The Evolution of Vertical Database Architectures – A Historical Review
(Keynote Talk)

Per Svensson

Dept. of Decision Support Systems, Swedish Defence Research Agency,
SE 164 90 Stockholm, Sweden
per.svensson@foi.se

1 Background

My intention in this lecture is to discuss the evolution of key concepts behind today's emerging vertical database architectures. The Cantor project [5, 7] pioneered the analysis and coordinated application of many of these concepts in relational systems, which is one reason why references to this work are a recurring theme in what follows. The other reason is that although the work was duly reported in reasonably well-known conference publications, it has left no trace in citations. Thus, from a strictly evolutionary perspective, Cantor was a dead branch which left no progeny, but from a historical perspective it might still provide useful lessons.

Transposed files as such were used in a number of early non-relational data base systems, mostly intended for statistical or scientific applications. A fairly comprehensive list of such systems was given by the paper [6] which is cited below. One great conceptual step that is now being taken is the realization that the adoption of transposed files opens a whole range of architectural opportunities. By careful combination of these opportunities dramatic performance gains may be provided, in particular of course when systems are used in those statistical and analytical kinds of application for which the concept was originally developed.

So what do these architectural opportunities consist of? Below is a list, however, due to space limitations it is not possible here to give a fair account of all of them:

1. column-wise storage of data, or *fully transposed files*
2. use of ordering
3. use of various kinds of "light-weight" data compression: RLE, minimum byte size, dictionary encoding, differencing
4. dynamically optimized combinations of these and other compression techniques
5. use of run-length encoding (RLE) for columns that are ordered
6. lazy decompression of data
7. use of vectorized method interfaces to reduce call overhead costs
8. special method interfaces for accessing RLE-coded data to enable higher-level search and join operations to work directly on compressed data when available
9. use of B-tree variants or other techniques to store and retrieve variable-length data in columns
10. conjunctive search and join algorithms working directly on ordered, RLE-compressed data

B. Ludäscher and Nikos Mamoulis (Eds.): SSDBM 2008, LNCS 5069, pp. 3–5, 2008.

11. use of the vectorized data flow network architecture paradigm and vectorized operations on data streams, to allow efficient query evaluation by interpretation of algebraic expressions rather than by compilation to low-level code.

Today's experiments and analyses are usually better planned and executed than those of the early days. It is therefore at least possible that the current research interest in vertical architectures will result in a better-founded kind of consensus than was achieved, and criticized in [6], see below, in the early 80´s for the standard tabular scheme for storage of relations.

2 The Effects of Modern Processor Architectures

The research group behind the Monet DBMS [9, 10] has made thorough analyses of the effect of modern computer hardware architectures on data base performance. In [9] a detailed discussion is presented of the impact of modern computer architectures, in particular with respect to their use of multi-level cache memories to alleviate the widening gap between DRAM and CPU speeds that has been a characteristic for computer hardware evolution since the late 70's. They show that it is progressively less appropriate to think of the main memory of a computer system as "random access" memory, and that accessing data sequentially also in main memory may provide significant performance advantages.

3 Transposed Files and Decomposed Storage Models

The term transposed file was used in early papers, such as [2,3,4], to denote what is today usually called "vertically fragmented" or "vertically decomposed" data structures [9], "vertical partitioning", "column-oriented" data bases [12] or "column store" data bases [11]. In my opinion, there is a need for more terminological consistency here.

The first published paper on transposed files and related structures that is widely recognized in recent literature is [6]. While the paper notes that some database systems use a fully transposed storage model, in this paper the *fully decomposed storage model* (DSM) is described.

A DSM is a *[fully] transposed storage model with surrogates included*. The authors conclude: "There seems to be a general consensus among the database community that the [conventional] n-ary approach is better. … Instead, we claim that the consensus opinion is not well founded and that neither is clearly better until a closer analysis is made."

4 Data Compression

Two recent papers on the use of data compression in relational data bases are [8, 12]. The most significant advantages are obtained when combining data compression with ordered, fully transposed file or DSM storage, but there are also approaches which use compression for row-oriented storage schemes. The paper [12] also addresses querying compressed data and is briefly discussed in the lecture. In [7], Cantor's

approach to data compression is described. It presupposes the existence of an efficient way to organize attribute sub-files containing varying length data.

5 Conclusions

Based on a literature review, it appears that most of the advantages of vertical storage in databases for analytical purposes have been known and exploited since the early 80's at least. As an early contributor to this technology, the author is happy to see a previous lack of interest at last reverse into what might be seen as a canonical vertical architecture, replacing the previous ill-founded consensus around the "flat file with indexes" approach.

About the Speaker. Per Svensson has been with the Swedish Defence Research Agency (FOI, previously FOA) since 1973 and is a Research Director since 1987. He was an Adjunct Professor of Scientific and Statistical Database Management at the Royal Institute of Technology (KTH), Department of Numerical Analysis and Computer Science, from 1996 to 2002. Previous employments include IBM Sweden and the Royal Institute of Technology (KTH). Dr. Svensson authored or co-authored 30 internationally published scientific papers, as well as about 25 technical reports in Swedish or English. He is the editor and one of the authors of the successful HiTS/ISAC proposal to EU PASR 2005.

References

1. Svensson, P.: Performance evaluation of a prototype relational data base handler for technical and scientific data processing. FOA Rapport C20281-D8. Swedish National Defence Research Institute, Stockholm (1978)
2. Batory, D.S.: On Searching Transposed Files. ACM TODS 4(4), 531–544 (1979)
3. Svensson, P.: On Search Performance for Conjunctive Queries in Compressed, Fully Transposed Ordered Files. In: Proc. VLDB 1979, pp. 155–163 (1979)
4. Turner, M.J., Hammond, R., Cotton, P.: A DBMS for Large Statistical Databases. In: Proc. VLDB 1979, pp. 319–327 (1979)
5. Karasalo, I., Svensson, P.: An overview of Cantor – a new system for data analysis. In: Proc. 2nd SSDBM (1983)
6. Copeland, G.P., Khoshafian, S.N.: A decomposition storage model. In: Proc. 1985 SIGMOD Conf. ACM, New York (1985)
7. Karasalo, I., Svensson, P.: The design of Cantor – a new system for data analysis. In: Proc. 3rd SSDBM (1986)
8. Westmann, T., Kossmann, D., Helmer, S., Moerkotte, G.: The implementation and performance of compressed databases. SIGMOD Record 29(3), 55–67 (2000)
9. Manegold, S., Boncz, P.A., Kersten, M.L.: Optimizing database architecture for the new bottleneck: memory access. VLDB Journal 9(3), 231–246 (2000)
10. Boncz, P., Zukowski, M., Nes, N.: MonetDB/X100: Hyper-pipelining query execution. In: Proc. of 2005 CIDR Conference, VLDB Endowment (2005)
11. Stonebraker, M., et al.: C-Store: A Column-oriented DBMS. In: Proc. VLDB 2005, pp. 553–564 (2005)
12. Abadi, D.J., Madden, S., Ferreira, M.C.: Integrating compression and execution in column-oriented database systems. In: Proc. of the 2006 SIGMOD Conf. ACM, New York (2006)