Chair of Applied Computer Science III
Prof. Dr. Guido Moerkotte
Email: moerkotte@uni-mannheim.de

Daniel Flachs
Email: daniel.flachs@uni-mannheim.de

Database Systems II — Solution to Exercise Sheet 10
Spring Semester 2019 — Created May 15, 2019

Exercise 1

Note: The following task can be solved as soon as Chapter 12 (Cardinality Estimation) from the script has been presented in the lecture. This will most likely be the case on May 13, 2019.

Assume you are given the following Relation R:

R

K	A
0	a
1	a
2	b
3	c
4	c
5	c
6	c
7	d
8	e
9	e

Exercise 1 a)

Build a frequency vector for Attribute A of relation R.

Solution

$f = \langle 2, 1, 4, 1, 2 \rangle$ in lexical order of the values in $R.A$.

Exercise 1 b)

The k^{th} frequency moment of a frequency vector f is defined as

$$F_k(f) = \sum_{i=1}^{n} f_i^k.$$

Compute the frequency moments for $k \in \{0, 1, 2\}$. What is the logical meaning of each of the values?

Solution

(i) $k = 0 : \sum_{i=1}^{5} f_i^0 = 2^0 + 1^0 + 4^0 + 1^0 + 2^0 = 5.$
Gives the number of distinct values.

(ii) $k = 1 : \sum_{i=1}^{5} f_i^1 = 2^1 + 1^1 + 4^1 + 1^1 + 2^1 = 10.$
Gives the number of values.

(iii) $k = 2 : \sum_{i=1}^{5} f_i^2 = 2^2 + 1^2 + 4^2 + 1^2 + 2^2 = 26.$
Gives the self-join size.

Exercise 1 c)

Sketches can be used to approximate the size of a join. Read *Sketches for Size of Join Estimation* by Rusu and Dobra, Section 1.1, and study Examples 2 (Section 3.1) and 3 (Section 3.2). In order to understand the examples, you may have to read more of Section (2 and) 3.

`http://faculty.ucmerced.edu/frusu/Papers/Journal/2008-05-tods.pdf`

Exercise 2

Note: The following task can be solved as soon as Chapter 13 (Parallelism) from the script has been presented in the lecture.

Amdahl's law allows you to compute the theoretical speed-up of the execution of a task when using n processors instead of 1. p denotes the proportion of the task that is parallelizable. The formula is given by

$$\frac{1}{(1-p) + \frac{p}{n}}.$$

Exercise 2 a)

Compute the speed-up according to Amdahl's law for $p \in \{0.1, 0.5, 0.9\}$ and $n \in \{2, 8, 32\}$.

Solution

		p		
		0.1	0.5	0.9
	2	1.05	1.33	1.81
n	8	1.09	1.77	4.71
	32	1.11	1.93	7.80

Exercise 2 b)

What are the implicit assumptions of Amdahl's law?

Solution

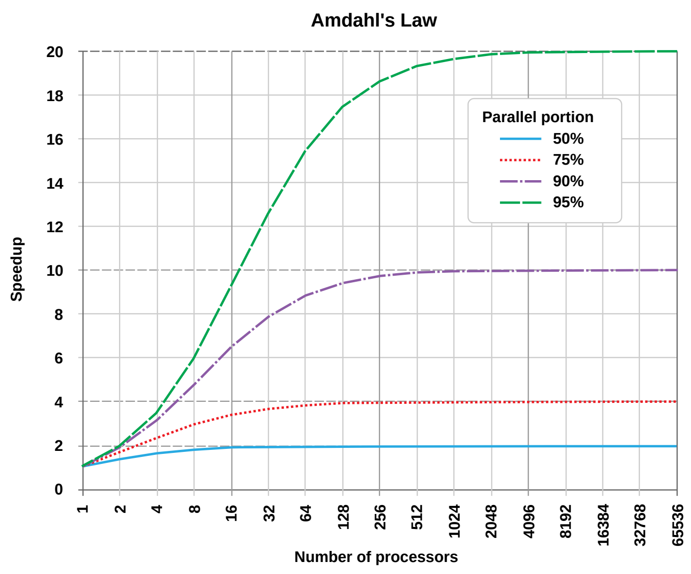
- Adding another processor/thread does not result in any overhead, e. g., for thread generation or synchronization. Therefore, Amdahl's law assumes perfect scalability, which is unrealistic.
- The parallelizable fraction p is divisible into arbitrarily small pieces, i. e., it can be distributed across arbitrarily many processors/threads. In mathematical terms, we can let $n \to \infty$. This is unrealistic. Consider the evaluation of a selection predicate on 1 000 000 tuples. Obviously, this can be parallelized since two tuples are independent. However, $n > 1\,000\,000$ does not make sense as the $1\,000\,001^{\text{th}}$ thread is not assigned any work.

Exercise 2 c)

In the script in Chapter 13, p. 117, you can find a figure showing the speedup factor (y-axis) according to Amdahl's law depending on the fraction $p \in [0, 1]$ of parallelizable instructions (x-axis) for different number of processors n.

Sketch a diagram that shows the speedup factor on the y-axis and the number of processors n on the x-axis. Draw graphs for $p \in \{0.50, 0.75, 0.90, 0.95\}$. To which limits do the curves converge for large n?

Solution



Source: `https://commons.wikimedia.org/wiki/File:AmdahlsLaw.svg` (10/05/2019)

For large n, i. e., $n \to \infty$, the term $\frac{1}{(1-p)+\frac{p}{n}}$ converges to $\frac{1}{1-p}$.

Exercise 2 d)

Discuss the difference between *inter-query parallelism* and *intra-query parallelism.*

Solution

- **inter-query parallelism**: run independent queries in parallel
- **intra-query parallelism**: partition relation and process partitions in parallel (within strands), process independent strands in parallel (bushy parallelism)

How to Study for the DBS II Exam

Recall the topics covered in the lecture:

- Hardware
- Hashing
- Compression
- Storage Layout
- Physical Algebra: Processing Modes & Implementation
- Expression Evaluation
- Indexing
- Boolean Expressions
- Cardinality Estimation
- Parallelism

We spent a fair amount of time on Hardware and Hashing. However, we are neither electrical engineers nor mathematicians, so questions regarding these topics will either be not super deep or *from the perspective of a computer scientist*.

Proofs and product-specific stuff is of minor importance for the exam.

Programming is necessary to implement the presented concepts and will be important for the exam. Despite we used C++, this is a not a C++-lecture. In terms of the exam, basic programming concepts are important. In the exam, if you do not know how to answer a question using C++, use pseudo code. Correct C++ code will always give you full points. Pseudo code tends to be more vague and hide certain implementation details, e.g. regarding efficiency. Therefore, make sure to be as specific as possible, especially with regard to the data structures used. The storage layout chapter is an ideal candidate for programming questions. It is suggested to take a look at the slides/script and the exercises.

Expect the exam questions to check for an basic understanding of the presented topics. We also ask questions like *Name n properties of Y* and *Give the definition of X* since these are easily gained points for many students. A few (hard) questions that are designed to be *unexpected* are possible and play a role if you aim for very good grade.