Bridging Paxos and Blockchain Consensus - Paper Review

By,
Upasana Ghosh

The distributed consensus problem has been a predominant problem plaguing distributed systems for a long time now. Various research works stemming from different perspectives and probable solutions have been put across in attempts to solve this problem in the most optimized manner. In this paper, the authors try to draw a parallel and bridge the gap between two such seemingly divergent concepts, namely Paxos and Blockchain, aiming to solve the same issue in their unique ways. The paper presents a view of how the experiences from the classical distributed consensus world and the world of novel Blockchains could interchangeably benefit each other. To achieve this view, the paper reviews various prevalent optimised flavours of blockchain protocols and compares them with the classical consensus protocols. It also suggests ways to incorporate popular techniques used while designing classical consensus algorithms to improve the design and model checking phase for Blockchain protocols and smart contracts. Finally, the paper presents an interesting analysis concerning the performance and scalability achieved by Paxos and Blockchain at the protocol level.

The paper starts by explaining how basic Paxos work and then moves on to compare it with the basic Blockchain protocol the Proof-of-Work (PoW), built on the Nakamoto consensus. It draws a parallel between the two by showcasing a list of similarities and differences portrayed by these two respectively deterministic and probabilistic protocols. Some of the comparisons mentioned in the paper are concerning the leader election phase, the way these protocols handle the multiple leaders' problem, their way of dealing with scalability issues and communication bottlenecks, and their approaches to deal with failures and attacks, in the presence of Byzantine nodes in private and public environments. The paper also compares the more advanced versions of Blockchains like Byzcoins and federated Blockchains with Paxos, in terms of providing stable leaders and achieving instant irreversibility. To reduce vulnerabilities and avoid events like DAO mishap in Blockchain-based and smart contract systems, the authors suggest the use of proven formal methods like TLA+ to identity and remove design flaws and handle edge cases during the protocol design phase itself. Although Paxos was built for solving consensus in small clusters and PoW protocol aim systems involving thousands of nodes, the authors still present a fair comparison of their performance in terms of throughput and scalability in this paper.

The most interesting thing to note about the paper is its attempt to compare and bridge two seemingly divergent solutions and its success in providing a fair picture. Classical consensus Protocols like Paxos have their own set of advantages, while novel protocols like Blockchain have their own. As the paper mentions, Paxos is ideally designed for private networks with a smaller cluster of nodes to make a system fail-safe and to provide strong consistency among the state replicated machines. Blockchain-based applications, on the other hand, is expected to solve consensus in open systems involving thousands of nodes prone to crashes, byzantine failures and Sybil attacks. Though we have classic byzantine protocols like byzantine Paxos that can be

compared to the Nakamoto consensus protocol to solve consensus in the presence of Byzantine nodes, it is interesting to note that Paxos, in general, does not scale as well as Blockchain protocols. The paper provides a fair comparison towards the end, in the form of a graph that depicts how the two protocols perform as the nodes in the cluster are increased. It is interesting to observe that even though Paxos provide better throughput for small clusters that keeps on degrading as the nodes in the cluster are increased, Blockchain maintains the same, although smaller throughput, throughout. Another important factor that impacts a system's performance is the number of communications that are exchanged between the nodes of the system. As highlighted by the authors, Paxos fails terribly while handling this communication bottleneck while making decisions. It's fascinating to see how innovatively and cleverly the Nakamoto consensus protocol bypasses this issue by adopting a simple approach. The comparison of Paxos with advanced Blockchain protocols is another interesting case study that the paper provides. It intriguing to observe how Bitcoin NG compares with multi Paxos in providing stable leader property to Blockchains by employing key blocks to provide temporary stability to elected leaders. This concept of allowing the same leader to add multiple micro blocks improves the performance and efficiency of the system by reducing mining overheads. Byzcoin employs the best of both worlds by employing PBFT along with PoW to provide instant consensus irreversibility to Blockchains. It introduces PBFT to provide immediate irreversibility by employing small councils of nodes and uses PoW to elect the council on a rolling basis to prevent colluding nodes from overpowering the council. The optimization that Federated blockchains provide on top of Byzcoin is immensely fascinating. The division of system nodes into federations or quorum slices bypasses the need for council elections using PoW, which saves a lot of computations and improves latency of the system. The paper provides a strong case study to support the viewpoint that using formal methods prevalent in distributed systems to find flaws and corner cases would be beneficial to refine Blockchain and smart-contract based algorithms during the design phase. The demonstration of the DAO hack using TLA+ sends across this point very well.

One of the main drawbacks that the author themselves mention in the paper is the challenges that modelling blockchain protocols bring to the TLA+ framework. Blockchain protocols require support for their probabilistic components. These components require TLA+ to provide basic math libraries to generate hashes, public/private signatures, etc which are not yet available to the TLA+ framework. Again, to be able to properly model blockchain protocols, TLA+ framework needs to provide a way to check loose invariants, which is currently not available. Hence, it could be said that though formal methods provide a strong guarantee to solve a good amount of issues prevalent in designing Blockchain-based applications, they still lack many of the features that are absolutely necessary to model check this genre of applications. To derive an analogy of the performance of both the protocols, the paper compares basic Paxos with Blockchain's PoW protocol. It should be taken into consideration that Paxos fails in the presence of Byzantine failures whereas PoW surpasses this issue. A better analysis of performance would have been between PBFT and PoW, as both these protocols handle Byzantine failures. Another consideration that the paper does not highlight enough is the latency comparison between advanced Blockchain protocols like Bitcoin and Paxos. Bitcoin typically supports seven transactions per second. This is way less than the thousands of transactions that Paxos support in the same amount of time. Inter-node communication poses an important bottleneck for Paxos based protocols. To reduce

this overhead to some extent, Multi Paxos introduces a concept of stable leaders i.e., maintaining the same leader for multiple rounds until the leader fails or the leader's lease expires. To maintain these leases and to detect failures, an orthogonal mechanism like a failure detection service is deployed. This is again an overhead, as this failure detection service also needs to be maintained. Bitcoin-NG introduces similar optimizations to Blockchain algorithms without the need for an extra mechanism to provide temporary stability to the elected leaders. Though the paper compares Paxos with Bitcoin-NG, it does not emphasize enough on this overhead from an extra resource requirement standpoint.

As a future extension to this work, steps should be taken to better equip the formal methods prevalently adopted by the distributed systems community like the TLA+/PlusCal framework, to incorporate novel protocols like Blockchain. As mentioned by the authors, it would be great if TLA+/PlusCal can be used to develop a formal framework and domain specific language to model check and prove the properties specific to blockchain protocols and systems. This would be extremely beneficial as it would provide a way to check for flaws and verify the immunity of a smartcontracts-based or a Blockchain based system to crashes, byzantine failures, stretched timelines and network partitions. Along with that TLA+ already provides stepwise refinement, hence having a TLA+ based framework for Blockchain protocols would provide avenues to design hybridized/hierarchical versions of the blockchain protocol that would provide improved efficiency and performance.

Acknowledgements:

- Professor Murat Demirbas' lecture notes on Blockchains - Blockchains Introduction
- Professor Murat Demirbas' lecture notes on Paxos - Paxos