The M-OLAP Cube Selection Problem: A Hyper-polymorphic Algorithm Approach

Jorge Loureiro[1] and Orlando Belo[2]

[1] Departamento de Informática, Escola Superior de Tecnologia e Gestão,
Instituto Politécnico de Viseu, Campus Politécnico de Repeses, 3505-510 Viseu, Portugal
jloureiro@di.estv.ipv.pt

[2] Departamento de Informática, Escola de Engenharia, Universidade do Minho
Campus de Gualtar, 4710-057 Braga, Portugal
obelo@di.uminho.pt

Abstract. OLAP systems depend heavily on the materialization of multidimensional structures to speed-up queries, whose appropriate selection constitutes the cube selection problem. However, the recently proposed distribution of OLAP structures emerges to answer new globalization's requirements, capturing the known advantages of distributed databases. But this hardens the search for solutions, especially due to the inherent heterogeneity, imposing an extra characteristic of the algorithm that must be used: adaptability. Here the emerging concept known as hyper-heuristic can be a solution. In fact, having an algorithm where several (meta-)heuristics may be selected under the control of a heuristic has an intrinsic adaptive behavior. This paper presents a hyper-heuristic polymorphic algorithm used to solve the extended cube selection and allocation problem generated in M-OLAP architectures.

Keywords: Hyper-Heuristics; Distributed Data Cube Selection; Multi-Node OLAP Systems Optimization.

1 Introduction

Today, we know that the success of *On-Line Analytical Processing* (OLAP) systems depends largely on their multidimensional visions' mechanisms and on their support for fast query answering, independently of the aggregation level of the required information. Yet, they also carry some seeds of failure. In fact, answering an OLAP query may imply the scanning and aggregation of huge amounts of data, something that is often incompatible with the inherent on-line characteristic. The materialization of multidimensional structures, denoted as materialized views or subcubes, was devised as the answer to such problem, and has been a condition of performance. However, the increasing needs of OLAP users forced these structures to attain an inordinate size and complexity, implying new approaches to their optimization, beyond the classical cube selection solutions, as greedy heuristics [4], genetic approaches [8], or hill climbing with simulated annealing [6]. So, we think that distribution is the key.

C. Fyfe et al. (Eds.): IDEAL 2010, LNCS 6283, pp. 194–201, 2010.

Accompanying the trends of organizations' infrastructures, and given the principle of locality, the distribution of the materialized structures will increase the probability of local satisfaction of OLAP needs, having several advantages: a sustained growth of processing capacity without an exponential increase of costs, an increased availability of the system, and the avoidance of bottlenecks. Here, we focus in only one distribution type, something that we denoted as a *Multi-node OLAP* (M-OLAP) architecture, generated by the distribution of the OLAP cube by several storage and processing nodes, named *OLAP Server Nodes* (OSN), with a known processing power and storage space, inhabiting in close or remote sites, interconnected by communication facilities, available to the (possibly spreaded) users' community. Conventional cube selection problem deals only with the appropriate selection of the materialized structures, attending to a given query profile. Now, a new factor is included into the optimizing equation: space. In fact, it is not enough to select the most beneficial subcubes; they also have to be conveniently located.

The optimization of the distributed OLAP approach needs an extended cost model, which was proposed in [1], as the distributed aggregation lattice. This work also proposed a distributed node set greedy algorithm that addressed the distributed view selection problem, being shown that this algorithm has a superior performance than the corresponding standard greedy algorithm, using a benefit per unit space metric. In [10], this distributed lattice framework is used, but extended, to include real communication cost parameters and processing node power, which led to heterogeneity in the nodes and the network. This model was used as a framework to build a simulated OLAP environment, which disposes a set of estimation cost algorithms [10] that used the intrinsic parallel nature of the distributed OLAP architecture and time as the cost unit. Framed on this simulated environment, several metaheuristics have been applied to the selection and allocation of cubes in M-OLAP systems: genetic co-evolutionary [11] and *Discrete Particle Swarm Optimization* (Di-PSO) [9]. It was also used a simulated annealing metaheuristic, but using a more comprehensive cost model framing the simulated OLAP environment. This work pursued the research that has been conducted, but trying another approach based on two evidences: 1) it is known that each optimizing heuristic (or meta-heuristic) has virtues (and some limitations); one side of its approach was the design of a huge amount of hybrids, which try to collect the better of two worlds: compensate the weaknesses of a given algorithm with the strengths of another; in this approach we have two (meta-) heuristics which are applied in a pre-defined sequence; 2) but the concept may be extended by using a bundle of (meta-) heuristics, an emerging concept known as hyper-heuristic, building an algorithm where several (meta-) heuristics may be selected under the control of a heuristic. In this paper, we'll try to solve a general optimizing problem, as we have a complex and heterogeneous environment, the M-OLAP architecture.

2 Hyper-heuristic Approach

According to Burke et al. [2], hyper-heuristics are defined as a procedure of "using (meta-)heuristics to choose (meta-)heuristics to solve the problem in hand". At first sight, this emerging approach operates on macro-level. But, when we are facing very