On Scaling Ethereum Transactions and Other Blockchain Based Cryptocurrencies

Rajkumar Kamala Prasad Singh
Department of Computer Science and Engineering
Mumbai University, India

Abstract— The increasing of blockchain-based transactions has made cryptocurrencies scalability a primary and urgent concern. Here we have tried to analyze how the fundamental and circumstantial bottlenecks in Ethereum limits the ability of its current peer-to-peer overlay network to support substantially higher throughputs and lower latencies. Our results suggest that reparameterization of block size is only first increment toward achieving upscaling, high-load blockchain protocols, and major advances will additionally require a basic rethinking of technical approaches. We offer a structured perspective on the design space for such approaches. Within this perspective, we enumerate and briefly discuss a number of recently proposed protocol ideas and offer several new ideas.

Key words: Ethereum, Blockchain

I. INTRODUCTION

Increasing adoption of cryptocurrencies has raised concerns about their ability to scale. Since Ehereum is a self-regulating system that works by discovering blocks at approximate intervals, its highest transaction throughput is effectively capped at maximum block size divided by block interval ever-increasing block sizes on Ethereum portends a potential problem where the system will reach its maximum capacity to clear transactions, probably by 2018-2019.

As a result, there has been discussing techniques for improving the scalability of blockchains in general, and Ethereum in particular. Today's representative blockchain such as Ethereum takes 10 min or longer to confirm transactions, achieves 15 transactions/sec maximum throughput. In comparison, a mainstream payment processor such as Visa credit card confirms a transaction within seconds, and processes 2000 transactions/sec on average, with a peak rate of 56,000 transactions/sec. Clearly, a large gap exists between where Ethereum is today and the scalability of a mainstream payment processor.

Therefore, the key questions are,

Can decentralized blockchains be scaled up to match the performance of a mainstream payment processor? What does it take to get there?

This paper aims to place exploration of blockchain scalability on offer three contributions that illuminate the problem of scaling Ethereum and blockchains generally to achieve high-performance, decentralized systems:

Here we present experimental measurements of a range of metrics that characterize the resource costs and performance of today's operational Ethereum network.

As a first step to-ward better scalability in Ethereum, the community has put forth various proposals to modify the key system parameters of block size and block interval. Our results hinge on the key metric of effective throughput in the overlay network. If the transaction rate exceeds the 90% effective throughput, then 10% of the nodes in the network would be unable to keep up, potentially resulting in denied services to users and reducing the network's effective mining power. To ensure at least 90% of the nodes in the current overlay network have sufficient throughput, we offer the following two guidelines:–

1) [Throughput limit.] The block size should not exceed 4MB, given today's10 min. average block interval (or a reduction in block interval time. A 4MBblock size corresponds to a maximum throughput of at most 27 transactions/sec.
2) [Latency limit.] The block interval should not be smaller than 12s, if full utilization of the network's bandwidth is to be achived.

II. ETHEREUM SCALABILITY TODAY: A REALITY CHECK

We analyze some of the key metrics of the Ethereum system as it exists today.

A. Maximum throughput.

The maximum throughput is the maximum rate at which the blockchain can confirm transactions. Today, Ethereum's maximum through-put is 3.3–7 transactions/sec [1]. This number is constrained by the maximum block size and the inter-block time.

B. Latency. Time for a transaction to confirm.

A transaction is considered confirmed when it is included in a block, roughly 10 minutes in expectation. Bootstrap time. The time it takes a new node to download and process the history necessary to validate the current system state. Presently in Ethereum, the bootstrap time is linear in the size of the blockchain history, and is roughly four days (averaged over five fresh t2.medium Amazon EC2 nodes that we connected to the network running the most recent master software).Cost per Confirmed Transaction (CPCT). The cost in USD of resources consumed by the entire Ethereum system to confirm a single transaction. The CPCT encompasses several distinct resources, all of which can be further decomposed into operational costs (mainly electricity) and capital equipment costs:

1) Mining:

Expended by miners in generating the proof of work for each block. Although we define latency in Ethereum as the time to obtain a single confirmation, some payment processors accept "zero-confirmation" transactions, while others follow common advice to wait for 6 confirmations before accepting a payment.

2) Transaction validation:

The cost of computation necessary to validate that a transaction can spend the outputs referenced by its inputs, dominated by cryptographic verifications.

3) Bandwidth:

The cost of network resources required to receive and transmit transactions, blocks, and metadata.

4) Storage:

The cost (1) of storing all currently spendable transactions, which is necessary for miners and full nodes to perform transaction validation, and of storing the blockchain's (much larger) historical data, which is necessary to bootstrap new nodes that join the network.

Table presents our estimates of these various costs. As the table shows, the majority of the cost is attributable to mining.

	At max throughput at defacto throughput c.p.c percentage	c.p.c percentage	
Mining proof-of-work	~Rs.55.40 - Rs.117.73	~56% ~249.30	~56%
Mining Hardware	~Rs.41.55 - Rs.90.03	~42% ~186.98	~42%
Transaction Validations	~Rs.0.14	~0.2% ~0.55	~0.2%
Bandwidth	~Rs.1.30	~2% ~5.54	~2%
Storage(Running Cost)	~Rs.0.055/ 5 Years		

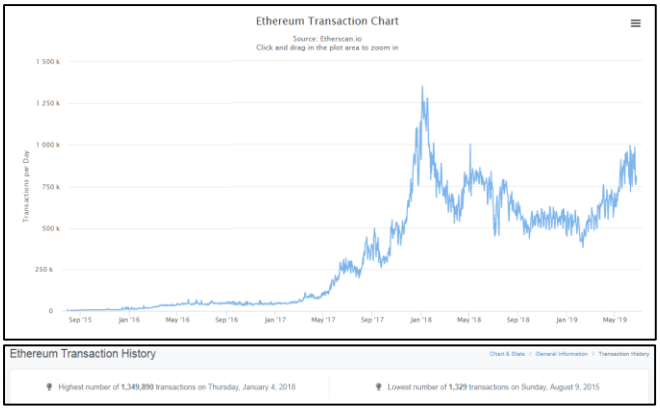
Table 1: Ethereum Cost Breakdown. Includes cost incurred by all nodes.

Our calculation suggests that, at the maximum throughput, the cost per confirmed transaction is Rs.96.95 – Rs.200.83,where 57% is electricity consumption for mining. If the de facto Ethereum through-put is assumed, the CPCT is as high as Rs.429. We proceed to explain our cost-estimation methodology. To measure the cost per transaction for Ethereum, we perform a back-of-the-envelope calculation by summing up the electricity consumed by the network as a whole, as well as the hardware's based on the AntMiner S5+ mining hardware, which is the currently available hardware that has the highest hash rate per joule, and the highest hash rate per rupees according to this comparison as of Aprill 2019. We assume a 1-year eff ective lifetime for the hardware, and that the average hashing rate of the network is 450,000,000 GH/s based on statistics from Aprill 2019.

Based on the power consumption of the selected hardware (0.445 W/GH), the total power consumed by the network will be about 200 MegaWatt. Furthermore,we assume the average price per KWh is Rs.6.93.There are two interesting scenarios: The first scenario is when the Ethereum network is operating at maximum throughput, namely 13.14 transactions/sec. This maximum throughput is mainly constrained by Ethereum's 1MB maximum block size and the variable transaction size. The lower bound of the maximum throughput is inferred from the current average transaction size, about 500 bytes, while the upper bound is based on an oft-cited estimate from which corresponds to unusually small (250 byte) transactions. The second scenario is the defacto average throughput for the Ethereum network, which is, based on statistics collected in Aprill 2019, 1.57 transactions/sec.

We note that it is a fallacy to assume that transaction costs necessarily have to be off set by transaction fees. In particular, the operational costs of running full nodes may be off set by financial externalities, such as being able to confirm one's own transactions without trusting third parties, or by network eff ects, such as selling items whose costs factor in the cost of operating a node. Miners, however, are bereft of these two factors and need to be compensated in the steady state, especially as the block subsidy is reduced over time.

III. Ethereum Transaction Chart:



IV. Rethinking the Design of a Scalable Blockchain

We organize our discussion around a decomposition of the Ethereum system into a set of abstraction layers that we call planes. Ordered in a hierarchy of dependency from bottom to top, the five planes we consider are the Consensus, Sharding, Side Planes, Lightning, SegWit, Plasma.

A. Consensus Plane

The function of the Consensus Plane is to designate a globally accepted set of transactions for processing, as well as a total or partial order on these transactions. As a general abstraction, this plane ingests messages from the Network Plane and outputs transactions for insertion into the system ledger. In Ethereum, the Consensus Plane is the functionality that mines blocks and reaches consensus on their integration into the blockchain. Improving proof-of-work protocols. Ethereum's blockchain protocol introduces a trade off among consensus speed, bandwidth, and security. By improving the former two, one introduces an increased number of forks, leading to a loss of the mining power that secures the system and to reduced fairness. Many cryptocurrencies favor consensus speed over security, employing a standard Ethereum blockchain with a high block-generation frequency. This three-way tradeoff , however, is not inherent in decentralized cryptocurrencies.

The GHOST protocol of Sompolinsky et al. as well as Lewenber get demonstrate that fairness and mining power utilization can be improved by changing the chain selection rule, in particular, by being inclusive to forks outside the main chain as well. In more recent work, Ethereum -NG demonstrates that the inherent trade off s in Ethereum can be eliminated with an alternative blockchain protocol, off ering a consensus delay and bandwidth limited only by the Network Plane.

1) Proof of stake.

Various proposals use proof of stake to achieve consensus, eliminating the computational expense of proofs of work. In proof of stake, principals gain the right to create blocks by depositing funds they own. These techniques, however, lack formal guarantees of system convergence.

Note: In proof of stake, a validator can validate the transactions based on the amount of crypto coins he/she holds.

B. Consortium consensus.

Decentralization carries a performance cost. A trust model with stronger assumptions than those in Ethereum can support a more efficient consensus protocol, achieving better latency and throughput with less computation, bandwidth, and storage. Specifically, using a standard Byzantine Fault Tolerant (BFT) replication protocol with a small number of pre-designated trusted entities removes many of the scaling obstacles in Ethereum. Settings involving BFT protocols executed by small sets of trusted entities have received little treatment in the academic literature, but are of consider able interest in practice, and mainstream financial institutions are actively exploring their use [45]. They are sometimes referred to as "consortium blockchains."Consortium blockchains are worth investigation both as an alternative to decentralized cryptocurrencies and to characterize the performance cost that decentralized blockchains incur by distributing trust. In the full version of this paper, we present performance figures and micro-benchmarking results on experiments with a popular BFT protocol (PBFT) across a range of diff erent system parametrizations and with nodes dispersed across eight geographies worldwide.

Our results illustrate the attractiveness of BFT as a basis for the consensus layer in a cryptocurrency (given acceptance of its strong trust assumptions. Even with dozens of nodes, PBFT greatly outperforms Ethereum in both transaction latency and throughput. For example, 64 nodes processing batches of 8192 transactions can achieve a throughput of 4.5K tx/sec., average trans-action latency of 1.79 sec., and an estimated resource cost per transaction of just $3.95 ×10−7.2Scaling to hundreds of nodes, however, would greatly de-grade the performance of the system. As we now explain, a promising approach to scaling and an open research direction is how to shard a BFT protocol.

C. Sharding.

One possible technique for improving the scalability of the Consensus Plane is to shard it, that is, split up the task of consensus among concurrently operating sets of nodes, with the aim of improving throughput and reducing per-node processing and storage requirements. Sharding is commonly employed in distributed databases, such as Dynamo, MongoDB, MySQL, and BigTable, although performance typically does not grow linearly with shard count. This is due to the need to reach consensus among the shards when operations span multiple shards. One possibility, explored in a non-Byzantine environment in past work, is to use a separate consensus protocol, such as Paxos, to achieve agreement among the shards. Such schemes, however, can incur substantial overhead when cross-shard coordination is required in a Byzantine setting, so sharding protocols for blockchains are an open area of research.2Transactions in our experiments are 190 bytes long, all that is needed for a basic money transfer; given the roughly 500 byte average size of Ethereum transactions, the system would achieve 1.7k tx/sec.

D. Side Plane:

Much as side chains allow off -the-main-chain consensus, we can consider off -chain functionalities. Off -chain transactions have been demonstrated in payment networks, in which payments are routed along paths of pre-established "collateral" channels. Each such channel represents a quantity of ethereum reserves set aside, such that parties can repeatedly adjust their relative stake by exchanging out-of-band messages until the channel is finalized (and the reserves paid out). While payment networks have been heralded as a solution to Ethereum's inherent limitations, much of their operation, and the guarantees they can off er rely critically on the nature of the links formed between parties. Even when payment networks use the same underlying transaction format as Ethereum, as do the Lightning Network [42] and full duplex channels [22], they essentially form a separate an independent, peer-to-peer Consensus Plane, backed by Ethereum. As a result, their capacity, ability to find routes, achieved throughput, latency, and privacy guarantees depend fundamentally one merge properties of the payment network graph, such as the value capacity of peer-to-peer channels, the discoverability of routes, the online status of nodes involved, and so on. Further, payment channels may embody a similar trade off between performance and centralization in the payment network; a centralized hub-and-spoke topology that simplifies routing embodies inherent problems with centralization, such as loss of privacy. The design of protocols for efficient, scal-able, privacy-preserving payment networks is an ongoing area of research: it is far from a given that they can outperform Ethereum's Network and Consensus layers overall.

E. Lightning

The Lightning Network enables users to enter into payment channels outside of the main Ethereum blockchain and transact cheaply. Ethereum's state channels are very similar. State channels allow users to transact state updates outside of the main Ethereum blockchain. It's important to note that both the Lightning Network and State Channels give users the same level of finality they would get by transacting on the core blockchain. This is because users have to store cryptographic messages offline as a way to prove that the final balance is accurate. Because the transactions are just between me and you and don't need to be broadcast to the whole network, they are almost instantaneous. And because there are no miners that need incentivizing, transaction fees are low or even non-existent.

1) How it works

First, two parties who wish to transact with each other set up a multisig wallet (which requires more than one signature to enact a transaction). This wallet holds some amount of Ethereum. The wallet address is then saved to the Ethereum blockchain. This sets up the payment channel. The two parties can now conduct an unlimited number of transactions without ever touching the information stored on the blockchain. With each transaction, both parties sign an updated balance sheet to always reflect how much of the Ethereum stored in the wallet belongs to each. When the two parties have done transacting, they close out the channel, and the resulting balance is registered on the blockchain. In the event of a dispute, both parties can use the most recently signed balance sheet to recover their share of the wallet.

It is useful to note that it is not necessary to set up a direct channel to transact on lightning – you can send payments to someone via channels with people that you are connected with. The network automatically finds the shortest

route. Development of the technology got a significant boost with the adoption of SegWit on the Ethereum and litecoin networks. Without the upgrade's transaction malleability fix, transactions on the lightning network would have been too risky to be practical. Without the security of the blockchain behind it, the lightning network will not be as secure, which implies that it will largely be used for small or even micro transactions which carry a lower risk. Larger transfers that require decentralized security are more likely to be done on the original layer.

F. Plasma

Another layer 2 scaling solution for Ethereum is Plasma. Plasma is a framework for creating child blockchains that are rooted to the main Ethereum network. Plasma chains can take on varying complexity, and many are being created to enable the processing of a large number of transactions per second. Apart from moving transactions off of the main chain, many developers are experimenting with new consensus mechanisms. The most common way to achieve distributed consensus outside of Proof-of-Work is Proof-of-Stake (PoS). In PoS, block producers are in charge of verifying sactions. Block producers are chosen randomly, but the odds of becoming one increase proportional to the number of tokens an entity holds. When block producers submit a block, they are forced to bond a number of tokens to their decisions and can be penalized if they behave maliciously. This is supposed to keep validators economically incentivized to act in the best interest of the network.

V. CONCLUSION

This paper has tried to explored the challenges in scaling Ethereum and blockchains in general. Supported by measurement studies, we showed that reparametrization of the block size and interval in Ethereum is only a first step toward substantial throughput and latency improvements while retaining significant system decentralization. Aggressive scaling will in the longer term require fundamental protocol redesign and establishment. Through a structured presentation of the design landscape for blockchain protocols, we illustrated the variety of potentially successful approaches to such scaling, categorized a range of recently proposed and new ideas, and framed a number of important open technical challenges for the community.

REFERENCES

[1] https://en.bitcoin.it/wiki/Scalability.
[2] https://en.bitcoin.it/wiki/Mining_hardware_comparison.
[3] https://blockchain.info/charts/hash-rate.
[4] https://blockchain.info/charts/n-transactions-per-block.
[5] https://bitnodes.21.co/dashboard/?days=365.
[6] https://gist.github.com/gavinandresen/e20c3b5a1d4b97f79ac2.
[7] Amazon ec2 pricing. http://aws.amazon.com/ec2/pricing/. accessed 2015-10-30.
[8] Antminer s5+ hardware. https://bitmaintech.com/productDetail.htm?pid=000201508140753265504JMKzsM067B. accessed 2015-10-30.
[9] Litecoin, open source P2P digital currency. https://litecoin.org.
[10] On scaling decentralized blockchains — a postion paper. http://www.initc3.org/scalingblockchain/.
[11] https://www.quora.com/What-are-the-problems-in-ethereum
[12] How a Visa transaction works. http://web.archive.org/web/20160121231718/http://apps.usa.visa.com/merchants/become-a-merchant/how-a-visa-transaction- works.jsp, 2015.
[13] J. Garzik. Block size increase to 2mb (bip 102). https://github.com/bitcoin/bips/blob/master/bip-0102.mediawiki, retrieved October 2015.
[14] J. Garzik. Making decentralized economic policy. http://gtf.org/garzik/bitcoin/BIP100-blocksizechangeproposal.pdf, retrieved October 2015.
[15] C. Georgiou, S. Gilbert, R. Guerraoui, and D. R. Kowalski. Asynchronous gossip.J. ACM, 60(2), May 2013.30. L. Glendenning, I. Beschastnikh, A. Krishnamurthy, and T. Anderson. ScalableConsistency in Scatter. In SOSP, 2011.

