Intelligent Retrieval Planning

Jonathan J. King
Computer Science Department
Stanford University

A. Introduction

Intelligent retrieval planning is the application of artificial intelligence techniques to the task of efficient retrieval of information from very large databases. * Using such techniques, significant increases in efficiency can be obtained. Some of these improvements are not available through standard methods of database query optimization. Intelligent retrieval planning presents interesting issues related to other artificial intelligence planning research: planning with limited resources[2], optimizing the combined planning and execution process[9], and pursuing plans whose success depends upon the current contents of the database[5]. An experimental system has been implemented to demonstrate the novel kinds of query optimizations and to test strategies for controlling the inference of constraints.

The problem of query optimization has arisen with the development of high level logical data models and nonprocedural query languages ([1], [3]). These free a user from the need to understand the physical organization of the database when posing a query. However, the user's statement of the query may lead to very inefficient processing. Standard techniques of query optimization ([8], [11], [12]) manipulate the set of retrieval operations contained in the query to find a relatively inexpensive sequence. The manipulations are independent of the meaning of the query, depending entirely on such factors as the size of the referenced files.

The essential advance of intelligent retrieval planning over standard techniques of database query optimization is to combine knowledge about the semantics of the application domain with knowledge about the physical organization of the database. Domain knowledge makes it possible to use the constraints in a database query to infer additional constraints which the retrieved data must satisfy. These additional constraints may make it possible to use more efficient retrieval operations or permit the execution of a sequence of operations that has a lower cost. Knowledge of the physical organization of the database can be used to limit the attempts to make such inferences so that the combined process of retrieval and inference is cost effective.

* The research described here is part of the Knowledge Base Management System Project at Stanford and SRI, supported by the Advanced Research Projects Agency of the Department of Defense under contract MDA903-77-C-0322.

B. Finding semantic equivalents of a database query

The techniques of intelligent retrieval planning will be illustrated with a simple example relational database with data about the deliveries of cargoes by ships to ports. The database contains three files, SHIPS, PORTS, and VISITS, with the attributes indicated:

SHIPS: (Shipname Type Length Draft Capacity)

PORTS: (Portname Country Depth Facilities)

VISITS: (Ship Port Date Cargo Quantity)

Semantic knowledge of the application domain is represented as a set of rules. The database is forced, via update restrictions, to conform to this set of rules.

The general semantic knowledge for our sample database consists of these rules:

Rule R1. "A ship can visit a port only if the ship's draft is less than the channel depth of the port."

Rule R2. "A ship can deliver no more cargo than its rated capacity."

Rule R3. "Only liquefied natural gas (LNG) is delivered to ports that are specialized LNG terminals."

Rule R4. "Only tankers deliver oil".

Rule R5. "Only tankers can be over 500 feet long."

During intelligent retrieval planning, the use of the rules is shifted from checking updates to inferring constraints. That is, given certain query constraints, it is possible to infer new constraints that the desired items must meet. For example, suppose a query requests the names of all ships that are longer than 650 feet. By rule R5, it can be inferred that a semantically equivalent retrieval request is for the names of tankers that are longer than 650 feet. This inferred description of the items to be retrieved may permit more efficient processing than the original description.

C. The physical organization of a database

Inferred semantically equivalent sets of constraints can be exploited for intelligent retrieval only if the physical organization of the database, and hence the cost of processing queries, is taken into account. Often, the physical organization has been arranged so that the cost of retrieving a restricted subset of data depends upon the data attributes that have been restricted. For instance, a file may have an auxiliary "index" on one of its attributes. If such an index exists, then the data pages that contain items that meet a constraint on that attribute can be identified directly and only those pages will be fetched. An indexed scan will be much less expensive than a scan through an entire file, measured in terms of pages fetched from disk. A discussion of retrieval costs for different physical database organizations is contained in [4].

Thus, given a query that constrains only unindexed attributes, a reasonable semantic retrieval strategy (subject to qualifications discussed in [4]) is to attempt to infer constraints on indexed attributes. Suppose that the SHIPS file has an index on the Type attribute. In that case, the best way to retrieve all the ships longer than 650 feet would be to fetch all the tankers by means of an indexed scan on Type, and then to check the Length value of each record fetched into main memory by that scan.

D. Novel query optimization based on the use of domain semantics

A query optimization method that uses domain semantics is interesting to the extent that it achieves significant increases in efficiency that are not available by other methods. One unique strategy that can arise when semantics are considered is the inclusion of an extra file in the set of files examined when a query is processed.

For example, suppose a query requests the quantity of liquefied natural gas delivered for each known visit to ports with a channel depth of less than 20 feet. With no inference, a typical query processor would retrieve all PORTS records with a Depth value of less than 20. For each one, it would retrieve all VISITS whose Port attribute was the same as the Portname for the PORTS record and whose Cargo attribute was liquefied natural gas. The cost of the retrieval varies as the product of the sizes of the PORTS and VISITS files.

However, with appropriate rules and indexes, intelligent retrieval planning can provide a much faster retrieval method. Suppose that the VISITS file has an index on the Ship attribute. In effect, this means that the database has been set up to provide inexpensive access from each ship to the set of its visits, while the set of visits to a specific port is much costlier to retrieve. Using rule R1, it can be inferred that the visits requested by the query could have been made only by ships with a draft of less than 20 feet.

It is now possible to retrieve SHIPS with Draft less than 20, then retrieve their associated VISITS (using the index), and finally, for each VISITS record with a Cargo value of liquefied natural gas, retrieve the associated PORTS record to check the value of Depth. If the Draft constraint substantially restricts SHIPS (and therefore the associated VISITS as well), then the overall cost will be much lower than that of the straightforward method, despite the fact that an extra file and an extra retrieval operation have been added. In a simulation test of this method using a cost model based on the System R relational database system[7] in which the VISITS file is much larger than the PORTS and SHIPS files, the simulated retrieval cost was reduced by more than order of magnitude.

E. Controlling the inference of additional constraints

Intelligent retrieval planning is complicated by the need to weigh possible gains in retrieval efficiency against the cost of performing inferences. The amount of planning done in the intelligent retrieval planning system in processing a particular query is determined by the cost of answering the unimproved query, and the possible improvements. The inference control mechanism has these key features:

(1) The specific retrieval problem determines which constraints to try to infer (for example, an attempt is made to add constraints to indexed fields).

(2) Knowledge about both the structure and the content of the database determines the effort to devote to attempting some inference.

(3) Retrieval from the database is an inherent part of the inference process. The ability to carry out an inference (and hence the shape of the whole retrieval plan) may depend upon the current contents of the database.

These features can be illustrated briefly in another example. Suppose the VISITS file is indexed only on Cargo, and a query requests data on visits to the port of Zamboanga. The retrieval strategy mentioned in section 3 suggests an attempt to infer a constraint on Cargo from the given constraint on Port.

Given the number of records in the VISITS file, it is possible to compute the effort needed to perform a sequential scan. The effort alloted to inference will be a function of this. There is no guarantee that a helpful constraint can be found for any particular query. This suggests a policy to allot to the inference process a fixed small fraction of the effort which the original retrieval would take. With such a policy, the effort to plan the retrieval will result in a minor increase in response time if the inference attempt fails, but may provide a major improvement if it succeeds. Although the policy is intuitively plausible, other strategies for alloting effort during problem solving under uncertainty, such as those discussed in [9], are being investigated.

Control of the inference process can be viewed as control of the moves in a space of constraints on attributes. Constraints can be moved either by applying a rule, by retrieving items restricted on one attribute and observing their values on other attributes, or by matching constraints on attributes defined on the same underlying set of entities. Continuing the example, starting with a constraint on the Port attribute of VISITS, new constraints can be found by retrieving from VISITS or by assigning the

value "Zamboanga" to the Portname field of PORTS. The first choice is rejected because the objective is to reduce the cost of that very retrieval. With a constraint on Portname in PORTS, a retrieval from PORTS can be performed. In this case, just a single record will be obtained because Portname is the unique identifier in that file. With appropriate access methods, such as hashing, the retrieval will be very inexpensive.

When the PORTS record for "Zamboanga" has been obtained, rules R1 and R3 may apply. If rule R3 applies, that is, if Zamboanga is a specialized liquefied natural gas terminal, then a strong constraint will be obtained on the goal attribute Cargo, and retrieval from VISITS will take place by means of an indexed scan rather than by means of a more expensive complete scan. If the data on Zamboanga does not support that inference, then other inference paths will have to be considered. This illustrates the possible dependence of retrieval planning on the current contents of the database.

The cost of each inference step: generating new inference path nodes, testing rules, and retrieving from the database itself, is taken from the allotment of planning resources. Planning terminates if a strong goal constraint is found, if no potential inference path can be extended, or if planning resources are exhausted.

F. Conclusion

Intelligent retrieval planning can provide novel and significant improvements in query processing efficiency. It draws on a knowledge of the physical organization of the database and on semantic knowledge of the application modelled in the database. The outcome of retrieval planning, both the retrieval method chosen and its cost, can depend upon the current contents of the database.

An experimental system exists that performs inferences on queries stated in a subset of the SODA relational database query language[6]. The system uses a simple retrieval cost model to select the least expensive semantically equivalent expression of the retrieval request. The cost model is used in conjunction with a planning executive to limit the inference of additional constraints.

Work is under way to codify intelligent retrieval strategies which, though they are specific to a given class of physical database organizations, are independent of the application domain. The eventual aim of this work is to develop a system which, given the set of domain rules and the description of the physical organization for a database, can provide the functions of intelligent retrieval planning described in this paper, much as the EMYCIN system[10] provides knowledge acquisition functions independently of the knowledge base to be built.

Acknowledgments

Many thanks for perceptive comments by Jim Bennett, Jim Davidson, Larry Fagan and Jerry Kaplan of Stanford University, and Barbara Grosz of SRI International.

References

1. Codd, E.F., *A relational model for large shared data banks*, Commun. ACM 13:6 (1970), 377-387.

2. Garvey, Thomas D., *Perceptual strategies for purposive vision*, Technical Note 117, SRI International, Menlo Park, California, September 1976.

3. Kim, Won, *Relational database systems*, ACM Computing Surveys 11:3 (1979), 185-212.

4. King, Jonathan J., *Exploring the use of domain knowledge for query processing efficiency*, Technical Report HPP-79-30, Heuristic Programming Project, Computer Science Department, Stanford University, December 1979.

5. Klahr, Philip, *Planning techniques for rule selection in deductive question-answering*, In Pattern Directed Inference Systems, D.A.Waterman and F. Hayes-Roth (Eds.), Academic Press, 1978.

6. Moore, Robert C., *Handling complex queries in a distributed data base*, Technical Note 170, SRI International, Menlo Park, California, October, 1979.

7. Selinger, P. Griffiths et. al. *Access path selection in a relational database management system*, In Proc. ACM-SIGMOD 1979, Boston, Mass., pp. 23-34.

8. Smith J.M. and P. Chang, *Optimizing the performance of a relational algebra data base interface*, Commun. ACM 18:10 (1975), 568-579.

9. Sproull, Robert F., *Strategy construction using a synthesis of heuristic and decision-theoretic methods*, Report CSL-77-2, Xerox Palo Alto Research Center, Palo Alto, California, July 1977.

10. Van Melle, William, *A domain-independent production-rule system for consulation programs*, IN Proc. IJCAI-79 Tokyo, Japan, 1979, pp. 923-925.

11. Yao, S. Bing, *Optimization of query evaluation algorithms*, ACM Transactions on Database Systems 4:2 (1979) 133-155.

12. Youssefi, Karel A. Allen, *Query processing for a relational database system*, Memorandum UCB/ERL M78/3, Electronics Research Laboratory, University of California, Berkeley, California, January 1978.