Dynamic Distributed Data Warehouse Design

Karima Tekaya, ISI ARIANA, Tunisia; E-mail: Karima.Tekaya@isi.rnu.tn

ABSTRACT

The fragmentation of a data warehouse into multiple data mart makes the task of the administrator very difficult. He must manage all operations of insertion, update and delete at the level of sources, Extraction, Transformation and Load (ETL) operations, logical Models, fragmentation and allocation. While nearly touching all levels of the architecture of a data warehouse, the administrator's task is going to become especially as difficult as the level of fragmentation is raised. Of this fact, it will be interesting to develop an auto-evolutionary system permitting to maintain a data warehouse up to date without interrupting its working and while keeping all the time one same level of performance. This article presents a dynamic distributed data warehouse design, proposes some basic concepts, develops a formulation of the problem and integrates an automatic system of administration based on intelligent agents.

Keywords: data warehouse, dynamic, fragmentation, ETL, Replication and design.

1. INTRODUCTION

A data warehouse is generally characterized by a very big volume of data; it is used, contrary to the transactional data bases, exclusively in consultation, all operations of up dating are taken in charge by the administrator. A data warehouse can be centralized or distributed. Many researches to date investigate building distributed data warehouses with particular emphasis placed on distribution design for data warehouse environment and the dynamic aspect is the current subject of work in the autonomic data management area.

The following section, presents in more details the problematic. The section 3 summarizes the state of art concerning data warehouse fragmentation techniques and dynamic data bases. The section 4, presents the contribution of this article and in the section 5, a dynamic distributed data warehouse design solution will be developed.

2. PROBLEMATIC

The problem of data warehouse fragmentation was carefully developed in [7], [8] and in [9]. But, the realized works always gave a static fragmentation that requires to be updated every time there is a change.

All operations of up dating are taken in charge by the administrator; the partitioning of data into multiple data mart is going to make very difficult his task. He must manage all operations of insertion, updating and deleting, at the level of sources, Extraction, Transformation and Load (ETL) operations, logical Models, fragments and allocation. While nearly touching all levels of the architecture of a data warehouse, the administrator's task is going to become especially as difficult as the level of fragmentation is raised.

3. STATE OF THE ART

Distributed Data Warehouse and Fragmentation Techniques

In [4] and in [5], authors proposed an architecture for distributed data warehouse. It is based on the ANSI/SPARC architecture that has three levels of schemas: internal, conceptual, and external. This work is based on TOP/DOWN approach and presents two fundamental issues: fragmentation and allocation of the fragment to various sites. Authors proposed a horizontal fragmentation algorithm for a fact table of a data warehouse. In [1], we have proposed a methodology for relational distributed data warehouse design. For this purpose, we develop a set of matrix: 'Matrix of data partitioning', 'matrix of data allocation' and a 'matrix of data source and in [2] we adapt the same methodology to the multidimensional environment. In [5], [4], [1] and in [2], a basic architecture of a distributed data warehouse has been proposed; we suggest the implementation of these works in a distributed environment.

Several works shows the importance of fragmentation in a data warehouse context, it represents today a more challenging stake that in a relational or objects database context. [6],[7],[8],[9],[10],[11],[12],[13]. In addition, several commercial products showed the utility of fragmentation in the process of queries optimization: In [1], we proposed a matrix of fragmentation, 'Horizontal Matrix of fragmentation ', making abstraction to the approach of modeling, this matrix permits from a logical table to generate a set of fragments and it has as input: queries and their frequencies of utilization. Thereafter, we proposed a matrix of allocation permitting to allocate every fragment to the most adequate site. One data can be a table or a fragment of table. In [2] the same solution was adapted in a dimensional environment; we have experiment the solution through an example. In [3] the same problematic was presented and some arguments are showed to prove the importance of fragmentation in distributed data warehouses environment. Otherwise, several works of research and the commercial products showed the utility of fragmentation techniques in the process of queries optimization [13]. Horizontal fragmentation in data warehouses is more challenging compared to that in relational and object databases. This challenge is due to the several choices of partitioning schemas that can be found in [8]

Dynamic Data Warehouse

We are not aware of any research work addressing the dynamic data warehouse. Some works developed the idea of a incremental design of a data warehouse [14], and other works are focused on dynamic operational data bases [15] , [16].

4. CONTRIBUTIONS

The contribution brought by this article consists in proposing a methodology for a dynamic distributed data warehouse design. This solution is essentially based on the extension of the classic solution of a centralized data warehouse. Some basic concepts are added, a formalism of presentation is developed and the integration of an automated administration system is done to maintain the data warehouse continuously up to date. The proposed solution can be adapted to centralized and so to distributed data warehouse.

5. DYNAMIC DISTRIBUTED DATA WAREHOUSE DESIGN (FIGURE 1)

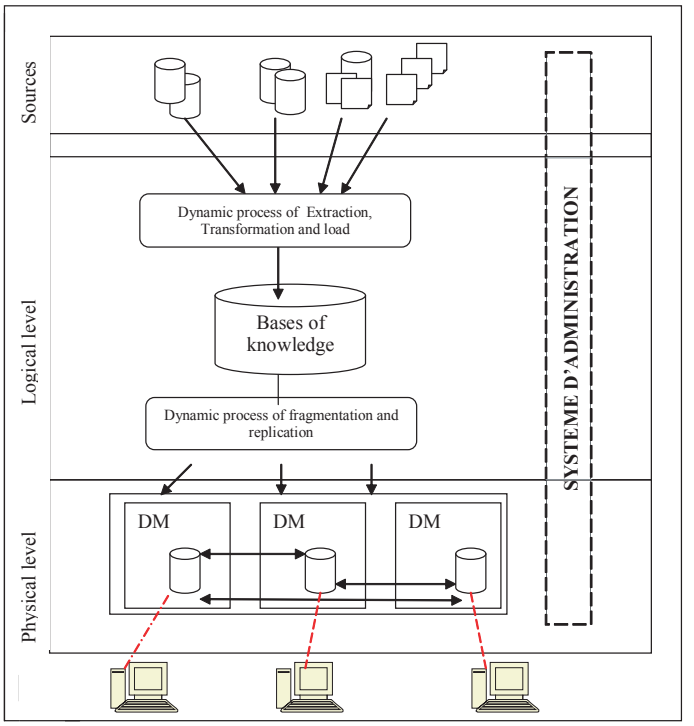
5.1 Basic Concepts

The proposed solution is an extension of work achieved in [2], this work is essentially based on the integration of a process of fragmentation and replication into the design level.

We propose the following basic concepts:

- *Dynamic process of extraction, transformation and load*: We add the term dynamic because this system is activated each time there is a change in data sources, function of transformation and load operations.
- *Bases of knowledge*: describes each data in the data warehouse. A data is characterized by a source, function of transformations and the allocation.

Figure 1. Architecture of a dynamic distributed data warehouse



- *Dynamic process of fragmentation and replication*: added to generate the derived horizontal fragmentation to the different tables and to found the best allocation for each data or fragment. We add the term dynamic because this system is activated each time there is a change in query frequency or structure.

- *System of administration*: integrate all functionalities to maintain a data warehouse up to date. The system of administration is touching all levels of the architecture of a distributed data warehouse (Figure 1). It's role is
  - o (1) to detect needs of refreshment of each data in every data mart,
  - o (2) to detect changes in function of transformation in every data mart,
  - o (3) to detect needs of updating in the bases of knowledge
  - o (4) to detect changes in query structure or frequency

5.2 Formalism

We are going to formalize the different concepts proposed through computational multi-dimensional matrixes.

a) The Matrix of Data Sources Integration (Table 1)

The d*ynamic process of extraction, transformation and load* can be simplified with the Matrix of Data Sources Integration MDSI (Table1), for each attribute A_{jL} ((1<=j=k), (1<=L=m); k the number of dimension or fact table (Ti) and m the number of attribute in Ti from dimensional model. We put for each data (1) *the source* (from which Logical Model LMr (1<=r<=s; s the number of source used for the integration process. (2) for each data A_{rz}((1<=r<=s, 1<=z<=x ; x the number of attribute for each LMr) the transformation function applied into data source to make an attribute A_{jL} adapted to the specification of the integrated data warehouse.

A transformation can be elementary or composite.

An elementary transformation is gotten while calculating the function $f_{tel}(\mathbf{A}_{rz})$, it gives as result an attribute $\mathbf{a}_{jL}$ that will be integrated in the data warehouse.

A transformation can be composite (CT), that means, calculated according to several attributes sources. The result will be gotten while applying the function $f_{tel}(\mathbf{A}_{r1},...,\mathbf{A}_{rx})$. This matrix has been implemented in [2].

b) Matrix of Primary and Derived Horizontal Fragmentation (Table2)

The dynamic process of fragmentation can be simplified by the Matrix of Primary and Derived Horizontal Fragmentation (MPDHF). The MPDHF consists to the definition of the uses of data by the treatments t_{ipi} (1<=i<=n, 1<=pi<=qi ; n numbers it of site, q numbers it of treatment by site. A treatment t_{ipi} can use one or several tables DTj dimension DTj (1<=j<=k; k number of dimension table from the star

Table 1. The matrix of data sources integration

Dimension tables	Attributs	Data source												CT
		LM1				**LM r**				**LM s**				
		$\mathbf{A}_{1l}$	$\mathbf{A}_{1z}$	$\mathbf{A}_{1x}$		$\mathbf{A}_{rl}$	$\mathbf{A}_{rz}$	$\mathbf{A}_{rx}$		$\mathbf{A}_{sl}$	$\mathbf{A}_{sz}$	$\mathbf{A}_{sx}$		
$\mathbf{DT}_1$	$\mathbf{a}_{1l}$	$f_{tel}(A_{1l})$												
	…													
	$\mathbf{a}_{1L}$	X		X		X				X				$f_{tc(\,)}$
	…		$f_{tel}(A_{1z})$											
	$\mathbf{a}_{1m}$													
$\mathbf{DT}_j$	$\mathbf{a}_{j1}$			$f_{tel}(A_{1x})$										
	…													
	$\mathbf{a}_{jL}$						$f_{tel}(A_{rz})$							
	…													
	$\mathbf{a}_{jm}$											$f_{tel}(A_{sx})$		
$\mathbf{DT}_k$	$\mathbf{a}_{k1}$	X	X			X	X			X	X			$f_{tc(\,)}$
	…							$f_{tel}(A_{rx})$						
	$\mathbf{a}_{kl}$													
	…		X	X			X	X			x	X		$f_{tc(\,)}$
	$\mathbf{a}_{km}$													

Table 2. Matrix of primary and derived horizontal fragmentation

		Treatments	DIMENSION TABLES		
			$DT_1(a_{1,1}, \ldots a_{1,L1}, \ldots a_{1,m1})$	$DT_j(a_{j,1}, \ldots a_{jLj}, \ldots a_{j,mj})$	$DT_k(a_{k,1}, \ldots, a_{k,Lk}, \ldots a_{k,mk})$
SITES	S_1	$t_{1,1}$ ... $t_{1,p1}$ .... $t_{1,q1}$	$Fhp_{11}=f_p(DT_1, \{a_{1,l1,..,} a_{1,Ll}\},CF)$ $Fhd_{11}=f_d(FT, Fhp_{11,} CJ)$	…	…
	…				
	S_i	$t_{i,1}$ ... $t_{i,pi}$ ... $t_{i,qi}$	$Fhp_{1i}=f_p(DT_1, \{a_{1,l1,..,} a_{1,Ll}\},CF)$ $Fhd_{11}=f_d(FT, Fhp_{11,} CJ)$	$Fhp_{ji}=f_p(DT_j, \{a_{j,1,..,} a_{j,Lj}\},CF)$ $Fhd_{ji}=f_d(FT, Fhp_{ji,} CJ)$	…
	…				
	S_n	$t_{n,1}$ ... $t_{n,pn}$ ... $t_{n,qn}$	…	…	$Fhp_{nk}=f_p(DT_k, \{a_{k,1,..,} a_{k,Lk}\},CF)$ $Fhd_{nk}=f_d(FT, Fhp_{nk,} CJ)$

model) and/or one or several attributes a_{jLj} (1<=L<=m; m number of attribute of a dimension table DTj. For example, a fragment horizontal primary Fhp_{11} is generated following the use of one or several attributes $a_{j,Lj}$ of the dimension table DT1 and by the treatment $t_{1,1}$.

The list of the primary horizontal fragments results from the application of the function f_p that takes in entry the dimension table to fragment DT_j, the list of the attributes concerned by fragmentation $\{a_{1,l1,..,} a_{1,Ll}\}$ and the criteria of fragmentation CF. It returns as result a fragment of the dimension table DT_j.

The list of the derived fragments results from the application of the function f_d that takes in entry the concerned FT, the primary horizontal fragment Fhp and join condition. It return's as result a derived fragment from the fact table.

c) Allocation Matrix (Table3)

The primary and derivative fragmentation matrix generates a set of data used by the different DM. Let Du be a generated data by the MPDHF (1<=u<=t ; t the number of data generated by the fragmentation process) Du can be a fact table, a dimension table a derived fragment or a primary fragment. If Du is used by Si then it will be automatically allocated to Si. And it will be called Persistent Data (PD). The persistence of one data in a DM depends on its use frequency (Fu). If Fu is equal to 0 it will be suppressed from the site and pass in the absent data (AD) state. The function f_a permits to determine for a data D_u to an instant t and according to the frequency of use Fu if it is PD or AD in a data mart DMi.

d) Matrix of knowledge bases (Table 4)

It gives for every allocated data: the source, transformations and allocation information. A transformation can be elementary $\boldsymbol{f_{tel}(A_{rz})}$ using only one source attribute $\mathbf{A_{rz}}$ or composite $\boldsymbol{f_{tcf}(A_{1z,} A_{1x,} A_{r1,} A_{s1})}$ while using several source attributes.

6. SYSTEM OF ADMINISTRATION (FIGURE 2)

(1) Agent of Refreshment

A data can be a table or a fragment of a table. This one is in evolution; changes can touch its structure (change of source, suppression of one or several attributes, additions of news data on the system…etc.) or the content. All operations of refreshment, updating or suppression on data are activated by the ADR. The ADR is activated by changes witch can occur in data source.

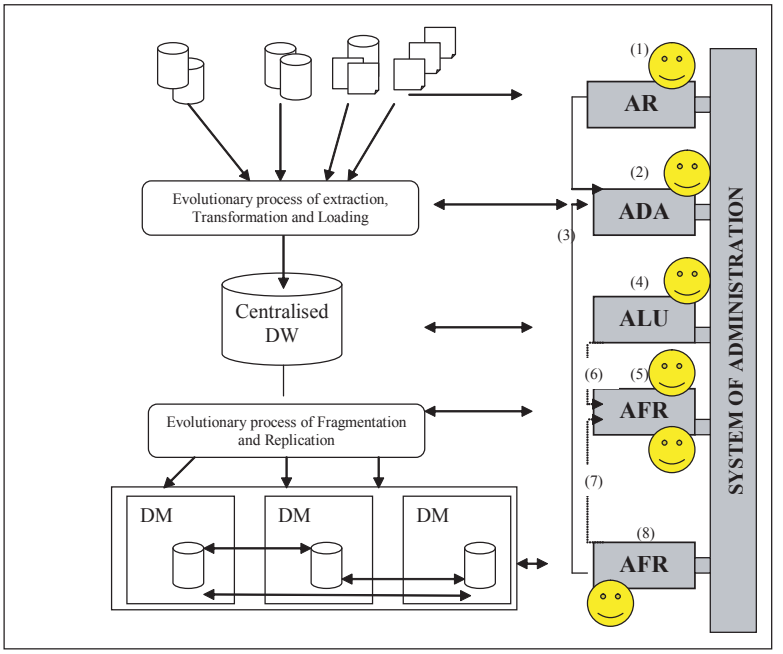
Table 3. Allocation matrix

		FACT TABLES / DIMENSION TABLES / PRIMARY HORIZONTAL FRAGMENTS /DERIVED HORIZONTAL FRAGMENT		
		D_1	D_u	D_t
DESTINATIONS	DM_1	$f_a(D_{1,} DM_1,t,F_u)$=PD/AD	$f_a(D_{1,} DM_1,t,F_u)$=PD/AD	$f_a(D_{1,} DM_1,t,F_u)$=PD/AD
	...			
	DM_i	$f_a(D_{1,} DM_1,t,F_u)$=PD/AD	$f_a(D_{1,} DM_1,t,F_u)$=PD/AD	$f_a(D_{1,} DM1,t,Fu)$=PD/AD
	...			
	DM_n	$f_a(D_{1,} DM_1,t,F_u)$=PD/AD	$f_a(D_{1,} DM_1,t,F_u)$=PD/AD	$f_a(D_{1,} DM_1,t,F_u)$=PD/AD

Table 4 : Matrix of knowledge bases

Destinations \ Sources			LM_l				LM_r				LM_s			TC
			A_{l1}	A_{lz}	A_{lx}	A_{r1}	A_{rz}	A_{rx}	A_{s1}	A_{sz}	A_{sx}	
DM_l	LM_l	D_{l1}	$f_{te}(A_{l1})$									
		…										
		D_{lu}	X		X	X			X			$f_{tc}()$
		…		$f_{te}(A_{lz})$			$f_{te}(A_{rz})$					
		D_{lt}										
…											$f_{te}(A_{sx})$	
DM_i	LM_i	D_{i1}										
		…					$f_{te}(A_{rz})$					
		D_{iu}										
		…		$f_{te}(A_{lz})$							$f_{te}(A_{sx})$	
		D_{it}	X	X		X	X		X	X		$f_{tc}()$
…												
DM_n	LM_n	D_{n1}										
		…		X	X		X	X		X	X	$f_{tc}()$
		D_{nu}										
		…	$f_{te}(A_{l1})$									
		D_{nt}										

Figure 2. Integration of intelligent agent



(2) Agent of Data Adaptation: (ADA)
Data mart are placed on the different sites, they can have different DBMS, different OS and can have some different features. Data must be adapted to specifications of every site. Then the applied transformation operations on data coming from different sources can undergo changes that vary according to data Marts, and it can also evolve in the time what requires a Refreshment of the corresponding data base. Therefore, the system of administration must take account of all changes concerning the applied transformation operations. The new needs of transformation are detected by the Agent of Data Mart Administration (ADMA) and thereafter communicated to the ADA **(3)**.

(4) Agent of Logical Updating of the Global Data Warehouse Model (ALU):
Its role consists in bringing stakes to necessary updating to the global model of data warehouse if there is a detected change in (1) or in (2).

(5) Agent of Fragmentation and Replication of the Logical Data (AFR)
If we detect one or some modifications at the level of the global logical model of the warehouse **(6)**, it is necessary to regenerate the Evolutionary process of Fragmentation and Replication and to see the impact on the physical models of different data mart. The Criteria of fragmentation essentially are based on decisional queries, these evolve according to the informational needs of the enterprise, this evolution has two types: query frequency and query structure, if there is detection of change (7), The system must verify the impact of this one on the already made fragmentation and so necessary to regenerate the Evolutionary process of Fragmentation and Replication, It will have as consequences the modification of one or some physical models (or data mart).

(7) Agent of Data Mart Administration (ADMA)
The role of this agent is to detect changes in the operation of transformation done by the ADA. It communicates changes to ADA so that it regenerates the evolutionary Process of extraction, transformation and loading. It can also detect changes in decisional queries, these changes can touch their frequencies as well that their structures. It sends an order thereafter to the AFR so that it regenerates the process of fragmentation and replication of the logical data warehouse.

7. CONCLUSION

A data warehouse includes a set of information for the decisional system. It can be centralized or distributed. A data warehouse is dynamic; it can face several evolutions. All operations of refreshment are taken in account by the administrator. The administrator's task becomes difficult if the data warehouse is distributed. To face this problem, we proposed, a methodology of a distributed and dynamic data warehouse design, we proposed a set of basic concepts and a set of matrixes permitting the formalization of the dynamic ETL process, basis of the knowledge, dynamic process of fragmentation and the dynamic process of replication. As perspectives we propose the implementation of the solution.

8. REFERENCES

[1] Karima Tekaya, Abdellaziz Abdellatif, 'Modélisation de la répartition des données d'un data warehouse'. Eighth Maghrebian Conference on Software Engineering and Artifical Intelligence, (MCSEAI'04). 2004.

[2] Karima Tekaya, Abdellaziz Abdellatif, 'Modélisation de la répartition des données d'un data warehouse'. Journal International des Sciences de l'Information et de la Communication (ISDM05)

[3] Karima Tekaya, Abdellaziz Abdellatif 'Fragmentation and Replication process in data warehouse environment' The 3rd ACS/IEEE International Conference on Computer Systems and Application (AICCSA'05).

[4] A.Y. Noaman, and K. Barker, 'A Horizontal Fragmentation Algorithm for the fact relation in a Distributed Data Warehouse'. Proceeding of the Eigth International Conference on Information and Knowledge Management (CIKM'99). November 1999, Pages154-161.

[5] A.Y. Noaman and K. Barker. 'Distributed data warehouse architectures'. Journal of Data Warehousing, 2(2):37-50, April 1997.

[6] Ladjel Bellatreche, Kamalakar Karlapalem, Mukesh Mohenia, 'What can partitioning do for your data warehouses and data marts', 2000.Ladjel Bellatreche**, Kamalakar Karlapalem, Mukesh Mohania. 'Some issues in design of data warehousing systems'. Data warehousing and web engineering".** IRM Press Hershey, PA, United States. Pages 22 – 76. 2002.

[7] Ladjel Bellatreche and Kamel Boukhalfa, 'La fragmentation dans les entrepôts de données: une approche basée sur les algorithmes génétiques', Revue des Nouvelles Technologies de l'Information (EDA'2005), Juin, 2005, pp. 141-160.

[8] L. Bellattreche, M. Schneider, H. Lorinquer, et M. Mohenia. 'Bringing together partitioning materialized views and indexes to optimize performance of relational data warehouse'. Proceeding of the international conference Data Warehousing and Knowledge Discovery (DAWAK'2004), pages 15-25, September 2004.

[9] Ladjel Bellatreche, **Kamalakar Karlapalem**, **Mukesh Mohania**. **'Some issues in design of data warehousing systems'. Data warehousing and web engineering.** IRM Press Hershey, PA, United States. Pages 22 – 76. 2002.

[10] M. T. Özsu et P. Valduriez. 'Principles of distributed data base systems'. Prentice Hall. 1991.

[11] S. Ceri et G. Pelagatti, 'Distributed databases principles and systems'. McGraw-Hill, 1984.

[12] Sanjay, V. R. Narasayya, et B. Yang. 'Integrating vertical and horizontal partitioning into automated physical database design'. Proceedings of the ACM SIGMOD International Conference on Management of data, pages 359-370, Juin 2004.

[13] Dimitri Theodoratos, Timos Sellis, 'Dynamic Data Warehouse Design'. Data Warehousing and Knowledge Discovery . Springer-Verlag, LNCS, 1999.

[14] Mariano Consens, Denilson Barbosa, Adrian M. Teisanu, Laurent Mignet, '**Goals and benchmarks for autonomic configuration recommenders', International Conference on Management of Data. Proceedings of the 2005 ACM SIGMOD international conference on Management of data,** Baltimore, Maryland. 2005.

[15] Stratos Papadomanolakis and Anastassia Ailamaki. Proceedings 'Automating Schema Design for Large Scientific Databases Using Data Partitioning**',** The 16th International Conference on Scientific and Statistical Database Management (SSDBM), Santorini, Greece, June 2004