Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

December 12, 2019

Databases IIB: DBMS-Implementation
— Exercise Sheet 8 —

Please read Part a) and mark questions which you want to discuss in class. You only have to submit Part d) to h). Please upload your solution into the StudIP file folder called "Hausaufgabe_8" in the StudIP entry of the lecture. The deadline is December 17 (the day before the next lecture).

It is permitted to form groups of up two members, but please make sure that both members can fully explain all homeworks submitted by the group. Please upload only one file per group. Name the file such that it contains the names of all group members.

Not all submitted homeworks will be corrected, but all homework exercises will be discussed in class. If you should have questions about your homework, please ask! A precondition for getting credit for this course is that you submit solutions to two thirds of the homeworks. Obviously wrong or very incomplete submissions do not count.

Repetition Questions

a) What would you answer to the following questions in an oral exam?

- Name some storage characteristics and compare main memory with disks.
- What is a "logical block access", a "physical block access", a "cache hit" and a "cache miss"? How is the "hit ratio" defined?
- Suppose there were 1000 logical block accesses and 100 physical block accesses. What is the cache hit ratio?
- What are reasonable cache hit ratios for a normal OLTP database?
- Why does one often get reasonable cache hit ratios even if the buffer in main memory is much smaller than the data files on disk? E.g. if the main memory buffer is only 20% of the total database size, one would still expect a better hit ratio than 20%. Why?
- Suppose, one has a relatively large database (e.g. 60 GB), but also a large server machine with e.g. 128 GB RAM. What would be advantages and disadvantages of a standard DBMS with a cache for disk blocks compared to a system that first reads the entire data to memory?
- Why does a query run much faster if one executes it a second time (soon after the first time)?

- Suppose you want to compare the performance of two database management systems. You create the same database in both systems and measure the runtime of a few example queries. Why might the system vendors (or their salespeople) consider the result questionable?
- Explain the interface of a typical buffer manager.
- Explain the LRU replacement strategy. What do the letters "LRU" stand for?
- What is the problem with "sequential flooding" of the buffer? What is the solution in Oracle?
- What information might the upper layers of a DBMS have that they should pass to the buffer manager for improving the replacement strategy?
- What is the difference between "consistent gets" and "db block gets" in Oracle?
- How can one get information to determine the cache hit ratio in Oracle?
- Suppose you detect that the cache hit ratio is bad in your Oracle database. What can you do?
- What does the `CACHE` parameter of the `CREATE TABLE` do in Oracle?
- What is the purpose of the buffer pools `DEFAULT`, `KEEP` and `RECYCLE` in Oracle? How can they be used to improve the cache hit ratio?
- What is the main idea of the "Five Minute Rule"? Why might it be necessary to buy more disks although the overall disk space is sufficient? Why can buying more RAM solve the problem?

Some Classical Papers

b) Have a look at at least one of the following papers:

- Wolfgang Effelsberg and Theo Haerder:
  Principles of Database Buffer Management.
  ACM Transactions of Database Systems, Vol. 9, No. 4, Dez. 1984, pp. 560–595.
  [http://dx.doi.org/10.1145/1994.2022]
- Jim Gray and Gianfranco R. Putzolu: The 5 Minute Rule for Trading Memory for Disk Accesses and The 10 Byte Rule for Trading Memory for CPU Time.
  Proceedings of the ACM SIGMOD Conference, 1987, pp. 395–398.
  [http://dl.acm.org/citation.cfm?doid=38713.38755]
- Goetz Graefe: The Five-minute Rule: 20 Years Later and How Flash Memory Changes the Rules.
  ACM Queue, Vol 6, Issue 4, September 24, 2008.
  [http://queue.acm.org/detail.cfm?id=1413264]

You can access the PDFs of articles published by the ACM from IP-addresses of the university, because the university pays for using the ACM digital library.

Rules of the Competition

c) Please start thinking about an implementation exercise that will be the basis of a competition. You do not yet have to submit something. Before you really start programming, you probably should know something about index structures.

- You will get a file in the same format as `homeworks.txt`, but quite large (many MB). I.e. each line in the file represents a a row in a table with columns `FIRST_NAME`, `LAST_NAME`, `EXERCISE_NO`, `POINTS`. The values are separated by a vertical bar:

  ```
  Ann|Smith|1|10
  ```

- The first step is that you read the input file and create your own file with the data. Your program will be called with three command line arguments: The option "`-c`" (for "create"), the path to the input file (replacing "`homeworks.txt`") and the name of the data file that you may write.

  The size of your data file may not be more than three times the size of the input file. The CPU time for this step will be measured. Furthermore, the memory your program uses ("maximum resident set size") will be measured. You will be disqualified if it is more than ten times the size of the input file.

  It is legal that you simply copy the file if you think that the file format is already good for the queries (see below). The input file has no particular order.

  You are not allowed to use Linux commands like "`cp`", you must program everything yourself in C++. You are also not allowed to use any non-standard libraries. Furthermore, you are not allowed to use complex data structures such as maps from the C++ library, you must program that yourself. (If you like, you can create an alternative version that uses any library you want. We will measure the times for that alternative version, too, but the program will not be part of the competition.)

- Now, your program will run ten times with first name and last name given on the command line. You have to print the result of some simple computation of the exercise numbers and points of the table rows for the given student. For instance, this could be the sum of the homework points this student has reached. However, the exact formula will be told to you after you have created your data file. The purpose of this rule is that you cannot precompute the value for each student in the first step and store already the result in your data file. You must be able to access homework numbers and points for the given student. It is suggested that your program contains the code for computing the sum of the homework points. In the competition, you have 15 minutes to change this part of the program for the then published formula. For instance, it could also be the sum of the exercise numbers (although this would not make much sense in practice). Note that exercise numbers are not necessarily sequential, and could possibly span the entire range of (non-negative) integers.

  The runtime (CPU time) will be measured for each student name, and the all the runtimes will be added (e.g. the times for ten searches and one preparation of the file). The result is the measure for the computation.

  Again, you will be disqualified if your program has a resident set size (in RAM) of more than 10 times the size of the `homeworks.txt` file. Note also that you are not allowed to access the `homeworks.txt` file in the queries — all necessary data must be in your own file. Note also that your program must terminate after each query, i.e. you cannot keep the data in RAM between the queries.

Homework Exercise 8

d) Write a `CREATE TABLE` statement with the `TABLESPACE`, `BUFFER_POOL` and `CACHE` parameters. The table should contain two columns for a statistics number and a statistics value as in `V$SYSSTAT` or `V$SESSTAT`:

- [https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/V-SYSSTAT.html]
- [https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/V-SESSTAT.html]

You may add additional columns if these are useful for g) below. If needed, the `CREATE TABLE` reference is here:

- [https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-TABLE.html]

e) Check whether the initialization parameter "`TIMED_STATISTICS`" is true or false.

- [https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/TIMED_STATISTICS.html]

You can set it locally for your session with the `ALTER SESSION` command:

- [https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/ALTER-SESSION.html]
- [https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/ALTER-SYSTEM.html]

However, when you use Adminer, each call is executed as a separate session. You may, however, put several SQL statements, separated by "`;`", into the SQL input window. If you execute that, they will run in one session.

The parameter can be set globally with the `ALTER SYSTEM` command. However, that would require rights that you do not have yet. I might do that.

f) Copy at least the three statistics values you need for computing the cache hit ratio into your table. You may store additional data.

g) Run the following query:

```
SELECT COUNT(*) FROM DICT_COLUMNS
```

Now write a query that computes the difference of the current statistics values and the values stored in your table (which were current before you executed the query), and outputs these differences together with the names of the statistics values and in addition the cache hit ratio. You might also output additional performance measures if you want. The result should be a table with two columns, the name of the performance indicator in the first column (such as "`Hit Ratio`"), and the value in the second.

h) Consider a buffer cache that operates with the LRU method, and has only four buffer frames. Suppose that the following blocks are accessed (pinned and immediately unpinned):

10, 12, 15, 20, 30, 12, 40, 15, 10, 12.

Which blocks are in the buffer at the end, and what was the hit ratio?