Efficient Transaction Processing in Byzantine Fault Tolerant Environments

Suyash Gupta Jelle Hellings Thamir Qadah Sajjad Rahnama Mohammad Sadoghi
Exploratory Systems Lab
Department of Computer Science
University of California, Davis

Abstract

The overarching goal of any state-of-the-art blockchain application [5] is ensuring the integrity of the client transactions. A blockchain in its simplest form is a *linked-list*, which helps to maintain a stable and reliable storage of client transactions. Blockchain has acted as a resolve to challenges in food production [7], energy trading [21], managing health care [4, 9, 16], and insurance fraud prevention [14]. The key reasons behind widespread interest in blockchain systems are their fundamental characteristics: transparency, integrity and decentralization.

A blockchain supports transparency and decentralization, by employing the core database principle of *replication*. Each blockchain is maintained by several replicas. These replicas need to reach an agreement on the order of client transactions, which is resolved by employing a *consensus* protocol. Thus, at the core of any blockchain system is its underlying consensus protocol. Intial blockchain applications, such as Bitcoin [24] and Ethereum [25] encourage a *permissionless* setting through the use the Proof-of-Work [10, 15] (PoW) protocol to attain consensus. PoW requires each replica to spend its resources in finding a solution for some hard cryptographic puzzle, which acts as a proof that the replica has generated the next block. Hence all the replias reach a consensus when at least one replica transmits its proof to the majority of replicas. However, PoW protocol can lead to the case where multiple replicas can claim to have generated the proof, which could create a *fork* in the blockchain.

Further, these permissionless protocols need to provide replicas with financial incentives to maximize non-malicious participation. These requirements necessitated the design of *permissioned* systems [1, 10], which build on top of the proven consensus protocols as the identities of the replicas are known apriori. We realize this goal of a safe and efficient, permissioned blockchain system through our design of ExpoDB framework. ExpoDB provides an in-memory distributed transactional framework that allows implementing and evaluating different concurrency control and agreement protocols. Our past experience with ExpoDB includes: (i) design of an efficient concurrency control protocol, QueCC [22], (ii) design of a fast non-blocking two-phase agreement protocol, EasyCommit [11], (iii) design of toplogoy-aware geoscale agreement protocol, GeoScale EasyCommit [12], and (iv) implementation of scalable storage layer, LStore [23]. These protocols, although exciting, do not meet the requirements of a practical blockchain application. Hence we extend the ExpoDB framework and set out goals for designing efficient blockchain systems.

Our current design of ExpoDB includes implementation of several byzantine fault-tolerant consensus algorithms, such as PBFT [6], Zyzzyva [18], RBFT [2], PoW [15], and Algorand [8]. ExpoDB also implements two popular blockchain systems, Hyperledger Fabric [1] and RapidChain [26]. Note that our focus is mainly in the design of efficient byzantine fault-tolerant (BFT) consensus protocols. These protocols differ from the Paxos-style [19, 20] consensus protocols as they allow replicas to be malicious.

The existence of malicious replicas influences the design of underlying consensus protocol, which in turn affects the performance of the system. Prior works [6, 18] have noted the expensive costs associated with the BFT consensus protocols and have advocated for efficient future designs. We tackle this cost trade-off by envisioning a manifold design. ExpoDB customizes each replica by associating with it an elaborate parallel pipeline architecture. These parallel pipelines allow us to process multiple batches of client transactions, in parallel, without comprising on linearizability [13]. Further, we employ efficient cryptographic constructs [3]: (i) to digitally sign the message (256-bit ED25519 [17]) between clients and replicas, (ii) to ensure integrity of the message (128-bit CMAC-AES) between replicas, and (iii) to authenticate the message contents using hashing (SHA256).

The use of efficient pipelines and constructs can only boost the system performance to a limited extent. The key component still affecting the system throughput is the underlying consensus protocol. Although the PBFT [6] protocol is the first to design of a practical BFT consensus, it requires three phases of which two require quadratic communication. One of the most noteworthy improvement over PBFT is Zyzzyva [18], which requires only linear communication through speculative execution. However, its safety and liveness is dependent on existence of good clients.

We believe there is ample opportunity to develop efficient BFT protocols that work in fewer phases than PBFT and do not face the limitations of Zyzzyva. Further, the existing BFT designs do not discuss the challenges of a geographically large setup, where replicas are spread across continents and latencies are the major concerns. Existing blockchain applications also assume existence of a single chain, which is maintained across all the replicas. Although this chain helps to ensure a single order, it restricts parallelism and requires massive storage. We believe through the use of *sharding*, a sharded-chain can help scaling the blockchain systems.

Author Info

- **Suyash Gupta**: https://gupta-suyash.github.io
- **Jelle Hellings**: http://jhellings.nl/
- **Thamir Qadah**: http://thamir.qadah.com/
- **Sajjad Rahnama**: https://sajjadrahnama.com/
- **Mohammad Sadoghi**: https://msadoghi.github.io/

1. REFERENCES

[1] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. W. Cocco, and J. Yellick. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *CoRR*, abs/1801.10228, 2018.

[2] P.-L. Aublin, S. B. Mokhtar, and V. Quéma. RBFT: Redundant Byzantine Fault Tolerance. In *Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems*, ICDCS '13, pages 297–306, Washington, DC, USA, 2013. IEEE Computer Society.

[3] E. Barker. Recommendation for Key Management, Part 1: General. *NIST Special Publication*, 800(57):1–160, Jan. 2016.

[4] B. Blechschmidt. Blockchain in Europe: Closing the strategy gap. Technical report, Cognizant Consulting, 2018.

[5] C. Cachin and M. Vukolic. Blockchain Consensus Protocols in the Wild. *CoRR*, abs/1707.01873, 2017.

[6] M. Castro and B. Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, pages 173–186, Berkeley, CA, USA, 1999. USENIX Association.

[7] L. Ge, C. Brewster, J. Spek, A. Smeenk, and J. Top. Blockchain for agriculture and food: Findings from the pilot study. Technical report, Wageningen University, 2017.

[8] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, pages 51–68, New York, NY, USA, 2017. ACM.

[9] W. J. Gordon and C. Catalini. Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability. *Computational and Structural Biotechnology Journal*, 16:224–230, 2018.

[10] S. Gupta and M. Sadoghi. *Blockchain Transaction Processing*, pages 1–11. Springer International Publishing, Cham, 2018.

[11] S. Gupta and M. Sadoghi. EasyCommit: A Non-blocking Two-phase Commit Protocol. In *Proceedings of the 21st International Conference on Extending Database Technology*, EDBT. Open Proceedings, 2018.

[12] S. Gupta and M. Sadoghi. Efficient and non-blocking agreement protocols. *Distributed and Parallel Databases*, Apr 2019.

[13] M. P. Herlihy and J. M. Wing. Linearizability: A Correctness Condition for Concurrent Objects. *ACM TOPLAS*, 12(3), 1990.

[14] M. Higginson, J.-T. Lorenz, B. Mnstermann, and P. B. Olesen. The promise of blockchain. Technical report, McKinsey&Company, 2017.

[15] M. Jakobsson and A. Juels. Proofs of Work and Bread Pudding Protocols. In *Proceedings of the IFIP TC6/TC11 Joint Working Conference on Secure Information Networks: Communications and Multimedia Security*, CMS '99, pages 258–272. Kluwer, B.V., 1999.

[16] M. N. Kamel Boulos, J. T. Wilson, and K. A. Clauson. Geospatial blockchain: promises, challenges, and scenarios in health and healthcare. *International Journal of Health Geographics*, 17(1):1211–1220, 2018.

[17] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007.

[18] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong. Zyzzyva: Speculative Byzantine Fault Tolerance. In *Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles*, SOSP '07, pages 45–58, New York, NY, USA, 2007. ACM.

[19] L. Lamport. The Part-time Parliament. *ACM Trans. Comput. Syst.*, 16(2):133–169, May 1998.

[20] B. M. Oki and B. H. Liskov. Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems. In *Proceedings of the Seventh Annual ACM Symposium on Principles of Distributed Computing*, PODC '88, pages 8–17, New York, NY, USA, 1988. ACM.

[21] PwC. Blockchain – an opportunity for energy producers and consumers?, 2016.

[22] T. M. Qadah and M. Sadoghi. QueCC: A Queue-oriented, Control-free Concurrency Architecture. In *Proceedings of the 19th International Middleware Conference*, Middleware '18, pages 13–25, New York, NY, USA, 2018. ACM.

[23] M. Sadoghi, S. Bhattacherjee, B. Bhattacharjee, and M. Canim. L-Store: A Real-time OLTP and OLAP System. EDBT. OpenProceeding.org, 2018.

[24] N. Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.

[25] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. 2015.

[26] M. Zamani, M. Movahedi, and M. Raykova. Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 931–948, New York, NY, USA, 2018. ACM.