CHAPTER 23

Parallel and Distributed Transaction Processing

We studied transaction processing in centralized databases earlier, covering concurrency control in Chapter 18 and recovery in Chapter 19. In this chapter, we study how to carry out transaction processing in parallel and distributed databases. In addition to supporting concurrency control and recovery, transaction processing in parallel and distributed databases must also deal with issues due to replication of data, and of failures of some nodes.

Both parallel and distributed databases have multiple nodes, which can fail independently. The main difference between parallel and distributed databases from the view point of transaction processing is that the latency of remote access is much higher, and bandwidth lower, in a distributed database than in a parallel database where all nodes are in a single data center. Failures such as network partitioning and message delays are much less likely within a data center than across geographically distributed sites, but nevertheless they can occur; transaction processing must be done correctly even if they do occur.

Thus, most techniques for transaction processing are common to both parallel and distributed databases. In the few cases where there is a difference, we explicitly point out the difference. And as a result, in this chapter, whenever we say that a technique is applicable to distributed databases, it should be interpreted to mean that it is applicable to distributed databases as well as to parallel databases, unless we explicitly say otherwise.

In Section 23.1, we outline a model for transaction processing in a distributed database. In Section 23.2, we describe how to implement atomic transactions in a distributed database by using special commit protocols.

In Section 23.3 we describe how to extend traditional concurrency control techniques to distributed databases. Section 23.4 describes concurrency control techniques for the case where data items are replicated, while Section 23.5 describes further extensions including how multiversion concurrency control techniques can be extended to deal with distributed databases, and concurrency control can be implemented with

heterogeneous distributed databases. Replication with weak degrees of consistency is discussed in Section 23.6.

Most techniques for dealing with distributed data require the use of coordinators to ensure consistent and efficient transaction processing. In Section 23.7 we discuss how coordinators can be chosen in a distributed fashion, robust to failures. Finally, Section 23.8 describes the distributed consensus problem, outlines solutions for the problem, and then discusses how these solutions can be used to implement fault-tolerant services by means of replication of a log.

Bibliographical Notes

Textbook discussions of distributed databases are offered by [Ozsu and Valduriez (2010)]. [Breitbart et al. (1999b)] presents an overview of distributed databases.

The implementation of the transaction concept in a distributed database is presented by [Gray (1981)] and [Traiger et al. (1982)]. The 2PC protocol was developed by [Lampson and Sturgis (1976)]. The three-phase commit protocol is from [Skeen (1981)]. [Mohan and Lindsay (1983)] discusses two modified versions of 2PC, called *presume commit* and *presume abort*, that reduce the overhead of 2PC by defining default assumptions regarding the fate of transactions.

Distributed clock synchronization is discussed in [Lamport (1978)]. Distributed concurrency control is covered by [Bernstein and Goodman (1981)].

bibnotes on 3PC, including its non-blocking unsafe variants, and blocking variants. [Skeen (1981)] (Skeen 82a, 82b and 82c among others). Mention also extended 3PC to avoid blocking in the event of reconstitution of quorum after multiple failures. [Keidar and Dolev (1998)]

The problem of concurrent updates to replicated data was revisited in the context of data warehouses by [Gray et al. (1996)]. [Anderson et al. (1998)] discusses issues concerning lazy replication and consistency. [Breitbart et al. (1999a)] describes lazy update protocols for handling replication.

The user manuals of various database systems provide details of how they handle replication and consistency. [Huang and Garcia-Molina (2001)] addresses exactly-once semantics in a replicated messaging system.

[Knapp (1987)] surveys the distributed deadlock-detection literature.

Distributed optimistic concurrency control web.cs.ucdavis.edu/~wu/ecs251/ecs251_DMVOCC.pdf Distributed optimistic concurrency control with reduced rollback [Agrawal et al. (1987)]

Distributed snapshot isolation is described in [Binnig et al. (2014)] and [Schenkel et al. (1999)]. A distributed snapshot isolation technique with serializability checks based on timestamps, along wih an implementation of the techniques on HBase, is discussed in [Padhye and Tripathi (2015)].

Experience in building a database using Amazon's S3 cloud-based storage is described in [Brantner et al. (2008)]. An approach to making transactions work correctly in cloud systems is discussed in [Lomet et al. (2009)].

Transaction processing in federated database systems is discussed in [Mehrotra et al. (2001)]. The ticket scheme is presented in [Georgakopoulos et al. (1994)]. 2LSR is introduced in [Mehrotra et al. (1991)].

Techniques for combining concurrency control with commit protocols based on consensus, to reduce overheads, are described by [Kraska et al. (2013)], [I. Zhang and Ports (2015)] and [Mu et al. (2016)].

Bibliography

[Agrawal et al. (1987)] D. Agrawal, A. Bernstein, P. Gupta, and S. Sengupta, "Distributed optimistic concurrency control with reduced rollback", *Distributed Computing*, Volume 2, Number 1 (1987), pages 45–59.

[Anderson et al. (1998)] T. Anderson, Y. Breitbart, H. F. Korth, and A. Wool, "Replication, Consistency and Practicality: Are These Mutually Exclusive?", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1998), pages 484–495.

[Bernstein and Goodman (1981)] P. A. Bernstein and N. Goodman, "Concurrency Control in Distributed Database Systems", *ACM Computing Surveys*, Volume 13, Number 2 (1981), pages 185–221.

[Binnig et al. (2014)] C. Binnig, S. Hildenbrand, F. Furber, D. Kossmann, J. Lee, and N. May, "Distributed snapshot isolation: global transactions pay globally, local transactions pay locally", *VLDB Journal*, Volume 23, Number 6 (2014), pages 987–1011.

[Brantner et al. (2008)] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska, "Building a Database on S3", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (2008), pages 251–263.

[Breitbart et al. (1999a)] Y. Breitbart, R. Komondoor, R. Rastogi, S. Seshadri, and A. Silberschatz, "Update Propagation Protocols For Replicated Databases", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1999), pages 97–108.

[Breitbart et al. (1999b)] Y. Breitbart, H. Korth, A. Silberschatz, and S. Sudarshan. *Distributed Databases*. John Wiley and Sons (1999).

[Georgakopoulos et al. (1994)] D. Georgakopoulos, M. Rusinkiewicz, and A. Seth, "Using Tickets to Enforce the Serializability of Multidatabase Transactions", *IEEE Transactions on Knowledge and Data Engineering*, Volume 6, Number 1 (1994), pages 166–180.

[Gray (1981)] J. Gray, "The Transaction Concept: Virtues and Limitations", In *Proc. of the International Conf. on Very Large Databases* (1981), pages 144–154.

[Gray et al. (1996)] J. Gray, P. Helland, and P. O'Neil, "The Dangers of Replication and a Solution", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1996), pages 173–182.

[Huang and Garcia-Molina (2001)] Y. Huang and H. Garcia-Molina, "Exactly-once Semantics in a Replicated Messaging System", In *Proc. of the International Conf. on Data Engineering* (2001), pages 3–12.

[I. Zhang and Ports (2015)] A. S. A. K. I. Zhang, N. K. Sharma and D. R. K. Ports, "Building consistent transactions with inconsistent replication", In *Proc. of the ACM Symposium on Operating System Principles* (2015).

[Keidar and Dolev (1998)] I. Keidar and D. Dolev, "Increasing the Resilience of Distributed and Replicated Database Systems", *Journal of Computer and System Sciences*, Volume 57, Number 3 (1998), pages 309–324.

[Knapp (1987)] E. Knapp, "Deadlock Detection in Distributed Databases", *ACM Computing Surveys*, Volume 19, Number 4 (1987), pages 303–328.

[Kraska et al. (2013)] T. Kraska, G. Pang, M. J. Franklin, S. Madden, and A. Fekete, "MDCC: multi-data center consistency", In *Eurosys Conference* (2013), pages 113–126.

[Lamport (1978)] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System", *Communications of the ACM*, Volume 21, Number 7 (1978), pages 558–565.

[Lampson and Sturgis (1976)] B. Lampson and H. Sturgis, "Crash Recovery in a Distributed Data Storage System", Technical report, Computer Science Laboratory, Xerox Palo Alto Research Center, Palo Alto (1976).

[Lomet et al. (2009)] D. Lomet, A. Fekete, G. Weikum, and M. Zwilling, "Unbundling Transaction Services in the Cloud", In *Proc. 4th Biennial Conference on Innovative Data Systems Research (CIDR)* (2009).

[Mehrotra et al. (1991)] S. Mehrotra, R. Rastogi, H. F. Korth, and A. Silberschatz, "Non-Serializable Executions in Heterogeneous Distributed Database Systems", In *Proc. of the International Conf. on Parallel and Distributed Information Systems* (1991), pages 245–252.

[Mehrotra et al. (2001)] S. Mehrotra, R. Rastogi, Y. Breitbart, H. F. Korth, and A. Silberschatz, "Overcoming Heterogeneity and Autonomy in Multidatabase Systems.", *Inf. Comput.*, Volume 167, Number 2 (2001), pages 137–172.

[Mohan and Lindsay (1983)] C. Mohan and B. Lindsay, "Efficient Commit Protocols for the Tree of Processes Model of Distributed Transactions", In *Proc. of the ACM Symposium on Principles of Distributed Computing* (1983), pages 76–88.

[Mu et al. (2016)] S. Mu, L. Nelson, W. Lloyd, and J. Li, "Consolidating Concurrency Control and Consensus for Commits under Conflicts", In *Symp. on Operating Systems Design and Implementation (OSDI)* (2016), pages 517–532.

[Ozsu and Valduriez (2010)] T. Ozsu and P. Valduriez, *Principles of Distributed Database Systems*, 3rd edition, Prentice Hall (2010).

[Padhye and Tripathi (2015)] V. Padhye and A. Tripathi, "Scalable Transaction Management with Snapshot Isolation for NoSQL Data Storage Systems", *IEEE Transactions on Services Computing*, Volume 8, Number 1 (2015), pages 121–135.

[Schenkel et al. (1999)] R. Schenkel, G. Weikum, N. Weisenberg, and X. Wu, "Federated Transaction Management with Snapshot Isolation", In *Eight International Workshop on Foundations of Models and Languages for Data and Objects, Transactions and Database Dynamics* (1999), pages 1–25.

[Skeen (1981)] D. Skeen, "Non-blocking Commit Protocols", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1981), pages 133–142.

[Traiger et al. (1982)] I. L. Traiger, J. N. Gray, C. A. Galtieri, and B. G. Lindsay, "Transactions and Consistency in Distributed Database Management Systems", *ACM Transactions on Database Systems*, Volume 7, Number 3 (1982), pages 323–342.

Credits

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