

Cost-Based Object Query Optimization

Quan Wang {quan@cse.ogi.edu}

Supervising Professor: David Maier {maier@cse.ogi.edu}
Oregon Graduate Institute of Science and Technology

March 1, 2000

1 The Goal

The long-term goal of our research is to develop a top-down transformation-based and cost-based optimizer for object queries. Relational optimization techniques have been widely and successfully adopted. Thus, the primary task of object query optimization is to adapt relational techniques, and meanwhile to invent new techniques, to address the new features in object query languages such as method invocation, path expressions, user-defined data type, reference attributes, collection-valued attributes and multiple collection types. Previous work [GM93, H95, S98] dealt with path expressions, method invocation and user-defined data types. Multiple collection types and collection-valued attributes (CVAs), being important features, have not been investigated thoroughly regarding their impact on traditional optimization techniques. This thesis proposed an algebraic framework for object query optimization with attention paid to multiple-collection types and CVAs. In the following, we first observe the problems in building a cost-based object query optimizer using the current body of knowledge. Then we present our solutions to these problems. Finally, we report the current status of this research.

2 The Problems

The Algebra Problem

The existing object algebra [SZ90, V93, S95] provides either incomplete mapping from OQL queries to algebraic expressions, or insufficient support for optimization. For instance, the AQUA algebra [V93] does not provide constructors and operators for collection types other than the set and the bag. The EXCESS algebra provides a full range of array operations. However, most array operations are difficult to optimize, because they cannot be reordered with traditional set-oriented operators such as join, projection and selection. These limitations make the existing algebra inappropriate for optimizing object queries.

The Unnesting Problem

Most query unnesting work is based on calculus or source-to-source transformations. Algebraic unnesting is desirable because it can be easily included in top-down transformation-based optimizers. Also, algebraic unnesting allows interleaving of unnesting rules and other transformation rules, which help generating good plans early, facilitating effective pruning. However, the existing algebraic unnesting methods [CM93, S95] are not complete, in the sense that they cannot unnest certain queries with CVAs and the queries contain duplicates in its intermediate results. More general unnesting techniques are desirable but not yet available.

The Materialization Problem

To optimize object queries, Blakeley et al. [BMG93] introduced the materialize operator to explicitly indicate at the logical level the need to resolve reference attributes. Materialize can be implemented by pointer-based joins [SC90, KMG89] or value-based joins [BMG93]. The pointer-based approach applies to all materialization situations, but it is inefficient when attributes to be materialized are shared. The value-based approach is efficient for shared single-valued attributes, but inefficient for shared CVAs. Also the valued-based approach requires the presence of appropriate type extents. As sharing appears frequently in base collections and

intermediate query results, alternative materialization techniques for shared attributes with less restriction and better performance are desired.

The Evaluation Model Problem

Query evaluation can be based on either the relational data model or higher-ordered data models. The choice, however, is rather difficult. Using the relational model allows full exploitation of classical sorting, indexing and hashing algorithms, but may exclude the algorithms developed under higher-ordered data models [DL92]. Alternatively, using higher-ordered model requires more complicated plan generation and cost estimation, and tends to generate many inefficient plans because most higher-ordered operators do not have more efficient algorithms than nested-loops evaluation. How to make an appropriate choice becomes an unavoidable question when developing an object query optimizer.

The Cost Model Problem

Cost model is an important component in cost-based optimization. The optimality of the optimizer is based on correct prediction of relative plan costs. Several object cost models have been proposed [GGT96, BF97]. But none of them addresses the issue of representing and propagating the properties of intermediate results that are required as parameters for cost functions. Also, the cost functions for non-relational operators such as map [V93] and recursive join [DI92] have not been investigated.

3 Thesis Statement and Contributions

My thesis is that effective optimization for object queries can be achieved by non-trivial adaptation of the existing object algebra and traditional optimization techniques. This dissertation addresses the problems observed in the previous section, presenting an algebraic framework and several key techniques for OQL query optimization.

The Algebra

We developed an algebra equipollent to the OQL query language. It includes the d-join operator [CM93] and its variations, which play a key role in our unnesting approach. The algebra also includes the operators manipulating multiple collection types and supports transformations involving those operators.

A Complete Algebraic Unnesting Approach

We developed a complete algebraic approach to unnest OQL queries. This approach can handle nested queries with CVAs and multiple collection types. The general idea is to represent a nested query using d-join operator and its variants, then to reduce these operators into relational operators in a deterministic manner. We proved the soundness and completeness of this unnesting approach [WMS99].

Hybrid Reference Materialization

We proposed a hybrid technique [WMS99] that combines the advantages of both the pointer-based and value-based approaches. This technique relaxes the limitations of the value-based technique, while preserving much of its performance advantage over the pointer-based technique. The hybrid technique shows even stronger performance advantages when moving from single-valued to collection-valued attributes. We also showed how to enhance the performance of value-based techniques on collection-valued attributes when inverse relationships are available. Both

the hybrid and enhanced value-based techniques can be easily incorporated into rule-based query optimizers, using the transformations we present. Analysis and experiments demonstrate that both techniques are complementary to current materialization approaches and achieve superior performance for shared attributes and CVAs.

Evaluation Model

We examined two alternative data models at the query evaluation level: the relational model and the nested model. For the relational model, we proposed an order-tracking technique that allows nested structural information to be encoded in the ordering of flat intermediate results, in order to include some efficient nested algorithms. In most cases, the mapping between nested to flat data model, as well as the mapping from nested to flat algorithms is supported. The overheads of both mappings are bound by linear complexity.

We also demonstrated that nested data model can benefit queries with CVAs, as well as some relational queries. Both data models are supported in our optimizer.

An Adequate Cost Model

We developed a parameter model that represents data properties as a shared complex structure. The model is fully compatible with the relational catalog structure and allows further extensions to accommodate more sophisticated statistics. We also provide the corresponding propagation mechanism. In our previous work [WM97], we developed and initially validated a cost model for object queries involving path expressions. Further, we will develop and validate a cost model for general object queries. For validation purposes, we defined a metrics, called the adequacy of a cost model, to measure the relative deviation of an estimated optimal plan from the actual optimal plan. A formula that computes the metrics based on some basic experimental data is provided in the thesis.

4 Progress

To evaluate the approaches proposed in this thesis, we implemented a cost-based top-down query processor for OQL queries. The query processor consists of several components: a parser, a cost-based optimizer extended from the Columbia query optimizer framework [SMB98] and a query evaluator implemented on the Gemstone/J object database system [G96]. The parser interprets user queries into algebraic expressions. The optimizer consists of a search space that stores equivalent expressions and candidate plans, a search engine that controls various transformations including hybrid materialization rules and unnesting rules and a cost model that estimates the relative costs of equivalent evaluation plans. The query evaluator consists of both relational and non-relational physical operators.

We employed theoretical and empirical methods to validate our approaches. The soundness and completeness of the unnesting algorithm was formally proved. The correctness of transformation rules was verified using set-theoretic reasoning. The effectiveness of hybrid materialization was validated through empirical evaluation. The performance of the two evaluation models will be experimentally contrasted and evaluated. For the cost model, both theoretical and empirical approaches will be employed. First, the accuracy of the cost function for individual physical operators will be measured experimentally. Then quantitative analysis for the adequacy of the cost model will be conducted. The overall optimizer performance will be measured empirically through a set of benchmark queries.

Reference

- [BCK98] R. Braumandl, J. Claussen, A. Kemper. Evaluating functional joins along nested reference sets in object-relational and object-oriented databases, *1998 VLDB*.
- [BMG93] J. A. Blakeley, W. J. McKenna, G. Graefe. Experiences building the Open OODB query optimizers, *1993 SIGMOD*.
- [BF97] E. Bertino, P. Foscoli. On Modeling Cost Functions for Object-Oriented Databases. *IEEE Transaction on Knowledge and Data Engineering*, Vol. 9, No. 3, 1997.
- [CB97] R. G. G. Cattell, D. K. Barry. The object database standard: ODMG 2.0, Morgan Kaufmann Publishers, Inc, 1997.
- [CM93] S. Cluet, G. Moerkotte, Nested queries in object bases, *Proc. Int. Workshop on Database Programming Languages*, 1993.
- [C89] L. Colby. A recursive algebra and query optimization for nested relations, *1989 ACM SIGMOD*.
- [CZ98] M. Cherniack, S. Zdonik. Changing the Rules: Transformations for Rule-Based Optimizers. *1998 ACM SIGMOD*.
- [DL92] V. Deshpande, P. A. Larson, The Design and Implementation of a Parallel Join Algorithm for Nested Relations on Shared-Memory Multiprocessors, *Proc. 18th Intel' Conference on Data Engineering*, February, 1992.
- [G96] *GemStone System Documentation*, Gemstone Inc., 1996.
- [GGT96] G. Gardarin, J. R. Gruser, Z. H. Tang. *Cost-based selection of path expression processing algorithms in object-oriented databases*, *1996 VLDB*.
- [KMG91] T. Keller, G. Graefe, D. Maier. *Efficient assembly of complex objects*, 1991 ACM SIGMOD.
- [ODE95] C. Ozkan, A. Dogas, C. Everndilek. *A heuristic approach for optimization of path expressions*, Technical Report, Middle East Technical University, 1995.
- [RKS88] M. A. Roth, H. F. Korth, A. Silberschatz: Extended Algebra and Calculus for Nested Relational Databases. *TODS* 13(4): 389-417 (1988)
- [S95] H. J. Steenhagen. Optimization of Object Query Language, PH.D thesis, University of Twente, 1995.
- [SMB98] L. Shapiro, D. Maier, K. Billings, Y. Fan, B. Vance, Q. Wang, H. Wu. Safe pruning in the Columbia query optimizer, www.cs.pdx.edu/~len/pruning.doc or [pruning.doc.zip](http://www.cs.pdx.edu/~len/pruning.doc.zip).
- [SC90] E. J. Shekita, M. J. Carey. A performance evaluation of pointer-based joins, *1990 ACM SIGMOD*.
- [SS86] H. J. Schek, M. J. Scholl. The relational model with relation-valued attributes, *Information Systems*, 11(2), 1986.
- [SS87] H-J. Schek, M. H. J. Scholl, The Two Roles of Nested Relations in the DASDBS Project, *LNCS 361*, 1987.
- [V93] S. Vandenberg. Algebras for Object-Oriented Query Languages. Ph.D. Dissertation, University of Wisconsin-Madison, 1993.
- [WM97] Q. Wang. A Cost Model for OQL Query Optimization. Oregon Graduate Institute, RPE Paper, 1997.
- [WMS99] Q. Wang, D. Maier, L. Shapiro. Revisiting Reference Materialization Techniques for Object Query Processing, submitted for publication. <http://cse.ogi.edu/~quan/hybrid.doc>. Technical Report CSE-99-004, Oregon Graduate Institute, 1999.
- [WMS99] Q. Wang, D. Maier, L. Shapiro. Algebraic Unnesting for Nested Queries, submitted for publication. <http://cse.ogi.edu/~quan/unnest.doc>. Technical Report CSE-99-013, Oregon Graduate Institute, 1999.