PLASTIC: Reducing Query Optimization Overheads through Plan Recycling

Vibhuti S. Sengar Jayant R. Haritsa*
Dept. of Computer Science & Automation
Indian Institute of Science, Bangalore 560012, INDIA

1. INTRODUCTION

Query optimization is a computationally intensive process, especially for the complex queries that are typical in current data warehousing and mining applications. The inherent overheads of query optimization are compounded by the fact that a new query is typically optimized afresh, providing no opportunity to amortize these overheads over prior optimizations. While current commercial query optimizers do provide facilities for reusing execution plans generated for earlier queries (e.g. "stored outlines" in Oracle 9i), the query matching is extremely restrictive – only if the incoming query has a close textual resemblance with one of the stored queries is the associated plan re-used to execute the new query.

Recently, in [1], we proposed a tool called PLASTIC (Plan Selection Through Incremental Clustering) to significantly increase the scope of plan reuse. The tool is based on the observation that even queries which differ in projection, selection and join predicates, as also in the base tables themselves, may still have identical *plan templates*, that is, they share a common database operator tree. By identifying such similarities in the plan space, we can materially improve the utility of plan cacheing.

PLASTIC attempts to capture these similarities by characterizing queries in terms of a feature vector that includes structural attributes such as the number of tables and joins in the query, as well as statistical quantities such as the sizes of the tables participating in the query. Using a distance function defined on these feature vectors, queries are grouped into clusters. Each cluster has a representative for whom the template of the optimizer-generated execution plan is persistently stored, and this plan template is used to execute all future queries assigned to the cluster. In short, PLASTIC recycles plan templates based on the expectation that its clustering mechanism is likely to assign an execution plan that is identical to what the optimizer would have produced on the same query.

Experiments with a variety of TPC-H-based queries showed that PLASTIC predicts the correct plan choice in most cases, thereby providing significantly improved query optimization times. Further, even when errors were made, the additional execution cost incurred due to the sub-optimal plan choices was marginal. PLASTIC also improves query execution efficiency by making it feasible for optimizers to always run at their highest optimization level. Further, the benefits of "plan hints", a common technique for influencing optimizer plan choices for specific queries, automatically percolate to the entire set of queries that are associated with this plan. Lastly, since the association of queries with clusters is based on database statistics, the plan choice for a given query is *adaptive* to the current state of the database.

*Contact Author: haritsa@dsl.serc.iisc.ernet.in. This work was supported in part by a Swarnajayanti Fellowship from the Dept. of Science & Technology, Govt. of India.

2. THE PLASTIC PROTOTYPE

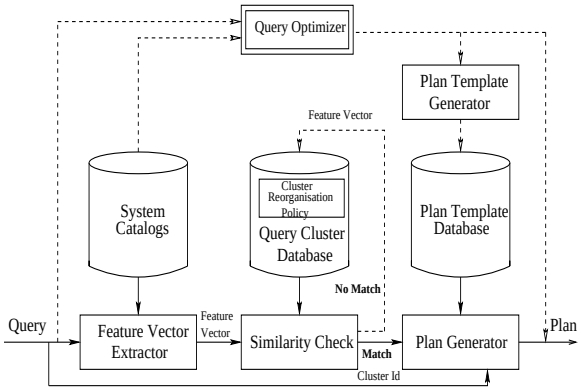

Figure 1: The PLASTIC Architecture

A block-level diagram of the PLASTIC architecture is shown in Figure 1 (the solid lines show the sequence of operations in the situation where a matching cluster is found for the new query, while the dashed lines represent the converse situation where no match is available and a fresh cluster is created). Based on this architecture, we have built a prototype implementation of PLASTIC. The tool is designed to be fully contained in the database system without requiring any support from the operating system; non-intrusive with respect to the optimizer by treating it as a black box and not requiring any internal modifications; easily portable across platforms and optimizers; extensible to incorporate changes to the tool modules; and with a variety of visual interfaces, including a plan diagram [1] generator, to help analyze and improve the tool's performance.

In the demo, a Java implementation of PLASTIC working with Oracle 9i and IBM DB2 is presented, and its potential for amortizing query optimization overheads is highlighted.

3. REFERENCES

[1] A. Ghosh, J. Parikh, V. Sengar and J. Haritsa, "Plan Selection based on Query Clustering", *Proc. of 28th Intl. Conf. on Very Large Data Bases (VLDB)*, August 2002.