Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery

by Gerhard Weikum and Gottfried Vossen

Reviewer: Marc H. Scholl, University of Konstanz (Germany), Marc.Scholl@uni-konstanz.de

Summary of the book

The book gives both a comprehensive overview and an in-depth presentation of the field of transactional data processing covering the latest findings of the research community as well as practical experiences with state-of-the art algorithms and efficient implementation techniques. It reflects the advances of the field, particularly over the last decade. In the formal foundations, the authors emphasize a novel approach that unifies the discussion of concurrency control and recovery aspects. This is a significant improvement over traditional text books that cover the latter more on the level of implemenation details rather than giving a solid formal background, integrated with the body of concurrency control theory. Also, their approach extends nicely from a standard page-oriented model to a multi-level, object model of transaction processing.

The book develops the necessary formal background and does not require a deep database background. The authors argue that the scope of transaction technology is wider than classical database systems, aiming at new fields, such as today's e-commerce or communication systems. From time to time, the authors dive deeper into formal arguments, but the text always allows to skip details. The reader is further guided through the material by pairs of "Goal and Overview"—"Lessons Learned" sections as well as illustrative examples, extraordinarily comprehensive bibliographic notes, and a suite of exercises of varying complexity in each chapter.

A walk through the contents

The two constituents of transactional technology, concurrency control and recovery, are presented in the two core Parts II and III of the book. Part I starts out with a broad discussion of background and motivation, ending in the presentation of the two computational models for transactional processing, the page and the object model, and the definitions of the basic notions.

Also, this part provides the necessary understanding of database system concepts for those readers not familiar with that material. Part IV is devoted to the coordination of distributed transactions and the last, Part V, gives an outlook on topics not covered in the book.

The notions and techniques of concurrency control in Part II are presented in eight chapters (3–10). Chapter 3 introduces potential problems, the notions of schedules and histories, defines the several correctness criteria in terms of final state, view, conflict, and commit serializability, and elaborates on the (inclusion) relationships between the corresponding classes of histories. Next, algorithms for guaranteeing correct execution of concurrent transactions are given in Chapter 4, among them are the well-known locking protocols (two-phase locking, 2PL) as well as non-locking and hybrid protocols. The improved potential for parallelism offered by multi-version concurrency control are analyzed in Chapter 5. While everything has been considered within the simple page model up to now, the authors turn to the object model in Chapter 6 discussing the notions of correctness and Chapter 7, presenting algorithms. The next two chapters, 8 and 9, investigate special opportunities for further optimization given in the relational database context and on search structures respectively. Well-known efficient implementation strategies are explained nicely within the more general framework by incorporating knowledge about the semantics of transactional operations. The presentation of concurrency control issues is concluded in Chapter 10 with implementation and pragmatic issues. The authors give example data structures and skeletons of realization algorithms, discuss several tuning options, and elaborate on the control of overload situations.

The second core part of transaction management, recovery, is covered in Part III of the book. The initial chapters, 11 and 12, set the scene by first introducing the notions, issues, and correctness criteria for transaction recovery, handling failures of single transaction programs (in Chapter 11), and then for system crashes (in Chapter 12). One of the most innovative aspects of the presentation chosen in this book is the fact that recovery aspects are integrated with concurrency control strategies by the introduction of so-called expanded schedules. An expanded schedule contains explicit undo-operations (as inverses of forward-processing operations) for aborted transactions. Using this approach, one can derive classes of syntactically characterized schedules, and investigate their relationships with serializability classes. One of the classes, prefix-reducible schedules, can be characterized in such a way that allows for a straightforward extension of standard concurrency control protocols. Algorithms for crash recovery are discussed separately in the following two

Chapters 13 for the page model and Chapter 14 for the object model. Chapter 15 focusses on index recovery, savepoints, parallel restarts, main memory data-servers, and data-sharing clusters. Media recovery based on log files and storage redundancy, and disaster recovery are the topic of Chapter 16, before application-level recovery is discussed in Chapter 17, concluding this part. The authors present recovery techniques based on queues for workflow and more general stateful applications.

Part IV of the book covers distributed transactions in homogeneous and heterogeneous contexts. Concurrency control along with deadlock detection is presented in Chapter 18 and recovery algorithms, among them 2-phase and 3-phase commit protocols, in Chapter 19. Finally, Chapter 20 of Part V wraps up by summarizing the current achievements for three classes of readers: ready-to-use solutions for developers, state-of-the-art techniques for advanced system builders, and new challenges for researchers. The authors' future perspectives for several application domains conclude the book.

Target audience

This book "emphasizes scientific fundamentals of long-term validity and value, and does not cover specific system products" (quote from the authors' mission statement). As such, this is clearly an advanced textbook oriented towards university students, a fact that also becomes obvious from the intensive use of formal models in the advanced material. Nonetheless, the book can also serve as a "refresher" for practitioners who have studied transaction management earlier and want to get an up-to-date overview of the latest achievements, both in terms of new concurrency control and recovery techniques, as well as in terms of a novel way of presenting those techniques and the interactions between them. The reader who is not familiar with basic database system technology or "database speak" can also get a good picture of what transaction technology can deliver to other fields within computer science.

The book itself, in its preface, gives advice on how to use the contents for university courses on the undergraduate and/or graduate levels. The authors offer suggestions for a 15-week, 4 hours-per-week course and also for a 10-week, 2 or 4 hours-per-week course, indicating what they feel mandatory and optional material.

Reviewer's appreciation

Jim Gray, in his foreword as a series editor for the book, expresses his expectation that "this book is likely to become the standard reference in our field for many years to come."—What could I add to this?

I have been lucky enough to be able to use a preliminary version of Part II on concurrency control for a 12-week, 2 hours-per-week advanced undergraduate course at the University of Konstanz, Germany. Even though preliminary and incomplete, the text served extremely well as a reference. Not being an expert on the advanced material myself, it was easy to develop class material (such as slides and assignments) from the text. And, after all, it was also quite easy to become a bit more of an expert! The rigorous formal models, the convincing rationale between the steps, and the perspectives on practical systems architectures made teaching based on this book real fun. As far as the reviewer can tell, students also appreciated the presentation very well. Some of the chapters from Part III have been used later by two students for preparing seminar presentations on recovery basics, they came up with very clean, easy-to-follow presentations.

If you stay on the informal side, that is, follow the text, the examples and the theorems, you get a brilliant and very accessible tour of transaction technology. Once you dive into the details, you enjoy a step-by-step development of the research achievements of the last thirty years in the field. In any case, the authors succeeded in developing a rigorous and consistent presentation of theoretical as well as practical aspects.

The book is heavy-weight (each US cent buys you approximately 0.21g, which sums up for a total of 1.7kg), but on the other hand you get an estimate of some 30k value of well-written characters for each US$ list price. As for the characters: the reviewer found the typesetting with standard LaTeX-fonts more readible than the somewhat heavy fonts used in the book. Some of the shading in the figures is rather dark and thus makes the text hard to read.