Uni Freiburg, Web Science Group Prof. Peter Fischer Systems Infrastructure for Data Science - Winter 2014/15

Exercise Sheet #5: Query Processing and Optimization

Nov 27, 2014

Exercise 6.1 : Pipelining vs. Materialization

Consider the following statements. Decide whether they are true or false. If you believe a statement is false, briefly explain why.

- A. Pipelining reduces query response time, but increases memory requirements.
- B. Merge-join is a blocking operator.
- C. Any sort operator is necessarily blocking.
- D. A fully pipelined query plan may have significantly higher execution time than an otherwise identical query plan, which fully materializes the result of each operator.

Exercise 6.2 : From Queries to Execution Plans

Consider the following partial schema of a database which keeps track of students and the courses they take:

Student(<u>matrikel_no</u>, first_name, last_name) Class(<u>class_id</u>, class_name) Takes(<u>class_id</u>, <u>matrikel_no</u>)

Primary keys are underlined. You may assume the obvious foreign-key relations and indexes on primary and foreign key columns. The table cardinalities are as follows:

|Student| = 1,000 |Class| = 100 |Takes| = 5,000

- A. Express the following natural-language query in SQL: Give me the last names of all students who take the course called "Information Systems".
- B. Convert the SQL statement to a straight-forward query-plan using only cartesian products, not joins. Estimate the cardinality of each intermediate result and the final result.
- C. Push-down the selection operators below or into the cartesian products (turning them into joins). Estimate the cardinalities of each intermediate result using your knowledge of table cardinalities and foreign-key relations.
- D. Choose access paths and physical implementations of the various operators (particularly joins), taking advantage of indexes or sort order. Try to find a near-optimal plan. Revise your join order if necessary.