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

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

October 9, 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
Scientific Project: Modules

Bachelor

- **Module:** WPF FIN SMK (Schlüssel- und Methodenkompetenzen)
- 5 CP = 150h ⇒ 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 ⇒ 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!
# 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>29.10.2018</td>
<td>30.10.2018</td>
<td>31.10.2018</td>
<td>01.11.2018</td>
</tr>
<tr>
<td>Aca-I</td>
<td>05.11.2018</td>
<td>06.11.2018</td>
<td>07.11.2018</td>
<td>08.11.2018</td>
</tr>
<tr>
<td>Winter Break</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>07.01.2019</td>
<td>08.01.2019</td>
<td>09.01.2019</td>
<td>10.01.2019</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, 13:00-15:00, G22A - room 217
- 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..."")
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 (II)

**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 (III)

- 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 $\Rightarrow$ 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!
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

- Textbooks contain valuable expert knowledge about laws
- Contextual information extraction can be a tedious task
- Find a way of automating the current process

We’ve got

- Rule-based annotation process implementation in GATE
- Dbpedia and Stanford Core NLP plugins for GATE

Your Task

- Literature Research: Alternative implementation options for automation of rule-based annotation within or outside of GATE
- Understanding of the current workflow and its constraints
- Implementation of an automated alternative workflow to the current GATE implementation
- Assessment which techniques require adaptation or replacement
- Documentation of time taken by current workflow compared to alternative proposal
Intro

- GridFormation is our work-in-progress solution to optimizing physical design with reinforcement learning. We have several implementations using OpenAI Gym and TensorFlow-Slim.
- Collaborate with us in growing to a multi-agent setting, where agents with different workloads collaborate to reach a Nash equilibrium.

Your Task

- Literature research: data partitioning, multi-agent reinforcement learning (RL), deep RL, DQN, DPPO, MCTS (experimental).
- Prototypical implementation: adapt existing agents.
- Experimental evaluation, with Hyrise or Peloton, and analysis.
Topic 3 - JOB19- Join Order Bot 2019

Intro

- Join order optimization is an NP-hard problem, where approximate solutions are error-prone.
- Traditionally optimizers do not gain from experience.
- How well can (deep) reinforcement learning (RL) be applied, so optimizers improve with time and learn how to avoid computation?

Your Task

- Literature research: join order optimization, deep RL.
- Evaluate the performance of your solution using the JOB benchmark, compare with a dynamic programming approach.
- Suggest improvements and outline limitations.
Topic 4 - Don’t Bust the Memory Bank: Recycling Slots in our String Dictionary

Intro

▶ "Remove" operations are required in operational systems
▶ A lot of "Remove" operations leads to sparse vectors + bookkeeping
▶ Continuous elements are required

We’ve got

▶ String Dictionary Implementation
▶ Modularized frameworks and C data structure library
▶ Vector-based Implementation as "Ground Truth"

Your Task

▶ Implementation of Slot-Map-based Alternative
▶ Evaluation of As-Is and Slot-Map based Alternative (w.r.t. runtime, memory consumption)
▶ Feel free to plug-in own ideas and concepts
▶ Low-Level Implementation Skills in C (Experience is required!)
Intro

- Benchmarking database system under atomic operations and queries is reasonable
- Requires to have the same evaluation system at hand and systems to compare
- Driver-implementations for systems of different classes (Document Stores, Key-Value Stores, Disk-based Relational, and In-Memory Relational Systems) are required

We’ve got

- Benchmark interface to 12 queries + a few setup functions ("Driver API")
- Example drivers for "Dummy", "REDIS", ...

Your Task

- Implementation of 3-5 system drivers for our benchmark
- Writing setup environment scripts (install, prepare, remove,... scripts) that are system-dependent
- Execute and report what the benchmarks tell you about your systems
- Low-Level Implementation Skills in C (Experience is required!)
Topic 6 - One Predicate Many Queries: Optimizer

Intro

- OLAP queries have multiple conjunctive selection statements
- Naive optimizer executes the selection together for efficient processing
- This leads to re-running same selection predicate multiple times
- Optimizing multiple queries can reduce the number of executed predicates

We’ve got

- A nice framework for connecting OpenCL with C++
- A growing set of primitives - primitives are granular operations within each DB operation.

Your Task

- Literature Research: Multi query optimization and ”One Operator Many Query”
- Identifying necessary selection primitives variants
- Identifying the implementation method for the selection primitives
- Implementation of a kernel generator based on the input characteristic
- Develop and test the optimizer for efficient selection groups
Topic 6 - One Predicate Many Queries: Optimizer
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
Topic 7 - Cold Data Avoidance for Elf

Beispieldaten

<table>
<thead>
<tr>
<th>C_1</th>
<th>C_2</th>
<th>C_3</th>
<th>C_4</th>
<th>...</th>
<th>TID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>T_1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>T_2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>T_3</td>
</tr>
</tbody>
</table>

Performanzgewinne

Elf-Datenstruktur

Reorganisation

David Broneske et al. Scientific Project 27
Intro

- Elf stores all information of a table → storage structure
- Currently used for selections... what about joins?

We’ve got

- Elf implementation
- Ideas and code for a straight-forward join on Elf

Your Task

- Literature Research: Join implementations, sort-based joins
- Understanding of the Elf and its optimization concepts
- Implement a straight-forward join based on selections and an own native join algorithm
- Performance comparison between state-of-the-art joins (e.g., Radix Join) and your implementation
Finding your Team

Topics:

- Topic 1 - Information Extraction from Legal Textbooks
- Topic 2 - Collaborative Adaptive Layouts
- Topic 3 - JoinOrderBot 2019
- Topic 4 - Don’t Bust the Memory Bank: Recycling Slots in our String Dictionary
- Topic 5 - YCSB Sharp: Driver Implementation
- Topic 6 - One Predicate Many Queries: Optimizer
- Topic 7 - Cold Data Avoidance for Elf
- Topic 8 - Joins on Elf

When do we meet for the programming test?
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
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
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** - Information Extraction from Legal Textbooks
- **Topic 2** - Collaborative Adaptive Layouts
- **Topic 3** - JoinOrderBot 2019
- **Topic 4** - Don’t Bust the Memory Bank: Recycling Slots in our String Dictionary
- **Topic 5** - YCSB Sharp: Driver Implementation
- **Topic 6** - One Predicate Many Queries: Optimizer
- **Topic 7** - Cold Data Avoidance for Elf
- **Topic 8** - Joins on Elf