A Constraint-based Mining Approach for Multi-attribute Index Selection

B. Ziani[1], F. Rioult[2] and Y. Ouinten[1]
[1]*LIM, Computer Science Department, University of Laghouat, Laghouat, Algeria*
[2]*GREYC (CNRS - UMR 6072), Université de Caen, Caen, France*

Keywords: Data Warehouse Physical Design, Bitmap Join Index Selection, Data Mining, Constraint Mining.

Abstract: The index selection problem (ISP) concerns the selection of an appropriate indexes set to minimize the total cost for a given workload under storage constraint. Since the ISP has been proven to be an NP-hard problem, most studies focus on heuristic algorithms to obtain approximate solutions. The problem becomes more difficult for indexes defined on multiple tables such as bitmap join indexes, since it requires the exploration of a large search space. Studies dealing with the problem of selecting bitmap join indexes mainly focused on proposing pruning solutions of the search space by the means of data mining techniques or heuristic strategies. The main shortcoming of these approaches is that the indexes selection process is performed in two steps. The generation of a large number of indexes is followed by a pruning phase. An alternative is to constrain the input data earlier in the selection process thereby reducing the output size to directly discover indexes that are of interest for the administrator. For example, to select a set of indexes, the administrator may put limits on the number of attributes or the cardinality of the attributes to be included in the indexes configuration he is seeking. In this paper we addressed the bitmap join indexes selection problem using a constraint-based approach. Unlike previous approaches, the selection is performed in one step by introducing constraints in the selection process. The proposed approach is evaluated using APB-1 benchmark.

1 INTRODUCTION

Data Warehousing and On-line Analytical Processing (OLAP) are becoming critical components of decision support. They are especially designed to enable executives, managers, and analysts to take better and faster decisions. Data warehouses are generally modelled according to a star schema that contains a central, large fact table, and several dimension tables that describe the facts (Inmon, 2002), (Kimball and Ross, 2007).

Queries defined on a star schema are called *star join queries*. They are complex and use several join operations that are very costly. Such queries will be performed on tables having potentially billions of records. As a result, it becomes crucial to accelerate query evaluation. Among the techniques adopted in relational data warehouses to improve query performance, materialized views and indexes are presumably the most effective ones (Chaudhuri and Narasayya, 2007). Data warehouses administrators then handle the fastidious task of choosing an advantageous configuration of indexes to enhance the system performance.

For a given data warehouse, the total number of distinct indexes can be extremely large; hence it is not always practicable to create all the indexes due to the limited amount of storage space that we can physically maintain. The approaches dealing with the index selection problem are composed of two steps:

1. Generation of candidate indexes for a given workload;
2. Selection of a final configuration that minimizes the cost of the workload, while observing the storage space limit.

The first step reduces the space of potential indexes by eliminating non relevant attributes. The final configuration (step 2) is mostly selected using greedy algorithms (Agrawal et al., 2000). The proposed approaches prune the set of generated indexes so that the constraint space is satisfied. However, this pruning process is performed **after** the generation of a large number of candidate indexes.

An alternative is to constrain the generation of indexes in order to produce fewer and more relevant outputs. In this paper we propose a constraint-based mining approach to solve the index selection problem. We believe that constraint-based mining will enable

Ziani B., Rioult F. and Ouinten Y..
A Constraint-based Mining Approach for Multi-attribute Index Selection.
DOI: 10.5220/0003964600930098
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 93-98
ISBN: 978-989-8565-10-5

administrators to focus on a subset of most advantageous indexes and that it avoids the generation of unwanted indexes.

The remainder of this paper is organized as follows: in Section 2 we present existing works related to bitmap join indexes selection problem and constraint-based mining. Section 3 describes the proposed approach for the bitmap join indexes selection. We experimentally study the efficiency of our approach in Section 4. We conclude the paper and present future directions in Section 5.

2 RELATED WORK

2.1 Bitmap Join Index Selection

The index selection problem has been studied first in traditional databases context (Chaudhuri and Narasayya, 1997), (Agrawal et al., 2000), (Chaudhuri et al., 2004), (Feldman and Reouven, 2003), (Frank et al., 1992), (Valentin et al., 2000). With the advent of data warehouse, indexation has become an important option in physical design and its importance is well recognized (Golfarelli et al., 2002). The index selection problem has been proven to be NP-hard (Chaudhuri et al., 2004). Thus, most studies in the literature have focused on finding approximate solutions using greedy strategies or heuristics-based approaches.

The aim of the proposed approaches is to determine a set of candidate indexes from a given workload of queries, then to propose a final indexes configuration providing the best profit, under storage space constraint. However, considered indexes usually concern one table. Bitmap join indexes are multi-attribute indexes involving several tables. Selecting a suitable configuration of Bitmap join indexes is more complicated than the classical mono-table indexes, since it requires the exploration of a large search space. To the best of our knowledge, only few studies dealing with the problem of selecting bitmap join indexes are carried out (Aouiche et al., 2005), (Bellatreche et al., 2007), (Bellatreche and Boukhalfa, 2010), (Ziani and Ouinten, 2011). Due to the large number of candidate indexes, the proposed approaches mainly focused on pruning the search space of potential indexes. They have used frequent itemsets (Aouiche et al., 2005), (Bellatreche et al., 2007), (Ziani and Ouinten, 2011) or heuristic strategies (Bellatreche and Boukhalfa, 2010) to perform the pruning process. In (Aouiche et al., 2005), (Bellatreche et al., 2007) the *Close* algorithm (Pasquier et al., 1999) for mining closed frequent itemsets is used to prune the search space of candidate indexes. Due to the large number of indexes generated as closed frequent itemsets, the authors in (Ziani and Ouinten, 2011) propose a maximal frequent itemsets based approach to perform the selection.

In (Bellatreche and Boukhalfa, 2010), the authors propose an intuitive algorithm for bitmap join indexes selection. As an initial configuration, the algorithm selects an index for each query having indexable attributes. When the size of the configuration exceeds the storage capacity $\mathcal{S}$, some selected indexes should be reduced until the satisfaction of $\mathcal{S}$.

The principal weakness of the proposed approaches is the large number of generated indexes, that is very difficult to manage, according to the system limitations (number of indexes per table and storage space constraint). Indeed, the pruning is done after the generation of the indexes configuration.

An alternative is to constrain the input data earlier in the selection process, thereby reducing the output size to directly discover indexes that are of interest for the administrator. We believe that a *constraint-based approach* will help to mine a reduced and more relevant indexes configuration.

2.2 Constraint-based Pattern Mining

Mining frequent itemsets (FI) in datasets is a demanding task common to several important data mining applications, that look for interesting patterns within databases (*e.g.*, association rules, correlations, sequences, episodes, classifiers, clusters). It was originally proposed in (Agrawal and Srikant, 1994), (Agrawal et al., 1993) with the Apriori algorithm.

The drawback of mining frequent itemsets is that, if there is a large frequent itemset with size s, then almost all 2^s candidate subsets of the itemset might be generated and tested. Furthermore, the number of frequent itemsets grows very quickly as the minimum support threshold decreases.

Moreover, the huge size of the output complicates the task of the analyst, who has to extract useful knowledge from a very large amount of frequent patterns. To overcome this problem, the paradigm of pattern discovery based on constraints was introduced with the aim at providing a tool for driving the discovery process towards potentially interesting patterns. Using constraints can be of a great help to purge a lot of patterns that are irrelevant for the user.

Constraint-based mining has then been widely addressed, with really different approaches. The mostly used constraints are the minimum or maximum support threshold, including (or being included in) some specific itemset, aggregated computation (sum, aver-

age, min, max, when items are associated to a measure). As this paper does not specifically contribute to the field of constraint-based mining, we just briefly recall below the main contributions.

Most of them combine anti-monotone constraints and monotone one (Pei and Han, 2000; Bucila et al., 2003). A constraint is monotone (resp. anti-) if it preserved while itemset specialization (resp. generalization). Many useful constraints fall within the anti-monotone category, such as the minimum support threshold or uper-bounding the aggregated sum. This allows for powerful pruning of the search space, because this space is built through specialization. The maximum support threshold is a typical monotone constraint.

Other approaches directly prune the dataset (Bonchi et al., 2003) or consider the problem as an inductive database issue and formalize the constraints as queries, in a dedicated constraint-base mining environment (Boulicaut et al., 2005),(Jeudy and Boulicaut, 2002).

3 CONSTRAINT-BASED INDEX SELECTION

To illustrate the motivation of our approach, let us see an example. Suppose that a given approach recommends a set $C_{idx} = \{I_1, I_2, \ldots, I_k\}$ of k indexes. The administrator may keep an index I_j knowing that it needs acceptable storage space, or reject it because it has previously shown negligible improvement for the system performance.

Indeed, depending on the cardinality of the attributes, the indexing process may be more or less efficient. If the cardinality is very large or very small, an index might not bring a very significant improvement (Vanichayobon and Gruenwald, 1999). On the other hand, it is not beneficial to create an index on a small table. Hence, table size is another parameter which can be taken into account. The administrator decides whether a table is large or not, and only the indexes on attributes belonging to large tables are selected.

More formally, let $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ be the set of indexable attributes and $\mathcal{D}$ the extraction context (Query/Attributes) for a given workload $\mathcal{W}$. If $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ is a set of k functions, denoting the properties of interest (constraints) for each index $I \subseteq \mathcal{A}$, our approach to solve the index selection problem requires to compute all the itemsets (indexes) occurring in the extraction context $\mathcal{D}$ and satisfying the set of constraints $\mathcal{C}$, i.e:

$$\{I \subseteq \mathcal{A} | C_1(I) \wedge C_2(I) \wedge \cdots \wedge C_k(I)\}$$

The architecture of our approach is illustrated in Figure 1. As data mining based approaches, it constructs an extraction context by identifying the indexable attributes from a given workload. Then, it performs a constraint-based extraction (involving administrator expertise) to generate the desired indexes. Unlike the classical frequent itemsets mining based approaches, we do not built an initial indexes configuration and we do not need to use a greedy algorithm to recommend a final configuration.

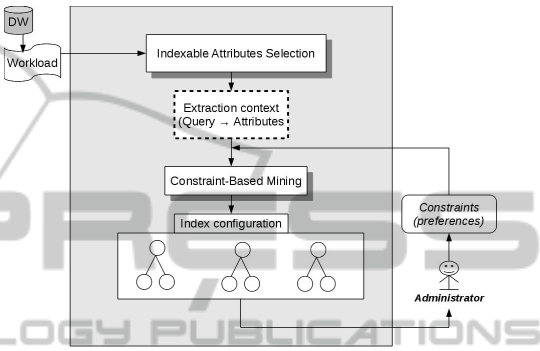

Figure 1: Constraint-based indexes selection.

4 EXPERIMENTAL STUDY

4.1 Description of the Experiment

The aim of our experiments is to evaluate our approach by observing the impact of using various constraints on the selected indexes. In the first experiment we study the impact of including constraints in the mining process on the number of generated indexes and the corresponding storage space. In the second expriments, we compare the performance of our approach with the baseline case where no indexes are created as well as the approaches using classical frequent itemsets mining, where the attributes frequency is the unique parameter used to generate a configuration of indexes. To evaluate the interestingness of the indexes generated by our approach, we use the same cost models proposed in similar works (Aouiche et al., 2005).

To perform a constraint-based selection, we have used the `MUSIC-dfs` tool (Mining with a User-SpecifIed Constraint, Depth-First Search approach) (Soulet et al., 2006). This tool provides a flexible and rich constraint query language. The user can iteratively develop complex constraints integrating various knowledge types.

In this study we are more interested in the quality of the generated indexes. Thus, our comparisons

are performed with no restrictions on available disk space. We have used the APB-1 Benchmark of the OLAP Council (OLAP-Council, 1998). The APB-1 Benchmark simulates a star schema data warehouse. It consists of one fact table *Actvars* and four dimension tables *ProLevel, TimeLevel, CustLevel, and ChanLevel*. We have considered 12 indexable attributes (Table 1).

Table 1: Characteristics of the indexable attributes.

Code	Attribute	Cardinality	Size of the dimension table
A	Class_Level	605	9
B	Quarter_Level	4	900
C	Group_Level	300	9
D	Family_Level	75	9
E	Line_Level	15	9
F	Division_Level	4	9
G	Year_Level	2	900
H	Month_Level	12	900
I	Retailer_Level	99	9000
J	Gender_Level	2	9000
K	All_Level	5	24
L	City_Level	4	9000

We have also used the same workload used in (Bellatreche and Boukhalfa, 2010). It consists of 60 star join queries involving aggregation operations and multiple joins between the fact table and dimension tables. We considered the following constraints:

- the support (frequency) of the generated indexes. This support shows the representativity of the index in the workload of queries;
- the length (number of attributes) of the generated indexes. It directly impacts on the width of the space needed for storing the index;
- the cardinality of the attributes in the generated indexes. This factor also impacts on the storage space width;
- the size of the dimension table to which an attribute belongs, that impacts on the height to the index.

4.2 Experimental Results

Experiment 1: Number and Size of Indexes vs Constraints. We begin with a baseline experiment where the frequency is the unique parameter taken into account (as classical approaches). Then, we conducted several experiments using the MUSIC-dfs to compute the generated configurations, for different combinations of constraints. The experiments depicted here are performed with a minimum support of 5% (3 queries). Using this threshold, the workload basically generates 56 indexes. This represents a very high value when compared to the size of the workload (60 queries).

Consequently, constraints are added to improve the number of generated indexes. Figures 2 and 3 show respectively the number and the total size occupied by the generated indexes for different constraint combinations. We applied different constraints to examine their impact on the generated configurations. It is interesting to observe that the characteristics of a generated configuration (i.e, number of indexes and total indexes size) depends on the complexity of the constraint (i.e., the number of combinations). This behavior allows the administrator to experiment with a broad set of configurations to select the most interesting one.

Experiment 2: Workload Cost vs Constraints. We compare the performance of our approach with the baseline case where no indexes are created as well as the approaches using classical frequent itemsets technique. For each constraint, we evaluate the workload cost using the generated indexes. The results we obtained are ploted in Figure 4. They show that we achieve a better performance using constraints on both the support (frequency) of indexes and the cardinality of the attributes. For complex constraints, there are very few or no generated indexes and thus the cost of the workload increases.

5 CONCLUSIONS

In this paper we have proposed a constraint-based framework for the index selection problem. Our approach leverages and extends principled methods of mining frequent itemsets for the index selection problem. The key contribution is that we show how constraint-based mining can be adapted in a flexible way that balances the characteristics of the workload and the administrator preferences for the index selection problem.

The existing approaches consist of a fully automatic procedure. Like any conventional process of data mining, this can lead to obvious, unhelpful, or undesirable knowledge (indexes). Our approach associates, on the one hand the high capacity of automatic selection mining frequent itemsets, and in the other hand, the necessary expertise of the administrator. Experimental results show that our approach is effective because it allows for more directly computing the useful indexes with precisely describing the requirements.

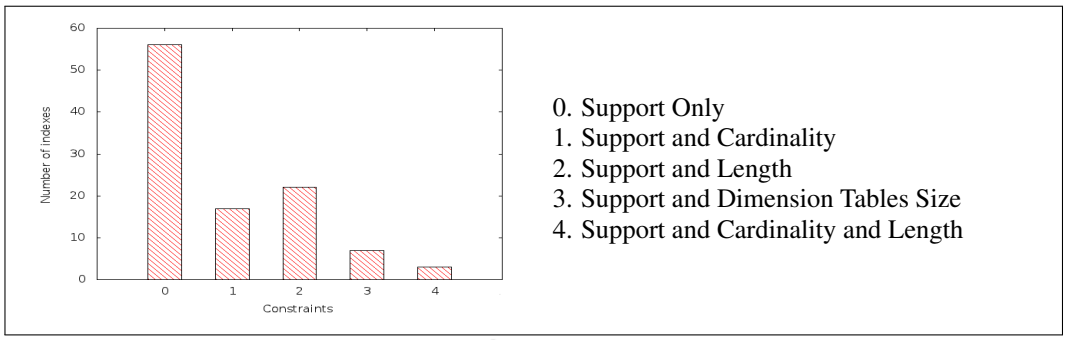

Figure 2: Number of generated indexes vs constraints.

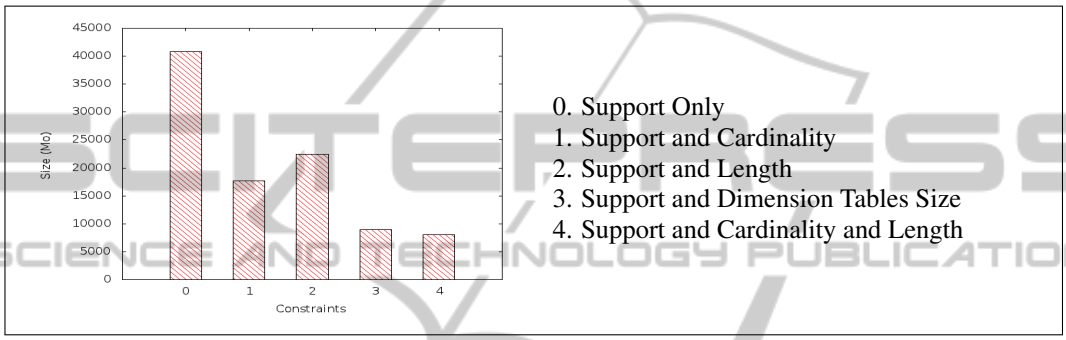

Figure 3: Size of generated indexes vs constraints.

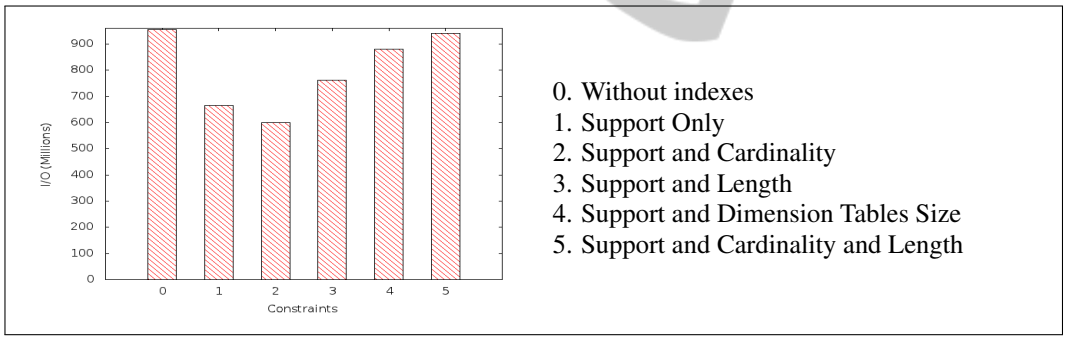

Figure 4: Workload cost vs constraints.

REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *ACM SIGMOD International Conference on Management of Data, Washington, D.C*, pages 207–216.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *International Conference on Very Large Data Bases, Santiago de Chile, Chile*, pages 487–499.

Agrawal, S., Chaudhuri, S., and Narasayya, V. (2000). Automated selection of materialized views and indexes in sql databases. In *VLDB*, pages 496–505.

Aouiche, K., Darmont, J., Boussaid, O., and Bentayeb, F. (2005). Automatic selection of bitmap join index in data warehouses. In *7th International Conference, DaWaK, Copenhagen, Denmark*, pages 64–73.

Bellatreche, L. and Boukhalfa, K. (2010). Yet another algorithms for selecting bitmap join index. In *12th International Conference, DAWAK , Bilbao, Spain*, pages 105–116.

Bellatreche, L., Missaoui, R., Necir, H., and Drias, H. (2007). Selection and pruning algorithms for bitmap index selection problem using data mining. In *9th International Conference, DaWaK, Regensburg, Germany*, pages 221–230.

Bonchi, F., Giannotti, F., Mazzanti, A., and Pedreschi, D.

(2003). Exante: Anticipated data reduction in constrained pattern mining. In *Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03), Cavtat-Dubrovnik, Croatia*, pages 47–58.

Boulicaut, J.-F., Raedt, L. D., and Mannila, H., editors (2005). *Constraint-Based Mining and Inductive Databases, European Workshop on Inductive Databases and Constraint Based Mining, Hinterzarten, Germany, March 11-13, 2004*, volume 3848 of *Lecture Notes in Computer Science*. Springer.

Bucila, C., Gehrke, J. E., Kifer, D., and White, W. (2003). Dualminer: A dual-pruning algorithm for itemsets with constraints. *Data Mining and Knowledge Discovery*, 7(4):241–272.

Chaudhuri, S., Datar, M., and Narasayya, V. (2004). Index selection for databases: A hardness study and a principled heuristic solution. *IEEE Trans. Knowl. Data Eng*, 16:1313–1323.

Chaudhuri, S. and Narasayya, V. (1997). An efficient cost-driven index selection tool for microsoft sql server. In *23rd International Conference on Very Large Data Bases*, pages 146–155.

Chaudhuri, S. and Narasayya, V. (2007). Self-tuning database systems: a decade of progress. In *33rd international conference on Very large data bases*, pages 3–14.

Feldman, Y. A. and Reouven, J. (2003). A knowledge-based approach for index selection in relational databases. *Expert Syst. Appl.*, 25:15–37.

Frank, M., Omiecinski, E., and Navathe, S. (1992). Adaptive and automated index selection in rdbms. In *3rd International Conference on Extending Database Technology, Vienna, Austria*, pages 277–292.

Golfarelli, M., Rizzi, S., and Saltarelli, E. (2002). Index selection for data warehousing. In *4th Intl. Workshop DMDW, Toronto, Canada*.

Inmon, W. (2002.). *Building the Data Warehouse*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition.

Jeudy, B. and Boulicaut, J.-F. (2002). Constraint-based discovery and inductive queries: Application to association rule mining. In Hand, D. J., Adams, N. M., and Bolton, R. J., editors, *Pattern Detection and Discovery*, volume 2447 of *LNCS*, pages 110–124. Springer.

Kimball, R. and Ross, M. (2007). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition.

OLAP-Council (1998). Apb-1 olap benchmark, release ii. *http://www.olapcouncil.org/*.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *7th International Conference on Database Theory*, pages 398–416.

Pei, J. and Han, J. (2000). Can we push more constraints into frequent pattern mining? In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD'00)*, pages 350–354, Boston, USA. New York : ACM Press.

Soulet, A., Klema, J., and Crmilleux, B. (2006). Efficient mining under flexible constraints through several datasets. In *Workshop on Knowledge Discovery in Inductive Databases co-located with PKDD'06*.

Valentin, G., Zuliani, M., Zilio, D., Lohman, G., and Skelley, A. (2000). Db2 advisor: An optimizer smart enough to recommend its own index. In *ICDE*, pages 101–110.

Vanichayobon, S. and Gruenwald, L. (1999). Indexing techniques for data warhouses queries. Technical report, University of Oklahoma, School of computer science.

Ziani, B. and Ouinten, Y. (2011). Enhancing multi-attribute indexes selection using maximal frequent itemsets. In *EGCM, Tanger, Morocco*, pages 65–77.