Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Information Systems 30 (2005) 47–70

www.elsevier.com/locate/infosys

Iterative-improvement-based declustering heuristics for multi-disk databases☆, ☆☆

Mehmet Koyutürk[a,1], Cevdet Aykanat[b,*]

[a] *Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, USA*
[b] *Computer Engineering Department, Bilkent University, Ankara 06800, Turkey*

Received 10 July 2002; received in revised form 21 May 2003; accepted 29 August 2003

Abstract

Data declustering is an important issue for reducing query response times in multi-disk database systems. In this paper, we propose a declustering method that utilizes the available information on query distribution, data distribution, data-item sizes, and disk capacity constraints. The proposed method exploits the natural correspondence between a data set with a given query distribution and a hypergraph. We define an objective function that exactly represents the aggregate parallel query-response time for the declustering problem and adapt the iterative-improvement-based heuristics successfully used in hypergraph partitioning to this objective function. We propose a two-phase algorithm that first obtains an initial K-way declustering by recursively bipartitioning the data set, then applies multi-way refinement on this declustering. We provide effective gain models and efficient implementation schemes for both phases. The experimental results on a wide range of realistic data sets show that the proposed method provides a significant performance improvement compared with the state-of-the-art declustering strategy based on similarity-graph partitioning.

Keywords: Parallel database systems; Declustering; Hypergraph partitioning; Iterative improvement; Weighted similarity graph; Max-cut graph partitioning

1. Introduction

Minimizing query-response times is a crucial issue in designing high-performance database systems for application domains such as scientific, spatial and multimedia. These systems are often used interactively and amounts of data to be retrieved for individual queries are quite large. In such database systems, the I/O bottleneck is overcome through parallel I/O across multiple disks. Disks are accessed in parallel while processing a query, so response time for a query can be

☆Recommended by K.A. Ross.
☆☆This work was partially supported by The Scientific and Technical Research Council of Turkey under grant 199E013.

*Corresponding author. Department of Computer Engineering, Bilkent University, Ankara 06533, Turkey. Tel.: +90-312-290-1625 fax: +90-312-266-4047.

E-mail addresses: koyuturk@cs.purdue.edu (M. Koyutürk), aykanat@cs.bilkent.edu.tr (C. Aykanat).

[1]The author was an M.Sc. student at the Computer Engineering Department of Bilkent University during this work.

0306-4379/$ - see front matter
doi:10.1016/j.is.2003.08.003

minimized by balancing the amount of data to be retrieved on each disk. Therefore, data is distributed across multiple disks, respecting disk capacity constraints, in such a way that data items that are more likely to be retrieved together are located into separate disks. This operation is known as declustering.

There have been considerable amounts of research on developing strategies to effectively decluster data on several disks in order to achieve minimum I/O cost. Many declustering strategies were developed on declustering multi-dimensional data structures such as cartesian product files, grid files, quad trees and R-trees [1–8], multimedia databases [9–13], parallel web servers [14], signature files [15], spatial databases and geographic information systems (GIS) [16,17].

In the literature, there exists a vast amount of work on mapping-function-based declustering techniques such as Coordinate Modulo Declustering (CMD) [5], Field-wise Exclusive-OR Distribution [18], Hilbert Curve Method [3,19], Lattice Allocation Method [2], and Cyclic Allocation Scheme [8]. Commonly these methods scatter the data into disks in such a way that the neighboring data items in multi-dimensional space are placed into different disks. The applications of these methods are restricted to spatial databases and multi-attribute data sets. Furthermore, if there exists information about query distribution and data sizes, these methods do not exploit such available information.

Recently, Shekhar and Liu [16] proposed a novel declustering technique which can exploit information about query distribution and handle heterogeneous data-item sizes, non-uniform data distributions, and constraints on disk sizes. They model the declustering problem as max-cut partitioning of a weighted similarity graph (WSG). The nodes of WSG correspond to data items and weights associated with edges represent similarity between respective data-item pairs. Here, the similarity between a pair of data items refers to the likelihood that the pair will be accessed together by queries. Hence, maximizing the edge cut in a partitioning of WSG relates to maximizing the chance of assigning similar data items to separate disks. This model was reported [16] to outperform all mapping-function-based strategies in experiments with grid files. In this work, we show that the objective function of max-cut graph partitioning does not accurately represent the cost function of declustering. This flaw is because of the fact that WSG is an indirect model and it represents each query defining a single multi-way relation by multiple pairwise relations.

In this work, we propose a direct model for solving the declustering problem by exploiting the correspondence between a data set with a given query distribution and a hypergraph. Each data item and query in the database system correspond, respectively, to a vertex and a hyperedge (net) of the hypergraph. The hypergraph partitioning (HP) problem has been widely encountered in VLSI layout design [20,21] and partitioning irregular computational domains for parallel computing [22,23]. We define an objective function that exactly represents the total query response time for the declustering problem and adapt the iterative-improvement-based HP algorithms to this objective function. We propose a two phase algorithm that first obtains an initial K-way declustering by recursively bipartitioning the data set, then applies multi-way refinement on this declustering. We provide effective gain models and efficient implementation schemes for both phases. Experimental results on a wide range of realistic data sets show that the proposed model provides significantly better declusterings than the WSG model, which is the most promising strategy in the literature.

We define the declustering problem and introduce the notation in Section 2. In Section 3, we introduce the state-of-the-art WSG model and discuss the flaws of this model. We introduce our model and the adaptation of iterative-improvement techniques to the problem in Section 4. In Section 5, we report the experimental results and evaluate the performance of the proposed method.

2. Basic definitions on declustering

Declustering problem can be defined in various ways depending on the application. Shekhar and Liu [16] define the problem in a database