Research Proposal
Koç Üniversitesi Research Session

Misrab M. Faizullah Khan
Advisor: Alptekin Küpçü

Koç Üniversitesi Computer Engineering Department

faizullah.misrab@gmail.com, akupcu@ku.edu.tr

December 2017

Title

Which came first, the storage or the compute? An alternative paradigm to Ethereum for scalable blockchain applications, using IPFS and Filecoin.

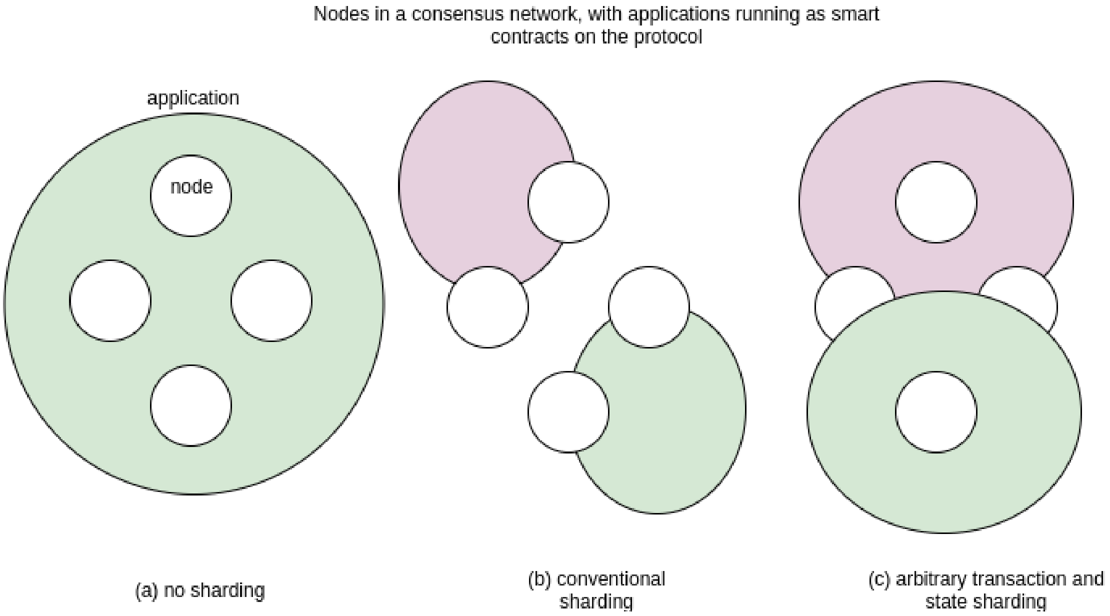
Introduction and Literature

Traditional Proof-of-Work blockchains [1], such as Ethereum [2] in its current state, require each node in the network to process each transaction. If c denotes the computational capacity of a single node, transaction capacity is then bounded by $O(c)$. This currently translates to about 10 transactions per second [3], with modest applications running into bottlenecks [4].

For most applications to become mainstream, several orders of magnitude improvement will be required. There are a few approaches to solving this problem, including: payment channels [5], more general state channels [6], and sharding. This last approach is often seen as the holy grail of scalability, and takes several forms such as purely transactional sharding [7], quadratic transaction and storage sharding, and the more challenging super-quadratic sharding [8].

All of the above methods conform to the prevalent blockchain paradigm: build a protocol that manages fundamentals like peer-to-peer networking, transaction processing, and state; start with the simplest form of Nakamoto consensus [9]; evolve the protocol to gradually introduce scalability.

There is however an alternative paradigm: smart contracts built on top of IPFS [10] and Filecoin [11]. In this universe, the building blocks of peer-to-peer networking and content-addressed storage over a DHT [12] are first solved. Next, consensus based on useful work through Proof-of-Storage (name Proof-of-Replication and Space-time [13]) enables an incentivised distributed storage system. Most interestingly, one can then construct a wide range of sharded consensus systems using only *Put* and *Get* primitives, without reinventing the building blocks. Smart contract code can be deployed with arbitrary redundancy, and contract state can be stored on the same or different nodes. This is illustrated in one of many possible ways below. It is worth noting that the consensus mechanism is not specified here: IPFS allows for a blockchain, or more general Merkle [14] tree structures.


Nodes in a consensus network, with applications running as smart contracts on the protocol
application
node
(a) no sharding
(b) conventional sharding
(c) arbitrary transaction and state sharding

The aim of this research will be to explore the spectrum of consensus architectures for smart contract applications on top of such a system, alongside their performance (scalability, safety, liveness) and security (incentive-compatibility) characteristics. In the simplest extreme, one could replicate Nakamoto consensus on top of IPFS and Filecoin. On the other hand, one could tune the degrees of sharding based on application-specific requirements. This provides an exciting alternative to the current one-size-fits-all approach, and is an avenue certainly worth exploring.

Results and Impact

At the end of this research, we should have a classical distributed system understanding of the characteristics of several consensus approaches on top of IPFS and Filecoin, a game-theoretic view of security through incentive compatibility, as well as implications on higher-level application and smart contract architecture.

It is no understatement that distributed ledger technology can fundamentally revolutionise power structures and markets. Yet the field is still highly experimental, and it remains to be seen what dominant consensus mechanisms emerge. Now is a perfect time to explore different paradigms with an open mind, and develop a sound theoretical and practical understanding of the tradeoffs involved.

References

[1] Bitcoin Proof-of-Work, https://bitcoin.org/bitcoin.pdf

[2] Ethereum White Paper, https://github.com/ethereum/wiki/wiki/White-Paper

[3] Ethereum Transaction Bottleneck, https://twitter.com/VitalikButerin/status/94074909633954

[4] Cryptokitties Congestion, https://qz.com/1145833/cryptokitties-is-causing-ethereum-network-congestion/

[5] Lightning Network, https://lightning.network/

[6] Plasma, https://plasma.io/

[7] Elastico, https://www.comp.nus.edu.sg/ loiluu/papers/elastico.pdf

[8] Ethereum Sharding, https://github.com/ethereum/sharding/blob/develop/docs/doc.md

[9] Nakamoto Consensus, https://www.quora.com/What-is-the-Nakamoto-consensus

[10] , The InterPlanetary File System, https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv

[11] Filecoin, https://filecoin.io/filecoin.pdf

[12] Distributed Hash Table (DHT), https://github.com/ipfs/go-ipfs/issues/1396

[13] Proof-of-Replication and Spacetime, https://filecoin.io/proof-of-replication.pdf

[14] Merkle DAGs, https://github.com/ipfs/specs/tree/master/merkledag