CHAPTER 18

Concurrency Control

We saw in Chapter 17 that one of the fundamental properties of a transaction is isolation. When several transactions execute concurrently in the database, however, the isolation property may no longer be preserved. To ensure that it is, the system must control the interaction among the concurrent transactions; this control is achieved through one of a variety of mechanisms called *concurrency-control* schemes. In this chapter, we consider the management of concurrently executing transactions, and we ignore failures. In Chapter 19, we shall see how the system can recover from failures.

As we shall see, there are a variety of concurrency-control schemes. No one scheme is clearly the best; each one has advantages. In practice, the most frequently used schemes are *two-phase locking* and *snapshot isolation*.

Bibliographical Notes

[Gray and Reuter (1993)] provides detailed textbook coverage of transaction-processing concepts, including concurrency-control concepts and implementation details. [Bernstein and Newcomer (2009)] provides textbook coverage of various aspects of transaction processing including concurrency control.

The two-phase locking protocol was introduced by [Eswaran et al. (1976)]. The tree-locking protocol is from [Silberschatz and Kedem (1980)]. Other non-two-phase locking protocols that operate on more general graphs are described in [Yannakakis et al. (1979)], [Kedem and Silberschatz (1983)], and [Buckley and Silberschatz (1985)]. [Korth (1983)] explores various lock modes that can be obtained from the basic shared and exclusive lock modes.

The locking protocol for multiple-granularity data items is from [Gray et al. (1975)]. A detailed description is presented by [Gray et al. (1976)]. [Kedem and Silberschatz (1983)] formalizes multiple-granularity locking for an arbitrary collection of lock modes (allowing for more semantics than simply read and write). This approach includes a class of lock modes called *update* modes to deal with lock conversion. [Carey (1983)] extends the multiple-granularity idea to timestamp-based concurrency control. An extension of the protocol to ensure deadlock freedom is presented by [Korth (1982)].

The timestamp-based concurrency-control scheme is from [Reed (1983)]. A timestamp algorithm that does not require any rollback to ensure serializability is presented by [Buckley and Silberschatz (1983)]. The validation concurrency-control scheme is from [Kung and Robinson (1981)].

Multiversion timestamp order was introduced in [Reed (1983)]. A multiversion tree-locking algorithm appears in [Silberschatz (1982)].

Degree-two consistency was introduced in [Gray et al. (1975)]. The levels of consistency—or isolation—offered in SQL are explained and critiqued in [Berenson et al. (1995)]; the snapshot-isolation technique was also introduced in the same paper. Serializable snapshot-isolation was introduced by [Cahill et al. (2009)]; [Ports and Grittner (2012)] describes the implementation of serializable snapshot isolation in PostgreSQL.

[Fekete et al. (2005)] describes how to ensure serializable executions under snapshot isolation, by rewriting certain transactions to introduce conflicts; these conflicts ensure that the transactions cannot run concurrently under snapshot isolation; [Jorwekar et al. (2007)] describes an approach, that given a set of (parameterized) transactions running under snapshot isolation, can check if the transactions are vulnerable to nonserializability.

Concurrency in B$^+$-trees was studied by [Bayer and Schkolnick (1977)] and [Johnson and Shasha (1993)]. The techniques presented in Section 18.10.2 are based on [Kung and Lehman (1980)] and [Lehman and Yao (1981)]. The technique of key-value locking used in ARIES provides for very high concurrency on B$^+$-tree access and is described in [Mohan (1990)] and [Mohan and Narang (1992)]. [Ellis (1987)] presents a concurrency-control technique for linear hashing.

[Faerber et al. (2017)] provide a survey of main-memory databases, including coverage of concurrency control in main-memory databases. Main memory concurrency control is discussed in [Larson et al. (2011)] and [Neumann et al. (2015)]. While many techniques for avoiding phantoms in validation based techniques perform a rescan of indices to check if there are any new records that were missed in the earlier scan, [Neumann et al. (2015)] take a different approach optimized for large in-memory scans which are common in main-memory databases. In their technique, scan predicates are recorded, and all updates performed by concurrent transactions are checked against the scan predicates to check if they satisfy the predicate. Since updates are generally far fewer than record reads, this approach is usually more efficient. The Bw-Tree data structure [Levandoski et al. (2013)] provides latch-free access to indices that can be main-memory resident, or resident on flash.

Bibliography

[Bayer and Schkolnick (1977)] R. Bayer and M. Schkolnick, "Concurrency of Operating on B-trees", *Acta Informatica*, Volume 9, Number 1 (1977), pages 1–21.

[**Berenson et al. (1995)**] H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O'Neil, and P. O'Neil, "A Critique of ANSI SQL Isolation Levels", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1995), pages 1–10.

[**Bernstein and Newcomer (2009)**] P. A. Bernstein and E. Newcomer, *Principles of Transaction Processing*, 2nd edition, Morgan Kaufmann (2009).

[**Buckley and Silberschatz (1983)**] G. Buckley and A. Silberschatz, "Obtaining Progressive Protocols for a Simple Multiversion Database Model", In *Proc. of the International Conf. on Very Large Databases* (1983), pages 74–80.

[**Buckley and Silberschatz (1985)**] G. Buckley and A. Silberschatz, "Beyond Two-Phase Locking", *Journal of the ACM*, Volume 32, Number 2 (1985), pages 314–326.

[**Cahill et al. (2009)**] M. J. Cahill, U. Röhm, and A. D. Fekete, "Serializable isolation for snapshot databases", *ACM Transactions on Database Systems*, Volume 34, Number 4 (2009), pages 20:1–20:42.

[**Carey (1983)**] M. J. Carey, "Granularity Hierarchies in Concurrency Control", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1983), pages 156–165.

[**Ellis (1987)**] C. S. Ellis, "Concurrency in Linear Hashing", *ACM Transactions on Database Systems*, Volume 12, Number 2 (1987), pages 195–217.

[**Eswaran et al. (1976)**] K. P. Eswaran, J. N. Gray, R. A. Lorie, and I. L. Traiger, "The Notions of Consistency and Predicate Locks in a Database System", *Communications of the ACM*, Volume 19, Number 11 (1976), pages 624–633.

[**Faerber et al. (2017)**] F. Faerber, A. Kemper, P.-A. Larson, J. Levandoski, T. Neumann, and A. Pavlo, "Main Memory Database Systems", *Foundations and Trends in Databases*, Volume 8, Number 1-2 (2017), pages 1–130.

[**Fekete et al. (2005)**] A. Fekete, D. Liarokapis, E. O'Neil, P. O'Neil, and D. Shasha, "Making Snapshot Isolation Serializable", *ACM Transactions on Database Systems*, Volume 30, Number 2 (2005), pages 492–528.

[**Gray and Reuter (1993)**] J. Gray and A. Reuter, *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann (1993).

[**Gray et al. (1975)**] J. Gray, R. A. Lorie, and G. R. Putzolu, "Granularity of Locks and Degrees of Consistency in a Shared Data Base", In *Proc. of the International Conf. on Very Large Databases* (1975), pages 428–451.

[**Gray et al. (1976)**] J. Gray, R. A. Lorie, G. R. Putzolu, and I. L. Traiger, *Granularity of Locks and Degrees of Consistency in a Shared Data Base*, Nijssen (1976).

[**Johnson and Shasha (1993)**] T. Johnson and D. Shasha, "The Performance of Concurrent B-Tree Algorithms", *ACM Transactions on Database Systems*, Volume 18, Number 1 (1993), pages 51–101.

[**Jorwekar et al. (2007)**] S. Jorwekar, A. Fekete, K. Ramamritham, and S. Sudarshan, "Automating the Detection of Snapshot Isolation Anomalies", In *Proc. of the International Conf. on Very Large Databases* (2007), pages 1263–1274.

[**Kedem and Silberschatz (1983)**] Z. M. Kedem and A. Silberschatz, "Locking Protocols: From Exclusive to Shared Locks", *Journal of the ACM*, Volume 30, Number 4 (1983), pages 787–804.

[**Korth (1982)**] H. F. Korth, "Deadlock Freedom Using Edge Locks", *ACM Transactions on Database Systems*, Volume 7, Number 4 (1982), pages 632–652.

[**Korth (1983)**] H. F. Korth, "Locking Primitives in a Database System", *Journal of the ACM*, Volume 30, Number 1 (1983), pages 55–79.

[**Kung and Lehman (1980)**] H. T. Kung and P. L. Lehman, "Concurrent Manipulation of Binary Search Trees", *ACM Transactions on Database Systems*, Volume 5, Number 3 (1980), pages 339–353.

[**Kung and Robinson (1981)**] H. T. Kung and J. T. Robinson, "Optimistic Concurrency Control", *ACM Transactions on Database Systems*, Volume 6, Number 2 (1981), pages 312–326.

[**Larson et al. (2011)**] P. Larson, S. Blanas, C. Diaconu, C. Freedman, J. Patel, and M. Zwilling, "High-Performance Concurrency Control Mechanisms for Main-Memory Databases", *PVLDB*, Volume 5, Number 4 (2011).

[**Lehman and Yao (1981)**] P. L. Lehman and S. B. Yao, "Efficient Locking for Concurrent Operations on B-trees", *ACM Transactions on Database Systems*, Volume 6, Number 4 (1981), pages 650–670.

[**Mohan (1990)**] C. Mohan, "ARIES/KVL: A Key-Value Locking Method for Concurrency Control of Multiaction Transactions Operations on B-Tree indexes", In *Proc. of the International Conf. on Very Large Databases* (1990), pages 392–405.

[**Mohan and Narang (1992)**] C. Mohan and I. Narang, "Efficient Locking and Caching of Data in the Multisystem Shared Disks Transaction Environment", In *Proc. of the International Conf. on Extending Database Technology* (1992), pages 453–468.

[**Neumann et al. (2015)**] T. Neumann, T. Mühlbauer, and A. Kemper, "Fast Serializable Multi-Version Concurrency Control for Main-Memory Database Systems", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (2015), pages 677–689.

[**Ports and Grittner (2012)**] D. R. K. Ports and K. Grittner, "Serializable Snapshot Isolation in PostgreSQL", *Proceedings of the VLDB Endowment*, Volume 5, Number 12 (2012), pages 1850–1861.

[**Reed (1983)**] D. Reed, "Implementing Atomic Actions on Decentralized Data", *Transactions on Computer Systems*, Volume 1, Number 1 (1983), pages 3–23.

[**Silberschatz (1982)**] A. Silberschatz, "A Multi-Version Concurrency Control Scheme With No Rollbacks", In *Proc. of the ACM Symposium on Principles of Distributed Computing* (1982), pages 216–223.

[**Silberschatz and Kedem (1980)**] A. Silberschatz and Z. Kedem, "Consistency in Hierarchical Database Systems", *Journal of the ACM*, Volume 27, Number 1 (1980), pages 72–80.

[**Yannakakis et al. (1979)**] M. Yannakakis, C. H. Papadimitriou, and H. T. Kung, "Locking Protocols: Safety and Freedom from Deadlock", In *Proc. of the IEEE Symposium on the Foun-*

dations of Computer Science (1979), pages 286–297.

Credits

The photo of the sailboats in the beginning of the chapter is due to ©Pavel Nesvadba/Shutterstock.