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.
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


**Credits**

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