Tientso Ning

Seminar Report

27 Apr 2020

The presentation was regarding the protocol known as IPFS (Interplanetary file system) which is a protocol whose aim is to facilitate the transfer of large datasets (using ideas from git and bittorrent). The idea is that data blocks can be uploaded, and connected peers can communicate that data rather than a centralized distributed server (like the HTTPS protocol). There are ideas of copies and version control, built into the idea of CIDs, allowing page updates, hashes, and other techniques to facilitate transfer of data. The content is worldwide, and the protocol structure makes censorship impossible. There is no central point of failure, and the structure allows (theoretically) faster transfer, since more local peers can be responsible for distribution of data. This has built in security against DDOS, as well as providing some benefits such as data integrity checking at the content level rather than just the URL level. There is a sort of chain of blocks that allows content to be viewed in all its forms/versions (much like git).

I had a question regarding the sharing of data that is private or illegal, and how that concern of privacy is addressed under the protocol. However, it is pointed out that this data is not protected, and once upload occurs, the data is always uploaded. Additionally, applying encryption on blocks removes any accountability on what data is being uploaded. This is a huge privacy concern for me, and gives me a lot of pause regarding this protocol, although I understand that some of the same concerns exist today regarding HTTPS. I would say that it is less a concern that a dire situation could occur, but rather that this privacy question is not one that seems to be a high priority of concern for a protocol that is very adept at enabling dire situations of distribution of sensitive data.

However, beyond this, the protocol introduction is theoretically very interesting.

Notes

Interplanetary file system – goal is to transfer large datasets (some solution between git and bittorrent)

kinda like git when you put something up, it connects and communicates to all the peers, and puts it up

then the peers will be able to access

copies and version control

concept of pinning

supports links, but operates off of ipfs CID, and if the page changes, the hash will change and the new hash will be able to be updated

a protocol, content addressed file system, coordinating content delivery, combining concepts from fair use of peer to peer, torrent peer2peer, and version control of git.

content is worldwide, no central point of failure, peer2 peer

censorship is not an option

uses bittorrent-like bandwidth distribution

protocol reduces load on the server **question: what about security concerns? create a peer that is always the closest, and puts up incorrect/altered information

and prevents censorship

prevents ddos because no central server

no more missing webpages, old content is now accessible **question: how do we combat content that is not supposed to be uploaded (personal content, illegal content, etc)

we need to solve – immutability (integrity check via hashing, URLs today are stable, but not the content. IPFS changes this)

caching and de-duping prevents duplications

to store files, uses Merkle DAGs, (directed acyclic graphs)

we need to consider block sizes (too big = DOS via big blocks, no single blocks from multiple peers because then we won't know who sent the bad block)

works for streaming because root block points to first block

bitswap to get data from connected peers (replies are with recipe on how to calculate CID, not the actual CID, to make sure each is checked, and also a bit smaller)

keeping blocks locally is called pinning

encryption of data and shared CID means that any data can be communicated, unrestricted, and there is no accountability