Course overview

Rasmus Pagh
The lectures at a glance

• SR: Tree Indexes.
• RP: Hash Indexes, Index Tuning
• SR: Data storage, external sorting, lower bound.
• SR: Implementation of relational operations
• RP: Query Optimization, Query tuning
• RP: Concurrency control
• SR: Spatial databases
• SR: Temporal databases
• SR: Text indexing
• RP: Decision support, OLAP
• Invited lecture
• RP: ITU research in databases
Tree indexes

- **B-trees**, a generalization of binary search trees, is the most important index type in DBMSs.
- You will get an understanding of what functionality B-trees offer, and how they are updated when the data changes.
- **Buffered B-trees**, a new B-tree variant that has exceptionally good update performance, is presented.
Hash indexes, index tuning

- External memory hash tables generalize hash tables as you know them.
- Faster than B-trees in some situations.
- Need to understand to choose!
- We will discuss general issues about how to choose the right indexes.
Data storage, sorting

• We consider how relations themselves are represented on disk
  – Sorted vs unsorted
  – How to cope with updates (variable size data)

• Sorting data on disk is an important primitive that we will need later on
  – External memory mergesort
  – Argument that this algorithm is best possible
Relational algebra operations

• The building blocks in DBMS query evaluation are algorithms that implement relational algebra operations.

• May be based on:
  – sorting,
  – hashing, or
  – using existing indexes

• The DBMS knows the characteristics of each approach, and attempts to use the best one in a given setting.
Query optimization, query tuning

- Query optimization is the process where the DBMS tries to find the “best possible” way of evaluating a given query.
- Standard approach builds on finding a “good” relational algebra expression and then choosing how and in what order the operations are to be executed.
- Query tuning is a “manual” effort to make query execution faster.
Concurrency control

- For databases with many users, the concurrency control mechanisms of a DBMS can cause performance problems.
- DBMSs are distinguished by their design of concurrency control system
  - Pessimistic (locking based) vs optimistic
  - Granularity
- To handle concurrency control problems, an understanding of the system in use is often required.
Spatial databases

• Many large databases contain geographical data.

• In general, many data sets can be viewed as points in a multi-dimensional space. **Example**: (salary, age) pairs.

• Need for efficient indexes that allow the DBMS to find part of the space. **Example**: “Find all tuples with age below 30 and salary above 500,000″.
Temporal databases

• It is increasingly feasible to never delete data (i.e., keep old versions)
• ⇒ Demand for capability to query old data.
• Need indexing capability also for old data!
• You will see a surprisingly efficient way of doing this.
Text indexing

• Many database applications contain lots of text
• ... but the relational model is not well suited to represent the structure of text.
• Result: Text datatype that may contain long strings that have to be handled in queries.
• We look at two topics:
  – B-trees optimized for strings
  – Full-text indexing
Decision support (OLAP)

- OLAP systems are specialized databases for decision support applications.
- Idea: Read-only (or write-rarely), optimized for fast answers to queries.
- Special indexing techniques for read-only data are used (bitmap indexing).
- Precomputation of aggregates important for performance.
Invited lecture

• We will invite an interesting person who has worked with database efficiency issues, to give an invited lecture.

• Name and affiliation to be announced.
ITU research in databases

• An overview of some results by ITU researchers on (or related to) performance aspects of databases.
• Mainly theoretical work - chance to be the first in the world to implement and test!
• Especially meant to serve as inspiration for formulating possible thesis projects.
The project

• Database development project.
• Use of the database will be simulated by a java program supplied to you.
• Your task:
  – Make a good database design.
  – Implement various query and update ops.
  – Tune for performance.
• More information on Tuesday...