Stats on intermediate tables

Yiwen Tseng

February 24, 2003

The role of statistics in a query optimizer

- Cost estimation
  - What statistics to collect
  - When to collect them
  - Under which circumstances to re-optimize the query
- These statistics are then used to optimize the execution of the query

What we have seen

- Runtime collection of statistics
  - Dynamic collectors of Tukwila
    - Gather statistics about each operation dynamically
  - Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans
    - Filter and statistics collector
- Static collection from the start
  - SIT and MNSA
    - The strategies are for cardinality estimation and not for plan generation

Traditional optimizer:

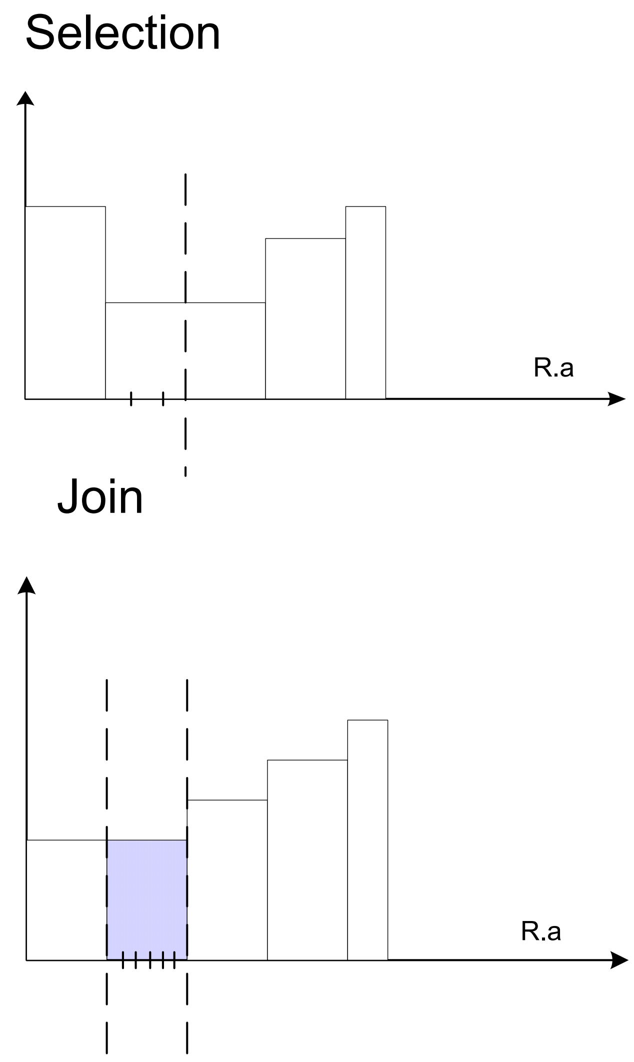
- Quality of the query execution plan depends on the accuracy of cost estimates.
- Cost estimation depends on cardinality estimation of various intermediate results.
- Optimizer uses statistics built over base tables
- Problem:
  - Propagation statistics lead to large estimation errors

Core Idea

- SIT
  - Histograms
  - Materialized view
    - Do some intensive work of the results in advance
  - Use existing SITs to model the distribution of tuples on intermediate nodes
- MNSA
  - Select a small subset of SITs that are sufficient to increase the quality of the query plans by the optimizer

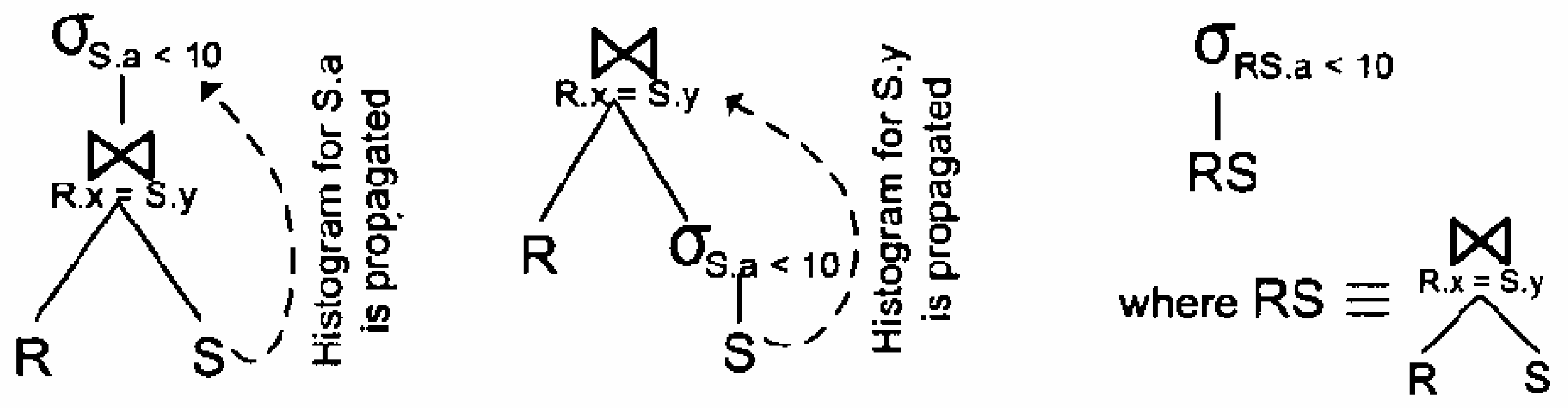
Traditional query optimizer

- Cardinality estimation using histograms
- Selection
- Join



Problem

- propagation of statistics through predicates



(a) S.a is propagated (b) S.y is propagated (c) Extra information is used

Materialized view

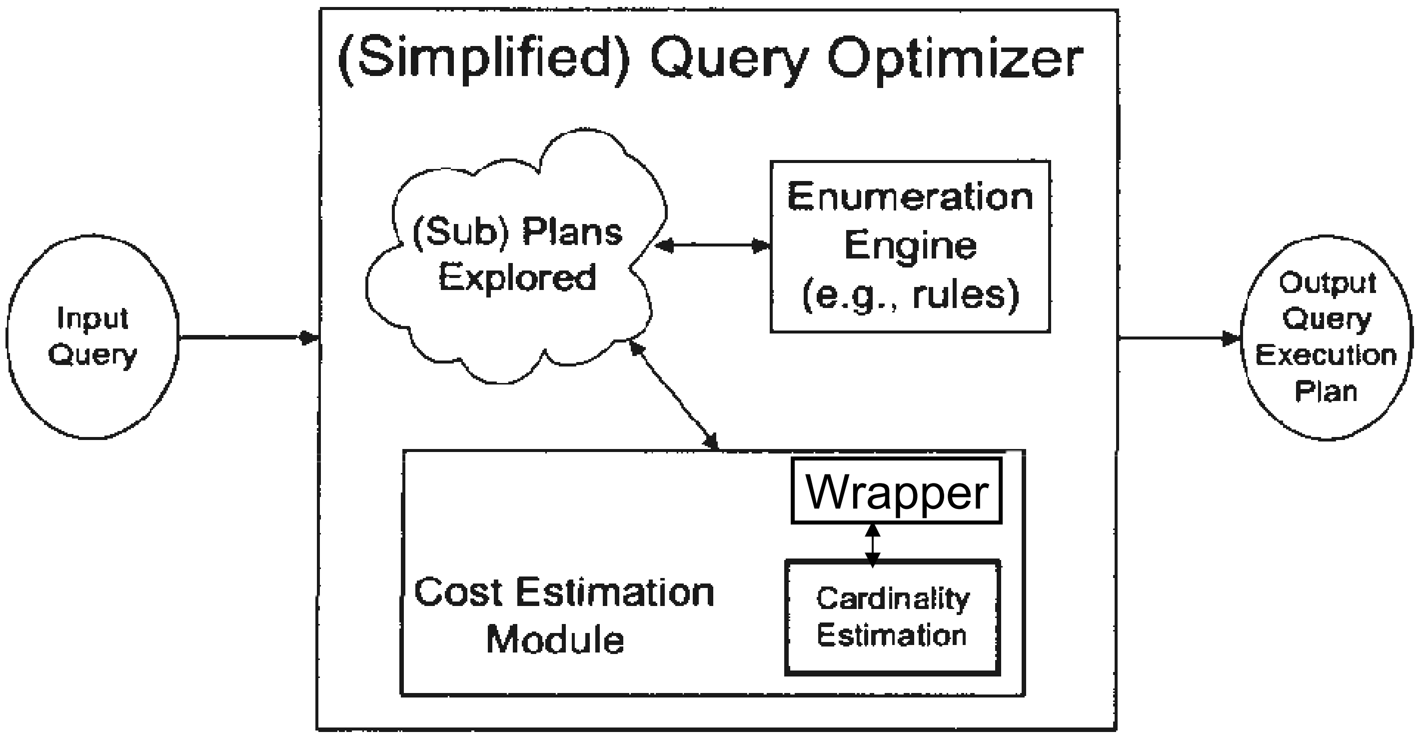
- What is a materialized view?
  - Like a normal view yet in that it contains ACTUAL data
  - Data for the view is assembled when the view is created or refreshed
- Motivation
  - Saving the overhead of performing the work already done by the materialized view

SIT: Statistics on Query Expressions

- Applies
  - Histograms
  - Materialized view
- SITs are created by the system and we assume that they are dependable sources
  - Real computation
  - Approximate query processing

Implementation

- By implementing a wrapper on top of the original cardinality estimation module of the RDBMS



Cardinality estimation using SITs

- Analyze the input
- Identify and apply relevant SITs
  - SIT-Sets should be applied to the query
  - If predicates of the query are not covered by the SIT, apply the auxiliary SITs
- Estimation and return the cardinality of the transformed query plan

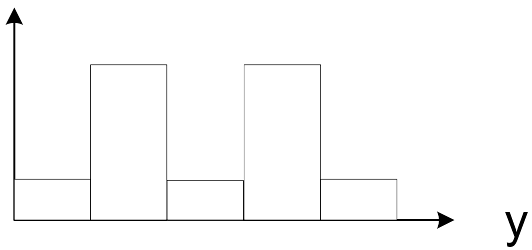
MNSA: Magic Number Sensitivity Analysis

- Goal:
  - To select the most influenced subset of SITs that are sufficient to increase the quality of the query plans
- Approach: Consider workload information
  - Given a query workload and a space constraint
  - Find the set of SITs that fits in the available space
  - So that actual cost is minimized or substantially reduced

MNSA algorithm

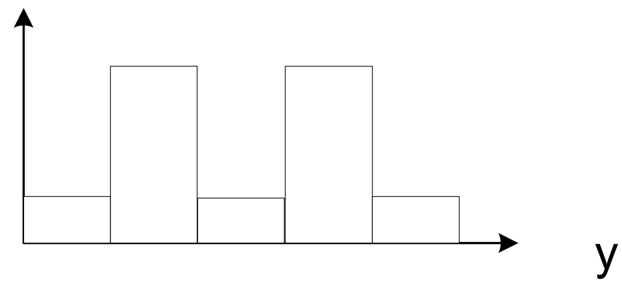
- Incrementally identifies and builds new statistics over the base tables until no additional statistic is needed
  - Magic selectivity number (extreme predicted selectivities) to estimate the absence of statistics
  - Verifies whether the optimized query plans are t-optimizer-cost equivalent
- Problem:
  - MNSA can not apply directly to the optimizer system

Independence strategy

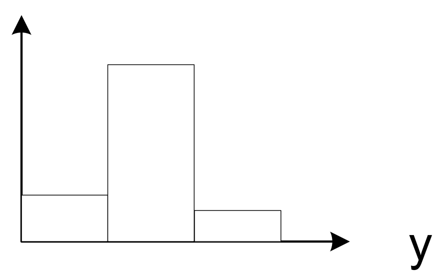


Select * from x<= 5
Selectivity = 1/5

Independence



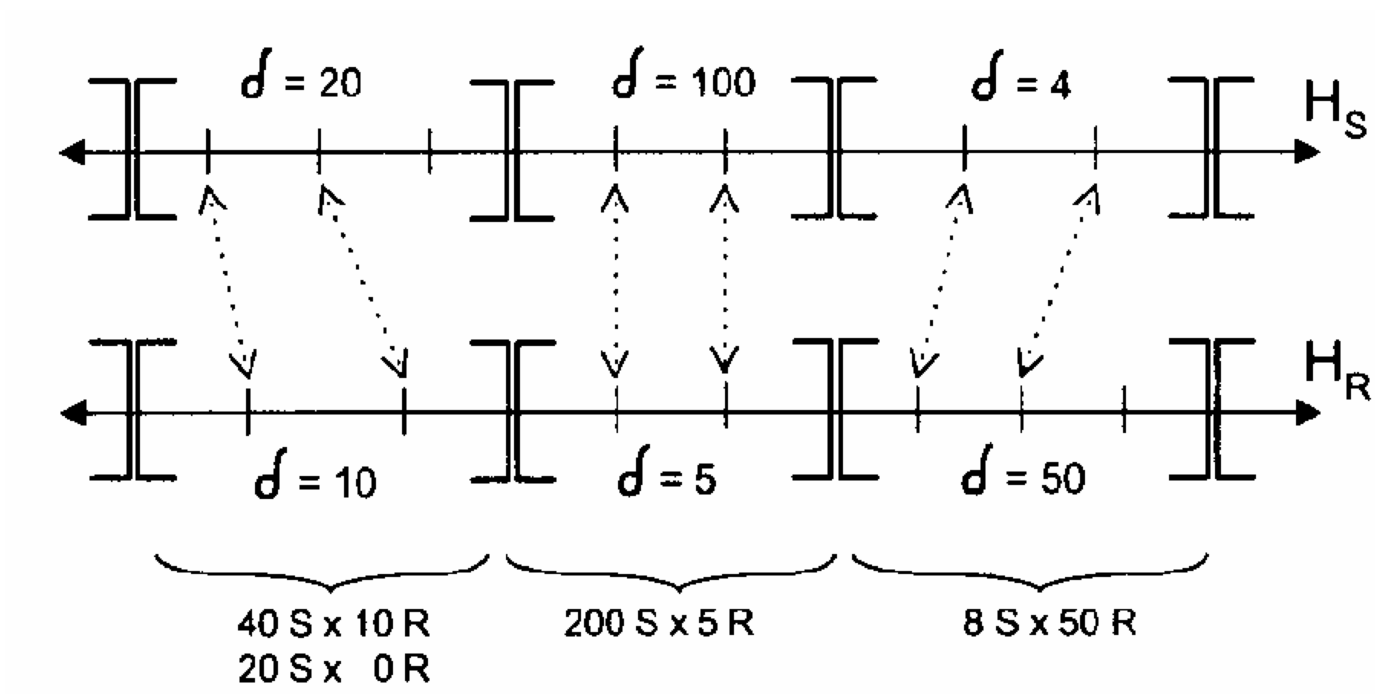
y=x*4



Alternative to get the cardinality estimation : Extreme cardinality estimation

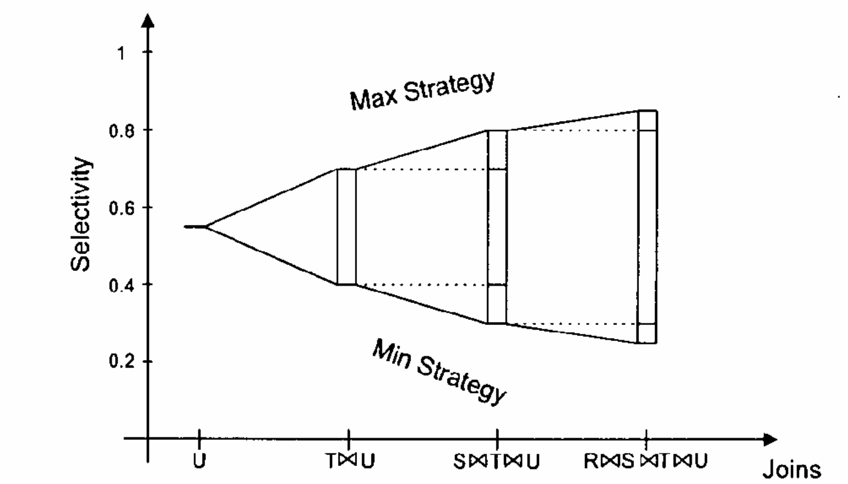
- Max and Min strategy
  - Get the max(min) number of tuples in the join results

S.A < 10 (30 tuples)



Selecting SITs

- Which generating query to use for some SIT
  - Max and Min cost differences should be significant
  - Selectivity estimation for the Min and Max strategies
  - Score of SIT
- Discard non-essential statistics



Conclusions

- Benefits:
  - Better performance
  - Do not need to store and maintain materialized views but only build statistics over those views
- Weakness:
  - Where does the materialized view come from?
  - Histograms
  - The quality of SITs
- Future Works:
  - Extending and evaluate the methodology ofr more complex queries and more complex statistics