Effectively and Efficiently Designing and Querying Parallel Relational Data Warehouses on Heterogeneous Database Clusters: The *F&A* Approach

Ladjel Bellatreche, LIAS/ENSMA, Poitiers University, Futuroscope Chasseneuil Cedex, France

Alfredo Cuzzocrea, ICAR-CNR, ItalyF&AUniversity of Calabria, Renede, Italy

Soumia Benkrid, National High School for Computer Science (ESI), Algiers, Algeria

ABSTRACT

In this paper, a comprehensive methodology for designing and querying Parallel Rational Data Warehouses (PRDW) over database clusters, called Fragmentation & Allocation (F&A) is proposed. F&A assumes that cluster nodes are heterogeneous in processing power and storage capacity, contrary to traditional design approaches that assume that cluster nodes are instead homogeneous, and fragmentation and allocation phases are performed in a simultaneous manner. In classical approaches, two different cost models are used to perform fragmentation and allocation, separately, whereas F&A makes use of one cost model that considers fragmentation and allocation parameters simultaneously. Therefore, according to the F&A methodology proposed, the allocation phase/decision is done at fragmentation. At the fragmentation phase, F&A uses two well-known algorithms, namely Hill Climbing (HC) and Genetic Algorithm (GA), which the authors adapt to the main PRDW design problem over heterogeneous database clusters, as these algorithms are capable of taking into account the heterogeneous characteristics of the reference application scenario. At the allocation phase, F&A introduces an innovative matrix-based formalism capable of capturing the interactions among fragments, input queries, and cluster node characteristics, driving the data allocation task accordingly, and a related affinity-based algorithm, called F&A-ALLOC. Finally, their proposal is experimentally assessed and validated against the widely-known data warehouse benchmark APB-1 release II.

Keywords: *Database Clusters,FragmentationF&AAllocation (F&A), Genetic Algorithm (GA), Hill Climbing (HC), Matrix-Based Formalism, Processing Power, Relational Data Warehouses (PRDW), Storage Capacity*

DOI: 10.4018/jdm.2012100102

INTRODUCTION

In this paper, we focus the attention to the context of query optimization techniques over *relational Data Warehouses* (RDW) developed on top of *cluster environments* (Lima *et al.*, 2009). A RDW is usually modeled by means of a *star schema* consisting of a huge *fact table* and a number of *dimension tables*, similarly to what shown in Figure 1 as related to the widely-known data warehouse benchmark *APB-1 release II* (OLAP Council, 2010). Here, the fact table *Sales* is joint to the following four dimension tables: *Product*, *Customer*, *Time*, *Channel*. *Star queries* are typically executed against RDW. Star queries retrieve aggregate information (e.g., based on standard SQL aggregate operators like SUM, COUNT etc) from *measures* stored in the fact table by applying *selection conditions* on joint dimension table columns, and they are extensively used as conceptual basis for more complex *OLAP queries*, which, in turn, are exploited to extract useful summarized knowledge from RDW for decision making purposes.

Unfortunately, evaluating OLAP queries over RDW typically demands for a high-performance that is difficult to ensure over large amounts of multidimensional data, even because such queries are usually complex in nature (Bellatreche*F&A*Boukhalfa, 2005). This complexity is mainly due to the presence of joins and aggregation operations over huge fact tables, which very often involve billions of tuples to be accessed and processed. In order to speed-up OLAP queries over RDW, several optimization approaches, mainly inherited from classical database technology, have been proposed in literature. Among others, we recall *materialized views* (Gupta, 1999), *indexing* (Sarawagi, 1997), *data partitioning* (Bellatreche *et al.*, 2009), *data compression* (Cuzzocrea*F&A*Serafino, 2009) etc. Despite this, it has been demonstrated that the sole use of these approaches singularly is not sufficient to gain efficiency during the evaluation of OLAP queries over RDW (Stöhr *et al.*, 2000). As a consequence, in order to overcome limitations deriving from these techniques, high-performance in database technology, including RDW (Furtado, 2004; DeWitt *et al.*, n.d.), has traditionally been achieved by means of *parallel processing methodologies* (Özsu*F&A*Valduriez, 1999).

The main motivation of using parallel processing technologies in data warehouses relies not only in the need for performance improvement, but also in the fact that parallel

Figure 1. Logical schema of the data warehouse benchmark APB-1 release II