary the switching constrains and the cloud upload speed limits and measure the degree of improvement in download time of the involved clients using our algorithm (BitTorrent and HTTP together) compared to the use of HTTP alone. The results show important improvements in the download time experienced by the peers.

The remainder of this paper is organized as follows: first, we discuss related work in Section 2 and present some background on BitTorrent and personal clouds in Section 3. Second, we present the architecture of our personal cloud system and highlights the main differences compared to the classic personal clouds in Section 4. In Section 5, we state the bandwidth allocation problem and propose an algorithm to solve it. The algorithm is evaluated later in Section 6 using a trace of a real personal cloud system. Finally, Section 7 concludes the paper.

2. Related work

Various related works focus on how to efficiently distribute content to a set of users, from classical content distribution networks (CDNs) to online, multicast streaming of live content. Peer-to-peer is considered one of the technologies that has proven its efficiency in reducing the load on the servers and improving the distribution time of the clients (Karagiannis et al., 2005). BitTorrent (Cohen, 2003) is one of the peer-to-peer protocols that has attracted the researchers' attention and several studies were dedicated to the analysis, modeling and measurements of the BitTorrent ecosystem (Qiu and Srikant, 2004; Izal et al., 2004; Pouwelse et al., 2005; Piatek et al., 2007).

BitTorrent was also introduced in cloud environments and many previous works have focused on reducing download times for large contents using BitTorrent in cloud settings. BitTorrent has proven its efficiency not only for bulk synchronous content distribution (Sweha et al., 2011), but also for reducing transfer times for cloud virtual images (Wartel et al., 2010; Schmidt et al., 2010; Reich et al., 2010). For example, in Reich et al. (2010), authors demonstrate that their BitTorrent-based solution for distributing virtual machines delivers up to an $30\times$ speedup over traditional remote file system approaches.

In this paper, we tackle the problem of the efficient distribution of the files to end-user in personal cloud systems. These systems have become very popular lately and there have been important works related to the benchmark and design of these services (Drago et al., 2012, 2013; Gracia-Tinedo et al., 2013; Garcia-Lopez et al., 2014; Tinedo et al., 2015). To the best of our knowledge, we were the first to propose to use both HTTP and BitTorrent together in cloud systems (Chaabouni et al., 2013, 2014).

In Chaabouni et al. (2013), we introduced the idea of transparent switching from HTTP to BitTorrent upon detection of a certain critical mass demand on a specific content. The threshold was placed on the number of users requesting the same files. The system tested with each new request whether the current number of requesters of the corresponding file passed the predefined threshold or not. When the threshold was reached, the system decided to adopt BitTorrent instead of HTTP in order to avoid bottlenecks on the one hand, and to save cloud bandwidth on the other hand.

In Chaabouni et al. (2014), we investigated further the threshold at which the system should switch from HTTP to BitTorrent. Instead of placing a static condition on the number of peers requesting the same file, we elaborated a complete analysis and experimental evaluation of a dynamic threshold that takes into consideration not only the number of peers, but also their corresponding bandwidths and the size of the shared file. With recourse to previous studies related to the study of HTTP and BitTorrent protocols (Qiu and Srikant, 2004; Wei et al., 2005; Kumar and Ross, 2006; Carbunaru et al., 2014), we proposed accurate estimations of the download times in both protocols and introduced some evaluation metrics to evaluate the efficiency of each of them. These metrics provide accurate estimations of the gain in download time and the amount of data contributed by the peers.

This paper differs from Chaabouni et al. (2014) in that it considers personal clouds with a fixed bandwidth budget constraint. Various related works focused on the allocation of the seed's bandwidth within the BitTorrent swarms when the bandwidth budget is limited (Leon et al., 2014; Peterson and Sirer, 2009; Sharma et al., 2014). However, we believe we are the first to propose an algorithm that manages to distribute the limited cloud bandwidth between the different sets of swarms each downloading the requested content using one of two possible download protocols: HTTP and BitTorrent.

3. Background

In this section, we start by exposing some detailing about the BitTorrent protocol. Later, we propose a definition of personal clouds and detail the general architecture of these systems.

3.1. The BitTorrent protocol

BitTorrent (Cohen, 2003) is a peer-to-peer protocol whose goal is to facilitate fast downloads of files by taking advantage of the upload bandwidth of the peers. A user who wants to share a certain content via BitTorrent has to generate first a *.torrent* file that contains the related meta-data information and a link to a *tracker*. A tracker is a server that assists in the communication between peers interested in the same content. After the generation of the meta-data file, the user has to make the content to be shared available through a BitTorrent node acting as a *seed*. A seed is a node that has a complete copy of a particular content, whereas a *leecher* is one that has only a partial

[1] https://wiki.ubuntu.com/UbuntuOne