Scaling Decentralized Finance

Robie Gonzales

ABSTRACT: Decentralized finance has emerged as the most popular use case for blockchain technology. Smart contract compatible blockchains such as Ethereum have enabled the offering of complex financial services in decentralized form. As a result, there has been an increasing number of users driving up the cost of using the network. For blockchain ecosystems to scale with this growth of activity, there needs to be changes in the underlying architecture. This paper will explore the scaling solutions being researched, which can be classified as Layer 1 and Layer 2 solutions.

I. INTRODUCTION

Blockchains offer a mechanism through which multiple, possibly mistrusting entities can cooperate in the absence of a trusted third party. With the development of smart contract compatible blockchains such as Ethereum [1], many complex use cases, particularly in the area of financial services, are being realized. However, the permissionless nature of these blockchains limit their scalability. Vitalik Buterin, founder of Ethereum [1], presents this problem as the **blockchain trilemma**, which is similar to CAP theory in traditional distributed systems [2]. The trilemma states that regarding scalability, security and decentralization, any improvement in one of these aspects negatively impacts at least one of the other two [3]. As a result, there has been extensive research into scaling solutions. While many of these solutions revolve around transactions directly with the blockchain (Layer 1 and Layer 2), there have been developments in scaling the smart contract layer through novel implementations of market makers. This paper will explore areas of decentralized finance, why scalability is needed and current research resulting from it.

II. DECENTRALIZED FINANCE

Decentralized finance (DeFi) has quickly evolved from just simple cryptocurrencies. Smart contracts, a set of rules specified in computer program form that is then stored on the blockchain, effectively act as **protocols** allowing

for independent entities to communicate with each other. As a result, many DeFi protocols have emerged offering traditional financial services such as exchanges, lending & borrowing and derivatives.

A. DECENTRALIZED EXCHANGES

Decentralized exchanges (DEXs) account for a majority of all DeFi activity [4]. Centralized exchanges (CEXs) often utilize a traditional order book model, which relies on market markers to provide liquidity and depth to the market [5]. Market makers are entities that are always willing to buy or sell an asset, allowing trades to be executed seamlessly. Instead, DEXs leverage an **automated market maker** (AMM) to determine asset pricing in a **liquidity pool**.

Liquidity pools are smart contracts that lock in the reserves of two or more tokens [5]. AMMs then use a pricing algorithm where the inputs are the quantities of the tokens within a pool. This allows liquidity takers to directly trade against the pool offering benefits such as instantaneous settlement, immediate access to liquidity and lower barriers of entry.

B. DERIVATIVES

The revolutionary mechanism of AMMs and liquidity pools have enabled the creation of innovative financial products. Of these, **perpetual contracts** have become the most popular DeFi derivative product. Perpetual contracts are essentially futures contracts without an expiry date, allowing them to be held or traded for an indefinite amount of time [6]. As a result, they are a popular option for high frequency trading.

AMMs are a smart contract layer solution to scalability concerns in the DeFi space. They remove the need for **liquidity providers** to constantly monitor the market and adjust bid and ask prices which reduces the load on the network. Furthermore, there has been research into more advanced types of AMMs such as a **time-weighted average market maker** (TWAMM) [7] which is optimized for large orders. However, for DeFi activity to reach traditional finance levels, there needs to be changes in the fundamental architecture.

III. SCALING SOLUTIONS

With the tremendous growth DeFi has seen in recent years [4], the Ethereum blockchain has reached certain capacity limitations. This has driven up the cost of using the network and has resulted in record high gas fees [8]. As explored earlier, while some solutions have been made directly on the smart

contract layer, they do not address the underlying issues associated with the blockchain architecture. Since every node in the Ethereum network stores the entire state and processes all transactions, the blockchain is bottlenecked by any single node. As a result, Bitcoin can only process 3-7 transactions per second [9] while Ethereum can process 7-15 transactions per second [9]. Compared to Visa's 65000 transactions per second [10], it is clear there needs to be scalable solutions. The main goals of blockchain scalability are to increase transaction speed (confirmation finality) and transaction throughput (transactions per second) [11]. Scaling solutions can be categorized into Layer 1 (on-chain) or Layer 2 (off-chain).

A. LAYER 1 SCALING

Layer 1 scaling revolves around the on-chain design of the blockchain itself, including the structure of blocks, consensus algorithms and the structure of the main network. While some blockchains like Solana [12] attempt to scale by using alternative consensus algorithms (Proof of History), Ethereum's main Layer 1 solution is *sharding*.

Sharding is the process of splitting a database horizontally to spread the load. In the context of blockchains, sharding is the process of creating new chains where each chain is only responsible for its own subset of transactions. Whenever a sharded chain verifies a new block, they publish a signature attesting to the fact they did so. Other sharded chains now only have to verify the signature rather than the transactions within that new block. More formally, we can represent the computational capacity of a single node as $O(C)$. A traditional blockchain can process blocks of size $O(C)$ whereas a sharded chain can process $O(C)$ blocks **in parallel**. Furthermore, since each sharded chain is responsible for a fixed subset of signatures, the cost to each node to verify a block is $O(1)$. Since each block has $O(C)$ capacity, the sharded chain's total capacity is $O(C^2)$ which results in **quadratic sharding** [3].

B. LAYER 2 SCALING

Instead of putting all transaction activity on the blockchain directly (Layer 1), Layer 2 scaling involves moving the bulk of activity off-chain into a Layer 2 protocol. The three major types of Layer 2 scaling are state channels, Plasma and *rollups* [13]. Rollups and state channels interact directly with the main blockchain and thus, receive all the security benefits from a Layer 1 solution. Conversely, plasma chains and sidechains involve the creation of new, independent chains

with their own consensus mechanisms which provide alternative security benefits for various use cases.

1. ROLLUPS

Rollups are solutions that move computation off-chain, but post some transaction data on-chain. Since some transaction data remains on Layer 1, rollups inherit all the security properties of the main chain while benefiting from the reduced load of expensive computations. They accomplish this by having a smart contract on-chain (Layer 1) that maintains a **state root**, which is the Merkle root of the state of the rollup [14]. When **batches** of off-chain transactions are posted back to Layer 1, the state root is updated. The mechanisms for verifying that the post-state roots in the batches are correct can be classified into *ZK-rollups* and *optimistic rollups.*

a. ZK-ROLLUPS

Zero-knowledge rollups bundle (rollup) off-chain computations and generate a cryptographic proof called a **SNARK** (succinct non-interactive argument of knowledge), also known as a **validity proof** [14]. This proof is then posted on Layer 1 and is required for any state changes on Layer 2. This means that ZK-rollups only need to verify the ZK-SNARK rather than all of the transaction data. This provides many scalability benefits as redundant data is no longer required to validate transactions.

ZK-SNARKS leverage some very powerful cryptographic primitives, particularly those of polynomials. Since polynomials are a single mathematical object that can contain an unbounded amount of information, a single equation between polynomials can represent an unbounded number of equations between numbers [15]. For example, consider the equation $A(x) + B(x) = C(x)$. If this equation is true, then it is also true that $A(0) + B(0) = C(0)$, $A(1) + B(1) = C(1)$, $A(2) + B(2) = C(2)$, etc. In other words, it is possible to encode a large number of computations into a single polynomial expression, which when verified implicitly verifies all of the equations simultaneously. The verification of these polynomial equations is further optimized by using a special type of polynomial hash called a **polynomial commitment**, which allows efficient verification even if the underlying polynomials are very large.

b. OPTIMISTIC ROLLUPS

Optimistic rollups are “optimistic” in the sense that they assume all transactions are valid by default and only run computations when there is a dispute. If one were to discover an invalid batch, they could publish a **fraud proof** onto Layer 1, which when verified by the on-chain smart contract will revert that batch and all batches after it.

Fraud proofs are a system where to accept a batch, a staked deposit must be signed. Disputers must also stake a deposit and whoever of the two parties are wrong loses their deposit. While computation based on fraud proofs are scalable, fraud proofs cannot be used to verify availability of data unlike ZK-SNARKS. However, this can be solved by **data availability sampling**, which uses ZK-SNARKS and other primitives to guarantee the presence of data [14].

IV. CONCLUSION

Scalability remains a problem in decentralized finance. As the number of users continue to rise and types of products continue to expand, so does the need for a scalable infrastructure. The two main scaling solutions can be classified as Layer 1 and Layer 2. Layer 1 revolves around the on-chain design of transactions with sharding being the main method of doing so. Layer 2 is the idea of moving transactions off-chain with rollups being the primary mechanism. Rollups can further be classified as ZK-rollups or optimistic rollups. As the DeFi and blockchain ecosystem continue to evolve, there exists no single scaling solution to rule them all. Multiple scaling solutions are needed to fulfill the vision of decentralized finance.

REFERENCES

[1] Buterin, V. (2014). *A next-generation smart contract and decentralized application platform.* https://ethereum.org/en/whitepaper/

[2] IBM. (2021). *CAP Theorem.* https://www.ibm.com/topics/cap-theorem

[3] Buterin, V. (2021). *Why sharding is great: demystifying the technical properties.* https://vitalik.ca/general/2021/04/07/sharding.html

[4] DefiLlama. (2021). DefiLlama. DeFi Dashboard. https://defillama.com

[5] Zhou, L., Qin, K., Torres, C., Le, D., Gervais, A. (2020). *High-Frequency Trading on Decentralized On-Chain Exchanges*. arXiv:2009.14021

[6] BitMEX. (2021). *Perpetual Contracts Guide.* https://www.bitmex.com/app/perpetualContractsGuide

[7] Paradigm. (2021). *TWAMM.* https://www.paradigm.xyz/2021/07/twamm/

[8] Etherscan. (2021). *Ethereum Blockchain Explorer.* https://etherscan.io/gastracker
[9] Ethereum Wiki. (2020). *On sharding blockchains FAQs*. https://eth.wiki/sharding/Sharding-FAQs
[10] Visa. (2021). *VisaNet: Global Electronic Payment Network.* https://www.visa.ca/en_CA/about-visa/visanet.html
[11] Monte, G. Pennino, D. Pizzonia, M. (2020). *Scaling Blockchains Without Giving up Decentralization and Security.* https://doi.org/10.1145/3410699.3413800
[12] Yakovenko, A. (2017). *Solana: A new architecture for a high performance blockchain*. https://solana.com/solana-whitepaper.pdf
[13] Ethereum.org. (2021). *Scaling*. https://ethereum.org/en/developers/docs/scaling/
[14] Buterin, V. (2021). *An Incomplete Guide to Rollups*. https://vitalik.ca/general/2021/01/05/rollup.html
[15] Buterin, V. (2021). *An approximate introduction to how zk-SNARKs are possible*. https://vitalik.ca/general/2021/01/26/snarks.html