Feedback on Group Three Presentation:
"Inclusion of New Types in Relational Database Systems"

CSci 8701: Overview of Database Research
University of Minnesota

Michael Goshey, Group 12

Tuesday, October 03, 2006
Introduction

Assignment summary

The assignment was for every member of each group in CSci 8701 to document substantive feedback on the analytical write-ups and presentations of two other groups, resulting in a rich body of constructive commentary for each group to use in evaluating its work.

Selected Article

Group Three selected M. Stonebraker’s Inclusion of New Types in Relational Database Systems as the target of its analysis. The article discusses and concretely demonstrates adding new, non-standard data types to a DBMS.

Feedback

(Note: bulleted items with (w) are for the write-up and (s) are for the slides. Bolded text reflects my emphasis.)

Technical Accuracy

I felt that the analysis was a fairly close rendering of the paper’s content and therefore technically accurate. The following are technical questions I had when reviewing the analysis.

→ (w) "Such operators could be any of the set \{=, <, >\}." I was unsure why <>, >=, <=, etc. were not included in this set.

→ (w) "With this information in place, the query optimizer has enough information to produce an optimal query path when a query is issued on user-defined data types." This statement seems to need a qualifier such as: "...to produce a more optimal query path than random selection."

→ (w) "Therefore, he seems only to be discussing the vision and rational on how to implement the constructs..." I assume by this we mean rationale?

→ (s) On the “Critique/Rewrite” slide: "Unique contributions, however, are not well verified”. This is perhaps a difference of opinion but I found Stonebraker’s contributions to be well-verified. He provides detailed treatment of all relevant aspects of the problem including specific implementation how-to’s. I find the approach similar to how computer scientists treat pseudo-code. We read and understand/verify the soundness of the approach without requiring it be implemented (e.g. in Java), compiled and run.

Presentation Clarity

Both the paper and presentation were relatively clear and easy to follow and have a readable style. The below are several areas for which additional clarity may be helpful:

→ (w) “Access methods are the routines for managing access to disk-based data structures supported by the system.” Since routine is such an overloaded concept in the computer science literature it might be worthwhile to provide an example of a simple routine with actual lines of actual code, etc.

→ (w,s) Both the write-up and the slides include a detailed depiction of a B+ Tree index. I found it curious to single out this particular access method given that in the context of the paper “new” access methods seem more germane. An explanation/diagram of an R Tree might be closer to the “sweet spot” of the paper.
Analysis and Insights

In the respective Rewrite sections of the write-up and slides one finds several potentially interesting suggestions that could be developed into insightful additions to the paper. The following are two examples:

→ (w,s) "...more discussion for transactions and concurrency control." This sounds very interesting but is a bit vague. It would be stronger if the group also provides context, important related questions and/or an example of why it is relevant.

→ (w,s) "...add discussion on query rewrite...". This should be explained and justified/supported with a statement establishing its relevance.

Ideas for Improvement

→ (w,s) I think it would be very interesting to identify (and perhaps challenge) the assumption that it is empirically better to include facilities for implementing new data types rather than limiting oneself to modeling non-standard data types using only standard (built-in) ones. Stonebraker seems to limit his justification to the reduction in query logic complexity but I think it would be an interesting debate to challenge whether all the custom work (and long-term support) involved in implementing custom data types is justified by the benefits.

→ (w,s) Both the write-up and slides take the time to define query optimization. I think several other terms in the paper should also be defined/referenced such as hash join, merge sort, concurrency control and query rewrite, to name a few.

→ (w,s) I would replace the material on the B+ Tree index with a description and visual depiction of an R Tree.