Reminiscences on Influential Papers

Kenneth A. Ross, editor

See `http://www.acm.org/sigmod/record/author.html` for submission guidelines.

Peter Boncz, CWI, `P.Boncz@cwi.nl`.

[Annita Wilschut, Jan Flokstra, Peter M.G. Apers. Parallel evaluation of multi-join queries. Proc. SIGMOD Conf. 1995, San Jose, CA, pages 115–126.]

In the PRISMA project, a 100-node parallel machine was constructed by Philips Electronics and Dutch academia, along with a parallel main memory database sytem (PRISMA/DB). When this paper was written, the project had just finished (along the with idea of database machines).

This paper abstracts from the particular features of PRISMA/DB, and evaluates and analyzes the performance trade-offs for a wide range of parallel query processing strategies. Its clear style of presentation, along with careful attention to previous work both in its discussion as well as in the experiments and analysis, make this paper into a concise introductory or "refereshment" text for researchers interested in parallel query execution.

Currently, parallel databases may no longer be in vogue, but this line of work, in particular the non-blocking symmetric hash-join also described here, is nowadays frequently cited in articles on stream query processing and peer-to-peer databases.

For me personally, a reminiscence of this paper automatically becomes a reminiscence of its primary author, Annita Wilschut, who is no longer active in our field after health problems. She was the coordinator of the (post-PRISMA) research project containing my Ph.D. track. Her sense of community and willingness to share knowledge, resulted in her teaching me to be a scientist, and enthusistically transmitting basic "database values", so crucial for our research field (e.g., the importance of abstract query languages that speciy "what" not "how", allowing both data independence and query optimization, etc. etc.).

Ihab Ilyas, University of Waterloo, `ilyas@uwaterloo.ca`.

[Ronald Fagin, Amnon Lotem and Moni Naor. Optimal aggregation algorithms for middleware. Proceedings of the 2001 ACM Symposium on Principles of Database Systems, pp. 102–113; full version in the Journal of Computer and System Sciences 66 (2003), pp. 614-656.]

There are many papers that significantly influenced my research; hence, I decided to pick the paper with the most direct impact on my Ph.D. thesis. I became first aware of this paper by attending the best paper award presentation in PODS 2001, Santa Barbara. I still remember the lively presentation of the paper by Ronald Fagin. I was impressed by the fact that the algorithms are extremely intuitive and the optimality results are very strong. I read the paper during the conference and its extended version from Ronalds website, later on. I could immediately see a strong relationship to what I was doing at Purdue; I was working on providing efficient query processing techniques for top-k similarity queries, addressing the inefficiency of current query engines in handling this type of query, which is dominant in many applications.

This paper introduced a family of optimal rank-aggregation algorithms for combining multiple ranked lists according to some monotone combining function. The objective was to get the overall top-k objects using the minimum number of accesses to the inputs lists. Reading the paper raised several research questions, one of which was how to leverage these simple and extremely efficient algorithms in relational databases to develop *rank-aware* query processing. In my Ph.D. thesis, I focused on generalizing these algorithms for joins and other relational operations and on developing relevant cost models for use in query optimization.

Although it has been only a few years since this paper has been published, it has been heavily referenced in many contexts, especially in top-k query processing. Because of their simplicity and optimality, the algorithms are highly applicable in many emerging applications and research problems.

Volker Markl, IBM Almaden Research Center, `marklv@us.ibm.com`.

[Rudolf Bayer, Edward M. McCreight. Organization and Maintenance of Large Ordered Indices. Acta Inf. 1: 173-189 (1972).]

The B-tree data structure and its seminal paper "Organization and Maintenance of Large Ordered Indices" have heavily influenced my research and work from my undergraduate years until today. I read this paper first as a student in 1988, and — always having been interested in data structures and algorithms — found very interesting the idea of having a tree grow "upside down", by splitting the root, challenging the way I was thinking about data structures. This paper had severe impact on me in several ways: (1) Based on this paper I chose the advisor for my Ph.D. thesis, as I wanted to work with one of the persons who had revolutionized data structures in this unconventional way. (2) The paper served as a basic guidance for my Ph.D. thesis, and beyond. I still find it to be a very interesting read and can discover a wealth of ideas and principles in it to be transferred into other areas. The design principles of B-trees not only motivated my research in data structures in the late 1990s, but also taught me the importance of self-tuning systems, as the B-Tree structure self-optimizes when processing insert or delete operations. In addition, it taught me that robustness has to be a major point of concern when inventing algorithms or building systems: if research is to be practical, it must give worst-case guarantees for the overall performance of an algorithm or system. I learned that merely finding a few cases where an algorithm A outperforms another algorithm B is not good enough to justify that algorithm A is worthwhile. Solid work requires guarantees for both space consumption and response time, for both query and modification operations — aspects which are often forgotten in current data structure research. Both self-tuning systems and robustness are the focus of my current research, not anymore in the area of data structures, but in the broader context of query processing, information management, and information integration. It seems that it takes more advanced methods and technologies, from machine learning, statistics, and information theory, to achieve the design objectives of B-Trees in these areas. However, the simplicity of the B-Tree and its algorithms constitute its beauty, and the design goals the B-Tree achieves give a roadmap to any researcher of what needs to be taken into account when conducting successful, relevant, and practical research.

Vasilis Vassalos, Athens University of Economics and Business, `vassalos@aueb.gr`.

[Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv and Divesh Srivastava. Answering Queries Using Views. Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 22-25, 1995, San Jose, California, pages 95-104. ACM Press, ISBN 0-89791-730-8.]

I read this paper as soon as it came out in 1995, as a young PhD student. Our Stanford group was working on information integration, in the TSIMMIS project, and I had started exploring query processing and optimization issues in information integration. This paper provided not only tools and insight useful in my own research but also the theoretical framework for modeling and understanding some of the unique query processing challenges in information integration. Together with the paper Answering Queries using Templates with Binding Patterns, by Rajaraman, Sagiv and Ullman, which was published in the same conference proceedings, it provided the basis for much of the theoretical work on information integration.

The Levy et al. paper was not the first one to deal with the problem of rewriting queries so that they use views instead of base tables - there was for example already the work by Yang and Larson on query transformation for SPJ queries and the work by Tsatalos, Solomon and Ioannidis on GMAP. The paper made a difference a) by providing semantic conditions for deciding the applicability of views, b) by demonstrating the applicability of the existing elegant work on intentional query optimization (while extending it) and c) by formalizing the problem in a way that allowed subsequent work, even as it expanded the scope of the investigation in many directions, to use a similar framework and terminology, which significantly accelerated progress.
