Distributed Web Applications with IPFS, Tutorial

David Dias mail@daviddias.me and Juan Benet juan@benet.ai

Protocol Labs

Abstract The contents of this document describe the tutorial session delivered at ICWE 2016, on Building Distributed Web Applications with IPFS. IPFS, the InterPlanetary File System, is the distributed and permanent Web, a protocol to make the Web faster, more secure and open. The tutorial format focuses in key elements of IPFS and how to use it to build applications with.

Keywords: IPFS, Web, Distributed, P2P, Cryptography, MerkleTree, MerkleDAG, IPLD, Go, JavaScript, Application, Apps, Blockchain, Hash, Secure, Data, File System, Files, Graphs, Database

1   Introduction

IPFS [1], the InterPlanetary File System, is the distributed and permanent Web, a protocol to make the Web faster, more secure, open and available. IPFS can be described "as Git meets a BitTorrent swarm", exchanging objects within one Git repository. In other words, IPFS provides a high throughput content-addressed block storage model, with content-addressed hyperlinks. This forms a generalised MerkleDAG, a data structure that can beused to build versioned file systems, blockchains, unix like file systems, amongst other options. IPFS combines a Distributed Hash Table, an incentivised block exchange and a self-certifying namespace. IPFS has no single point of failure, and nodes do not need to trust each other.

The tutorial will focus on the IPFS Application Stack, including: libp2p, the networking layer; bitswap for data exchange;IPLD and the MerkleDAG, the thin waist data structure of IPFS and how to use IPFS interface to build distributed applications. The full length of the tutorial is 6 hours.

2   Motivations and goals

Distributed Applications are naturally complex, it is known that a ton of work and energy has been put by professionals from the Academia and the Industry to solve some of the toughest problems to deliver guarantees and semantics that applications can use and trust.

IPFS builds on top of the knowledge gathered along the last decades, offering semantics that are similar to those of a file system, but built with the Web in mind. This tutorial covers the core challenges of building a distributed application and offers a structure to solve those problems through IPFS.

3   IPFS, the InterPlanetary FileSystem

A complete IPFS description can be found on the IPFS paper [1] and the specifications repository [1] inside the IPFS organization on Github. Here we present a distilled version on each layer of IPFS.

3.1   libp2p

libp2p [2] is the networking layer library built for IPFS, which exists as a standalone project. libp2p is a modularized and extensible approach to the network stack, that reuses existing transports and system capabilities to offer NAT Traversal, Peer Discovery, Routing, Stream Multiplexing, Protocol Multiplexing, Encryption, Authentication and more.

3.2   Data Exchange

The IPFS Data Exchange takes care of negotiating bulk data transfers, exchanging blocks that are wanted by a given peer in a market like approach.

3.3   MerkleDAG

The ipfs merkledag is a directed acyclic graph whose edges are merkle-links. This means that links to objects can authenticate the objects themselves, and that every object contains a secure representation of its children.

This is a powerful primitive for distributed systems computations. The merkledag simplifies distributed protocols by providing an append-only authenticated datastructure. Parties can communicate and exchange secure references (merkle-links) to objects. The references are enough to verify the correctness of the object at a later time, which allows the objects themselves to be served over untrusted channels. Merkledags also allow the branching of a datastructure and subsequent merging, as in the version control system git. More generally, merkledags simplify the construction of Secure CRDTs, which enable distributed, convergent, commutative computation in an authenticated, secure way.

3.4   Data Structures

Any kind of Data Structure that can be described in a graph format can be mounted on top of the IPFS MerkleDAG. A pratical example is the included UnixFS, a Data Structure that is built into IPFS offering a native way to handle files in familiar way to filesystems in Unix environments.

---

[1] https://github.com/ipfs/specs
[2] https://github.com/ipfs/specs/tree/master/libp2p

3.5 Naming

Naming inside IPFS is governed by IPNS, the InterPlanetary Naming System. IPNS takes ideas from SFS [2] to enable the creation of cryptographically signed mutable pointers, which can be used to the creation of name records inside the network.

4 Tutorial

The format of the tutorial is a sum of a lecture and a laboratory class. Participants will get a common base of understanding of what is IPFS through a lecture, followed by an hands on laboratory class, where each individual will work in parallel and build a Distributed Web Application with IPFS. This tasks will be divided in a series of steps, starting from the network, followed by the data model, design decisions on desired guarantees and so on.

During these practical steps, participants will interact with the several layers of IPFS and their respective APIs.

4.1 Learning outcomes

During this session, it is expected that attendees learn how to:

– install IPFS
– use IPFS CLI
– use IPFS HTTPitemAPI
– use IPFS client libraries
– load an application from IPFS
– dial between peers, using IPFS (libp2p)
– learn how to build data structures on top of the Object (DAG) API
– learn how to use UnixFS

4.2 Target audience

General interest in cryptography, distributed systems, data structures and P2P protocols is recommended. Familiarity with JavaScript or the Go programming languages are a bonus, however any knowledge of CS programming will suffice.

4.3 Curriculum

This tutorial covers the IPFS stack and topics to have into account when building distributed applications, including:

– Merkle Trees
– Merkle'lized Data Structures
– MerkleDAG
– IPLD

- Hashing functions
- Distributed Hash Tables
- Routing strategies and protocols
- Peer Discovery
- Transport protocls
- IPFS HTTP API
- IPFS Core API
- IPFS CLI
- Data Exchange, namely bitswap
- Self Describable Data Formats

The tutorial is custom build for ICWE2016 audience, taking into consideration participants experience with the subject and interest.

5   Presenter

David Dias is a P2P Software Engineer and Researcher at Protocol Labs (http://ipn.io), the company behind IPFS. Before, David worked on the security and web development industry at Lift Security.

David holds a P2P Masters in Science, having built the first P2P DHT using WebRTC specifically for the Web Platform for job execution distribution.

David's speaking history with regards to P2P, security and distributed systems has been the following:

- Dec 2015 | Linux Foundation Node.js Interactive Conference. Stellar Module Management, using IPFS for code package distribution.
- May 2015 | Data Terra Nemo, the P2P Conf. "webrtc-explorer", distributed browser computing platform using volunteered shared resources
- Feb 2015 | OpoJS. Resource Discovery for the Web Platform on top of a P2P Overlay Network powered by WebRTC
- Apr 2014 | JSConf Brazil. Securing Node.js Applications by the community and for the community

Currently I'm also an invited Professor at the University of Lisbon, having developed a new post graduation course on modern web development.

Other relevant previous work:

- Developed a 2 day WebRTC Training for O'Reilly Fluent Conf 2015
- Node Security researcher
- Part of the european research project SynergyVM while a researcher on the distributed systems group at INESC-ID

6   Conclusion

With this tutorial, participants will become knowledgeable of the components that compose IPFS, its architecture design decisions and be confortable to build Web Applications with IPFS.

The materials built for this tutorial will be made available under MIT License, under the IPFS organization on GitHub[3]. Participants are encouraged to consult them at any time or distribute amonsgt their colleagues in their organization. We will continue to improve the content as IPFS evolves.

We would like to acknowledge and send our appreciation to the organizing committee behind ICWE 2016 for having invited us to make this Tutorial happen.

References

1. Benet, J.: IPFS - Content Addressed, Versioned, P2P File System (2014)
2. Mazieres, D., Kaashoek, F.: Self-certifying file system (2000)

All links were last followed on March 10, 2016.

---

[3] https://github.com/ipfs