Canyon: Permanent Storage Layer for Limitless Scalability

Liu-Cheng Xu

Abstract

Canyon is a permanent storage network built on Substrate, which records the hashes of files on chain and stores the files off-chain. By blending PoS and a probabilistic proof-of-storage scheme inspired by Arweave, Canyon greatly reduces the barriers to entry for miners, incentivizing them to store as much data as possible for the block rewards.

Contents

1 Introduction

1.1 Motivation

TODO

2 Background

2.1 Filecoin

Filecoin[1] proposes a sophisticated cryptographic solution based on zero-knowledge proofs (ZKPs) to prevent the common attacks on the decentralized storage verification, which uses pure mathematical methods and is able to achieve high security guarantees. The whole mining process consumes too much computing power, making the actual storage cost prohibitively expensive. Furthermore, the high hardware requirements largely raise the threshold for small miners, preventing these with pure common commodity hardwares from joining the network, the storage distribution today becomes centralized.

Due to the lack of authentic storage needs and system design of Filecoin, the miners themselves are economically impelled to store tons of garbage data in the network, increasing the storage power and earning more mining rewards. Despite the Filecoin team has proposed the off-chain governance approach like Filecoin Plus[1], it does not mitigate this problem substantially, leaving the promotion of the useful storage still a huge unresolved challenge in the Filecoin network.

2.2 Crust

To solve the problem of decentralized storage verification, Crust[2] adopts a hardware-based solution Trusted Environment Execution (TEE) to make sure the miners store a specific number of data copies as promised. Each of the storage nodes is required to enroll on Crust chain through TEE before it's allowed to deal with the storage orders from client. The TEE module of the nodes will periodly check and report whether the files are properly stored in the local storage space in a trusted way.

There are three major TEE providers with different implementations: Software Guard Extensions(SGX) on the Intel platform, Secure Encrypted Virtualization (SEV) on the AMD platform, and TrustZone on the ARM platform. The SGX of Intel is the mostly widely used TEE platform. Although Crust has a relatively low hardware demand for the mining compared with Filecoin, the miners are heavily dependent on a fairly narrow range of of hardwares that support TEE, thus being greatly affected by the hardware manufacturers.

2.3 Arweave

Unlike the ephemeral storage services such as Filecoin, Crust, Arweave[3] serves as a permanent storage layer using a probabilistic and incentive-driven approach to maximize the number of redundant copies of any individual piece of data in the network, which fills in a crucial aspect of decentralized storage need in Web 3.0. The natual feature of perpetual storage is perfectly suitable for the NFTs, and is the cornerstone of new storage-based computation paradigm like everFinance [2].

Without the periodical audit of the data replications, Arweave is much effortless than the other solutions in terms of the constraint and cost of storage consensus. Nevertheless, the deficiency

[1] https://docs.filecoin.io/store/filecoin-plus

[2] https://medium.com/everfinance/a-storage-based-computation-paradigm-enabled-by-arweave-de799ae8c424

of this scheme is the potential risk of data centralization that ultimately there possibly will be a single storage provider serves the whole data. The Arweave team has introduced and delopyed an improved version of consensus Succinct Proofs of Random Access, attempting to disincentivize miners from retrieving data on demand from the network.

3 Design purposes

3.1 Security and privacy

4 System design

4.1 Consensus

Canyon network is profoundly inspired by Arweave, especially the storage consensus, aiming to be a permenant decentrialized storage network using Substrate framework. The key contribution of Canyon is that it adapts PoS into the storage consensus of Arweave, whereas Arweave uses PoW, making Canyon a more scalable and environment-friendly network.

In order to mine a block, a legitimate POA, which proves the block author has the access to the data of a random historical block, is required to be included in the block header. According to the computed result of POA, we can estimate the proportion of data stored locally by a node in the network-wide data. Based on this, we link the PoS rewards to the estimated storage capacity of a node. The more data a node stores, the more rewards it can receive. Besides, as the blocks mined by the node increase, the estimation is getting closer to the actual value.

4.1.1 Proof of Access

$$P(\text{win}) = P(\text{has recall block}) * P(\text{finds hash first}) \tag{1}$$

$$P(\text{win}) = P(\text{has recall block}) * P(\text{claims slot}) \tag{2}$$

Algorithm 1: Generation of POA

```
Input  :
         The random seed $S$;
         The weave size $W$;
Output:
         The proof of accesing the recall block $POA$;
Initialize the number of repeats $x$ with 1;
repeat
    Draw a random byte $B$ with $\textsc{MultiHash}(S, x) \mod W$;
    Find the $TX$ in which the random byte $B$ is included;
    $x \leftarrow x + 1$;
until The data of TX is available;
$POA \leftarrow \textsc{Construct}\text{POA}(TX)$;
return $POA$;
```

The steps are as follows:

1. Input the value of `parent_hash` and `weave_size` respectively. And depth is initialized to 1.
2. Compute `recall_byte`, i.e. a random byte in the network-wide data history that ranges from 0 to `weave_size`, excluding `weave_size`. For the RecallByte formula, please see AR source code.
3. Find out the random byte belongs to which `recall_tx` included in which block given `recall_byte`.
4. If you have stored the data of `recall_tx` locally, extract the original data of `recall_tx` and split the data into a Chunk (256KB) list, then create a `poa { data_root, tx_path, data_path, Chunk }` .
5. If the node does not have the data of `recall_tx` locally, increase the value of `depth` by 1 and repeat the step 2 to 4.
6. Return poa.

As you can see, if a node stores 100% of data, the value of 'depth' always remains 1. The node only needs to go through the above steps once every time when it attempts to generate a 'poa' proof. If the node stores 50% of data, the value of 'depth' is expected to approach 2 as it produces enough blocks. If the node stores 10% of data, the value of `depth` approaches 10.

If a node produces a total of N blocks, and the poa.depth of each block is $depth_{i \in 1,2,...,N}$, the average 'depth' value of the N blocks $\hat{x}$ is:

$$\hat{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

If the node produces enough blocks (i.e. $N \to \infty$) , we can calculate what proportion of data stored locally by the node in the network-wide data more precisely with the value of $\hat{x}_{N\to\infty}$. The formula is as follows:

$$R = \frac{1}{\hat{x}_{N\to\infty}}$$

With the knowledge of the storage capacity of nodes R, we can incentivize miners to store more users' data by allocating PoS rewards to them accordingly.

4.1.2 Proof of Stake

4.1.2.1 Staking Rewards

4.1.2.2 Stake

4.2 Economy Model

4.2.1 Transaction Fee

The fee of data storage transaction is composed of the perpetual storage cost and one-shot bandwidth cost. The perpetual storage cost is basically modeled after Arweave.

4.2.2 Data Oblivion

TODO

4.2.3 Payments

Canyon has an inherent advantage of easier interoperablity with the Polkadot ecosystem by using Substrate, which is the same framework Polkadot is built on. Canyon will support the users pay the storage fee using the stable coins like USDT.

5 Roadmap

TODO

6 Conclusion

6.1 Future work

References

[1] Protocol Labs, Filecoin: A Decentralized Storage Network, `https://filecoin.io/filecoin.pdf`.

[2] `https://crust.network/`

[3] Sam Williams, Viktor Diordiiev, Lev Berman, India Raybould, Ivan Uemlianin. Arweave: A Protocol for Economically Sustainable Information Permanence.

[4] Gavid wood. Polkadot: Vision for A Heterogeneous Multi-chain Framework. `https://polkadot.network/PolkaDotPaper.pdf`

[5] `https://github.com/paritytech/substrate`

[6] Vitalik buterin. An Incomplete Guide to Rollups, `https://vitalik.ca/general/2021/01/05/rollup.html`

[7] A Storage-based Computation Paradigm Enabled by Arweave, `https://medium.com/everfinance/a-storage-based-computation-paradigm-enabled-by-arweave-de799ae8c424`

[8] `https://ever.finance/`

A Appendix