CHAPTER 22

Parallel and Distributed Query Processing

In this chapter, we discuss algorithms for query processing in parallel database systems. We assume that the queries are **read only**, and our focus is on query processing in decision support systems. Such systems need to execute queries on very large amounts of data, and parallel processing of the query across multiple nodes is critical for processing queries within acceptable response times.

Our focus in the early parts of this chapter is on relational query processing. However, later in the chapter, we examine issues in parallel processing of queries expressed in models other than the relational model.

Transaction processing systems execute large numbers of queries that perform updates, but each query affects only a small number of tuples. Parallel execution is key to handle large transaction processing loads; however, this topic is covered in Chapter 23.

Bibliographical Notes

Parallel sorting is discussed in [DeWitt et al. (1992)]. Parallel sorting on multicore and multithreaded processors is discussed in [Garcia and Korth (2005)] and [Chen et al. (2007)]. Parallel join algorithms are described by [Nakayama et al. (1984)], [Richardson et al. (1987)], [Kitsuregawa and Ogawa (1990)], and [Wilschut et al. (1995)], among other works.

Skew handling in parallel joins is described by [Walton et al. (1991)], [Wolf (1991)], and [DeWitt et al. (1992)].

Parallel query-optimization techniques are described by [Lu et al. (1991)] and [Ganguly et al. (1992)].

The adaptation of database-system design and query-processing algorithms to multicore and multithreaded architectures is discussed in the proceedings of the Interna-

tional Workshop on Data Management on Modern Hardware (DaMoN), held annually since 2005.

As of 2017, most of the SQL on MapReduce implementations above (including Hive, Impala, Asterix, Flink, and Spark, with the exception of Apache HAWQ and Microsoft PDW) had only heuristics-based optimizers; adding support for cost-based optimization was an ongoing effort.

[Stocker et al. (2001)] describe integration of semi-join reducers into an optimizer.

Bibliography

[Chen et al. (2007)] S. Chen, A. Ailamaki, P. B. Gibbons, and T. C. Mowry, "Improving Hash Join Performance through Prefetching", *ACM Transactions on Database Systems*, Volume 32, Number 3 (2007), page 17.

[DeWitt et al. (1992)] D. DeWitt, J. Naughton, D. Schneider, and S. Seshadri, "Practical Skew Handling in Parallel Joins", In *Proc. of the International Conf. on Very Large Databases* (1992), pages 27–40.

[Ganguly et al. (1992)] S. Ganguly, W. Hasan, and R. Krishnamurthy, "Query Optimization for Parallel Execution", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1992), pages 9–18.

[Garcia and Korth (2005)] P. Garcia and H. F. Korth, "Multithreaded Architectures and the Sort Benchmark", In *Proc. of the First International Workshop on Data Management on Modern Hardware (DaMoN)* (2005).

[Kitsuregawa and Ogawa (1990)] M. Kitsuregawa and Y. Ogawa, "Bucket Spreading Parallel Hash: A New, Robust, Parallel Hash Join Method for Skew in the Super Database Computer", In *Proc. of the International Conf. on Very Large Databases* (1990), pages 210–221.

[Lu et al. (1991)] H. Lu, M. Shan, and K. Tan, "Optimization of Multi-Way Join Queries for Parallel Execution", In *Proc. of the International Conf. on Very Large Databases* (1991), pages 549–560.

[Nakayama et al. (1984)] T. Nakayama, M. Hirakawa, and T. Ichikawa, "Architecture and Algorithm for Parallel Execution of a Join Operation", In *Proc. of the International Conf. on Data Engineering* (1984), pages 160–166.

[Richardson et al. (1987)] J. Richardson, H. Lu, and K. Mikkilineni, "Design and Evaluation of Parallel Pipelined Join Algorithms", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1987), pages 399–409.

[Stocker et al. (2001)] K. Stocker, R. Braumandl, A. Kemper, and D. Kossmann, "Integrating Semi-Join-Reducers into State-of-the-Art Query Processors", In *Proc. of the International Conf. on Data Engineering* (2001), page 0575.

[Walton et al. (1991)] C. Walton, A. Dale, and R. Jenevein, "A Taxonomy and Performance Model of Data Skew Effects in Parallel Joins", In *Proc. of the International Conf. on Very Large Databases* (1991), pages 537–548.

[Wilschut et al. (1995)] A. N. Wilschut, J. Flokstra, and P. M. Apers, "Parallel Evaluation of Multi-Join Queues", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1995), pages 115–126.

[Wolf (1991)] J. Wolf, "An Effective Algorithm for Parallelizing Hash Joins in the Presence of Data Skew", In *Proc. of the International Conf. on Data Engineering* (1991).

Credits

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