- 跳轉到博客目錄頁面<---[在線閱讀]　　[本地] --->*點擊此查看html網頁格式　pdf格式*

特別推薦文章

- **鄧紫棋解解的小說——2507抬上帝入天坑** <---[點擊此前往github在線閱讀]　本地模式 ---> html網頁版　pdf版本
- 無眠月照無情門 . 失去自由的歌手 <---[點擊此前往github在線閱讀]　　本地模式---> html網頁版　pdf版本

西子

世人皆唱東坡詞，無人知我歌中淚。
胭脂淚痕君不見，肚藏淚酒君不知。
法律珠鍊人中鳳，舞臺深處天牢夢。
鍍金屠門千豬過，三寸魷魚萬人舔。
君入西子渡津口，她閥君腦千秋雪。

real_web3.0_and_IPFS

如何用家庭IOT做分佈式雲存儲和雲計算的終端，用閒置的電腦鑽錢

阿柄

```
An real web3.0 ecosystem should be like this, where web3.0 is not
dependent on anyCryptocurrency platorm, where self-media platform
and home IOT system is decentralized and connected to each other,
where ereryone can rent their free disk space, and where everyone
can rent their computer computation acting as an edge side of
distributed cloud computing,thus thousands and hundreds of ants can
move an elephant. In one world a real web3.0 is fully decentralized
and a lot of opportunities are distributed among individuals.
click below link to read more ...
一個真的web3.0生態系統應如是：不依賴任何加密平臺，所以不會淪爲別人的韭菜，其中去中心
化家庭IOT自媒體平臺之間自由互連，每個人將可以成爲分佈式雲計算的運算終端出租電腦存儲空間
```

An easy way to understand web3.0 and IPFS

伍成和 wu chenghe

chenghewu@gmail.com

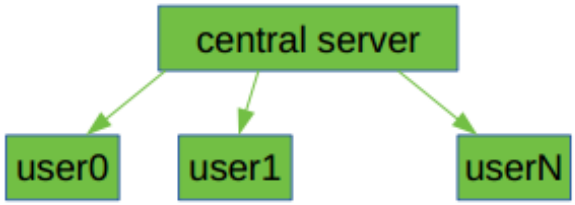
wuchenghe@vk.com

www.decensormedia.org

www.decensormedia.org/ppt

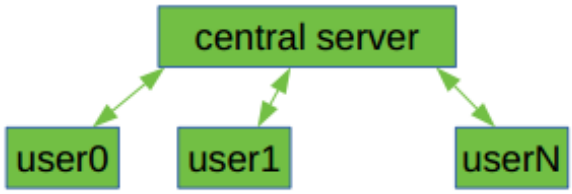
we can intuitively consider the evolution of web as following:

1. web 1.0 . read only, user can only view the content of a website, can not interact with the website



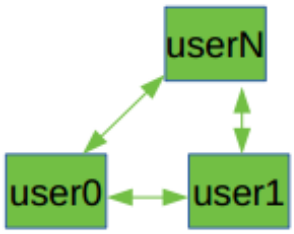
Web 1.0

2. web 2.0. read and write. The web we use today, user can interact with the server, and our current social media is based on this. But the drawback is that it is centralized system, all data are controlled by the central server. We sacrifice our privacy for convenience and laziness ! And users have to go thru central server to connect to any other users on the internet, so that censorship is a problem for such network.



Web 2.0

3. web 3.0, mainly focus on decentralization and privacy! So there is no central server is required for connecting users in the network.



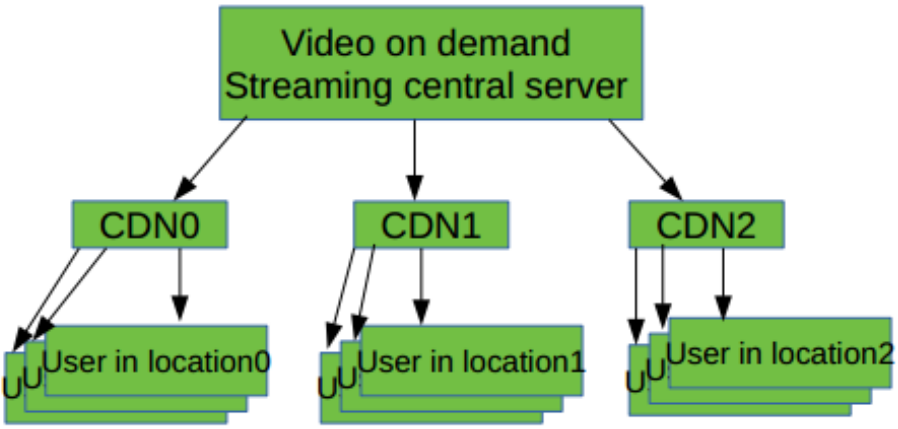
Web 3.0

IPFS (InterPlanetary File System)

Here I am not talking about exact the same IPFS as explained here in this paper : IPFS - Content Addressed, Versioned, P2P File System , I explain the basic principle and design a new IPFS in a way I believe could be better for application. After you know how to look at IPFS in a different aspect, you would better understand the original paper.

Let's start with a question : why we need IPFS ? We now enter an era where we need to store and move around petabyte data, and ideally we want them to be accessible anywhere in realtime. The typical application scenario is ultra-high definition video on demand, we have to store the huge amount of videos on the cloud, and need a lot of CDN (Content Delivery Network) to deliver the video to any corner of the world.

For people who are not familiar with video streaming system, we can intuitively think of it as depicted in the following picture :



Web 2.0 VOD streaming

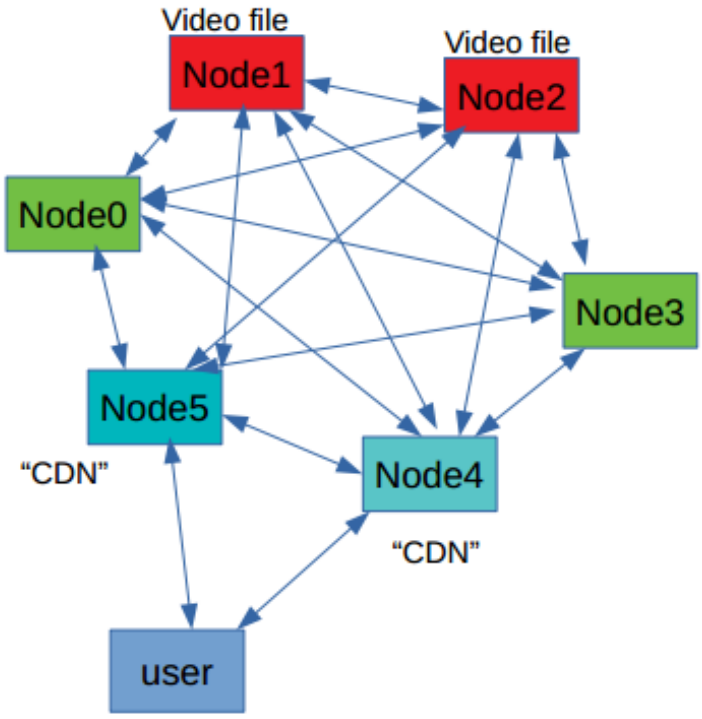
the above picture is the VOD(video on demand) streaming architecture based on web 2.0, we need a central cloud server (VPS, Virtual Private Server , simply think of cloud server as a powerful computer without display in a datacenter, like google, amazon,etc...) to store all the videos we have, but it for distance reasons, it cannot serve people all over the world, because of the bandwidth and latency issues. So we have to first move the corresponding videos to the VPS (named CDN server) which is located as near to the user as possible. So that when user watch videos, it can stream in realtime.

The problem of current VOD architecture is that the rent of the CDN is quite expensive, we does not have users watching the videos all the time thru out the year, so resource and money are wasted.

How should we solve this problem ?

We already have global distributive file system, for instance, BitTorrent. To cost down the expense, we can make use of the millions of individual computers.

So similar to the idea of home-sharing where people make use of his spared rooms, in web 3.0 every node in the IPFS network is acting like a room provider!



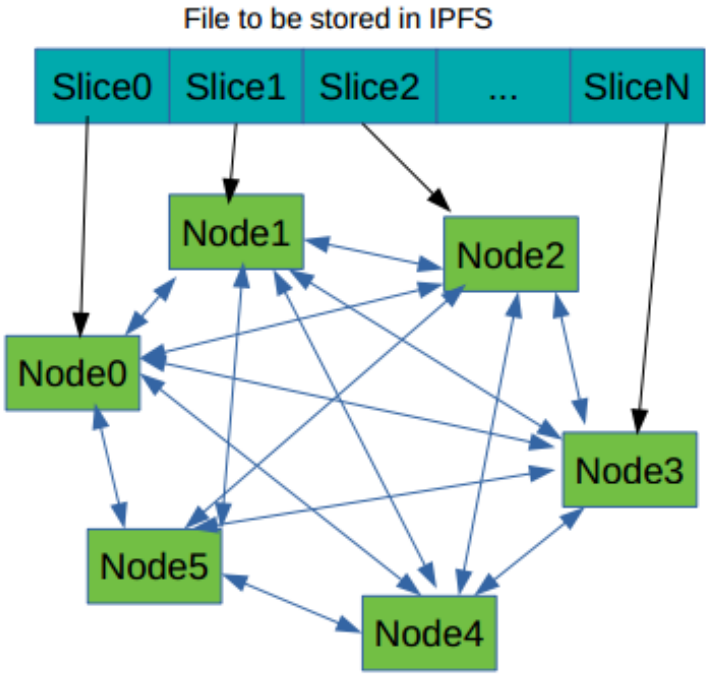
IPFS network

Every node can be both node and at the same time user, any computer install the software like BitTorrent download app, he is connected to the IPFS network.

The difference between BitTorrent is that we we don't save things for free for others, BitTorrent is a free sharing network, which is essentially violating the creator's IP, in IPFS we should like home-sharing, we rent our space in a very cheap price during a period, consider this as a micro-vps. This is different from the original IPFS paper!

As shown from the above picture, user is far away from the video file node1 and node2, we can pay very very few money to Node4 and Node5, and ask them to serve as temporary "CDN" to stream the video to user near them. In thus a way, the cost of traditional CDN is greatly reduced. It is more like a retailer with zero inventory!

To better protect the privacy of the tenant, we slice the files into several trunks,each stored in different nodes, thus the node owner can not get the information of the files stored in his "home-sharing alike" storage rooms, because it is only a fraction of the original file. Only the file owner have links to all the trunks stored among the IPFS! As illustrated below :



since the price is very low,for safety reasons, one can replicate his files on multiple nodes, just in case some nodes act dirtily and delete your file. Offcourse he should get punished accordingly in a way,like paying penalty.

And since every node is acting like a user at the same time, you may also have some files stored in other nodes for some reasons, so the money he spends for storage on other nodes will be compensated by his tenants, so this is equivalent to the dynamic storage exchange.

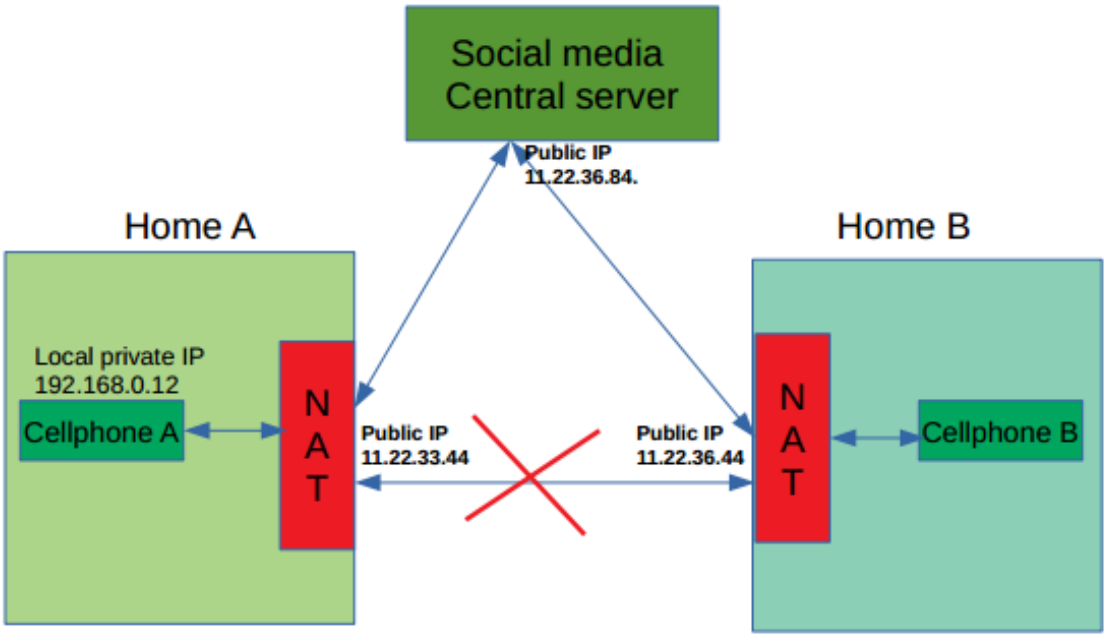
How do we pay the bill for IPFS storage, just like we pay for VPS

Anykind of money they agree on.

Is IPFS real peer to peer

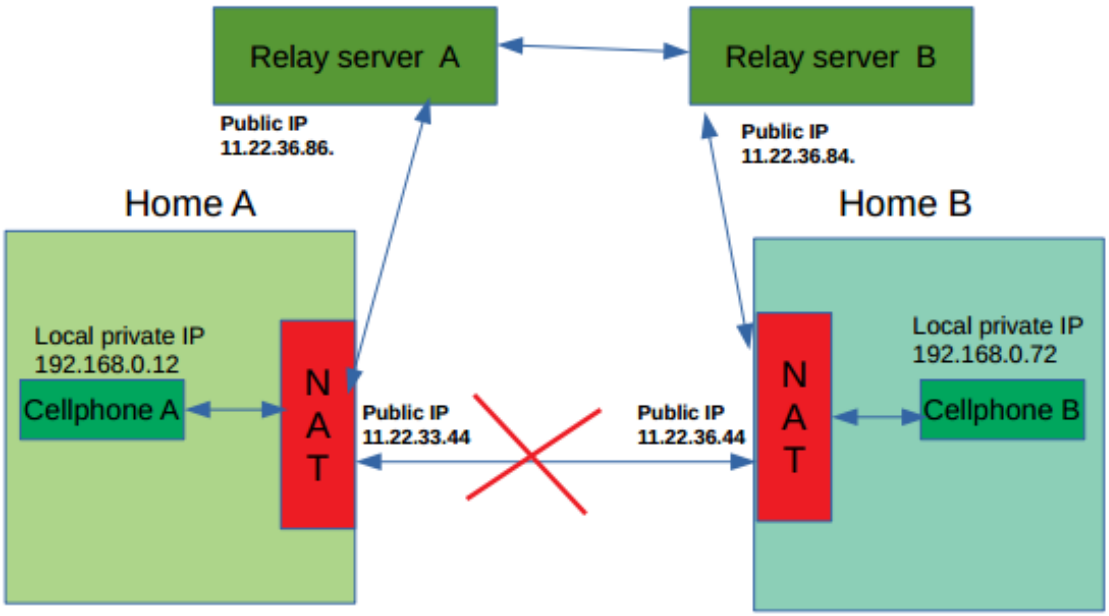
What does peer to peer mean? It means device like computers or cellphones talks to each other without going thru third party.

Well, to answer this question, first let us see is it possible for a user device A find and talk to user device B.

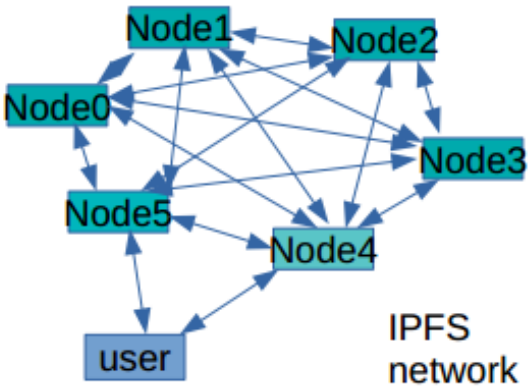


NAT: network address translation, it maps private local IP to the home's public IP. It monitors outgoing network data and incoming network data, for security reasons, some Home NAT refuse the incoming network data whose source public IP is not recorded in the NAT list, intuitively speaking, every incoming data to home NAT need invitation.

so for most home NAT, since most devices behind home NAT need “invitation” to get permission from to get pass the NAT and talk to the devices, the direct peer to peer connection is not available, we need to go thru some kind of relay,just like above social media central server, but the drawback of such system is that the social media have the power to censors your content and connections. To overcome such centralized relay, we can rend our own VPS as a relay. Like below :



so from the above discussion, we know that in physical layer to build below IPFS ideal peer to peer network we need to have a trusted relay node for most device behind NAT to join the peer to peer network.

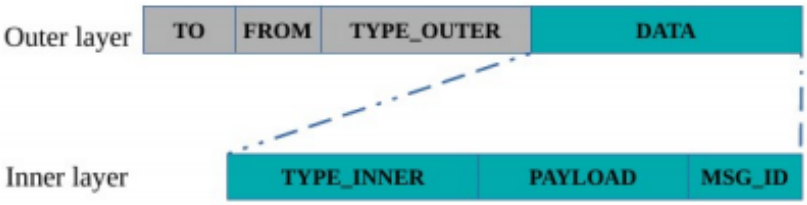


So we can safely say that IPFS is not real peer to peer for most use case where some device behind the NAT need a relay to join the IPFS network. But when devices behind the NAT are able to communicate with each other directly, the it is true peer to peer !

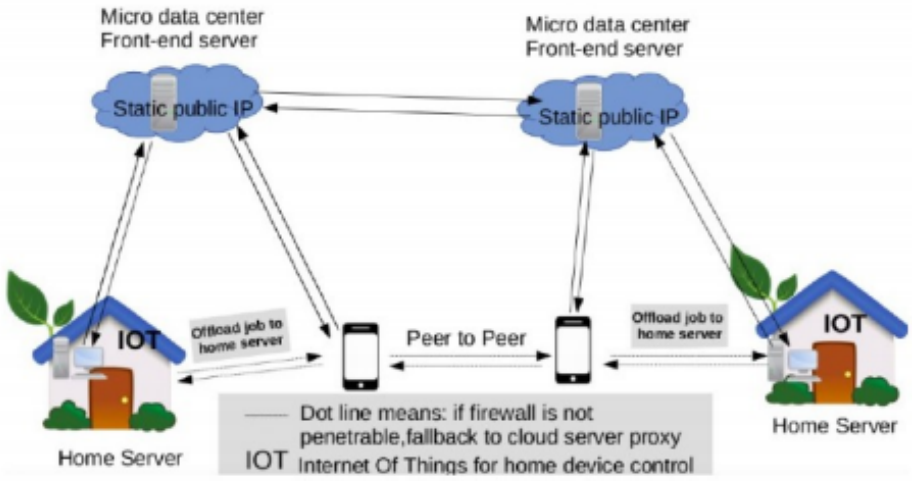
So under the hook, the basic physical network layout looks like this :

DeCensorMedia Protocol : 大道至簡!

In order for individual website to talk to each other, we design such a simple protocol :
網絡世界的大同：你的前端可以和我的後端做基本的 API 通信，我的前端也可以和你的後端做基本的 API 通信。



The encryption methods of outer layer and inner layer are different, cloud server (front end server) can only encrypt the outer layer, home server can encrypt inner layer, inner layer is for data transmission between frontend device and homeserver.



2022年12月20日