Consensus in Data Management: From Distributed Commit to Blockchain

Other titles in Foundations and Trends® in Databases

Modern Datalog Engines
Bas Ketsman and Paraschos Koutris
ISBN: 978-1-63828-042-2

Natural Language Interfaces to Data
Abdul Quamar, Vasilis Efthymiou, Chuan Lei and Fatma Özcan
ISBN: 978-1-63828-028-6

Trends in Explanations: Understanding and Debugging Data-driven Systems
Boris Glavic, Alexandra Meliou and Sudeepa Roy
ISBN: 978-1-68083-880-0

Differential Privacy for Databases
Joseph P. Near and Xi He
ISBN: 978-1-68083-850-3

Database Systems on GPUs
Johns Paul, Shengliang Lu and Bingsheng He
ISBN: 978-1-68083-848-0

Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases
Gerhard Weikum, Xin Luna Dong, Simon Razniewski and Fabian Suchanek
ISBN: 978-1-68083-836-7

Consensus in Data Management: From Distributed Commit to Blockchain

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

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

the essence of knowledge
Boston — Delft

Foundations and Trends® in Databases

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

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.

ISBN: 978-1-63828-161-0

Foundations and Trends® in Databases

Volume 12, Issue 4, 2023

Editorial Scope

Topics

Foundations and Trends® in Databases publishes survey and tutorial articles in the following topics:

- Data Models and Query Languages
- Query Processing and Optimization
- Storage, Access Methods, and Indexing
- Transaction Management, Concurrency Control and Recovery
- Deductive Databases
- Parallel and Distributed Database Systems
- Database Design and Tuning
- Metadata Management
- Object Management
- Trigger Processing and Active Databases
- Data Mining and OLAP
- Approximate and Interactive Query Processing
- Data Warehousing
- Adaptive Query Processing
- Data Stream Management
- Search and Query Integration
- XML and Semi-Structured Data
- Web Services and Middleware
- Data Integration and Exchange
- Private and Secure Data Management
- Peer-to-Peer, Sensornet and Mobile Data Management
- Scientific and Spatial Data Management
- Data Brokering and Publish/Subscribe
- Data Cleaning and Information Extraction
- Probabilistic Data Management

Information for Librarians

Foundations and Trends® in Databases, 2023, Volume 12, 4 issues. ISSN paper version 1931-7883. ISSN online version 1931-7891. Also available as a combined paper and online subscription.

To my Sun, Moon, and Heart;
Saud, Huda, and Reem
- Faisal Nawab

To my wife, my mother
Nasim Sadoghi, Lili Taghavi
- Mohammad Sadoghi

Contents

Consensus in Data Management: From Distributed Commit to Blockchain

Faisal Nawab[1] and Mohammad Sadoghi[2]

[1] *University of California, Irvine, USA; nawabf@uci.edu*
[2] *University of California, Davis, USA; msadoghi@ucdavis.edu*

ABSTRACT

The problem of distributed consensus has played a major role in the development of distributed data management systems. This includes the development of distributed atomic commit and replication protocols. In this monograph, we present foundations of consensus protocols and the ways they were utilized to solve distributed data management problems. Also, we discuss how distributed consensus contributes to the development of emerging blockchain systems. This includes an exploration of consensus protocols and their use in systems with malicious actors and arbitrary faults.

Our approach is to start with the basics of representative consensus protocols where we start from classic consensus protocols and show how they can be extended to support better performance, extended features, and/or adapt to different system models. Then, we show how consensus can be utilized as a tool in the development of distributed data management. For each data management problem, we start by showing a basic solution to the problem and highlighting its shortcomings that invites the utilization of consensus. Then,

Faisal Nawab and Mohammad Sadoghi (2023), "Consensus in Data Management: From Distributed Commit to Blockchain", Foundations and Trends® in Databases: Vol. 12, No. 4, pp 221–364. DOI: 10.1561/1900000075.

we demonstrate the integration of consensus to overcome these shortcomings and provide desired design features. We provide examples of each type of integration of consensus in distributed data management as well as an analysis of the integration and its implications.

1

Introduction

Consensus [49], [132]—which is the problem of making distributed nodes reach agreement—has influenced data management systems and research for many decades. This influence is due to consensus being a basic building block that can be used in more complex distributed data management systems while retaining correctness guarantees of the state of the data and its recovery.

Consensus becomes relevant to data management systems when data is distributed across multiple nodes. When multiple nodes are working together, many complexities arise due to communication uncertainties and the possibility of machine failures. This is the case in fundamental data management problems such as distributed atomic commitment and database replication [21], [56], [108], [129], [146]. Solving the intricacies of distributed coordination, network uncertainties, and failures in such complex data management problems is a daunting challenge. This has led many systems designers to utilize consensus as a tool to build more complex distributed protocols.

Consensus is solved in different ways depending on the system model and assumptions. One major factor in the design of consensus protocols is the failure model. The failure model can be a benign model—such as crash fault-tolerance—where a node fails by stopping to engage in the protocol. Also, it can be a *byzantine* failure model [106], [132]—where a failed node can act in arbitrary ways including acting maliciously to influence the system negatively. In addition to the failure model, the network communication model also has an influence on the design and practicality of the proposed protocol. Communication models vary in a spectrum between a synchronous model—where time bounds on message reception are assumed—and an asynchronous model—where messages can be delayed indefinitely.

Variants of consensus algorithms are designed to answer unique challenges in different environments. Protocols that work best in a tightly-connected cluster might not be suitable for a distributed network separated by wide-area latency. Similarly, the workload plays an influence on whether to optimize for reaching consensus or learning about prior consensus outcomes. The goals of the protocol also play a part in how consensus algorithms are designed. Many protocols focus on achieving higher performance. However, some might optimize for lower latency while others optimize for higher throughput. Other than performance, a consensus algorithm might optimize for load balancing, faster recovery, or ease of understanding and implementation.

Consensus has renewed interest in the data management community in response to new problems. This interest started when consensus algorithms were utilized in replication and atomic commit protocols in distributed data management systems. With the growing interest in cloud computing in the 2000s, consensus has been explored as a means to design highly-available systems that are replicated across commodity machines. As cloud computing continued growing, consensus has also been explored in disaster recovery and multi-data center environments where data is copied and distributed across large geographic locations. More recently, cryptocurrency and blockchain-based applications reignited the interest in consensus and introduced a new breed of consensus algorithms that allow unique properties such as open membership to anonymous nodes [124], [155]. Data management systems has explored the use of such blockchain-based systems and consensus for applications spanning supply-chain management and decentralized finance, among others.

This monograph presents consensus as well as how it has been used to solve various distributed data management problems. The goal of this monograph is to provide a foundation for the reader to understand the landscape of using consensus protocols in data management systems as well as empower data management researchers and practitioners to pursue work that utilizes and innovates in consensus for their data management applications. This monograph is not meant to be a survey of consensus protocols nor it is a survey of data management systems that uses consensus. Rather, it presents the foundations of consensus and consensus in data management by presenting in more detail work that has been influential or representative of the data management areas we explore.

The monograph starts with a section to introduce the principles of consensus (Section 2). This section builds the foundation needed for the rest of the monograph to understand the consensus problem as well as the core consensus protocols that are widely-used in data management systems. Specifically, we will formally present the consensus problem and its guarantees as well as the space of system model and assumptions used by different protocols. Then, we present the paxos protocol in detail. Paxos [98], [99] is one of the most influential consensus algorithms that has been used—along with its variants—in many data management systems. We then present other consensus algorithms in different levels of detail to provide an intuition of the space of consensus algorithms including variants of the paxos protocol. Finally, we present how consensus is typically used in real systems using the abstraction of state-machine replication and what are other distributed systems problems that share properties with the consensus problem.

Section 3 presents background on the use of consensus in data management which provides an intuition of why and how consensus influences data management systems and the types of data management problems that invite the use of consensus protocols. This is done by providing a historical perspective of the development of distributed data management systems and how consensus has played a role in the various steps of this development. This section also presents background on data management systems that is needed for the rest of this monograph. It presents the system and data model of data management systems that we utilize for the rest of the monograph. Also, it introduces the problems of transaction processing, concurrency control and recovery, as they are typically concerns that are involved while using consensus algorithms in distributed data management systems.

Section 4 presents how consensus is used for the distributed atomic commit problem, which is one of the most important problems in distributed data management systems. The section begins with an overview of the problem of atomic commitment and the significance of this problem in distributed and partitioned databases. This includes a detailed description of seminal protocols such as Two-Phase Commit (2PC) [9], [56], [108]. Then, we present more details about distributed atomic commit protocols that use consensus as a foundation. We present in more detail the paxos commit protocol [58] to represent a class of atomic commit protocols using consensus. We start from that description to discuss other atomic commit protocols that use consensus in different ways. We conclude the section with a discussion on the relation between the atomic commit and consensus problems. This relation stems from both protocols aiming to reach agreement across distributed nodes and show how many elements of atomic commit protocols and consensus protocols overlap and aim to provide similar properties.

Section 5 presents how consensus is used in replication protocols where data copies are distributed across different nodes. This section begins with an introduction to the problem of data replication and its significance in data management systems for performance and fault-tolerance. This includes presenting some early work on data replication and the ensuing concurrency control concerns. Then, we discuss how consensus can be used to solve the replication problem. In particular, we show how the state-machine replication abstraction has been used to enable multiple nodes to maintain copies of data that are consistent and recoverable. We also discuss how replication of individual participants in atomic commit protocols can be used as an alternative to the approaches we have shown in Section 4. We also present different variations of how consensus is used in different environments. In particular, we discuss the use of consensus in replicating for highly-available systems that gained popularity in cloud computing. Also, we present how consensus is adapted and used in environments that span large geographic locations such as multi-data center and geo-replicated systems.

Section 6 expands the scope of the crash-tolerant commit protocols to handle arbitrary failures. To this end, we explore in-depth the seminal fault-tolerant consensus protocol known as PBFT (Practical Byzantine Fault Tolerance) [34]. We present PBFT as the foundation for navigating and examining the consensus landscape. We further explore speculative, optimistic, linearized, and concurrent consensus designs. We conclude this section by examining the topology of consensus in the context of cross-shard and cross-chain designs. Our ultimate aim is to simplify and make the design of these intricate protocols accessible to a wide range of audiences, a stepping stone to further advancing this field.

Section 7 concludes the monograph with a summary and a discussion of future directions. We discuss the potential impact of utilizing and extending consensus in the areas of serverless computing, decentralized applications, and edge-cloud systems.

References

[1] M. Abebe, B. Glasbergen, and K. Daudjee, "Dynamast: Adaptive dynamic mastering for replicated systems," in *36th IEEE International Conference on Data Engineering (ICDE)*, 1381–1392. (2020), 2020.

[2] M. Abebe, B. Glasbergen, and K. Daudjee, "Morphosys: Automatic physical design metamorphosis for distributed database systems," *Proceedings of the VLDB Endowment*, vol. 13, no. 13, 2020, 3573–3587. (2020).

[3] I. Abraham, N. Crooks, N. Giridharan, H. Howard, and F. Suri-Payer, "Brief announcement: It's not easy to relax: Liveness in chained BFT protocols," in *36th International Symposium on Distributed Computing (DISC)*, vol. 246, 39:1–39:3. (2022), 2022.

[4] I. Abraham, G. Gueta, D. Malkhi, L. Alvisi, R. Kotla, and J. Martin, "Revisiting fast practical byzantine fault tolerance," *CoRR*, vol. abs/1712.01367. (2017), 2017. arXiv: 1712.01367.

[5] I. Abraham, G. Gueta, D. Malkhi, and J. Martin, "Revisiting fast practical byzantine fault tolerance: Thelma, velma, and zelma," *CoRR*, vol. abs/1801.10022. (2018), 2018. arXiv: 1801.10022.

[6] A. Adya, “Weak consistency: A generalized theory and optimistic implementations for distributed transactions,” Ph.D. dissertation, Massachusetts Institute of Technology (MIT), Deptartment of Electrical Engineering and Computer Science (EECS). (1999), 1999.

[7] D. Agrawal, G. Alonso, A. El Abbadi, and I. Stanoi, “Exploiting atomic broadcast in replicated databases,” in *European Conference on Parallel Processing (Euro-Par)*, 496–503. (1997), 1997.

[8] 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, 2019, 211–223. (2019).

[9] 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)*, 562–570. (1976), 1976.

[10] M. J. Amiri, D. Agrawal, and A. E. Abbadi, “SharPer: Sharding permissioned blockchains over network clusters,” in *ACM International Conference on Management of Data (SIGMOD)*, 76–88. (2021), 2021.

[11] M. J. Amiri, C. Wu, D. Agrawal, A. E. Abbadi, B. T. Loo, and M. Sadoghi, “The Bedrock of BFT: A unified platform for BFT protocol design and implementation,” *CoRR*, vol. abs/2205.04534. (2022), 2022. arXiv: 2205.04534.

[12] A. M. Antonopoulos and G. Wood, *Mastering ethereum: building smart contracts and dapps*. O’reilly Media. (2018), 2018.

[13] V. Arora, T. Mittal, D. Agrawal, A. El Abbadi, X. Xue, *et al.*, “Leader or majority: Why have one when you can have both? improving read scalability in raft-like consensus protocols,” in *USENIX Workshop on Hot Topics in Cloud Computing (HotCloud). (2017)*, 2017.

[14] G. Ateniese, I. Bonacina, A. Faonio, and N. Galesi, “Proofs of Space: When space is of the essence,” in *Security and Cryptography for Networks*, 538–557. (2014), 2014.

[15] H. Attiya, A. Bar-Noy, D. Dolev, D. Koller, D. Peleg, and R. Reischuk, "Achievable cases in an asynchronous environment," in *Annual Symposium on Foundations of Computer Science (SFCS)*, 337–346. (1987), 1987.

[16] P. Bailis and A. Ghodsi, "Eventual consistency today: Limitations, extensions, and beyond," *Communications of the ACM*, vol. 56, no. 5, 2013, 55–63. (2013).

[17] 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)*, 223–234. (2011), 2011.

[18] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *Financial Cryptography and Data Security*, 142–157. (2016), Springer, 2016.

[19] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake (extended abstract)," *SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, 2014, 34–37. (2014).

[20] H. Berenson, P. A. Bernstein, J. Gray, J. Melton, E. J. O'Neil, and P. E. O'Neil, "A critique of ANSI SQL isolation levels," in *ACM International Conference on Management of Data (SIGMOD)*, 1–10. (1995), 1995.

[21] P. A. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency control and recovery in database systems*, vol. 370. Addison-Wesley, 1987. URL: http://research.microsoft.com/en-us/people/philbe/ccontrol.aspx.

[22] A. Bessani, M. Santos, J. Felix, N. Neves, and M. Correia, "On the efficiency of durable state machine replication," in *USENIX Annual Technical Conference (ATC)*, 169–180. (2013), 2013.

[23] 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)*, 111–120. (2012), 2012.

[24] K. P. Birman and R. V. Renesse, *Reliable distributed computing with the Isis toolkit*. IEEE Computer Society Press. (1993), 1993.

[25] BitShares-Core Contributors, *Bitshares documentation*, 2020. URL: https://how.bitshares.works/_/downloads/en/master/pdf/.

[26] M. Bravo, G. V. Chockler, and A. Gotsman, "Making byzantine consensus live," in *International Symposium on Distributed Computing (DISC)*, ser. LIPIcs, vol. 179, 23:1–23:17. (2020), 2020.

[27] M. Bravo, G. V. Chockler, and A. Gotsman, "Making byzantine consensus live," *Distributed Computing*, vol. 35, no. 6, 2022, 503–532. (2022).

[28] M. F. Bridgland and R. J. Watro, "Fault-tolerant decision making in totally asynchronous distributed systems," in *ACM Symposium on Principles of Distributed Computing*, 52–63. (1987), 1987.

[29] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Ph.D. dissertation, University of Guelph. (2016), 2016.

[30] E. Buchman, J. Kwon, and Z. Milosevic, "The latest gossip on BFT consensus," *CoRR*, vol. abs/1807.04938. (2018), 2018. arXiv: 1807.04938.

[31] M. Burrows, "The chubby lock service for loosely-coupled distributed systems," in *Symposium on Operating Systems Design and Implementation (OSDI)*, 335–350. (2006), 2006.

[32] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *CoRR*, vol. abs/1710.09437. (2017), 2017. arXiv: 1710.09437.

[33] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 173–186. (1999), 1999.

[34] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, 2002, 398–461. (2002).

[35] T. D. Chandra, R. Griesemer, and J. Redstone, "Paxos made live: An engineering perspective," in *ACM Symposium on Principles of Distributed Computing (PODC)*, 398–407. (2007), 2007.

[36] T. D. Chandra, V. Hadzilacos, and S. Toueg, "The weakest failure detector for solving consensus," *Journal of the ACM (JACM)*, vol. 43, no. 4, 1996, 685–722. (1996).

[37] 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, 2008, 1–26. (2008).

[38] A. Charapko, A. Ailijiang, and M. Demirbas, "Pigpaxos: Devouring the communication bottlenecks in distributed consensus," in *ACM International Conference on Management of Data (SIGMOD)*, 235–247. (2021), 2021.

[39] J. Chen, S. Gupta, S. Rahnama, and M. Sadoghi, "Power-of-Collaboration: A sustainable resilient ledger built democratically," *IEEE Data Engineering Bulletin*, vol. 45, no. 2, 2022, 25–36. (2022).

[40] 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, 2013, 1–22. (2013).

[41] G. Danezis, L. Kokoris-Kogias, A. Sonnino, and A. Spiegelman, "Narwhal and tusk: A dag-based mempool and efficient BFT consensus," in *ACM European Conference on Computer Systems (EuroSys)*, 34–50. (2022), 2022.

[42] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *ACM International Conference on Management of Data (SIGMOD)*, 123–140. (2019), 2019.

[43] D. Dolev and H. R. Strong, "Distributed commit with bounded waiting," in *IEEE Symposium on Reliability in Distributed Software and Database Systems*, 53–59. (1982), Jul. 1982.

[44] D. Dolev, C. Dwork, and L. Stockmeyer, "On the minimal synchronism needed for distributed consensus," *Journal of the ACM (JACM)*, vol. 34, no. 1, 1987, 77–97. (1987).

[45] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl, "Reaching approximate agreement in the presence of faults," *Journal of the ACM (JACM)*, vol. 33, no. 3, 1986, 499–516. (1986).

[46] C. Dwork, N. Lynch, and L. Stockmeyer, "Consensus in the presence of partial synchrony," *Journal of the ACM (JACM)*, vol. 35, no. 2, 1988, 288–323. (1988).

[47] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, "Proofs of Space," in *Advances in Cryptology (CRYPTO)*, 585–605. (2015), 2015.

[48] R. Elmasri and S. B. Navathe, *Database systems*, vol. 9. Pearson Education. (2011), 2011.

[49] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM (JACM)*, vol. 32, no. 2, 1985, 374–382. (1985).

[50] H. Garcia-Molina and D. Barbara, "How to assign votes in a distributed system," *Journal of the ACM (JACM)*, vol. 32, no. 4, 1985, 841–860. (1985).

[51] S. Gazzaz, V. Chakraborty, and F. Nawab, "Croesus: Multi-stage processing and transactions for video-analytics in edge-cloud systems," in *IEEE International Conference on Data Engineering (ICDE)*, 1463–1476. (2022), 2022.

[52] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," in *ACM Symposium on Operating Systems Principles (SOSP)*, 29–43. (2003), 2003.

[53] D. K. Gifford, "Weighted voting for replicated data," in *ACM Symposium on Operating Systems Principles (SOSP)*, 150–162. (1979), 1979.

[54] G. Golan Gueta, I. Abraham, S. Grossman, D. Malkhi, B. Pinkas, M. Reiter, D.-A. Seredinschi, O. Tamir, and A. Tomescu, "SBFT: A scalable and decentralized trust infrastructure," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 568–580. (2019), 2019.

[55] 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, 2015, 2711–2724. (2015).

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

[57] J. Gray, “The transaction concept: Virtues and limitations (invited paper),” in *International Conference on Very Large Data Bases (VLDB)*, 144–154. (1981), 1981.

[58] J. Gray and L. Lamport, “Consensus on transaction commit,” *ACM Transactions on Database Systems (TODS)*, vol. 31, no. 1, 2006, 133–160. (2006).

[59] R. Guerraoui, “Revisiting the relationship between non-blocking atomic commitment and consensus,” in *International Workshop on Distributed Algorithms*, 87–100. (1995), 1995.

[60] H. S. Gunawi, M. Hao, R. O. Suminto, A. Laksono, A. D. Satria, J. Adityatama, and K. J. Eliazar, “Why does the cloud stop computing? lessons from hundreds of service outages,” in *ACM Symposium on Cloud Computing (SoCC)*, 1–16. (2016), 2016.

[61] S. Gupta, M. J. Amiri, and M. Sadoghi, “Chemistry behind agreement,” in *Conference on Innovative Data Systems Research (CIDR). (2023)*, 2023.

[62] 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)*, 301–312. (2021), 2021.

[63] S. Gupta, J. Hellings, and M. Sadoghi, “Brief announcement: Revisiting consensus protocols through wait-free parallelization,” in *International Symposium on Distributed Computing (DISC)*, vol. 146, 44:1–44:3. (2019), 2019.

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

[65] 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)*, 1392–1403. (2021), 2021.

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

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

[68] S. Gupta, S. Rahnama, S. Pandey, N. Crooks, and M. Sadoghi, "Dissecting BFT consensus: In trusted components we trust!" In *ACM European Conference on Computer Systems (EuroSys)*, 2023.

[69] S. Gupta, S. Rahnama, and M. Sadoghi, "Permissioned blockchain through the looking glass: Architectural and implementation lessons learned," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 754–764, 2020.

[70] S. Gupta and M. Sadoghi, "EasyCommit: A non-blocking two-phase commit protocol," in *International Conference on Extending Database Technology (EDBT)*, 157–168. (2018), 2018.

[71] V. Hadzilacos and S. Toueg, "A modular approach to fault-tolerant broadcasts and related problems," Cornell University. (1994), Tech. Rep., 1994.

[72] T. Haerder and A. Reuter, "Principles of transaction-oriented database recovery," *ACM computing surveys (CSUR)*, vol. 15, no. 4, 1983, 287–317. (1983).

[73] J. M. Hellerstein, J. Faleiro, J. E. Gonzalez, J. Schleier-Smith, V. Sreekanti, A. Tumanov, and C. Wu, "Serverless computing: One step forward, two steps back," in *Conference on Innovative Data Systems Research (CIDR). (2019)*, 2019.

[74] J. Hellings, S. Gupta, S. Rahnama, and M. Sadoghi, "On the correctness of speculative consensus," *CoRR*, vol. abs/2204.03552. (2022), 2022. arXiv: 2204.03552.

[75] J. Hellings, D. P. Hughes, J. Primero, and M. Sadoghi, "Cerberus: Minimalistic multi-shard byzantine-resilient transaction processing," *CoRR*, vol. abs/2008.04450. (2020), 2020. arXiv: 2008.04450.

[76] J. Hellings and M. Sadoghi, "Brief announcement: The fault-tolerant cluster-sending problem," in *International Symposium on Distributed Computing (DISC)*, 45:1–45:3. (2019), 2019.

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

[78] J. Hellings and M. Sadoghi, "Byzantine cluster-sending in expected constant communication," *CoRR*, vol. abs/2108.08541. (2021), 2021. arXiv: 2108.08541.

[79] J. Hellings and M. Sadoghi, "The fault-tolerant cluster-sending problem," in *International Symposium on Foundations of Information and Knowledge Systems (FoIKS)*, 168–186. (2022), 2022.

[80] M. Herlihy, L. Shrira, and B. Liskov, "Cross-chain deals and adversarial commerce," *Proceedings of the VLDB Endowment*, vol. 13, no. 2, 2019, 100–113. (2019).

[81] H. Howard, A. Charapko, and R. Mortier, "Fast flexible paxos: Relaxing quorum intersection for fast paxos," in *International Conference on Distributed Computing and Networking*, 186–190. (2021), 2021.

[82] H. Howard, D. Malkhi, and A. Spiegelman, "Flexible paxos: Quorum intersection revisited," in *International Conference on Principles of Distributed Systems (OPODIS)*, vol. 70, 25:1–25:14. (2016), 2017.

[83] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, "ZooKeeper: Wait-free coordination for internet-scale systems," in *USENIX Annual Technical Conference (ATC). (2010)*, 2010.

[84] E. Jonas, J. Schleier-Smith, V. Sreekanti, C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. J. Yadwadkar, J. E. Gonzalez, R. A. Popa, I. Stoica, and D. A. Patterson, "Cloud programming simplified: A berkeley view on serverless computing," *CoRR*, vol. abs/1902.03383. (2019), 2019. arXiv: 1902.03383.

[85] 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)*, 245–256. (2011), 2011.

[86] M. F. Kaashoek and A. S. Tanenbaum, "Group communication in the amoeba distributed operating system," in *International Conference on Distributed Computing Systems*, 222–230. (1991), 1991.

[87] M. Kazhamiaka, B. Memon, C. Kankanamge, S. Sahu, S. Rizvi, B. Wong, and K. Daudjee, "Sift: Resource-efficient consensus with rdma," in *International Conference on Emerging Networking Experiments And Technologies*, 260–271. (2019), 2019.

[88] I. Keidar, E. Kokoris-Kogias, O. Naor, and A. Spiegelman, "All you need is DAG," in *ACM Symposium on Principles of Distributed Computing (PODC)*, A. Miller, K. Censor-Hillel, and J. H. Korhonen, Eds., 165–175. (2021), 2021.

[89] B. Kemme, R. Jiménez-Peris, and M. Patiño-Martínez, "Database replication," *Synthesis Lectures on Data Management*, vol. 5, no. 1, 2010, pp. 1–153.

[90] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure Proof-of-Stake blockchain protocol," in *Advances in Cryptology (CRYPTO)*, 357–388. (2017), Springer, 2017.

[91] S. King and S. Nadal, *PPCoin: Peer-to-peer crypto-currency with Proof-of-Stake. (2012)*, 2012. URL: https://www.peercoin.net/whitepapers/peercoin-paper.pdf.

[92] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *IEEE Symposium on Security and Privacy (S&P)*, 583–598. (2018), 2018.

[93] J. Kończak, P. T. Wojciechowski, N. Santos, T. Żurkowski, and A. Schiper, "Recovery algorithms for paxos-based state machine replication," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, 2019, 623–640. (2019).

[94] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative byzantine fault tolerance," *ACM Transactions on Computing Systems*, vol. 27, no. 4, 2009, 7:1–7:39. (2009).

[95] 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)*, 113–126. (2013), 2013.

[96] P. Kuznetsov, A. Tonkikh, and Y. X. Zhang, "Revisiting optimal resilience of fast byzantine consensus," in *ACM Symposium on Principles of Distributed Computing (PODC)*, 343–353. (2021), 2021.

[97] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM (CACM)*, vol. 21, no. 7, 1978, 558–565. (1978).

[98] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems*, vol. 16, no. 2, 1998, 133–169. (1998).

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

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

[101] L. Lamport, "Fast paxos," *Distributed Computing*, vol. 19, no. 2, 2006, 79–103. (2006).

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

[103] L. Lamport, D. Malkhi, and L. Zhou, "Vertical paxos and primary-backup replication," in *ACM symposium on Principles of Distributed Computing (PODC)*, 312–313. (2009), 2009.

[104] L. Lamport, D. Malkhi, and L. Zhou, "Reconfiguring a state machine," *ACM SIGACT News*, vol. 41, no. 1, 2010, 63–73. (2010).

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

[106] L. Lamport, R. Shortak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, 1982, 382–401. (1982).

[107] B. Lampson and D. Lomet, "A new presumed commit optimization for two phase commit," in *International Conference on Very Large Data Bases (VLDB)*, 630–640. (1993), 1993.

[108] 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.

[109] B. W. Lampson, "How to build a highly available system using consensus," in *International Workshop on Distributed Algorithms*, 1–17. (1996), 1996.

[110] J. Li, E. Michael, N. K. Sharma, A. Szekeres, and D. R. Ports, "Just say NO to paxos overhead: Replacing consensus with network ordering," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 467–483. (2016), 2016.

[111] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing Proof-of-Stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, 297–315. (2017), Springer, 2017.

[112] 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)*, 103–115. (2006), 2006.

[113] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *ACM SIGSAC Conference on Computer and Communications Security*, 17–30. (2016), 2016.

[114] H. Mahmoud, F. Nawab, A. Pucher, D. Agrawal, and A. El Abbadi, "Low-latency multi-datacenter databases using replicated commit," *Proceedings of the VLDB Endowment*, vol. 6, no. 9, 2013, 661–672. (2013).

[115] S. Maiyya, F. Nawab, D. Agrawal, and A. E. Abbadi, "Unifying consensus and atomic commitment for effective cloud data management," *Proceedings of the VLDB Endowment*, vol. 12, no. 5, 2019, 611–623. (2019).

[116] 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)*, 369–384. (2008), 2008.

[117] P. J. Marandi, M. Primi, N. Schiper, and F. Pedone, "Ring paxos: A high-throughput atomic broadcast protocol," in *IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, 527–536. (2010), 2010.
[118] M. Mihai Letia, N. Preguica, and M. Shapiro, "Crdts: Consistency without concurrency control," *RR-6956, INRIA. (2009)*, 2009.
[119] N. Mittal and F. Nawab, "Coolsm: Distributed and cooperative indexing across edge and cloud machines," in *IEEE International Conference on Data Engineering (ICDE)*, 420–431. (2021), 2021.
[120] C. Mohan and B. Lindsay, "Efficient commit protocols for the tree of processes model of distributed transactions," *ACM SIGOPS Operating Systems Review*, vol. 19, no. 2, 1985, 40–52. (1985).
[121] C. Mohan, R. Strong, and S. Finkelstein, "Method for distributed transaction commit and recovery using byzantine agreement within clusters of processors," in *Proceedings of the annual ACM symposium on Principles of distributed computing (PODC)*, 89–103 (1983), 1983.
[122] I. Moraru, D. G. Andersen, and M. Kaminsky, "There is more consensus in egalitarian parliaments," in *ACM Symposium on Operating Systems Principles (SOSP)*, 358–372. (2013), 2013.
[123] S. Mu, L. Nelson, W. Lloyd, and J. Li, "Consolidating concurrency control and consensus for commits under conflicts," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 517–532. (2016), 2016.
[124] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2009. URL: https://bitcoin.org/bitcoin.pdf.
[125] F. Nawab, "Wedgechain: A trusted edge-cloud store with asynchronous (lazy) trust," in *IEEE International Conference on Data Engineering (ICDE)*, 408–419. (2021), 2021.
[126] 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)*, 1221–1236. (2018), 2018.

[127] 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)*, 8–17. (1988), 1988.

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

[129] M. T. Ozsu and P. Valduriez, *Principles of distributed database systems*, vol. 2. Springer. (1999), 1999.

[130] S. Park, A. Kwon, G. Fuchsbauer, P. Gaži, J. Alwen, and K. Pietrzak, "SpaceMint: A cryptocurrency based on proofs of space," in *Financial Cryptography and Data Security*, 480–499. (2018), Springer, 2018.

[131] S. Patterson, A. J. Elmore, F. Nawab, D. Agrawal, and A. E. Abbadi, "Serializability, not serial: Concurrency control and availability in multi-datacenter datastores," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, 2012, 1459–1470. (2012).

[132] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM (JACM)*, vol. 27, no. 2, 1980, 228–234. (1980).

[133] F. Pedone, R. Guerraoui, and A. Schiper, "Exploiting atomic broadcast in replicated databases," in *European conference on parallel processing (Euro-Par)*, 513–520. (1998), 1998.

[134] F. Pedone, R. Guerraoui, and A. Schiper, "The database state machine approach," *Distributed and Parallel Databases (DAPD)*, vol. 14, no. 1, 2003, 71–98. (2003).

[135] 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)*, 2:298–2:311. (2022), 2022.

[136] R. Ramakrishnan, J. Gehrke, and J. Gehrke, *Database management systems*, vol. 3. McGraw-Hill New York. (2003), 2003.

[137] M. Sadoghi and S. Blanas, *Transaction Processing on Modern Hardware*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers. (2019), 2019.

[138] N. Santos and A. Schiper, "Optimizing paxos with batching and pipelining," *Theoretical Computer Science*, vol. 496, 2013, 170–183. (2013).

[139] F. B. Schneider, "Implementing fault-tolerant services using the state machine approach: A tutorial," *ACM Computing Surveys (CSUR)*, vol. 22, no. 4, 1990, 299–319. (1990).

[140] A. Sharov and A. S. A. M. M. Stokely, "Take me to your leader! online optimization of distributed storage configurations," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, 2015, 1490–1501. (2015).

[141] D. Skeen, "Nonblocking commit protocols," in *ACM International Conference on Management of Data (SIGMOD)*, 133–142. (1981), 1981.

[142] D. Skeen and M. Stonebraker, "A formal model of crash recovery in a distributed system," *IEEE Transactions on Software Engineering*, no. 3, 1983, 219–228. (1983).

[143] J. Sousa and A. Bessani, "Separating the wheat from the chaff: An empirical design for geo-replicated state machines," in *IEEE Symposium on Reliable Distributed Systems (SRDS)*, 146–155. (2015), 2015.

[144] A. Spiegelman, N. Giridharan, A. Sonnino, and L. Kokoris-Kogias, "Bullshark: DAG BFT protocols made practical," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2705–2718. (2022), 2022.

[145] I. Stanoi, D. Agrawal, and A. El Abbadi, "Using broadcast primitives in replicated databases," in *International Conference on Distributed Computing Systems (ICDCS)*, 148–155 (1998), 1998.

[146] R. H. Thomas, "A majority consensus approach to concurrency control for multiple copy databases," *ACM Transactions on Database Systems (TODS)*, vol. 4, no. 2, 1979, 180–209. (1979).

[147] A. Thomson, T. Diamond, S.-C. Weng, K. Ren, P. Shao, and D. J. Abadi, "Calvin: Fast distributed transactions for partitioned database systems," in *ACM International Conference on Management of Data (SIGMOD)*, 1–12. (2012), 2012.

[148] R. Van Renesse and D. Altinbuken, “Paxos made moderately complex,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, 2015, 42:1–42:36. (2015).

[149] 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, 2014, 472–484. (2014).

[150] G. S. Veronese, M. Correia, A. N. Bessani, and L. C. Lung, “Spin one’s wheels? byzantine fault tolerance with a spinning primary,” in *IEEE Symposium on Reliable Distributed Systems (SRDS)*, 135–144. (2009), 2009.

[151] M. Whittaker, “Compartmentalizing state machine replication,” Ph.D. dissertation, University of California, Berkeley. (2021), 2021.

[152] M. Whittaker, N. Giridharan, A. Szekeres, J. Hellerstein, H. Howard, F. Nawab, and I. Stoica, “Matchmaker Paxos: A Reconfigurable Consensus Protocol,” *Journal of Systems Research (JSys)*, vol. 1, no. 1, Sep. 2021.

[153] M. J. Whittaker, N. Giridharan, A. Szekeres, J. M. Hellerstein, H. Howard, F. Nawab, and I. Stoica, “Matchmaker paxos: A reconfigurable consensus protocol [technical report],” *CoRR*, vol. abs/2007.09468. (2020), 2020. arXiv: 2007.09468.

[154] M. J. Whittaker, N. Giridharan, A. Szekeres, J. M. Hellerstein, and I. Stoica, “Bipartisan paxos: A modular state machine replication protocol,” *CoRR*, vol. abs/2003.00331. (2020), 2020. arXiv: 2003.00331.

[155] G. Wood, *Ethereum: A secure decentralised generalised transaction ledger*, 2016. URL: https://gavwood.com/paper.pdf.

[156] Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, and H. V. Madhyastha, “Spanstore: Cost-effective geo-replicated storage spanning multiple cloud services,” in *ACM Symposium on Operating Systems Principles (SOSP)*, 292–308. (2013), 2013.

[157] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, “HotStuff: BFT consensus with linearity and responsiveness,” in *ACM Symposium on Principles of Distributed Computing (PODC)*, 347–356. (2019), 2019.

[158] V. Zakhary, D. Agrawal, and A. E. Abbadi, "Atomic commitment across blockchains," *Proceedings of the VLDB Endowment*, vol. 13, no. 9, 2020, 1319–1331. (2020).

[159] V. Zakhary, F. Nawab, D. Agrawal, and A. El Abbadi, "Global-scale placement of transactional data stores.," in *International Conference on Extending Database Technology (EDBT)*, 385–396. (2018), 2018.

[160] V. Zakhary, F. Nawab, D. Agrawal, and A. El Abbadi, "Db-risk: The game of global database placement," in *International Conference on Management of Data (SIGMOD)*, 2185–2188. (2016), 2016.

[161] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 931–948. (2018), 2018.

[162] I. Zhang, N. K. Sharma, A. Szekeres, A. Krishnamurthy, and D. R. Ports, "Building consistent transactions with inconsistent replication," *ACM Transactions on Computer Systems (TOCS)*, vol. 35, no. 4, 2018, 1–37. (2018).