CHAPTER 15

Query Processing

Query processing refers to the range of activities involved in extracting data from a database. The activities include translation of queries in high-level database languages into expressions that can be used at the physical level of the file system, a variety of query-optimizing transformations, and actual evaluation of queries.

Bibliographical Notes

[Graefe (1993)] presents an excellent survey of query-evaluation techniques. [Faerber et al. (2017)] describe main-memory database implementation techniques, including query processing techniques for main-memory databases, while [Kemper et al. (2012)] describes techniques for query processing with in-memory columnar data. [Samet (2006)] provides a textbook description of spatial data structures, while [Shekhar and Chawla (2003)] provides a textbook description of spatial databases, including indexing and query processing techniques. Textbook descriptions of techniques for indexing documents, and efficiently computing ranked answers to keyword queries may be found in [Manning et al. (2008)].

A query processor must parse statements in the query language, and must translate them into an internal form. Parsing of query languages differs little from parsing of traditional programming languages. Most compiler texts cover the main parsing techniques, and present optimization from a programming-language point of view.

[Knuth (1973)] presents an excellent description of external sorting algorithms, including an optimization called *replacement selection*, which can create initial runs that are (on the average) twice the size of memory. [Nyberg et al. (1995)] shows that due to poor processor-cache behavior, replacement selection performs worse than in-memory quicksort for run generation, negating the benefits of generating longer runs. [Nyberg et al. (1995)] presents an efficient external sorting algorithm that takes processor cache effects into account. Query evaluation algorithms that take cache effects into account have been extensively studied; see, for example, [Harizopoulos and Ailamaki (2004)].

According to performance studies conducted in the mid-1970s, database systems of that period used only nested-loop join and merge join. These studies, including [Blasgen and Eswaran (1976)], which was related to the development of System R, determined that either the nested-loop join or merge join nearly always provided the optimal join method. Hence, these two were the only join algorithms implemented in System R. However, [Blasgen and Eswaran (1976)] did not include an analysis of hash-join algorithms. Today, hash joins are considered to be highly efficient and widely used.

Hash-join algorithms were initially developed for parallel database systems. Hybrid hash join is described in [Shapiro (1986)]. [Zeller and Gray (1990)] and [Davison and Graefe (1994)] describe hash-join techniques that can adapt to the available memory, which is important in systems where multiple queries may be running at the same time. [Graefe et al. (1998)] describes the use of hash joins and *hash teams*, which allow pipelining of hash joins by using the same partitioning for all hash joins in a pipeline sequence, in the Microsoft SQL Server.

[Brinkhoff et al. (1993)] discusses an implementation of spatial joins using R-trees. [Lo and Ravishankar (1996)] and [Patel and DeWitt (1996)] present partitioning-based methods for computation of spatial joins. [Samet and Aref (1995)] provides an overview of spatial data models, spatial operations, and the integration of spatial and nonspatial data.

Bibliography

[Blasgen and Eswaran (1976)] M. W. Blasgen and K. P. Eswaran, “On the Evaluation of Queries in a Relational Database System”, *IBM Systems Journal*, Volume 16, (1976), pages 363–377.

[Brinkhoff et al. (1993)] T. Brinkhoff, H.-P. Kriegel, and B. Seeger, “Efficient Processing of Spatial Joins Using R-trees”, In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1993), pages 237–246.

[Davison and Graefe (1994)] D. L. Davison and G. Graefe, “Memory-Contention Responsive Hash Joins”, In *Proc. of the International Conf. on Very Large Databases* (1994).

[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.

[Graefe (1993)] G. Graefe, “Query Evaluation Techniques for Large Databases”, *ACM Computing Surveys*, Volume 25, Number 2 (1993).

[Graefe et al. (1998)] G. Graefe, R. Bunker, and S. Cooper, “Hash Joins and Hash Teams in Microsoft SQL Server”, In *Proc. of the International Conf. on Very Large Databases* (1998), pages 86–97.

[Harizopoulos and Ailamaki (2004)] S. Harizopoulos and A. Ailamaki, “STEPS towards Cache-resident Transaction Processing”, In *Proc. of the International Conf. on Very Large Databases* (2004), pages 660–671.

[Kemper et al. (2012)] A. Kemper, T. Neumann, F. Funke, V. Leis, and H. Mühe, "HyPer: Adapting Columnar Main-Memory Data Management for Transaction AND Query Processing", *IEEE Data Engineering Bulletin*, Volume 35, Number 1 (2012), pages 46–51.

[Kim (1995)] W. Kim, editor, *Modern Database Systems*, ACM Press (1995).

[Knuth (1973)] D. E. Knuth, *The Art of Computer Programming, Volume 3*, Addison Wesley, Sorting and Searching (1973).

[Lo and Ravishankar (1996)] M.-L. Lo and C. V. Ravishankar, "Spatial Hash-Joins", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1996), pages 247–258.

[Manning et al. (2008)] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press (2008).

[Nyberg et al. (1995)] C. Nyberg, T. Barclay, Z. Cvetanovic, J. Gray, and D. B. Lomet, "AlphaSort: A Cache-Sensitive Parallel External Sort", *VLDB Journal*, Volume 4, Number 4 (1995), pages 603–627.

[Patel and DeWitt (1996)] J. Patel and D. J. DeWitt, "Partition Based Spatial-Merge Join", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1996), pages 259–270.

[Samet (2006)] H. Samet, *Foundations of Multidimensional and Metric Data Structures*, Morgan Kaufmann (2006).

[Samet and Aref (1995)] H. Samet and W. Aref. "Spatial Data Models and Query Processing", In *[Kim (1995)]*, pages 338–360 (1995).

[Shapiro (1986)] L. D. Shapiro, "Join Processing in Database Systems with Large Main Memories", *ACM Transactions on Database Systems*, Volume 11, Number 3 (1986), pages 239–264.

[Shekhar and Chawla (2003)] S. Shekhar and S. Chawla, *Spatial Databases: A TOUR*, Pearson (2003).

[Zeller and Gray (1990)] H. Zeller and J. Gray, "An Adaptive Hash Join Algorithm for Multiuser Environments", In *Proc. of the International Conf. on Very Large Databases* (1990), pages 186–197.