Bern University of Applied Sciences

Informatics Seminar

InterPlanetary File System
IPFS

Author:
Martin Schmidli

Teacher:
Kai Brünnler

Bern, April 23, 2017

Contents

Chapter 1

Abstract

IPFS (InterPlanetary Filesystem) is an opensource protocol which can be used to run a distributed filesystem. IPFS was invented to tackle the drawbacks of the Internet and the Internetprotocol suite we are using today.

To understand why IPFS was invented, we first have to understand the issues we have today. This report will outline how IPFS works, what kind of issues it tries to resolv and analyze the obstacles which have to be overcome to establish IPFS as a accpted and widely used technology.

Chapter 2

Introduction

IPFS stands for InterPlanetary Filesystem. It's an opensource Internet Protocol which can be used to run a distributed filesystem. The developers didn't invent IPFS from scratch. In its core IPFS takes advantage of existing technologies like Kademlia DHT, BitTorrent and Git. The main intention behind IPFS is to tackle the issues we are having in todays Internet/Web. In the eyes of the inventors the protocol should be seen as an upgrade or even as a replacemnet of the existing technolgies like HTTP. Compared to HTTP where data is adressed by URL's is IPFS the data is adressed by its contents hash. Global data distribution should be simplified and be implemented in the protocol itself rather than prescribe a separate distribution mechnism [1].

2.1 Origin

The Development of IPFS was started in 2014 by Juan Benet, former Stanford Student and founder of the company Protocol Labs . Protocol Labs and contributors of the Community are developing IPFS further.

2.2 Name

The name was choosen as a tribute to J. C. R. Licklider, a computer scientist who came up with an idea of a "intergalactic network" of computers in 1962. He imagined a global network of computers, able to talk to each other and exchange data. During his time working at DARPA (Defense Advanced Research Projects Agency) he influenced many peoble with his ideas. DARPA later startet the ARPANET Project and laid the foundation stone for todays Internet. Many important technologies for example: TCP/IP where invented or funded during this project.

2.3 Project state

As of 19.04.2017 the Specifications of the IPFS Protocol are still being developed and hasn't completed yet. The developers state, that the core parts of the specs have reached a reliable or stable state. No RFC request have been submitet. An Implmenetation of the Protocol, written in the programming language Go and some utilities have aready been published. Implementation in other programming languages Javascript and Phyton are in developing.

Chapter 3

Todays Problems

The developers of IPFS state: The Internet/Web of today has many issues. The existing protocols we have today, have some major design issues or are not good enouth anymore to satisfy the needs of todays Web and his Users [2].

In my opinion the following list is just a small subset of issues we are faching today. It would be folish to believe that just one new protocol could solve all the existing problems.

3.1 Offline functionality

Imagine you are sitting with your colleagues at work. You all work together on a document. You are using a WebApplication to collaborate with each other. Suddenly the internet connection is lost. You are all sitting in the same network but you are unable to share your version of the document with the others. All the data needs to be synced with the backbone service for example Google Docs. and then down again to the other clients / your colleagues. The Applications should be able to talk to each other and shouldn't be dependant of a service somewhere in the internet.

3.2 Protocol

The most used Protocol from the Internet Protocol Suite by far is HTTP. It follows the Server-Client communication model. The Client establishes a connection to one server, sends requests and get responses back from the server. Even if there are load balancing mechanisms in place it will be served by the by one host only.

The alternative communication model would be Peer to Peer short. p2p. The Clients establishes a connection to mutlible "peers" which will serve the requested data. Every peer will deliver a fraction of the data. Those data "pieces" will be downloaded simultainously.

3.3 Permenancy

I guess every person which used an Internet Browser before has seen a message like ”Error 404” or ”Site not found”.

Figure 3.1: Github 404 Error Page

The specific content you tried to reach has been deletet or has been moved by the sites provider. All the links provided are now useless. Every day content gets moved around or is deleted forever. Sometimes its just a silly picture of a cat but mostly its data somebody actually would have needed.

Nevertheless there are several companies already trying to solve this issue by archiving all the data available on the internet. The most well known would be ”Internet Archive” with its ”Wayback Machine” Webapplikation. They are using web crawlers to get the data and store them. The User is able to use theWayback Machine to view the webpage back then when the Internet Archive made a snapshot of it [3]. By October 2016 they managed to archive 273 billion webpages from over 361 million websites, which resulted in a data size of 15 Petabytes (15’000 Terabytes) [4].

3.4 Centralization & Proxy

Big data hosting companies like Google or Amazon own datacenters worldwide. Developer can buy the service from them to cluster their application data and distrbute it all over the world. This ensures the availability of your Application even in case of a datacenter outage caused by a natural catastrophe or other technical issues. Those incidents aren't fictions and cost millions if they happen. Latest example, 28.02.2107 the Amazon S3 System located in Northern Virginia gots unavailable. Thousands of webpages and webapplications stopped working. A Amazon technican wantet to remove a small subset of servers. He entered the command wrong and more servers than planned were removed. This led to an 4h outage of the hole Site [5]. Estimated cost 150 Million US Dollar [6].

Clustering of data is not standard today. To make your data highly available is cost intensive. To achieve the same level of high availability cost effective without any kind of big service provider like Google is near impossible.

A second group of big data handlers are the so called CDNs. These Content Delivery Networks host data worldwide and act as a proxy for data requests. Services of these providers are used to bring content closer to the customers to reduce latency and download time.

3.5 Security

Man in the middle, armor the content not the tunnel

3.6 Privacy & Consumer Protection

In these days one of the main concerns of social media platforms are privacy issues and hate messages. Those companies, for example Facebook and Twitter, are facing huge fines in several countries because goverments think they don't do enouth to protect their users. [7]. Content shared with a service in the internet, mostly stayes in the internet as its rapidly shared and distributed. If it's out there it's out there forerver. The content itself is hard to locate and remove. The data is adressed with links. The links or URL changes from hoster to hoster. If the affacted user tries to notify the hoster of the data, for example a picture with the intetion to damage the users reputation, it might alread be hosted by several other platforms.

Chapter 4

Data Handling

4.1 Data Adressing

4.1.1 HTTP

Data which is access by HTTP is adressd with its hostname, port, path and the filename or searchword [8].

Example 4.1.1.1. URL schema used for HTTP

http : // hostname [: port] / path [? searchwords]
https://raw.githubusercontent.com/github/gitignore/master/TeX.gitignore

If the location of the file changes and no redirection was created, the link gets useless.

4.1.2 IPFS

IPFS goes a different way. Data added to the global IPFS can be adressed by the hash of its content. When the content of the file changes so does the hash value. The changed content will be added to the Network. The older content will still be accessible with the old hash value. This garanties the persistance and the versioning of the data.

Example 4.1.2.1. Add a file

We create a demo file

We add the file foo.txt to the IPFS Network. The content gets hashed and creates an key to this specific data. **QmYNmQKp6SuaVrpgWRsPTgCQCnpxUYGq76YEKBXuj2N4H6** is the key to our file with the content "foo".

```
[N-MACBOOK-3333:Desktop martinschmidli$ ipfs add foo.txt
added QmYNmQKp6SuaVrpgWRsPTgCQCnpxUYGq76YEKBXuj2N4H6 foo.txt
```

This hash value cann now be used to access the content from any IPFS client. Example with cat

```
N-MACBOOK-3333:Desktop martinschmidli$ ipfs cat QmYNmQKp6SuaVrpgWRsPTgCQCnpxUYGq76YEKBXuj2N4H6
foo
```

Let's change the content of "foo.txt".

```
N-MACBOOK-3333:Desktop martinschmidli$ echo "new Foo" >> foo.txt
N-MACBOOK-3333:Desktop martinschmidli$ cat foo.txt
foo
new Foo
```

Add the file again. Because we changed the content, the hash value will be different this time. New Key: **QmbYzb3nScopAnfkoUpRWUFVv856uSWpSRc2KYM1FSBJxr**

```
N-MACBOOK-3333:Desktop martinschmidli$ ipfs add foo.txt
added QmbYzb3nScopAnfkoUpRWUFVv856uSWpSRc2KYM1FSBJxr foo.txt
```

The system is activly deduplicating data. If existing contents gets added for example two times the same picture they will get the same hash value, but the raw data wil be stored just once in the IPFS network.

Example 4.1.2.2. Deduplicating

We add two files, same content but different names **foo2.txt** and **foo3.txt** with the content **"new Foo"**

```
N-MACBOOK-3333:Desktop martinschmidli$ echo "new Foo" >> foo2.txt
N-MACBOOK-3333:Desktop martinschmidli$ echo "new Foo" >> foo3.txt
N-MACBOOK-3333:Desktop martinschmidli$ ipfs add foo2.txt
added QmYxEG2M5LhqPwoRXhvhTWPVKtapZNXMHtL1TgPypgqNDK foo2.txt
N-MACBOOK-3333:Desktop martinschmidli$ ipfs add foo3.txt
added QmYxEG2M5LhqPwoRXhvhTWPVKtapZNXMHtL1TgPypgqNDK foo3.txt
```

Both times the generated hash is
QmYxEG2M5LhqPwoRXhvhTWPVKtapZNXMHtL1TgPypgqNDK

4.2 Mutability - IPNS

4.3 Cryptography

4.4 Versioning

4.5 Storage

4.6 Naming System

Chapter 5

Who profits

Chapter 6

Future

The Question is: ”Is there a future for IPFS”?

Bibliography

[1] https://github.com/ipfs/ipfs, 24.07.2017

[2] https://ipfs.io/, Sector: The web of tomorrow needs IPFS today, 22.04.2017

[3] https://en.wikipedia.org/wiki/Internet_Archive, 22.04.2017

[4] https://blog.archive.org/2016/10/23/defining-web-pages-web-sites-and-web-captures, 22.04.2017

[5] https://aws.amazon.com/message/41926/, 23.04.2017

[6] http://www.npr.org/sections/thetwo-way/2017/03/03/518322734/amazon-and-the-150-million-typo, 23.04.2017

[7] https://www.nytimes.com/2017/03/14/technology/germany-hate-speech-facebook-tech.html?_r=0, 23.04.2017

[8] https://www.w3.org/Addressing/HTTPAddressing.html, 23.04.2017

List of Figures

Chapter 7

RESEARCH

Research: Merkel DAG, Bittorrent Swarm, Distributed Hash Table, Bittorrent MainlineDHT, multihash, Bitswap