Towards Environmental-Adaptive and Performance-Resilient Consensus in Distributed Ledger Technology

Christian Berger
University of Passau
Passau, Germany
cb@sec.uni-passau.de

ABSTRACT

Many recent research works have proposed distributed ledger technology (DLT) that employs Byzantine fault-tolerant (BFT) consensus protocols as the underlying core primitive to create a total order among all transactions. Compared to many Proof-of-Work (PoW) blockchains, this design typically benefits from increased performance, energy efficiency and proven liveness and safety characteristics. While BFT protocols have the potential to create highly resilient infrastructures, some questions yet remain to be answered. This paper sketches our current and future research on how DLTs can benefit from making the underlying BFT protocol adaptive towards the system's environment (e.g., geographic decentralization or system scale) and resilient against attacks of malicious replicas that are targeted at degrading the overall system performance.

CCS CONCEPTS

• **Computing methodologies → Distributed algorithms**; *Modeling and simulation*; • **Computer systems organization → Dependable and fault-tolerant systems and networks**; • **General and reference** → Validation.

KEYWORDS

Byzantine Fault Tolerance, Consensus, Blockchain, Distributed Ledger Technologies, Resilience, Scalability, Performance, Adaptivity

ACM Reference Format:
Christian Berger. 2020. Towards Environmental-Adaptive and Performance-Resilient Consensus in Distributed Ledger Technology. In *21st International Middleware Conference Doctoral Symposium (Middleware '20 Doctoral Symposium), December 7–11, 2020, Delft, Netherlands.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3429351.3431740

1 MOTIVATION AND RELATED WORK

State machine replication (SMR) is a well-studied approach that allows developing resilient distributed systems. It coordinates client interactions with independent server replicas, thus achieving fault-tolerance [18]. In the Byzantine fault model [12], faulty nodes may present different symptoms towards different observers [8], which is an interesting fault model for blockchain networks. This is, because blockchains ideally achieve an overall trustworthy infrastructure on top of a network of nodes of which a fraction may exhibit potentially malicious behavior.

With the recent research interest in blockchain infrastructures, Byzantine fault-tolerant (BFT) SMR protocols have also been getting more and more attention over the last few years. To serve as an example, the Hyperledger Fabric (HLF) [2] blockchain platform has been adopted to incorporate the BFT-SMaRt [6] library as an ordering service [20] to achieve high-performance and resilient service execution. Interestingly, further improvements can be made at the protocol level to make consensus in DLTs *more practical*: for instance, weight-enabled active replication (WHEAT) [19] is an optimization that decreases latencies in a geographically dispersed environment, and thus can speed up the ordering process for blocks in Hyperledger Fabric when deployed in a wide-area network environment [20].

BFT consensus for DLT. Typically, traditional (PBFT-like [7]) BFT consensus protocols can serve as a key ingredient of distributed ledger technologies (DLTs) as they are employed to order transactions in a group of replicas without relying on the expensive Proof-of-Work (PoW) mechanism [17] and also give stricter guarantees, e.g., on consensus finality [21]. The main ambition is to reach higher throughput

Middleware '20 Doctoral Symposium, December 7–11, 2020, Delft, Netherlands

ACM ISBN 978-1-4503-8200-7/20/12…$15.00
https://doi.org/10.1145/3429351.3431740

and lower latency than PoW achieves – however, for larger system sizes the limited scalability of traditional BFT protocols can become a problem [21]. Many current research papers improve the scalability by proposing novel BFT protocols [9–11, 13–16, 22] which employ several interesting ideas, such as efficient communication topologies [10], parallelization of transaction processing [11], trusted hardware components [14], representative committees [9], cryptographic primitives [16, 22] or a combination of several ideas.

The road ahead. Along the path of many possible protocol-level improvements, we conduct research on optimizations that are *adaptive*, thus can make the consensus protocol react to environmental conditions at runtime, such as scale or geographic dispersion. A further goal is to investigate how protocols can be designed (and modeled) in a way that allows reasoning about their performance behavior in case of attacks, such as when a specific threshold of replicas harmfully tries to degrade the system performance. While BFT protocols are often published with a security analysis that, e.g., proves certain safety and liveness characteristics, it has been shown that many BFT protocols are *in practice* prone to degradation attacks or can not give reasonable guarantees [1]. In our work, we will shift the focus towards the *practical aspects* of BFT protocols when employed *within a DLT*. This also includes exploring realistic deployment scenarios and understanding the practical constraints they involve.

2 RESEARCH QUESTIONS

Although BFT protocols are being studied for decades now [7, 12], in practice, these algorithms often have deficits in respect to scalability, delivering steady performance under attacks, or the feasibility for validation, which may tarnish their practical eligibility for DLTs. We aim to investigate on research questions that focus on improving the *practical eligibility* of BFT protocols when used in DLT:

R1 How can the adaptivity of BFT protocols be improved without diminishing the resilience of the overall protocol? And which gains can we achieve?

R2 How can we design BFT systems that can deliver a predictable and acceptable performance even when 'worst-case' situations like attacks or faults occur?

R3 How can the design process be enhanced by suitable validation techniques to make sure that the actually implemented BFT systems work as intended?

3 CURRENT RESEARCH

Scaling consensus. At first, we analyzed a broad variety of novel BFT protocols. Many of them were crafted to specifically fit the needs of DLT. We focused on the question which novel techniques and designs these protocols employ to improve scalability. Then, we summarized our findings [3], concluding that great potential lies in a clever combination of efficient communication strategies like gossip and cryptographic primitives, e.g., threshold-signatures and the use of randomness, e.g., for cryptographic sortition to elect a committee or leader as well as trusted execution environments.

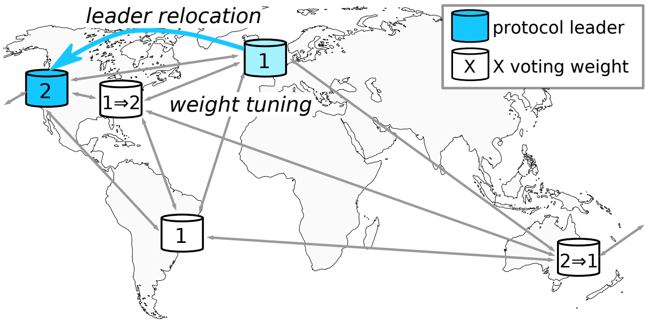
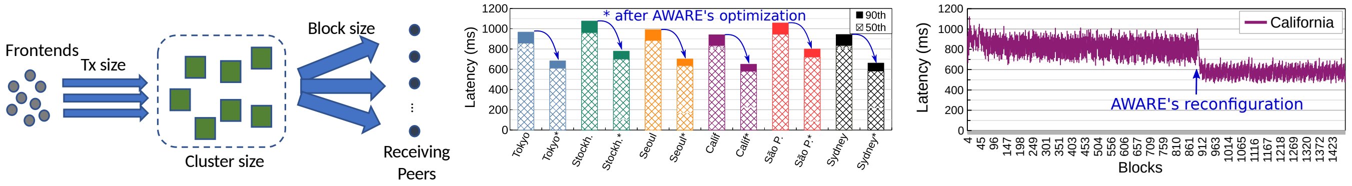


Figure 1: AWARE automatically optimizes a WHEAT configuration at run-time [5].

Adaptivity in geo-replication. Moreover, we developed and implemented[1] a protocol called AWARE (Adaptive Wide-Area REplication) [4] which is an extension of the BFT-SMaRt / WHEAT protocol. The WHEAT protocol is an optimization to achieve latency gains in geographically dispersed environments. Its core innovation is weighted replication: voting weights allow giving faster replicas more importance in quorum formation, so a proportionally smaller quorum can be probed to make progress. AWARE extends this idea by automating voting weights tuning as well as leader positioning, thus aiming for latency gains at run-time by selecting a fast performing system configuration (see Figure 1). The AWARE approach consists of (1) reliable self-monitoring of inter-replica connection latencies as decision-making basis and (2) a deterministic algorithm for self-optimization which adapts voting weights and leader position to minimize consensus latency. Moreover, this algorithm uses a prediction model that can accurately forecast the system's performance in regard to consensus latency for different configurations. Subsequently, the algorithm safely reconfigures the system. Evaluation results have shown that world-spanning Byzantine consensus systems can benefit from such a dynamic self-optimizing approach, because it allows the system to automatically adapt to changing environmental conditions [4].

Experiments with Hyperledger Fabric. Further, we conducted experiments (see Figure 2) with Hyperledger Fabric using

[1] The open-source implementation of AWARE is available at https://github.com/bergerch/aware.



(a) Ordering service model [20].

(b) Latency gains across frontends that are deployed in different regions [5].

(c) Run-time observation of a frontend deployed in California [5].

Figure 2: AWARE consensus as self-adapting ordering service for Hyperledger Fabric leads to latency gains observed by frontends distributed across the globe.

AWARE as consensus substrate to order transactions [5]: in HLF, a modular ordering service has the responsibility to achieve agreement on which block is appended next to the blockchain. We think our protocol is particularly helpful as a consensus substrate for DLTs that (1) should be optimized for geographic scalability with ordering nodes being spread across different regions in the world, (2) employ the Byzantine fault model for high resilience, and (3) aim to achieve adaptiveness to their environments.

In our experiments on Amazon AWS ordering nodes and frontends are placed in the regions *Sydney*, *São Paulo*, *California*, *Tokio* and *Stockholm*. *Frontends* send transactions (*envelopes*) to the geo-distributed *ordering cluster* for transaction ordering and block generation, blocks are then delivered to the *receiving peers* for validation (see Figure 2a). AWARE reconfigures the system by shifting voting weight from *São Paulo* to *California* resulting in latency gains. Figure 2c shows the run-time observation for *California*, and Figure 2b shows the latency gains after optimization as observed by frontends deployed in different regions.

4 FUTURE RESEARCH

Scaling consensus. To improve the scalability of consensus (**R1**), the consideration of suitable communication topologies (e.g., tree-like, hierarchical, randomized overlay) is a promising approach of state-of-the-art research. To advance these ideas further, we think it can be beneficial to make the structure of communication as flexible as possible and hide it behind an interface. In our approach, we intend to separate the management of a robust and efficient network (such as the construction of an overlay, choosing a suitable topology) from the remaining tasks of the BFT protocol. Here, the communication may be realised in a distinct layer, that is accessed by the BFT protocol over the interface. This abstraction may allow to reduce the complexity of the composed system and, at the same time, make the protocol more adaptive towards run-time conditions, e.g., an increasing scale of the system in terms of number of replicas.

Orthogonaly to this, the use of trusted hardware based approaches like Intel SGX can help to boost scalability by managing some otherwise costly tasks like replacing asymmetric cryptographic primitives by symmetric ones. Efficient communication may also require the use of certain cryptographic primitives like multi- or threshold signatures that are used for message aggregation techniques.

Modeling BFT protocols. Further, to help forecasting the behavior of a BFT protocol when under attack (**R2**), we will explore how employing models (e.g., using decomposition into building blocks) can help to reason about probabilistic lower bounds on the protocol performance, such as the latency of the composed system. The overall goal is to study the impact of attacks in terms of decreased performance of BFT protocols when integrated in DLTs and when deployed in realistic scenarios. Employing a prediction model can help to identify bottlenecks in different layers of a system (and in different configurations) even before an empirical evaluation of the system takes place. This can be helpful for optimizing the system at design time.

Until now, we have made some experiences with a prediction model for consensus latency that is based on simulation [4] (amortized simulation of the protocol run over multiple rounds) and heuristics to efficiently traverse a search space of system configurations [5]. We plan to extend these by (1) considering malicious nodes and attacks, (2) extending to different consensus protocols and (3) considering other DLT building blocks than just consensus, e.g., interaction with frontends. Apart from simulation, a variety of interesting modeling techniques exists (like timed or stochastic Petri nets) that may also help to model the system behavior in uncivil execution, such as when deployed in potentially harmful environments.

Validating practical eligibility. Moreover, to improve testing the implementation of a scalable BFT protocol into a practical system for validity (**R3**), we aim to follow an approach that joins methods of three different domains: first, using suitable modeling techniques to derive the validity of a composed protocol from validated building blocks in a constructive, bottom-up manner. Second, incorporating automated testing procedures that can, to some extent, simulate malicious behavior against BFT protocols by generating stress testing scenarios. Third, developing a tool (or extending an existing one like Hyperledger Caliper[2]) for automated deployment and benchmarking which can make use of these generated test scenarios. The overall goal is to combine complementary techniques to support a continuous design process of DLT systems, reaching from the specification of a BFT consensus protocol to its implementation to its employment in a more complex blockchain infrastructure (like Hyperledger Fabric).

5 SUMMARY

With recent developments in distributed ledger technology, resilient consensus systems become increasingly practical and necessary. In our work, we investigate on improving the *practical eligibility* of consensus protocols for their use with DLT. This includes designing consensus-based systems to be *adaptive towards their environment*, e.g., by dynamically selecting the characteristics of the ordering protocol that match best with a given set of conditions. Further, *validation techniques* and *performance models* can help to ascertain that consensus-based system work as intended and deliver acceptable performance even in case of failures or attacks.

ACKNOWLEDGMENTS

I would like to thank my supervisor Hans P. Reiser for his ongoing supervision and support. Also, I would like to thank the anonymous Doctoral Symposium reviewers for their feedback on my work. Further, I acknowledge the financial support by the Federal Ministry of Education and Research of Germany (grant nr. 01IS18068, SORRIR).

REFERENCES

[1] Yair Amir, Brian Coan, Jonathan Kirsch, and John Lane. 2011. Prime: Byzantine replication under attack. *IEEE Trans. on Dependable and Secure Comp.* 8, 4 (2011), 564–577.

[2] Elli Androulaki et al. 2018. Hyperledger Fabric: a distributed operating system for permissioned blockchains. In *Proc. of the 13th EuroSys Conf.* ACM, 30.

[3] Christian Berger and Hans P. Reiser. 2018. Scaling Byzantine Consensus: A Broad Analysis. In *Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers (SERIAL)*.

[4] Christian Berger, Hans P Reiser, João Sousa, and Alysson Bessani. 2019. Resilient Wide-Area Byzantine Consensus Using Adaptive Weighted Replication. In *Proceedings of the 38th IEEE International Symposium on Reliable Distributed Systems (SRDS)*.

[5] Christian Berger, Hans P. Reiser, João Sousa, and Alysson Bessani. 2020. AWARE: Adaptive Wide-Area Replication for Fast and Resilient Byzantine Consensus. *IEEE Transactions on Dependable and Secure Computing* [to be published]. https://doi.org/10.1109/TDSC.2020.3030605

[6] Alysson Bessani, João Sousa, and Eduardo EP Alchieri. 2014. State machine replication for the masses with BFT-SMaRt. In *44th Annu. IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN), 2014*. 355–362.

[7] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*. 173–186.

[8] Kevin Driscoll, Brendan Hall, Michael Paulitsch, Phil Zumsteg, and Hakan Sivencrona. 2004. The real byzantine generals. In *The 23rd Digital Avionics Systems Conference (IEEE Cat. No. 04CH37576)*, Vol. 2. IEEE, 6–D.

[9] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 51–68.

[10] Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. 2016. Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th USENIX Security Symposium (USENIX Security 16)*. 279–296.

[11] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. 2018. OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding. In *2018 IEEE Symposium on Security and Privacy (SP)*. 583–598.

[12] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* 4, 3 (July 1982), 382–401.

[13] Peilun Li, Guosai Wang, Xiaoqi Chen, Fan Long, and Wei Xu. 2020. Gosig: A Scalable and High-Performance Byzantine Consensus for Consortium Blockchains *(SoCC '20)*. Association for Computing Machinery, New York, NY, USA, 223–237. https://doi.org/10.1145/3419111.3421272

[14] Jian Liu, Wenting Li, G Karame, and N Asokan. 2018. Scalable Byzantine Consensus via Hardware-assisted Secret Sharing. *IEEE Trans. Comput.* (2018).

[15] Giuliano Losa, Eli Gafni, and David Mazières. 2019. Stellar Consensus by Instantiation. In *33rd International Symposium on Distributed Computing (DISC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[16] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The honey badger of BFT protocols. In *Proc. of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 31–42.

[17] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/bitcoin.pdf (last Accessed: 22/09/2020).

[18] Fred B Schneider. 1990. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys (CSUR)* 22, 4 (1990), 299–319.

[19] João Sousa and Alysson Bessani. 2015. Separating the WHEAT from the Chaff: An Empirical Design for Geo-Replicated State Machines. In *34th IEEE Symp. on Reliable Distributed Systems (SRDS)*. IEEE, 146–155.

[20] João Sousa, Alysson Bessani, and Marko Vukolić. 2018. A Byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *48th Annu. IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN)*. IEEE, 51–58.

[21] Marko Vukolić. 2015. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *International Workshop on Open Problems in Network Security*. Springer, 112–125.

[22] Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. 2019. HotStuff: BFT Consensus with Linearity and Responsiveness. In *Proc. of the 2019 ACM Symp. on Principles of Distributed Computing* (Toronto ON, Canada) *(PODC '19)*. ACM, New York, NY, USA, 347–356.

[2] See https://www.hyperledger.org/use/caliper.