# Optimization Query Process of Mediators Interrogation Based on Combinatorial Storage

L. Cherrat, M. Ezziyyani, M. Essaaidi
University Abdelmalek Essaadi, LaSIT
F.S. Tetuan, Morocco

*Abstract*—**In the distributed environment where a query involves several heterogeneous sources, communication costs must be taken into consideration. In this paper we describe a query optimization approach using dynamic programming technique for set integrated heterogeneous sources. The objective of the optimization is to minimize the total processing time including load processing, request rewriting and communication costs, to facilitate communication inter-sites and to optimize the time of data transfer from site to others. Moreover, the ability to store data in more than one centre site provides more flexibility in terms of Security/Safety and overload of the network. In contrast to optimizers which are considered a restricted search space, the proposed optimizer searches the closed subsets of sources and independency relationship which may be deep laniary or hierarchical trees. Especially the execution of the queries can start traversal anywhere over any subset and not only from a specific source.**

***Keywords—Mediation; Datawarehouse; Optimisation; Classification***

## I. Introduction

The challenge created by the increase and the diversity of information sources on the web, and by the need of organizations to interoperate database systems not only consists of the need to use tools for integrating data [3][5][9][10] among multiple users and heterogeneous information sources, but also the necessity of these tools to overcome the limitations of current search engines by allowing not only users to ask queries more sophisticated than simple keywords, but also being able to aggregate other elements of answers from different sources to build, in the most optimized possible way by time and space research, the analytical global response to the user query. This need is becoming increasingly relevant for medical information, especially with the existence of a multitude of web sources specific to medicine areas and the trend towards computerization of patient medical records [2].

Since query processing of data integration [1][6] [11][12] requires access to the data from numerous wide distribution sources over network, it is crucial to investigate how to deal with the expensive communication over head and the response time. In this paper, we present an efficient approach for processing distributed sources with the existence of an execution order graph [2]dependency of the integration system. In the first of a given set of sources, the algorithm classifies the integrated sources into non-exclusive groups (local data warehousing), such that the associate operations can be locally processed without data transfer. Local data warehousing offers many benefits: reduced costs, increased flexibility, and simplified data access with greater agility. Indeed, local data warehousing offers power to interrogate several centralized sources, but also the possibility to analyze the data more efficiently and with low cost on any server based on availability and needs. This solution effectively enabling more users to access more and more data with more ease. Thanks to the Distributed Databases Solution, we can migrate critical data on data centers and improve the response time of readjustment and equilibration of the data distribution. In this perspective, the use of the principle of local data warehousing report a very suitable solution for the integration systems [8].

Our goal is to create disjoint subsets of sources with low coupling the maximum possible. The question is: on what criteria we will classify sources into a disjoint data warehouse? To do this, we develop a relevant algorithm for grouping the sources into subsets based on a new classification method that we propose in this paper.

In the remainder of this paper, we start with Section II by introducing our query optimization method based on the sources classification and the used classification techniques. We next, in Section III, present the sources classification principle and the generated algorithms. In Sections IV, we develop a new method for refining the regrouping result in the aim to readjust the subsets generated by our hybrid classification algorithm. We then in section V, study the performance of data transmission on the network during the interrogation of the mediator with the presence of local data warehouses generated by the algorithm that we proposed. Section VI concludes with future agenda.

## II. Query Optimization Based on the Sources Classification

In order to optimize the process of querying sources integrated by the mediator [2][4], we proceed to the construction of partitions of a set of homogeneous sources with known distances and similarities between pairs of sources. Both functions are defined by the degree of dissimilarity and similarity [24] between sources based on the structure of the schema and the data recorded in sources. To do this, we use the approaches and methods of partitioning based on the optimization algorithm which allows us to find a lower cost solution for each partition with the consideration of the homogeneity of the sources in the same partition. Generally, each partition founded by the global optimization algorithm cannot meet the basic constraints of the predefined partitioning. The algorithm then, proceeds to the error correction intra-partition. Such a process is called refining process, which consists of the refining of a partition to increase its homogeneity.

Refining algorithms are used to distribute the sources in the partitions satisfied the constraints of homogeneity and distribution and they have two common objectives: (i) find a partition such that the objective functions, distance and similarity, take respectively the minimum and maximum value. (ii) find a partition such that the variance of homogeneity partitioning is respected as much as possible. The difference between partitioning methods vary according to the order of priorities between these two objective functions. In our algorithm, we give more importance to minimize the distance function, when the similarity functions [19] (load distribution), it will have as a primary goal to respect the homogeneity of inner-partitions. In this paper, we propose two new methods for classification using a hybrid combination of the two following classic classification techniques [23] :

*a) Hierarchical approach*: It is based on the following principle: create a set of partition distributed hierarchically into disjoint groups (ie into partitions with less and less parts). Each new partition is obtained by successive grouping of parts of the partition immediately preceding in the hierarchy. The sets of sources are divided into two groups to form a tree whose top node is represented by the set of sources and the subset element by the two partitions and so on for each subset created.

*b) Mobile centers Method:* This is an iterative method that consists of calculating the center of gravity for each part of the partition, and to recreate a partition where each part consists of the nearest elements to the center of gravity. The center of gravity is calculated based on the weight of the global schema. The distance between the global schema and a source is calculated based on the similarity function between the source and the global schema. The next section presents our hybrid method for partitioning sources.

## III. Sources Classification Rules

The natural solution to this question is to maintain a distributed data warehouse, consisting of multiple local sources adjacent to the collection points, together with a coordinator. In order for such a solution to make sense, we need a technology for the data classification process [7]. We have developed a new algorithm for this task. This algorithm translates a set of sources into distributed distinct subsets and generates distributed data warehouses, with the following rules: (i) each generated data warehouse performing some computation and communicating the query result to the coordinator, and (ii) the coordinator synchronizing the results and (possibly) communicating with the data warehouses. The semantics of the subqueries generated by system ensure that the amount of data that has to be shipped between data warehouses are independent and use the underlying data. The solution allows for a wide variety of optimizations that are easily expressed in the interrogation and thus readily integrated into the query optimizer. The optimization algorithm included in our prototype contributes to the minimization of synchronization traffic and the optimization of the data processing at the local sites. Significant features of the this approach are the ability to perform both distribution and optimization that reduce the data transferred and the number of evaluation rounds.

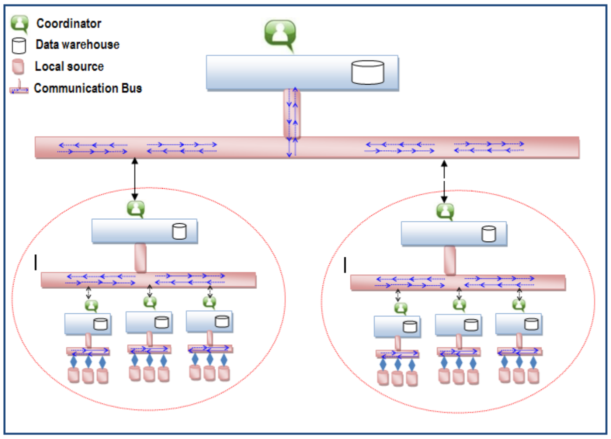

Fig .1. Sources communication

### A. *Principle of classification*

The basic idea of this solution is: data in the network is transmitted as a small fragment from set of sources to others, which is obviously a non-redundant way. When data is transferred to another venue, not every datum is involved in connection operation nor useful. Therefore, the data is not involved in the connection, for useless data needs not be transmitted circularly in the network. The basic principle of this optimization strategy is to use the local data warehouse to only transmit the data involved in the connection in the network as far as possible.

The interrogation of the hybrid mediator generally performed in accordance with the created relationships between local data warehouses. Its advantage and deficiency is not considered how to optimize the order of the sub-query to further reduce the network communication costs. But we consider this task it was taken on consideration at the order optimizer process. The solution in this paper is presented according to the deficiency of general algorithm, that is, through the cost estimate to generate the local data warehouses and interrogation process to improve how to further reduce the transfer data cost of sub-query. In this paper we will demonstrate, the results data generated by performing all the sub-queries and generating the final result are regarded as the decisive factor of the creation of the local data warehouses, and the optimization benefits of the order execution.

### B. *Source classification algorithms.*

In this section, we propose a new method for the classification of sources based on the principle of the top-down hierarchical method and the mobile centers method. Indeed, this hybrid classification method is based on the knowledge of a distance function and a function of dissimilarity between all pairs of sources of the set integrated by the mediator. In the first, we propose a solution which is based on the principle of top-down hierarchical in the perspectives to improve it with the introduction of the method of mobile centers.

To do this, we define a function that calculates the distance between pairs of sources. However, there is no immediate relationship between distances for all sources of a graph of sources. If the relationship of a distance can be established, it is generally very expensive to implement, especially for non-related graphs.

Therefore, classification methods by graph partitioning are generally impracticable. To some extent, the ascending hierarchical methods could be used without the knowledge of the distance between each source. In this case, they will work nearer to nearer from the known distances between neighboring elements. In this adaptation, each element is a top of the graph, and the distance between neighbors is the cost associated with the edge connecting this top to another top. In fact, such partitioning approaches for graph of sources, are known as the methods of expanding region.

### *C. Used Functions for the classification of sources*

In this section, we are interested to grouping the sources into subsets such that the sources of the same set react similarly to changes of user queries. These problems are often treated with automatic classification methods [20][21] to identify groups of data sources with a homogeneous behavior or quasi-homogeneous to generate a result for the same query to form groups of homogeneous sources, i.e. groups of sources such as sources are as similar as possible within a group (compactness criterion), and the groups are as dissimilar of the similarity and the dissimilarity is based on the set of the following variables :

as possible (criterion of dissimilarity). The measurement The structure of the database schema.

- The nature and number of attributes of entities.
- The size and occurrence of records.
- The inter- entities relationship.
- Results of requests for canonical query (Standard).

#### *1) Distance Function:*

- $\mathrm{NbrE}(\mathrm{S_i})$ **:** is the number of entities in the source $\mathrm{S_i}$
- $\mathrm{NbrE}(\mathrm{S_j})$ : is the number of entities in the source $\mathrm{S_j}$
- $\mathrm{NbrAtt}\left(\mathrm{E_k^{S_i}}\right)$ : is the attributes number of the entity $\mathrm{E_k^{S_i}}$
- $\mathrm{NbrAttId}\left(\mathrm{E_k^{S_i}}, \mathrm{E_l^{S_j}}\right)$ : is the number of the identical attributes between the two entities$\mathrm{E_k^{S_i}}$ and $\mathrm{E_l^{S_j}}$.

#### *2) Similarity function and coupling function*

To measure the similarity between two sources, we adopt the cosinus rules [26]. With the sets are  the sources and elements are the entities. Therefore, we define a similarity function between two sources: it is the report of intergroup rapprochement. The second function is the function allowing to calculate the intra-sources coupling ratio between two sources of the same group. Both functions are based on the weight of the source for each entity. In the next section, we present the data mathematical model used to define these functions.

Let $\mathrm{S} = \mathrm{E_1}, \mathrm{E_2} \ldots, \mathrm{E_N}$ a set of entities of the source S. We define the weight of the entity by the number of attributes, the

We define in this section the Distance function [25] between two sources $\mathrm{Distance}\left(\mathrm{S_i}, \mathrm{S_j}\right)$ which is mainly based on the difference between the metadata of the two sources. Indeed, the value of the distance function depends on the number of distinct attributes between all pairs of entities from two sources. The principle of this function is to calculate the distance between two vectors in space. To do this, we assume that each source is a vector whose coordinates are the entities of the source. Thus, the distance between the two sources is the Euclidean Distance between two vectors. We therefore define this function as follows:

$$Distance\left(S_i, S_j\right) = \sqrt[2]{\sum_k^{NbrE(S_i)} \sum_l^{NbrE(S_j)} (DE(E_k^{S_i}, E_l^{j}))^2} \quad 1)$$

With:

$DE(E_k^{S_i}, E_l^{S_j})$ : Is the distance between the entity $\mathrm{E_k^{S_i}}$ of the source $\mathrm{S_i}$and $\mathrm{E_l^{S_j}}$ of the source $\mathrm{S_j}$.  such as:

$$DE\left(E_k^{S_i}, E_l^{S_j}\right) = \frac{NbrAtt\left(E_k^{S_i}\right) - NbrAttId\left(E_k^{S_i}, E_l^{S_j}\right)}{NbrAtt\left(E_k^{S_i}\right) + NbrAtt\left(E_k^{S_i}\right)}$$

$$+ \frac{NbrAtt\left(E_l^{S_j}\right) - NbrAttId\left(E_k^{S_i}, E_l^{S_j}\right)}{NbrAtt\left(E_k^{S_i}\right) + NbrAtt\left(E_k^{S_i}\right)}$$

$$DE\left(E_k^{S_i}, E_l^{S_j}\right) = \frac{NbrAtt\left(E_k^{S_i}\right) + NbrAtt\left(E_l^{S_j}\right)}{NbrAtt\left(E_k^{S_i}\right) + NbrAtt\left(E_k^{S_i}\right)}$$

$$+ \frac{-2 * NbrAttId\left(E_k^{S_i}, E_l^{S_j}\right)}{NbrAtt\left(E_k^{S_i}\right) + NbrAtt\left(E_k^{S_i}\right)}$$

With:

number of recordsets and the relation with the other entities of the same source.

$$\mathrm{P_E} = \sum_{\mathrm{i}} (\mathrm{Card(E)}) * (\|\mathrm{E}\| - \mathrm{Nb_{FK}(E)}))$$

$\mathrm{Card(E)}$ : Number of recordsets of the entity E.

$\|\mathrm{E}\|$ :  Number of Attributes of the entity E.

$\mathrm{Nb_{FK}(E)}$: Number of external key of the entity E.

$\mathrm{Similitude}(\mathrm{S_i}, \mathrm{S_j})$ is the degree of similarity between two sources Si and Sj. that is to say, the similarity between two sources regarding the schema structure  constituting the two sources In this case, the value of the $\mathrm{Similitude}(\mathrm{S_i}, \mathrm{S_j}) \in$ [0,1]. To calculate the similarity between the two sources, we use the Cosinus similarity. Indeed, given two sources $\mathrm{S_i}$ and $\mathrm{S_j}$. The similarity $\mathrm{Cosinus}\left(\theta_{(i,j)}\right)$ is represented by using a scalar product and a grandeur value, which is defined as follows:

$$\mathrm{Similitude}(\mathrm{S_i}, \mathrm{S_j}) = \mathrm{Cosinus}\left(\theta_{(i,j)}\right) = \frac{\mathrm{S_i} \cdot \mathrm{S_j}}{\|\mathrm{S_i}\| \|\mathrm{S_j}\|}$$

$$= \frac{\sum_{k}^{n}(P_{E_i^k}) \times \left(\sum_{l}^{m}(P_{E_i^l})\right)}{\sqrt{m \times \sum_{p}^{n}((P_{E_i^k})^2} \times \sqrt{n \times \sum_{l}^{m}(P_{E_i^l})^2}}$$

The resulting similarity ranges which tend to 0 means exactly that two sources are disjoint. If the value is 1, it means that the two sources are identical. For other values, it indicates the degree of similarity or dissimilarity between the two sources.

Subsequently, we define the coupling function between two sources $S_i$ and $S_j$, signify the probability of executing a query with the interrogation of the two sources $S_i$ and $S_j$ to generate the result. To do this, we use the Jaccard similarity coefficient [25][26]. The Jaccard coefficient measures the similarity between two sources, it is defined as the ratio of the number of common attributes between the two sources on the number of the union of attributes of two sources:

$$Jaccard(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \quad (1)$$

The Jaccard distance $Jaccard(S_i, S_j)_\delta$ measure the dissimilarity between sets. It consist simply to subtract the Jaccard coefficient to 1 $(1 - Jaccard(S_i, S_j))$. Therefore, the coupling function between two sources Sj and Si is a function that gives the degree of similarity between two sources (inter-group) belonging to the same group. This is the relation between the similarity and the distance between two sources proportionally to the weight of the intersection of the two sources.

$$Coupling(S_i, S_j) = P_{S_i \cap S_j} \times \frac{Jaccard(S_i, S_j)}{Jaccard(S_i, S_j)_\delta}$$

$$Coupling(S_i, S_j) = P_{S_i \cap S_j} \times \frac{Jaccard(S_i, S_j)}{1 - Jaccard(S_i, S_j)}$$

$$Coupling(S_i, S_j) = P_{S_i \cap S_j} \times \frac{Jaccard(S_i, S_j)}{\frac{|S_i \cup S_j| - |S_i \cap S_j|}{|S_i \cup S_j|}}$$

$$Coupling(S_i, S_j) = P_{S_i \cap S_j} \times |S_i \cup S_j| \times \frac{Jaccard(S_i, S_j)}{|S_i \cup S_j| - |S_i \cap S_j|}$$

$$Coupling(S_i, S_j) = P_{S_i \cap S_j} \times |S_i \cup S_j| \times \frac{Jaccard(S_i, S_j)}{Distance(|S_i \cup S_j|, |S_i \cap S_j|)}$$

Note: The function $Coupling(S_i, S_j,)$ is used during the process of refining and readjustment groups (section V).

### *D. First classification method""*

This classification method into clusters seeks to find for each source, all other sources such as distances to this source is minimal and the similarity is maximal. To do this, we use the deviation parameter of the distance $\boldsymbol{\varepsilon_{S_i}^{Dis}}$ and the similarity $\boldsymbol{\varepsilon_{S_i}^{Sim}}$ for the sources **$S_i$**.

The deviation of distances for a given source **$S_i$** is :

$$\boldsymbol{\varepsilon_{S_i}^{Dis}} = \sqrt{\frac{1}{NbrS - 1} \sum_{j=k}^{NbrS} \left(Distance(S_i, S_j) - Moy_{Dis}(S_i)\right)^2}$$

$$Moy_{Dis}(S_i) = \frac{1}{NbrS} \sum_{j=k}^{NbrS} \left(Distance(S_i, S_j)\right)$$

With: NbrS : is the total number of sources

The Deviation for similarities for a given source **$S_i$** is:

$$\boldsymbol{\varepsilon_{S_i}^{Sim}} = \sqrt{\frac{1}{NbrS - 1} \sum_{i=k}^{NbrS} \left(Similitude(S_i, S_i) - Moy_{Sim}(S_i)\right)^2}$$

$$Moy_{Sim}(S_i) = \frac{1}{NbrS} \sum_{j=k}^{NbrS} \left(Similitude(S_i, S_j)\right)$$

With: NbrS : is the total number of sources

We define the group of the source **$S_i$** by the intersection of the two groups $G_{S_i}^D$ et $G_{S_i}^S$ as follows:

$$\boldsymbol{G_{S_i}^D = \{S_j \in S \,/\, Min_{S_i}^{Dis} - \varepsilon_{S_i}^{Dis} < Distance(S_i, S_j) < Min_{S_i}^{Dis} + \varepsilon_{S_i}^{Dis}\}}$$

and

$$\boldsymbol{G_{S_i}^S = \{S_j \in S \,/\, Max_{S_i}^{Sim} - \varepsilon_{S_i}^{Sim} < Similitude(S_i, S_j) < Max_{S_i}^{Sim} + \varepsilon_{S_i}^{Sim}\}}$$

thus :

$$\boldsymbol{G_{S_i} = G_{S_i}^D \cap G_{S_i}^S}$$

To generate different classification groups with a recursive manner, we follow the following steps:

1) *Initialize G by set of sources.*
2) *Wile G $\not\equiv \phi$*
- Select one source $S_x \in G$, we search group $G_{S_x}$ identical to $G_{S_i}$. Thus, the group is the union of all groups.
- $G \Leftarrow G \setminus G_{S_x}$

Below the generic algorithm of the classification method presented in the previous section:

*1) Selection of distances and the grouping method.*

*2) Calculating the distance between all pairs of individuals (matrix).*

*3) Each individual is considered a cluster.*

*4) Research of the two clusters to combine (cf. clustering method [14][15][16][22]).*

*5) Merging of the two clusters and update the distance matrix.*

6) *Repeat from steps 4 until you have only one cluster*.

*1) Analysis of result.*

The problem is due to the global differences in the degrees of belonging of a source to a given set. It may happen that a source has a distribution of entities similar on two sets, but for one of the two, the degree of belonging is always smaller than the other. We can consider that this source stores the same data and that one of the two sources includes the other, or that one source have a cardinality less than the other. However, as the Euclidean distance is based on absolute differences, these two sources are probably distant and therefore classified in different categories. We say that there is a "Size Effect".

We can overcome this problem by generating two new sources by bursting the source in question. But this transformation does not solve all problems. Indeed, if several variables are related to the same underlying phenomenon, they will be correlated between them and provide the same information several times.

To avoid this drawback, this method can be improved by the use of, on one hand, a fixed number of predefined subsets. Each subset has a center of gravity represented by the local schema. On the other hand, by the separate use of the distance function and the similarity (see the following section) that engages the first for defining sets and the second for correcting error of intra-group belonging.

## E. The second method " Gravity Center"

This algorithm aims to build a set of disjoint partitions of all the integrated sources. At the beginning of the algorithm, it is necessary to fix a number k of groups and choose an initial partition. The number of the partition can be inspired by a priori knowledge of the application areas integrated by the mediator. In this method, we adopt the rules of the center of gravity based on the sources local schemas (SL) for all predefined sub-domains. This requires prior knowledge of the primary domain integrated by system and the sub-domains its which composed with local schemas. For each sub-domain, we define a local schema to represent the center of gravity $\mathrm{Cg_i}$ for a group around the center. Then, based on both distance and similarity functions presented in Section V3, to seek all sources belonging to this group. To do this, we minimize the distance and we maximize the similarities between the sources and the center of gravity of each group according to the values of the deviations $\varepsilon_{\mathrm{G_i}}^{\mathrm{Dis}}$ and $\varepsilon_{\mathrm{G_i}}^{\mathrm{Sim}}$. The calculation of the latter depends on the number of sources, the number of groups and the average distance from sources to gravity center. The process starts by the generation of the group whose gravity center has the greatest weight while taking all sources into account. Subsequently, the second group will be formed with the inclusion of unaffected sources to the previous groups, and so on. The group's center of gravity $\mathrm{Cg_i}$ is the intersection of the two groups $\mathrm{G_{S_i}^D}$ and $\mathrm{G_{S_i}^S}$ such:

$$G_{g_i}^{Dis} = \{S_j \in S \,/\, \boldsymbol{Dis}(Cg_i, \boldsymbol{S_j}) < \boldsymbol{\varepsilon_{G_i}^{Dis}} \text{ and } \boldsymbol{S_j} \notin G_{g_k}^{Dis}, \forall k < i\}$$

And

$$G_{g_i}^{Sim} = \{ S_j \in S \,/\, \boldsymbol{\varepsilon_{G_i}^{Sim}} < \boldsymbol{Sim}(Cg_i, \boldsymbol{S_j}) \text{ and } \boldsymbol{S_j} \notin G_{g_k}^{Sim}, \}$$

We assume that $P_{G_{g_i}^{Dis}} < P_{G_{g_j}^{Dis}}, \forall i < j$ and $P_{G_{g_i}^{Sim}} < P_{G_{g_j}^{Sim}}, \forall i < j$

With $\varepsilon_{\mathrm{G_i}}^{\mathrm{Dis}}$ and $\varepsilon_{\mathrm{G_i}}^{\mathrm{Sim}}$ are the standard deviations of the set S.

Thus : $Gg_i = G_{g_i}^{Dis} \cap G_{g_i}^{Sim}$

The classification algorithm using the method of gravity center is as follows:

*1) Initialize $\boldsymbol{S}$ by all sources.*

*2) Determine all the centers of gravity $\boldsymbol{Cg_i}$ represented by the local schema of application sub-domain (K centres).*

*3) Calculate the weight of each center of gravity.*

*4) Sort the K centers of gravity in descending order by weight.*

*5) For each center of gravity $\boldsymbol{Cg_i}$ , calculate the standard deviations of the set $\boldsymbol{S}$ : $\boldsymbol{\varepsilon_{G_i}^{Dis}}$ **and** $\boldsymbol{\varepsilon_{G_i}^{Sim}}$*

*a. Compute the set$G_{g_i}^{Dis}$.*

*b. Compute the set$G_{g_i}^{Sim}$.*

*c. Determine the group $Gg_i = G_{g_i}^{Dis} \cap G_{g_i}^{Sim}$.*

*d. Initialize $S = S \setminus Gg_i$.*

*e. If $S = \{\emptyset\}$, $Exit\ \ loop$.*

# IV. Refining the Regrouping Result

## A. Refining principle

The execution of the hybrid classification algorithm that we proposed in the previous section can automatically generate a set of groups (subsets) that respects the basic constraints defined by the objective function of the hybrid classification algorithm, but does not take into account the general context of the application domain. Therefore, two sources of the same subset generated by the algorithm may have a low semantic relationship, but belong to the same subset according to the principle of the gravity center classification algorithm used in our algorithm. Otherwise, two different sources can have a strong semantic relationship between them, but belong to two different subsets. This means that a refining processor is essential for readjustment of subsets generated by our classification algorithm.

This step aims to minimize the cost of data exchange during the execution of subqueries on geographically remote sites. We propose in this section, the refining process with double treatment: Inter-subset and intra-subset. To define this refining process, we describe in the next section a coupling function between two subsets which gives the degree of correlation (and/or isolation) between groups. Generally, we separate between three possible situations:

### *1) Isolated Subset*

Isolated subset is a subset without data replication which has very low coupling (NULL) with other subsets generated by the algorithm of classification. In this case, we can ignore the cost of exchanges between the two subsets. Therefore, we do not apply the refining process on this set for a readjustment. So, it is the very high condition of the end refining process.

### *2) Low couplet subset with other subsets*

This is a subset with a data replication and low coupling between all other subsets generated by the algorithm. The threshold value of the low coupling is defined during the configuration of quality of service (QoS) parameters of the classification algorithm. In this case also, we can ignore the cost of exchanges between tow subsets. Therefore, we do not apply the refining process on this set for a readjustment. This is the condition acceptably low for the end of the refining process.

### *3) Highly (or strong) couplet subset with other subsets*

This is a subset with a data replication and highly coupling between all other subsets generated by the algorithm. In this case, the cost of exchanges between the two subsets may influence the quality of the algorithm. Therefore, we apply the refining process on this set for a readjustment. This is the condition for the continuation of the refining process. In this case, we proceed to the creation of another subset of sources such that the new subset will allow us to minimize the exchange of data between sources during a query process.

## *B. Subsets readjustment algorithm*

The basic idea of the subsets readjustment proposed in this paper is to either move the sources of low coupling with other sources of the same group to groups of highly coupling, or to create a new groups.

The transfer or change of sources is based on the criterion of belonging. The criterion for membership of a source to a group depends on the threshold value proposed by the administrator system as a parameter of quality of service as we will define in the next section. For a description of this algorithm, we propose the following data model:

- Threshold (G) : the minimum threshold for the validation of belonging a source to a group G. It is defined as follows:

$$\text{Threshold (G)} = (\text{Max Coupling}(\text{Si}, \text{Sj}) \times \text{Min Distance (Si,Sj)})$$

- We assume that the source S belongs to the group G. We define the belonging degree of S to G and we denoted by DA(S,G), by the proportional ratio of the sum of the similarities of the source S with other sources of the same group and the sums of the distances from the source S to the sources of the other groups.

$$\text{DA}(\text{S}, \text{G}) = \sum \text{Coupling}(\text{S}, \text{Si}) \times \frac{\sum \text{Similitude}(\text{S}, \text{S}_\text{i}))}{\sum \text{Distance}(\text{S}, \text{S}_\text{j}))}$$

With: $S_i \in G$ and $S_j \notin G$.

- Therefore, we define LowCoupling (G) by all sources of low coupling of the group G. This is the set of sources with the value of the degree of belonging validation is less than the threshold of G.

$$\text{LowCoupling (G)} = \{\text{Si} / \text{DA (Si, G)} < \text{threshold(G)}\}$$

So during the adjustment process or the refining of each group G, we begin with the generation of all the sources of low coupling for each not empty group G (LowCoupling (G)) and for each source $S_i$ of this set, we proceed to the following steps:

- If all the belonging degrees of the source to the other groups are less than the belonging validation thresholds, we assign this source Si to a new group.
- If not, the source Si is added to the group of a maximal validation belonging degree.

**Algorithm :**

*1) Let S = {S1, S2, ..., Sn} a set of sources*

*2) Let G={G1,G2, ……, Gm}a set of groups generated by the distribution algorithm.*

*3) For each element of untreated group $G_i$, we proceed to the following iterations:*

*a. Mark up the group $G_i$ and calculate LowCoupling set ($G_i$).*

*b. If the set LowCoupling($G_i$) is not empty then:*

*c. For each sources $S_i$ in LowCoupling ($G_i$), do:*

*1) Calculate the DA (Si, Gj), for all other groups Gj such that i # j, and store them in a indexed table by the groups $G_{i\,:}$ TabDegre[$G_i$] in descending order.*

*2) Traverse the table TabDegre from the first element until the verification of the condition:*

*3) TabDegre[$G_k$]> threshol ($S_i$,$G_k$)*

*4) If any group $G_k$ from the table TabDegre don't validate this condition, we will create a new marked group Gn that contains the source $S_i$.*

*5) If not, we add Si in the group $G_k$.*

## V. Study and Evaluation Of Network Overload

Generally, the aim of using the data warehouse is to ensure access to data in a distributed environment and minimizing network overhead. In this section, we study the performance of data transmission on the network during the interrogation of the mediator with the presence of local data warehouses generated by the algorithm that we proposed in the previous section. We will use analytical modeling and statistical analysis of simulation results. In particular, we examine the statistics of the packets transmission on the network, and we propose a comparative study on the distribution of network load among the proposed solutions. We then establish the relationship between the data warehouse system efficiency of data replication, which could be used to adjust dynamically the degree of replication depending on the bandwidth of the network, optimizing the tradeoff between storage and data accessibility. To do this, we consider the following parameters:

$\mathrm{Taille\_Reponse}(\mathrm{S_i})$: The average size of a response to a request asking the source $\mathrm{S_i}$.

NbrSources : Total number of sources.

Taille_Rep_Moy(Ei): the average size of a response to a request asking the data warehouse Ei.

Nbr_Sources($E_i$): The number of sources comprising data warehouse $E_i$.

Nbr_Entrepot : The number of data warehouses.

Nbr_Requete : Number of queries .

$P(S_i)$: The probability to have a new response from the source $S_i$.

Let R a user query and $\{\mathrm{R_1}, \mathrm{R_2}, \ldots, \mathrm{R_n}\}$ a set of subqueries after rewriting by mediator and n sources $\{S_1, \ldots\ldots., S_n\}$.

*A. Without using classification methods*

In this case we assume that the sources are integrated by the mediator are independent, and for each source $\mathrm{S_i}$, the mediator generates a subquery $\mathrm{R_i}$.

$$\mathrm{TailleReponse}(R_1) = \mathit{TailleRepMoy}(S_1) \times \mathrm{P(S_1)}$$

$$\mathrm{TailleReponse}(R_2) = \mathrm{TailleRepMoy(S_1)} \times (\mathrm{P(S_2)} - \mathrm{P(S_1)})$$

$$\mathrm{TailleReponse}(R_3) = \mathit{TailleRepMoy}(S_3) \times \big(P(S_3) - P(S_1 \cap S_2\,)\big)$$

$$T\mathrm{ailleReponse}(\mathrm{R_n}) = \mathrm{TailleRepMoy}(\mathrm{S_n}) \times \left(\mathrm{P}(\mathrm{S_n}) - \mathrm{P}\left(\bigcap_{\mathrm{i=1}}^{\mathrm{n-1}} \mathrm{S_i}\right)\right)$$

$$T\!a\mathrm{illeReponse}(\mathrm{R_n}) = \mathrm{TailleRepMoy}(\mathrm{S_n}) \times \left(\mathrm{P}(\mathrm{S_n}) - \prod_{\mathrm{i=1}}^{\mathrm{n-1}} \mathrm{P}(\mathrm{S_i})\right)$$

$$T\mathrm{aille}(\mathrm{R}) = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{TailleReponse}(\mathrm{R_i})$$

$$T\mathrm{aille}(\mathrm{R}) = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{TailleRepMoy}(\mathrm{S_i}) \times \left(\mathrm{P}(\mathrm{S_i}) - \prod_{\mathrm{k=1}}^{\mathrm{i-1}} \mathrm{P}(\mathrm{S_k})\right)$$

For reasons of simplicity, we assume that the probability $\mathrm{P(S_i)}$ and average response size TailleRepMoy $(\mathrm{S_i})$ identical for all sources$\mathrm{S_i}$.

We represent this parameters respectively by P, and TM then :

$$\mathrm{Taille}(R) = \sum_{i=1}^{n} TM \times \left(P - \prod_{k=1}^{i-1} P\right)$$

$$\mathrm{Taille}(R) = TM\ \times \sum_{i=1}^{n-1} \left(P - P^{i-1}\right)$$

*B. With using classification methods*

In this case we consider another data duplication factor $\mathrm{D(E_i)}$ in a data warehouse $\mathrm{E_i}$. This factor represents the probability of data duplication in the responses of sources. Therefore:

$$P(E_i) = D(E_i) \times \prod_{j=1}^{Nbr(E_j)} P(S_j)$$

With, $\mathrm{Nbr(E_i)}$ is the number of sources of data warehouse $\mathrm{E_i}$ . Also, we suppose K data warehouse generated after applying one of the classification algorithms. The overall size of the result after executing a query R is:

$$TailleReponse(R_1) = TailleRepMoy(E_1) \times P(E_1)$$

$$TailleReponse(R_2) = TailleRepMoy(E_1) \times (P(E_2) - P(E_1))$$

$$TailleReponse(R_3) = TailleRepMoy(E_3) \times \big(P(E_3) - P(E_1 \cap E_2\,)\big)$$

$$TailleReponse(R_k) = TailleRepMoy(E_k) \times \left(P(S_k) - P\left(\bigcap_{i=1}^{k-1} S_i\right)\right)$$

$$TailleReponse(R_k) = TailleRepMoy(E_k) \times \left(P(S_k) - \prod_{i=1}^{k-1} P(E_i)\right)$$

$$Taille(R) = \sum_{p=1}^{k} TailleReponse\big(R_p\big)$$

$$= \sum_{p=1}^{k} \mathrm{TailleRepMoy}\big(E_p\big) \times \left(P\big(E_p\big) - \prod_{i=1}^{p-1} P(E_i)\right)$$

$$= \sum_{p=1}^{k} \text{TailleRepMoy}(E_p) \times \left( \left( D(E_p) \times \prod_{i=1}^{Nbr(E_p)} P(S_i) \right) - \prod_{i=1}^{p-1} \left( D(E_i) \times \prod_{j=1}^{Nbr(E_i)} P(S_j) \right) \right)$$

(F1)

For the sake of simplicity, we assume that the probability $\mathrm{P(S_i)}$, the replication factor $\mathrm{D(E_i)}$ and the size of the average response $\mathrm{TailleRepMoy\ (S_i\,)}$ are regular for all sources. We represent these parameters respectively by P, D, and T then:

$$Taille(R) = \sum_{p=1}^{k} TMRE \times \left( \left( D \times \prod_{i=1}^{Nbr(E_p)} P \right) - \prod_{i=1}^{P-1} \left( D \times \prod_{j=1}^{Nbr(E_i)} P \right) \right)$$

$$Taille(R) = TMRE \times \sum_{p=1}^{k} \left( \left( D \times P^{Nbr(E_p)} \right) - \left( D^{p-1} \times P^{Nbr(E_i) \times (p-1)} \right) \right)$$

(F2)

*1) Analysis:*

According to the two formulas (F1) and (F2) the size of response to a query, it can be concluded that the size of data exchange on the network with the use of classification methods, following a series of interrogation of the mediator, is lower than without the use of classification methods in different situations. But the degree of difference depends on the factor of duplication, the average size of the query result, the number of remote sources and number of data warehouses generated by classifiers.

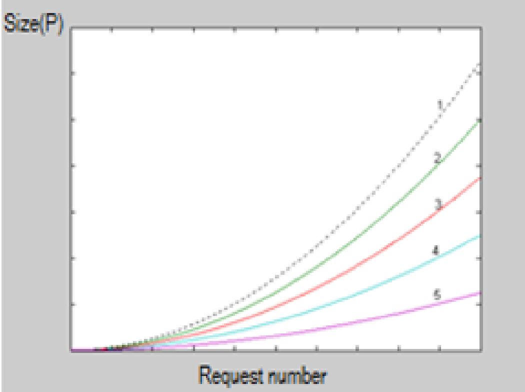

Fig .2. Influence of the probability P

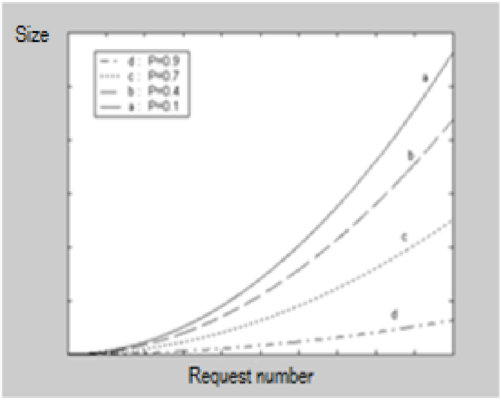

Fig .3. Influence of the duplication factor D

Generally, in the case of use of classification methods it can be seen that the rate of communication and exchange of data decreases to a certain level of interrogation. Therefore, the cost estimate takes into account other additives parameters influence on the basic parameters studied previously.

For example, we assumed that the probabilities, P and D are constant for the any new responses from a remote source regardless of the number K of warehouse generated by the classification algorithms. But these probabilities depend heavily on this number K and the number of queries N.

The average size of the response decreases for each new query. This degradation depends mainly to the identical records of sources in a warehouse$\mathrm{E_i}$.

Indeed, in the first experiment, we fixe two parameters: the duplication factor D and the average number of warehouses K, and we changed the number of queries N. The results are shown in the following figure.

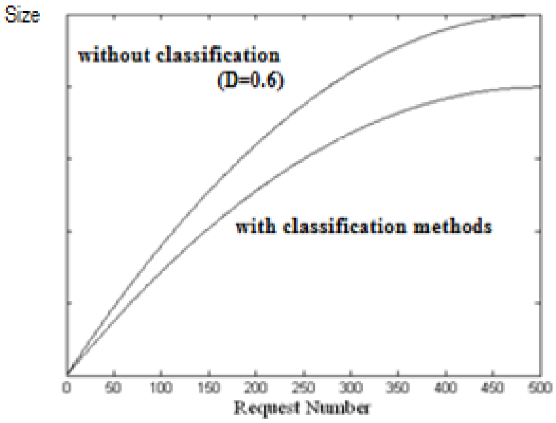

Fig .4. Comparison between methods (D=0.6)

According to this graph, the size of the exchanges on the network without the use of classification methods is always higher than that with the use of classification methods. The difference becomes important with the increase of the number of questions. This means that duplication of data stored in the warehouse (as the duplication factor D) influence on the exchange rate.

In the second experiment we can observe the effect of the variation of the duplication factor D on the exchange rate. According to figure 5, we note that if the duplication factor D decreases, the number of warehouses increases, therefore the exchange rate also increases. For D = 0, this meant that there is no classification groups. This shows that the classification methods guarantee a better system performance.

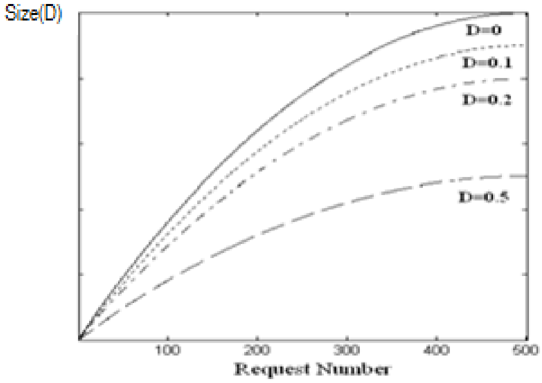

Fig .5. Influence of the duplication factor D on the size.

## VI. Conclusion

In this paper, we have presented a new approach for query optimization using dynamic programming technique for set integrated heterogeneous sources. To do this, we have developed a relevant algorithm which grouping the sources into subsets based on our new classification method that we proposed in this paper. In fact, we have shown with the study of the performance of data transmission on the network during the interrogation of the mediator with the presence of local data warehouses generated by our proposed algorithm and the evaluation of network overload that our classification methods using datawarehousing offer many benefits: minimized the cost of data exchange during the execution of subqueries on geographically remote sites, increased flexibility, and simplified access to data with greater agility. Indeed, local data warehousing offers space and interrogation power for several sources centers, but also the possibility of analyzing the data more efficiently on any remote server based on availability and needs. This solution effectively enables more users to access to more and more data without difficulty. Thanks to the Distributed Databases solution, we can migrate critical data on a data centers and improve the response time of readjustment and equilibration of the distribution data localization. In the perspective, we will study the performance of the different solutions by a comparative study.

## Refrences

[1] K. Asghari, A S. Mamaghani and M R. Meybodi, “An Evolutionary Approach for Query Optimization Problem in Database”, In Proc. of Int. Joint Conf. on Computers, Information and System Sciences, and Engineering (CISSE2007), England, springer, 2007.

[2] L. Cherrat, M. Ezziyyani and M. Essaaidi, “Automatic Generation of Query Order Execution Plan for Hybrid Mediator with Medical Sources”, In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), International Conference on Digital Object Identifier, Beijing, pp. 558 - 565, 10-12 Oct. 2011.

[3] A. Cali, D. Calvanese, G. Giacomo and M. Lenzerini, “ On the Expressive Power of Data Integration Systems”, In Proc. of the 21st Inter. Conf. on Conceptual Modeling, pp.338-350, 2003.

[4] M. Ezziyyani, M. Bennouna and L.Cherrat, “Mediator of the heterogeneous information systems based on application domains specification : AXMed Advanced XML Mediator”, IEEE Journal,, Vol.3, No.2, pp. 25-45, 2006.

[5] L M. Haas, “Beauty and the beast: The theory and practice of information integration”, In Database Theory ICDT'2007, pp.28-43, 2007.

[6] D. Calvanese, and G D. Giacomo, “Data Integration: A Logic-Based Perspective”, AI Magazine, Vol.26, No.1, 2005.

[7] I. Jellouli, M. El Mohajir and E. Zimanyi, “Classification conceptuelle et ontologie de domaine pour l’intégration sémantique des données”, La revue électronique des technologies d'information e-TI, No. 5, 5 novembre 2008.

[8] S. Kermanshahani, “Semi-materialized framework: a hybrid approach to data integration”, In Proc. of the 5th inter. conf. on Soft computing as transdisciplinary science and technology CSTST '08, pp. 600-606, New York, NY, USA, 2008.

[9] A. Halevy, A. Rajaraman. and J. Ordille, “Data integration: the teenage years”, Proceedings of the 32nd international conference on Very large data bases VLDB '06, pp. 9-16, Seoul, Korea, September 12-15, 2006.

[10] M S. Hacid, and C. Reynaud, “L'intégration de sources de données”, Revue Information - Interaction - Intelligence I3, Vol.4, No. 2, 2004.

[11] Z G. Ives, AY. Levy, D S. Weld, D. Florescu and M.. Friedman, “Adaptive Query Processing for Internet Applications”, In IEEE Computer Society Journal, Vol.23, No. 2, pp. 19-26, June 2000.

[12] Z G. Ives, “Efficient query processing for data integration”, Doctoral thesis at the University of Washington, 2002.

[13] H P. Kriegel, P. Kunath, M. Pfeifle and M. Renz, “Approximated Clustering of Distributed High-Dimensional Data”, In Proc. of the 9 th Pacific-Asia conference (PAKDD 2005), Hanoi, Vietnam, pp. 432-441, May 18-20, 2005.

[14] E Johnson. and H. Kargupta, “Hierarchical Clustering From Distributed, Heterogeneous Data”, In Computer Science Journal, pp. 221-244. Springer-Verlag, 1999.

[15] A K. Jain., M N. Murty and P J. Flynn, “Data Clustering: A Review. In ACM Computing Surveys”, Vol. 31, No. 3, pp. 265-323, Sep. 1999.

[16] N F. Samatova, G. Ostrouchovand, A. Geist and A V. Melechko, “RACHET: An Efficient Cover-Based Merging of Clustering

Hierarchies from Distributed Datasets", In Distributed and Parallel Databases, Vol. 11, No.2, pp. 157-180, Mars 2002.

[17] E. Januzaj, H P. Kriegel, and M. Pfeifle, "DBDC: Density Based Distributed Clustering", In Proc. 9th Int. Conf. on Extending Database Technology (EDBT 2004), pp. 88-105, Heraklion, Greece, 2004.

[18] E. Januzaj, H P. Kriegel and M. Pfeifle, "Scalable Density-Based Distributed Clustering", In Proc. 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Pisa, Italy, 2004.

[19] H P. Kriegel, S. Brecheisen, P. Kröger, M. Pfeifle and M. Schubert, "Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects", In Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'03), pp.587-598, San Diego, CA, 2003.

[20] M C. Tu, D. Shin and D.Shin, "A Comparative Study of Medical Data Classification Methods Based on Decision Tree and Bagging Algorithms", In Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing , pp.183-187, 2009.

[21] A. S. Kumar and S. Sahni, "Comparative Study of Classification Algorithms for Spam Email Data Analysis", In International Journal on Computer Science and Engineering (IJCSE), Vol. 3, No. 5, May 2011.

[22] P. Berkhin, "A Survey of Clustering Data Mining Techniques In Grouping Multidimensional Data", pp. 25-71, 2006.

[23] J P. Nakache and J. Confais , "Approche pragmatique de la classification", In livre editions TECHNIP, 2004.

[24] A. Lelu, "Evaluation de trois mésures de Similarité utilisées en Sciences de l'information"., In Information Sciences for Decision Making , Vol.6, pp.14-25, 2003.

[25] S H. Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions", International Journal Of Mathematical Models And Methods In Applied Sciences, Vol. 1, No. 42007.

[26] S H. Cha, "Taxonomy of Nominal Type Histogram Distance Measures", In American Conference On Applied Mathematics (Math '08), Harvard, Massachusetts, USA, March 24-26, 2008.