The Volcano Optimizer Generator: Extensibility and Efficient Search

Presentation: Alfred
Discussion: Kati

The Volcano Optimizer Generator
- Object-oriented and scientific database systems
- Allowing query optimization to be more tuned towards the application = higher performance
- (Expert) User optimize

The Generator Paradigm

Model Specification
- Optimizer Generator
- Optimizer Source Code
- Compiler and Linker

Query
- Optimizer
- Plan

Volcano Requirements
- Useable as standalone tool
- Efficient
- Support physical properties
- Expressive – heuristics, directed search, cost functions

Optimizer Generator
- This is not the first time for this approach (EXODUS).
- Volcano improves on the work of EXODUS: ease of use, expressiveness.

Design Principles
Design Principles
- Relational algebra (logical and physical), especially to support OO
- Rules-based => modularization
- Map queries to same algebraic equivalent as Volcano’s input
- Rule compilation rather than interpretation
- Dynamic programming

Optimizer: Input/Output
- Input: User Query => Logical algebra expression
- Output: Algorithms to access physical storage => Physical algebra expression
- How does this relate back to the optimizer generator?

Volcano: Input/Output (1)
- Input: Quite a few things!
- Output: Generated optimizer

Volcano: Input/Output (2)
- Set of logical operators
- Algebraic transformation rules (logical -> physical)
- Algorithms and enforcers
- Implementation rules (operators to algorithms)
- Cost functions
- Logical properties
- Physical property vector
- Applicability function for each algorithm and enforcer
- Property function for each operator algorithm and enforcer

Volcano Plan Search Engine
- Search engine is same for all generated optimizers
- Directed dynamic programming; goal-oriented (driven by needs rather than by possibilities)
- Find costs of promising moves (transform, algorithm, or enforcer)

Volcano Plan Search Engine
- EXODUS did not consider the logical expressions together with the physical properties in the optimization costing. (Whereas Volcano does)
- In OO systems, this can be used to more properly cost access of complex objects.
- Volcano algorithm is top-down (lower levels are explored only when warranted).
Comparison to Starburst

- Starburst has a hierarchy of intermediate levels; harder to see interactions. Volcano uses an algebraic approach which paper claims to be easier to understand.

Comparison to Starburst

- Query rewrites in Starburst does not include cost estimates. (Heuristic)
- Although paper is critical of this, Volcano does allow for heuristic transformations to be specified.

Discussion

Form groups and discuss:
Which of the query optimizers is the most effective and why do you think so? Please write down key points either supporting Starburst or Volcano. What were the advantages/disadvantages provided by both systems? Does the authors’ criticism of Starburst have merit?

How good was it?

- Comparison between Volcano and EXODUS.
- Example used a small data model, consisting of relational select and join operators only.
- As similar data model descriptions as possible were specified for Volcano and EXODUS.

How good was it?

- How long did the optimization take?
- Did the optimizers come out with similar estimates?
How good was it?

- Volcano took less time to optimize.
- EXODUS optimizer generator measurements were quite volatile and took a lot of memory.
- EXODUS’s generated optimizer and search engine do not explore and exploit physical properties and interesting orderings.

Summary

- Tools not just relational databases, but also object-oriented and scientific databases.
- Extensibility using optimizer generator.
- Separation of logical and physical algebras.

Discussion

The authors seem to be very proud of their work. Throughout their paper it sometimes feels as if you were reading an advertisement. Are there issues you are concerned about? Think about the user-friendliness of the system, and other things.

What features do you think might have been improved by following DBMS (OO features ...)?