Database Tuning, ITU, Spring 2008

Rasmus Pagh

May 8, 2008

Project – 5th part and full report

Deadline: May 21, 15.00 Danish time. Hand in at the exam office in *three* copies.

Purpose

The purpose of the final part of the project is to allow students to pursue their own interests, within the scope of the project. While the "core" project is supposed to be done by all group members together, extensions may be pursued by any subset of the group (e.g., by single group members), based on differences in interests or ambition.

The full report documents each group's work on the project, and the ability to communicate clearly on complicated technical matters. It also documents the ability to independently formulate a small research project (a possible thesis project) within the subject area. In addition, it forms the basis for the examination and grade.

Suggested topics

Below you can find some inspiration for topics to address in the last part of the project. Feel free to modify or extend the suggestions, or come up with your own.

Sampling. In the course you have seen several uses of sampling: Query result size estimation, "on-line aggregation", and producing early join results. Use the `SAMPLE` operator of Oracle to experiment with sampling. Can sampling be used to improve, in some sense, any of the queries studied in the project? How do sampling-based estimates of query result sizes compare with the estimates made by the Oracle query optimizer, and with the actual sizes? Is it efficient to maintain a materialized view that is a sample of a relation?

OLAP and bitmap indexing. Come up with some OLAP-like queries in the context of the project, and try to implement them such that response time is small. You may use materialized views and "on-line aggregation". While bitmap indexes are not explicitly supported in Oracle XE, it is possible to use a normal (B-tree) index to achieve a similar effect. The idea is to build a materialized view where each tuple contains a small part of the bitmap. To access the bitmap, use the `bitand(x,y)` function which takes two binary integers, and returns the integer resulting from a bit-wise conjunction of the bit strings.

Spatial indexing. Try out some spatial indexing technique for the "nearest point" query on the geographical data in the VXL database. At least two indexing techniques can be tried, by reduction to a one-dimensional query: Space-filling curves (Z-ordering is the easiest) and grid files. Let's call a grid file *uniform* if all the grid rectangles are squares of the same size. Consider a uniform grid file, based on the GPS coordinates, for road segment endpoints. Implement this such that each square has at most 1000 points in it, and such that the total number of squares is (close to) mimimal. Use a clustered B-tree index to make sure that points in the same square are stored physically together.

Temporal databases and OLAP. Do a literature study on subjects in the intersection of temporal databases and OLAP, e.g., read (part of) the papers [1, 2]. You should focus on writing a short overview of selected aspects, aiming at a reader who is familiar with OLAP and temporal databases, but not familiar with ways of combining them. In particular, you should make the relationship to the contents of DBT clear.

Index design. There is a rich literature giving practical advice on database performance tuning. Tapio Lahdenmäki (`www.tapio1.com`) has received a lot of attention in recent years for his method for choosing indexes. Describe the main aspects of Lahdenmäki's method (described in the book *Relational Database Index Design and the Optimizers*), and evaluate it based on the knowledge and experience you have from the course. What are the strengths and weaknesses of the method? Alternatively, describe and evaluate the main points in the articles *Optimizers are not Perfect*, available at `www.tapio1.com`.

The full report

The report should be written as if the reader is a student who has followed the lectures of DBT, but not worked on the project. If it seems relevant, the report may summarize parts of the material covered by the DBT lectures, but it is the *application* of this knowledge, with accompanying analysis and arguments, that is the main focus.

The full report should contain:

- An introductory section describing the relevant prerequisites (courses and projects) of each group member. Also, you must clearly describe how each group member contributed to the project and report, to enable individual grading.
- The same information as in answers to the four deliverables. You may, and probably should, revise this material according to the feedback received. It is perfectly fine to arrange the material differently than according to deliverable number and question number.
- You are encouraged to extend your discussion to aspects that seem relevant for the VXL case, but not touched upon by the deliverables. For example, if there are cases where acceptable performance could not be achieved, you may discuss what could be done to alleviate this, including restrictions on what (or when) queries are performed, hardware enhancements, etc. If you need more specific knowledge than what is available, make reasonable assumptions.
- An experience report: What aspects of the work were harder (or easier) than expected? Was this because of special restrictions (or features) of the DBMS software, hardware limitations, or other constraints? Was there any information that you would have liked to have at the beginning of the project? Etc.
- Documentation of the programming performed: Printouts of Java programs, DDL statements, and other relevant files. These should be easily understandable in the context of the report (e.g., through a combination of in-code comments and overview sections in the report).
- A proposal for at least one thesis project related to performance of databases. You are supposed to independently formulate a small "research problem" that would be interesting to pursue, based on the knowledge you have obtained in DBT. Also, you should outline a possible methodology using which the problem could be addressed. You are *not* expected to investigate the literature on the subject to find out if the problem has been addressed before.

If some section of the report (and the programming work behind it) was done by a proper subset of the group members, this should be indicated below the section heading. At the exam, you will be expected to be able to discuss any section on the "core" part of the project, whereas a section on an extension will not be discussed with students who did not contribute to it.

Of course, if you have used (even for inspiration only) any papers or books when doing the report, these should be appropriately cited.

References

[1] Michael H. Böhlen, Johann Gamper, and Christian S. Jensen. Multi-dimensional aggregation for temporal data. In Yannis E. Ioannidis, Marc H. Scholl, Joachim W. Schmidt, Florian Matthes, Michael Hatzopoulos, Klemens Böhm, Alfons Kemper, Torsten Grust, and Christian Böhm, editors, *EDBT*, volume 3896 of *Lecture Notes in Computer Science*, pages 257–275. Springer, 2006.

[2] Jun Yang and Jennifer Widom. Incremental computation and maintenance of temporal aggregates. *VLDB J*, 12(3):262–283, 2003.