Consensus in Data Management

With Use Cases in Edge-Cloud and Blockchain Systems

Faisal Nawab
University of California, Irvine
Irvine, CA
nawabf@uci.edu

Mohammad Sadoghi
University of California, Davis
Davis, CA
msadoghi@ucdavis.edu

ABSTRACT

Consensus is a fundamental problem in distributed systems, involving the challenge of achieving agreement among distributed nodes. It plays a critical role in various distributed data management problems. This tutorial aims to provide a comprehensive primer for data management researchers on the topic of consensus and its fundamental and modern applications in data management. We begin by exploring the basic principles of consensus, including the problem statement, system models, failure scenarios, and various consensus algorithms such as Paxos and its variants. The tutorial then delves into the applications of consensus in distributed data management, focusing on distributed atomic commitment and data replication. We explain how consensus is integral to these areas and present examples of research and industry work that apply consensus to data management. The tutorial extends to modern use cases of consensus in the evolving landscapes of edge-cloud systems and blockchain technology. We discuss how consensus mechanisms are being adapted and applied in these areas, highlighting their growing importance in emerging areas of data management. By exploring these cutting-edge applications, participants will gain insights into how consensus is shaping ongoing and future research on distributed data management.

The tutorial builds on the authors' recent book "Consensus in Data Management: from Distributed Commit to Blockchain". The book will serve as the foundation and reading material for the tutorial. This tutorial targets data management researchers and practitioners to equip them with the knowledge and perspective needed to innovate in these emerging fields. This includes graduate students and junior researchers starting their careers in the area of distributed data management. Also, it includes researchers in other areas of data management who wish to explore the area of distributed data management with the goal of utilizing it in their own fields.

KEYWORDS

Consensus, Data Management, Atomic Commit, Replication, Blockchain

PVLDB Reference Format:
Faisal Nawab and Mohammad Sadoghi. Consensus in Data Management. PVLDB, 17(12): 4233-4236, 2024.
doi:10.14778/3685800.3685843

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685843

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://nawab.me/main/consensus-in-data-management-book/.

1 INTRODUCTION AND OVERVIEW

1.1 Why "consensus in data management"?

Consensus [1] is a fundamental problem in distributed systems. At its core, the problem of consensus involves ensuring that a group of distributed nodes agree on the value of a variable. This problem has significant applications in distributed data management systems and has been a cornerstone in numerous applications for many decades, particularly in scenarios where data is distributed or replicated across multiple nodes.

One classic application of consensus is in the area of distributed atomic commit [2–4], where nodes must atomically commit transactions across multiple nodes. Additionally, consensus has been extensively applied in the field of data replication to ensure consistency across data replicas. These classical applications of consensus in data management were reincarnated in various stages of the development of data management research. This includes data management for peer-to-peer systems that require consensus to coordinate across the different peers. More recently, cloud computing systems brought unique challenges in distributed data management due to the scale and unique characteristics of data centers and the utilization of commodity machines in large-scale applications. Consensus played a pivotal role in the development of various cloud computing data management systems [5–7].

Beyond these classical applications, consensus is increasingly playing a pivotal role in modern data management contexts, such as edge-cloud systems catering to IoT and edge applications [8–14], as well as blockchain and decentralized applications [15–18]. In these systems, a distributed set of nodes needs to coordinate to achieve a common goal, and consensus mechanisms are instrumental in facilitating this coordination.

1.2 Target Audience

This tutorial is specifically designed for two groups of researchers and practitioners with distinct but complementary needs. The first group includes graduate students, junior researchers, and practitioners who are building a career in distributed data management. For these individuals, understanding consensus and its application in distributed data management is crucial. It provides them with the essential expertise required to navigate and contribute to this field. The tutorial will offer them foundational knowledge and practical insights, enabling them to grasp how consensus mechanisms are integrated into various data management solutions.

The second group comprises researchers and practitioners from other fields of data management who are looking to extend their solutions and research into distributed settings. With the growth of data sizes and the increasing demand for scalable solutions in numerous areas of data management, the adoption of distributed data management technology has become commonplace. This group will benefit from learning about the foundations of distributed data management and consensus, as provided in the tutorial. This knowledge will empower them to innovate at the intersection of their existing expertise and distributed data management, allowing them to develop more robust, scalable, and efficient data management solutions for their research and products.

1.3 Topics Overview

Tutorial Approach: Our approach involves presenting both the theoretical foundations of each topic and showcasing real-world examples from contemporary systems and research. This dual approach ensures a robust understanding of the theoretical underpinnings, as well as practical insights into how these concepts are applied in real-world scenarios. The presenters will build on their recent books in the topic "Consensus in Data Management: From Distributed Commit to Blockchain" [1] as well as "Fault-Tolerant Distributed Transactions on Blockchain" [19]. In these books, we also follow the approach of covering both foundational as well as practical aspects of the problem that we will utilize in this tutorial.

Foundations of Consensus: The tutorial starts with a comprehensive introduction to the consensus problem, exploring its foundational aspects and various solutions. We will delve into the typical practical applications of consensus in distributed systems, establishing a solid base for understanding its role and significance in data management.

Overview of Distributed Data Management: Participants will be provided with a quick overview of distributed data management, focusing on critical areas such as ACID properties, concurrency control, and recovery. This overview sets the stage for understanding how consensus mechanisms integrate into broader data management contexts.

Consensus in Distributed Atomic Commit: The tutorial will then explore the problem of distributed atomic commit from a data management perspective. We will discuss how consensus-based solutions offer unique properties compared to non-consensus approaches, providing insights into the role of consensus in ensuring atomicity across distributed nodes.

Consensus in Data Replication and Consensus: A significant part of the tutorial will focus on the application of consensus in data replication, a process involving copies of data across multiple locations. We will discuss the challenges of data replication and how consensus mechanisms can overcome these challenges. Various real-world solutions incorporating consensus in data replication will be examined, particularly those relevant to cloud computing solutions.

Consensus in Edge-Cloud Systems: We will also discuss the use of consensus in edge-cloud systems, particularly for IoT and edge applications. This emerging area, with its promising growth prospects, integrates consensus extensively for coordination across edge nodes. The tutorial will highlight the importance and potential of consensus in these innovative environments.

Consensus in Blockchain Systems: Finally, the tutorial will address consensus in blockchain systems, focusing on the unique aspects of these environments, such as the lack of trust between nodes. We will explore how specific consensus protocols are utilized to ensure agreement among honest nodes, even in the presence of malicious actors. The session will cover how these protocols contribute to building robust data management systems for blockchain and decentralized environments [19–33].

1.4 Tutorial Duration and Structure

The targeted duration of the tutorial is 2 hours. It will be structured in the following way: The first 20 minutes will be allocated to introduce consensus and its solutions as well as present an overview of distributed data management. Then, 30 minutes are allocated to cover consensus in distributed atomic commit protocols and data replication. Then, 40 minutes are allocated for consensus in blockchain. The remaining time will be allocated for questions and answers as well as a presenting use cases in edge-cloud and blockchain applications.

1.5 Presenters

The tutorial is presented by two presenters:

Faisal Nawab: Faisal is an Assistant Professor in the Computer Science Department of the University of California, Irvine (UCI). His work focuses on building massively-distributd edge-cloud data management systems.

Mohammad Sadoghi: Mohammad is an Associate Professor of Computer Science at the University of California, Davis. His research is focused to pioneer a resilient data platform at scale. He has over 100 publications and has filed 35 U.S. patents.

2 TUTORIAL CONTENT SAMPLE

In this section, we provide a sample of the detailed content of the tutorial. Given the space constraints, we only show the detailed contents of the first part of the tutorial, namely consensus foundations. (Section 1.3 presents a summary of the full list of topics covered in the tutorial.)

Consensus Foundations. The first part of our tutorial is dedicated to an in-depth exploration of consensus, laying a solid foundation for the audience. We begin by presenting the system model and defining the consensus problem, setting the stage for a comprehensive understanding of this critical concept in distributed systems. A key focus of this section will be on the Paxos algorithm [34], a seminal and widely-used consensus protocol with significant applications in data management. Paxos has not only been influential in academic research but has also inspired numerous industry systems. We will present the details of the algorithm of Paxos, shedding light on its operational mechanisms as well as its properties concerning safety and liveness. the discussion is centered around the normal-case operation of paxos that consists of two phases of communication, a leader election and replication phases (Figure 1). Then, we start adding more complexities to the normal-case operation and observe how paxos can tolerate these

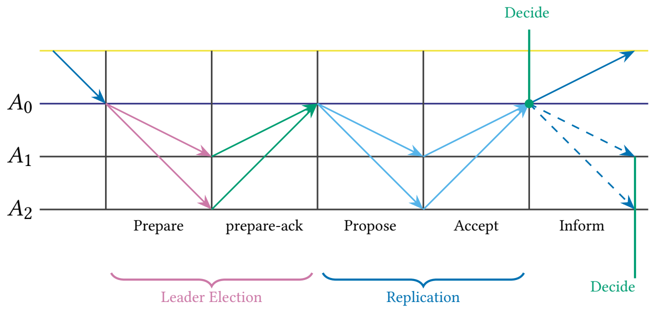

Figure 1: A schematic representation of the normal-case operation of deciding a value using the paxos protocol.

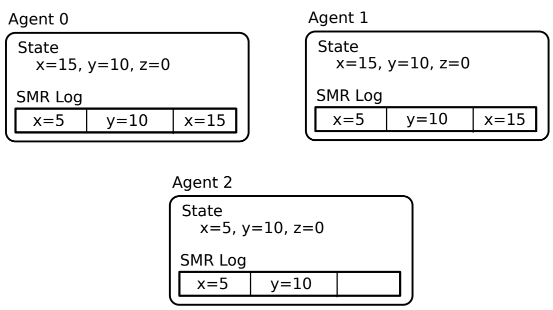

Figure 2: An example of a replicated system with three nodes running a SMR protocol. The initial state of each node consists of three variables, x, y, and z, that are initialized to 0. A consensus process is used to write to each log position to ensure agreement and fault-tolerance. In the example, agents 0 and 1 have three entries in their log and the state of the agent reflects processing the first three requests. Agent 2, however, did not receive the third entry yet and its state reflects processing the first and second entries only.

complexities—such as failures and communication delays—while guaranteeing safety properties.

Further, we delve into the application of Paxos in solving the state-machine replication problem (Figure 2.) This part of the tutorial explains how Paxos facilitates the implementation of a replicated log across distributed nodes, ensuring consistency in the logs. This discussion will provide practical insights into how consensus protocols like Paxos support the reliability and efficiency of distributed data management systems.

Concluding this section, we broaden our exploration to include other consensus protocols, comparing and contrasting their unique features and trade-offs. This comparative analysis will enhance the participants' understanding of the diverse landscape of consensus mechanisms and their suitability for different data management scenarios.

Many of the other consensus protocols are inspired and built on paxos. This includes Fast Paxos [35]—that introduces the concept of a fast path to decide c-values using a single round of communication—and Generalized Paxos [36]—that extends paxos to enable reaching agreement on partial order rather than total order. A body of work extends paxos to manage reconfiguration [37–42]—which is the problem of changing the set of agents running the consensus protocol. Load balancing is an important extension to paxos to avoid the overhead that is placed on the leader. This led to extensions such as S-Paxos [43] that distributed some of the work of the leader to other replicas which reduces the load on the leader. Work such as DPaxos [8] and WPaxos [44] explore extensions such as hierarchy, locality, and sharding to enable better performance in geo-replication settings. Utilizing variants of paxos such as Fast Paxos [35] and Generalized Paxos [36] has also been explored to reduce the amount of wide-area communication in protocols such as MDCC [45]. Mencius [46] is a multi-leader system that is based on paxos. It aims to enable faster latency by partitioning log entries across agents and serving client requests using the closest agent and its assigned log entries. Moraru et. al. [47] propose Egalitarian Paxos (EPaxos) that aim to reduce communication complexity by utilizing a fast path design and information about conflicts.

Other than paxos, there has been a number of consensus protocols. Viewstamp Replication [48] is a primary-based replication system that in normal-case operation acts similarly to multi-paxos. Viewstamp Replication utilizes a special view change process to tolerate failures. RAFT [49] is a consensus-based replication system that aims to provide a more understandable solution for consensus. Zab [50] is an atomic broadcast algorithm that is used to propagate state from a primary to a set of backup nodes in a consistent and fault-tolerant manner. These various protocols have many common features but also differ in various ways. Vive La Difference [51] studies these similarities and differences.

3 CONCLUSION

The tutorial we propose covers the topic of utilizing consensus as a tool to enable better designs of distributed data management systems. The tutorial presents a primer about consensus and its general applications. Then, the tutorial discussed various problems in data management and how consensus can be beneficial in their design. We start with the problem of distributed atomic commitment that ensures the safety of committing transactions across distributed nodes. Then, we discuss data replication and how consensus can be applied in various ways to implement consistent replication across nodes. We present use cases for this type of application of consensus in various application areas including cloud computing, geo-replication, and edge-cloud systems. Finally, we present the implications of consensus in data management in the area of blockchain and decentralized applications. We then discuss how data management applications are built on top of these new consensus primitives.

4 ACKNOWLEDGMENTS

The work done in this tutorial proposal is supported in part by NSF grants 1815212, 2245372, 2245373, and 2321121, and gifts from Meta and Roblox. The tutorial proposal is based in part on a monograph written by the presenters published in Foundations and Trends in Databases [1].

REFERENCES

[1] F. Nawab and M. Sadoghi, "Consensus in data management: From distributed commit to blockchain," *Foundations and Trends® in Databases*, vol. 12, no. 4, pp. 221–364, 2023.

[2] P. A. Alsberg and J. D. Day, "A principle for resilient sharing of distributed resources," in *Proceedings of the 2nd International Conference on Software Engineering (ICSE)*, 1976, pp. 562–570. (1976).

[3] B. Lampson and H. E. Sturgis, "Crash recovery in a distributed data storage system," in *Computer Science Lab, Xerox Parc, Palo Alto, CA, Technical Report. (1976)*, 1976.

[4] J. N. Gray, "Notes on data base operating systems," in *Operating Systems, An Advanced Course*, ser. Lecture Notes in Computer Science. Springer, 1978, vol. 60, pp. 393–481. (1978).

[5] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild *et al.*, "Spanner: Google's globally distributed database," *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 3, pp. 1–22. (2013), 2013.

[6] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, pp. 1–26. (2008), 2008.

[7] J. Baker, C. Bond, J. C. Corbett, J. J. Furman, A. Khorlin, J. Larson, J. Leon, Y. Li, A. Lloyd, and V. Yushprakh, "Megastore: Providing scalable, highly available storage for interactive services," in *Innovative Data Systems Research (CIDR)*, pp. 223–234. (2011).

[8] F. Nawab, D. Agrawal, and A. El Abbadi, "Dpaxos: Managing data closer to users for low-latency and mobile applications," in *ACM International Conference on Management of Data (SIGMOD)*, 2018, pp. 1221–1236. (2018).

[9] D. Abadi, O. Arden, F. Nawab, and M. Shadmon, "Anylog: a grand unification of the internet of things," in *Conference on Innovative Data Systems Research (CIDR '20)*, 2020.

[10] F. Nawab and M. Shadmon, "The tipping point of edge-cloud data management." in *Conference on Innovative Data Systems Research (CIDR '24)*, 2024.

[11] S. Gupta, S. Rahnama, E. Linsenmayer, F. Nawab, and M. Sadoghi, "Reliable transactions in serverless-edge architecture," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 301–314.

[12] A. Khan, S. Mehrotra, and F. Nawab, "Transedge: Supporting efficient read queries across untrusted edge nodes." 2023.

[13] N. Mittal and F. Nawab, "Coolsm: Distributed and cooperative indexing across edge and cloud machines," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 420–431.

[14] F. Nawab, D. Agrawal, and A. El Abbadi, "Nomadic datacenters at the network edge: Data management challenges for the cloud with mobile infrastructure." in *EDBT*, 2018, pp. 497–500.

[15] J. Hellings and M. Sadoghi, "Byshard: Sharding in a byzantine environment," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 2230–2243. (2021), 2021.

[16] M. El-Hindi, C. Binnig, A. Arasu, D. Kossmann, and R. Ramamurthy, "BlockchainDB: A shared database on blockchains," *Proc. VLDB Endow.*, vol. 12, no. 11, pp. 1597–1609, 2019.

[17] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: a scalable blockchain database," *white paper, BigChainDB*, pp. 53–72, 2016.

[18] J. Fang, F. Habibi, K. Bruhwiler, F. Alshammari, A. Singh, Y. Zhou, and F. Nawab, "Pelopartition: Improving blockchain resilience to network partitioning," in *2022 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2022, pp. 274–281.

[19] S. Gupta, J. Hellings, and M. Sadoghi, *Fault-Tolerant Distributed Transactions on Blockchain*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers. (2021), 2021.

[20] S. Gupta, M. J. Amiri, and M. Sadoghi, "Chemistry behind agreement," in *13th Conference on Innovative Data Systems Research, CIDR 2023, Amsterdam, The Netherlands, January 8-11, 2023*. www.cidrdb.org, 2023.

[21] M. J. Amiri, C. Wu, D. Agrawal, A. E. Abbadi, B. T. Loo, and M. Sadoghi, "The bedrock of byzantine fault tolerance: A unified platform for BFT protocols analysis, implementation, and experimentation," in *21st USENIX Symposium on Networked Systems Design and Implementation, NSDI 2024, Santa Clara, CA, April 15-17, 2024*, L. Vanbever and I. Zhang, Eds. USENIX Association, 2024, pp. 371–400.

[22] S. Gupta, J. Hellings, and M. Sadoghi, "RCC: resilient concurrent consensus for high-throughput secure transaction processing," in *IEEE International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 1392–1403. (2021).

[23] S. Gupta, J. Hellings, S. Rahnama, and M. Sadoghi, "Proof-of-Execution: Reaching consensus through fault-tolerant speculation," in *International Conference on Extending Database Technology (EDBT)*. OpenProceedings.org, 2021, pp. 301–312. (2021).

[24] J. Chen, S. Gupta, S. Rahnama, and M. Sadoghi, "Power-of-Collaboration: A sustainable resilient ledger built democratically," *IEEE Data Eng. Bull.*, vol. 45, no. 2, pp. 25–36, 2022.

[25] J. Hellings and M. Sadoghi, "ByShard: sharding in a byzantine environment," *VLDB J.*, vol. 32, no. 6, pp. 1343–1367, 2023.

[26] S. Gupta, S. Rahnama, J. Hellings, and M. Sadoghi, "ResilientDB: Global scale resilient blockchain fabric," *Proceedings of the VLDB Endowment*, vol. 13, no. 6, pp. 868–883. (2020), 2020.

[27] D. Kang, S. Rahnama, J. Hellings, and M. Sadoghi, "SpotLess: Concurrent rotational consensus made practical through rapid view synchronization," in *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, Netherlands, May 13-17, 2024*. IEEE, 2024.

[28] J. Hellings, D. P. Hughes, J. Primero, and M. Sadoghi, "[problem] cerberus: Minimalistic multi-shard byzantine-resilient transaction processing," *Journal of Systems Research*, vol. 3, no. 1, 2023.

[29] S. Rahnama, S. Gupta, R. Sogani, D. Krishnan, and M. Sadoghi, "RingBFT: Resilient consensus over sharded ring topology," in *International Conference on Extending Database Technology (EDBT)*. OpenProceedings.org, 2022, pp. 2:298–2:311. (2022).

[30] S. Gupta, S. Rahnama, S. Pandey, N. Crooks, and M. Sadoghi, "Dissecting BFT consensus: In trusted components we trust!" in *Proceedings of the Eighteenth European Conference on Computer Systems, EuroSys 2023, Rome, Italy, May 8-12, 2023*, G. A. D. Luna, L. Querzoni, A. Fedorova, and D. Narayanan, Eds. ACM, 2023, pp. 521–539.

[31] F. Nawab and M. Sadoghi, "Blockplane: A global-scale Byzantizing middleware," in *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*. IEEE, 2019, pp. 124–135.

[32] J. Hellings and M. Sadoghi, "[Solution] Byzantine cluster-sending in expected constant cost and constant time," *J. Syst. Res.*, vol. 3, no. 1, 2023.

[33] ——, "The fault-tolerant cluster-sending problem," in *Foundations of Information and Knowledge Systems - 12th International Symposium, FoIKS 2022, Helsinki, Finland, June 20-23, 2022, Proceedings*, ser. Lecture Notes in Computer Science, I. Varzinczak, Ed., vol. 13388. Springer, 2022, pp. 168–186.

[34] L. Lamport, "Paxos made simple," *ACM Sigact News*, vol. 32, no. 4, pp. 18–25. (2001), 2001.

[35] ——, "Fast paxos," *Distributed Computing*, vol. 19, no. 2, pp. 79–103. (2006), 2006.

[36] ——, "Generalized consensus and paxos," *Technical Report MSR-TR-2005-33, Microsoft Research. (2005)*, 2005.

[37] L. Lamport, D. Malkhi, and L. Zhou, "Stoppable paxos," *TechReport, Microsoft Research. (2008)*, 2008.

[38] ——, "Vertical paxos and primary-backup replication," in *ACM symposium on Principles of Distributed Computing (PODC)*, 2009, pp. 312–313. (2009).

[39] ——, "Reconfiguring a state machine," *ACM SIGACT News*, vol. 41, no. 1, pp. 63–73. (2010), 2010.

[40] V. Gramoli, L. Bass, A. Fekete, and D. W. Sun, "Rollup: Non-disruptive rolling upgrade with fast consensus-based dynamic reconfigurations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2711–2724. (2015), 2015.

[41] L. Lamport and M. Massa, "Cheap paxos," in *International Conference on Dependable Systems and Networks*, 2004, pp. 307–314. (2004).

[42] J. R. Lorch, A. Adya, W. J. Bolosky, R. Chaiken, J. R. Douceur, and J. Howell, "The smart way to migrate replicated stateful services," in *ACM SIGOPS European Conference on Computer Systems (EuroSys)*, 2006, pp. 103–115. (2006).

[43] M. Biely, Z. Milosevic, N. Santos, and A. Schiper, "S-paxos: Offloading the leader for high throughput state machine replication," in *IEEE Symposium on Reliable Distributed Systems (SRDS)*, 2012, pp. 111–120. (2012).

[44] A. Ailijiang, A. Charapko, M. Demirbas, and T. Kosar, "Wpaxos: Wide area network flexible consensus," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 1, pp. 211–223. (2019), 2019.

[45] T. Kraska, G. Pang, M. J. Franklin, S. Madden, and A. Fekete, "Mdcc: Multi-data center consistency," in *ACM European Conference on Computer Systems (EuroSys)*, 2013, pp. 113–126. (2013).

[46] Y. Mao, F. P. Junqueira, and K. Marzullo, "Mencius: Building efficient replicated state machine for wans," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 369–384. (2008).

[47] I. Moraru, D. G. Andersen, and M. Kaminsky, "There is more consensus in egalitarian parliaments," in *ACM Symposium on Operating Systems Principles (SOSP)*, 2013, pp. 358–372. (2013).

[48] B. M. Oki and B. H. Liskov, "Viewstamped replication: A new primary copy method to support highly-available distributed systems," in *ACM Symposium on Principles of Distributed Computing (PODC)*, 1988, pp. 8–17. (1988).

[49] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *USENIX Annual Technical Conference (ATC)*, 2014, pp. 305–319. (2014).

[50] F. P. Junqueira, B. C. Reed, and M. Serafini, "Zab: High-performance broadcast for primary-backup systems," in *IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, 2011, pp. 245–256. (2011).

[51] R. Van Renesse, N. Schiper, and F. B. Schneider, "Vive la différence: Paxos vs. viewstamped replication vs. zab," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 472–484. (2014), 2014.