MOMIS Goes Multimedia: WINDSURF and the Case of Top-K Queries [⋆]

Ilaria Bartolini[1], Domenico Beneventano[2], Sonia Bergamaschi[2], Paolo Ciaccia[1],
Alberto Corni[2], Mirko Orsini[3], Marco Patella[1], Marco Maria Santese[2]

[1] DISI - Università di Bologna - Italy
[2] Department of Engineering "Enzo Ferrari" - University of Modena and Reggio Emilia - Italy
[3] DataRiver - Spin-off of the University of Modena and Reggio Emilia - Italy

Abstract. In a scenario with "traditional" and "multimedia" data sources, this position paper discusses the following question: "How can a multimedia local source (e.g., Windsurf) supporting *ranking queries* be integrated into a mediator system without such capabilities (e.g., MOMIS)?" More precisely, "How to support ranking queries coming from a multimedia local source within a mediator system with a "traditional" query processor based on an *SQL*-engine?" We first describe a *naïve* approach for the execution of *range* and *Top-K* global queries where the MOMIS query processing method remains substantially unchanged, but, in the case of Top-K queries, it does not guarantee to obtain K results. We then discuss two alternative modalities for allowing MOMIS to return the Top-K best results of a global query.

1 Introduction

In this paper we discuss how to extend the MOMIS Data Integration System in order to query "traditional" and "multimedia" data sources. The proliferation of multimedia data, and the consequent need of their management and integration with traditional information, represents nowadays a critical issue in many different contexts, such as, for example, the medical one. The solution we propose here is based on Windsurf [2,4], a general framework for the efficient retrieval of complex multimedia data.

MOMIS is characterized by a classical wrapper/mediator architecture [10] based on a Global Virtual Schema (Global Virtual View - *GVV*) and a set of traditional data sources [7]. The data sources contain the real data, while the GVV provides a reconciled, integrated, and virtual view of the underlying local sources. A query over the GVV (*global query*) must be rewritten as an equivalent set of queries expressed on the local schemata (*local queries*), i.e., the mediator must translate global queries to the native contexts for source execution.

Considering scenarios where "traditional" and "multimedia" data sources coexist, this position paper addresses the following problem:

[⋆] An open source version of the MOMIS system is delivered by the academic UNIMORE spin-off DataRiver (`www.datariver.it`); this work is developed in the context of the "DataRiver Data Integrator" project, funded to DataRiver by the Italian Ministry of Economic Development, "Fondo per l'Innovazione Tecnologica - FIT Start-Up".

> "How to support *ranking (Top-K) queries* coming from a multimedia local source (e.g., Windsurf), keeping substantially unchanged the query processing method of the mediator system (e.g., MOMIS)?"

This is an important issue when the mediator system is not extensible/alterable, for some reasons, to support ranking queries coming from a multimedia local source. In [6] we proposed a solution where ranking queries can be expressed on the GVV without requiring multimedia processing capabilities at the mediator level (i.e., at the MOMIS System), since they are managed by the multimedia data management system MILOS [1]. However, this solution requires, at the mediator level, a completely new query processing method (based on the *MEDRANK* algorithm [8]) with respect to the one used for traditional data (based on a *Full Outer Join* operation and then performed by an SQL-engine).

Here we discuss two alternative approaches to support ranking queries on the GVV that fully exploit the capabilities offered by the Windsurf query engine and that do not require to alter the overall query processing logic adopted by MOMIS.

The paper is organized as follows. We briefly describe the MOMIS system and the Windsurf framework in Section 2. In section 3 we describe a *naïve* approach for the execution of multimedia queries which requires to modify neither Windsurf nor MOMIS; however this approach does not guarantee the completeness of results for Top-K queries, i.e., (much) less than K results might be returned. Then, in Section 4 we discuss the two alternative modalities for allowing MOMIS to also support Top-K queries.

2 Background

In this background section we briefly describe the MOMIS query processor and the Windsurf framework.

2.1 The MOMIS Query Processor

Query processing in MOMIS is performed in two steps:

Query Unfolding MOMIS follows a Global-As-View (*GAV*) approach based on the idea that the content of each class `G` of the GVV is characterized in terms of a view (called *mapping query*) over the sources; then the query translation is performed by means of *query unfolding*, i.e., by expanding a global query on a global class `G` of the GVV according to the definition of the mapping query.

Data Fusion Local queries are executed on the local sources and local queries results are fused to obtain the answer to the global query. To perform *Data Fusion*, we assume that *Object Identification*, i.e., the identification of the same object in different data sources, has been already performed and thus a shared object identifier (`ID`) exist among different sources. Multiple records with the same `ID` are fused into a single record by means of the Full Outer Join Merge operator proposed in [9] and adapted to MOMIS in [7]. Intuitively, it corresponds to the following two operations: (1) Computation of the Full Outer Join and (2) Application of the Resolution Functions (to solve conflicts).

To give an example, let us consider two relational sources, `L1` and `L2`, with schema `L1(ID,A,B)` and `L2(ID,A,C)`, respectively. The global class `G` obtained by integrating `L1` and `L2` is the relation `G(ID,A,B,C)`; for the common attribute `A`, the `AVG` resolution function is used, i.e., `A=AVG(L1.A,L2.A)`. Given the global query `Q`:

```
Q =  SELECT A,B
     FROM G
     WHERE A >= 18 AND B = 12
```

the Query Unfolder module produces the following local queries:

```
LQ1:                LQ2:
SELECT ID,A,B       SELECT ID,A
FROM   L1           FROM   L2
WHERE  B = 12
```

Intuitively, since the attribute `A` is defined as `AVG(L1.A,L2.A)`, the *global predicate* `A >= 18` cannot be *pushed* at the local sources because the `AVG` function has to be computed at a global level. On the other hand, since the attribute `B` is only present into `L1`, the *global predicate* `B = 12` can be pushed at the local source `L1`. The local queries are executed on the local sources; the data fusion performed by the Join Engine can be represented by the following full outer join query (we use the same symbol `LQ` for a local query and its result):

```
Q_FOJ =  SELECT ID, A=AVG(LQ1.A,LQ2.A),B
         FROM LQ1 LEFT OUTER JOIN LQ2 USING (ID)
         WHERE AVG(LQ1.A,LQ2.A)>=18
```

where the full join is simplified into a left join since the predicate `B = 12` can only be satisfied in `L1` and, then, an object of the result must necessarily be an object of `LQ1`.

2.2 The Windsurf framework

Windsurf is a general framework for the efficient retrieval of complex multimedia data which are at the heart of several modern applications, such as image/video retrieval and the comparison of collection of documents [2,4]. With the goal of allowing a seamless management of such data, Windsurf provides a unified model for the representation of complex multimedia data.

The Windsurf software library [5] (`www-db.disi.unibo.it/Windsurf/`) provides a framework for evaluating the performance of alternative query processing algorithms for efficient retrieval of multimedia data. Important features of the Windsurf library are its generality, flexibility, and extensibility. These are guaranteed by the appropriate instantiation of the different templates included in the library; in this way, each user can implement her particular retrieval model.[4]

The Windsurf framework offers a number of appealing features, including:

[4] The Windsurf library is released under the “QPL” license and is freely available for personal use, education and research purposes.

Extensibility and personalization Different types of low-level multimedia object representation, multimedia object segmentation, feature extraction, and local and/or global comparison criteria can coexist and be compared.

Efficient processing of distance-based and preference-based queries For distance-based queries the (dis)similarity between documents is numerically assessed by way of a *document distance function* d that combines together elementary distances between the constituting documents' elements. On the other hand, preference queries are based on the Skyline model [3] that does not rely on the specification of a numerical document distance function, rather document D_a is considered better than D_b for the query Q iff D_a does no worse than D_b on all query elements and there exists at least one query element on which D_a is strictly better than D_b. The result of a Skyline query necessarily includes those documents that would be the best alternatives according to some specific document distance function.[5]

For the purpose of this paper, we focus on Windsurf query processing functionalities when applied in the context of the well known content-based *image* retrieval (CBIR) problem. Given an image database, where each image I is described through some features representing its visual content, a query image Q and an image distance function d, that for each pair of images measures their dissimilarity (using their features), determine the set of *best* database images wrt Q. The rationale is that image I_a is considered better than I_b for the query Q iff $d(Q, I_a) < d(Q, I_b)$ holds.

The basic distance-based queries supported by Windsurf are:

Range queries Given a maximum distance threshold δ, return all images sufficiently close to the query, i.e., those images I for which $d(Q, I) \leq \delta$. Clearly, this requires some knowledge about distribution of distances with respect to the queries.

Top-K queries Given a maximum result cardinality K, return the K images closest to the query. This type of query is usually preferred over range queries because it is easier for the user to control the cardinality of the result.

nextNN queries Performs a sorted access to the image database, i.e., images are returned one-by-one sorted by non decreasing values of $d(Q, I)$.

3 A *naïve* approach for integrating traditional and multimedia data

In order to integrate the capabilities of Windsurf within the MOMIS architecture, we introduce the concept of Windsurf Local Source (*WLS*) to represent and query, within the MOMIS system, a local source managed by the Windsurf system. The local schema of *WLS* is the relation `WLS(ID,IMAGE,DISTANCE)` where `ID` is the shared identifier, `IMAGE` is the image returned by Windsurf, and `DISTANCE` is computed according to the (dis)similarity with the target image. Windsurf does not include the shared identifier `ID` and uses an internal image identifier `IMG_ID`; the (partial) association between `ID` (objects) and `IMG_ID` (images) is given by the relation `RM(ID, IMG_ID)`; the relation

[5] In this paper we only consider distance-based queries.

WLS(ID,IMAGE,DISTANCE) is then obtained by joining the RM(ID,IMG_ID) table with the data managed by Windsurf.

The *naïve* approach to perform range and Top-K global queries does not require any modification of the Windsurf system and no changes to the logic of the MOMIS Query Processor, i.e., the new local source WLS is simply included into the MOMIS Query Processing (described in section 2.1). For instance, with reference to the example in Section 2.1:

1. The *multimedia* global class MG obtained by integrating L1, L2 and WLS is the relation MG(ID,A,B,C,IMAGE,DISTANCE).
2. The *multimedia* global query MQ is obtained by adding to the "traditional" global query Q a range or Top-K query specification. A multimedia local query LQM with such specification is produced by the unfolding process.
3. LQM is executed on WLS and its result is joined with the traditional query Q_FOJ, so obtaining the MQ result:

   ```
   MQ = SELECT *
        FROM Q_FOJ JOIN LQM USING(ID)
   ```

   A join operation is used since an object of the MQ result must necessarily be an object of both the traditional query Q_FOJ and the multimedia query LQM.
   It has to be clear that, in the case of Top-K queries, MQ does not guarantee to obtain K results, since an image returned by Windsurf is not guaranteed to join an object in the result of Q_FOJ. As a consequence, in the worst case it is possible that MQ returns no results at all.

This *naïve* approach is fully implemented in the context of the "DataRiver Data Integrator" project. From an implementation point of view, the MOMIS mediator is on a network server and every MOMIS wrapper can be run on a different network server and accessed by a web services protocol. This layout is also applied in the case of a *Windsurf wrapper* (the software module that interfaces MOMIS to Windsurf).

4 Supporting Top-K Queries in MOMIS

In this section we discuss two alternative modalities for allowing MOMIS to return the Top-K best results of a global query. We remind (see Section 3) that the relation WLS(ID,IMAGE,DISTANCE) is actually obtained by joining the RM(ID,IMG_ID) table with the data managed by Windsurf (which does not include the shared identifier ID).

4.1 The Semi-join Exact Method

The first method we consider is able to deliver the correct result, yet it requires to perform some modifications to both MOMIS and Windsurf. The idea goes as follows:

1. First, MOMIS executes the global query *without* any reference to WLS, i.e., no multimedia data and ranking are considered. This step yields the table Q_FOJ(ID,...) with the shared identifier ID and all the attributes needed for the final result.
2. Second, Q_FOJ is projected on ID, yielding a table RES(ID).
3. MOMIS then sends to the Windsurf wrapper the local query LQM by providing as input parameters, besides the target image and the required number of results, K, also the list of shared identifiers, RES(ID).
4. By means of the RM table, the Windsurf Wrapper transforms the list into a corresponding list of image identifiers, FILTER(IMG_ID). Notice that the cardinality of this list may be less than that of RES(ID) if some tuples do not have a matching image.
5. Windsurf then executes a series of nextNN calls (see Section 2.2), adding a returned image to its result only if the corresponding IMG_ID is also present in the FILTER list.
6. Eventually, K images are returned to the wrapper, which will use them to populate the LQM result table and return it to MOMIS.
7. As a last step, the join between Q_FOJ(ID,...) and LQM is performed and the result of the global query returned to the user.

By definition, this method guarantees that the correct result is always obtained. Because of step 5, it can well be regarded as a *semi-join strategy*, in which the list of joinable values, FILTER(IMG_ID), is used by Windsurf to avoid returning to MOMIS non-matching images.

From an implementation point of view, the semi-join method requires to slightly extend Windsurf, by making it able to filter results on the basis of an IMG_ID list, and also to slightly alter the normal flow of execution of the MOMIS Query Processor. However, the overall logic of query processing remains unaltered.

4.2 An Estimate-based Approximate Method

The second method we discuss requires no modification of the Windsurf system and no changes to the logic of the MOMIS Query Processor. Rather, a new module able to exploit data statistics, and also incorporating a probabilistic model, needs to be developed. Although this method can sometimes yield less than K results, this can be controlled through the usage of statistics and a probabilistic model.

The intuition about this second method is that, should one be able to guess how many tuples are in RES (see step 2 above) and how many of them have a matching image, then one could use this information to retrieve from Windsurf a number K' ($K' > K$) of images so that at least K of them are guaranteed, with high confidence, to have a match in RES (and thus in Q_FOJ).

In order to simplify the presentation, and without loss of generality, we consider that, besides the Windsurf local source, there is only another traditional source, LS(ID,A,B,...). Further, we assume that the number of images, I, indexed by Windsurf and the number of matching pairs, M, in the RM(ID,IMG_ID) table are both known quantities.

We start by analyzing the simplest case in which no predicate is present for LS, i.e, RES coincides with LS. In this scenario the only reason that could lead to discard from the result an image returned by Windsurf is that such image has no match in the `RM(ID,IMG_ID)` table. Let $P = M/I$ denote the fraction of images with a match in `RM`, i.e., P is the probability that a randomly chosen image will be present in `RM`. Assuming that having a match in `RM` and belonging to the result of a ranking query are two independent events, it is immediate to derive that setting $K' = K/P$ will yield, on the average, K results for the global query.

Setting $K' = K/P$ is indeed a simplistic way of proceeding, since it provides no guarantees on the likelyhood that, for a specific query, the number of results will indeed be at least K. To obviate this, the key observation is that, given K' images returned by Windsurf, the number of them with a match in `RM` is a random variable X with *hypergeometric distribution* and parameters I (no. of images), M (no. of images with a match), and K' (no. of selected images), that is:

$$\Pr_H\{X = x\} = \binom{M}{x}\binom{I-M}{K'-x}\bigg/\binom{I}{K'} \tag{1}$$

Since usually it is $K' \ll I$, the above can be approximated with negligible errors by the simpler *binomial distribution*, with parameters K' and $P = M/I$:

$$\Pr_B\{X = x\} = \binom{K'}{x} P^x (1-P)^{K'-x} \tag{2}$$

Then, by setting K' so that $\Pr_B\{X < K\} \leq \varepsilon$, we obtain that, with confidence at least $1 - \varepsilon$, K results are indeed returned. In order to compute $\Pr_B\{X < K\}$ one could either directly use Equation 2 or resort to known bounds for the tail of the binomial distribution. For instance, Hoeffding's inequality applied to our scenario yields:

$$\Pr_B\{X < K\} \leq \exp\left(-2\frac{(K'P - (K-1))^2}{K'}\right) \tag{3}$$

which can be easily solved in the K' unknown. As an example, when $P = 0.5$ and $K = 10$, in order to have $\varepsilon = 0.01$ it has to be $K' \geq 37$, whereas $K' \geq 43$ ensures, with a 99.9% confidence, that at least 10 images are returned.

Let us now consider the general case in which one or more predicates on `LS` are present. It should be clear that, in order to apply probabilistic arguments similar to those above described, one needs an estimate of how many tuples in `LS` satisfy the predicates, i.e., the *selectivity* F of these predicates. Given F, and assuming that the tuples satisfying the predicates are independent of those with a matching image, the number of *candidate tuples* can be estimated as $F * M$ (where M we remind is the cardinality of `RM(ID,IMG_ID)`). The problem then reduces to determine a value of K' high enough to guarantee, with confidence at least $1 - \varepsilon$, that at least K of such images will find a match among the candidates.

From an implementation point of view, this method implies no changes at all to Windsurf, whereas it requires that MOMIS be slightly extended so as to estimate the selectivity of a query. Further a module in charge of implementing the above-sketched probabilistic model is needed to derive a proper value for the number of images, K', to be requested to Windsurf. Again, no major changes in the query processing logic of MOMIS are required.

5 Conclusions

We have discussed two alternative solutions for extending a mediator system like MOMIS so as to support ranking multimedia queries. Both solutions are appealing in that the needed changes to MOMIS query processing logic are minor ones.

For simplicity of exposition, in this paper we have considered only the case in which a single multimedia source is present. However, our solutions can be easily extended to the case of multiple multimedia sources, again guaranteeing that ranking issues will not affect MOMIS logic.

References

1. Amato, G., Gennaro, C., Rabitti, F., Savino, P.: Milos: A multimedia content management system for digital library applications. In: Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science, vol. 3232, pp. 14–25 (2004)
2. Ardizzoni, S., Bartolini, I., Patella, M.: Windsurf: Region-based image retrieval using wavelets. In: 10th International Workshop on Database & Expert Systems Applications, Florence, Italy, September 1-3, 1999, Proceedings. pp. 167–173 (1999)
3. Bartolini, I., Ciaccia, P., Patella, M.: Efficient sort-based skyline evaluation. ACM Trans. Database Syst. 33(4) (2008)
4. Bartolini, I., Ciaccia, P., Patella, M.: Query processing issues in region-based image databases. Knowl. Inf. Syst. 25(2), 389–420 (2010)
5. Bartolini, I., Patella, M., Stromei, G.: The Windsurf library for the efficient retrieval of multimedia hierarchical data. In: SIGMAP 2011 - Proceedings of the International Conference on Signal Processing and Multimedia Applications, Seville, Spain, 18-21 July, 2011. pp. 139–148 (2011)
6. Beneventano, D., Gennaro, C., Bergamaschi, S., Rabitti, F.: A mediator-based approach for integrating heterogeneous multimedia sources. Multimedia Tools and Applications 62(2), 427–450 (2013)
7. Bergamaschi, S., Beneventano, D., Guerra, F., Orsini, M.: Data integration. In: Embley, D.W., Thalheim, B. (eds.) Handbook of Conceptual Modeling: Theory, Practice and Research Challenges. Springer Verlag (2011)
8. Fagin, R., Kumar, R., Sivakumar, D.: Efficient similarity search and classification via rank aggregation. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. pp. 301–312. SIGMOD ’03, ACM (2003)
9. Naumann, F., Freytag, J.C., Leser, U.: Completeness of integrated information sources. Inf. Syst. 29(7), 583–615 (2004)
10. Wiederhold, G.: Intelligent integration of information. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993. pp. 434–437. ACM Press (1993)