Computer Networks 000 (2016) 1–17

Contents lists available at ScienceDirect

Computer Networks

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

Workload models and performance evaluation of cloud storage services

Glauber D. Gonçalves[a,*], Idilio Drago[b], Alex B. Vieira[c], Ana Paula Couto da Silva[a], Jussara M. Almeida[a], Marco Mellia[b]

[a] Universidade Federal de Minas Gerais, Brazil
[b] Politecnico di Torino, Italy
[c] Universidade Federal de Juiz de Fora, Brazil

ARTICLE INFO

Article history:
Received 2 October 2015
Revised 3 March 2016
Accepted 30 March 2016
Available online xxx

Keywords:
Cloud storage
Models
Measurements

ABSTRACT

Cloud storage systems are currently very popular with many companies offering services, including worldwide providers such as Dropbox, Microsoft and Google. These companies as well as providers entering the market could greatly benefit from a deep understanding of typical workload patterns their services have to face in order to develop cost-effective solutions. Yet, despite recent studies of usage and performance of these systems, the underlying processes that generate workload to the system have not been deeply studied.

This paper presents a thorough investigation of the workload generated by Dropbox customers. We propose a hierarchical model that captures user sessions, file system modifications and content sharing patterns. We parameterize our model using passive measurements gathered from fourdifferent networks. Next, we use the proposed model to drive the development of CloudGen, a new synthetic workload generator that allows the simulation of the network traffic created by cloud storage services in various realistic scenarios. We validate CloudGen by comparing synthetic traces with actual data from operational networks. We then show its applicability by investigating the impact of the continuing growth in cloud storage popularity on bandwidth consumption. Our results indicate that a hypothetical 4-fold increase in both user population and content sharing could lead to 30 times more network traffic. CloudGen is a valuable tool for administrators and developers interested in engineering and deploying cloud storage services.

1. Introduction

Cloud storage is a data-intensive Internet service that synchronizes files with the cloud and among different end user devices, such as PCs, tablets and smartphones. It offers the means for customers to easily backup data and to perform collaborative work, with files being automatically uploaded and synchronized. Cloud storage is already one of the most popular Internet services, generating a significant amount of traffic [1]. Well-established players such as Dropbox, as well as giants like Google and Microsoft, face a fierce competition for customers. Currently, Dropbox leads the market with more than 400 million users by 2015,[1] and Google and Microsoft show a fast growth [2], thanks to solutions that are more and more integrated into Windows or Android Operating Systems.

Both established and new players could greatly benefit from a deep understanding of the workload patterns that cloud storage services have to face in order to develop cost-effective solutions. However, several aspects make the analysis of cloud storage workloads a challenge. As the stored content is private and synchronization protocols are mostly proprietary, the knowledge of how these systems work is limited outside companies running the services. It is thus very hard to obtain large scale data that could drive workload analyses, given also the widespread use of encryption for both data and control messages [3].

* Corresponding author. Tel.: +553192555988.
E-mail addresses: ggoncalves@dcc.ufmg.br, gdgnew@gmail.com (G.D. Gonçalves), idilio.drago@polito.it (I. Drago), alex.borges@ufjf.edu.br (A.B. Vieira), ana.coutosilva@dcc.ufmg.br (A.P. Couto da Silva), jussara@dcc.ufmg.br (J.M. Almeida), marco.mellia@polito.it (M. Mellia).

[1] https://www.dropbox.com/news/company-info

http://dx.doi.org/10.1016/j.comnet.2016.03.024
1389-1286/© 2016 Elsevier B.V. All rights reserved.

Indeed, we are aware of only a few recent efforts to analyze the characteristics of cloud storage services [1], focusing either on architectural design aspects [4,5], quality of experience [6,7], or benchmark-driven performance studies [3,8,9]. Yet, a characterization of the underlying client processes that generate workload to the system, notably the data transfer patterns that emerge from such processes, is only partly addressed by our preliminary efforts presented in [10,11]. Such knowledge is key to analyze the impact of these services on network bandwidth requirements as well as to the design of future services.

Towards filling this gap, this paper performs a thorough investigation of the workload experienced by Dropbox, the currently most popular cloud storage service. We propose a hierarchical model that describes client behavior in successive Dropbox sessions. Within each session, our model captures file system modifications and content sharing among users and devices, as well as client interactions with Dropbox servers while storing and retrieving files. We parameterize the model based on the analysis of Dropbox traffic traces collected in four different networks, namely two university networks – one in South America and one in Europe, and two European Internet Service Provider (ISP) networks. We offer these traces to the community, to foster future studies.

Next, we use the model to drive the development of a new synthetic workload generator, called *CloudGen*. Our validation experiments, comparing real and synthetic workloads, show that CloudGen is able to reproduce the behavior of Dropbox customers. CloudGen is a valuable tool that allows network administrators and cloud storage developers to analyze the traffic volume of cloud storage as well as to evaluate future system optimization and developments in various synthetic, but still realistic, workload scenarios.

Given the steady increase in Dropbox user base [12], we illustrate the applicability of CloudGen by evaluating how network traffic would change in edge networks, such as ISP and campus networks, when customer population grows and when users share more content via cloud storage. Our results suggest that a hypothetical 4-fold increase in both user population and content sharing could lead to 30 times more cloud storage traffic, eventually challenging the capacity of such systems.

In sum, our main contributions are the following:

- We propose a hierarchical model of the Dropbox client behavior and parameterize it using passive measurements gathered from four different networks. To the best of our knowledge, we are the first to model the client behavior and data sharing in Dropbox.
- We develop and validate a synthetic workload generator (CloudGen) that reproduces user sessions, traffic volume and data sharing patterns. We offer CloudGen as free software to the community.[2]
- We illustrate the applicability of CloudGen in a case study, where the impact of larger user populations and more content sharing are explored.

The paper is organized as follows. Section 2 presents the background on Dropbox. Our data collection methodology is presented in Section 3. Section 4 describes our client behavior model and characterizes its components, whereas our workload generator is introduced in Section 5. We illustrate CloudGen applicability in Section 6, exploring possible future scenarios for cloud storage usage. Section 7 reviews related work, while Section 8 summarizes our contributions and lists future work. Finally, additional details of the characterization of our model components are provided in Appendix A.

2. Overview of Dropbox

2.1. Architecture

Dropbox relies on thousands of servers that are split into two major groups: *control* and *storage* servers. Control servers are responsible for: (i) user authentication; (ii) management of the Dropbox file system metadata; and (iii) management of device sessions. Storage servers are in charge of the actual storage of data, and are outsourced to the Amazon cloud by the time of writing.

Users can link several devices to Dropbox using both PC and mobile clients. Web interfaces are also available, providing an easy way to access files in the cloud. We will only discuss PC clients in this work, since they are responsible for most workload in cloud storage [1], although our model and workload generator can be extended to other usage scenarios as well.

Every user registered with Dropbox is assigned a unique *user ID*. Every time Dropbox is installed on a PC, it provides the PC a unique *device ID*. Users need to select an initial folder in their PCs from where files are kept synchronized with the remote Dropbox servers. Users might decide to share content with other users. This is achieved by selecting a particular folder that is shown to every participant in the sharing as a *shared folder*.

Internally, Dropbox treats the initial folders selected at configuration time and all shared folders indistinctly. They are all called *namespaces*, and each is identified by a unique *namespace ID*. Namespaces are usually synchronized with all devices of the user and, for shared folders, with all devices of all users participating in the sharing. Each namespace is also attached to a monotonic *journal ID (JID)*, representing the namespace latest version [13].

2.2. Client protocols

Our work takes as reference the protocols used by Dropbox in its client v2.8.4. A device connecting to Dropbox contacts control servers to perform user authentication and to send its list of namespaces and respective journal IDs. This step allows servers to decide whether the device is updated or if it needs to be synchronized. If outdated, the device executes storage operations (exemplified next), until it becomes synchronized with the cloud. After that, the device finishes the initial synchronization phase and enters in steady state.

When a Dropbox device identifies new local changes (e.g., new files, or file modifications), it starts a synchronization cycle. The device first performs several local operations to prepare files for synchronization. For example, Dropbox splits files in chunks of up to 4 MB and calculates content hashes for each of them. Chunks are compressed locally, and they might be grouped into bundles to improve performance [3].

As soon as chunks/bundles are ready for transmission, the device behaves as depicted in Fig. 1a. It first contacts a Dropbox control node, sending a list of new chunk hashes – see the *commit* request in the left-hand side of Fig. 1a. The servers answer with the subset of hashes that is not yet uploaded (see *need blocks (nb)* reply). Note that servers can skip requesting chunks it already has, a mechanism known as client-side deduplication [14]. If any hash is unknown to servers, the device executes several storage operations directly with the storage servers (see *store* requests).[3] Finally, the device contacts again the Dropbox control node, repeating the initial procedure to complete the transaction.

[2] CloudGen is available at http://cloudgen.sourceforge.net.

[3] Dropbox also deploys a protocol for synchronizing devices in Local Area Networks, called `LAN Sync`. Devices in a LAN can use it to exchange files without retrieving content from the cloud. Since such traffic does not travel outside the LAN, we ignore the effects of LAN Sync.