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

David Broneske, Andreas Meister, Marcus Pinnecke, Roman Zoun, Gunter Saake

April 4, 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 \(\Rightarrow\) 42h presence time (3 SWS) + 108h autonomous work

**Master**

- **Module:** Scientific 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: 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 for C++/JAVA
- 1h test (date/room will be discussed)
- Half of the team members have to pass the test
- Topic: Program comprehension and general programming guidelines
# 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><strong>Introduction</strong></td>
<td>03.04.17</td>
<td>04.04.17</td>
<td>05.04.17</td>
<td>06.04.17</td>
</tr>
<tr>
<td><strong>MS-I</strong></td>
<td>01.05.17</td>
<td>02.05.17</td>
<td>03.05.17</td>
<td>04.05.17</td>
</tr>
<tr>
<td><strong>Aca-I</strong></td>
<td>08.05.17</td>
<td>09.05.17</td>
<td>10.05.17</td>
<td>11.05.17</td>
</tr>
<tr>
<td><strong>Aca-II</strong></td>
<td>15.05.17</td>
<td>16.05.17</td>
<td>17.05.17</td>
<td>18.05.17</td>
</tr>
<tr>
<td><strong>MS II</strong></td>
<td>22.05.17</td>
<td>23.05.17</td>
<td>24.05.17</td>
<td>25.05.17</td>
</tr>
<tr>
<td></td>
<td>29.05.17</td>
<td>30.05.17</td>
<td>31.05.17</td>
<td>01.06.17</td>
</tr>
<tr>
<td></td>
<td>05.06.17</td>
<td>06.06.17</td>
<td>07.06.17</td>
<td>08.06.17</td>
</tr>
<tr>
<td><strong>MS III</strong></td>
<td>12.06.17</td>
<td>13.06.17</td>
<td>14.06.17</td>
<td>15.06.17</td>
</tr>
<tr>
<td></td>
<td>19.06.17</td>
<td>20.06.17</td>
<td>21.06.17</td>
<td>22.06.17</td>
</tr>
<tr>
<td></td>
<td>26.06.17</td>
<td>27.06.17</td>
<td>28.06.17</td>
<td>29.06.17</td>
</tr>
<tr>
<td><strong>MS Final</strong></td>
<td>03.07.17</td>
<td>04.07.17</td>
<td>05.07.17</td>
<td>06.07.17</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, G22A-128
- 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 . . .
The Idea . . .

Role-playing game . . . Imagine
You

We
The Idea . . .

Role-playing game . . . Imagine
You
  ▶ . . . are an upcoming project team
  ▶ . . . are searching for innovative DB solutions
  ▶ Research is your key to success
We
Role-playing game... Imagine

You

▸ ... are an upcoming project team
▸ ... are searching for innovative DB solutions
▸ Research is your key to success

We

▸ ... are the customer
▸ ... want the best solution that gives us a unique selling proposition
▸ ... want you to provide evidence of your scientific capabilities
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)
- 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
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 ⇒ 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)
- Every task can be chosen **once**

**Projects**
- **Theoretical part**
  - State of the art
  - New ideas
- **Practical part**
  - Usually in C++ / JAVA
  - Prototypical implementation
  - Evaluation part
Topic 1 - Join-Order Optimization

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

We’ve got
▶ Optimization Framework
▶ DP + Top-Down-Enumeration

Your Task
▶ Literature Research: Join-Order optimization
▶ Understanding of the Optimization Framework
▶ What are used optimization within Top-Down-Approaches?
▶ Prototypical implementation of optimization approach in C++
▶ Tune algorithms to performance (e.g., using a profiler)
▶ Evaluate their performance and draw conclusions
▶ Compare to other algorithms
Topic 2 - Cardinality Estimation

Intro

▶ Join-Order Optimization depends on cost-estimation
▶ Efficient query processing can only be ensured with good cost-estimation
▶ Cost-estimation needs at lease some cardinality estimations

We've got

▶ Optimization Framework
▶ DP + Top-Down-Enumeration

Your Task

▶ Literature Research: Cardinality Estimations (Top-Down-Approaches ...)
▶ Implement simple in-memory processing framework
▶ Integrate suitable benchmark for Join-order optimization
▶ Prototypical implementation of Cardinality Estim. in C++
▶ Tune algorithms to performance (e.g., using a profiler)
▶ Evaluate their performance and draw conclusions
Topic 3 - Graph Traversals in Relational DBSs

Intro

- Explorational queries important
- Backbone: Efficient single step traversals
- Latest idea: Integrate traversal operators into RDBMSs

Your Task

- Based on literature to study (see below): Implementation of GRAPHITE for evaluation purposes (in Modern C++)
- Understanding of the GRAPHITE concepts
- Evaluation of GRAPHITE on existing data sets with pre-defined queries
- Optional: Suggestion of optimization knobs

Literature to Study

Intro

- Tile-based architecture, recently proposed by Arulraj et al.
- Columns and rows are partitioned in tiles.
- Processing happens over logical tiles.
- Can tiles efficiently support non-relational data models?

Your Task

- Literature Research: Tile-based architectures, Non-relational data model of choice.
- Prototypical implementation of: a) table → tiles, b) tiles → non-relational model, c) at least one non-relational operator.
- Evaluate the transformation steps.
- Suggest improvements and outline limitations.
Topic 5 - Build your own Octopus (OctopusDB)

Intro

- OctopusDB, recently proposed by Dittrich and Jindal as a way to support HTAP.
- At the beginning there are no tables in this system, just a log.
- Table-like pieces (views) are created from the log, according to queries.
- Different types of views: Row, Column, Pax, Index.
- How to improve the implementation of such a system?

Your Task

- Literature research: OctopusDB, Database cracking.
- Prototypical implementation of: a) central log, b) log → views, c) processing over views and maintenance.
- Evaluate the creation and maintenance of views.
- Suggest improvements and outline limitations.
Topic 6 - Metaproteom Analysis Tools

Intro

▶ Proteomics is the research of a species in their proteins
▶ Metaproteomics is the proteomics of bio communities
▶ Protein search algorithms
▶ Find bottlencks and revolutionize the search task

We’ve got

▶ mass spectra data
▶ X!tandem c++ code
▶ Test environment

Your Task

▶ Literature Research: mass spectrometry, protein search
▶ Understanding X!tandem implementation and parameters
▶ Implementation evaluation steps of each component
▶ Evaluation the Components with different data Inputs
▶ Define an optimization approach for the algorithm and evaluate them
Elf

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

- Insert/Update/Delete are important Operations on an index structure
- Linearization hinders straight-forward approaches
- Instead of doing direct updates: build a second Elf
- Merge of two Elfs necessary

We’ve got

- Elf implementation

Your Task

- Literature Research: Merge Implementations + SIMD versions
- Understanding of the Elf and its optimization concepts
- Implementation of a scalar merge and a SIMD merge for Elf
- Performance evaluation of the variants
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 - Cardinality Estimation
- Topic 3 - Graph Traversals in Relational DBSs
- Topic 4 - Tiles and non-relational models
- Topic 5 - Build your own Octopus (OctopusDB)
- Topic 6 - Metaproteom Analysis Tools
- Topic 7 - (SIMD) Merge of several Elfs
- Topic 8 - SIMD for 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
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 - Cardinality Estimation
▶ Topic 3 - Graph Traversals in Relational DBSs
▶ Topic 4 - Tiles and non-relational models
▶ Topic 5 - Build your own Octopus (OctopusDB)
▶ Topic 6 - Metaproteom Analysis Tools
▶ Topic 7 - (SIMD) Merge of several Elfs
▶ Topic 8 - SIMD for Elf

When do we meet for the programming test?