Prof. Dr. Jens Dittrich
Teaching Assistant: Josiane Xavier Parreira

Assignment 2

SAARLAND UNIVERSITY
COMPUTER SCIENCE

1 Buffer Management in DBMS versus Disk Cache

Modern disks often have their own main memory caches, from 8 to 32 MB nowadays, and use this to prefetch pages. The rationale for this technique is the empirical observation that, if a disk page is requested by some (not necessarily database!) application, 80% of the time the next page is requested as well. So the disk gambles by reading ahead.

1. Give a nontechnical reason that a DBMS may not want to rely on prefetching controlled by the disk.
2. Explain the impact on the disk's cache of several queries running concurrently, each scanning a different file.
3. Is this problem addressed by the DBMS buffer manager prefetching pages? Explain.
4. Modern disks support *segmented caches*, with about four to six segments, each of which is used to cache pages from a different file. Does this technique help, with respect to the preceding problem?

2 SSDs vs. Hard Disks

One major disadvantage of Solid-state drives is their limited write cycles. This problem is reduced by spreading writes over the entire device (so-called wear leveling), rather than rewriting files in place. Assuming the following specs of a SSD disk:

Capacity:	64GB
Write endurance rating:	1 Mio. cycles
Sustained write speed:	50MB/sec

1. Assuming a perfect wear leveling, how long have you got before the disk is trashed if you perform non-stop writing operations?
2. Discuss the differences between SSDs and hard disks in terms of read/write operations an sequential/random access.
3. Give (at least) one example of a DB operation where SSDs outperform hard disks by a large amount.

3 Horizontal vs. Vertical Partitioning

Consider a table with 10 attributes and we want to perform an operation that involves only 2 of these attributes. For each of the following storing models: NSM, DSM, PAX, and Fractured Mirrors, answer:

1. What percentage of the data is loaded into the memory and the L1 cache, respectively?
2. If you want to write one row into the table, how many I/O operations are needed?

4 Compression

Given an algorithm that requires 50 CPU clocks per byte to decompress a given data. If we have a single 2GHz CPU machine and the I/O operations can be done at a speed of 100MB/s, what is the minimum compression ratio that justifies the use of the compression?