Online Index Selection in RDBMS by Evolutionary Approach*

Piotr Kołaczkowski and Henryk Rybiński
Institute of Computer Science, Warsaw University of Technology
{pkolaczk,hrb}@ii.pw.edu.pl

Abstract. In recent years, many algorithms for automatic physical database tuning have been proposed and successfully used in tools for administration of relational database management systems. The novel method described in this paper uses a steady-state evolutionary approach to continuously give index recommendations so that the database management system can adapt to changing workload and data distribution. Contrary to online algorithms offering recommendations on a per-query basis, our solution takes into account index reuse across different queries. The experiments show that the quality of the recommendations obtained by the proposed method matches the quality of recommendations given by the best offline index selection algorithms. Moreover, high performance and low memory footprint of the method make it suitable for autonomic database tuning systems.

1 Introduction

Relational Database Management Systems (RDBMS) have been continuously developed for more than three decades now and became very complex. To administer them, a lot of experience and knowledge is required. The costs of employing professional database administrators are often much higher than the costs of database software licensing [4]. Recently, we have been observing a high demand on solutions reducing these costs. Especially intelligent, automatic tools for solving complex administration problems are very helpful. One of such complex problems is performance tuning. In this work we consider the aspect of proper index selection (IS), which often significantly affects the overall database application performance. The importance of proper IS increases with the size of a database. A perfect tool for automatic IS should not only provide good index recommendations, but also be able to continuously monitor the database system, and adapt the selected index set appropriately whenever the database structure, content or workload change. This self-tuning activity should not pose a significant load on the database system or consume large amounts of resources, because otherwise the main purpose of using it, i.e. increasing the overall performance of the system, would be defeated.

* The work has been supported by the Ministry of Science and Higher Education grant No N N 516 375936.
There has been a lot of research into automatic physical database tuning recently. Some of these methods, successfully employed in commercial RDBMS, e.g. [2] provide very high quality of index recommendations. However, high recommendation quality usually comes at a price of high computational cost and high memory usage, so these methods are not well suited for self-tuning database systems. On the other hand, some methods (e.g. [10][11]) propose continuous online tuning algorithms that reduce the tuning overheads by carefully choosing a small subset of indexes that need to be concerned as candidates for materialization. However, they make simplifying assumptions about the underlying cost model, e.g. they assign benefits to single indexes, as if these benefits were independent on the existence of other indexes. Methods based on that approach are fast, but may give results far from the optimum.

Our aim was to develop a method that would give near-optimal results and simultaneously be capable of running in real-time mode. The method is based on an evolutionary approach. Evolutionary algorithms tend to find good solutions fast for large and complex combinatorial problems, and can adapt the solution to the changing external conditions. This approach has been successfully used to tackle the offline ISP [5].

One of the most costly operation when finding optimal index set is evaluating the quality of the various candidate index configurations. This often requires several query planner invocations to find the optimal plans for the tasks that might use some of the candidate indexes. Finding the optimal plan for the given task is itself a hard optimisation problem. The number of query planner invocations can be reduced by caching parts of the plans that are not affected by the existence and usage of indexes [9], however this increases memory usage for storing many equivalent query plans for each query. In [7] we have proposed a method to explore the space of index configurations without invoking the query planner at all. The algorithm treats the IS problem (ISP) as a multi-query optimisation problem, with objective to find the set of query plans that minimise the summary execution costs, assuming the optimiser is allowed to introduce whatever indexes it wishes. Then, the set of optimal indexes is retrieved a posteriori from the query plans, by linear scanning the plan nodes.

In the following section of this paper we describe a variant of algorithm [7], capable of real-time index selection. In Section 3 we present results of experiments comparing efficiency and performance of our method to the greedy per-query index selection [12] and relaxation based IS [2]. Finally we discuss the strong and weak points of the proposed method and sketch plans for future research.

2 Index Selection Algorithm

The method consists of three main components: (1) the incremental workload compression module responsible for reducing the number of input queries and adjusting their weights according to how frequent they appear in the workload, (2) the evolutionary IS module responsible for continuously providing the best index configurations, and (3) the index materialization service responsible for