CHAPTER 14

Indexing

Many queries reference only a small proportion of the records in a file. For example, a query like “Find all instructors in the Physics department” or “Find the total number of credits earned by the student with *ID* 22201” references only a fraction of the *instructor* or *student* records. It is inefficient for the system to read every tuple in the *instructor* relation to check if the *dept_name* value is “Physics”. Likewise, it is inefficient to read the entire *student* relation just to find the one tuple for the *ID* “22201”. Ideally, the system should be able to locate these records directly. To allow these forms of access, we design additional structures that we associate with files.

Bibliographical Notes

Discussions of the basic data structures in indexing can be found in [Cormen et al. (2009)]. B-tree indices were first introduced in [Bayer and McCreight (1972)] and [Bayer (1972)]. B^+-trees are discussed in [Comer (1979)],[Bayer and Unterauer (1977)], and [Knuth (1973)]. [Gray and Reuter (1993)] provide a good description of issues in the implementation of B^+-trees.

Index structures optimized for flash storage include the Lazy-Adaptive tree of [Agrawal et al. (2009)], and the FD tree of [Li et al. (2010)]. The Cache-Sensitive B^+-tree (CSB$^+$-tree) presented by [Rao and Ross (2000)] is a B^+-tree variant designed to minimize cache misses. The Bw-tree [Levandoski et al. (2013)] is a B^+-tree variant optimized for main-memory, and features latch-free traversal and updates which minimize concurrency control overheads (concurrency control is discussed later, in Chapter 18). [Faerber et al. (2017)] provide a survey of main-memory databases, including coverage of main-memory indexing techniques.

The Log-Structured Merge (LSM) tree is presented in [O’Neil et al. (1996)], while the Stepped Merge tree is presented in [Jagadish et al. (1997)]. The buffer tree is presented in [Arge (2003)]. [Vitter (2001)] provides an extensive survey of external-memory data structures and algorithms.

Several alternative tree and treelike search structures have been proposed. Skip-lists [Pugh (1990)] are a probabilistic alternative to balanced tree structures, and are used for in-memory indexing in some in-memory databases.

Bitmap indices, and variants called **bit-sliced indices** and **projection indices**, are described in [O'Neil and Quass (1997)]. They were first introduced in the IBM Model 204 file manager on the AS 400 platform. They provide very large speedups on certain types of queries, and are today implemented on most database systems. Research on bitmap indices includes [Wu and Buchmann (1998)],[Chan and Ioannidis (1998)], [Chan and Ioannidis (1999)], and [Johnson (1999)].

[Samet (2006)] provides a textbook coverage of spatial data structures. [Samet (1995)] provides an overview of the large amount of work on spatial index structures. An early description of the quad tree is provided by [Finkel and Bentley (1974)]. [Samet (1990)] and [Samet (1995)] describe numerous variants of quad trees. [Bentley (1975)] describes the k-d tree, and [Robinson (1981)] describes the k-d-B tree. The R-tree was originally presented in [Guttman (1984)]. Extensions of the R-tree are presented by [Sellis et al. (1987)], which describes the R^+ tree, and [Beckmann et al. (1990)], which describes the R^* tree. These structures provide better worst case complexity guarantees for search than R-trees, but at a higher space cost. [Roussopoulos et al. (1995)] describe algorithms for nearest neighbor search on R-trees.

[Lomet and Salzberg (1989)], and [Lomet and Nawab (2015)] describe index structures for temporal data. [Becker et al. (1996)] present an asymptotically optimal multiversion B-tree structure which can be used to index temporal data.

Bibliography

[Agrawal et al. (2009)] D. Agrawal, D. Ganesan, R. Sitaraman, Y. Diao, and S. Singh, "Lazy-Adaptive Tree: An Optimized Index Structure for Flash Devices", *Proceedings of the VLDB Endowment*, Volume 2, Number 1 (2009), pages 361–372.

[Arge (2003)] L. Arge, "The Buffer Tree: A Technique for Designing Batched External Data Structures", *Algorithmica*, Volume 37, Number 1 (2003), pages 1–24.

[Bayer (1972)] R. Bayer, "Symmetric Binary B-trees: Data Structure and Maintenance Algorithms", *Acta Informatica*, Volume 1, Number 4 (1972), pages 290–306.

[Bayer and McCreight (1972)] R. Bayer and E. M. McCreight, "Organization and Maintenance of Large Ordered Indices", *Acta Informatica*, Volume 1, Number 3 (1972), pages 173–189.

[Bayer and Unterauer (1977)] R. Bayer and K. Unterauer, "Prefix B-trees", *ACM Transactions on Database Systems*, Volume 2, Number 1 (1977), pages 11–26.

[Becker et al. (1996)] B. Becker, S. Gschwind, T. Ohler, B. Seeger, and P. Widmayer, "An Asymptotically Optimal Multiversion BTree", *VLDB Journal* (1996).

[Beckmann et al. (1990)] N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1990), pages 322–331.

[Bentley (1975)] J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching", *Communications of the ACM*, Volume 18, Number 9 (1975), pages 509–517.

[Chan and Ioannidis (1998)] C.-Y. Chan and Y. E. Ioannidis, "Bitmap Index Design and Evaluation", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1998), pages 355–366.

[Chan and Ioannidis (1999)] C.-Y. Chan and Y. E. Ioannidis, "An Efficient Bitmap Encoding Scheme for Selection Queries", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1999), pages 215–226.

[Comer (1979)] D. Comer, "The Ubiquitous B-tree", *ACM Computing Surveys*, Volume 11, Number 2 (1979), pages 121–137.

[Cormen et al. (2009)] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd edition, MIT Press (2009).

[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.

[Finkel and Bentley (1974)] R. A. Finkel and J. L. Bentley, "Quad Trees: A Data Structure for Retrieval on Composite Keys", *Acta Informatica*, Volume 4, (1974), pages 1–9.

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

[Guttman (1984)] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1984), pages 47–57.

[Jagadish et al. (1997)] H. V. Jagadish, P. P. S. Narayan, S. Seshadri, S. Sudarshan, and R. Kanneganti, "Incremental Organization for Data Recording and Warehousing", In *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB '97 (1997), pages 16–25.

[Johnson (1999)] T. Johnson, "Performance Measurements of Compressed Bitmap Indices", In *Proc. of the International Conf. on Very Large Databases* (1999), pages 278–289.

[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).

[Levandoski et al. (2013)] J. Levandoski, D. B. Lomet, and S. Sengupta, "The BwTree: A B-tree for New Hardware Platforms", In *Proc. of the International Conf. on Data Engineering* (2013), pages 302–313.

[Li et al. (2010)] Y. Li, B. He, R. J. Yang, Q. Luo, and K. Yi, "Tree Indexing on Solid State Drives", *Proc. VLDB Endow.*, Volume 3, Number 1-2 (2010), pages 1195–1206.

[Lomet and Nawab (2015)] D. Lomet and F. Nawab, "High Performance Temporal Indexing on Modern Hardware", In *Proc. of the International Conf. on Data Engineering* (2015).

[Lomet and Salzberg (1989)] D. Lomet and B. Salzberg, "Access Methods for Multiversion Data", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1989), pages 315–324.

[O'Neil and Quass (1997)] P. O'Neil and D. Quass, "Improved Query Performance with Variant Indexes", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1997), pages 38–49.

[O'Neil et al. (1996)] P. O'Neil, E. Cheng, D. Gawlick, and E. O'Neil, "The Log-structured Merge-tree (LSM-tree)", *Acta Inf.*, Volume 33, Number 4 (1996), pages 351–385.

[Pugh (1990)] W. Pugh, "Skip lists: A probabilistic alternative to balanced trees", *Communications of the ACM*, Volume 33, Number 6 (1990).

[Rao and Ross (2000)] J. Rao and K. A. Ross, "Making B+-Trees Cache Conscious in Main Memory", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (2000), pages 475–486.

[Robinson (1981)] J. Robinson, "The k-d-B Tree: A Search Structure for Large Multidimensional Indexes", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1981), pages 10–18.

[Roussopoulos et al. (1995)] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest Neighbor Queries", In *Proc. of the ACM SIGMOD Conf. on Management of Data* (1995), pages 71–79.

[Samet (1990)] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison Wesley (1990).

[Samet (1995)] H. Samet. "Spatial Data Structures", In *[Kim (1995)]*, pages 361–385 (1995).

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

[Sellis et al. (1987)] T. K. Sellis, N. Roussopoulos, and C. Faloutsos, "The R^+-Tree: A Dynamic Index for Multi-Dimensional Objects", In *Proc. of the International Conf. on Very Large Databases* (1987), pages 507–518.

[Vitter (2001)] J. S. Vitter, "External Memory Algorithms and Data Structures: Dealing with Massive Data", *ACM Computing Surveys*, Volume 33, (2001), pages 209–271.

[Wu and Buchmann (1998)] M. Wu and A. Buchmann, "Encoded Bitmap Indexing for Data Warehouses", In *Proc. of the International Conf. on Data Engineering* (1998), pages 220–230.

Credits

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