

Contents

Foreword	xv
Preface	xvii
Chapter 1 VERSO: A RELATIONAL BACKEND DATABASE MACHINE	1
1.0. Abstract	1
1.1. Motivations and Objectives	1
1.2. The Relational Interface	3
1.2.1. Motivation	3
1.2.2. Informal Description	4
1.2.3. Formal Definition	4
1.3. Software Description	5
1.3.1. Query and Update Processing	5
1.3.2. Logical Relations	6
1.3.3. Formatted Relations	6
1.3.4. Physical Relations	8
1.3.5. Filtering and the Finite-State Automation	8
1.3.6. Dynamic Aspects	9
1.3.7. Concurrency Control in VERSO	10
1.4. Hardware Architecture	11
1.4.1. General description	11
1.4.2. The V_0 Processor	12
1.4.3. Filter design	13
1.4.4. The Controller System	15
1.5. Current State of Development	16

Chapter 2 SABRE: A RELATIONAL DATABASE SYSTEM FOR A MULTIMICROPROCESSOR MACHINE	19
2.0. Abstract 19	
2.1. Introduction 20	
2.2. Objectives 21	
2.3. Basic Concepts 21	
2.3.1. The Data Manipulation Protocol 21	
2.3.2. A Hierarchy of Views 23	
2.3.3. The Metabase 24	
2.3.4. Request Decomposition and Optimization 24	
2.3.5. Intrarequest Parallelism 24	
2.3.6. Associative Partitions 26	
2.3.7. The Reduced Cover Tree 26	
2.4. Functional and Operational Architectures of SABRE 27	
2.4.1. Basic Principles 27	
2.4.2. The Functional Architecture 27	
2.4.3. The Operational Architectures 29	
2.5. SABRE Version 2—A True Multiprocessor Database Computer 31	
2.5.1. The Operational Architecture 31	
2.5.2. A Hardware Overview 32	
2.6. The Conclusion 33	
Chapter 3 RDBM: A DEDICATED MULTIPROCESSOR SYSTEM FOR DATABASE MANAGEMENT	36
3.0. Abstract 36	
3.1. Introduction 37	
3.1.1. Objectives and Background 37	
3.1.2. The Search Computer (SURE) System 38	
3.2. An Overview of the RDBM Architecture 41	
3.2.1. Design Goals 41	
3.2.2. Data Structure and User Interface 42	
3.2.3. Overall Architecture 42	
3.2.4. Basic Processor Hardware 44	
3.3. Detailed Discussion of RDBM 44	
3.3.1. Synchronization of Processor Actions 44	
3.3.2. The Secondary Storage System 52	
3.3.3. Data Filtering 55	
3.3.4. Sorting in RDBM 60	
3.3.5. Interrecord Operations 65	
3.3.6. Data Input/Output 70	
3.3.7. Control of the RDBM Multiprocessor System 72	
3.4. The Current State of the Project 83	
3.4.1. System Components 83	
3.4.2. The State of Implementation 84	
3.4.3. Further Project Activities 85	
3.5. Acknowledgements 85	
3.6. References 85	

Chapter 4 THE ARCHITECTURE OF A RELATIONAL DATABASE COMPUTER KNOWN AS DBMAC

87

- 4.0. Abstract 87
- 4.1. Introduction 88
 - 4.1.1. A Domain-Based Internal Schema 88
 - 4.1.2. Database Machine Modularity 90
 - 4.1.3. Functional Characteristics of DBMAC 92
- 4.2. The Logical Architecture 93
 - 4.2.1. The Logical High System 95
 - 4.2.2. Performance Evaluation of the High System 97
 - 4.2.3. The Logical Low System 101
- 4.3. The Physical Architecture 102
 - 4.3.1. The Physical High System 103
 - 4.3.2. The Physical Low System 104
 - 4.3.3. The Physical Architecture: A Complete View 105
- 4.4. Conclusions 106
- 4.5. References 107

Chapter 5 AN ASSESSMENT OF THE QUERY-PROCESSING CAPABILITY OF DBMAC

109

- 5.0. Abstract 109
- 5.1. Introduction 109
- 5.2. The Workload Model 110
 - 5.2.1. Queries 110
 - 5.2.2. Data-Pool Primitives 112
- 5.3. The Simulation Model 113
 - 5.3.1. The Physical Architecture 113
 - 5.3.2. Data Allocation 115
 - 5.3.3. Primitive Execution 116
 - 5.3.4. Query-Execution Graphs 117
 - 5.3.5. Query Activation and Execution 118
- 5.4. System Parameters 119
- 5.5. Some Results 120
 - 5.5.1. Query Execution 121
 - 5.5.2. Data Integration 123
 - 5.5.3. Device Utilization 124
- 5.6. Further Experiments 125
- 5.7. Some Remarks 126
- 5.8. Conclusions 128
- References 128

Chapter 6 DESIGN CONSIDERATIONS FOR AN INFORMATION QUERY COMPUTER

130

- 6.0. Abstract 130
- 6.1. Introduction 130
 - 6.1.1. Database Machines Research at NEC 132
 - 6.1.2. The Information Query Computer (IQC) 134

6.2.	Logical and Physical Interfaces	136
6.2.1.	The Logical Interface	136
6.2.2.	The Physical Interface	140
6.2.3.	Host-Database Machine Interface Problems	141
6.3.	Architectural Considerations	142
6.3.1.	Considerations on Basic Database Operations	142
6.3.2.	Expected Database Machine Performance	148
6.3.3.	Storage Hierarchy	151
6.4.	RASIS Considerations	152
6.4.1.	Reliability, Availability and Serviceability	152
6.4.2.	Data Integrity	153
6.4.3.	Data Security	154
6.5.	The Implementation of an Information Query Computer	154
6.5.1.	A System Overview	154
6.5.2.	The IQC System Architecture	156
6.5.3.	Design Highlights	160
6.6.	Conclusions	164
Appendix 6.1. Execution-Time Evaluation of Index (B-Tree Index) Search Operation 167		
6.7.	Acknowledgement	165
References		165

Chapter 7 A DATA-STREAM DATABASE MACHINE WITH LARGE CAPACITY

168

7.0.	Abstract	168
7.1.	Introduction	169
7.2.	Architectural Considerations	170
7.2.1.	High Performance	171
7.2.2.	Large Capacity	173
7.2.3.	Data Types of the Attributes	173
7.2.4.	Database Processing with an Encoded Database	174
7.3	Abstract Architecture	175
7.3.1.	Encoder/Decoder	175
7.3.2.	Kernel Database Machine	179
7.4.	Basic Functional Components	182
7.4.1.	Binary Trie Engine	182
7.4.2.	Search Engine	183
7.4.3.	Sort Engine	185
7.4.4.	Associative Disk Controller and Disk Cache	187
7.5.	Subsystem Architecture	188
7.5.1.	Encoder/Decoder	188
7.5.2.	Tuple Selector	192
7.5.3.	Tuple Constructor	192
7.5.4.	Aggregate Operator	192
7.6.	A Multiplicity of Subsystems	196
7.6.1.	Sorting Distributor Network	196
7.6.2.	Sorter Bank	198

7.7. Concluding Remarks 200

References 201

**Chapter 8 DESIGN AND ANALYSIS OF JOIN OPERATIONS
OF DATA BASE MACHINES**

203

8.0. Abstract 203

8.1. Introduction to the Hardware Organization 204

 8.1.1. The Hardware Organization 204

 8.1.2. The Basic Join Operation Involving Four Memories 206

 8.1.3. Phases of the Basic Join Operation 206

8.2. An Algorithmic Analysis of the Basic Join Operation 207

 8.2.1. Phases of Source Input and A-Block Calculation 208

 8.2.2. Phases of Target Input and Target Propagation 208

 8.2.3. Phases of A-Block Calculation and Join 208

 8.2.4. Time Complexity of the Basic Join Operation 209

8.3. Design Analysis of the Basic Join Operation 210

 8.3.1. Analysis of the A-Memory Size 210

 8.3.2. Results on the A-Memory Size 211

 8.3.3. Analysis of the B-Memory Size 212

 8.3.4. Interpretation of the Results On Memory Size 212

 8.3.5. Results on the B-Memory Size 214

 8.3.6. B-Memory Size as Influenced by the A-Memory
Block-Access Time 219

 8.3.7. Preferred Characteristics 226

8.4. Design of an Optimal Chip for the Join Operation 226

 8.4.1. Designing an Optimal Chip for Today 227

 8.4.2. Designing an Optimal Chip for 1992 231

8.5. Inequality and m -Way Join Operations 234

 8.5.1. Requirements for and Phases of the Inequality Join 234

 8.5.2. Operations of the m -Way Join 236

8.6. Comparison With Other Methods 237

8.7. Conclusions 244

8.8. Acknowledgements 244

References 245

Appendix 8.1. Interarrival-Time and Service-Time Distributions
in the Single-Processor Case 246

Appendix 8.2. Interarrival-Time and Service-Time Distributions
in the Multiprocessor Case 251

Appendix 8.3. The Heavy-Traffic Approximation 254

Appendix 8.4. The Diffusion Approximation 255

**Chapter 9 THE ARCHITECTURE AND OPERATION OF A LARGE,
FULL-TEXT INFORMATION-RETRIEVAL SYSTEM**

256

9.0. Abstract 256

9.1. Introduction 257

 9.1.1. Text Information-Retrieval Systems 257

 9.1.2. Classes of Text-Retrieval Systems 260

9.1.3.	An Example Application	262
9.2.	A Hardware-Enhanced Information-Retrieval System	264
9.2.1.	Host-Processor Operations	264
9.2.2.	Backend Processors	265
9.3.	The Text Searcher	266
9.3.1.	The Partitioned Finite-State Automation	268
9.3.2.	A PFSA Implementation	268
9.3.3.	State-Table Partitioning	273
9.3.4.	Operation Characteristics	277
9.3.5.	VLSI Implementation of the PFSA	279
9.3.6.	The Query Resolver	284
9.4.	Index Structures and Processing	285
9.4.1.	Previous Indexing Systems	286
9.4.2.	The Page-Indexing Method	288
9.4.3.	Specialized Index Processors	293
9.5.	Requirements for the Example	297
9.6.	Summary	298
	References	299

**Chapter 10 THE IMPLEMENTATION OF A MULTIBACKEND
DATABASE SYSTEM (MDBS):
PART I—AN EXERCISE IN DATABASE SOFTWARE ENGINEERING** 300

10.0.	Abstract	300
10.1.	Introduction	301
20.1.1.	Why Do We Have an Exercise?	301
20.1.2.	What Is the Exercise?	301
10.1.3.	What Benefits Do We Expect?	301
10.1.4.	What Results Do We Have to Date?	301
10.1.5.	What Is the Database System Being Prototyped?	302
10.1.6.	The Organization of the Chapter	304
10.2.	Why Implement MDBS?	304
10.2.1.	Validation of Simulation and Analytic Results	304
10.2.2.	Toward a Methodology for Database Applications Classification	305
10.2.3.	Benchmarking the System Performance	306
10.3.	An Evolutionary Implementation Strategy—What and Why?	306
10.3.1.	Version I—A Very Simple System: Single Mini without Concurrency Control and with Simplified Directory Management	307
10.3.2.	Version II—A Simple System: Single Mini with Concurrency Control	308
10.3.3.	Version III—The First “Real” System: Multiple Minis with Concurrency Control	308
10.3.4.	Version IV—The Real System with “Good” Directory Management	308
10.3.5.	Version V—The Full System with all the Designed Features Included	308

10.4.	Team Organization and Monitoring the Development Effort	309
10.4.1.	A Three-Level Chief-Programmer-Team Organization without Librarians	309
10.4.2.	The Structured Walkthrough	310
10.4.3.	A Uniform Documentation Standard	312
10.5.	The Design, Coding, and Testing Stages of the System Versions	314
10.5.1.	A Top-Down Design Strategy and the Use of Data and Service Abstractions	314
10.5.2.	A Formal Systems Specification Language (SSL)	314
10.5.3.	The Use of Jackson Charts	317
10.5.4.	A Measure of Modularity	319
10.5.5.	A Practice of Structured Coding	321
10.5.6.	A “Black-Box” Testing Approach	321
10.6.	Conclusions	322
10.6.1.	Our Year-Long Experience in the Exercise of Database Software Engineering	322
10.6.2.	Our Specific Findings about the Exercise in Database Software Engineering	323
10.7.	Acknowledgement	325
	References	325

Chapter 11 THE IMPLEMENTATION OF A MULTI-BACKEND DATABASE SYSTEM (MDBS): PART II—THE DESIGN OF A PROTOTYPE MDBS	327
--	------------

11.0.	Abstract	327
11.1.	Logical Description of MDBS	328
11.1.1.	The Data Model and the Data Manipulation Language	329
11.1.2.	The Notion of Record Clusters	334
11.1.3.	The Entire Process of Request Execution	338
11.1.4.	An Overview of the MDBS Implementation	342
11.1.5.	The Organization of the Rest of the Chapter	347
11.2.	A Functional Description of MDBS	347
11.2.1.	Functions of the Controller	347
11.2.2.	Functions of Each Backend	349
11.2.3.	Request Execution in MDBS	351
11.2.4.	Process Structure of MDBS	358
11.3.	An Implementation of the Controller Functions	359
11.3.1.	Design and Implementation Goals for the Controller	359
11.3.2.	The Concept of “Traffic Unit”	360
11.3.3.	The Structure of the Controller	360
11.3.4.	The Process Structure of the Controller	363
11.4.	An Implementation of Backend Functions	364
11.4.1.	The Process of Primary-Memory-Based Directory Management	365

11.4.2. The Record-Processing Functions	373
11.4.3. Concurrency Control	378
11.5. Current Status of the Implementation	384
11.6. Acknowledgements	385
References	385

