• Summary
  - What we learned
  - What we didn’t learn
• Feedback
• Exams
• Advertisement
Contents

• Introduction
• Overview and architecture
• Storage and access methods
  – B*-Trees, Extensible hashing, index-sequential files …
  – Multidimensional indexing: Grid-files, kd-Trees, R-Trees …
• Query processing and optimization
  – Physical relational operators
  – Cost-based optimization
• Recovery
• Transactions and concurrency control
Price versus Access Time

- Really expensive
  - Register: 1-10 ns / byte

- Very expensive
  - Cache: 10-60 ns / cache line

- ~ 15 € / GB
  - Main Memory: 100-300 ns / block

- ~ 0.04 € / GB
  - Disk: 10-20 ms / block

- Tape: sec – min

RAID 5: RAID4, Verteilung der Parity

- Parity immer auf der Platte, die keinen der Blöcke enthält
- Wesentlich **bessere Lastbalancierung** als RAID4
  - Paritätsplatte kein Flaschenhals mehr
  - Schreiben erfordert X-1 Platten für die Daten und 1 weitere Platte für Parity
- Langsamer als RAID10, aber platzsparender
Oracle Block Structure

- **DBA**: Data Block Header: block address (global and relative in tablespace)
- Block type: data, index, redo, ...
- Table directory: tables in this block (for clustered data)
- Row directory: offset of tuples in block
- **ITL**: Interested transaction list – locks on rows in block
  - There is no „lock manager“ in Oracle
  - ITL grows and shrinks – “ITL wait”, INITTRANS, MAXTRANS
  - Locks are not cleaned upon TX end – next TX checks TX-ID
Inserting Values

Current content

- 40 = 101000
- 32 = 100000
- 18 = 010010
- 13 = 001101
- 12 = 001100
- 7 = 000111
- 6 = 000110
- 4 = 000100

- 000: 32, 40; t=3
- 001: 4, 12; t=3
- 01: 6, 18; t=2
- 00: 7, 13; t=1

INSERT(28)
- 28 = 011100
- h(28) = 001110

Overflow

d=t;
B-Trees

- B-Tree is a multi-level index with variable number of levels
- Height adapts to table growth / shrinkage
Example: 2D objects

- Objects are **points** in a 2D space
- Queries
  - Exact: All objects with coordinates \((X_1, Y_1)\)
  - Box: Find all points in a given rectangle
  - Partial: All points/rectangles with \(X\) (\(Y\)) coordinate between ...
Exemplary Workflow

SQL query → parse → parse tree → convert → logical query plan → estimate result sizes → "improved" l.q.p → consider physical plans → l.q.p. + sizes → \{P1, P2, \ldots\} → estimate costs → \{(P1, C1), (P2, C2), \ldots\} → execute → pick best → Pi → statistics / adaptation → answer → \{P1, P2, \ldots\}
Comparing Join Methods

Nested-Loops-Join

Merge-Join

Hash-Join
Correlated Subquery without Aggregation

\[
\text{SELECT} \ o.o\_id \\
\text{FROM} \ \text{order} \ o \\
\text{WHERE} \ o.o\_id \ \text{IN} \ (\text{SELECT} \ d.o\_id \\
\text{FROM} \ \text{delivery} \ d \\
\text{WHERE} \ d.o\_id = o.o\_id \ \text{AND} \\
\text{d.date-o.date<5})
\]

• Subquery materialization not possible
• **Naïve** computation requires one execution of subquery for each tuple of outer query
• Solution: **Rewrite into join**
  - Again: Caution with duplicates (if o:d is 1:n, DISTINCT required)

\[
\text{SELECT} \ \text{DISTINCT} \ o.o\_id \\
\text{FROM} \ \text{order} \ o, \ \text{delivery} \ d \\
\text{WHERE} \ o.o\_id = d.o\_id \ \text{AND} \\
\text{d.date-o.date<5}
\]
Joins and Projection/Selection

• Rule 6. Exchange of selection and join

  If \( \text{Cond} \) contains only attributes of \( E_1 \), then:

  \[ \sigma_{\text{Cond}}( E_1 \bowtie_{\text{Cond}_1} E_2 ) \equiv \sigma_{\text{Cond}}( E_1 ) \bowtie_{\text{Cond}_1} E_2 \]

• Rule 7. Exchange of selection and union/difference

  \[ \sigma_{\text{Cond}}( E_1 \cup E_2 ) \equiv \sigma_{\text{Cond}}( E_1 ) \cup \sigma_{\text{Cond}}( E_2 ) \]

  \[ \sigma_{\text{Cond}}( E_1 - E_2 ) \equiv \sigma_{\text{Cond}}( E_1 ) - \sigma_{\text{Cond}}( E_2 ) \]

• Rule 8. Exchange of selection and natural join

  \[ \sigma_{\text{Cond}}( E_1 \bowtie E_2 ) \equiv \sigma_{\text{Cond}}( E_1 ) \bowtie \sigma_{\text{Cond}}( E_2 ) \]
Equi-Depth

- Chose borders such that total frequency in each bucket is app. equal
  - Problem if one value is more frequent than \(|R|/|\text{buckets}|\) - use / combine with other types of histograms (later)
In the following, we talk of "objects"
- Usually means tuple
- Could also be block or attribute value (more later)
Relationships

- RC: Recoverable schedules
- ACA: Schedules avoiding cascading aborts
- ST: Strict schedules
  - Usually, we want strict schedules in databases
- SR: Serializable schedules
• Inspired by functional programming concepts *map* and *reduce*
• Operates on key-value pairs

**Map**

- Process key-value pairs individually → with UDF
- Generates key/value pairs
- Example (LISP):
  
  \[
  \text{(mapcar '1+ '(1 2 3 4)) } \Rightarrow (2 \ 3 \ 4 \ 5)
  \]

**Reduce**

- Merges intermediate key-value pairs with same key
- Example (LISP):
  
  \[
  \text{(reduce '+ '(1 2 3 4)) } \Rightarrow 10
  \]
Further Topics

• Distributed Databases, 2-phase-commit
• Parallel databases
• Special-purpose extensions (multimedia, spatial, text)
• Techniques for very large databases (DWH)
• Techniques for integrating heterogeneous data (Info Int)
• More complex data analysis (OLAP, data mining, analytics)
• Database programming and tuning
• XML, Xpath
• Object-oriented databases, OQL
• …
Bücher

• Vor allem

• Eher weniger
Werbung

• **Studien- und Diplomarbeiten**

• **Unsere Forschungsthemen**
  - Text Mining (in biomedizinischen Texten)
  - Verteilte Analyse sehr großer Datenmengen (BigData)
  - Statistische Analyse biomedizinischer Daten
  - Ihr Lieblingsthema (wenn es halbwegs zum Profil passt)

• **Oft interdisziplinär**
  - Charite, MPI‘s, FUB, Linguisten, Firmen, ...

• **Immer: Intensive Betreuung**
Diplom- / Magister/ Masterarbeiten

Anne Isbener: Similarity Search on Tabular Data (Exposé)
Diplomarbeit Informatik
März 2016 - Oktober 2016
Betreuung: Ulf Leser, Felix Naumann (HPI Potsdam)

David Salomon: Bestimmung der Recurrence Matrix für die Recurrence Quantification Analyse mittels Approximate Nearest Neighbour Search (Exposé)
Diplomarbeit Informatik
März 2016 - Oktober 2016
Betreuung: Ulf Leser, Norbert Marwan (GFZ Potsdam), Tobias Rawald

Tim Repke: Extraction of Citation Data from Websites based on Visual Cues (Exposé)
Masterarbeit Informatik
März 2016 - September 2016
Betreuung: Ulf Leser, Niels Pinkwart

Maurice Bleuel: Implementation and Evaluation of the TPC-DI Benchmark for Data Integration Systems (Exposé)
Masterarbeit Informatik
Januar 2016 - Juli 2016
Betreuung: Ulf Leser

Sascha Baese: Skalierbare Indexierung humaner Mutationsprofile durch Inverted Files (Exposé)
Diplomarbeit Informatik
November 2015 - August 2017
Betreuung: Ulf Leser, Stefan Sprenger, NN

Peter Moor: Development of a Mutation Panel for Neuroendocrine Tumor Research (Exposé)
Masterarbeit Bioinformatik (FU)
Dezember 2014 - Februar 2016
Betreuung: Ulf Leser, Heike Siebert (FUB)

Studien-/ Bachelorarbeiten

Monika Leung: Parallelization of a Bioinformatics application with a Workflow Language: A critical analysis of a parallel grid search optimization of the LIMMA algorithm based on the Cuneiform Workflow Language (Exposé)
Bachelorarbeit Informatik
May 2016 - September 2016
Betreuung: Ulf Leser, Raik Otto, Joachim Fischer

Jan-Niklas Rössler: Eine kritische, komparative Analyse von Methoden zur Untersuchung differentieller Genexpression (Exposé)
Bachelorarbeit Bioinformatik
May 2016 - August 2016
Betreuung: Ulf Leser, Raik Otto, Rosario Michael Piro (FUB)
Ausblick

• Sommer 2017
  - BA: Algorithmen und Datenstrukturen
  - BA: Einführung in die Bioinformatik (Starlinger)
  - BA: 2*Proseminar: Wissenschaftliches Arbeitens
  - MA: Seminar Text Classification