Management and Content Delivery for Smart Networks: Algorithms and Modeling

Lab. 2: Simulation of Cloud Storage Synchronization

The objective of this laboratory is to evaluate strategies for the synchronization of devices with cloud storage. You will compare current solutions to alternatives based on P2P, answering questions such as whether the workload in central storage servers can be *significantly* reduced by changing the synchronization protocols.

While doing that, you will practice different simulation techniques, e.g., traffic generation, trace-driven analysis and the estimation of model parameters from traces.

You will use datasets of real usage of cloud storage to parameterize the models.

Scenario

Consider a typical cloud storage solution – e.g., Dropbox, Microsoft OneDrive etc. Each *user* of the service registers a set of *devices*. Each device contains many *shared folders*, i.e., paths in the PC from where files are synchronized with the cloud.

All content in a shared folder is synchronized with all devices of the given user. Moreover, users might share content with others by inviting them to shared folders, which then become synchronized with all devices of all users participating in the sharing. This scheme results in a content sharing network, which is illustrated in Fig. 1.

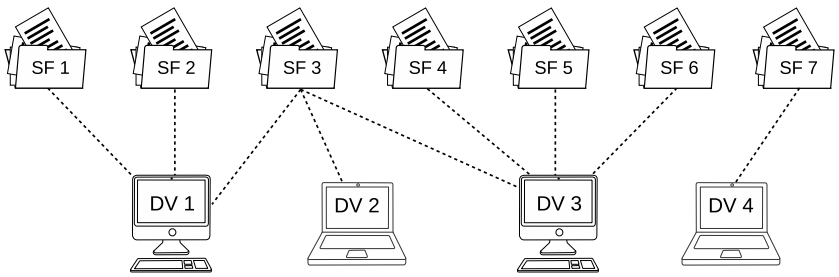

Figure 1: Content sharing network – *DV* stands for device, *SF* for shared folder.

Assume the synchronization protocol of Dropbox: The addition of any content in a shared folder triggers content propagation. All devices having Dropbox and participating in the sharing retrieve the content immediately if on-line, or as soon as they come back on-line.

Exercise 1: Simulation of Cloud Storage Workload

In this exercise, use the architecture of Dropbox and most other storage providers as reference. Synchronization takes place using a centralized infra-structure in the cloud. Devices connect to servers and send/receive files to/from the cloud.

Simulate each client as an independent process, following the model depicted in Fig. 2. The model captures three aspects of cloud storage clients: (i) devices' sessions – i.e., devices becoming on-line and off-line; (ii) the time between uploads of files to the cloud, once a device is active; (iii) the transmission of files in the network.

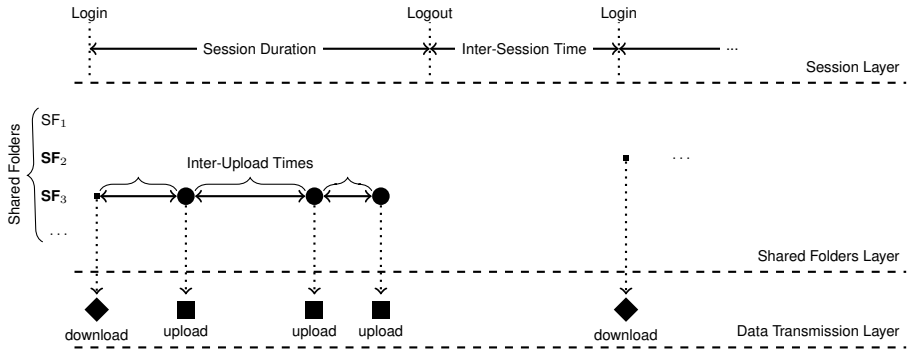

Figure 2: Model of cloud storage client behavior.

A session starts when the device logs in, and it lasts until the device logs out. This interval is called *session duration*. The time between the client logout and its next login is the *inter-session time*.

When a device becomes on-line, it first *downloads* all content produced by other peers in its shared folders. Then, it may *upload* one of more files, which is controlled by the *inter-upload time*. Assume that a device always upload content to the same shared folder in a session, which is chosen at random. Each upload contains a unique file, characterized by the *file ID* (e.g., a hash key) and the *file size*. Assume also that files, once uploaded, are never deleted from the shared folders.

As soon as a file is uploaded, other devices participating in the sharing are notified and start downloading the content immediately if on-line. Otherwise, the file is queued to be downloaded next time the peers appear on-line, as described above.

Your tasks in this exercise are:

1. Implement the simplified model for a Dropbox device depicted in Fig. 2. You are given a skeleton for the simulator in SimPy, which creates a representation for the synchronization network shown in Fig. 1, using parameters described in the next table.
2. Parameterize the upper-layers of the model with the following distributions:

Component	Model	Parameters
Session Duration (seconds)	Log-normal	$\mu = 8.492$ $\sigma = 1.545$
Inter-Session Time (seconds)	Log-normal	$\mu = 7.971$ $\sigma = 1.308$
Inter-Upload Time (seconds)	Log-normal	$\mu = 3.748$ $\sigma = 2.286$
Shared Folders per Device	Negative Binomial	$r = 0.470$ $p = 1.119$
Devices per Shared Folder	Negative Binomial	$r = 0.231$ $p = 0.537$

3. Simulate file sizes and IDs using the information about files stored by real users on UbuntuOne. In the references below you will find a link to download traces of UbuntuOne usage.
4. **Estimate how the workload on the cloud servers, in terms of active devices, upload and download traffic, varies according to the number of devices in the simulation.**

Note that, for the latter, you will need to simulate upload/download rates for each transfer, for an arbitrary time granularity. In the references below you will find links to data about typical throughout experienced by Dropbox users when uploading and downloading files while connected to ADSL lines in Italy. Assume that the server-side bandwidth capacity is infinity.

Exercise 2: Syncronization using P2P

Again, each device is an independent process and behaves as in the model described in Fig. 2 in terms of sessions and uploads. However, the download of content now may happen without the cloud – i.e., assume that devices may exchange files using a P2P protocol.

A device always sends files to the cloud first. As before, cloud servers notify other devices participating in the sharing about the new content to be downloaded. When sending these notifications, based on the file IDs, servers inform devices about *all* other devices already holding the file – i.e., even those not participating in the sharing. If a device holding the file is on-line at that moment, P2P synchronization takes place. If no other device is on-line to provide the content, or if the download fails for any reasons (e.g., the peer goes offline before the download is complete), a normal download from the cloud takes place.

Your tasks in this exercise are:

1. **Estimate how the workload on cloud servers, in terms of download traffic, is reduced by the P2P synchronization scheme.**
2. Study the traffic served via P2P: How much bandwidth do individual peers contribute to the P2P synchronization?
3. There are two major factors limiting the P2P approach: (i) files must be shared by different devices; (ii) users must be online simultaneously. Which one is predominant based on your simulations?

Groups and Final Reporting

You are expected to work on groups of up to three students. Each group is required to prepare a short report describing results obtained during all labs in the course. This report must not exceed 10 pages.

You have as starting point the skeleton of a network simulator written in Python. You are however free to pick other programming languages you are familiar with to code your solution, provided that all members of the group are able to code and explain me the solution.

You need to delivery both the written report and your source code by June 16.

References

[1] Datasets and support code are available in https://sites.google.com/site/idiliod/links/
[2] SimPy in 10 Minutes. `https://simpy.readthedocs.io/en/latest/simpy_intro/`
[3] G. Goncalves, I. Drago, A. Vieira, A. Silva, J. Almeida, and M. Mellia. Workload Models and Performance Evaluation of Cloud Storage Services. In: Computer Networks. 2016.
[4] R. Gracia-Tinedo, Y. Tian, J. Sampe, H. Harkous, J. Lenton, P. Garca-Lopez, M. Sanchez-Artigas, and M. Vukolic. Dissecting UbuntuOne: Autopsy of a Global-Scale Personal Cloud Back-end. In Proc. of the ACM IMC, 2015.