ResilientDB: Global Scale Resilient Blockchain Fabric

Suyash Gupta[*] Sajjad Rahnama[*] Jelle Hellings Mohammad Sadoghi
Exploratory Systems Lab
Department of Computer Science
University of California, Davis
{sgupta,srahnama,jhellings,msadoghi}@ucdavis.edu

■Research - Short■

ABSTRACT

Recent developments in blockchain technology have inspired innovative new designs in resilient distributed and database systems. At their core, these blockchain applications typically use Byzantine fault-tolerant consensus protocols to maintain a common state across all replicas, even if some replicas are faulty or malicious. Unfortunately, existing consensus protocols are not designed to deal with *geo-scale deployments* in which many replicas spread across a geographically large area participate in consensus.

To address this, we present the Geo-Scale Byzantine Fault-Tolerant consensus protocol (GeoBFT). GeoBFT is designed for excellent scalability by using a topological-aware grouping of replicas in local clusters, by introducing parallelization of consensus at the local level, and by minimizing communication between clusters. To validate our vision of high-performance geo-scale resilient distributed systems, we implement GeoBFT in our efficient ResilientDB permissioned blockchain fabric. We show that GeoBFT is not only sound and provides great scalability, but also outperforms state-of-the-art consensus protocols by a factor of six in geo-scale deployments.

1. INTRODUCTION

Recent interest in *blockchain technology* has renewed development of distributed *Byzantine fault-tolerant* (Bft) systems that can deal with failures and malicious attacks of some participants [3, 5, 6, 9, 11, 16, 19, 21, 24, 25, 27, 30]. Although these systems are safe, they attain low throughput, especially when the nodes are spread across a wide-area network (or *geographically large distances*). We believe this contradicts the central promises of blockchain technology: *decentralization* and *democracy*, in which arbitrary replicas at arbitrary distances can participate [13, 14, 19].

At the core of any blockchain system is a Bft consensus protocol that helps participating replicas to achieve resilience. Existing blockchain database systems and data-processing frameworks typically use *permissioned blockchain designs* that rely on traditional Bft consensus [15, 17, 20, 29, 26, 28]. These permissioned blockchains employ a *fully-replicated design* in which all replicas are known and each replica holds a full copy of the data (the blockchain).

Table 1: Real-world inter- and intra-cluster communication costs in terms of the ping round-trip times (which determines *latency*) and bandwidth (which determines *throughput*). These measurements are taken in Google Cloud using clusters of `n1` machines (replicas) that are deployed in six different regions.

	Ping round-trip times (ms)						*Bandwidth* (Mbit/s)					
	O	I	M	B	T	S	O	I	M	B	T	S
Oregon (O)	≤ 1	38	65	136	118	161	7998	669	371	194	188	136
Iowa (I)		≤ 1	33	98	153	172		10004	752	243	144	120
Montreal (M)			≤ 1	82	186	202			7977	283	111	102
Belgium (B)				≤ 1	252	270				9728	79	66
Taiwan (T)					≤ 1	137					7998	160
Sydney (S)						≤ 1						7977

1.1 Challenges for Geo-scale Blockchains

To enable geo-scale deployment of a permissioned blockchain system, we believe that the underlying consensus protocol must distinguish between *local* and *global* communication. This belief is easily supported in practice. For example, in Table 1 we illustrate the ping round-trip time and bandwidth measurements. These measurements show that global message latencies are at least 33–270 times higher than local latencies, while the maximum throughput is 10–151 times lower, both implying that communication between regions is *several orders of magnitude* more costly than communication within regions. Hence, a blockchain system needs to recognize and minimize global communication if it is to attain high performance in a geo-scale deployment.

In the design of geo-scale aware consensus protocols, this translates to two important properties. First, a geo-scale aware consensus protocol needs to be *aware of the network topology*. This can be achieved by clustering replicas in a region together and favoring communication within such clusters over global inter-cluster communication. Second, a geo-scale aware consensus protocol needs to be *decentralized*: no single replica or cluster should be responsible for coordinating all consensus decisions, as such a centralized design limits the throughput to the outgoing global bandwidth and latency of this single replica or cluster.

Existing state-of-the-art consensus protocols do not share these two properties. The influential Practical Byzantine

[*] Both authors have equally contributed to this work.

Third International Symposium on Foundations and Applications of Blockchain (FAB '20) May 1, 2020, Santa Cruz, California, USA.

Fault Tolerance consensus protocol (Pbft) [7, 8] is centralized, as it relies on a single primary replica to coordinate all consensus decisions, and requires a vast amount of global communication (between all pairs of replicas). Protocols such as Zyzzyva improve on this by reducing communication costs in the optimal case [4, 22, 23]. However, these protocols still have a highly centralized design and do not favor local communication. Furthermore, Zyzzyva provides high throughput only if there are no failures and requires reliable clients [1, 10]. The recently introduced HotStuff improves on Pbft by simplifying the recovery process on primary failure [31]. This allows HotStuff to efficiently switch primaries for every consensus decision, providing the potential of decentralization. However, the design of HotStuff does not favor local communication, and the usage of threshold signatures strongly centralizes all communication for a single consensus decision to the primary of that round. Another recent protocol PoE provides better throughput than both Pbft and Zyzzyva in the presence of failures, this without employing threshold signatures [15]. Unfortunately, also PoE has a centralized design that depends on a single primary. Finally, the geo-aware consensus protocol Steward promises to do better [2], as it recognizes local clusters and tries to minimize inter-cluster communication. However, due to its centralized design and reliance on cryptographic primitives with high computational costs, Steward is unable to benefit from its topological knowledge of the network.

1.2 GeoBFT: Towards Geo-scale Consensus

In this work, we improve on the state-of-the-art by introducing GeoBFT, a topology-aware and decentralized consensus protocol. In GeoBFT, we group replicas in a region into clusters, and we let each cluster make consensus decisions independently. These consensus decisions are then shared via an optimistic low-cost communication protocol with the other clusters, in this way assuring that all replicas in all clusters are able to learn the same sequence of consensus decisions: if we have two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ with $\mathbf{n}$ replicas each, then our optimistic communication protocol requires only $\lceil \mathbf{n}/3 \rceil$ messages to be sent from $\mathcal{C}_1$ to $\mathcal{C}_2$ when $\mathcal{C}_1$ needs to share local consensus decisions with $\mathcal{C}_2$. In specific, we make the following contributions:

1. We introduce the GeoBFT consensus protocol, a novel consensus protocol that performs a topological-aware grouping of replicas into local clusters to minimize global communication. GeoBFT also decentralizes consensus by allowing each cluster to make consensus decisions independently.

2. To reduce global communication, we introduce a novel global sharing protocol that *optimistically* performs minimal inter-cluster communication, while still enabling reliable detection of communication failure.

3. The optimistic global sharing protocol is supported by a novel *remote view-change protocol* that deals with any malicious behavior and any failures.

4. We prove that GeoBFT guarantees *safety*: it achieves a unique sequence of consensus decisions among all replicas and ensures that clients can reliably detect when their transactions are executed, this independent of any malicious behavior by any replicas.

Table 2: The normal-case metrics of Bft consensus protocols in a system with z clusters, each with n replicas of which at most f, n > 3f, are Byzantine. GeoBFT provides the lowest global communication cost per consensus decision (transaction) and operates decentralized.

Protocol	Decisions	Communication		Centralized
		(Local)	*(Global)*	
GeoBFT (our paper)	$\mathbf{z}$	$\mathcal{O}(2\mathbf{z}\mathbf{n}^2)$	$\mathcal{O}(\mathbf{f}\mathbf{z}^2)$	No
↳ *single decision*	1	$\mathcal{O}(4\mathbf{n}^2)$	$\mathcal{O}(\mathbf{f}\mathbf{z})$	No
Steward	1	$\mathcal{O}(2\mathbf{z}\mathbf{n}^2)$	$\mathcal{O}(\mathbf{z}^2)$	Yes
Zyzzyva	1	$\mathcal{O}(\mathbf{z}\mathbf{n})$		Yes
Pbft	1	$\mathcal{O}(2(\mathbf{z}\mathbf{n})^2)$		Yes
PoE	1	$\mathcal{O}((\mathbf{z}\mathbf{n})^2)$		Yes
HotStuff	1	$\mathcal{O}(8(\mathbf{z}\mathbf{n}))$		Partly

5. We show that GeoBFT guarantees *liveness*: whenever the network provides reliable communication, GeoBFT continues successful operation, this independent of any malicious behavior by any replicas.

6. To validate our vision of using GeoBFT in geo-scale settings, we present our ResilientDB fabric [18] and implement GeoBFT in this fabric.[1]

7. We also implemented other state-of-the-art Bft protocols in ResilientDB (Zyzzyva, Pbft, HotStuff, and Steward), and evaluate GeoBFT against these Bft protocols using the YCSB benchmark [12]. We show that GeoBFT *achieves up-to-six times more throughput* than existing Bft protocols.

In Table 2, we provide a summary of the complexity of the normal-case operations of GeoBFT and compare this to the complexity of other popular Bft protocols.

2. CONCLUDING REMARKS

In this paper, we present our Geo-Scale Byzantine Fault-Tolerant consensus protocol (GeoBFT), a novel consensus protocol with great scalability. To achieve great scalability, GeoBFT relies on a topological-aware clustering of replicas in local clusters to minimize costly global communication, while providing parallelization of consensus. As such, GeoBFT enables geo-scale deployments of high-performance blockchain systems. To support this vision, we implement GeoBFT in our permissioned blockchain fabric—ResilientDB—and show that GeoBFT is not only correct, but also attains up to *six times* higher throughput than existing state-of-the-art Bft protocols.

This paper is part of the proceedings of VLDB Endowment, *Volume* 13, *No* 6 and can be accessed at: `https://dl.acm.org/doi/10.14778/3380750.3380757`.

3. REFERENCES

[1] Abraham, I., Gueta, G., Malkhi, D., Alvisi, L., Kotla, R., and Martin, J. Revisiting fast practical byzantine fault tolerance, 2017.

[2] Amir, Y., Danilov, C., Dolev, D., Kirsch, J., Lane, J., Nita-Rotaru, C., Olsen, J., and Zage, D. Steward: Scaling byzantine fault-tolerant replication

[1] We have open-sourced our ResilientDB fabric at https://resilientdb.com/.

to wide area networks. *IEEE Transactions on Dependable and Secure Computing 7*, 1 (2010), 80–93.

[3] Association, G. Blockchain for development: Emerging opportunities for mobile, identity and aid, 2017.

[4] Aublin, P.-L., Guerraoui, R., Knežević, N., Quéma, V., and Vukolić, M. The next 700 BFT protocols. *ACM Transactions on Computer Systems 32*, 4 (2015), 12:1–12:45.

[5] Blechschmidt, B. Blockchain in Europe: Closing the strategy gap. Tech. rep., Cognizant Consulting, 2018.

[6] Casey, M., Crane, J., Gensler, G., Johnson, S., and Narula, N. The impact of blockchain technology on finance: A catalyst for change. Tech. rep., International Center for Monetary and Banking Studies, 2018.

[7] Castro, M., and Liskov, B. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation* (1999), USENIX Association, pp. 173–186.

[8] Castro, M., and Liskov, B. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems 20*, 4 (2002), 398–461.

[9] Christie's. Major collection of the fall auction season to be recorded with blockchain technology, 2018.

[10] Clement, A., Wong, E., Alvisi, L., Dahlin, M., and Marchetti, M. Making byzantine fault tolerant systems tolerate byzantine faults. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation* (2009), USENIX Association, pp. 153–168.

[11] Compert, C., Luinetti, M., and Portier, B. Blockchain and GDPR: How blockchain could address five areas associated with gdpr compliance. Tech. rep., IBM Security, 2018.

[12] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM Symposium on Cloud Computing* (2010), ACM, pp. 143–154.

[13] Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., and Tan, K.-L. BLOCKBENCH: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data* (2017), ACM, pp. 1085–1100.

[14] Gupta, S., Hellings, J., Rahnama, S., and Sadoghi, M. An in-depth look of BFT consensus in blockchain: Challenges and opportunities. In *Proceedings of the 20th International Middleware Conference Tutorials* (2019), pp. 6–10.

[15] Gupta, S., Hellings, J., Rahnama, S., and Sadoghi, M. Proof-of-Execution: Reaching Consensus through Fault-Tolerant Speculation, 2019.

[16] Gupta, S., Hellings, J., and Sadoghi, M. Brief announcement: Revisiting consensus protocols through wait-free parallelization. In *33rd International Symposium on Distributed Computing (DISC 2019)* (2019), vol. 146 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 44:1–44:3.

[17] Gupta, S., Hellings, J., and Sadoghi, M. Scaling blockchain databases through parallel resilient consensus paradigm, 2019.

[18] Gupta, S., Rahnama, S., and Sadoghi, M. Revisiting fast practical byzantine fault tolerance, 2019.

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

[20] Herlihy, M. Blockchains from a distributed computing perspective. *Communications of the ACM 62*, 2 (2019), 78–85.

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

[22] Kotla, R., Alvisi, L., Dahlin, M., Clement, A., and Wong, E. Zyzzyva: Speculative byzantine fault tolerance. In *Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles* (2007), ACM, pp. 45–58.

[23] Kotla, R., Alvisi, L., Dahlin, M., Clement, A., and Wong, E. Zyzzyva: Speculative byzantine fault tolerance. *ACM Transactions on Computer Systems 27*, 4 (2009), 7:1–7:39.

[24] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system, 2009.

[25] Nawab, F., and Sadoghi, M. Blockplane: A global-scale byzantizing middleware. In *35th International Conference on Data Engineering* (2019), IEEE, pp. 124–135.

[26] Özsu, M. T., and Valduriez, P. *Principles of Distributed Database Systems*, 3th ed. Springer New York, 2011.

[27] Pisa, M., and Juden, M. Blockchain and economic development: Hype vs. reality. Tech. rep., Center for Global Development, 2017.

[28] Tel, G. *Introduction to Distributed Algorithms*, 2nd ed. Cambridge University Press, 2001.

[29] van Steen, M., and Tanenbaum, A. S. *Distributed Systems*, 3th ed. Maarten van Steen, 2017.

[30] Wood, G. Ethereum: a secure decentralised generalised transaction ledger, 2016. EIP-150 revision.

[31] Yin, M., Malkhi, D., Reiter, M. K., Gueta, G. G., and Abraham, I. HotStuff: BFT consensus with linearity and responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing* (2019), ACM, pp. 347–356.