Seminar: Query Optimization

Organizational Meeting

Chair of Database Systems
Chair of Data Science & Engineering
Garching, July 11, 2019
Overview

Weekly Meeting

- Mondays, 16:00 - 18:00, presumably starting Nov 07, 2019
- Room MI 02.09.014
- 2 presentations per meeting
- There will be an attendance log

Required Work

- Seminar paper (8 pages)
- Sample implementation (C++)
- Presentation (20 minutes + 10 minutes discussion)
- Moderate one discussion (act as the "devil’s advocate", you should pair up for this)
Organization & Due Dates

Check in via email (radke@in.tum.de) or personally

1. Check in soon after matching for paper recommendations (preferences considered FCFS)
2. Check in when rough structure is planned
3. Check in when first draft is ready

Due Dates

- Structure: ca. 4 weeks prior to presentation date
- Presentation slides: 1 week prior to presentation date
- Seminar paper and sample implementation: 2 weeks after presentation date (strict!)
Topics - Join Ordering

- Greedy Operator Ordering [1, 2]
- IKKBZ [3, 4]
- Mixed Integer Linear Programming [5]
- Dynamic Programming [6, 7, 8]
- GroupBy Push Down [9, 10]
- Linearized DP [11, 12]
- Transformative Approaches [13]
- Iterative DP [14]
- Randomized Approaches [15, 16]
Topics - Cardinality Estimation

- Distinct Values [17, 18]
- Reservoir Sampling [19, 20, 21]
- Index-Based Join Sampling [22]
- Tighter Upper Bounds for Join Size Estimation [23]
- Learned Cardinalities [24, 25]
http://db.in.tum.de/teaching/ws1920/seminarAnfrageOpt

Bernhard Radke radke@in.tum.de

Have fun!
References

[1] Leonidas Fegaras.
A new heuristic for optimizing large queries.

Polynomial heuristics for query optimization.

On the optimal nesting order for computing n-relational joins.
References II

Optimization of nonrecursive queries.

[5] Immanuel Trummer and Christoph Koch.
Solving the join ordering problem via mixed integer linear programming.

Access path selection in a relational database management system.
References III

Dynamic programming strikes back.

On the correct and complete enumeration of the core search space.

Dynamic programming: The next step.
References IV


References V

    The complexity of transformation-based join enumeration.

    Iterative dynamic programming: a new class of query optimization algorithms.

    Join order selection - good enough is easy.
References VI

[16] César A. Galindo-Legaria, Arjan Pellenkoft, and Martin L. Kersten.
Uniformly-distributed random generation of join orders.

Towards estimation error guarantees for distinct values.

Every row counts: Combining sketches and sampling for accurate group-by result estimates.
References VII

Random sampling with a reservoir.

Adaptive-size reservoir sampling over data streams.

Scalable reservoir sampling on many-core cpus.
References VIII

Cardinality estimation done right: Index-based join sampling. 

Tighter upper bounds for join cardinality estimates. 

Learned cardinalities: Estimating correlated joins with deep learning. 
[25] Lucas Woltmann, Claudio Hartmann, Maik Thiele, Dirk Habich, and Wolfgang Lehner.
Cardinality estimation with local deep learning models.