
Indexing Techniques for Advanced Database Systems

INDEXING TECHNIQUES FOR ADVANCED DATABASE SYSTEMS

ELISA BERTINO

Dipt. di Scienze dell'Informazione, Università di Milano, Via Comelico 39/41, 20135
Milano, Italy

BENG CHIN OOI

Dept. Information Systems & Computer Science, National University of Singapore,
Lower Kent Ridge Road, Singapore 119260

RON SACKS-DAVIS

Collaborative Inf. Tech. Research Institute, RMIT & The University of Melbourne,
723, Swanston St, Carlton, Victoria, Australia 3053

KIAN-LEE TAN

Dept. Information Systems & Computer Science, National University of Singapore,
Lower Kent Ridge Road, Singapore 119260

JUSTIN ZOBEL

Collaborative Inf. Tech. Research Institute, RMIT & The University of Melbourne,
723, Swanston St, Carlton, Victoria, Australia 3053

BORIS SHIDLOVSKY

Rank Xerox Research Centre, Grenoble Laboratory, 6, chemin de Maupertuis, 38240
Meylan, France

BARBARA CATANIA

Dipt. di Scienze dell'Informazione, Università di Milano, Via Comelico 39/41, 20135
Milano, Italy

Kluwer Academic Publishers
Boston/Dordrecht/London

Contents

List of Figures	vii
List of Tables	xi
Preface	xiii
1. OBJECT-ORIENTED DATABASES	1
1.1 Object-oriented data model and query language	3
1.2 Index organizations for aggregation graphs	7
1.3 Index organizations for inheritance hierarchies	20
1.4 Integrated organizations	29
1.5 Caching and pointer swizzling	36
1.6 Summary	38
2. SPATIAL DATABASES	39
2.1 Query processing using approximations	40
2.2 A taxonomy of spatial indexes	42
2.3 Binary-tree based indexing techniques	46
2.4 B-tree based indexing techniques	56
2.5 Cell methods based on dynamic hashing	64
2.6 Spatial objects ordering	70
2.7 Comparative evaluation	71
2.8 Summary	75
3. IMAGE DATABASES	77
3.1 Image database systems	78
3.2 Indexing issues and basic mechanisms	80
3.3 A taxonomy on image indexes	84
3.4 Color-spatial hierarchical indexes	91
	v

3.5	Signature-based color-spatial retrieval	105
3.6	Summary	109
4.	TEMPORAL DATABASES	113
4.1	Temporal databases	114
4.2	Temporal queries	119
4.3	Temporal indexes	121
4.4	Experimental study	142
4.5	Summary	148
5.	TEXT DATABASES	151
5.1	Querying text databases	152
5.2	Indexing	157
5.3	Query evaluation	169
5.4	Refinements to text databases	175
5.5	Summary	181
6.	EMERGING APPLICATIONS	185
6.1	Indexing techniques for parallel and distributed databases	186
6.2	Indexing issues in mobile computing	194
6.3	Indexing techniques for data warehousing systems	203
6.4	Indexing techniques for the Web	211
6.5	Indexing techniques for constraint databases	214
	References	228

List of Figures

1.1	An object-oriented database schema.	4
1.2	Instances of classes of the database schema in Figure 1.1.	8
1.3	Multi-index for path $P_1 = \text{Author.books.publisher.name}$.	10
1.4	Indexing graphs: a) multi-index; b) join indexes; c) nested index; d) path-index; e) access support relation.	11
1.5	JI organization for path $P_1 = \text{Author.books.publisher.name}$.	12
1.6	Nested index for path $P_1 = \text{Author.books.publisher.name}$.	14
1.7	A nested index for path $P_1 = \text{Author.books.publisher.name}$ clustered on OIDs of instances of the class at the beginning of the path.	15
1.8	Path index for path $P_3 = \text{Organization.staff.books.publisher.name}$.	15
1.9	Access support relation for path $P_3 = \text{Organization.staff.books.publisher.name}$.	17
1.10	Indexing graphs for advanced techniques: a) Path splitting; b) ASR decomposition; c) join index hierarchy.	19
1.11	Derived join index between Author and Publisher.name.	19
1.12	SC-index organization for the inheritance hierarchy rooted at class Book.	21
1.13	Entries of CH-tree for the inheritance hierarchy rooted at class Book.	22
1.14	Fragment of the H-tree organization for the inheritance hierarchy rooted at class Book.	23
1.15	Objects from hierarchy rooted at Book as a 2-dimensional search plane.	26

1.16	Class-division: a) Example hierarchy; b) Binary tree on the class-dimension; c) A CH-query against class C in the 2-dimensional data space.	27
1.17	Nested-inherited index for path $P=Organization.Author.Book.Publisher.$	30
1.18	Example of index contents in a nested-inherited index.	32
1.19	Indexing graph of the nested-inherited index for path $P=Organization.Author.Book.Publisher.name.$	32
2.1	Evolution of spatial index structures.	45
2.2	The organization of data in a kd -tree.	47
2.3	The K-D-B-tree structure.	49
2.4	The hB-tree structure.	50
2.5	The structure of a spatial kd -tree.	53
2.6	The structure of an R-tree.	57
2.7	The structure of an R^+ -tree.	61
2.8	The structure of a BV-tree.	63
2.9	The grid file layout.	65
2.10	Intersection search region in the grid file.	67
2.11	The R-file.	69
2.12	Ordering based on locational keys.	71
2.13	Comparison of R-tree and R^* -tree.	74
3.1	Architecture of an image database system.	79
3.2	A taxonomy of image indexing schemes.	85
3.3	The two-level B^+ -tree structure.	93
3.4	The three-tier color index.	97
3.5	Three images and their 8 largest clusters.	99
3.6	The SMAT structure.	102
3.7	A height-balanced SMAT.	105
3.8	An image partitioned into a 4×8 grid.	106
4.1	A key split of a leaf node in the TSB-tree based on p3.	123
4.2	Time splitting in TSB-tree.	124
4.3	The time index constructed from the <i>tourist</i> relation.	124
4.4	An AP-tree structure of order 3.	126
4.5	Append in the AP-tree.	127
4.6	A Nested ST-tree structure.	128
4.7	An interval B-tree after inserting t1, t2, t3 and t4.	129
4.8	The interval B-tree after insertion all tuples.	130
4.9	Spatial representation of the <i>tourist</i> relation.	131
4.10	The three orderings for points in the two-dimensional space.	131
4.11	Organizing the spatial representation of the <i>tourist</i> relation using a B^+ -tree and linearizing using the D-order.	132

4.12	Space partitioning in the R-tree.	134
4.13	Query regions for R-tree on the time dimension.	135
4.14	A SR-tree with spanning portion and remnant portion.	136
4.15	The five polygon shapes in TP-tree.	137
4.16	An TP-tree for the <i>tourist</i> relation.	138
4.17	A three-dimensional spatial rendition of the TP-tree.	139
4.18	Query regions for the TP-tree.	140
4.19	The two R-tree method.	141
4.20	Effect of arrival rate on time-slice intersection query.	145
4.21	Effect of longer lifespan on time-slice intersection query, $(\lambda, \mu) = (5, 500)$.	146
4.22	Performance of intersection search in key-range time-slice query $(\lambda, \mu) = (5, 200)$.	147
5.1	Example entry in newspaper database.	153
5.2	Arrangement of a simple inverted file..	158
5.3	Single-pass index construction algorithm using temporary files.	165
5.4	Two-pass index construction algorithm.	166
5.5	Elementary ranking algorithm using an array of accumulators.	171
5.6	Interleaved ranking algorithm using limited accumulators	174
5.7	SGML document illustrating hierarchical structure.	176
6.1	Organization of a file under linear hashing.	191
6.2	Message exchanges in distributed linear hashing when performing insertion of a new key.	193
6.3	Reference architecture of a mobile network.	195
6.4	MH and MSS interaction.	197
6.5	A general organization for broadcasted data.	198
6.6	The general protocol for retrieving broadcasted data.	199
6.7	Bcast organization in the (1-m) indexing method.	201
6.8	Bcast organization in the distributed indexing method.	202
6.9	Bcast organization for the flexible indexing method.	202
6.10	An example of star-schema database with a central fact table (SALES) and several dimension tables.	206
6.11	An example of a bitmap index entry.	208
6.12	An example of bitmap join index entry.	209
6.13	An example of projection index.	210
6.14	Relation r_1 (white) and r_2 (shadow).	216
6.15	Categories of possible intersections of a query interval with a database of intervals.	220
6.16	Reduction of the interval intersection problem to a diagonal-corner searching problem with respect to x .	220

- 6.17 (a) A polygon p representing the extension of a linear generalized tuple; (b) A pair of open polygons representing p in the dual plane, together with the points representing lines q_1, q_2, q_3, q_4 in the dual plan.

List of Tables

4.1	A <i>tourist</i> transaction time relation at time 0.	116
4.2	The <i>tourist</i> transaction time relation at time 3.	116
4.3	The <i>tourist</i> valid time relation at time 0.	117
4.4	The <i>tourist</i> valid time relation at time 3.	117
4.5	The <i>tourist</i> bitemporal relation at time 0.	118
4.6	The <i>tourist</i> bitemporal relation at time 5.	118
4.7	A <i>tourist</i> relation for running examples.	121

Preface

Database management systems are widely accepted as a standard tool for manipulating large volumes of data on secondary storage. To enable fast access to stored data according to its content, databases use structures known as indexes. While indexes are optional, as data can always be located by exhaustive search, they are the primary means of reducing the volume of data that must be fetched and processed in response to a query. In practice large database files must be indexed to meet performance requirements.

Recent years have seen explosive growth in use of new database applications such as CAD/CAM systems, spatial information systems, and multimedia information systems. The needs of these applications are far more complex than traditional business applications. They call for support of objects with complex data types, such as images and spatial objects, and for support of objects with wildly varying numbers of index terms, such as documents. Traditional indexing techniques such as the B-tree and its variants do not efficiently support these applications, and so new indexing mechanisms have been developed. As a result of the demand for database support for new applications, there has been a proliferation of new indexing techniques.

The need for a book addressing indexing problems in advanced applications is evident. For practitioners and database and application developers, this book explains best practice, guiding selection of appropriate indexes for each application. For researchers, this book provides a foundation for development of new and more robust indexes. For newcomers, this book is an overview of the wide range of advanced indexing techniques.

The book consists of six self-contained chapters, each handled by area experts: Chapters 1 and 6 by Bertino, Catania, and Shidlovsky, Chapters 2, 3 and 4 by Ooi and Tan, and Chapter 5 by Sacks-Davis and Zobel. Each of the first five chapters discusses indexing problems and techniques for a different

database application; the last chapter discusses indexing problems in emerging applications.

In Chapter 1 we discuss indexes and query evaluation for object-oriented databases. Complex objects, variable-length objects, large objects, versions, and long transactions cannot be supported efficiently by relational database systems. The inadequacy of relational databases for these applications has provided the impetus for database researchers to develop object-oriented database systems, which capture sophisticated semantics and provide a close model of real-world applications. Object-oriented databases are a confluence of two technologies: databases and object-oriented programming languages. However, the concepts of object, method, message, aggregation and generalization introduce new problems to query evaluation. For example, aggregation allows an object to be retrieved through its composite objects or based on the attribute values of its component objects, while generalization allows an object to be retrieved as an instance of its superclass.

Spatial data is large in volume and rich in structures and relationships. Queries that involve the use of spatial operators (such as spatial intersection and containment) are common. Operations involving these operators are expensive to compute, compared to operations such as join, and indexes are essential to reduction of query processing costs. Indexing in a spatial database is problematic because spatial objects can have non-zero extent and are associated with spatial coordinates, and many-to-many spatial relationships exist between spatial objects. Search is based, not only on attribute values, but on spatial properties. In Chapter 2, we address issues related to spatial indexing and analyze several promising indexing methods.

Conventional databases only store the current facts of the organization they model. Changes in the real world are reflected by overwriting out-of-date data with new facts. Monitoring these changes and past values of the data is, however, useful for tracking historical trends and time-varying events. In temporal databases, facts are not deleted but instead are associated with times, which are stored with the data to allow retrieval based on temporal relationships. To support efficient retrieval based on time, temporal indexes have been proposed. In Chapter 3, we describe and review temporal indexing mechanisms.

In large collections of images, a natural and useful way to retrieve image data is by queries based on the contents of images. Such image-based queries can be specified symbolically by describing their contents in terms of image features such as color, shape, texture, objects, and spatial relationship between them; or pictorially using sketches or example images. Supporting content-based retrieval of image data is a difficult problem and embraces technologies including image processing, user interface design, and database management.

To provide efficient content-based retrieval, indexes based on image features are required. We consider feature-based indexing techniques in Chapter 4.

Text data without uniform structure forms the main bulk of data in corporate repositories, digital libraries, legal and court databases, and document archives such as newspaper databases. Retrieval of documents is achieved through matching words and phrases in document and query, but for documents Boolean-style matching is not usually effective. Instead, approximate querying techniques are used to identify the documents that are most likely to be relevant to the query. Effectiveness can be enhanced by use of transformations such as stemming and methodologies such as feedback. To support fast text searching, however, indexing techniques such as special-purpose inverted files are required. In Chapter 5, we examine indexes and query evaluation for document databases.

In the first five chapters we cover the indexing topics of greatest importance today. There are however many database applications that make use of indexing but do not fall into one of the above five areas, such as data warehousing, which has recently become an active research topic due to both its complexity and its commercial potential. Queries against warehouses requires large number of joins and calculation of aggregate functions. Another example is the use of indexes to minimize energy consumption in portable equipment used in a highly mobile environment. In Chapter 6 we discuss indexing mechanisms for several such emerging database applications.

We are grateful to the many people and organizations who helped with this book, and with the research that made it possible. In particular we thank Timothy Arnold-Moore, Tat Seng Chua, Winston Chua, Cheng Hian Goh, Peng Jiang, Marcin Kaszkiel, Alan Kent, Ramamohanarao Kotagiri, Wan-Meng Lee, Alistair Moffat, Michael Persin, and Ross Wilkinson. We are also grateful to the Multimedia Database Systems group at RMIT, the RMIT Department of Computer Science, the Australian Research Council and the Department of Information Systems and Computer Science at the National University of Singapore.

Bertino
Catania
Ooi
Ron Sacks-Davis
Shidlosvky
Tan
Justin Zobel

