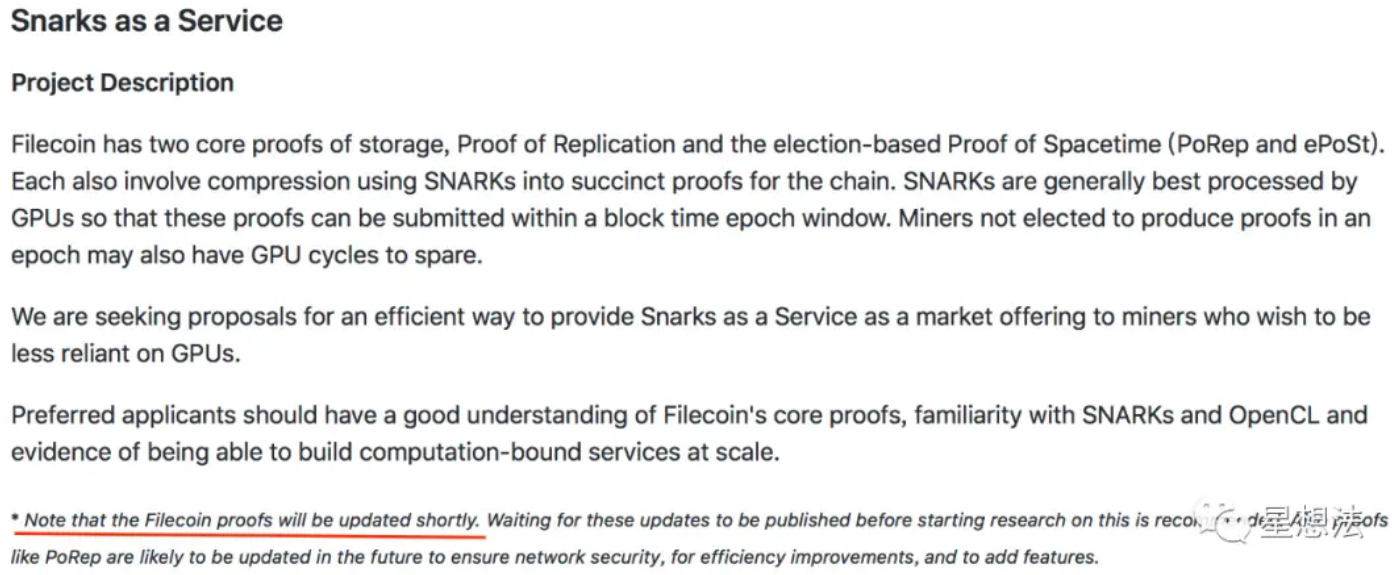
Filecoin-Snark as a Service data volume analysis

FileCoin (https://learnblockchain.cn/tags/FileCoin)

> Snark as a Service is a more interesting service, which specializes in providing zero-knowledge proof computing services in the Filecoin ecosystem. When the sector size is 32G, the amount of data required to prove is about 8M.

Filecoin has postponed again, which is actually expected. Friends looking at the code may find that lotus and rust-fil-proof projects are still updating a lot of code every day. For large-scale project development, code freeze will be performed before going online, and a pressure test will be conducted for a certain period of time. If there are no serious bugs, it can go online. At present, the project is still in the development stage, and there is no code freeze.

Since the last AMA, the Filecoin team has introduced the idea of Snark as a Service. It can be said that a new role is introduced in the entire Filecoin ecosystem, specifically providing zero-knowledge proof generation services. Ordinary miners can outsource the calculation of zero-knowledge proofs to this service. The Filecoin team has also opened up proposals for this service.



Snarks as a Service

Project Description

Filecoin has two core proofs of storage, Proof of Replication and the election-based Proof of Spacetime (PoRep and ePoSt). Each also involve compression using SNARKs into succinct proofs for the chain. SNARKs are generally best processed by GPUs so that these proofs can be submitted within a block time epoch window. Miners not elected to produce proofs in an epoch may also have GPU cycles to spare.

We are seeking proposals for an efficient way to provide Snarks as a Service as a market offering to miners who wish to be less reliant on GPUs.

Preferred applicants should have a good understanding of Filecoin's core proofs, familiarity with SNARKs and OpenCL and evidence of being able to build computation-bound services at scale.

** Note that the Filecoin proofs will be updated shortly. Waiting for these updates to be published before starting research on this is reco… …proofs like PoRep are likely to be updated in the future to ensure network security, for efficiency improvements, and to add features.*

Our Trapdoor Tech team is more interested in this kind of service. The calculation of zero-knowledge proof involves circuit generation, FFT and Multiexp calculation. On the basis of the current official version, there can be a performance improvement of **2.5 to 3 times** .

Snark as a Service, the most basic thing is to figure out the amount of data miners need to transmit to the service. This article analyzes the amount of data in detail. This article involves some professional terms, labeling/column hash/encoding and so on. Those who are not familiar with these terms can take a look at the article on the algorithm of SDR:

Filecoin-Deep understanding of SDR algorithm (http://mp.weixin.qq.com/s?__biz=MzU5MzMxNTk2Nw==&mid=2247486980&idx=1&sn=d525f288bd1305191a7cd7a4d26acb8c&chksm=fe131f14c9649602fdb874443a0b09ead774208e3fa1ca6dc4bb35322390299aa7eeeffe2141&scene=21#wechat_redirect)

Filecoin-PoREP circuit introduction (https://learnblockchain.cn/article/890)

Vanilla Proof is the data structure corresponding to the entire data volume, starting from the analysis of this structure.

01 Overview

Vanilla Proof is the structure of the data required for PoREP proof. The specific definition is in the Proof structure of storage-proofs/src/porep/stacked/vanilla/params.rs.

```
pub struct Proof {
 pub comm_d_proofs: MerkleProof,
 pub comm_r_last_proof: MerkleProof,
 pub replica_column_proofs: ReplicaColumnProof,
 pub labeling_proofs: HashMap>,
 pub encoding_proof: EncodingProof,
}
```

It can be seen that the entire Proof is composed of 2 MerkleProof, 1 ReplicaColumnProof, 1 HashMap of LabelingProof, and 1 EncodingProof. In the entire PoREP calculation, two Hash functions are used: Poseidon and Sha256. The domain size of these two Hash functions is 256bit, which is 32 bytes.

02 MerkleProof

MerkleProof includes the data needed to prove the leaf node on a Merkle tree, including leaf node (leaf), path (path) and tree root (root). The specific definition is in the structure of MerkleProof in storage-proofs/src/merkle.rs.

```
pub struct MerkleProof {
 pub root: H::Domain,
 path: Vec, usize)>,
 leaf: H::Domain,
}
```

Among them, root and leaf are 32 bytes. Path is implemented by a tuple's Vec. Each tuple specifies the information of the sibling node and the position of the sibling node (left or right). For a binary tree, a tuple consists of a sibling node and its position. For an octree, a tuple consists of 7 sibling nodes and positions. U in the MerkleProof structure represents the fork number of the Merkle tree, U2 represents the binary tree, and U8 represents the 8-ary tree.

In other words, the size of a MerkleProof can be calculated by the following formula:

```
MerkleProof = 32 * 2 + (树高-1)* (兄弟节点个数*32 + 4)
```

03 ReplicaColumnProof

The SDR algorithm in PoREP needs to calculate 11 layers of raw data. If the Sector is 32G, each layer is divided into 1^30 nodes. The data of all layers at the same node position form a Column, that is, a column.

All the data that proves all the columns are integrated in the ReplicaColumnProof structure. The specific definition is in the ReplicaColumnProof structure in storage-proofs/src/porep/stacked/vanilla/params.rs.

```
pub struct ReplicaColumnProof {
 pub c_x: ColumnProof,
 pub drg_parents: Vec>,
 pub exp_parents: Vec>,
}
```

Where c_x is the column information of the node position of a certain challenge, drg_parents is the column information corresponding to the node on which the base of the challenge node depends, and exp_parents is the column information corresponding to the node on which the exp of the challenge node depends.

3.1 Column

The information of each column is represented by the Column structure:

```
pub struct Column {
 pub(crate) index: u32,
 pub(crate) rows: Vec,
}
```

The index represents the number of the column, and rows represents the specific information of each row of a certain column.

3.2 ColumnProof

The proof data information of a single column is represented by the ColumnProof structure, which is specifically defined in the ColumnProof structure of storage-proofs/src/porep/stacked/vanilla/column_proof.rs.

```
pub struct ColumnProof {
 pub(crate) column: Column,
 pub(crate) inclusion_proof: MerkleProof,
}
```

Because the last level of node data in a column also constitutes a Merkle tree, the proof information provided in each column will include a MerkleProof information.

Therefore, the ReplicaColumnProof size is calculated as follows:

```
Column = 4 + 32*11 = 356
ColumnProof = Column + MerkleProof
ReplicaColumnProof =  (1+6+8)*ColumnProof
```

04 LabelingProof

The calculation of PoREP's SDR becomes Labeling. The so-called Labeling is to calculate new node information based on the dependent node information of base and exp.

LabelingProof is used to indicate the information required by a node for labeling calculation proof, which is specifically defined in the LabelingProof structure in storage-proofs/src/porep/stacked/vanilla/labeling_proof.rs.

```
pub struct LabelingProof {
 pub(crate) parents: Vec,
 pub(crate) node: u64,
}
```

The node is the node number, and the parents are the dependent node information of the base and exp that the node depends on. In the actual Labeling calculation, the information of the parents will be calculated multiple times, so the Parents in the LabelingProof will also be expanded from 14 node information to 37 node information (base and exp depend on node information replication).

The calculation formula for the length of a single LabelingProof is:

```
LabelingProof = 32 * 37 + 8 = 1192
```

Note that the labeling_proofs in Proof will contain the LabelingProof proof information for each layer.

05 EncodingProof

EncodingProof is the data required for the proof of the last layer of SDR calculation and the original data Encoding. The specific definition is in the EncodingProof structure of storage-proofs/src/porep/stacked/vanilla/encoding_proof.rs.

```
pub struct EncodingProof {
 pub(crate) parents: Vec,
 pub(crate) node: u64,
}
```

Similar to LabelingProof, it is necessary to prove that the calculation result of Encoding is correct, and all the node information that the node depends on is required.

The length calculation formula of a single EncodingProof is:

```
EncodingProof = 32 * 37 + 8 = 1192
```

06 Proof size calculation

Take the 32G Sector as an example, count all the data required for the proof, including: 9 partitions, 11 layers, and 16 challenges. In other words, the data required for the entire proof includes 9*16 = 144 Proofs.

```
comm_d_proof = 32 * 2 + 30 * (32 + 4)= 1144
comm_r_last_proof = 32 * 2 + 10 * (32*7 + 4)= 2344
replica_column_proofs = 15*(356+2344) = 40500
labeling_proofs = 11*1192 = 13112
encoding_proof = 1192
Proof = 1144 + 2344 + 40500 + 13112 + 1192 = 58292
```

The total data is: 144*58292 = 8394048 = **8M** .

to sum up:

Snark as a Service is a more interesting service, which specializes in providing zero-knowledge proof computing services in the Filecoin ecosystem. When the sector size is 32G, the amount of data required to prove is about 8M.

*There are many original and high-quality articles on the public account **star idea** , and you are welcome to scan the code and pay attention.*

This article participates in the DingChain community writing incentive plan (https://learnblockchain.cn/site/coins) , good articles are good for profit, and you are welcome to join as well.

Published on 2020-04-20 11:03 Reading (820) Credits (44)
Category: FileCoin (https://learnblockchain.cn/categories/FileCoin)

0 likes Favorites

Articles you may be interested in

Why is NFT different? How does Filecoin's distributed storage solution empower NFT? (https://learnblockchain.cn/article/2495) 21 views

IPFS Weekly 132 | The next gathering will showcase the cooperation between IPFS and NFT (https://learnblockchain.cn/article/2418) 69 views

IPFS helps expand ETH, Filecoin and DeFi to create the future together, and analyzes the powerful combination of IPFS and ETH (https://learnblockchain.cn/article/2390) 66 views

The Web3.0 China Summit came to a successful conclusion, Hu Feng, COO of Time Cloud: Filecoin needs long-termism! (https://learnblockchain.cn/article/2375) 94 views

Web 3.0 is coming, distributed storage plays an important role|Space Cloud invites you to participate in the Web3.0 China Summit and Distributed Storage Industry Conference (https://learnblockchain.cn/article/2349) 132 views

People's Daily Online: "Distributed storage opens up a market of 100 billion yuan", IPFS welcomes the new digital era! (https://learnblockchain.cn/article/2320) 186 views

Related questions

Which RPC interfaces are the three interface calls in the figure below the filecoin block explorer? ? ? (https://learnblockchain.cn/question/1505) 1 answer

What is the current progress of Filecoin? (https://learnblockchain.cn/question/4) 1 answer

0 comments

Please log in (https://learnblockchain.cn/login) to comment