Applied Computing and Informatics (2015) **xxx**, xxx–xxx

Saudi Computer Society, King Saud University

Applied Computing and Informatics

(http://computer.org.sa)
www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

Hierarchical simultaneous vertical fragmentation and allocation using modified Bond Energy Algorithm in distributed databases

Hossein Rahimi, Fereshteh-Azadi Parand *, Davoud Riahi

Math and Computer Science Department, Allameh Tabataba'i University, Ahmad Ghasir St., Beheshti Av., Tehran, Iran

Received 31 December 2014; revised 2 March 2015; accepted 3 March 2015

KEYWORDS

Bond Energy Algorithm;
Distributed database system;
Data allocation and fragmentation;
Clustering

Abstract Designing an efficient Distributed Database System (DDBS) is considered as one of the most challenging problems because of multiple interdependent factors which are affecting its performance. Allocation and fragmentation are two processes which their efficiency and correctness influence the performance of DDBS. Therefore, efficient data fragmentation and allocation of fragments across the network sites are considered as an important research area in distributed database design. This paper presents an approach which simultaneously fragments data vertically and allocates the fragments to appropriate sites across the network. Bond Energy Algorithm (BEA) is applied with a better affinity measure that improves the generated clusters of attributes. The algorithm simultaneously generates clusters of attributes, calculates the cost of allocating each cluster to each site and allocates each cluster to the most appropriate site.

* Corresponding author. Tel.: +98 2188713160.
E-mail addresses: s.rahimi@atu.ac.ir (H. Rahimi), parand@atu.ac.ir (F.-A. Parand), d.riahi@atu.ac.ir (D. Riah).

☆ Peer review under responsibility of King Saud University.

ELSEVIER | **Production and hosting by Elsevier**

http://dx.doi.org/10.1016/j.aci.2015.03.001

Results show more efficient clustering and allocation which gives better performance.

1. Introduction

Distributed databases reduce cost and increase performance and availability, but the design of Distribute Database Management Systems (DDBMS) is complicated. To make this process feasible it is divided into two steps: Fragmentation and Allocation. Fragmentation tries to split data into fragments, which should be allocated to sites over the network in the allocation stage. The process of fragmentation falls into two categories: Vertical Fragmentation and Horizontal Fragmentation. Vertical Fragmentation (VF) is partitioning relation R into disjoint sets of smaller relations while Horizontal Fragmentation (HF) is partitioning relation R into disjoint tuples. The allocation problem involves finding the optimal distribution of fragmentation to set F on site set S. There are four data allocation strategies applicable in a distributed relational database: centralized, fragmentation (partition), full replication, and partial replication (selective) [10]. When data is allocated, it might either be replicated or maintained as a single copy. So, fragment allocation can be either non-redundant or redundant. Under a non-redundant allocation scheme, exactly one copy of each fragment will exist across all the sites, while under redundant allocation schema, more than one copy of each fragment will exist across all the sites [12]. In this work, we combine fragmentation with partial replication of some clusters of attributes.

Allocation and fragmentation are interdependent and efficient data fragment allocation requires considering allocation constraints in the process of fragmentation, but in the most previous works these two steps are separated.

There are two general approaches toward solving the partitioning problem. One is to find the efficient solution by considering some of the constraints. In Hoffer [13] the storage capacity and retrieval cost constraints are the role factors. Each of these factors is weighted based on their amount of effect. The objective was to minimize the value of overall cost. The weights are calculated using linear programming approach so that the sum of the weights is equal to 1.

$$min(c_1 * storage\ cost + c_2 * retrieval\ cost) \tag{1}$$

Another good example of first set of approaches is proposed in Schkolnick [21]. The method tries to cluster records within an Information Management System (IMS) type hierarchical structure. The generated hierarchical tree is linear in the number of nodes. Heuristic grouping is used by the method presented in Hammer and Niamir [3]. It starts by assigning attributes to different positions. All potential types of grouping are considered and the one which represents the