Getting to know IPFS

Thu 03 August 2017 | by Mark Pors | in blockchain train journal

This article is part 3 of the Blockchain train journal, start reading here: Catching the Blockchain Train.

Let's read a bit (reading is good for you!)

I already listed this article in the previous episode, but I mention it again because it is a great place to start to understand IPFS and its context: HTTP is obsolete. It's time for the distributed, permanent web.

A couple of things we can pick up from it (a bit simplified):

- IPFS consists of a network of peer-to-peer nodes (aka computers that talk to each other directly)
- These nodes can store content (any type of file)
- Content is represented by a hash and is immutable (if the content changes, so does the hash)
- A node can request content from other nodes by using this hash. This is pretty cool: there is a permanent relation between the hash and the content itself. Unlike the current web where the content behind a URL can change.
- Nodes can decide to store a copy of any content
- The more nodes store a piece of content the harder it is to get rid of it (the permanent web).
- The network also gets faster that way, similar to bittorrent getting faster when the number of seeders goes up (IPFS is partly based on the bittorrent protocol, but one of the differences is that it prevents duplicate pieces of identical content)

A bit more detail:

- Content (files) is broken up into blocks, and each block gets a hash
- Duplicate blocks (identical content but different hash) are removed from the network
- IPFS has a directory concept (another hash) that points to the hashes of the content inside
- Git like version control is used for blocks
- To solve the problem of finding the last version of some content is solved by IPNS
- IPNS allows anyone to create a unique public link to any content that is out there.
- So an IPFS hash points to some immutable content, regardless the version; the IPNS address points to some file or directory determined by the creator of that address
- This IPNS address is implemented as a pubkeyhash which is similar to the public address in a Bitcoin wallet.
- IPNS hashes are long and ugly. Also, browsers don't speak IPFS. There are a couple of solutions to bridge the gap to the current time that we will look into later.

To read and discover more about IPFS, this page on github is a great starting point ipfs/ipfs. Make sure to check out the quick summary.

OK, it is all still a bit abstract I guess, so time to play.

Terminal time (move those fingers buddy!)

Now we understand the basics lets start playing a bit.

To get up and running and to be able to play along, first install (I recommend the ipfs-update approach) and follow the getting started.

Now it is time to set the first step into making this blog decentralized. BTW what a great domain this site has! Well, as long as you are reading the centralized version :)

I uploaded the statically generated website to my server and it looks like this:

```
├── archives.html
├── author
│   └── mark-pors.html
├── authors.html
├── catching-the-blockchain-train.html
├── categories.html
├── category
│   └── blockchain-train-journal.html
├── feeds
│   ├── all.atom.xml
│   └── blockchain-train-journal.atom.xml
├── getting-to-know-ipfs.html
├── index.html
├── picking-a-decentralized-storage-system.html
├── tags.html
└── theme
    ├── css
     ... // ...
            └── youtube.png
```

To shoot a file into the interplanetary file system, we simply do:

```
$ ipfs init # this creates your node's peer ID
$ ipfs daemon # start your local node
$ ipfs add catching-the-blockchain-train.html
added QmXyZcrThrfWQSTKzPiNT4Nd2RcqcVQ3tr7rmFqHYZ3fq4 catching-the-blockchain
```

Now my node can serve this file and tell the network there is a piece of content (my first blog post!) available, just in case someone requests this hash: `QmXyZcrThrfWQSTKzPiNT4Nd2RcqcVQ3tr7rmFqHYZ3fq4`.

So, in your terminal:

```
ipfs cat QmXyZcrThrfWQSTKzPiNT4Nd2RcqcVQ3tr7rmFqHYZ3fq4
```

should return the same as (for as long as I host it over HTTP of course):

```
curl http://decentralized.blog/catching-the-blockchain-train.html
```

Notice that when you request the file for the second time through ipfs, it gets it very fast. That is because your node now has a copy of the content locally. Pretty cool huh?

The ipfs node comes with an HTTP gateway, so we can also access the file in a browser: http://decentralized.blog:8080/ipfs/QmXyZcrThrfWQSTKzPiNT4Nd2RcqcVQ3tr7rmFqHYZ3fq4

This is what happened here (from the getting started document):

> *The gateway served a file from your computer. The gateway queried the DHT, found your machine, requested the file, your machine sent it to the gateway, and the gateway sent it to your browser.*

Note: this obviously introduces a central server into the picture, but at least anyone can run such a gateway. So it is still sort-of-decentralized. We have to look for something better later on.

Decentralize it! At last some progress

To push the whole website onto the interplanetary file system can't be more simple, just add the `-r` flag and point to the directory containing all static files.

```
$ ipfs add -r ../../www/dcb
added QmWu4hsywXoSrw5JRhjUadyMnDuxuvcgExwVD62cWeLVjb dcb/archives.html
added QmUCJ6z2EfikXEDUYbxfjdkrX2zw62XCAM93HHC61Qrmwg dcb/author/mark-pors.ht
added QmV13vUiKU64oHoxbp5MN7bnehraW46de82PLvndbisn64 dcb/authors.html
added QmXyZcrThrfWQSTKzPiNT4Nd2RcqcVQ3tr7rmFqHYZ3fq4 dcb/catching-the-blockc
added QmRiWHfYLQEJoKz2dDq2sXnW2keiKZG1nHbnaN55TTXN8b dcb/categories.html
added QmSCx2eV3BjHJWoXMBc7rKakPv3fAcewyroFCRmovcsYSy dcb/category/blockchain
added QmVxVLT22KLGasBLxzSv6yXofx3GVXmWEd3HqSjPp1X6Zz dcb/feeds/all.atom.xml
added QmdKJY5VVd8Qm5xsLt8Vj5XwvTbe5mCQhK6DnZ9LkqTHwz dcb/feeds/blockchain-tr
added QmYpyQ3MXYZum3q2E9pRxUWnJXK74cLvFbuMDYukK9ivK1 dcb/index.html
added QmXkfv9ZXb9evxPXTMJC8fvgeZymYpTQo3CVFv7mDoSRTN dcb/picking-a-decentral
added QmSyuSjmzhwkoj6qH7s4MmKeB98Ub1DZ1Hm5anBBnR9yVu dcb/tags.html
added QmSfLETTMQc1b9cBJENH1S4NK5qniXSf79KSP8h5CbU3U8 dcb/theme/css/main.css
... // ...
added QmU8CmEYWV5aUuqoToq1xiyqftdY8MHQ8MdViA2ex23uyV dcb/theme/images/icons/
added QmYBVeJkBRdJYj8jr3YTGWE537h26YqcfjvbX4ttjZ65X2 dcb/author
added QmPwQPX86RQi6MrZQigh2tWgztcbf46EjhRg1hzMzELvHN dcb/category
added QmfRxCPp417G3NTqEpBmwtGwtUKFzgQjBRvxNBq88HhC9F dcb/feeds
added QmYUnPcDPcnKA3WFembvNYG48m1oUWa7QLNWZxrkLv1vcE dcb/theme/css
added QmfBq8BDXxhu9cLAG4qhxqY6K5xDQCBWyoYqX5fTmQbfwW dcb/theme/images/icons
added Qme4Dt6vQ1eAVcEs7dY55cAMhTsRJxTakANhZjX9EKvGfF dcb/theme/images
added QmciTMAUiEqMpar2u9n1dJNJStmrP6fjMESyH6ZS5eo56H dcb/theme
added QmcPx9ZQboyHw8T7Afe4DbWFcJYocef5Pe4H3u7eK1osnQ dcb
```

Note: my first try failed when I tried to add a symlink to the network, that doesn't work.

See that the hash for `dcb/catching-the-blockchain-train.html` is the same as we previously got? That's the de-duplication in action.

Now when we take the bottom hash from this list and feed it to the gateway, we see the blog: http://decentralized.blog:8080/ipfs/QmcPx9ZQboyHw8T7Afe4DbWFcJYocef5Pe4H3u7eK1osnQ/. All relative links work just fine; it's beautiful!

In case my gateway is down, just try the one provided by IPFS: https://gateway.ipfs.io/ipfs/QmcPx9ZQboyHw8T7Afe4DbWFcJYocef5Pe4H3u7eK1osnQ/ or your local gateway: http://127.0.0.1:8080/ipfs/QmcPx9ZQboyHw8T7Afe4DbWFcJYocef5Pe4H3u7eK1osnQ/. Now is that cool or what?

There is still a small issue: it is the previous version of the blog because I'm still writing this article and when I upload it when I'm done the hash changes, but then the new hash won't be in the article...

There is a solution for that:

A permanent address for this blog with IPNS

The usage of IPNS is explained here: The Inter-Planetary Naming System.

So in our case, on the node that runs on my server, I do:

```
$ ipfs name publish QmcPx9ZQboyHw8T7Afe4DbWFcJYocef5Pe4H3u7eK1osnQ
Published to QmRf4ERGvYpVo6HRa2VueZT8pWi8YvyLS3rW6ad2y83tdN: /ipfs/QmcPx9ZQb
```

This results in an address for the blog that won't change: `QmRf4ERGvYpVo6HRa2VueZT8pWi8YvyLS3rW6ad2y83tdN`.

This address points (when I executed the command just right now) to the directory containing the static files (index.html is served when it is in that directory BTW).

When you read this, the address `QmRf4ERGvYpVo6HRa2VueZT8pWi8YvyLS3rW6ad2y83tdN` is still the home page address, but it will contain new content (I will do another `publish` when I'm done writing this post).

So our static blog is now decentralized and available through http://decentralized.blog:8080/ipns/QmRf4ERGvYpVo6HRa2VueZT8pWi8YvyLS3rW6ad2y83tdN/. Note the `ipns` part instead of `ipfs` this time.

IPFS is pretty cool, and it makes our lives much easier. Before we move on to improving our blog I'd like to look a little bit under the hood the understand the workings of IPFS (and IPNS) a bit more...

Read on here: Understanding the IPFS White Paper part 1.

social

atom feed

Github

Twitter