Using Knowledge of Redundancy for Query Optimization in Mediators

Vasilis Vassalos
Computer Science Department
Stanford University
Stanford, CA 94305
vassalos@cs.stanford.edu

Yannis Papakonstantinou
Dept. of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093
yannis@cs.ucsd.edu

1 Introduction

Many autonomous and heterogeneous information sources are becoming increasingly available to the user through the Internet – especially through the World Wide Web. In order to make the information available in a consolidated, uniform, and efficient manner, it is necessary to integrate the different information sources. The integration of Internet sources poses several challenges which have not been sufficiently addressed by work on the integration of corporate databases residing on an Intranet [LMR90]. We believe that the most important ones are heterogeneity, large number of sources, redundancy, source autonomy and diverse access methods and querying interfaces.

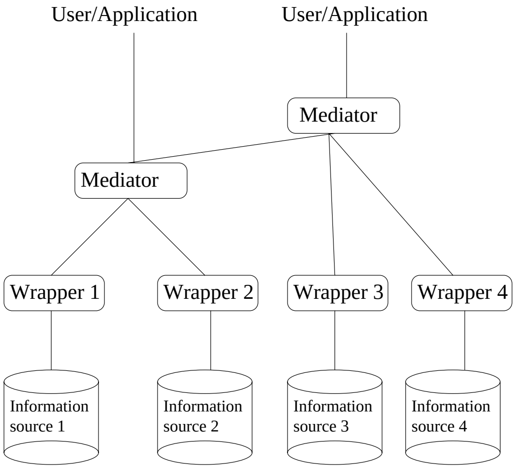

Figure 1: Common mediation architecture

With the exception of the challenges posed by information redundancy, the above mentioned challenges have been significantly addressed by a number of recent mediator systems (a.k.a. information gathering agents) [GM$^+$97a, LYV$^+$98, LRO96, TRV95, S$^+$, AKH93]. These systems have adopted a *mediation* [Wie92] architecture as shown in Figure 1 to address the heterogeneity and autonomy issues. The main component of these systems is the mediator, which is essentially a query processor (a.k.a. query planner).

The recent advances in query planning and optimization have not addressed the challenges and opportunities introduced by redundancy and overlap. First, the system must avoid retrieving the same set of data from multiple sites – a situation which is very likely in the World-Wide-Web. Second, redundancy can be exploited in order to increase query performance and system availability by having the system collect information from the most efficient available sources.

Using knowledge of redundancy we can reduce the number of source accesses that have to be performed to retrieve the answer to the user query. In Section 2, we discuss this problem in more detail and formulate it as a scheduling problem with AND-OR precedence constraints.

Partial Answers The amount of information available online often causes users to be satisfied with *partial answers* to their queries. For example, a user will be most interested in quickly receiving a big percentage of the books published in 1997 on the Middle East rather than waiting for a long time and spending computational resources to get the full set. We discuss how having probabilistic information about source overlap can help derive efficient query plans. However there are some challenges to be addressed:

1. The amount of information necessary to completely specify source overlaps is exponential in the number of sources.

2. The naive algorithm that uses the source overlap information and chooses the best sources to give the required partial answer is also exponential in the number of sources.

In Section 3 we describe the optimization framework and propose approximations that can make efficient use of the source overlap information and provide suboptimal solutions.

1.1 Related Work

Most theoretical work in the area of information integration (e.g., [LMSS95, LRU96, RSU95, DL97, DG97, PGMU96, VP97, VP, KW96]) has been in query processing and query planning. In particular, the generation of sound and complete plans for queries over multiple sources has been studied for a variety of data models and query languages.

There has been work on using local completeness information [FW97, Lev96] to create query plans that guarantee getting complete answers to a query while also identifying irrelevant sources. [AKL97] discuss a method for discriminating between useful and useless sources during query execution, by adding "sensing" subqueries to the query plan.

Using probabilistic information about overlap to help in query planning has recently been proposed in [FKL97], where the goal is to pick the k most useful sources to access. [FKL97] primarily use information about *domain* overlap, i.e., overlap between the collections of objects in the schema, because of the exponential blowup when using source overlap information.

2 Minimizing Source Accesses

The query planning algorithms employed by many mediator systems, such as the Information Manifold [LRO96] or TSIMMIS [GM$^+$97a, LYV$^+$98], transform the user query to queries accessing the sources (i.e., *source queries*) and collecting every relevant piece of information. Note that the same piece of information could have been collected from more than one sources. The source query results are finally integrated in the mediator. In the case of the simplest integration queries, the mediator just takes the union of these results. In the presence of redundancy in the data, such a strategy can be very inefficient.

If we can encode information replication, using for example constraints that state that parts of sources are equivalent, then the mediator can use that information to infer that certain source queries it generates are in fact equivalent. The problems of encoding redundancy information in a mediator and of using the encoding to make inferences about source queries are very interesting and challenging problems that we will not be addressing in this paper. We will focus on what needs to be done after such inferences have been made. A set of source queries has already been divided into equivalence classes; at least one query from each class needs to be executed to obtain a complete[1] answer to the user query. We want to pick these *representatives* from each class in a way that minimizes the total cost of answering the user query. In the following paragraphs we describe in more detail the optimization framework and express this query planning problem as a scheduling problem with AND/OR precedence constraints.

Framework and Cost Model Let $\mathcal{S} = S_1, \ldots, S_n$ be a set of information sources, and let C_i be the cost associated with accessing source i. Given a user query Q, let $\mathcal{P} = P_1, \ldots, P_m$ be the set of source queries under consideration and let $\mathcal{S}^{P_i} = S_1^{P_i}, \ldots, S_l^{P_i}$ be the set of sources that query P_i needs to access. Finally, let the source queries be divided in k equivalence classes, such that all queries in each class produce the same part of the answer to Q. We will denote the queries in class j by $P_{1_j}, \ldots, P_{m_j}$. Our objective is to pick at least one query from each class in a way that minimizes the total cost of executing the chosen queries.

We adopt a simple cost function: the cost of a source query P is the sum of the costs of accessing the sources $\mathcal{S}^P$, i.e., $C_P = \sum_{S_i \in \mathcal{S}^P} C_i$. Moreover, we assume that in the course of answering Q each source is accessed at most once, i.e., results of source accesses are cached so if a second plan needs to access the same source, it can reuse the cached copy.[2] In summary, we want to pick $\mathcal{P}_r = P_{i_1}, \ldots, P_{i_j}$ such that $\sum_{P_{i_l} \in \mathcal{P}_r} C_P$ is minimized. Notice that this formulation naturally allows us to model unavailable sources, by assigning them extremely high access cost.

In particular, assume a user query Q is decomposed by the mediator[3] in the following source queries: P_1, accessing sources $1, 2, 3$, P_2, accessing sources $3, 4$ and P_3, accessing sources $4, 5$. Let

[1] As complete as possible using the available sources.

[2] It is interesting of course to also consider more detailed cost models.

[3] We deliberately do not give any details on the decomposition, since we believe that the query optimization problem we describe is relevant for mediator systems, such as TSIMMIS and the Information Manifold, that follow quite different query processing strategies.

us also assume that the cost of accessing each source is equal to 1. We have determined that source queries P_1 and P_2 actually provide the same information, thus they are equivalent. Then it is obvious that the mediator should answer the user query by executing source queries P_2 and P_3, since this results in fewer source accesses.

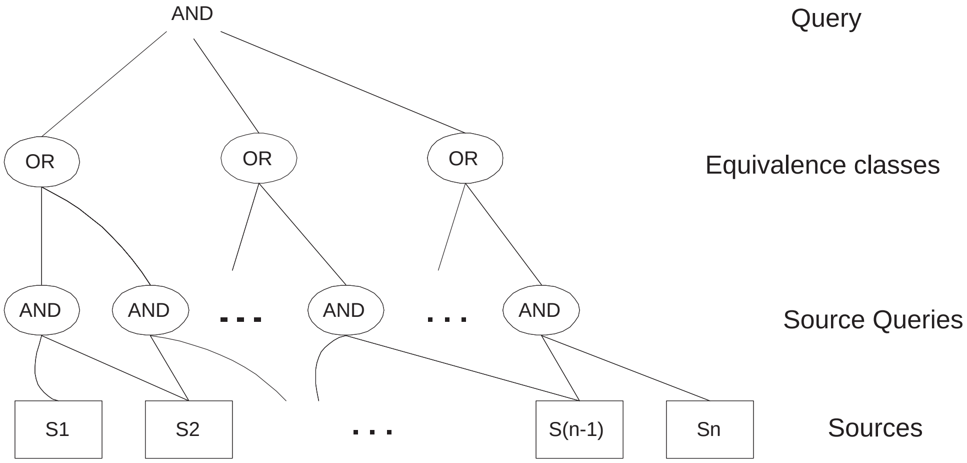

Figure 2: Query plan choice as AND/OR scheduling with internal-tree constraints

We can formulate the problem as an AND/OR scheduling problem with precedence constraints, as in Figure 2. We label each source as a *leaf task*, each source query as an AND-task (for obvious reasons), each equivalence class as an *OR-task* and the user query as an *AND-task*. An *AND-task* cannot be scheduled before all its *predecessors* are scheduled, while an *OR-task* cannot be scheduled before at least one of its predecessors is scheduled. Our goal is to schedule the query node and the optimization criterion is to minimize the number of *leaf-tasks* scheduled. It should be obvious that this is just a simplification of our original problem, where each source S_i has cost $C_i = 1$. Our instance of AND/OR scheduling has *internal-tree* constraints: there are no cycles and all non-leaf nodes have at most one incoming edge.[4]

Known Results and Open Problems The AND/OR scheduling problem is in general NP-complete [GM97b]. The more interesting question is whether there are good polynomial approximation algorithms for this problem. Our instance of the problem has two AND-levels and one OR-level.

- If there are only one AND-level and one OR-level, the problem becomes exactly set cover, so it is polynomially approximable to no better than a $\log n$ factor [Hoc97].
- If the graph has two AND-levels and two OR-levels (alternating) and *internal-tree* constraints, then the problem is not polynomially approximable to even a linear factor [GM97b].

[4] Notice that if an internal AND-node has two incoming edges from two OR-nodes, that means that the corresponding source query belongs to two equivalence classes that need to be collapsed because of transitivity; thus the two OR-nodes will become one, and the constraint will be satisfied.

The query planning problem under consideration is open; we believe it is as hard as AND/OR scheduling with two AND- and two OR- levels (alternating). In view of this, it is essential to evaluate any proposed heuristics in real-life query planning scenarios, to determine their behaviour in real situations.

3 Probablistic Source Overlap: Optimizing for Partial Answers

It is not always possible to provide precise logical constraints on Internet sources, and thus we may not be able to decide that any plans are equivalent to each other. For example, we may know that auto classified source A has 90% of all Honda ads that source B has, but still there may be Honda ads of A that do not appear in B and vice versa. Even though plans involving A might not be equivalent to plans involving B, the mediator should be able to use such information to obtain more efficient query plans for *partial answer* queries, i.e., queries that are satisfied with a percentage of the possible answers. In particular, the mediator should deliver the requested part of the answer and at the same time avoid retrieving overlapping information, hence reducing the computational cost (and possibly the financial cost as well.). Note that optimizing for parts of the answer serves perfectly users who browse. For example, if the user asks for 80% of ads for Honda cars, the mediator can infer that only source A needs to be accessed in order to get the required part of the answer.

For simplicity and clarity, we will discuss optimization of partial answer delivery for queries that perform just selections over unions. In the next paragraphs we present the optimization criteria of the mediator and propose directions for efficient heuristics. The performance of these heuristics will have to be evaluated with many real world experiments.

Framework and Optimization Criteria Let us first motivate the challenges by considering (i) n sources $S_i, i = 1, \ldots, n$, each one exporting a relation R_i and (ii) a query $\cup_{i=1,\ldots,n} R_i$. (We will later generalize to queries including selections.) The mediator has statistics estimating the percentage $p(R_i)$ of each available R_i in the union. We will discuss how to obtain these and other statistics later in the section. For example, $p(R_1) = 1/2$ if R_1 has half of the tuples of $R_1 \cup \ldots \cup R_n$.

We require that the mediator minimizes the total cost for the retrieval of at least a percent of the result, where the percentage a is provided by the user. (It is easy to see that the problem of retrieving the first a tuples is essentially identical.) We adopt a simple cost model, where the cost for accessing a source is equal to the amount of information that we retrieve from the source. The cost of retrieval of the result is the sum of the costs of accessing the sources. (We are planning to consider more detailed cost models in the future.) Therefore, the optimizer has to choose m relations $R_{i_1}, \ldots, R_{i_m}$ such that $\sum_{j=i_1,\ldots,i_m} p(R_j)$ is minimized and $p(\cup_{j=i_1,\ldots,i_m} R_j) >= a$.

An obvious way to solve this problem is to compute and store estimates for all possible $p(\cup_{j=i_1,\ldots,i_m} R_j)$. Given any union query and any percentage a we can then choose the right $R_{i_1}, \ldots, R_{i_m}$ in time linear to the number of sources available (using binary search). There are two serious problems with this solution:

- There is an exponential number of unions of which to estimate the size. In small enough integration scenarios it is not infeasible to do so, since secondary storage is these days available in abundance, and this information needs to be computed once. In particular, if we are integrating 20 sources, we need to keep 1 byte for each of 2^{20} subsets, or 1 gigabyte of data. But clearly this solution does not scale.

- Another important problem is that it is not easy to estimate these quantities: sampling methods cannot estimate union sizes directly. But we can use sampling to estimate source overlaps.

Using overlap information Given a table C of all possible source overlaps $p(\cap_{j=i_1,\ldots,i_m} R_j)$, we can always calculate any $p(\cup_{j=i_1,\ldots,i_m} R_j)$ from the entries of C using the inclusion-exclusion formula:

$$p(\cup_{j=i_1,\ldots,i_m} R_j) = \sum_j p(R_j) - \sum_{j<k} p(R_j \cap R_k) + \ldots + (-1)^m p(\cap_{j=i_1,\ldots,i_m} R_j)$$

Of course, C still requires exponential space.[5] Moreover, calculating even one $p(\cup_{j=i_1,\ldots,i_m} R_j)$ takes exponential time in the number of sources. We will discuss the space requirements later in the section. Let us briefly discuss efficient heuristics that use overlap information to generate efficient query plans for partial answer queries.

Algorithm for Partial Answer Query Plans Even if we have explicitly stored all k-wise overlaps for all $k \leq n$, we still need an efficient, if possible polynomial, algorithm for choosing m relations $R_{i_1}, \ldots, R_{i_m}$ such that $\sum_{j=i_1,\ldots,i_m} p(R_j)$ is minimized and $p(\cup_{j=i_1,\ldots,i_m} R_j) >= a$. This optimization problem is NP-complete by easy reduction from the exact set cover problem [GJ79] and is a variant on the set cover problem. The set cover problem is only approximable by a polynomial algorithm to a $\log n$ factor [Hoc97], where n is the size of the universe of the sets. A $(\log n)OPT$ polynomial time approximation to our problem is straightforward and is not presented for lack of space. We are currently investigating whether there is a provably better approximation algorithm for this problem. We are also planning to experimentally evaluate greedy algorithms for solving this problem.

Directions for Statistics Acquisition and Approximation The mediator will discover overlap statistics by analyzing the results of prior queries. It may also periodically sample the sources to derive statistics that cannot be derived with significant confidence from the results of prior queries. However, the novel challenge is to approximate the statistics; a possible approximation is to precompute a subset of the entries of C and use the maximum likelihood estimator of the others. The desiderata for this summarization are that it is space efficient and that it allows us to compute the coverage of unions without too much error in either direction — underestimating the coverage of a union means we are taking a hit in efficiency: we will end up computing a larger percentage

[5] A similar observation is made in [FKL97].

of the answer, at probably a higher cost. These approximations will have to be evaluated in a practical setting; we are looking for good average performance in real integration scenarios. Lower bounds for the error in the estimation of unions using approximate inclusion-exclusion are proven in [LN90]: for a union of n sets, if we are given all k-wise intersections of these sets for all $k \leq K$, *any* approximation of the union may err by a factor of $\Theta(n/K^2)$ if $K \leq \sqrt{(n)}$.[6]

The statistics acquisition and approximation problem is complicated by the fact that in practice queries will also impose selection conditions, say `make = 'Honda'`, on the source data. Clearly, it is very expensive to generate a separate set of statistics for each possible query. We should only keep statistics for predicates p that are significantly different than the (relevant) entries of C.[7] For example, assume that statistics information indicates that sites S_1 and S_2 have a 10% overlap on advertised cars and 60% overlap on advertised Hondas. In this case it is necessary to keep the Honda information because its estimate is very imprecise.

References

[AKH93] Y. Arens, C. A. Knoblock, and C.-N. Hsu. Retrieving and integrating data from multiple information sources. *International Journal of Intelligent and Cooperative Information Systems*, 2(2):127–158, 1993.

[AKL97] N. Ashish, C. A. Knoblock, and A. Levy. Information gathering plans with sensing actions. In *Fourth European Conference on Planning*, 1997.

[DG97] O. Duschka and M. Genesereth. Answering queries using recursive views. In *Proc. PODS Conf.*, 1997.

[DL97] O. Duschka and A. Levy. Recursive plans for information gathering. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 1997.

[FKL97] D. Florescu, D. Koller, and A. Levy. Using probabilistic information in data integration. In *Proc. VLDB Conf.*, 1997.

[FW97] M. Friedman and D. S. Weld. Efficiently executing information-gathering plans. In *Proc. IJCAI Conf.*, 1997.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability.* Freeman, 1979.

[GM$^+$97a] H. Garcia-Molina et al. The TSIMMIS approach to mediation: data models and languages. *Journal of Intelligent Information Systems*, 8:117–132, 1997.

[GM97b] M. Goldwasser and R. Motwani. Intractability of assembly sequencing: unit disks in the plane. In *Proc. of Workshop for Algorithms and Data Structures*, 1997.

[Hoc97] D. S. Hochbaum. *Approximation Algorithms for NP-hard Problems.* PWS Publishing Company, 1997.

[KW96] C. T. Kwok and D. S. Weld. Planning to gather information. In *Proc. AAAI Conf.*, 1996.

[6] If $K \geq \sqrt{(n)}$ then we can get a very good approximation, but this case is not very interesting, since we still need overlap information for $(\sqrt{2})^n$ source combinations.

[7] Notice that [FKL97] make the simplifying assumption that this is never the case.

[Lev96] A. Levy. Obtaining complete answers from incomplete databases. In *Proc. VLDB Conf.*, 1996.

[LMR90] W. Litwin, L. Mark, and N. Roussopoulos. Interoperability of multiple autonomous databases. *ACM Computing Surveys*, 22:267–293, 1990.

[LMSS95] A. Levy, A. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views. In *Proc. PODS Conf.*, pages 95–104, 1995.

[LN90] N. Linial and N. Nisan. Approximate inclusion-exclusion. In *Proc. STOC Conf.*, 1990.

[LRO96] A. Levy, A. Rajaraman, and J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proc. VLDB*, pages 251–262, 1996.

[LRU96] A. Levy, A. Rajaraman, and J. Ullman. Answering queries using limited external processors. In *Proc. PODS*, pages 227–37, 1996.

[LYV$^+$98] C. Li, R. Yerneni, V. Vassalos, H. Garcia-Molina, and Y. Papakonstantinou. Capability based mediation in tsimmis, 1998. Accepted for demonstration, SIGMOD98.

[PGMU96] Y. Papakonstantinou, H. Garcia-Molina, and J. Ullman. Medmaker: A mediation system based on declarative specifications. In *Proc. ICDE Conf.*, pages 132–41, 1996.

[RSU95] A. Rajaraman, Y. Sagiv, and J. Ullman. Answering queries using templates with binding patterns. In *Proc. PODS Conf.*, pages 105–112, 1995.

[S$^+$] V.S. Subrahmanian et al. HERMES: A heterogeneous reasoning and mediator system. http://www.cs.umd.edu/projects/hermes/overview/paper.

[TRV95] A. Tomasic, L. Raschid, and P. Valduriez. Scaling heterogeneous databases and the design of DISCO. Technical report, INRIA, 1995.

[VP] V. Vassalos and Y. Papakonstantinou. Expressive capabilities description languages and query rewriting algorithms. Stanford Technical Report.

[VP97] V. Vassalos and Y. Papakonstantinou. Describing and Using Query Capabilities of Heterogeneous Sources. In *Proc. VLDB Conf.*, pages 256–266, 1997.

[Wie92] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25:38–49, 1992.