Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Distribution Design for Complex Value Databases

Hui Ma

2007

A dissertation presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Information Systems at Massey University

Abstract

Distribution design for databases usually addresses the problems of fragmentation, allocation and replication. However, the main purposes of distribution are to improve performance and to increase system reliability. The former aspect is particularly relevant in cases where the desire to distribute data originates from the distributed nature of an organization with many data needs only arising locally, i.e., some data are retrieved and processed at only one or at most very few locations. Therefore, query optimization should be treated as an intrinsic part of distribution design. Due to the interdependencies between fragmentation, allocation and distributed query optimization it is not efficient to study each of the problems in isolation to get overall optimal distribution design. However, the combined problem of fragmentation, allocation and distributed query optimization is NP-hard, and thus requires heuristics to generate efficient solutions.

In this thesis the foundations of fragmentation and allocation in databases on query processing are investigated using a query cost model. The considered databases are defined on complex value data models, which capture complex value, object-oriented and XML-based databases. The emphasis on complex value databases enables a large variety of schema fragmentation, while at the same time it imposes restrictions on the way schemata can be fragmented. It is shown that the allocation of locations to the nodes of an optimized query tree is only marginally affected by the allocation of fragments. This implies that optimization of query processing and optimization of fragment allocation are largely orthogonal to each other, leading to several scenarios for fragment allocation. Therefore, it is reasonable to assume that optimized queries are given with subqueries having selection and projection operations applied to leaves. With this assumption some heuristic procedures can be developed to find an "optimal" fragmentation and allocation. In particular, cost-based algorithms for primary horizontal and derived horizontal fragmentation, vertical fragmentation are presented.

Acknowledgements

I would like to thank Professor Klaus-Dieter Schewe, my supervisor, for his attentive guidance, endless patience, invaluable advice and constant support during this research. Especially I am thankful for the opportunity offered by him to step into an academic career. It is he who showed me how to do research step by step. It is also he who always encouraged me to make me more and more confident and enthusiastic about doing research.

I am also thankful to my co-supervisor, Professor Sven Hartmann, for his support, patience, encouragement and guidance during this research. Especially, my deep appreciation owns to Professor Sven Hartmann for his always being available for discussing and answering questions, not only during week days, working hours but also during weekends, holidays, late nights, through the whole period of my study.

In the mean time, I would like to express my appreciation of the understanding and help from Markus Kirchberg and Sebastian Link, who have helped me during my study in their own ways.

Finally, I am grateful to my husband, my parents, and my son, for their support, patience and *love*. Without them this work would never have come into existence (literally).

Table of Contents

1 Introduction **1**
- 1.1 Distributed Databases: Definition and Motivation . . . 1
- 1.2 Database Distribution Design . . . 3
  - 1.2.1 Design Techniques: Fragmentation and Allocation . . . 3
  - 1.2.2 Alternative Design Strategies . . . 4
    - Top-down Approach. . . . 4
    - Bottom-up Approach. . . . 5
  - 1.2.3 The Objective of the Design of Data Distribution . . . 5
- 1.3 Distribution Design Dilemma . . . 6
  - 1.3.1 The Complexity of the Problems . . . 6
  - 1.3.2 Interdependencies with Query Optimization . . . 7
  - 1.3.3 Ad Hoc Solutions . . . 7
- 1.4 Contributions . . . 8
- 1.5 The Outline of the Thesis . . . 9

2 Literature Review on Distribution Design for Databases **11**
- 2.1 Horizontal Fragmentation . . . 11
  - 2.1.1 Primary Horizontal Fragmentation for Relational Databases . . . 11
  - 2.1.2 Primary Horizontal Fragmentation for Object Oriented Databases . . . 14
  - 2.1.3 Derived Horizontal Fragmentation for Relational Databases . . . 15
  - 2.1.4 Derived Horizontal Fragmentation for Object Oriented Databases . . . 16
- 2.2 Vertical Fragmentation . . . 18
  - 2.2.1 Vertical Fragmentation for Relational Databases . . . 18
    - Affinity-Based Approach . . . 18
    - Cost-Driven Approach . . . 20
  - 2.2.2 Vertical Fragmentation for Object Oriented Databases . . . 21
- 2.3 Mixed Fragmentation . . . 22
- 2.4 Allocation . . . 23
  - 2.4.1 Simple Query Environment . . . 23
  - 2.4.2 Query Site Strategy . . . 24
  - 2.4.3 Others . . . 26
- 2.5 Summary . . . 26

3 Complex Value Databases and Query Language **29**
- 3.1 The Data Model . . . 29
  - 3.1.1 Types and Trees . . . 29

3.1.2 Subtypes . . . 33
3.1.3 Rational Trees . . . 34
3.2 Query Algebra and Optimization . . . 36
3.2.1 A Generic Query Algebra . . . 37
3.2.2 Path Expressions . . . 38
3.2.3 A Simple Query Algebra . . . 38
3.3 A Query Processing Cost Model . . . 41
3.3.1 Query-Trees . . . 41
3.3.2 Size Estimation for Leaves of a Query Tree . . . 41
3.3.3 Size Calculation for Intermediate Nodes of a Query Tree . . . 42
3.3.4 Query Processing Costs . . . 43

4 Fragmentation Operations **45**
4.1 Horizontal Fragmentation . . . 45
4.2 Vertical Fragmentation . . . 47
4.3 Splitting . . . 49
4.4 Correctness Rules for Fragmentation . . . 50

5 Foundations of Fragmentation and Allocation **53**
5.1 The Impact of Splitting on Query Costs . . . 53
5.1.1 Scenario I . . . 54
5.1.2 Scenario II . . . 54
5.1.3 Scenario III . . . 55
5.2 The Impact of Horizontal Fragmentation on Query Costs . . . 56
5.2.1 Scenario I . . . 56
5.2.2 Scenario II . . . 57
5.2.3 Scenario III . . . 58
5.3 The Impact of Vertical Fragmentation on Query Costs . . . 59
5.3.1 Scenario I . . . 59
5.3.2 Scenario II . . . 60
5.3.3 Scenario III . . . 61
5.4 Summary . . . 61

6 Heuristics for Horizontal Fragmentation and Allocation **63**
6.1 Primary Horizontal Fragmentation and Allocation . . . 63
6.1.1 Fragmentation and Allocation Refinement for Horizontal Fragments . . . 69
6.1.2 An Example . . . 70
Fragmentation Step 1 . . . 72
Fragmentation Step 2 . . . 75
6.1.3 Simple Selection Predicates for Horizontal Fragmentation . . . 76
6.1.4 Experimental Evaluation . . . 78
6.2 Derived Horizontal Fragmentation . . . 80
6.2.1 A Motivating Example . . . 80
6.2.2 Some Terms . . . 83
6.2.3 Heuristics for Derived Horizontal Fragmentation . . . 85
6.2.4 An Example . . . 86
6.2.5 Discussion . . . 87

6.2.6 Experimental Evaluation 89

7 Heuristics for Vertical Fragmentation and Fragment Allocation 91
7.1 A Motivating Example 91
7.2 Some Terms 94
7.3 A Heuristic Approach for Vertical Fragmentation 96
7.4 Examples 98
7.5 Discussion 102

8 Conclusions 105
8.1 Summary 105
8.2 Open Problems and Challenges 106

Bibliography 109

List of Figures

3.1 HERM Diagram of the University Database 30
3.2 Example of a Query Tree 42

5.1 Scenario I for Query Tree Rewriting in Case of Splitting Fragmentation . . . 54
5.2 Scenario II for Query Tree Rewriting in case of Splitting Fragmentation . . . 55
5.3 Scenario III for Query Tree Rewriting in Case of Splitting Fragmentation . . 56
5.4 Scenario I for Query Tree Rewriting in Case of Horizontal Fragmentation . . 57
5.5 Scenario II for Query tree Rewriting in Case of Horizontal Fragmentation . . 58
5.6 Scenario III for Query Tree Rewriting in Case of Horizontal Fragmentation . 59
5.7 Scenario I for Query Tree Rewriting in Case of Vertical Fragmentation 60
5.8 Scenario II for Query Tree Rewriting in Case of Vertical Fragmentation . . . 61
5.9 Scenario III for Query Tree Rewriting in Case of Vertical Fragmentation . . . 62

6.1 Allocation without Fragmentation 71
6.2 Reallocation of Fragments: Step 1 72
6.3 Reallocation of Fragments: Step 2 74
6.4 Reallocation of Fragments: Step 3 76
6.5 Simple Predicate and Query Costs 79
6.6 Scenario I - Primary Fragmentation Only 81
6.7 Scenario II - Derived Fragmentation according to One Fragmentation Schema 81
6.8 Scenario III - Derived Fragmentation according to Two Fragmentation Schemata 82

7.1 Scenario I: Fragmentation with Two Queries at One Location 92
7.2 Scenario II: Fragmentation with Two Queries at Different Locations 92
7.3 Scenario III: Fragmentation with Two Queries at Different Locations 93
7.4 Scenario IV: Fragmentation with Two Queries at Different Locations 93
7.5 Allocation of Fragments for Example 7.3 99

List of Tables

3.1 An Instance of the University Schema 32
3.2 An Instance of the University Schema (continued from Table 3.1) 33

4.1 Horizontal Fragmentation of *db*(DEPARTMENT) 46
4.2 Horizontal Fragmentation of *db*(PAPER) 47
4.3 Derived Horizontal Fragmentation of *db*(LECTURE) 48
4.4 Vertical Fragmentation of *db*(LECTURE) 50

6.1 Atomic Fragment Request Matrix 67
6.2 Atomic Fragment Pay Matrix 67
6.3 Atomic Fragment Allocation 68
6.4 Horizontal Fragmentation of *db*(LECTURER) 68

7.1 Attribute Usage Frequency Matrix for Example 7.3 99
7.2 Attribute Request Matrix for Example 7.3 99
7.3 Attribute Pay Matrix for Example 7.3 99
7.4 Attribute Usage Frequency Matrix for Example 7.4 100
7.5 Attribute Request Matrix for Example 7.4 101
7.6 Transportation Cost Factors for Example 7.4 101
7.7 Attribute Pay Matrix for Example 7.4 101
7.8 Attribute Allocation for Example 7.4 102