KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Computer Science and Engineering

Second cycle, 120 credits

Distributed file system by advantaging online web services

An image says more than 1000 words

GLENN OLSSON

San Luis Obispo, California, USA, 2022

Distributed file system by advantaging online web services

An image says more than 1000 words

GLENN OLSSON

Master's Programme, Computer Science, 120 credits
Date: February 7, 2022

Supervisors: Hamid Ghasemirahni, Zachory Peterson
Examiner: Gerald Quentin Maguire Jr

Host company: Cal Poly
Swedish title: Distrubuerat filsystem genom utnyttjande av onlinebaserade webtjänster
Swedish subtitle: En bild säger mer än 1000 ord

Abstract

Today there are free online services that can be used to store files of arbitrary types and sizes, such as Google Drive. These services are often limited by a certain total storage size. My goal is to create a filesystem that similarly can store arbitrary amount and types of data but without any real limit. This is to be achieved by taking advantage of online webpages such as Twitter where text and files can be posted on free accounts with no visible limit. The goal is to have a filesystem that behaves like any other but where the actual data is stored for free on unsuspecting websites.

Keywords

Canvas Learning Management System, Docker containers, Performance tuning

Choose the most specific keyword from those used in your domain, see for example: the ACM Computing Classification System (https://www.acm.org/publications/computing-classification-system/how-to-use), the IEEE Taxonomy (https://www.ieee.org/publications/services/thesaurus-thank-you.html), PhySH (Physics Subject Headings) (https://physh.aps.org/), ...or keyword selection tools such as the National Library of Medicine's Medical Subject Headings (MeSH) (https://www.nlm.nih.gov/mesh/authors.html) or Google's Keyword Tool (https://keywordtool.io/)

Mechanics:

- The first letter of a keyword should be set with a capital letter and proper names should be capitalized as usual.
- Spell out acronyms and abbreviations.
- Avoid "stop words" - as they generally carry little or no information.
- List your keywords separated by commas (",").

Since you should have both English and Swedish keywords - you might think of ordering them in corresponding order (*i.e.*, so that the n^{th} word in each list correspond) - this makes it easier to mechanically find matching keywords.

Sammanfattning

Sammanfattning på svenska

Nyckelord

Canvas Lärplattform, Dockerbehållare, Prestandajustering

Acknowledgments

Thanks to:

- Andrew Guenther, for uploading this template
- I would like to thank xxxx for having yyyy.

San Luis Obispo, February 2022
Glenn Olsson

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Project Overview

This project intends to create a filesystem called *Fejk File System* (FFS) which takes advantage of online web services such as Twitter. The idea is to save the files by posting/sending an encrypted version as one or more posts/private messages on these services. The goal is to achieve storage of data in the same scale as the free accounts on online storage services such as Google Drive where users can store up to 15Gb of files. Accomplishing this would mean that one can store more data than that for free using this new filesystem. The data posted will be encrypted and not be comprehensible by anyone who would stumble upon a post, just like if anyone would analyze a regular encrypted filesystem on a disk.

The intention is not to create a revolutionary fast and usable filesystem but to instead to explore how well it is possible to utilizing the storage that Twitter and similar services provides by allowing users to post text and files, almost unmonitored. The performance will however be analyzed and compared to existing alternatives such as Google Drive.

1.2 Background

Year after year, people increase their total data storage usage for obvious reasons. Cameras get better leading to images and videos taking more space, and with storage being cheap and easily usable, files are not needed to be deleted meaning that the data usage accumulates (**CITATION NEEDED?**). This means that users will require more and more storage throughout their lifetime, and even potentially beyond their lifetime if dependants wants to keep

these files. System storage in our hardware devices often increase with new product cycles. Today you can keep hundreds of gigabytes in your pocket without spending a big fortune(**COMPARE iPHONE FROM LIKE 10 YEARS AGO AND TODAY - STORAGE AVAILIABLE, INCREASED. LOWEST TIER VS HIGHEST TIER**). Along with increasing device storage is cloud storage increasing. For instance Apple's service iCloud allows users to store up to 2TB of data in the cloud for a few bucks per month (**CITE??**). Even though the cost per month is not a lot, after many months this cost accumulates and you as a user get more and more dependant on this storage, especially as you don't want to spend time to look through all your data and maybe remove some to save space (**FIND SOMETHING THAT USERS NEVER DOWNGRADES THEIR STORAGE - MUST EXIST**). With increased pricing or necessary space upgrade, the cost will be even higher.

Social media platforms such as Twitter, Flickr and Facebook have many millions of daily users that post anything from texts to images for their cats or funny videos. According to Henna Kermani at Twitter, they processed about 200GB of image data every second in 2016[1]. A single user posting a few images per day does not significantly change the amount of data processed or saved at all for these tech giants - a few gigabytes here or there will probably go unnoticed (**PROBABLY?? IS THAT GOOD ENOUGH FOR A THESIS?**). (**MENTION HERE ABOUT POTENTIAL ANOMALY DETECTION?? OR LATER?+**). The difference between the photos posted on Twitter compared to the ones stored on cloud services such as iCloud is that the images on Twitter are stored for free for the users, indefinitely. While iCloud and similar services often have a free-tier of storage, Twitter does not have an upper limit of how many images or tweets one can make (**RIGHT?? I COULD NOT FIND ANYTHING WITH A QUICK GOOGLE SEARCH. LOOK AT TOS?**)

Chapter 2

Background

2.1 Filesystems

Filesystems are used to store data on for instance a hard drive of a computer on in the cloud. Google Drive is a filesystem that enables user to save their data online up to 15 GB for free[2] using their clusters of distributed storage devices, meaning that the data is saved on theirs servers which can be located wherever[3]. Paying customers can achieve higher amount of storage using the service.

A deniable filesystem is a system that does not expose files stored on this system without credentials - neither how many files are stored, their sizes, their content or even if there exists any files on the filesystem[4]. This is useful if for example one is to be exposed to an audit of their data by a totalitarian regim where they don't even want to disclose that they have data.

A unix filesystem uses a data structure called an *inode*. An inode keeps track of the metadata for the files in the filesystem, and a directory simply contains the file names, and each files/directory's inode id. Using a lookup, the system can then learn about the file - where it is located, for instance how big it is, as can be seen in Figure 2.1 (**CITATION NEEDED**). Each inode entry can contain any number of metadata information which might be relevant for the system, such as creation time and last updated.

Looking at the 4 main file systems of windows, they all have many, sometimes different, functionalities such as links and named streams as well limitations such as a defined theoretical maximum file size[5]. This is set to 16 exbibytes for NTFS, exFAT and UDF, and for FAT32 it is set to 4 gigabytes.

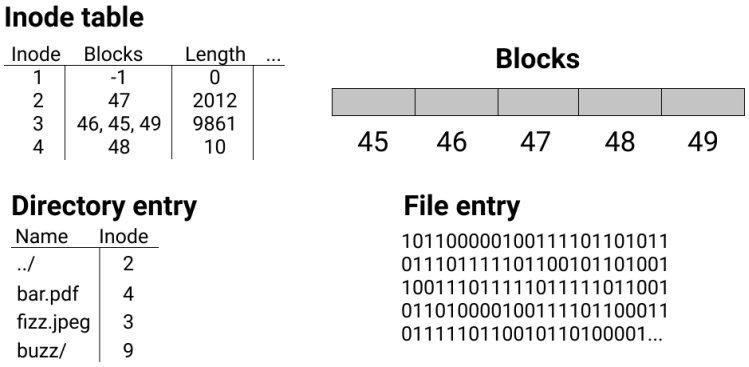

Figure 2.1: Basic structure of inode based filesystem

2.2 Threats

To consider a filesystem secure it is important to imagine different potential adversaries who might attack the system and. Considering that FFS has no real control of the data stored on the different services, all the data must be considered to be stored in an on an unsecure system. Even if we could hide the posts made on for instance twitter by making the profile private, we must still consider that twitter could be an adversary and therefore the data stored must always be unreadable without the correct authentication. We assume that any adversary has access to all knowledge about FFS, including how the data is converted, encrypted and posted. There are multiple secure ways of encrypting data, including AES which is one of the faster and secure encryption algorithms[6].

Other than adversaries for just FFS, we might also imagine that the underlying services might receive attacks that can potentially harm the security of the system or even have it go offline indefinitely. One solution is to use redundancy - by duplicating the data over multiple services we can more confidently believe that our data will be accessible as all services will probably not go offline.

2.3 Twitter

Twitter is a micro-blog online where users can sign up for a free account and create public posts using text, images and videos. Text posts are limited to 280 characters while images can be up to 5mb and videos up to 512mb[7]. There is also possibility to send private messages to other accounts, where

each message can contain up to 10’000 characters and the same limitations on files. If one would represent an arbitrary file of X bytes, each byte (0x00 - 0xFF) can be represented as a character and we can therefore represent this file as X different characters. Using the same set of characters for encoding and decoding we can get a symmetric relation for representing a file as a string of characters. This text can theoretically be posted on for instance Twitter, as long as the size is smaller than 280 or 10’000 bytes depending on if we would post a public post or a private message.

2.4 Related Work

Peters created a deniable filesystem using a log-based structure in 2014[4]. The filesystem of my project could be seen as a deniable system in the sense that the data is not actually stored on the device, and if the filesystem is not mounted it could be hard to prove that the user actually has data, even if they for instance would find the twitter account. This was also developed using FUSE[8] which I also will be using.

Zadok, Badulescu, and Shender created Cryptfs, a stackable Vnode filesystem that encrypted the underlying, potentially unencrypted, filesystem[9]. By making the filesystem stackable, any layer can be added on top of any other, and the abstraction occurs by each Vnode layer communicating with the one beneath. There’s potential to further stacking though continuing layers by using tools such as FiST[10].

Chapter 3

Method

3.1 FFS

The product of this thesis is the Fejk File System (FFS) which uses online services to store the data but behaves like a mountable disk for the users. The file system will however be very basic and not support all functionalities that other systems do such as links. The reasoning is that these behaviours are not required for a useable system, and when comparing the system to distributed filesystems such as Google Drive, they often do not support his either.

Figure 3.1 describes the basic outline of FFS which is based on the idea of inode filsystems. Instead of an inode pointing to specific blocks in a disk, the inodes of FFS will instead point keep track of the id numbers of the posts to online services where the file is located.

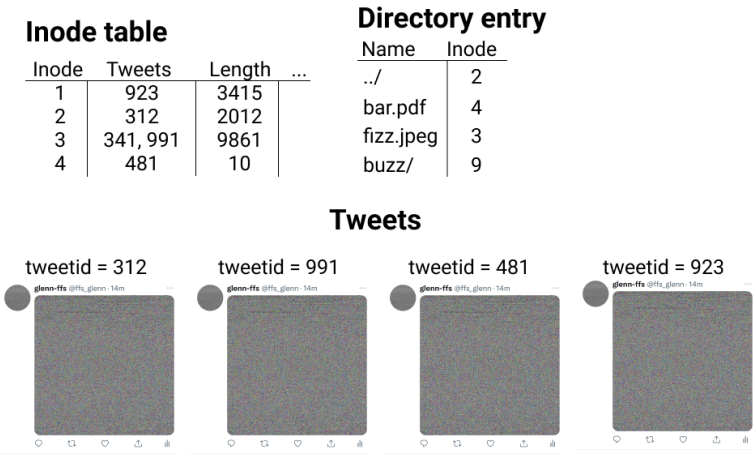

Figure 3.1: Basic structure of FFS inode-based structure

Chapter 4

Results and Analysis

Chapter 5

Discussion

Chapter 6

Conclusions and Future work

6.1 Future work

References

[1] *Mobile @Scale London Recap - Engineering at Meta*. URL: https://engineering.fb.com/2016/03/29/android/mobile-scale-london-recap/.

[2] *Cloud Storage for Work and Home – Google Drive*. URL: https://www.google.com/intl/sv/drive/ (visited on 10/26/2021).

[3] *Distributed Storage: What's Inside Amazon S3?* Cloudian. URL: https://cloudian.com/guides/data-backup/distributed-storage/ (visited on 10/26/2021).

[4] Timothy M Peters. "DEFY: A Deniable File System for Flash Memory". San Luis Obispo, California: California Polytechnic State University, June 1, 2014. DOI: 10.15368/theses.2014.76. URL: http://digitalcommons.calpoly.edu/theses/1230 (visited on 10/19/2021).

[5] mikben. *File System Functionality Comparison - Win32 Apps*. URL: https://docs.microsoft.com/en-us/windows/win32/fileio/filesystem-functionality-comparison (visited on 02/07/2022).

[6] Dr Prerna Mahajan and Abhishek Sachdeva. "A Study of Encryption Algorithms AES, DES and RSA for Security". In: *Global Journal of Computer Science and Technology* (Dec. 7, 2013). ISSN: 0975-4172. URL: https://computerresearch.org/index.php/computer/article/view/272 (visited on 02/07/2022).

[7] *Media Best Practices - Twitter*. URL: https://developer.twitter.com/en/docs/twitter-api/v1/media/upload-media/uploading-media/media-best-practices (visited on 10/26/2021).

[8] *Libfuse*. libfuse, Oct. 26, 2021. URL: https://github.com/libfuse/libfuse (visited on 10/26/2021).

[9] Erez Zadok, Ion Badulescu, and Alex Shender. “Cryptfs: A Stackable Vnode Level Encryption File System”. In: (), p. 14.

[10] *FiST: Stackable File System Language and Templates*. URL: https://www.filesystems.org/ (visited on 02/02/2022).

TRITA-EECS-EX- 2022:00

For DIVA

{
"Author1": { "Last name": "Olsson",
"First name": "Glenn",
"Local User Id": "u18orpa8",
"E-mail": "glennol@kth.se",
"organisation": {"L1": "School of Electrical Engineering and Computer Science",
}
},
"Degree1": {"Educational program": "Master's Programme, Computer Science, 120 credits"
,"programcode": "TCSCM"
,"Degree": "Degree of Master (120 credits)"
,"subjectArea": "Computer Science and Engineering"
},
"Title": {
"Main title": "Distributed file system by advantaging online web services",
"Subtitle": "An image says more than 1000 words",
"Language": "eng" },
"Alternative title": {
"Main title": "Distrubuerat filsystem genom utnyttjande av onlinebaserade webtjänster",
"Subtitle": "En bild säger mer än 1000 ord",
"Language": "swe"
},
"Supervisor1": { "Last name": "Ghasemirahni",
"First name": "Hamid",
"Local User Id": "u1fz5jtv",
"E-mail": "hamidgr@kth.se",
"organisation": {"L1": "",
"L2": "Computer Science" }
},
"Supervisor2": { "Last name": "Peterson",
"First name": "Zachory",
"E-mail": "znjpeterson@gmail.com",
"Other organisation": "Cal Poly"
},
"Examiner1": { "Last name": "Maguire Jr",
"First name": "Gerald Quentin",
"Local User Id": "u1d13i2c",
"E-mail": "maguire@kth.se",
"organisation": {"L1": "",
"L2": "Computer Science" }
},
"Cooperation": { "Partner_name": "Cal Poly"},
"National Subject Categories": "10201",
"Other information": {"Year": "2022", "Number of pages": "xv,17"},
"Series": { "Title of series": "TRITA-EECS-EX" , "No. in series": "2022:00" },
"Opponents": { "Name": "A. B. Normal & A. X. E. Normalè"},
"Presentation": { "Date": "2022-03-15 13:00"
,"Language":"eng"
,"Room": "via Zoom https://kth-se.zoom.us/j/ddddddddddd"
,Ädress": Ïsafjordsgatan 22 (Kistagången 16)"
,"City": "Stockholm" },
"Number of lang instances": "2",
"Abstract[eng]": €€€€

Today there are free online services that can be used to store files of arbitrary types and sizes, such as Google Drive. These services are often limited by a certain total storage size. My goal is to create a filesystem that similarly can store arbitrary amount and types of data but without any real limit. This is to be achieved by taking advantage of online webpages such as Twitter where text and files can be posted on free accounts with no visible limit. The goal is to have a filesystem that behaves like any other but where the actual data is stored for free on unsuspecting websites. €€€€,

"Keywords[eng]": €€€€

Canvas Learning Management System, Docker containers, Performance tuning €€€€,

"Abstract[swe]": €€€€

Sammanfattning på svenska €€€€,

"Keywords[swe]": €€€€

Canvas Lärplattform, Dockerbehållare, Prestandajustering €€€€,

}