Extensible Query Processing in Starburst

Presentation: Kati
Discussion: Andrew

Outline
- Motivation
- Solution: Extensible DBMS
- Language Processing
- Query Graph Model
- Query rewrite
- Summary

Motivation
- DBMSs disability to support other applications than administrative ones
- No sufficient support for the functions and data types needed by the engineering's applications
  - additional functions and data types needed

Starburst Project
- Extensibility
  - Language extensions
  - Internal processing extensions
  - Data management extensions
- Worth noting the time when it came up
  - Object-oriented phase

Two major components
- Corona: the query language processor
- Core: data manager

Starburst’s Language: Hydrogen
- Based on SQL
- Orthogonal
- Extensible
- Table expression
- Table function
  - very complex queries possible
Language processing

Two stages: compilation and execution

Query Graph Model

- Vertices
- Edges
- boxes

Query Rewrite

- A form of optimization and a big challenge
- New transformations required
- Rule-based approach
  - Creation of new rule system
  - Greater scope of optimization and improved execution plan

Rule-based Approach

- Rule language is C
- Two parts: condition and action, each written a C function
- Consistency
- Rule classes → Modularization

Rules- three classes

- Predicate migration
- Projection push-down
- Operation merging

Rule engine

- Independent of any rules
- Forward chaining
- Several control strategies:
  - Sequential
  - Priority
  - Statistical
  - budget
No! cost-based query-rewrite
- All alternatives are generated
- At the plan level cost-based
- BUT interaction desired
- SINCE number of alternatives grows tremendously

Cost-based optimization
- Plan generation
- Plan costing
- Search strategy
- Designed to be orthogonal

Plan generator
- Strategy alternative rules (STARs)
- A general-purpose STAR evaluator
- A search strategy that chooses the next STAR to evaluate
- An array of STARs

Summary
- Starburst: extensible DBMS
- Extensions to the language, the language processing and the data manager
- Table expressions allow orthogonality
- Orthogonality & Extensibility → complex queries possible

Summary
- Query internally a QGM
- QGM simplifies the DBC’s task, give him a great deal of flexibility and power
- Rule-based query rewrite
- Grammar-like rules to generate plans