Review of "Efficient Mid-query Reoptimization of Sub-optimal Query Execution Plans"

http://lsirpeople.epfl.ch/nbonvin/courses/adms07/
Nicolas Bonvin

November 15, 2007

Summary In this paper the authors describe a query optimizer which collects statistics at key points during the execution of a query in order to decide if the execution plan is suboptimal. In that case the optimizer tries to optimize the execution etther by changing the resource allocation or by changing the execution plan.

Innovative ideas

- *Annotated execution plan and live statistics collection*: at runtime the optimizer keeps track of actual cost of given operations (for example the current selectivity) and compare these costs with the initial estimates. When the estimated cost is too different then the query is reoptimized using the new statistics.
- *Estimated inaccuracy potentials for statistics*: The authors describe precisely these potentials under variation of the dataset and various operators. One is now able to insert *statistics collector* operators in the query execution plan.
- *Single-pass statistics collection*: if one inserts a *statistics collector* operator, the tuples will only goes trough this operator once. That is, one can only collect statistics that can be gathered in one pass over the data.

Most glaring problems

- *Needless effort in a pipelined execution*: in order to collect meaningful statistics and use them, one often need to suspend the execution of higher operators and put the current output tuples on disk which has an expensive I/O cost.
- *Reinvokation of the optimizer*: the optimizer needs to be launched and reanalyze the query with the new statistics. This may be an expensive operation due to the number of plans that have to be checked. .
- *Statistics collector overhead*: the statistics-collector insertion algorithm calculates the total execution time available for collection and decides how to split this time into several *statistics collector* operators. Each *statistics collector* operator introduce an overhead proportional to the amount of tuples that goes trough it.
- *Reimplementation of some operators*: in order to take full advantage of dynamic buffer management, some operators need to be changed. But the authors don't explain in detail which operator can be modified.

Conclusion Collecting useful statistics is still an open research area. Why not add these new statistics into the system catalog in order to improve the next "static" query plan generation ?