Filecoin-Sector processing logic changes in testnet3

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

The frequency of Lotus code updates has become faster, and a lot of code merges come in every day. The current zero-knowledge proof CRS has been updated from V20 to V24. At present, the test network has also entered the testnet3 stage. On the basis of the previous version of V20, the V24 version also has some changes to the sector processing.

In the previous article, I tried the WeChat official account, which is very interesting. Most of the paid are good friends I know, mainly for encouragement and support. Thank you friends for your encouragement, and will insist on writing down your understanding in time and sharing.

Some time ago, I saw Fu Sheng's reflection on the cliff. I have a deep feeling. I was on Cheetah Mobile at the time of the 16/17 big event in the article. Cheetah Mobile did predict that tools would be OSized very early, and the future of tool apps was rather confused. In 16 years, tool app guarantees revenue, 17 years in news, AI. The culture of Cheetah Mobile is to run in small steps and make quick trial and error. Indeed, in those few years, it felt that Cheetah Mobile was mainly setting targets and doing tasks. Did not find their own accumulation and barriers. Cheetah Mobile is also very hard. When it is most nervous, it releases updates basically every week. App is as big as Cheetah Mobile, it is not simple, and it also has its own relatively complete development process and system.

However, there is a big problem, which is income. The revenue model mainly depends on advertising. Overseas advertising is indeed easy to get stuck on advertising platforms. Still can't fight back, can only compromise.

A few feelings and perception of future trends, but it is very difficult to change your situation, especially your own DNA. Commercial closed loop is very important. Although a small point cannot form a closed loop at the beginning, it is important to form its own closed loop as it develops.

Anyway, Cheetah Mobile, has made a lot of attempts in the direction of AI and robots. I hope that the old club will get out of the trough and develop smoothly.

The frequency of Lotus code updates has become faster, and a lot of code merges come in every day. The current zero-knowledge proof CRS has been updated from V20 to V24. At present, the test network has also entered the testnet3 stage. On the basis of the previous version of V20, the V24 version also has some changes to the sector processing.

What does Filecoin-Lotus storage prove? (https://learnblockchain.cn/article/681)

This article introduces the logic of the sector processing of testnet3. The related logic is implemented in the rust-fil-proof project. The last submission information of the source code used in this article is as follows:

commit 14870d715f1f6019aba3f72772659e38184378bf (HEAD -> master, origin/master, origin/HEAD)
Author: Rod Vagg
Date: Fri Mar 20 22:30:18 2020 +1100

feat(filecoin-proofs): expose filecoin_proofs::pad_reader

commit 78da3a008a1407654db600e6d5161464a8595e85

01

Sector processing (Precommit) process

The precommit process is divided into two phases, phase1 and phase2, namely phase1 and phase2. The related interface functions are in the seal_pre_commit_phase1 and seal_pre_commit_phase2 functions in the filecoin-proofs/src/api/seal.rs file.

In general, there are two major changes in the logic of the Sector Precommit process: 1) The label encoding algorithm has changed from window SDR to SDR. 2) The process is divided into two stages.

The calculation process of SDR has been introduced in depth in the previous article.

Filecoin-Why is SDR so slow? (http://mp.weixin.qq.com/s?__biz=MzU5MzMxNTk2Nw==&mid=2247486980&idx=1&sn=d525f288bd1305191a7cd7a4d26acb8c&chksm=fe131f14c9649602fdb874443a0b09ead774208e3fa1ca6dc4bb35322390299aa7eeeffe2141&scene=21#wechat_redirect)

I won't talk about it here, but mainly introduce the two-stage related logic of Sector processing.

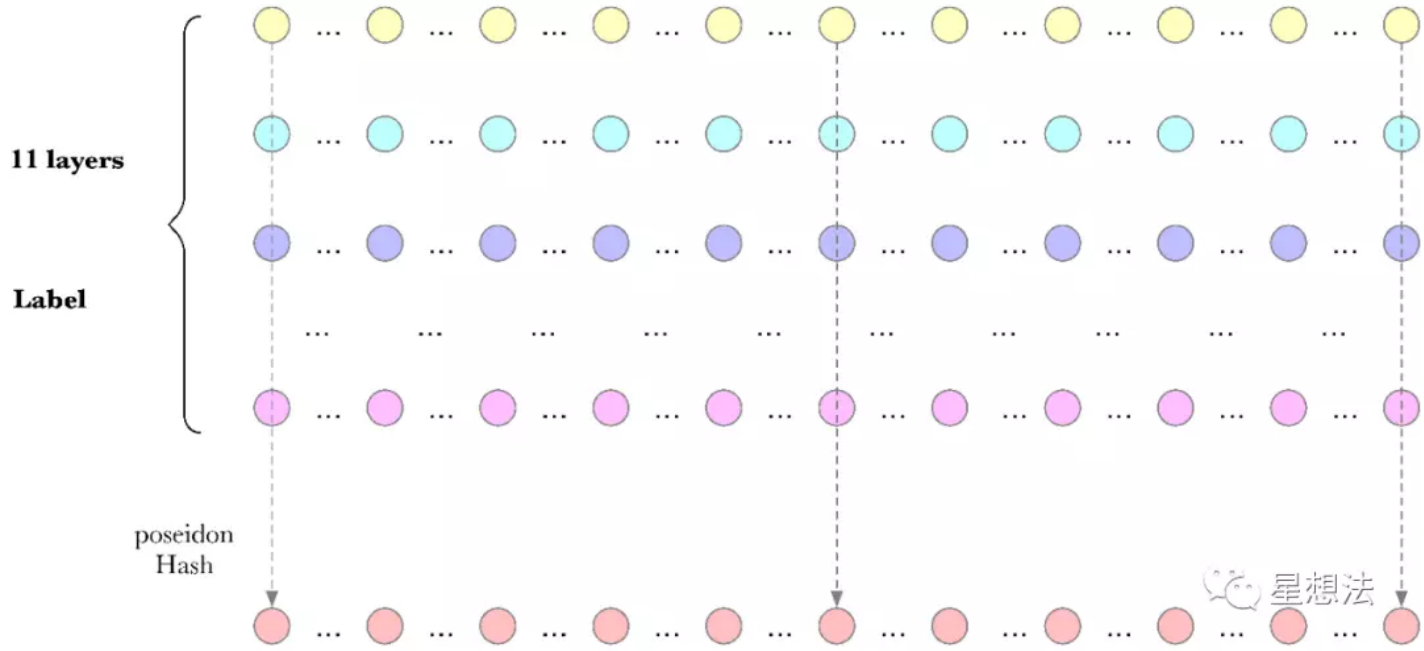
1.1 Precommit Phase1

The process of Phase1 is mainly a two-part calculation: 1) Calculating the merkle tree (binary tree, sha256 hash calculation) of the original data 2) Label, which is the calculation of SDR. The merkle tree of the original data (tree _d), the root of the tree is comm_d.

1.2 Precommit Phase2

The process of Phase2 is mainly a two-part calculation: 1) column hash 2) generate a merkle tree (octree, poseidon hash calculation) for the calculation result of column hash 3) perform an encoding again for the calculation result of the label to generate a merkle tree (Octree, poseidon hash calculation).

The calculation process of **column hash** is as follows:



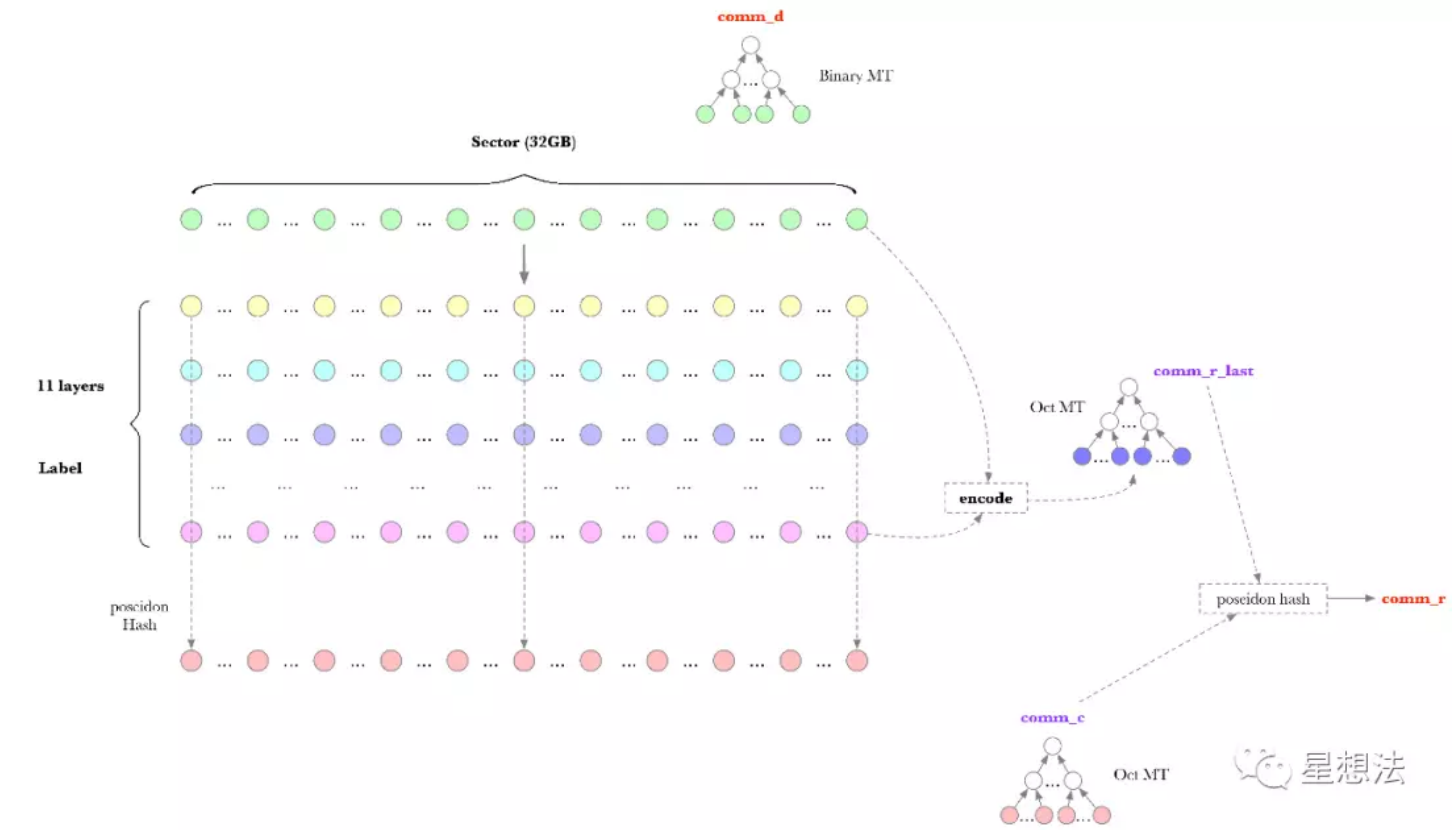
The 32GB Sector is divided into 1G nodes. The calculation of SDR will generate 11 layers of processing data, each layer is 32GB. The node data of the same number in each layer, the result of the hash after being combined is the calculation result of the column hash. The calculation result of Column hash is also 32GB.

According to the calculation result of column hash, an octree (tree_c) is generated, and the root of the tree is comm_c.

The calculation of **label encoding** is to encode the calculation result of SDR and the original data. The so-called encoding is currently the addition of large numbers. As a result of encoding, an octree (tree_r_last) is generated, and the root of the tree is comm_r_last.

There are two data on the chain: comm_d and comm_r. Among them, comm_r is the hash result of posedion of comm_c and comm_r_last.

The entire sector processing logic is summarized as follows:



02

Sector Proof (Commit) Process

The Commit process is divided into two phases, phase1 and phase2, namely phase 1 and phase 2. The sector certification process is closely related to the zero-knowledge certification process. For those who are not familiar with the theory and application of zero-knowledge proof zk-SNARK, you can check the related articles I wrote before. The related interface functions are in the seal_commit_phase1 and seal_commit_phase2 functions in the filecoin-proofs/src/api/seal.rs file.

Phase 1 of the sector certification is mainly to prepare the data needed for the circuit. These data are not completely the public data of the circuit, nor are they completely the private data of the circuit, but the original data required by the circuit data. In stage 1, it will not prove the 1G nodes corresponding to the 32G of Sector, but select some nodes to prove.

All these selected nodes are divided into 9 Partitions:

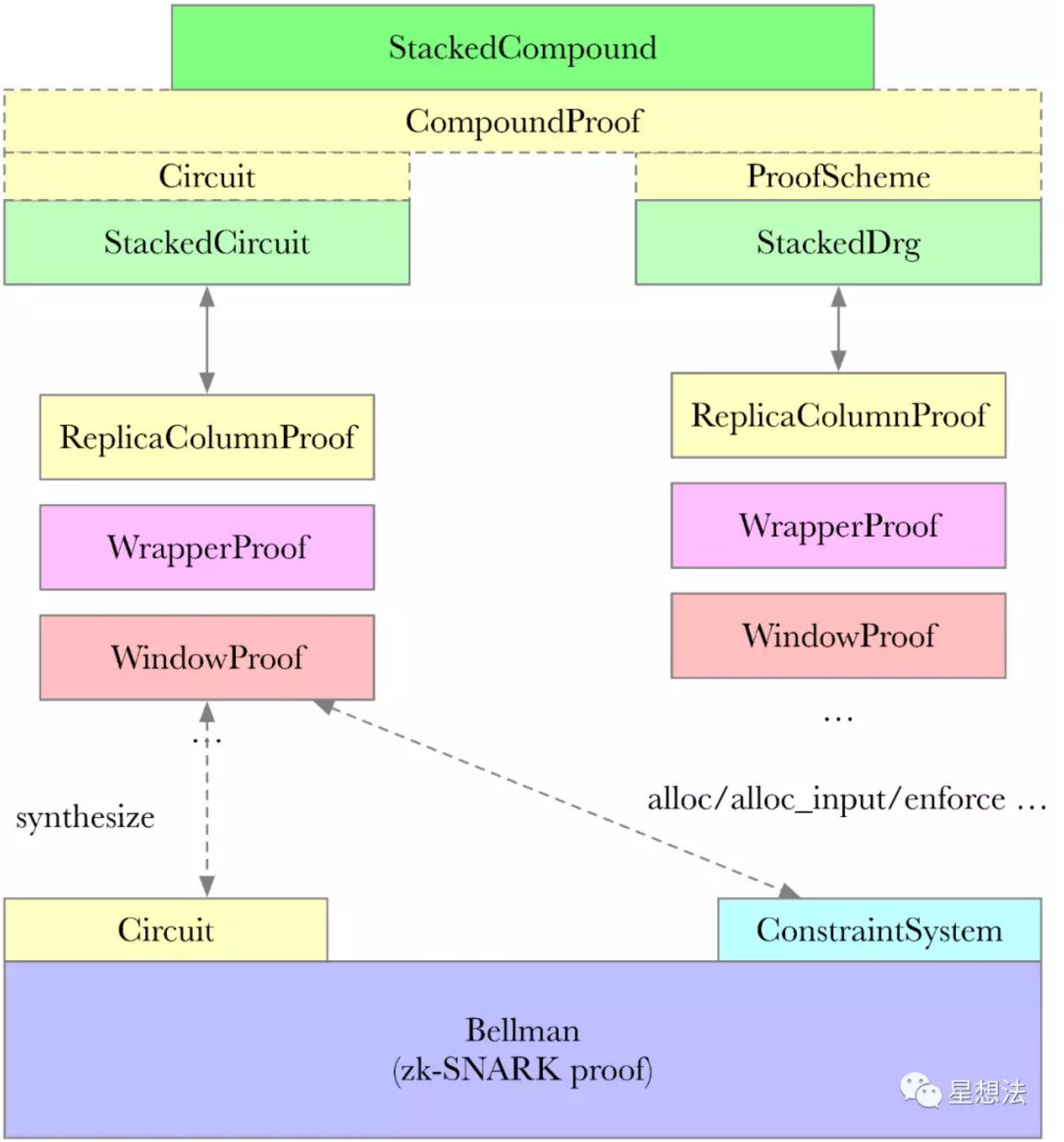
```
pub static ref POREP_PARTITIONS: RwLock>= RwLock::new(
[
(SECTOR_SIZE_2_KIB, 1),
(SECTOR_SIZE_8_MIB, 1),
(SECTOR_SIZE_512_MIB, 1),
(SECTOR_SIZE_32_GIB, 9)
]
```

Each Partition randomly selects some nodes. The number of selected nodes does not exceed the minimum challenge node number:

```
pub static ref POREP_MINIMUM_CHALLENGES: RwLock>= RwLock::new(
[
(SECTOR_SIZE_2_KIB, 2),
(SECTOR_SIZE_8_MIB, 2),
(SECTOR_SIZE_512_MIB, 2),
(SECTOR_SIZE_32_GIB, 138)
]
```

In other words, for the 32G Sector, there are 9 Parititons, and each Partition randomly selects 16 nodes to challenge. The specific algorithm randomly selected is in the derive_internal function of storage-proofs/src/porep/stacked/vanilla/challenges.rs.

Phase 2 of the sector proof is the circuit processing of the zero-knowledge proof and the process of generating the zero-knowledge proof. The logic of this part, the overall framework is the same as before, mainly due to some changes in the SDR algorithm changes. RUST-FIL-PROOF (FPS) implements StackedCompound, which is specifically used to implement Stacked DRG data processing certification. StackedCompound integrates two parts together, one part is the circuit (Stacked Circuit), and the other part is the Stacked Drg, which realizes the preparation of circuit data. These parts are divided into sub-functions (Window, Wrapper, ReplicaColumn, etc.). When calling Bellman to generate the proof, the synthesize interface of the corresponding circuit will be called to complete the process of generating R1CS for the entire circuit.



to sum up:

The source code of Lotus is updated frequently. Testnet3 divides Sector's Precommit and Commit processing into two stages (Phase1 and Phase2). SDR is that the algorithm change is the biggest change. The CRS of zero-knowledge proof has been updated to version V24.