Physical Data Modeling for Multidimensional Access Methods

Frank Ramsak *Volker Markl* *Rudolf Bayer*
ramsak@forwiss.de markl@forwiss.de bayer@in.tum.de

Bayerisches Forschungszentrum
für Wissensbasierte Systeme
Orleansstr. 34, 81667 München, Germany
http://mistral.in.tum.de

1 Introduction

Despite the fact that the database community has proposed a vast number of indexing methods over the years, no standard physical data model has been established like it has been achieved on the conceptual and logical level. How to optimize a given data model by using various indexing methods is still the ‚trade secret' of the database administrators. Only recently, some approaches have been tried to make this knowledge available to the normal database user by easy to use optimization tools (e.g., AutoAdmin-Tool of MS SQL Server 7.0). In addition, physical data modeling has concentrated on one-dimensional access methods, since these were the only ones available in commercial database management systems. As multidimensional access methods (MDAMs) are making their way from the research labs into commercial products, a general physical data model should also take MDAMs into account, especially since MDAMs have a high potential to improve processing in important application domains like OLAP, Data Mining, or Archiving Systems. Our research in this field concentrates on providing rules and heuristics for optimal physical data modeling with multidimensional access methods. Currently we are focusing on the application domain of relational OLAP (ROLAP). MDAMs have a high potential in ROLAP since most queries result in multi-attribute restrictions on a table [MZB99].

Arguing that physical data modeling is superfluous in the presence of MDAMs because one could just index all important attributes with one MDAM neglects the fact that the practical limits on the dimensionality of MDAMs lies around ten dimensions. As consequence, physical data modeling for MDAMs does not address the question of which index type to use in the first place, but the question of which attributes to select for indexing.

2 Related Work

A large body of work has been done in the field of index selection, especially in the context of decision support systems/OLAP [GHR+97, Sar97]. Another driving factor are database vendors who support their commercial systems with easy-to-use tools for database administration, like the "AutoAdmin"-Tool of MS SQL Server 7.0 [CN97, CN98]. An important result of [CN99] is the observation that cost based index selection yields significantly better results than selection based only

on structural analysis . To our knowledge all this work covers only one-dimensional index structures, and does not address the special issues of MDAMs.

For our analysis and experiments we are using the UB-Tree [Bay96, Bay97], but the resulting guidelines for physical data modeling also apply to any other MDAM with a disjoint space partitioning like Grid-Files or hB-Trees.

3 Clustering in High Dimensional Space

First of all note that there is a limit for the dimensionality of MDAMs in practice: for more than eight to ten dimensions the performance degenerates too much if not all dimensions are restricted in a query [WSB98]. In the following we present a formal analysis of the influence of the number of dimensions on query performance for a given query. In our model (see Figure 3-1) we assume a disjoint space partitioning where the space is recursively splitted l_i times in the middle of each dimension. Each resulting region of this partitioning maps directly to one page on secondary storage. A query box in this space is defined by the selectivity s_i in each dimension. As consequence, we do not take the position of the query box into account, but only its volume.

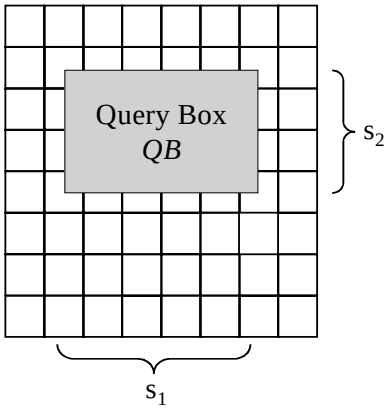

Figure 3-1 MDAM Model

Assume a d-dimensional MDAM $M=\{l_1,...,l_d\}$ on the first d attributes of a relation R with n ($n \geq d$) attributes A_1, ..., A_n, without loss of generality. We specify a query $Q=\{s_1,...,s_n\}$ on relation R by specifying the selectivities $s_i \in [0,1]$ for each attriubte A_i. The resulting query box $QB=\{s_1,...,s_d\}$ on M has a volume of $s_1*...*s_d$ of the complete universe. We are interested in the upper bound of the number of regions overlapping QB for a given d dimensional space partitioning, because this number gives us a cost estimation for the query execution. If we assume uniform data distribution and have l_i partitioning steps in dimension A_i then each resulting region has a selectivity of $1/2^{l_i}$ in A_i. As consequence, QB overlapps at most

$$\left\| \lceil s_i \rceil \right\|_{l_i} = \begin{cases} \lceil s_i \, 2^{l_i} \rceil + 1, \; if \; s_i < \dfrac{2^{l_i} - 1}{2^{l_i}} \\ 2^{l_i}, \; otherwise \end{cases}$$

regions in A_i. In total, the query box QB overlapps at most

$$\left\| \lceil QB \rceil \right\|_M = \prod_{i=1}^{d} \left\| \lceil s_i \rceil \right\|_{l_i}$$

regions of M.

Given this cost function, we are able to simulate the behaviour of MDAMs with varying the dimensionality of the MDAM and the query box. Figure 3-2 shows the simulation results for a table with $P=2^{24}$ pages and different organization. The d-dimensional query box has a selectivity of $0{,}2^d$.

The presented cost model is only applicable in the case where a table consists of $P = 2^{\sum_{i=1}^{d} l_i}$ pages; for any other value of *P* addtional uncomplete splits are introduced. In analogy to [Mar99] the cost model can be enhanced by introducing probabilities of the number of intersected regions by a query box.

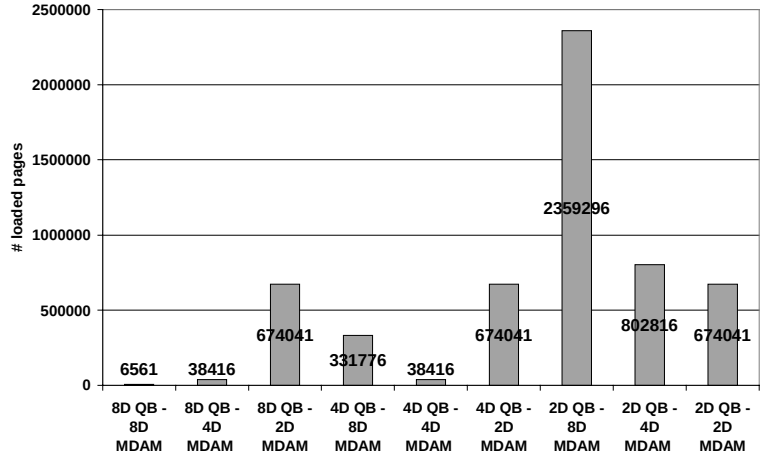

Figure 3-2 Simulation Results

Higher dimensional indexes may efficiently support high-dimensional queries, but if these queries are rare one should chose a dimensionality that corresponds better to the dimensionality of the common queries. Figure 3-3 shows the results of a four dimensional query on a table with two million 200 byte tuples (eight integer attributes, one char-string attribute; total 71249 pages) with different UB-Tree organiztions (four, six, and eight dimensional).

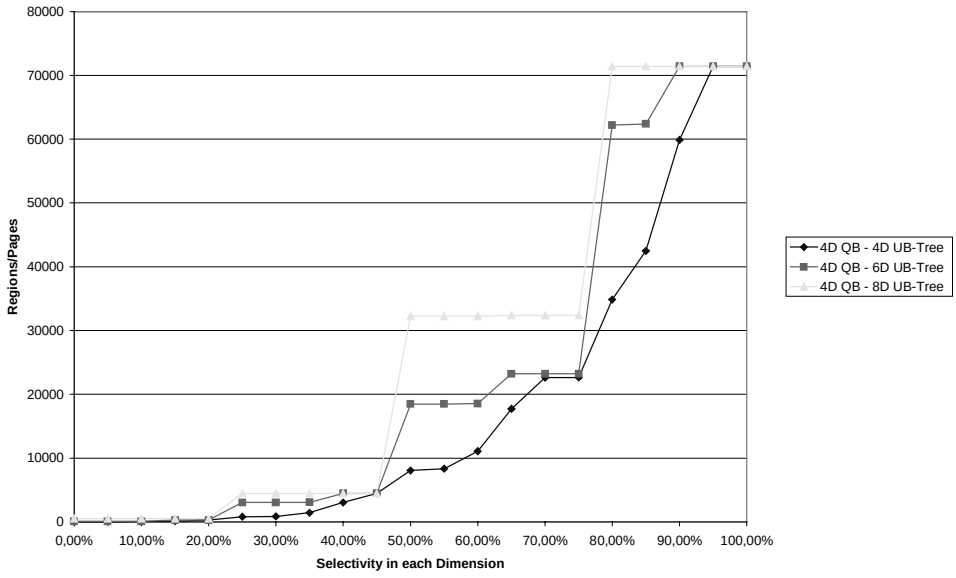

Figure 3-3 Number of pages loaded depending on the dimensionality of the UB-Tree

As expected, the four dimensional organization of the table shows the best performance and the eight dimensional organization the worst performance for four dimensional query boxes.

4 Heuristics for Physical Data Modeling

In the previous section we have shown how the dimensionality influences the multidimensional clustering and therefore the query performance. In this section we will provide rules of thumb and heuristics of which attributes to chose for indexing.

We classify the attributes of a relation into following categories according to their role in queries:

- ***Sorting attributes***: attributes that are mainly used to specify sorting and grouping operations
- ***Filter attributes***: attributes that are used to specify restrictions on the relation

Obviously, an attribute may be used for both, sorting and filtering, but usually one of the two 'roles' of an attribute outweighs the other. Even though sorting is a crucial operation supporting the typical ROLAP queries, even more important is the restriction of the huge data volume that is typical for OLAP applications. As consequence, filter attributes should be preferred to sorting attributes.

As an example we are looking at a query subset (Q5, Q6, Q7, Q8, and Q12) of the TPC-D benchmark [TPC97] and try to find the best multidimensional organization of the LINEITEM table for these queries. We are using the selectivities given by the benchmark specification and the introduced cost model for the index selection; the table size is set to $P=2^{16}=65536$ pages, which corresponds to TPC-D scaling factor SF=10. We assume that the restrictions on the dimension tables of the schema lead to a multidimensional range query on LINEITEM. Index candidates, i.e., attributes that are restricted by the queries are: ORDERKEY, PARTKEY, SUPPKEY, QUANTITY RETURNFLAG, SHIPDATE RECEIPTDATE, AND SHIPMODE. Each single query only restricts up to three attributes, thus an eight dimensional organization is not optimal (see Table 4-1). On the other side, a optimal solution for each query can not be achieved due to the fact that only one clustering MDAM can be created. Trying different organizations we found out that a four dimensional MDAM on ORDERKEY, SUPPKEY, SHIPDATE, and RECEIPTDATE yields the best performance for this query subset. In this case one

Table 4-1 Number of pages loaded for the queries depending on the MDAM organization

MDAM organization	Q5	Q6	Q7	Q8	Q12	Sum of all queries
8D	16384	18432	12288	8192	24576	79872
Best solution for each query	477	1870	308	48	2850	5553
4D on (Orderkey,Supplierkey, Shipdate,Receiptdate)	2560	16384	864	2560	16384	38752
Full Table Scan	65536	65536	65536	65536	65536	327680
1D on Orderkey	1874	65536	5244	3746	65536	141936

has to load significantly less pages than a full table scan (approx. 8 times more pages) and a one dimensional index on ORDERKEY (approx. 3.5 times more pages). However, due to the handling of multiple dimensions MDAMs perform worse than special one-dimensional data structures for one-dimensional queries. As consequence, MDAMs should not be used where one dimensional data structures would suffice.

For optimal physical data modeling knowledge of the application domain is required. In ROLAP hierarchies over dimensions play an important role in query processing as they are usually used as navigation and aggregation paths. In current systems support of hierarchies has been neglected. Multidimensional hierarchical clustering (MHC) as proposed by [MRB99] provides a method to use hierarchies efficiently in query processing.

5 Conclusion and Future Work

The 'curse of dimensionality' seems to be a strong limitation of MDAMs, but in our experience many problems can be modeled with four to eight dimensions. Furthermore, if most dimensions are restricted by a query, an MDAM is applicable very well even in higher dimensional space.

One important feature of MDAMs that is often required is the symmetry property of the indexed attributes, that is that the MDAM shows identical/similar query performance independent of which attributes are restricted. However, in many ROLAP applications not all dimensions that are used to organize the measures are equally important, i.e., some of them are favored in most of the queries. As consequence the MDAM should treat the dimensions according to their importance. We are currently investigating the feasibility of weighted UB-Trees, i.e., UB-Trees where the space partitioning and therefore the multidimensional clustering reflects the different importance of the dimensions. Note that a one-dimensional index over a key of concatenated attributes $A_1 \circ \ldots \circ A_n$ represents the special case of weighted dimensions, where A_1 is the most important one and A_n the least important one.

In addition to MHC, our goal is to provide a general framework for physical data modeling with MDAMs and for handling hierarchies on the logical and physical level.

References

[Bay96] R. Bayer. *The universal B-Tree for multidimensional Indexing*. Technical Report TUM-I9637, Institut für Informatik, TU München, 1996.

[Bay97] R. Bayer. *UB-Trees and UB-Cache – A new Processing Paradigm for Database Systems.* Technical Report TUM-I9722, Institut für Informatik, TU München, 1997.

[CN97] S. Chaudhuri and V. Narasayya. *An efficient, Cost-Driven Index Selection Tool for Microsoft SQL Server*. Proc. of VLDB, 1997.

[CN98] S. Chaudhuri and V. Narasayya. *AutoAdmin "What-If" Index Analysis Utility*. Proc. of SIGMOD, 1998.

[CN99] S. Chaudhuri and V. Narasayya. *Index Merging*. Proc. of ICDE, 1999.

[GHR+97] H. Gupta, V. Harinarayan, A. Rajaraman, and D. Ullman. *Index Selection for OLAP*. Proc. of ICDE, 1997.

[Mar99] V. Markl. *MISTRAL: Processing Relational Queries using a Multidimensional Access Technique.* Ph.D. Thesis Draft, Institut für Informatik, TU München. 1999

[MRB99] V.Markl, F. Ramsak, and R. Bayer. *Improving OLAP Performance by Multidimensional Hierarchical Clustering.* To appear in Proc. of IDEAS, 1999.

[MZB99] V. Markl, M. Zirkel, and R. Bayer. *Processing Operations with Restrictions in Relational Database Management Systems without external Sorting*. Proc. of ICDE, 1999.

[Sar97] S. Sarawagi. *Indexing OLAP data*. Data Engineering Bulletin 20 (1), 1997, pp. 36-43.

[TPC97] Transaction Processing Performance Council. *TPC Benchmark D (Decision Support)*. Standard Specification, Revision 1.2.3. June 1997.
URL: http://www.tpc.org

[WSB98] R. Weber, H.-J. Schek, and S. Bolt. *A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces*. Proc. of VLDB, 1998.