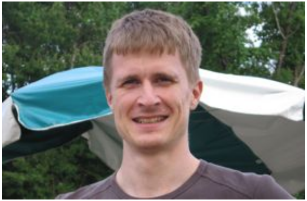
Data Processing on Modern Hardware

Jens Teubner, TU Dortmund, DBIS Group
jens.teubner@cs.tu-dortmund.de

Summer 2016

A Few Words About Me

Jens Teubner
DBIS Group (LS 6)
Otto-Hahn-Strasse 14, Room 333
`jens.teubner@cs.tu-dortmund.de`



1996–2001	Diploma in Physics, U Konstanz
2001–2005	Research assistant, DBIS Group, U Konstanz
2005–2007	Research assistant, Database Group, TU München
Oct 2006	PhD in Computer Science (XML query processing)
2007–2008	Postdoc, IBM T. J. Watson Research Center, NY, USA
2008–2013	Senior Researcher, Systems Group, ETH Zurich
since 4/2013	Full Professor, DBIS Group, TU Dortmund University

Topic: Database systems on modern computing hardware

A Motivating Example (Memory Access)

Task: sum up all entries in a two-dimensional array.

Alternative 1:

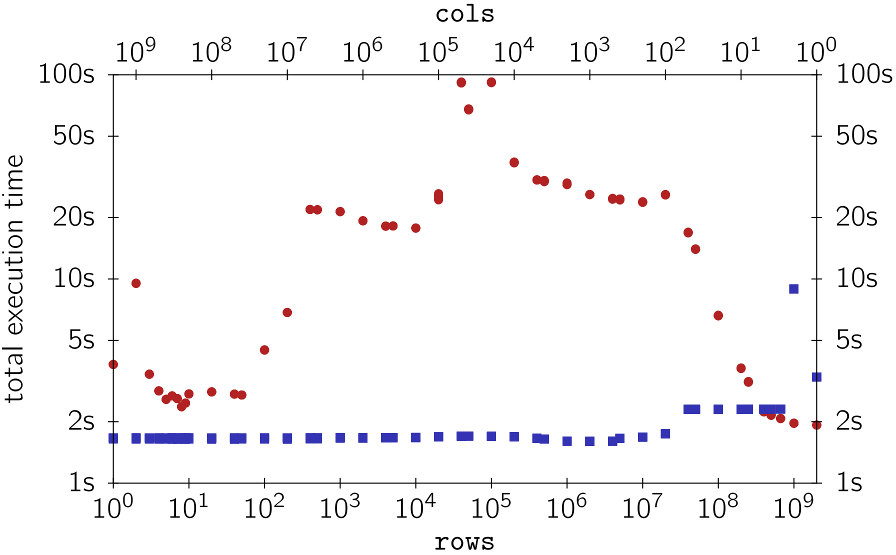
```
for (r = 0; r < rows; r++)
    for (c = 0; c < cols; c++)
        sum += src[r * cols + c];
```

Alternative 2:

```
for (c = 0; c < cols; c++)
    for (r = 0; r < rows; r++)
        sum += src[r * cols + c];
```

Both alternatives touch the same data, but in different order.

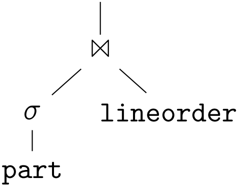
A Motivating Example (Memory Access)



A Motivating Example (Multi-Core)

Task: run parallel instances of the query

```
SELECT SUM(lo_revenue)
  FROM part, lineorder
 WHERE p_partkey = lo_partkey
   AND p_category <= 5
```



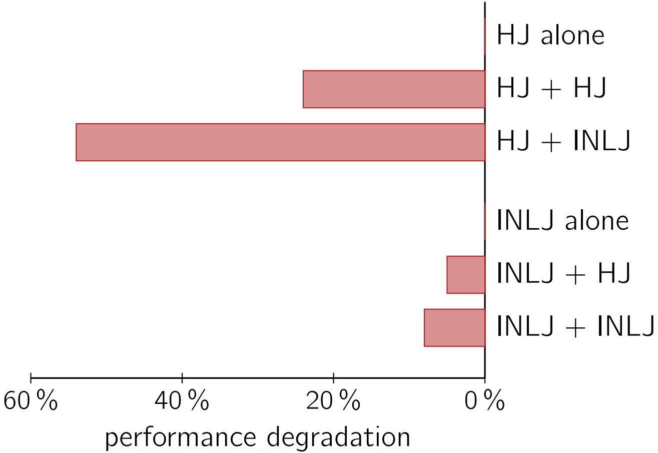
To implement $\bowtie$ use either

- a **hash join** or
- an **index nested loops join**.

Results taken from 'Lee, Ding, Chen, Lu, and Zhang. MCC-DB: Minimizing Cache Conflicts in Multi-core Processors for Databases. *VLDB 2009*.'

A Motivating Example (Multi-Core)

Co-run independent instances on different CPU cores.



Concurrent queries may seriously affect each other's performance.

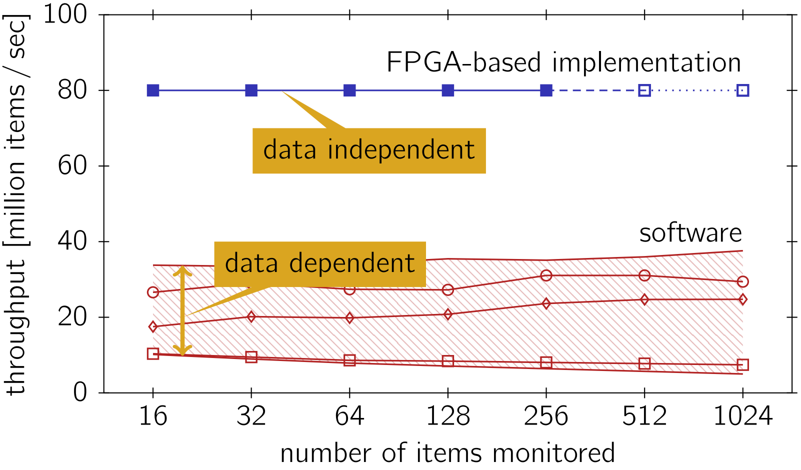
A Motivating Example (Non-Commodity Hardware)

Task: in a long stream of items, find those items that occur most often.

Algorithm *Space-Saving* (Metwally *et al.*, TODS, vol. 31(3), 2006):

```
foreach stream item $x \in S$ do
    find bin $b_x$ with $b_x$.item $= x$ ;
    if such a bin was found then
        $b_x$.count $\leftarrow b_x$.count $+ 1$ ;
    else
        $b_{\text{min}} \leftarrow$ bin with minimum count value ;
        $b_{\text{min}}$.count $\leftarrow b_{\text{min}}$.count $+ 1$ ;
        $b_{\text{min}}$.item $\leftarrow x$ ;
```

A Motivating Example (Non-Commodity Hardware)



Jens Teubner, René Müller, and Gustavo Alonso. FPGA Acceleration for the Frequent Item Problem. *ICDE 2010*.

Course Content

- Cache Awareness
    - How can we place data in memory and access it in a way that makes optimal use of memory caches?
- Query Execution
    - How can we tune our algorithms to fit modern processor architectures?
- Multi-Core Architectures
    - How can we exploit the parallelism provided by multi-core architectures?
- Specialized Hardware
    - How can we (mis-)use specialized hardware for data processing (*e.g.*, GPUs, FPGAs, modern NICs)?

Course Organization

Lecture:

- Mondays, 14–16h, OH 12, E.003
- Course website: `http://dbis.cs.tu-dortmund.de/cms/en/teaching/ss16/dp-new-hw/`
  Please visit this website **regularly**. We will frequently post new information during the semester.

Exercises: (also done by me)

- Mondays, 16–18h, OH 12, E.003
- First exercise: today
- Exercise material is part of the course content!

Exam

- Information about the exam will follow!

Course Setup

- I'd like to make this course highly **interactive**.
  - Please speak up, discuss, ask questions!
- The material we discuss is relevant in **practice**.
  - We'll provide practical examples and exercises.
  - You achieve maximum fun factor if you verify techniques on **your** machine.

Course Material

This is **not** a standard course (often even “bleeding edge”).

- There is **no real textbook** for this course.

  Computer architecture basics are covered in “Computer Architecture: A Quantitative Approach” by Hennessy and Patterson, though.

- I’ll make **lecture slides** available on the web.
- Most material is taken out of **research papers**.
  - I’ll give references to those papers.
  - These are all good and easy-to-read papers.
- The techniques that we discuss are related to classical database implementation techniques. If you’ve heard a course like “Architecture of DBMS” before, you migh recognize some ideas again.

Real Systems

MonetDB implements many of the techniques we'll talk about.

- MonetDB is open-source: `http://monetdb.cwi.nl/`
- Support for SQL and XQuery, multi-platform
- Numerous tools to look "under the hood" of MonetDB.
- Primary development: CWI Amsterdam
- Try it out yourself!

HyPer is another very successful "in-memory database."

- Developed originally at TU München
- Web site: `http://www.hyper-db.de/`
- Just acquired by Tableau Software

We want **You**

The topics that we discuss in this lecture are highly related to the topics of my research group.

- Our work is internationally very successful
    - *Avalanche* is a research project that I started at ETH Zurich and it is well-known by now in the research community.
- **You** could help us:
    - Bachelor/Master Theses, constribution as student assistant, etc.
- **Approach me** if you'd like to contribute to a vibrant project at the forefront of research.