Journal of Computational and Applied Mathematics 259 (2014) 905–913

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Minimizing roundtrip response time in distributed databases with vertical fragmentation

Rodolfo A. Pazos [a,*], Graciela Vázquez [b], José A. Martínez [a], Joaquín Pérez-Ortega [c], Gilberto Martínez-Luna [d]

[a] Instituto Tecnológico de Ciudad Madero, Av. 1o. de Mayo s/No., Col. Los Mangos, Cd. Madero, Tamaulipas, Mexico

[b] ESIME, Instituto Politécnico Nacional, Av. Inst. Politécnico Nal. s/No., Col. Lindavista Zacatenco, México, Mexico City, Mexico

[c] Centro Nacional de Investigación y Desarrollo Tecnológico, Interior Internado Palmira s/No., Col. Palmira, Cuernavaca, Morelos, Mexico

[d] Centro de Investigación en Computación, IPN, Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, Zacatenco, Mexico City, Mexico

ARTICLE INFO

Article history:
Received 15 February 2013
Received in revised form 5 September 2013

Keywords:
Distributed database design
Vertical fragmentation

ABSTRACT

The main purpose of this paper is to show the advantage of using a model proposed by us, which minimizes roundtrip response time versus traditional models that minimize query transmission and processing costs for the design of a distributed database with vertical fragmentation. To this end, an experiment was conducted to compare the roundtrip response time of the optimal solution obtained using our model versus the roundtrip response time of the optimal solution obtained using a traditional model. The experimental results show that for most cases the optimal solution from a traditional model yields a response time which is larger than the response time of the optimal solution obtained from our model, and sometimes it can be thrice as large.

1. Introduction

One of the challenges of the application of distributed database (DDB) systems is the possibility of expanding their utilization through the use of the Internet, so widespread nowadays. One of the most difficult problems in DDB systems deployment is the distribution design.

Traditionally, the DDB design problem has been defined as finding relation (tables) fragments and their allocation such that the overall costs incurred by query transmission and processing are minimized. It has been recognized for many years the importance of considering response time in the DDB modeling [1,2]. Unfortunately, traditional optimization models have not considered response time, as shown in Table 1, which summarizes the most relevant and recent works on fragmentation and the allocation of DDBs with vertical fragmentation.

The second, third and fourth columns of Table 1 show that some works have addressed only the fragmentation problem, many works have only dealt with the fragment allocation problem, and some works have addressed integrally both problems. The fifth column of Table 1 shows that all the previous works have considered transmission, access or processing costs. The seventh and eighth columns of Table 1 show that most works have proposed heuristic algorithms for addressing DDB fragmentation and allocation and only a few have proposed mathematical programming formulations.

In this paper a mathematical programming model is presented (VFA-RT), which describes the behavior of a DDB with vertical fragmentation and permits us to optimize its design taking into account the nonlinear nature of roundtrip response time (query transmission delay, query processing delay, and response transmission delay). In a previous paper by us [1]

* Corresponding author. Tel.: +52 8331676671.
E-mail address: r_pazos_r@yahoo.com.mx (R.A. Pazos).

0377-0427/$ – see front matter
http://dx.doi.org/10.1016/j.cam.2013.09.057

Table 1
Related works on fragmentation and allocation for DDBs with vertical fragmentation.

Works	Problems addressed			Value to minimize		Solution approach	
	Fragmentation	Allocation	Integrated fragmentation + allocation	Transmission or processing costs	Roundtrip response time	Heuristic algorithm	Mathematical programming formulation
Chakravarthy, 94 [7]	✓			✓		✓	
Pérez, 00 [3][a]			✓	✓			✓
Ma, 06 [4]			✓	✓		✓	
Tambulea, 08 [8]		✓		✓		✓	
Karimi, 09 [9]		✓		✓			[b]
Khan, 10 [10]		[c]		✓		[d]	
Sevinc, 10 [11]		✓		✓		✓	
Kamali, 11 [12]		✓		✓		✓	
Goli, 12 [13]	✓			✓		✓	
Song, 13 [14]			[c]	✓	[e]		[b]
Our approach			✓		✓		✓

[a] A previous model of ours.
[b] Problem constraints are enforced in the algorithm.
[c] Relation replication is considered.
[d] Axiomatic game theoretic mechanism.
[e] Times are assumed linearly proportional to transmission and processing costs.

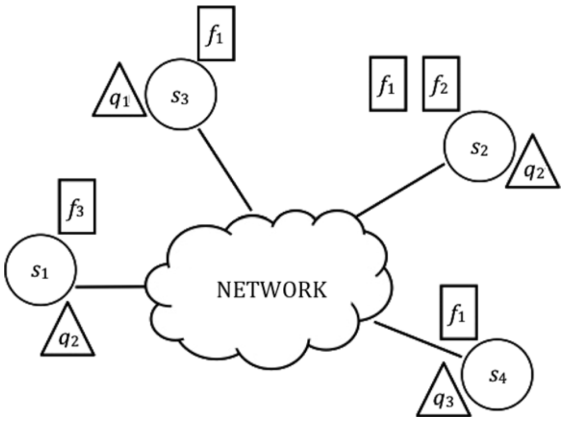

Fig. 1. Example of a distribution design.

we introduced for the first time the VFA-RT model, and in [2] we presented the experimental results of two metaheuristic algorithms (threshold accepting and tabu search) aimed at finding a good metaheuristic algorithm for solving the VFA-RT model. Unlike those papers, now we are interested in experimentally showing the advantage of using the proposed model with respect to traditional models which consider query transmission and processing costs; therefore, this paper also includes a brief description of the DFAR model [3], which considers this kind of costs. Finally, the paper includes a comparative experiment of the VFA-RT and DFAR models (using an exact algorithm for each) and a comparative experiment of an exact algorithm applied to the VFA-RT model versus a heuristic approach [4].

2. Description of the DDB design problem

A distributed database (DDB) is a database (DB) whose data are partitioned into portions that are stored in two or more different nodes, which are interconnected by a communication network. Fig. 1 shows a simple example of a DDB, in which the communication network is represented by a cloud and each network site (node) is represented by a circle tagged with a letter s and a sub-index.

We are interested in a type of data partition called vertical fragmentation, which is defined as follows [5] let R denote a relation (DB table) that has a set of attributes (table columns) $A = \{a_1, a_2, \ldots, a_L\}$; the vertical fragmentation of R consists of a partition of R into several sub-relations $R_1, R_2, \ldots, R_N$ such that each sub-relation is obtained by the following operation:

$$R_i = \Pi_{A_i} R$$

where Π represents the projection operator from the relational algebra, and $A_i \subset A$; additionally, $R_1, R_2, \ldots, R_N$ should be defined so that the original relation R can be reconstructed from its fragments by applying the join operation of the relational algebra to its fragments: $R = R_1$ JOIN R_2 JOIN …JOIN R_N.