REFERENCES

[1] [n.d.]. DBx1000. https://github.com/yxymit/DBx1000.

[2] [n.d.]. DBx1000 with Bamboo Implemented. https://github.com/ScarletGuo/Bamboo-Public.

[3] Divyakant Agrawal, Amr El Abbadi, Richard Jeffers, and Lijing Lin. 1995. Ordered shared locks for real-time databases. *The VLDB Journal* 4, 1 (1995), 87–126.

[4] Philip A Bernstein, Philip A Bernstein, and Nathan Goodman. 1981. Concurrency control in distributed database systems. *ACM Computing Surveys (CSUR)* 13, 2 (1981), 185–221.

[5] Philip A. Bernstein and Nathan Goodman. 1981. Concurrency Control in Distributed Database Systems. *CSUR* (1981), 185–221.

[6] Philip A Bernstein and Eric Newcomer. 2009. *Principles of transaction processing*. Morgan Kaufmann.

[7] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. 2010. Benchmarking Cloud Serving Systems with YCSB. In *SoCC*. 143–154.

[8] James C. Corbett and et al. 2012. Spanner: Google's Globally-Distributed Database. In *OSDI*. 251–264.

[9] Cristian Diaconu, Craig Freedman, Erik Ismert, Per-Ake Larson, Pravin Mittal, Ryan Stonecipher, Nitin Verma, and Mike Zwilling. 2013. Hekaton: SQL Server's Memory-Optimized OLTP Engine. In *SIGMOD*. 1243–1254.

[10] Bailu Ding, Lucja Kot, and Johannes Gehrke. 2018. Improving optimistic concurrency control through transaction batching and operation reordering. *Proceedings of the VLDB Endowment* 12, 2 (2018), 169–182.

[11] Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, Aditya Akella, Kuangching Wang, Glenn Ricart, Larry Landweber, Chip Elliott, Michael Zink, Emmanuel Cecchet, Snigdhaswin Kar, and Prabodh Mishra. 2019. The Design and Operation of CloudLab. In *Proceedings of the USENIX Annual Technical Conference (ATC)*. 1–14. https://www.flux.utah.edu/paper/duplyakin-atc19

[12] Tamer Eldeeb and Phil Bernstein. 2016. *Transactions for Distributed Actors in the Cloud*. Technical Report.

[13] K. P. Eswaran, J. N. Gray, R. A. Lorie, and I. L. Traiger. 1976. The Notions of Consistency and Predicate Locks in a Database System. *CACM* (1976), 624–633.

[14] Jose M. Faleiro and Daniel J. Abadi. 2015. Rethinking Serializable Multiversion Concurrency Control. *PVLDB* (2015), 1190–1201.

[15] Jose M Faleiro, Daniel J Abadi, and Joseph M Hellerstein. 2017. High performance transactions via early write visibility. *Proceedings of the VLDB Endowment* 10, 5 (2017), 613–624.

[16] Dieter Gawlick and David Kinkade. 1985. Varieties of concurrency control in IMS/VS fast path. *IEEE Database Eng. Bull.* 8, 2 (1985), 3–10.

[17] Goetz Graefe, Mark Lillibridge, Harumi Kuno, Joseph Tucek, and Alistair Veitch. 2013. Controlled lock violation. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 85–96.

[18] J. Gray, Pete Homan, H. Korth, and R. Obermarck. 1981. A Straw Man Analysis of the Probability of Waiting and Deadlock in a Database System. In *Berkeley Workshop*.

[19] Jim Gray and Andreas Reuter. 1992. *Transaction Processing: Concepts and Techniques* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[20] Zhihan Guo, Kan Wu, Cong Yan, and Xiangyao Yu. 2021. Releasing Locks As Early As You Can: Reducing Contention of Hotspots by Violating Two-Phase Locking (Extended Version). arXiv:2103.09906 [cs.DB]

[21] Ramesh Gupta, Jayant Haritsa, and Krithi Ramamritham. 1997. Revisiting Commit Processing in Distributed Database Systems. In *SIGMOD*. 486–497.

[22] Mark C. Jeffrey, Suvinay Subramanian, Cong Yan, Joel Emer, and Daniel Sanchez. 2015. A Scalable Architecture for Ordered Parallelism. In *MICRO*. 228–241.

[23] M. C. Jeffrey, S. Subramanian, C. Yan, J. Emer, and D. Sanchez. 2016. Unlocking Ordered Parallelism with the Swarm Architecture. *IEEE Micro* (2016), 105–117.

[24] Ryan Johnson, Ippokratis Pandis, Radu Stoica, Manos Athanassoulis, and Anastasia Ailamaki. 2010. Aether: a scalable approach to logging. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 681–692.

[25] Evan P.C. Jones, Daniel J. Abadi, and Samuel Madden. 2010. Low Overhead Concurrency Control for Partitioned Main Memory Databases. In *SIGMOD*. 603–614.

[26] Hideaki Kimura, Goetz Graefe, and Harumi A Kuno. 2012. Efficient locking techniques for databases on modern hardware.. In *ADMS@ VLDB*. 1–12.

[27] Hsiang-Tsung Kung and John T Robinson. 1981. On optimistic methods for concurrency control. *ACM Transactions on Database Systems (TODS)* 6, 2 (1981), 213–226.

[28] Per-Åke Larson, Spyros Blanas, Cristian Diaconu, Craig Freedman, Jignesh M. Patel, and Mike Zwilling. 2011. High-Performance Concurrency Control Mechanisms for Main-Memory Databases. *VLDB* (2011), 298–309.

[29] David Lomet, Alan Fekete, Rui Wang, and Peter Ward. 2012. Multi-Version Concurrency via Timestamp Range Conflict Management. In *ICDE*. 714–725.

[30] Dahlia Malkhi and Jean-Philippe Martin. 2013. Spanner's concurrency control. *ACM SIGACT News* 44, 3 (2013), 73–77.

[31] C Mohan. 1990. *ARIES/KVL: A Key-Value Locking Method for Concurrency Control of Multiaction Transactions Operating on B-Tree Indexes*. VLDB.

[32] Shuai Mu, Sebastian Angel, and Dennis Shasha. 2019. Deferred runtime pipelining for contentious multicore software transactions. In *Proceedings of the Fourteenth EuroSys Conference 2019*. 1–16.

[33] Flemming Nielson, Hanne R. Nielson, and Chris Hankin. 2010. *Principles of Program Analysis*. Springer Publishing Company, Incorporated.

[34] Daniel J Rosenkrantz, Richard E Stearns, and Philip M Lewis. 1978. System Level Concurrency Control for Distributed Database Systems. *ACM Transactions on Database Systems (TODS)* 3, 2 (1978), 178–198.

[35] Mohammad Sadoghi, Mustafa Canim, Bishwaranjan Bhattacharjee, Fabian Nagel, and Kenneth A. Ross. 2014. Reducing Database Locking Contention through Multi-Version Concurrency. *Proc. VLDB Endow.* 7, 13 (Aug. 2014), 1331–1342. https://doi.org/10.14778/2733004.2733006

[36] Dennis Shasha, Francois Llirbat, Eric Simon, and Patrick Valduriez. 1995. Transaction Chopping: Algorithms and Performance Studies. *ACM Transactions on Database Systems (TODS)* 20, 3 (1995), 325–363.

[37] Eljas Soisalon-Soininen and Tatu Ylönen. 1995. Partial strictness in two-phase locking. In *International Conference on Database Theory*. Springer, 139–147.

[38] Dixin Tang and Aaron J Elmore. 2018. Toward coordination-free and reconfigurable mixed concurrency control. In *2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18)*. 809–822.

[39] The Transaction Processing Council. 2007. TPC-C Benchmark (Revision 5.9.0).

[40] Alexander Thomasian. 1993. Two-phase locking performance and its thrashing behavior. *ACM Transactions on Database Systems (TODS)* 18, 4 (1993), 579–625.

[41] Stephen Tu, Wenting Zheng, Eddie Kohler, Barbara Liskov, and Samuel Madden. 2013. Speedy Transactions in Multicore In-Memory Databases. In *SOSP*.

[42] Tianzheng Wang and Hideaki Kimura. 2016. Mostly-optimistic concurrency control for highly contended dynamic workloads on a thousand cores. *Proceedings of the VLDB Endowment* 10, 2 (2016), 49–60.

[43] Zhaoguo Wang, Shuai Mu, Yang Cui, Han Yi, Haibo Chen, and Jinyang Li. 2016. Scaling multicore databases via constrained parallel execution. In *Proceedings of the 2016 International Conference on Management of Data*. 1643–1658.

[44] Gerhard Weikum and Gottfried Vossen. 2001. *Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery*. Elsevier.

[45] Chao Xie, Chunzhi Su, Cody Littley, Lorenzo Alvisi, Manos Kapritsos, and Yang Wang. 2015. High-performance ACID via modular concurrency control. In *Proceedings of the 25th Symposium on Operating Systems Principles*. 279–294.

[46] Cong Yan and Alvin Cheung. 2016. Leveraging Lock Contention to Improve OLTP Application Performance. *Proceedings of the VLDB Endowment* 9, 5 (2016), 444–455.

[47] Xiangyao Yu, George Bezerra, Andrew Pavlo, Srinivas Devadas, and Michael Stonebraker. 2014. Staring into the Abyss: An Evaluation of Concurrency Control with One Thousand Cores. *VLDB*, 209–220.

[48] Yang Zhang, Russell Power, Siyuan Zhou, Yair Sovran, Marcos K Aguilera, and Jinyang Li. 2013. Transaction chains: achieving serializability with low latency in geo-distributed storage systems. In *SOSP*. 276–291.