第一周周报

汪子龙

本周学习了 IPFS 系统的工作原理，了解其三大底层支持：Multiformats、libp2p 和 IPLD，还有 IPFS 网络的八层架构，以及其激励层 Filecoin。对激励层的共识机制和一些证明算法都有较浅的了解，下周打算深入学习这些算法并试图找到研究方向。

IPFS（InterPlanetary File System）是一种内容可寻址的对等超媒体分发协议，在 IPFS 网络中的节点构成一个分布式文件系统，由 Multiformats、Libp2p 和 IPLD 提供底层支持。

图 1 IPFS 的组成

一、加密协议 Multiformats

Mutiformats 是一系列 hash 加密算法和自描述方式（从值上就可以知道值是如何生成）的集合，具有 SHA1\SHA256\SHA512\Blake3B 等 6 种主流的加密方式，用以加密和描述 nodeID 以及指纹数据的生成。

Multiformats 协议包含以下协议：

1. multihash：自描述哈希，格式为<哈希函数><数据长度><数据值>。
2. multiaddr：自描述网络地址，格式为(/<网络协议>/<目的地址>)+。
3. multibase：自描述基编码，格式为<基编码字符><数据>，其中基编码字符表示编码的方式，需要查阅 multibase table 确定。
4. multicodec：自描述序列化。
5. multistream：自描述流网络协议。
6. multigram：自描述分组网络协议。

Multiformats 协议的支撑使得 IPFS 的存储开创了一种全新的安全模式，对所有的内容都进行加密，有效保证了数据的安全和用户的隐私权。

二、数据模型 IPLD

1 内容标识符

IPFS 根据文件内容，使用哈希算法生成 CID（Content Identifier)。CID 目前有两个版本，各版本组成如下：

1.CIDv0

以“Qm”开头的 46 位字符串，采用 base58 编码。

2.CIDv1

编码方式前缀，用于指明 CID 剩余部分的编码方式；

版本标识符，用于指明使用的 CID 版本；

文件格式标识符，用于说明目标文件的格式，便于取得文件后的后续操作；

基于文件内容的哈希值。

2 IPLD

IPFS 使用 IPLD(InterPlanetary Linked Data)数据模型，实现数据节点间的链接。IPLD 定义了一种简单的、适用于所有 MerkleDags、基于 JSON 的结构。以下是一个 IPLD 节点的 JSON 示例：

```
{
  "title": "As We May Think",
  "author": {
     "/": "QmBBB...BBB"
  }
}
```

假设该节点的 CID 为 QmAAA...AAA，该节点通过子路径 author 链接到 CID 为 QmBBB...BBB 的节点。我们可以使用路径 QmAAA...AAA/author 访问到节点 QmBBB...BBB 中的内容。

三、数据传输协议 Libp2p

Libp2p 模块在 IPFS 中主要负责数据的传递，提供了 p2p 文件系统一些核心功能的解决方案，如传输、安全、节点路由和网络发现等。

Libp2p 具有以下特点：

1.采用 multiaddr 协议寻址,如地址/ipv4/192.168.1.1/udp/80 表示使用 ipv4 协议，向地址为 192.168.1.1 的主机的 80 端口，发送 upd 数据包。

2.运输时使用的协议由开发者选择，开发者可以同时选择支持多种运输协议。

3.采用 secio 协议进行信道验证和加密,可以保证接收方身份的正确性和运输时的安全性。

4.每一个 Libp2p 节点都包含一个私钥和一个公钥（PeerId)，公钥使用哈希算法生成，采用 multihash 协议编码。

5.使用 Kademlia 算法生成 DHT（Distributed Hash Table）实现节点路由。

四、IPFS 八层协议

IPFS是一个基于P2P的分布式文件系统,IPFS协议具有八层子协议栈分别为:身份、网络、路由、交换、对象、文件、命名、应用。

1 身份层和路由层

IPFS 节点身份信息的生成以及路由规则是通过 Kademlia 协议生成制定。Kademlia协议构建了一个DHT,每个加入这个DHT网络的人都要生成自己的身份信息,然后才能通过这个身份信息去负责存储这个网络里的资源信息和其他成员的联系信息。

2 网络层

IPFS使用Libp2p协议可以支持任意传输层协议（见第三节）。

3 交换层

IPFS 在 BitTorrent 的基础上使用 BitSwap 协议实现与其他节点的块交换。IPFS每一个节点都维护了两个列表：已有的数据块（have_list）和想要的数据块（want_list）。与 BitTorrent 不同的是：BitSwap 可以从不属于本文件的其他文件获取数据块(只要数据块的哈希值一样,数据内容必然一样),从全局考虑,这使得BitSwap的效率相比于BitTorrent更高。

为了激励节点寻找并分享数据，BitSwap根据节点的之间的数据收发建立了一个信用体系：发送给其他节点数据可以增加信用值，从其他节点接受数据降低信用值。如果一个节点只接收数据而不分享数据，信用值就会降得很低而被其他节点忽略掉。

BitSwap 节点会记录下来和其他节点通信的账单（数据收发），可以保持节点间数据交换的历史和防止篡改。当两个节点之间建立连接的时候，BitSwap会相互交换账单信息，如果账单不匹配，则清除重新记账。恶意节点可能会故意“丢失”账单，以希望清除掉自己的债务。其它交互节点会把这些都记下来，如果总是发生，节点就会被拒绝。

4 对象层和文件层

IPFS的数据对象以MerkleDAG结构存在，此对象结构具有以下优点：

1.内容寻址：所有内容由其多重哈希校验和唯一标识，包括链接。

2.防篡改：所有内容都通过校验和进行验证。如果数据被篡改或损坏，IPFS会检测到它。

3.重复数据删除:保存完全相同内容的所有对象都是相同的,并且只存储一次。

文件层是一个新的数据结构，采用Git一样的数据结构来支持版本快照：

1.blob：包含一个可寻址的数据单元，并表示一个文件。

2.list：表示由多个连接在一起的blob组成的大型或非重复数据文件。

3.tree：它表示一个目录。

4.commit：版本历史记录中的快照。

5 命名层

具有自我验证的特性：当其他用户获取该对象时，使用指纹公钥进行验签，即验证所用的公钥是否与NodeId匹配，这验证了用户发布对象的真实性，同时也

获取到了可变状态。加入了 IPNS（InterPlanetary Name System）使得加密后的 DAG 对象名可定义，增强可阅读性。

五、激励层 Filecoin

Filecoin 是 IPFS 的激励层，采用区块链机制发行令牌 FIL。矿工挣取 FIL 的方式有两种：通过在存储空间市场向其他用户出售存储空间；通过挖矿产生新的区块并维护链上的共识机制。

Filecoin 使用存储能力共识机制（Storage Power Concensus），涉及到的算法包括：

1 EC（Expected Consensus）

每轮选举前，矿工节点会生成一个 Ticket（由上一轮区块的 PoSt 和节点的地址生成）。在每轮的选举中，节点生成的 Ticket 将被用来参与本轮的 leader 竞选，Ticket 符合条件的节点会得到 WinningTicket，表明该节点在本轮可以成为一个 leader 节点。leader 节点根据 WinningTicket 生成一个 Election Proof，通过这个证明，节点就可以开始生产区块并打包广播。

EC 共识机制每轮预期选取一个 leader 打包区块形成 tipset 上链并获得奖励。若没有符合条件的节点，则本轮不会有新的区块上链；若有多个符合条件的节点，则会计算每个 leader 提交 tipset（区块的集合）的权重，最大的作为主链，权重相同的选 Ticket 值较小者。

2 PoRep（Proof-of-Replication）

PoRep 算法的职能是用来证明一个存储系统确实存储了某一份数据的拷贝，而且每一份拷贝使用不同的物理存储，并用来抵御去中心化系统中三种常见的攻击。

Filecoin 证明机制所涉及到的定义包括：

1.挑战：系统对矿工发起提问，可能是一个问题或者一系列问题，矿工正确的答复，则挑战成功，否则失败。

2.证明者：向系统提供证明了完成系统发起的挑战。

3.验证者：向矿工发起挑战的一方，来检测是否矿工完成了数据存储任务。

4.数据：用户向矿工提交的需要存储或者矿工已经存储的数据。

5.证明：矿工完成挑战时候的回答。

Filecoin 使用的 PoRep 算法是 ZigZagDRG 算法，原始数据 data 首先依次分成一个个小数据（d1-d5），每个小数据将被计算出一个散列值，小数据本身也将散列值作为加密种子来进行编解码。这些小数据的散列值按照 DRG（Depth Robust Graph）建立连接关系。数据块的散列关系将构成 Merkle Tree 结构。这样是为了在进行挑战与检验的时候，无需针对所有数据块解码，即可以快速验证，也更好地抵御了攻击性。

整个 PoRep 的计算过程分为若干层，每一层的 DRG 关系的箭头方向是互斥的，上一层向右，下一层就向左，因此得名 ZigZag。数据解码过程中，每一层之间互不依赖，即可并行执行，相对于串行编码要更为快速。

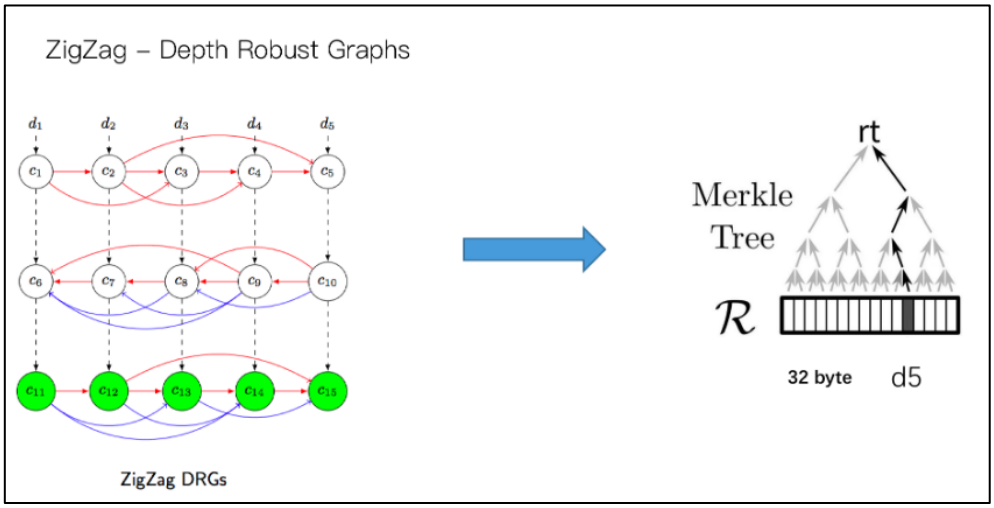

图 2 ZigZagDRG 算法示意

3 PoSt（Proof-of-Spacetime）

PoSt 可以理解为矿工一定时间内持续地生成复制证明和接受挑战和验证的过程。

Filecoin 的 PoSt 算法如下图所示：挑战者在 PoRep 循环重复执行的 i 轮，输入一个随机挑战参数 c，之后挑战参数 c 会被链式递归计算，即上一次的输出作为下一次的输入，重复 t 次之后，最后一次的结果作为 PoSt 的证明，接受反向验证。

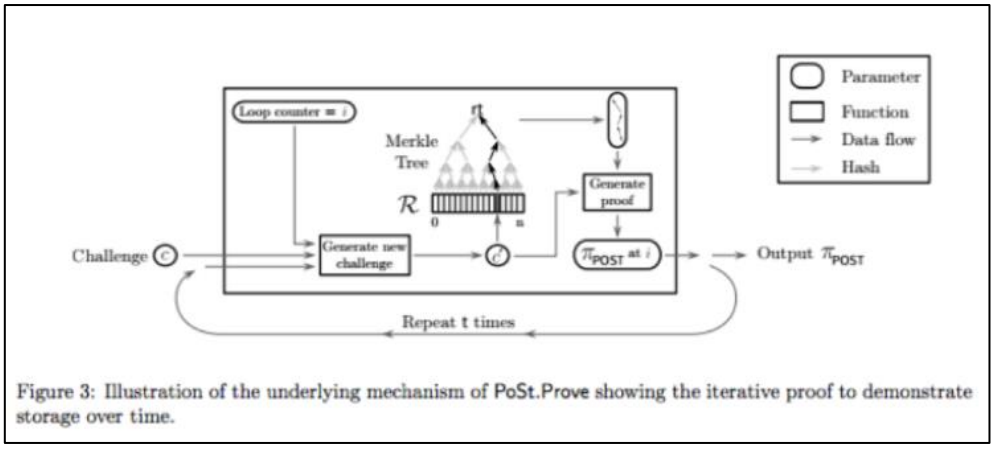

Figure 3: Illustration of the underlying mechanism of PoSt.Prove showing the iterative proof to demonstrate storage over time.

图 3 Filecoin 的 PoSt 算法示意

Filecoin 中定义每隔 20000 个区块（平均 6 天左右），存储矿工必须提供一次 PoSt，表明仍存有用户数据的证明。与此同时，存储市场会每隔 100 个区块，去对 PoSt 发起证明验证，以判断是否需要下发惩罚。