Database Systems II – Exercise #9

Sheet #9 (cont'd): CSB$^+$ Trees, Boolean Expressions

Sheet #10: Cardinality Estimation, Parallelism, Exam Preparation

Daniel Flachs

Chair of Practical Computer Science III:
Database Management Systems

15/05/2019

UNIVERSITY
OF MANNHEIM
School of Business Informatics
and Mathematics

Contents

Contents

Task 3

Assume you are given the following query:

```
SELECT *
FROM R
WHERE Age > 27 and Income > 30.000 and Weight < 75;
```

We refer to the predicates in the query by the set

$$P = \{Age > 27, Income > 30\,000, Weight < 75\} = \{p_1, p_2, p_3\}$$

Task 3

Furthermore, assume you are given the following sample taken from R:

ID	Age	Income	Weight
1	28	40,000	80
2	30	55,000	50
3	27	37,000	75
4	40	60,000	60
5	42	62,000	85
6	22	15,000	55
7	70	20,000	67
8	50	80,000	57
9	55	85,000	86
10	33	42,000	58

Recap: Gamma Sampling

- **Goal**: Given a sequence of z selections $\sigma_{p_1}(\sigma_{p_2}(\dots\sigma_{p_z}(R)\dots))$ on a relation R, determine the optimal (cheapest) order to evaluate them.
- The cost for evaluating a selection σ_{p_i} depends on
  - the number of tuples that need to be processed ($\rightarrow$ fewer tuples is better), and
  - the cost to evaluate the predicate p_i on one tuple.
- To determine the optimal order, we need a selectivity value for each subset of predicates $P' \subseteq P = \{p_1, \dots, p_z\}$.
- A subset of predicates P' corresponds to the logical conjunction $F_\beta(P') = \bigwedge_{p_i \in P'} p_i$. This respective selectivity is denoted by $\beta(P')$.
- Gamma sampling determines these β-selectivities indirectly using so-called γ-selectivities of logical minterms of the form

$$F_\gamma(P') = \bigwedge_{p_i \in P'} p_i \wedge \bigwedge_{p_i \notin P'} \neg p_i.$$

Recap: Gamma Sampling – Why Minterms?

$$F_\beta(P') = \bigwedge_{p_i \in P'} p_i \qquad F_\gamma(P') = \bigwedge_{p_i \in P'} p_i \wedge \bigwedge_{p_i \notin P'} \neg p_i$$

- In a minterm, each predicate appears exactly once, either positive or negative (negated). For z predicates, there are 2^z minterms.
- Two minterms X_1, X_2, $(X_1 \neq X_2)$ over the same z predicates are mutually exclusive, i. e., $X_1 \wedge X_2 \equiv$ false. – Why?

  Consequences

  - Each tuple from a relation R fulfills exactly one minterm from the set of all minterms.
  - The sum over all minterm selectivities must be 1.
- We can construct the β-selectivities from the γ-selectivities.
  Example: $P = \{p_1, p_2, p_3\}$, $P' = \{p_2, p_3\}$.

  $$\beta(\{p_2, p_3\}) = \gamma(\{p_1, p_2, p_3\}) + \gamma(\{\neg p_1, p_2, p_3\})$$

Task 3a

Which tuples fulfill which predicates?

$$P = \{p_1, p_2, p_3\} = \{Age > 27, Income > 30\,000, Weight < 75\}$$

ID	Age	Income	Weight	p_1	p_2	p_3
1	28	40,000	80	✓	✓	✗
2	30	55,000	50	✓	✓	✓
3	27	37,000	75	✗	✓	✗
4	40	60,000	60	✓	✓	✓
5	42	62,000	85	✓	✓	✗
6	22	15,000	55	✗	✗	✓
7	70	20,000	67	✓	✗	✓
8	50	80,000	57	✓	✓	✓
9	55	85,000	86	✓	✓	✗
10	33	42,000	58	✓	✓	✓

Task 3a

For each $P' \subseteq P$, compute the selectivity $\gamma(P')$.

$$\gamma = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.1 \\ 0.3 \\ 0.1 \\ 0.1 \\ 0.0 \\ 0.4 \end{pmatrix} \quad \begin{matrix} 000 \\ 100 \\ 010 \\ 110 \\ 001 \\ 101 \\ 011 \\ 111 \end{matrix} \quad \begin{matrix} \neg p_1 \wedge \neg p_2 \wedge \neg p_3 \\ p_1 \wedge \neg p_2 \wedge \neg p_3 \\ \neg p_1 \wedge p_2 \wedge \neg p_3 \\ p_1 \wedge p_2 \wedge \neg p_3 \\ \neg p_1 \wedge \neg p_2 \wedge p_3 \\ p_1 \wedge \neg p_2 \wedge p_3 \\ \neg p_1 \wedge p_2 \wedge p_3 \\ p_1 \wedge p_2 \wedge p_3 \end{matrix}$$

It holds: $\sum_{P' \subseteq P} \gamma(P') = 1$ ✓

Task 3b

Give the complete design matrix C that is associated with $|P| = 3$.

The complete design matrix can be found recursively by

$$C_{|P|} = \begin{pmatrix} C_{|P|-1} & C_{|P|-1} \\ 0 & C_{|P|-1} \end{pmatrix} \quad \text{and} \quad C_0 = \begin{pmatrix} 1 \end{pmatrix}.$$

$$C_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Task 3c

Compute $C\gamma$.

$$\beta = C\gamma = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.0 \\ 0.1 \\ 0.3 \\ 0.1 \\ 0.1 \\ 0.0 \\ 0.4 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.8 \\ 0.8 \\ 0.7 \\ 0.6 \\ 0.5 \\ 0.4 \\ 0.4 \end{pmatrix} \mathrel{\widehat{=}} \begin{pmatrix} \emptyset \\ p_1 \\ p_2 \\ p_1 \wedge p_2 \\ p_3 \\ p_1 \wedge p_3 \\ p_2 \wedge p_3 \\ p_1 \wedge p_2 \wedge p_3 \end{pmatrix}$$

Contents

Task 1

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

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 $R.A$.

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.

Task 1

c) Paper *"Sketches for Size of Join Estimation"* by Rusu and Dobra

- General overview on the topic of sketches for join size estimation.
- Two examples for AGMS and Fast-AGMS.
- Also, there are example implementations provided by the authors: faculty.ucmerced.edu/frusu/Projects/Sketches/sketches.html

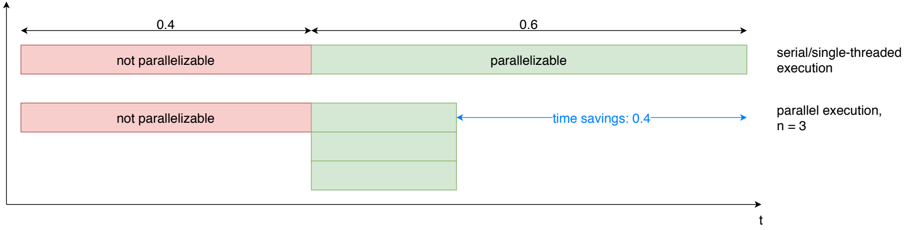
Task 2

Amdahl's law allows you to compute the theoretical speed-up of the execution of a task when using n processors/threads instead of 1:

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

- p denotes the fraction of the task that is parallelizable, i. e., this part of the program can be (evenly) divided into n parallel threads of execution.
- $1 - p$ is the part of the task that cannot be parallelized and can hence only be executed sequentially.
- The sequential and the parallelizable part (i. e., the whole task) sum up to 1: $p + (1 - p) = 1$.
- The formula's numerator represents the whole task if it is run without any parallelization.
- The denominator represents the fraction of the task that remains if p is parallelized.

Task 2



In the above example ($n = 3$, $p = 0.6$), the speedup is

$$\frac{1}{(1-0.6)+\frac{0.6}{3}} = \frac{5}{3} = 1.\overline{6}$$

i. e., parallelization is 1.6 times as fast as sequential execution.

Task 2

a) Compute the speed-up 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

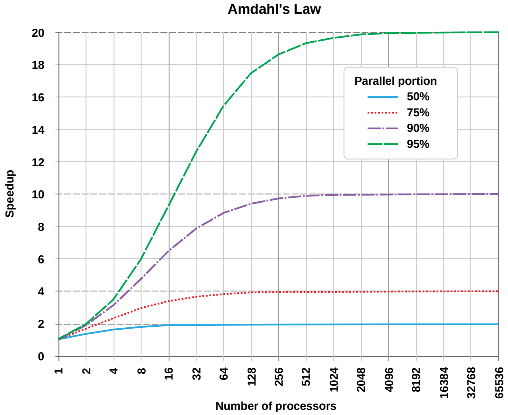
Task 2

b) What are the implicit assumptions of Amdahl's law?

1. **No Overhead**
   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.
2. **Arbitrarily Divisible Tasks**
   - 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.
   - **Example**: Consider the evaluation of a selection predicate on 1 000 000 tuples. Obviously, this can be parallelized since two arbitrary tuples are independent. However, $n > 1\,000\,000$ does not make sense as the $1\,000\,001^{\text{th}}$ thread would not be assigned any work.

Task 2

c) Sketch a diagram showing 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?



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

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

Task 2

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

- **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)

Task 3

How to Study for the DBS II Exam

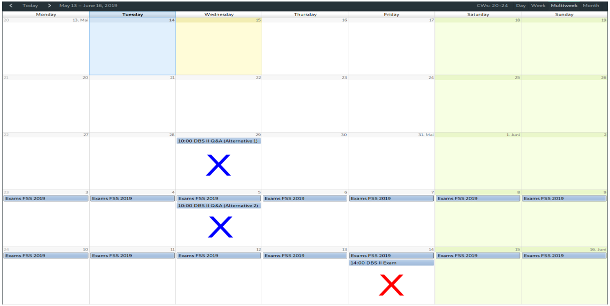
Recall the topics covered in the lecture:

1. Hardware
2. Hashing
3. Compression
4. Storage Layout
5. Physical Algebra: Processing Modes & Implementation
6. Expression Evaluation
7. Indexing
8. Boolean Expressions
9. Cardinality Estimation
10. Parallelism

Contents

Exam & Exam Preparation

- **Exam Date**: 14/06/2019, 14:00–15:30, room O 135 – might be subject to change, please check https://www2.uni-mannheim.de/studienbueros/pruefungen/pruefungstermine/ again!
- **Q&A Session**
  - 29/05/2019 (last week of lectures) *or* 05/06/2019 (first exam week), time: 13:45 as usual?



  - Please send me your questions beforehand so I can prepare an answer!