Scientific Project: Databases for Multi-dimensional Data, Genomics and modern Hardware

David Broneske, Gabriel Campero, Bala Gurumurthy, Andreas Meister, Marcus Pinnecke, Roman Zoun, Gunter Saake

October 13, 2017
Overview

- Concepts of this course
- Overview of project topics & forming project teams
- Course of action (milestones, presentations)
- How to perform literature research?
Overview

▶ Concepts of this course
▶ Overview of project topics & forming project teams
▶ Course of action (milestones, presentations)
▶ How to perform literature research?
▶ Further lectures:
  ▶ Academic writing (2-3 lectures)
Organization
Scientific Project: Course Grading

Bachelor

▸ **Module:** WPF FIN SMK (Schlüssel- und Methodenkompetenzen)

▸ 5 CP = 150h ⇒ 42h presence time (3 SWS) + 108h autonomous work

Master

▸ **Module:** Scientific Team Project

▸ 6 CP = 180h ⇒ 42h presence time (3 SWS) + 138h autonomous work

*Grade at the end of the course for the whole project team*
Scientific Project: Course Grading II

- Weighting the Grade:
  - Presentations: 30%,
  - Implementation: 30%,
  - Paper: 30%,
  - Soft Skills: 10%
- Binding registration: Second Milestone
Scientific Project: Prerequisite

▶ Successful programming test in C++/Java
▶ 1h theoretical test in a seminar room (data and place to be discussed)
▶ Half of the team members have to pass the test
▶ Topics:
# Scientific Project: Semester Plan

<table>
<thead>
<tr>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-I</td>
<td>30.10.2017</td>
<td><strong>31.10.2017</strong></td>
<td>01.11.2017</td>
<td>02.11.2017</td>
</tr>
<tr>
<td>Aca-I</td>
<td>06.11.2017</td>
<td>07.11.2017</td>
<td>08.11.2017</td>
<td>09.11.2017</td>
</tr>
<tr>
<td>Winter Break</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>08.01.2018</td>
<td>09.01.2018</td>
<td>10.01.2018</td>
<td>11.01.2018</td>
</tr>
<tr>
<td></td>
<td>15.01.2018</td>
<td>16.01.2018</td>
<td>17.01.2018</td>
<td>18.01.2018</td>
</tr>
<tr>
<td>MS Final</td>
<td>22.01.2018</td>
<td>23.01.2018</td>
<td>24.01.2018</td>
<td>25.01.2018</td>
</tr>
</tbody>
</table>
Scientific Project: Milestones

- Milestone I - Topic, schedule, and team presentation & first results of literature research
- Milestone II - Concept & additional literature research
- Milestone III - Implementation & evaluation setup
- Milestone IV - Final presentation (wrap-up + evaluation results)
Concepts & Content
Lecture & Presentation

▶ Time/Place: Friday, 9:00-11:00, G29 - room K058
▶ Lectures with content of course → all
▶ Presentation of main milestones (see time table)
  → each project team

Meetings (Exercise)

▶ Individual for each project team
▶ Time and room to be agreed in project teams!
▶ Presentation of all intermediate results/milestones (informal)
▶ Discussion, discussion, discussion . . .
Acquired skills, specific to research

- Performing literature research
- Understanding and structured reviewing of scientific work
- Autonomous, solution-based reasoning on research task (e.g., finding alternative solutions)
- How to ask? How to adapt a task (extend/reduce)?
- Academic writing
Objectives & Qualification (II)

Acquired skills, always needed

- Team management
- Project and time scheduling
- Presentation of results
- Flexibility regarding changing conditions
- Reasoning about solutions ("Why is this the best/not adequate...")
Deliveries

- 4 milestone presentations (*main milestones*)
- Each team member has to present at least once
- Reporting of (sub) milestones in exercises/meetings
- Written paper about literature research (technical report)
- Management report
- Prototypical implementation
Deliveries and Grading (I)

Technical Report

- Delivery of report at a given time (deadline)
- Quality/Quantity of literature research
- Number of pages
- Quality of paper structure and evaluation
- Own contribution
Deliveries and Grading (II)

Management Report

- Description of project realization (timeline, milestones)
- Separation of roles and contributions of single team members
- Meeting protocols
- Self-evaluation of member and group work (strengths, weaknesses)
Deliveries and Grading (III)

**Presentation & Discussion**
- Quality of scientific presentation (structure, references, time)
- Assessment regarding the content (e.g., results of particular milestones)
- Participation of discussion

**Organization**
- Strictness
- Communication (just-in-time answers, satisfying time constraints)
- Self-organization (Sharing tasks, internal reporting of current state-of-work, dealing with problems)
- Autonomous working
Task & Time Management

Task Management

- *Main milestones* have to be finished in time
- *(Sub) milestones* are less strict (but don’t be sloppy)
- Pre-defined work packages $\Rightarrow$ each project team
  - ...defines sub work packages
  - ...determines responsibilities for these packages
    (divide&conquer)

Time Management

- Planning of periods
- Regarding capacities and resources
- Considering other tasks and activities
- Reporting of delays immediately to project members !
Role Management

- Possible roles: team leader, developer, researcher, ...
- Work together vs. responsibilities: design, implementation, testing, writing, ...
- Delegate for important roles/work packages
- Assignment of (sub) tasks to role for each milestone
Topic & Project Teams

- Teams with 3 to 5 students (depends on the task)
- Most tasks can be chosen once
- **Projects**
  - Theoretical part
    - State of the art
    - New ideas
  - Practical part
    - Usually in C++ or Java
    - Prototypical implementation
    - Evaluation part
Topic 1 - Join-Order Optimization

Intro

- Join Order Optimization needed for efficient query processing
  → NP-hard problem
- For 3-5 Bachelor/Master Students

Task

- What are common techniques? (Top-Down-Approaches ...)
- What are used optimization within Top-Down-Approaches?
- Prototypical implementation using C++
- Tune algorithms to performance (e.g., using a profiler)
- Evaluate their performance and draw conclusions
- Compare to other algorithms
Topic 2 - GridFormation.FPM

Intro

▶ In the GridStore (our novel, in-house storage engine) a table is partitioned into grids. Grids can include replicated & non-consecutive parts of the table. They can also overlap.
▶ Collaborate with us in applying frequent pattern mining (FPM) to optimize data partitioning for the GridStore.
▶ For 3-5 Bachelor/Master Students

Your Task

▶ Literature Research: FPM, Data Partitioning, Flexible Storage Engines.
▶ Prototypical implementation and tuning of an FPM strategy for data partitioning. This prototype should incorporate novel features relevant to the use case (e.g. add biases towards some partitions, penalties for replication, or others).
▶ Experimental evaluation & suggestion of improvements.
Intro

- DBMSs should be able to use reliable workload forecasts, for optimizing their performance.
- Recurrent Neural Networks can produce such forecasts.
- What are the challenges and opportunities for integrating efficient RNN forecasting into a DBMS?
- For 3-5 Bachelor/Master Students

Your Task

- Literature research: Deep Learning for forecasting.
- Prototypical implementation and tuning of a RNN for forecasting database workloads, using TensorFlow.
- Evaluate the performance of the model (incl. GPU vs CPU), the quality of predictions and how a prototypical storage engine could benefit from them.
- Suggest improvements and outline limitations.
Topic 4 - SIMD-Accelerated Hash Based Aggregation

Intro

▶ Hashing can be used to perform grouped aggregation queries
▶ While probing, insertion could be necessary
▶ Can be optimized using SIMD
▶ For 3-5 Bachelor/Master students

We’ve got

▶ Framework for performing hash insertion while probing
▶ Cuckoo hashing and linear probing with SIMD

Your Task

▶ Literature Research: Find additional hashing techniques
▶ Understand working of these hashing techniques
▶ Invent a concept to perform SIMD on hashed aggregation
▶ Implementation of SIMD accelerated hash based aggregation
▶ Performance comparison of new hashing techniques with existing ones
Elf

- Multi-column selection predicates in DWH applications
- Overhead when scanning several columns
→ Elf: Multi-dimensional main-memory index structure\(^1\)
  - Tree structure with fixed search path depending on \# of columns
  - Prefix-redundancy elimination
  - Storage optimizations

\(^1\)www.elf.ovgu.de
Intro
- Cold data traversal for queries on a little amount of columns
- Worst case: Mono-column selection predicates
- For 3-5 Bachelor/Master students

We’ve got
- Elf implementation

Your Task
- Literature Research: Related index structures and cold data management
- Understanding of the Elf and its optimization concepts
- Implementation of Elfs for Mono-column selection predicates, Pointers into TID lists
- Performance evaluation of the variants
- Investigate ratio of storage overhead and performance gain
Topic 6 - SIMD for Elf

Intro

- Elf nodes similar to B-tree nodes
- Zeuch et al. introduced SIMD for B-tree
- For 3-5 Bachelor/Master students

We’ve got

- Elf implementation

Your Task

- Literature Research: SIMD for tree structures
- Understanding of the Elf and its optimization concepts
- Implementation of SIMD Elf and its performance evaluation
Intro

▶ Sorting is a data-intensive task
▶ Elf stores data already sorted → column order determines effectiveness
▶ For 3-5 Bachelor/Master students

We’ve got

▶ Elf implementation

Your Task

▶ Literature Research: Sorting queries on partially indexed data
▶ Understanding of the Elf and its optimization concepts
▶ Implementation of additional sortings for Elfs
▶ Performance evaluation compared to a sort from scratch
Finding your Team

Topics:

- Topic 1 - Join-Order Optimization
- Topic 2 - GridFormation.FPM
- Topic 3 - WFTF! - Workload Forecasting w/TensorFlow (&RNNs)
- Topic 4 - SIMD-Accelerated Hash Based Aggregation
- Topic 5 - Cold Data Avoidance for Elf
- Topic 6 - SIMD for Elf
- Topic 7 - Sort Queries in Elf
Literature Research
Efficient literature research requires

- Knowledge of *Where* to search
- Knowledge of *How* to search
- Finding adequate search terms
- Structured review of papers
- Knowledge of how to find information in papers
Where to Search (I)

- Different websites available that provide large literature databases

1. Google Scholar: http://scholar.google.de/
   - Key word and concrete paper search
   - Often, PDFs are provided

2. DBLP: http://www.informatik.uni-trier.de/~ley/db/
   - Search for keyword, conferences, journals, author(s)
   - BibTex and references to other websites

3. Citeseer: http://citeseerx.ist.psu.edu/about/site
   - keyword, fulltext, author, and title search
   - BibTex and (partially) PDFs are provided
Where to Search (II)

- Publisher sites are also a suitable target
- ACM Digital Library: http://portal.acm.org/dl.cfm
  - Keyword, author, conference/literature (proceedings), and title search
  - Bibtex, mostly PDFs and other information are provided
  - Similar to ACM, but only few PDFs
  - Extended access within university network
- Springer: http://www.springerlink.de/
  - Similar to previous
  - Extended access within university Network
- Further search possibilities: on author, research group or university sites
Some hints to not get lost in the jungle

- Use distinct keywords (*fingerprint vs. fingerprint data*)
- Keep keywords simple (at most three words)
- Otherwise, search for *whole* title
- Read abstract (and maybe introduction) ⇒ decision for relevance

First insights

- Read abstract, introduction and background/related work (coarse-grained) to
  - ... get a first idea of the approach
  - ... find other relevant papers
Information Retrieval

Finding the required information

- Read the paper carefully
- Omit formal parts/sections
- Try to classify (core idea, main characteristics) ⇒ develop classification/evaluation in mind
- Understand the big picture
- Make notes
- Do NOT translate each sentence
Finding your Team

Topics:

- Topic 1 - Join-Order Optimization
- Topic 2 - GridFormation.FPM
- Topic 3 - WFTF!- Workload Forecasting w/TensorFlow (&RNNs)
- Topic 4 - SIMD-Accelerated Hash Based Aggregation
- Topic 5 - Cold Data Avoidance for Elf
- Topic 6 - SIMD for Elf
- Topic 7 - Sort Queries in Elf

When do we meet for the programming test?