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

April 3, 2018
Overview

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

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

- **Module:** WPF FIN SMK (Schlüssel- und Methodenkompetenzen)
- 5 CP = 150h $\Rightarrow$ 42h presence time (3 SWS) + 108h autonomous work

**Master**

- **Module:** Scientific Team Project (Inf, IngInf, WIF, CV)
  - DKE: Methods 2 or Applications
  - DE: Interdisciplinary Team Project
- 6 CP = 180h $\Rightarrow$ 42h presence time (3 SWS) + 138h autonomous work

*Grade at the end of the course for the whole project team*
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:
  - Some language specifics
  - General program understanding
  - Control flow understanding
- You can take both tests and have to pass at least one!
<table>
<thead>
<tr>
<th></th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td>02.04.18</td>
<td>03.04.18</td>
<td>04.04.18</td>
<td>05.04.18</td>
<td>06.04.18</td>
</tr>
<tr>
<td></td>
<td>09.04.18</td>
<td>10.04.18</td>
<td>11.04.18</td>
<td>12.04.18</td>
<td>13.04.18</td>
</tr>
<tr>
<td></td>
<td>16.04.18</td>
<td>17.04.18</td>
<td>18.04.18</td>
<td>19.04.18</td>
<td>20.04.18</td>
</tr>
<tr>
<td><strong>MS-I</strong></td>
<td>23.04.18</td>
<td>24.04.18</td>
<td>25.04.18</td>
<td>26.04.18</td>
<td>27.04.18</td>
</tr>
<tr>
<td></td>
<td>30.04.18</td>
<td>01.05.18</td>
<td>02.05.18</td>
<td>03.05.18</td>
<td>04.05.18</td>
</tr>
<tr>
<td><strong>Aca-I</strong></td>
<td>07.05.18</td>
<td>08.05.18</td>
<td>09.05.18</td>
<td><strong>10.05.18</strong></td>
<td>11.05.18</td>
</tr>
<tr>
<td><strong>MS II</strong></td>
<td>14.05.18</td>
<td>15.05.18</td>
<td>16.05.18</td>
<td>17.05.18</td>
<td>18.05.18</td>
</tr>
<tr>
<td><strong>Aca-II</strong></td>
<td>21.05.18</td>
<td>22.05.18</td>
<td>23.05.18</td>
<td>24.05.18</td>
<td>25.05.18</td>
</tr>
<tr>
<td></td>
<td>28.05.18</td>
<td>29.05.18</td>
<td><strong>30.05.18</strong></td>
<td>31.05.18</td>
<td>01.06.18</td>
</tr>
<tr>
<td></td>
<td>04.06.18</td>
<td>05.06.18</td>
<td>06.06.18</td>
<td>07.06.18</td>
<td>08.06.18</td>
</tr>
<tr>
<td><strong>MS III</strong></td>
<td>11.06.18</td>
<td>12.06.18</td>
<td>13.06.18</td>
<td>14.06.18</td>
<td>15.06.18</td>
</tr>
<tr>
<td></td>
<td>18.06.18</td>
<td>19.06.18</td>
<td>20.06.18</td>
<td>21.06.18</td>
<td>22.06.18</td>
</tr>
<tr>
<td></td>
<td>25.06.18</td>
<td>26.06.18</td>
<td>27.06.18</td>
<td>28.06.18</td>
<td>29.06.18</td>
</tr>
<tr>
<td><strong>MS Final</strong></td>
<td>02.07.18</td>
<td><strong>03.07.18</strong></td>
<td>04.07.18</td>
<td>05.07.18</td>
<td>06.07.18</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, Meetings & Presentation**

*Lecture & Presentation*

- Time/Place: Tuesday, 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 . . .
Objectives & Qualification (I)

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
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...")
Progress of Course

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)
- 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 (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
Deliveries and Grading (IV)

▶ Grade consists of:
  ▶ Presentations: 30%,
  ▶ Implementation: 30%,
  ▶ Paper: 30%,
  ▶ Soft Skills: 10%

▶ Binding registration: Second Milestone
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 ⇒ 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
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
Intro

▶ Join Order Optimization needed for efficient query processing
  → NP-hard problem

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 - Applied GridFormation

Intro

- GridFormation is our developing concept for framing common physical design optimizations under a single reinforcement learning formulation. Currently, we use a Deep Q-Learning model.
- Collaborate with us in applying our solution to an existing DBMS.

Your Task

- Literature Research: Data Partitioning, Reinforcement Learning, Model Management.
- Prototypical implementation integrating our solution as an online solution for a DBMS.
- Experimental evaluation & novel contributions to our current design and implementation.
Topic 3 - Workload Forecasting w/RNNs

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?

Your Task

- Literature research: Deep Learning for forecasting.
- Prototypical implementations and tuning of an RNN for forecasting database workloads.
- Evaluate the performance of the model (possibly incl. GPU vs CPU), the quality of predictions and how a prototypical storage engine could benefit from them.
- Suggest improvements and outline limitations.
Intro
- Join and aggregate are compute intensive
- Use similar algorithm for processing
- Common functionalities can be shared

We’ve got
- OpenCL framework
- Data parallel processing primitives

Your Task
- Literature Research: Different join and grouped aggregation methods available
- Understanding data parallel primitives and their granularities
- Concepts for using primitives for join and grouped aggregation processing
- Implementation of your concept of primitive based execution
- Compare primitive execution with stand-alone algorithm
Intro

- Our graph database system GeckoDB comes with a powerful scriptable environment, the *shell*
- Modification and querying with Cypher-Like query language + scripting w/ GeckoScript
- Conditional execution is missing: IF *var* THEN *block* ELSE *block* END
- Loops are missing: WHILE *var* *block* END

We’ve got

- GeckoDB incl. system shell (variable-/stack based virtual machine)
- Working scripting language w/ (scoped) variables, functions, etc...
- Examples on function definitions, full access to sources (written in C)

Your Task

- **Implement** IF and WHILE statements for GeckoScript in C
- **Evaluate** *shell* incl. IF and WHILE on std. functions you have to write in GeckoScript (e.g., fibonacci) by comparing to 2+ languages of your choice (e.g. Phyton, JavaScript, LUA, ...)

Intro

- Our graph database system GeckoDB is built from ground-up to support modern hardware, e.g., massive parallelism.
- The heart of parallelism-support is a library that we developed which focuses on bulk-data operations, *BOLSTER*.
- Currently, threads are spawned on-demand which may lead to thousands of (short-living) threads.
- For 3-5 Bachelor/Master students.

We’ve got

- GeckoDB incl. system incl. example where "the pain" is.
- BOLSTER is ready to use and connected to the components, so everything is setup.

Your Task

- **Implement** a thread pool (or similar strategy) in C for BOLSTER to avoid short-living threads.
- **Evaluate** a given example w/ and w/o your extension.
Topic 7 - Cold Data Avoidance for Elf

Intro

- Cold data traversal for queries on a little amount of columns
- Worst case: Mono-column selection predicates

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
Intro

- Sorting is a data-intensive task
- Elf stores data already sorted \(\rightarrow\) column order determines effectiveness

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
Intro

- Elf nodes similar to B-tree nodes
- Zeuch et al. introduced SIMD for B-tree

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
Finding your Team

Topics:

- Topic 1 - Join-Order Optimization
- Topic 2 - Applied GridFormation
- Topic 3 - Workload Forecasting w/RNNs
- Topic 4 - Join and Grouped Aggregation Using Granular Operations
- Topic 5 - If and While in GeckoScript
- Topic 6 - Thread pool in GeckoDB/BOLSTER
- Topic 7 - Cold Data Avoidance for Elf
- Topic 8 - Sort Queries in Elf
- Topic 9 - SIMD for Elf
Literature Research
How to Perform 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
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
How to Search

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
**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 - Applied GridFormation
- Topic 3 - Workload Forecasting w/RNNs
- Topic 4 - Join and Grouped Aggregation Using Granular Operations
- Topic 5 - If and While in GeckoScript
- Topic 6 - Thread pool in GeckoDB/BOLSTER
- Topic 7 - Cold Data Avoidance for Elf
- Topic 8 - Sort Queries in Elf
- Topic 9 - SIMD for Elf

When do we meet for the programming test?