IPFS: Interplanetary File Systems

Introduction

- A blockchain is fundamentally a **decentralized system**.
- Alternatively, a **decentralized peer-to-peer system** can be realized independent of a blockchain.
- Interplanetary File System (IPFS) is a fine example of such a system.
- IPFS is a decentralized model for file transfer in contrast to the centralized namespace and transfer provided by the http family of protocols.
- Why Interplanetary??

Introduction

- HTTP operates in a centralized, hierarchical namespace: a web site is identified by http://www.jiit.ac.in where each item is resolved hierarchically by the domain name service (DNS).

- This peer-peer transfer of data is not new! It has been an age old quest.
- Recall or the **Napster and Gnutella** media sharing services. And the **bittorrent** services that is underlying many of our current data services.

- Juan Benet in the IPFS white paper describes it as a “Content Addressed, Versioned, P2P Filesystem”.

- IPFS is an alternative approach to file sharing, a decentralized approach.

Features

- Immutable Files
- Versioning
- Divide file into blocks
- Inbuilt Security using hashing
- Distributed (peer to peer)
- Content oriented

Key Ideas

Once again similar to how Bitcoin did, IPFS leverages and combines many successful peer-to-peer system ideas.

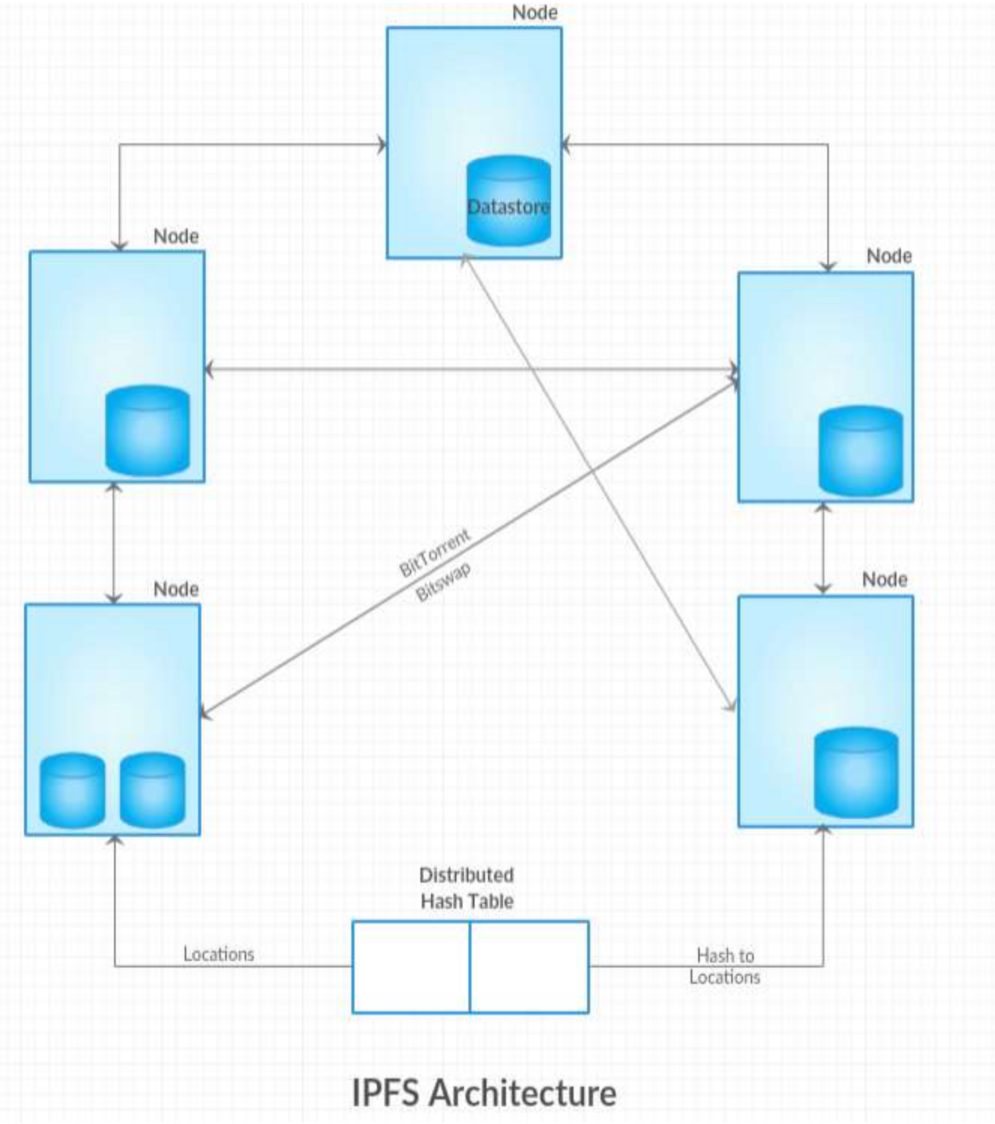
1. Global distributed file system: IPFS is about “distribution” decentralization.
2. Content-based identification with secure hash of contents; Resolving locations using Distributed Hash Table (DHT)
3. Block exchanges using popular Bittorrent peer-to-peer file distribution protocol
4. Incentivized block exchange using Bitswap protocol
5. Merkle DAG (Directed Acyclic Graph) version-based organization of files, similar Git version control system
6. Self-certification servers for the storage nodes for security

IPFS Architecture

This is a high level view of the architecture of IPFS.

Files in distributed storage, and distributed hash table, uses the hash of the file as a key to return the location of the file.

Once the location is determined, the transfer takes place peer-to-peer as a decentralized transfer.



Distributed Nodes

The nodes are the computers that holds the decentralized data file objects that form the global file system.

Nodes are identified by cryptographic hashes of public key. (Similar to our blockchain nodes).

They hold the objects that form the files to be exchanged.

Objects are identified by a secure hash and an object may contain sub objects each with its own hash.

That hash is used in the creation of the root hash of the object.

Recall our Merkle tree from course 1: Blockchain Basics.

Content Addressable

In the current world wide web protocol, we typically refer to a web resource or data by the server on which they are stored.

For example, https://www.jiit.ac.in/ actually refers to the server on which the JIIT page is hosted and to a particular directory and file on that server.

This is a “centralized” approach. (i.e. location addressable)

What if the resource is available in numerous and dynamically variable number of locations?

IPFS offers a decentralized solution for this.

Content Addressable

IPFS identifies the resources by a hash of **resource/file**. (Instead of identifying the resource by its location as in HTTP.)

IPFS identifies it by its content or by the secure hash of its content.

The file is addressed by a universally unique identifier instead of by location.

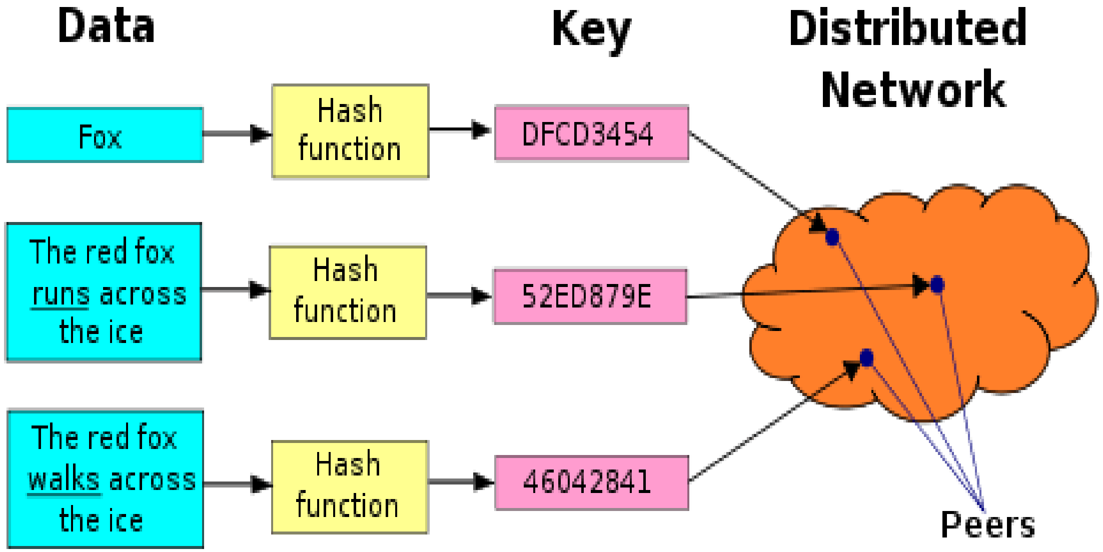
Then how do you resolve the location? Just like you have a URL or link of a website, you start with the **hash identifier** of the resource.

Send a request for anyone with a resource with this identifier; on a successful response, access it peer-to-peer.

If content addressable, how do you resolve location of objects?

Routing part of the IPFS protocol maintains a DHT (Distributed Hash Table) for the locating the nodes as well as for file objects.

A simple DHT hold the hash as the key and location as the value. Key can directly hash into the location.



Exchange the blocks of the file

The peer nodes holding the data blocks are searched by a protocol called BitSwap.

DHT **resolves to the closest location** to the key value.

Peer nodes have a **want_list** and **have_list** and some form of a **barter system** (without use of money i.e. goods exchange) is formed.

No free riding

Any imbalance is noted in form of a **BitSwap credit and debt;**

Exchange the blocks of the file

Bitswap protocol **manages the block exchanges** involving the nodes accordingly.

The nodes in the network thus have to provide value in the form of blocks.

(Hmm... This could be an ideal usecase for a “digital token?”; if you send a block you get a IPFS token that can be used when you need a block.)

The Bitswap protocol has **provisions for handling exceptions** such as:

Free loading node
Node wanting nothing
Node having nothing

Multiple versions maintainenance

Multiple versions maintained using **a Merkle Directed Acyclic Graph** data structure

The basic elements of the block, list of blocks, tree of block representing an instance, and commit that is snapshot of the tree.

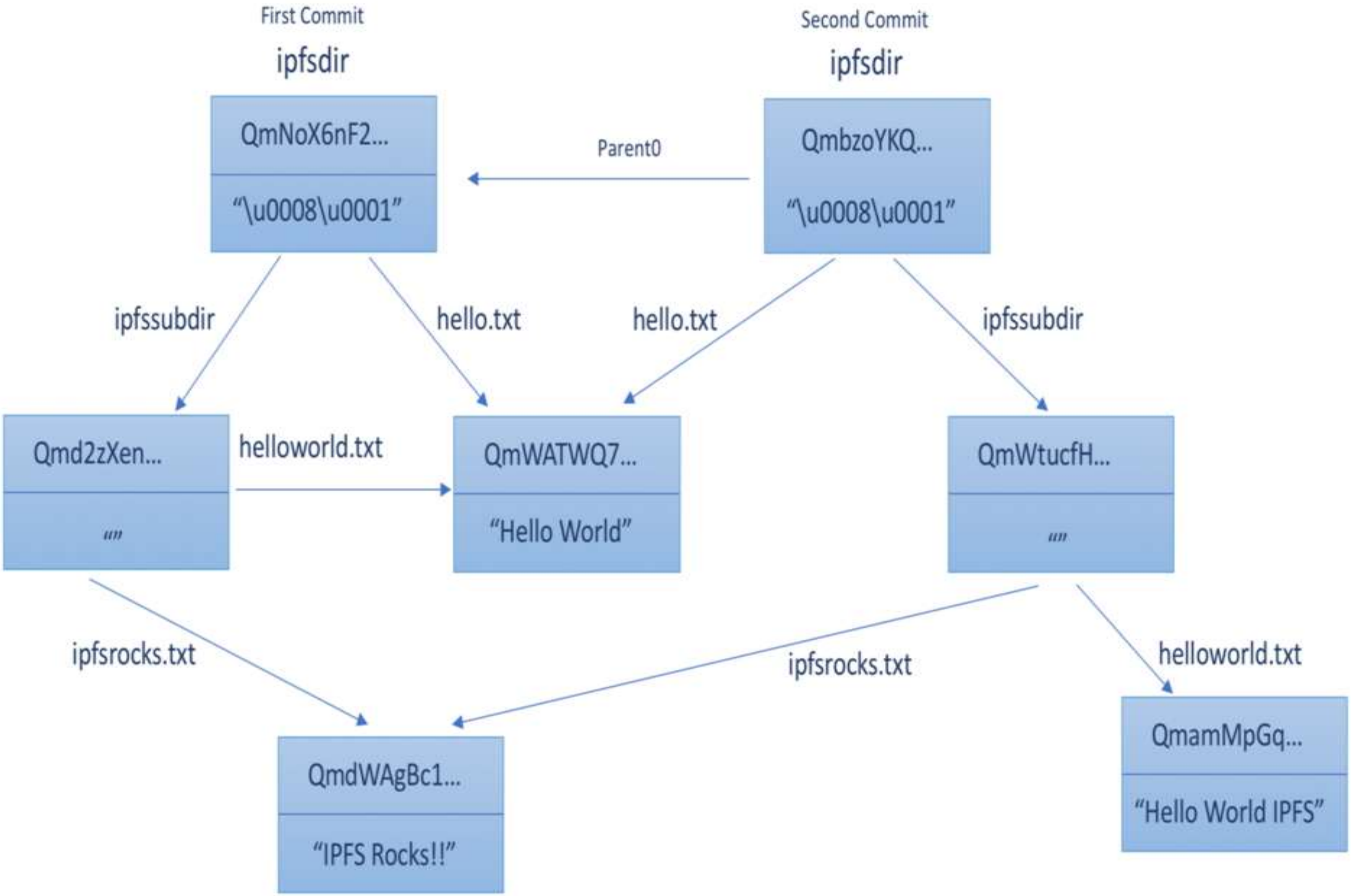
This Merkle DAG also helps in **checking** any **tampering** and also in **deduplication**.

Size of IPFS shared file

Default size 256 KB

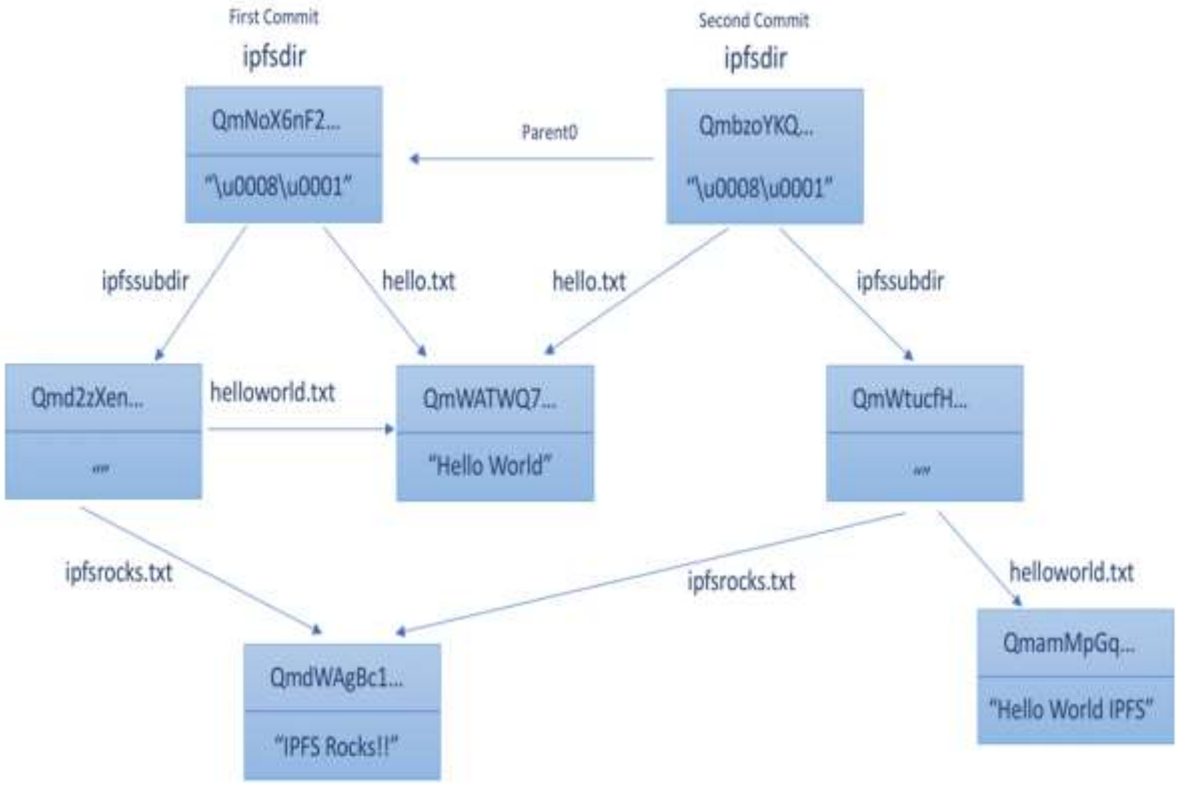
If file size > 256 KB (e.g. image/audio/video):

- Divide multimedia file into chunks of 256 KB
- Link all the chunks properly


First Commit
ipfsdir
QmNoX6nF2...
"\u0008\u0001"
Second Commit
ipfsdir
QmbzoYKQ...
"\u0008\u0001"
Parent0
ipfssubdir
hello.txt
hello.txt
ipfssubdir
Qmd2zXen...
""
helloworld.txt
QmWATWQ7...
"Hello World"
QmWtucfH...
""
ipfsrocks.txt
ipfsrocks.txt
helloworld.txt
QmdWAgBc1...
"IPFS Rocks!!"
QmamMpGq...
"Hello World IPFS"

IPFS shared files

The picture here depicts the You can observe in this picture two commits of the course3Dir, the four nodes on the left form the first commit and the three nodes on the right the second commit. The is DAG instead of a Merkle tree we have seen in Ethereum state root. You can observe deduplication, that is same files are shared. There are two shared files



Use cases and Relationship to Blockchain

IPFS can be a standalone decentralized file system.

It can be complementary to the existing HTTP based centralized system.

We discussed it in the context of blockchain systems because it can serve an important role of decentralized storage for blockchain application that have a lot of data, but will store only the hash on the blockchain.

In this case instead of a centralized store, IPFS can be the decentralized store that work in tandem with the decentralized ledger technology of the blockchain to create a powerful solution for many storage-rich business usecases.

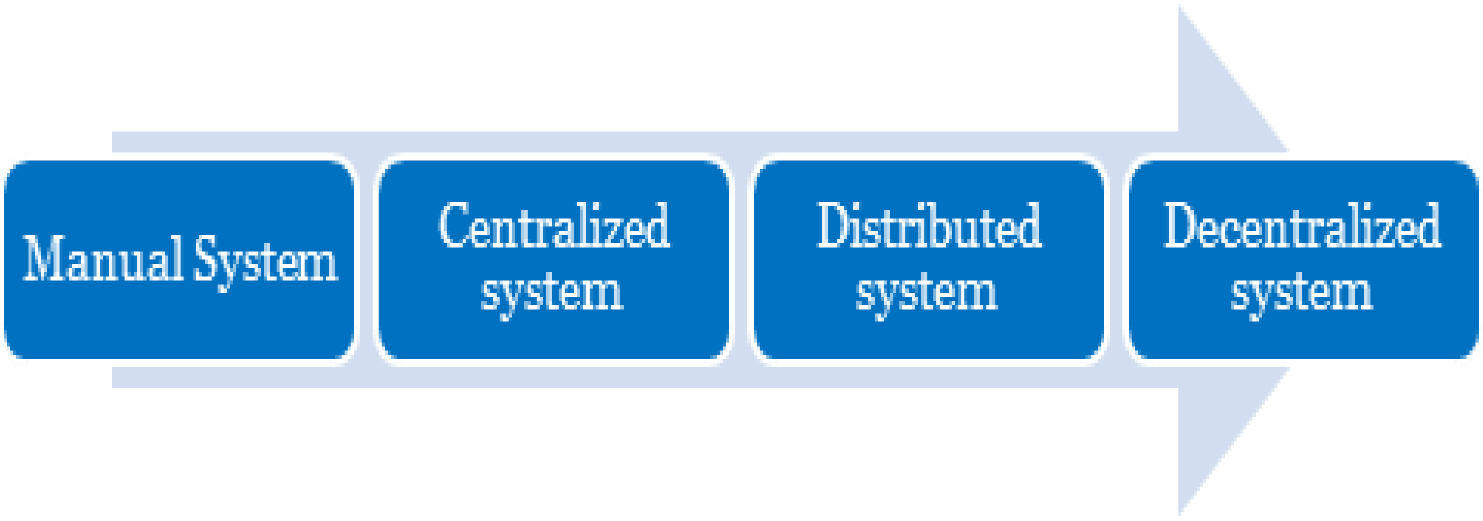
Limitations:

Unavailability of file issue?
Serving system off during downloading?

Summarizing:

We discussed the details of a decentralized storage system that can be used for storing the off-chain data for a blockchain application.

It is used in many genomic data applications for storing large genomic data and in dapps such as Openlaw for document storage.



References

- Juan Bennet’s IPFS whitepaper: https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf