

# Data Processing on Modern Hardware

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

Winter 2019/20

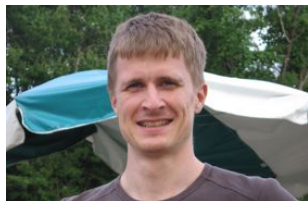
# A Few Words About Me

## Jens Teubner

DBIS Group (LS 6)

Otto-Hahn-Strasse 14, Room 333

[jens.teubner@cs.tu-dortmund.de](mailto: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:

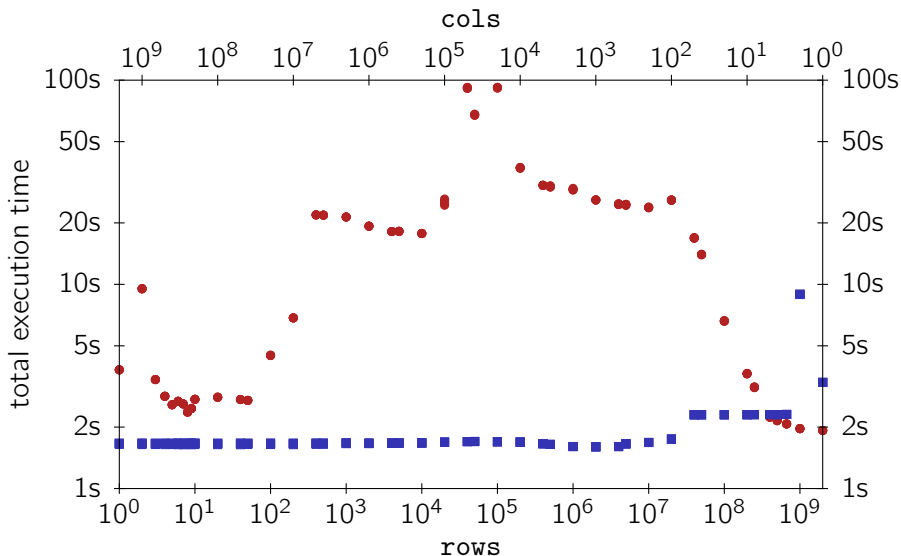
```
for (r = 0; r < rows; r++)  
    for (c = 0; c < cols; c++)  
        sum += src[r * cols + c];
```

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.

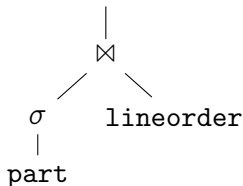
# A Motivating Example (Memory Access)



# 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
```



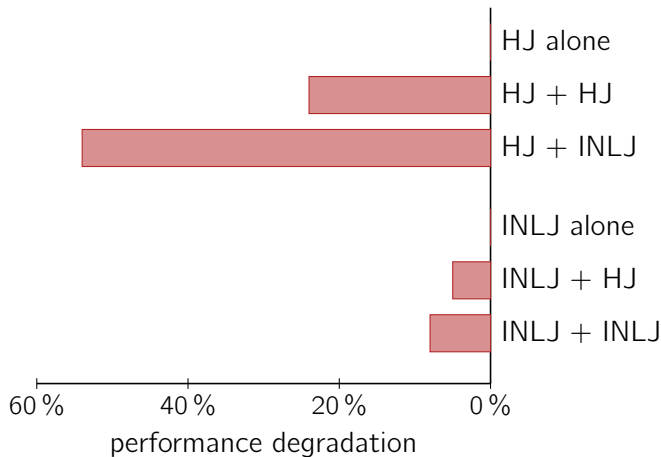
To implement  $\bowtie$  use either

- 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.

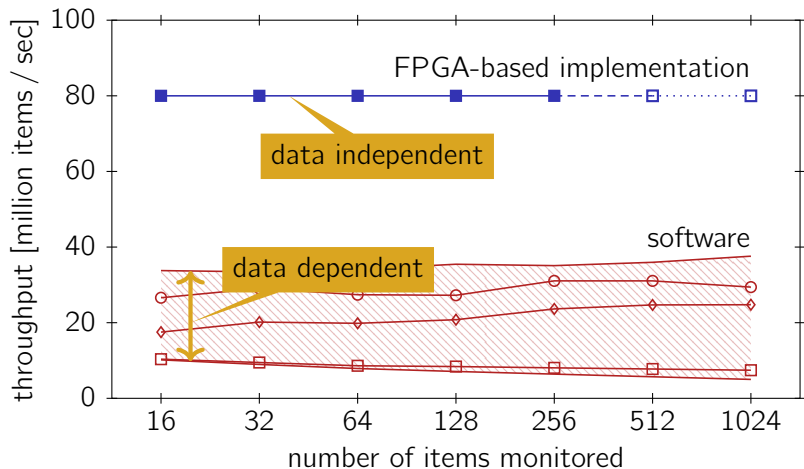
# A Motivating Example (Non-Commodity Hardware)

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):

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

# A Motivating Example (Non-Commodity Hardware)



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



- 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)?

## Lecture:

- Mondays, 14–16h, OH 12, E.003
- Course website: <http://dbis.cs.tu-dortmund.de/cms/en/teaching/ws1920/dp-new-hw/>  
Please visit this website **regularly**. We will frequently post new information during the semester.

## Exercises:

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

- Tuesday, February 4, 2020, 9:00h
- Wednesday, March 18, 2020, 11:00h
- Allowed material:
  - One sheet of DIN A4, handwritten (both sides)
  - Pocket calculator (but shouldn't be needed)

- 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.

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 might recognize some ideas again.

**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/>
- Meanwhile acquired by Tableau Software

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

- Our topics are highly **relevant to practice**
  - *E.g.*, SAP HANA is a very successful commercialization of many ideas that we'll discuss in this course.  
(And we collaborate, in fact, with the SAP HANA team.)
- **You** could help us:
  - Bachelor/Master Theses, contribution as student assistant, etc.
- **Approach me** if you'd like to contribute to a vibrant project at the forefront of research.