Bachelor’s/Master’s Thesis Colloquia

Prof. Volker Markl
Chair of Database Systems and Information Management (DIMA)
Agenda

1. New DIMA Courses
2. Requisites, DIMA Thesis Process, Resources
3. DIMA Thesis Proposal/Exposé Template
4. DIMA Thesis Evaluation
5. Representative Thesis Topics
6. Questions from the Audience
1. New DIMA Courses

Bachelor’s Thesis Colloquium
Master’s Thesis Colloquium
1. **Initiated in April 2023** (SS 2023), to be offered each academic year (both WS and SS)

2. Desirable **Prerequisites**
   (a) Computer science topics addressed in the TUB Bachelor’s curriculum, particularly, both ISDA and DBPRA (or their equivalents)
   (b) good programming skills in C, Java, and SQL

3. This course complements and must be taken concurrently with the writing of a **Bachelor’s Thesis in Data Management Systems** at DIMA.
Description of Teaching and Learning Methods
1. This course consists of lectures on thesis topic selection, preparing a thesis proposal, effective execution of a thesis, and student led presentations.
2. Participating students will attend scheduled presentations and benefit both from the peers' comments on their work and from the discussion of the work of fellow students.
3. Active participation and contributions to all parts of this course are essential.

Test Description
1. This course is either pass or fail.
2. To pass this course, students must prepare and defend a DIMA thesis proposal and deliver a final presentation upon completion of their thesis.
3. Participating students will attend the final presentations of their peers.

Registration Procedures: Contact your DIMA thesis advisor.

Maximum Number of Participants: Determined by the mentoring capacity of DIMA.
Master’s Thesis Colloquium (MTC)

1. Initiated in April 2023 (SS 2023), to be offered each academic year (both WS and SS)

2. Desirable Prerequisites
   (a) Computer science topics addressed in the TUB Bachelor’s curriculum, particularly, both ISDA and DBPRA (or their equivalents)
   (b) good programming skills in C, Java, and SQL
   (c) knowledge of master's level coursework in database technology (DBT) and database systems implementation (DBTLAB) and advanced information management courses (e.g., BDSPRO, DMH, MDS, ROC, BDASEM, IMSEM)
   (d) fluency in English

3. This course complements and must be taken concurrently with the writing of a Master’s Thesis in Data Management Systems at DIMA.
MTC

Description of Teaching and Learning Methods
1. This course includes lectures on the thesis process at DIMA, thesis topic selection, preparing a thesis proposal, evaluation criteria of Master’s theses, open thesis topics at DIMA, and student presentations (e.g., proposal, defense).
2. Participating students will attend scheduled presentations and benefit both from the peers' comments on their work and from the discussion of the work of fellow students.
3. Active participation and contributions to all parts of this course are essential.

Test Description
1. This course is either pass or fail.
2. To pass this course, students must prepare and defend a DIMA thesis proposal and deliver a final presentation upon completion of their thesis.
3. Participating students will attend the final presentations of their peers.

Registration Procedures: Contact your DIMA thesis advisor.

Maximum Number of Participants: Determined by the mentoring capacity of DIMA.
2. Requisites, Thesis Process, Resources
<table>
<thead>
<tr>
<th>Course Offerings</th>
<th>Bachelor's Courses</th>
<th>Data Analytics Lab</th>
<th>Master's Courses</th>
<th>Thesis Opportunities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summer Semester 2023 Courses</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Archive of Past Semesters</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

https://www.tu.berlin/en/dima
Thesis Opportunities

Pursuing a Thesis with DIMA

Theses in DIMA are often tied to ongoing research projects sponsored by funding agencies and companies. These are commonly written in English (and some in German). Problems are typically centered on topics in database systems, scalable and distributed data management, and machine learning systems, including (i) query processing and optimization, (ii) storage, indexing, and physical database design, (iii) streams, sensor networks, and complex event processing, (iv) parallel and distributed databases, (v) databases for emerging hardware, (vi) benchmarking and performance evaluation, (vii) machine learning for data management, (viii) data management for machine learning, (ix) transaction processing, (x) database monitoring and tuning, (xi) data warehousing, OLAP, and SQL, Analytics, (xii) database security, privacy, and access control, (xiii) data visualization, (xiv) graph data management, RDF, and social networks, (xv) knowledge discovery, clustering, and data mining, (xvi) spatio-temporal databases, and (xvii) very large data science applications/pipelines.

To pursue a thesis with us, students are generally required to possess:

- outstanding programming skills in C++, Java, or Scala,
- extensive knowledge in database systems (e.g., IBM DB2, Oracle) or big data analytics systems (e.g., Flink, Spark),
- basic knowledge in the use of an IDE (e.g., Eclipse, IntelliJ),
- basic knowledge in the use of a distributed version control system (e.g., SVN, Git).

Furthermore, to conduct a:

- Bachelor’s thesis, students must have successfully completed ISDA and DBPRA (at a minimum) with a grade of good or better and possibly several other Bachelor’s courses offered by DIMA, such as DBPRO and DBSEM.
- Master’s thesis, students must have successfully completed DBT and DBTLAB (at a minimum) with a grade of good or better and possibly several other Master’s courses offered by DIMA, such as BDASEM, BDPBRO, DMH, IMSIM, MDS, and ROC.

Moreover, depending on the thesis topic, additional knowledge may be required (e.g., compiler technology, distributed systems, networking, operating systems, systems programming, machine learning).

Available Thesis Topics

If you are interested in our research areas and are currently in search of a thesis topic, we invite you to contact us. For more information, click on more below.

Completed Bachelor’s and Master’s Theses

Many students have successfully completed their Bachelor’s and Master’s Theses. For more information, click on ‘more’ below.

Pursuing a Thesis with DIMA

• theses are usually tied to ongoing research, commonly written in English
• topics in **database / ML systems, scalable / distributed data management**, including
  • query processing and optimization
  • storage, indexing, and physical database design
  • streams, sensor networks, and complex event processing
  • databases for emerging hardware
  • benchmarking and performance evaluation
  • ML for data management and data management for ML
  • database monitoring and tuning
  • database security, privacy, and access control
  • data visualization
  • graph data management, RDF, and social networks
  • knowledge discovery, clustering, and data mining
To pursue a thesis with us, students should generally possess...

1. *outstanding programming skills* in C++, Java, or Scala

2. *extensive knowledge in database systems* (e.g., DB2, Oracle) or *big data analytics systems* (e.g., Flink, Spark)

3. *basic knowledge in the use of an IDE* (e.g., Eclipse, IntelliJ)

4. *basic knowledge in the use of a distributed version control system* (e.g., SVN, Git)

Moreover, successfully completed DIMA courses *(with a grade of good or better)*

- Bachelor’s Thesis students: at least **ISDA** and **DBPRA**, ideally **DBPRO** and **DBSEM**
- Master’s Thesis students: at least **DBT** and **DBTLAB**, ideally **BDSPRO**, **DMH**, **MDS**, **ROC**, **BDASEM** and **IMSEM**
Further (Topic Specific) Requirements

- Moreover, depending on the thesis topic, additional knowledge may be required
  - systems programming
  - compiler technology
  - distributed systems
  - networking
  - operating systems
  - machine learning
Thesis Process

1. Contact prospective advisor directly via email or in person

2. Based on capacity the advisor will meet with the student/discuss the topic

3. Upon mutual agreement, the student will prepare a proposal/run it by the advisor

4. Once complete, the student will present the proposal in the colloquium.

5. This will then be either accepted as is or require revision.

6. Once approved by Juan Soto, then the student can proceed to register the thesis.
Note: The DIMA Thesis Related Webpages will be revised in the coming months!

Thesis Process

Database Systems and Information Management

You are here: Database Systems and Information Management » Teaching » Thesis Opportunities » Available Topics

General Instructions

1. To get an overview of our currently available topics, download the document Thesis Opportunities at DIMA.

2. Students are encouraged to contact DIMA Senior Researchers, Postdocs, and Ph.D. Students (Research Associates) directly via email to learn more about their proposed thesis topics. Many of our thesis advisors are already working with several students, and it may be that they cannot currently take on another student. For this reason, you will need to inquire about their availability. Furthermore, when contacting a team member, be sure to submit your current CV and the completed Thesis Request Form.

3. Once you have identified an advisor(s) who is willing and able to advise you on a thesis topic, you can start writing your thesis proposal (see Thesis Proposal Template). After finishing your writing, your advisor(s) will review and accept the proposal before they forward it to Juan Soto for a final review.

4. Upon final acceptance, students can proceed to register their thesis with the responsible examination office. Note: The proposal need not be forwarded to the examination office and students are advised to track the registration step to ensure it has officially been processed in a timely manner (e.g., at most a few weeks).

Further Templates:

- For guidance on a representative example of a thesis outline, take a look at this document: Representative Thesis Outline.
- For those seeking a sample thesis template in LaTeX, we provide the following template: Latex Thesis Sample Template.

Resources

General Instructions

1. To get an overview of our currently available topics, download the document Thesis Opportunities at DIMA.

2. Students are encouraged to contact DIMA Senior Researchers, Postdocs, and Ph.D. Students (Research Associates) directly via email to learn more about their proposed thesis topics. Many of our thesis advisors are already working with several students, and it may be that they cannot currently take on another student. For this reason, you will need to inquire about their availability. Furthermore, when contacting a team member, be sure to submit your current CV and the completed Thesis Request Form.

3. Once you have identified an advisor(s) who is willing and able to advise you on a thesis topic, you can start writing your thesis proposal (see Thesis Proposal Template). After finishing your writing, your advisor(s) will review and accept the proposal before they forward it to Juan Soto for a final review.

4. Upon final acceptance, students can proceed to register their thesis with the responsible examination office. Note: The proposal need not be forwarded to the examination office and students are advised to track the registration step to ensure it has officially been processed in a timely manner (e.g., at most a few weeks).

Further Templates:

- For guidance on a representative example of a thesis outline, take a look at this document: Representative Thesis Outline.
- For those seeking a sample thesis template in LaTeX, we provide the following template: Latex Thesis Sample Template.

3. DIMA Thesis Proposal/Exposé Template
Elements of a Thesis Proposal

0. Preface
1. Introduction / Scientific Background / Related Work
2. Goal of the Thesis / Statement of the Research Problem
3. Thesis Approach
4. Implementation Plan and Timeframe
5. Bibliography
# Preface

**Database Systems and Information Management Group**  
Fak. IV Electrical Engineering and Computer Science  
Technische Universität Berlin

**Thesis Proposal**

**Instructions to Students**

(a) Complete the proposal below and ask your primary advisor to review it for completeness.
(b) Once it has been accepted, your primary advisor should forward it to Juan Soto for final approval.
(c) The proposal length **must not exceed** five pages, including the cover and bibliography pages.

<table>
<thead>
<tr>
<th>Thesis Type</th>
<th>Bachelor’s or Master’s</th>
<th>ECTS</th>
<th>12 or 30</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Student</strong></td>
<td>First Name Surname(s)</td>
<td>Mat. Number</td>
<td>123456</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:fname.lname@campus.tu-berlin.de">fname.lname@campus.tu-berlin.de</a></td>
<td>Primary Advisor</td>
<td>First Name Surname(s)</td>
</tr>
<tr>
<td>1st Examiner</td>
<td>Prof. Dr. Volker Markl</td>
<td>2nd Examiner (e.g., Prof. Dr. Odej Kao, Prof. Dr. Matthias Böhm)</td>
<td></td>
</tr>
<tr>
<td>Academic Program</td>
<td>(e.g., Computer Science, Information Systems Management)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thesis Title</td>
<td>(e.g., Towards Continuous Query Processing for Mobile IoT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duration Period</td>
<td>From MM/YYYY to MM/YYYY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Specify the *obstacle* to be overcome

1.2 Specify the *motivation* (Why is this problem interesting?)

1.3 Specify the *novelty* (Was this problem already solved?)

1.4 Specify the *anticipated impact* (how does solving this problem impact our world?)

1.5 Specify the *anticipated contributions* (journal publication? open-source software?)

1.6 Specify the *scientific background* (or foundational work) that your solution will build upon

1.7 Specify the *related work* (e.g., competing solutions corresponding to the research challenge)
2. Goal of the Thesis

2.1 State the overarching goal that you aim to achieve in your thesis.

...  

2.2 Provide a succinct, precise, and unambiguous statement of the research problem(s) or question to be solved, in order to attain the goal.

...  

2.3 Specification of the scope of the thesis.

2.3.1 State what is in scope, particularly, the subproblems that will be explored.

...  

2.3.2 State what is out of scope.

...
3. Thesis Approach

3.1 Describe the solution approach (e.g., algorithms, data, evaluation metrics, software, systems).

3.2 How does the proposed solution differ from the state-of-the-art?

3.3 How will you know if you have *succeeded* in attaining your goal, i.e., solved the research problem(s)?

3.3.1 How will you measure the *effectiveness* of your solution? (e.g., formal proof of correctness, validation through test cases where the correct results are known)

3.3.2 How and against which baseline(s) will you measure the *efficiency* of your solution? (e.g., improved scalability: higher throughput over a varying number of cores, when compared to approach XYZ, improved latency: lower latency across different workload and database sizes, when compared to approach XYZ)
4. Implementation Plan and Timeline

An *implementation plan* consists of 3 major steps corresponding to 5 content-based milestones (MS1 – MS5).

**Step 1. [Related Work]**
The identification and discussion (MS1) of relevant resources (e.g., books, papers, software).

**Step 2. [Scientific and Technical Contribution]**
The design (MS2) and implementation (MS3) of relevant algorithms and system components in a prototype.

**Step 3. [Demonstration of the Effectiveness and Efficiency of the Proposed Solution]**
The evaluation of the proposed solution via experiments (MS4) and/or analysis and the discussion and the interpretation of the findings (MS5).
## Customized Implementation Plan and Timeline

<table>
<thead>
<tr>
<th>Dates</th>
<th>Milestone</th>
<th>Tasks</th>
<th>Deliverable(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>??-?? - ??-??</td>
<td>MS1</td>
<td>1. conduct literature review&lt;br&gt;2. read through the identified papers</td>
<td>list of relevant papers&lt;br&gt;reading notes</td>
</tr>
<tr>
<td>??-??</td>
<td>MS2</td>
<td>3. gain familiarity with <em>technologies</em> or <em>systems</em>&lt;br&gt;4. design an algorithm/protocol</td>
<td>...</td>
</tr>
<tr>
<td>??-??</td>
<td>MS3</td>
<td>5. implement algorithm/protocol&lt;br&gt;6. integrate it into initial prototype/system&lt;br&gt;7. validate prototype/system effectiveness</td>
<td>...</td>
</tr>
<tr>
<td>??-??</td>
<td>MS4</td>
<td>8. define data, workload(s), evaluation metrics&lt;br&gt;9. design experiments to be conducted&lt;br&gt;10. conduct experiments</td>
<td>...</td>
</tr>
<tr>
<td>??-??</td>
<td>MS5</td>
<td>11. analyze/interpret results (explain anomalies)</td>
<td>...</td>
</tr>
<tr>
<td>??-??</td>
<td>TC</td>
<td>12. complete thesis writeup</td>
<td>submit thesis by the due date¹</td>
</tr>
<tr>
<td>??-??</td>
<td>SC</td>
<td>13. finalize prototype/system</td>
<td>publish code in an open repository</td>
</tr>
</tbody>
</table>
Acknowledgment and 5. Bibliography

Preparation for the Thesis Defense

*Students should sign and date that they have read the instructions below.*

<table>
<thead>
<tr>
<th>Date</th>
<th>Tasks</th>
<th>General Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>post thesis submission</td>
<td>14. prepare presentation slides</td>
<td>send slides (as a .pdf file) to advisor(s) and Juan, no later than 8:00 a.m. (GMT+1), the day of the scheduled defense</td>
</tr>
<tr>
<td>defense date</td>
<td>15. deliver the presentation</td>
<td>Time limited: 15’ for B.Sc. and 20’ for M.Sc.</td>
</tr>
<tr>
<td></td>
<td>16. answer raised questions</td>
<td>ca. 20’ for Q&amp;A</td>
</tr>
</tbody>
</table>

Signature: ___________________________  Date: ___________________

5. Bibliography

*Instructions.* Use the [ACM Bibliography Style](https://www.bibtex.com/s/bibliography-style-base-acm/) to list the set of reference sources you drew on to prepare your thesis proposal.

...
4. DIMA Thesis Evaluation
Evaluation Criteria

- **Form of the Thesis**
  - Organization
  - Appearance
  - Clarity and Precision
  - Grammar Usage

- **Context**
  - Motivation
  - Literature Review
  - Related Work
  - Future Research

- **Contributions**
  - Novelty
  - Execution
  - Breadth of the Research
  - Depth of the Research

- **Defense**
  - Oral Presentation
  - Q & A
Form of the Thesis

- length (B.Sc. 40-50, M.Sc. 70-100)
- appropriate thesis structure
- proper use of English grammar
- employs precise language
- proper use of the active voice
- proper citation of authors/papers
- technical terms defined

Context

- motivation discussed clearly
- literature review is appropriate
- background / foundations
- discussion about related work
- discussion about state-of-the-art
- student’s contributions
- future research
Contributions

Discussion of the Prep. / Setup

- research problem stated clearly
- evaluation metrics selection
- experimental setup
- experiment design

Discussion of the Solution

- arguments thoroughly developed
- identification/description of data
- research methodology approach
- algorithms/systems employed
- parameter settings specification
- empirical experiments conducted
- algorithms / systems performance
- interpretation/limitations/generalizability of results
Contributions

Follows Good Scientific Practices

• correct usage of statistical graphics
• diverse usage of statistical graphics
• specify repository w/ relevant artifacts
• evidence of scientific innovation
• sound strategy when solving the research problem

Thesis Execution / Management

• independence in the execution
• independence in the writing
• planning and meeting deadlines
• appropriate levels of communication
Defense

• **structure**/organization/logic (flow)
• **content** (technical terms defined, key elements presented)
• **quality of slides** (layout, font, use of figures)
• **presentation** (language use, thesis well laid out, findings/argumentation appropriate)
• **timing** (B.Sc. 15’, M.Sc. 20’)
• presentation pace appropriate (neither too fast, nor too slow)
• understood the questions raised during the defense
• correctly answered the questions raised during the defense
## Thesis Grade

<table>
<thead>
<tr>
<th>Thesis Grade</th>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Excellent</strong></td>
<td>≥ 95%</td>
<td>outstanding performance</td>
</tr>
<tr>
<td><strong>Very Good</strong></td>
<td>≥ 90%</td>
<td>very good performance</td>
</tr>
<tr>
<td><strong>Good</strong></td>
<td>≥ 85%</td>
<td>above average performance</td>
</tr>
<tr>
<td><strong>Satisfactory</strong></td>
<td>≥ 70%</td>
<td>average performance, complies with overall requirements</td>
</tr>
<tr>
<td><strong>Sufficient</strong></td>
<td>≥ 65%</td>
<td>performance which, despite some flaws, still complies with requirements</td>
</tr>
<tr>
<td><strong>Fail</strong></td>
<td>&lt; 50%</td>
<td>performance with significant flaws, does not comply with requirements</td>
</tr>
</tbody>
</table>
5. Representative Thesis Topics at DIMA

Open Topics / Research Directions
<table>
<thead>
<tr>
<th>Member (No Slides)</th>
<th>Member (w/ Slides)</th>
<th>Member (w/ Slides)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. Ariane Ziehn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problems Arising in Distributed Data Streaming Systems

Bachelor’s Topics
1. eHealth scenario for IoTropolis
2. traffic/public transport scenario for IoTropolis

Master’s Topics
1. efficiently combining batch/streaming data
2. efficiently joining data across data management systems
3. improving *Parquet* for the management of time series
Graph Data Management

- Graphs are evolving in nature
  - Dynamic
  - Streaming

- Potential Problems
  - Subgraph Counting
    - e.g. triangles, butterflies
  - Community Detection

How to incrementally refine our results when the graph is updated instead of computing them from scratch?
PolyDB: Efficient & Interoperable Cross-DB Query Processing

• Data scientists analyze disparate data sets
• Challenges: multiple interfaces, excessive data movement, limited DB interoperability
• XDB is a decentralized cross-database query processor with a lightweight mediator

Open Issues in PolyDB

1) Accelerating data science pipelines on disparate data sets
   Explore: Integration of data science frontends and optimization opportunities for cross-db environments
   See: “P2D: A Transpiler Framework for Optimizing Data Science Pipelines”. In DEEM@SIGMOD’23

2) Investigating cross-database optimization
   Explore: Cost-based, heuristic, and multi-objective optimization for cross-db environments
   See: “In-Situ Cross-Database Query Processing”. In ICDE’23

3) Composing modular data management systems
   Explore: Composability mechanisms to “plug-and-play” existing db components
   See: “Towards a Modular Data Management System Framework”. In CDMS@VLDB’22

… and many more, feel free to reach out!
Motivation: frequent transient and fail-stop failures

The unified environment:
- Is highly dynamic
- Encapsulate unreliable devices with limited resources

Research directions:
- Adaptivity in state preservation
- Unification of existing solutions
- Pathfinding
- Graceful failure handling
**Research Topic:** Efficient execution of stream processing workloads on heterogeneous hardware.

**General prerequisites:**
- Knowledge in C/C++ programming.
- Knowledge in GPUs.

**Goal:** Execute queries with machine learning inference on NebulaStream using GPUs.

**Underlying Problem:** How to efficiently integrate the GPU ML inference operator with NebulaStream’s query compiler?

---

**Master’s Thesis Topic:** Investigate Efficient Data Ingestion to GPUs

**Goal:** Processing Streaming Workload at a High Throughput on GPUs while Keeping Latency in Check.

**Underlying Problem:** Indirect data path by staging the incoming stream to the host’s main memory is inefficient. How to make the communication as direct as possible?

---

**Contact information:**
Dwi Nugroho
d.nugroho@tu-berlin.de
**Goal:** Execute queries with ML inference on NebulaStream using GPUs.

**Research Questions:**

1. *How to integrate the GPU ML inference operator with NebulaStream’s query compiler?*
2. *How can we leverage existing functionalities in libraries such as TensorFlow Lite?*
3. *How does the solution performs, i.e., in terms of throughput and latency?*
Investigate Efficient Data Ingestion to GPUs

**Goal:** Processing Streaming Workload at a **High Throughput** on GPUs while Keeping **Latency in Check**.

**Underlying Problem:** Indirect data path by staging the incoming stream to the host’s main memory is inefficient.

**Research Questions:**
1. To what extent does the impact of ingestion give to the overall SPE performance? What are the impacted use cases?
2. How to allow communication between the network card and GPU to be as direct as possible? What are the alternatives?
3. How does the integration with a GPU-Accelerated SPE look like?

**Status Quo:** Stage the ingested event streams in main memory.
Adaptive Stream Join Processing

- In-memory database systems [1-3]
- Memory wall ⇒ CPU performance grown faster than main-memory performance
- Join order enormously impacts the database performance [4]

**Goal of thesis:**
Develop and investigate adaptive join algorithm(s)
- Higher performance
- More resource efficient

Investigating ASP on FPGAs

- Exponential increase of data per year and numerous applications with high-velocity data streams
- Solution: **Approximate Stream Processing**
- FPGA as accelerators [7, 10]

**Goal of thesis:**
Integrate FPGAs into Stream Processing Engine, e.g., NebulaStream [5]


Security and privacy (S&P) are becoming key components of modern cloud DBMSes

We investigate how to provide S&P guarantees in the public cloud databases using Trusted Execution Environments (TEEs)

TEEs impose many limitations on the developers and introduce new performance bottlenecks

This area is at the intersection of systems security, query processing, and modern hardware

Example thesis projects (BSc and MSc):
- Development of relational algorithms for TEEs
- Combining encrypted query processing with TEEs
- Designing efficient and secure data exchange protocols for distributed DBMS using TEEs

Requirements: good skills in C/C++, experience with databases, interest in systems security
Data Management Challenges in Federated Learning

• Federated learning (FL) enables machine learning across multiple parties without data sharing

• Large potential for cross-organizational collaboration
  • (e.g., anti money laundering, supply chain management)

• Many open (data management) research questions remain
  • When is cross-silo FL worth it in the first place?
  • How to discover other interested parties?
  • How to create consistent participation incentives?
  • How to confidentially integrate heterogeneous data sources?
Materialized Views for Stream Processing Systems

Adrian Michalke

motchalke@tu-berlin.de – EN 708

Motivation:
Novel use cases require data management of real-time and historic data. Streaming materialized views enable the pre-aggregation of historic data while being kept fresh by stream-maintenance.

Potential Topics:
1. Maintaining streaming materialized views → Efficient data structures
2. Using streaming materialized views → Rewrite rules/cost-models
3. Selecting streaming materialized views → Cost-models

Propose your own topic!

Prerequisites:
Programming skill in C/C++; good understanding of materialized views and stream processing; (ideally) completed several DIMA courses (e.g., SDS, BDSPRO, BDASEM, DBT, ROC, DBTLAB, DMH, DBPRA, DBPRO, DBSEM, ISDA).

Related Work:
NebulaStream is a stream processing system that operates over edge-cloud infrastructure, which contains heterogeneous hardware, low-powered devices, and different networking capabilities. We aim to operate over a plethora of these devices, in the most cost-effective way possible.

**Bachelor’s:**

1. Performance evaluation of stream processing workloads under heterogeneous CPU (micro) architectures
2. Reducing network overhead with adaptive filtering

**Master’s:**

1. Evaluating the usage of Unikernels for ephemeral stream processing
Stream Query Optimization for a unified fog-cloud infrastructure

We are building next generation stream processing system for disaggregated data processing called NebulaStream (www.nebula.stream). NebulaStream is designed to operate over a unified fog-cloud infrastructure. Such infrastructure present several challenges and opportunities for optimizing stream queries. Currently, we are investigating how to efficiently place operators for a large number of queries over such an infrastructure. Furthermore, we are conducting research to mitigate the effect of transient failures as well as monitor the effect that running queries have on system performance.

Open Topic:

Investigating approaches to allow continuous stateful stream query execution over continuously evolving infrastructure.

Prerequisites: Good programming skills (preferably in C++); good writing skills; knowledge about distributed, stream processing, or database systems.
Query processing on heterogeneous CPU/GPU systems

Research questions for individual master theses:
• How do atomic operations influence GPU performance?
• What is the tradeoff between data transfer and cooperation on fast coherent interconnects?
• How can database leverage dedicated hardware for matrix operations?
• What is faster on GPUs, vectorization or query compilation?
• Do we have to adapt query plans to different processors?
• How can we model processor and data characteristics for operator implementations?

Ideally, you should:
• Have experience in GPU programming (CUDA, OpenCL, …)
• Understand how hardware architecture affects performance.
• Have solid C and C++ programming experience.
• Know the basics of machine learning.

If you are interested, read this paper: 
[1] Rosenfeld et al., “Query processing on heterogeneous CPU/GPU systems”, ACM CSUR ‘23

... and contact Viktor Rosenfeld <viktor.rosenfeld@tu-berlin.de>
Java UDFs in NebulaStream

**Bachelor thesis topic:**
- Goal: Speed up execution of Java UDFs in NebulaStream.
- Problem: JNI calls have lots of overhead.
- Solution: Use strided execution to move hot loop from C++ code to embedded JVM.

**Ideally, you should:**
- Have C++ programming experience.
- Enjoy devising end-to-end performance and micro benchmarks.

If you are interested, read these papers:
[1] Rosenfeld et al., “Processing Java UDFs in a C++ Environment”, ACM SoCC’17

... and contact Viktor Rosenfeld <viktor.rosenfeld@tu-berlin.de>
Compliant Data Processing

• Inevitable increase in the processing of personal data
• Governments have begun crafting data protection regulations

• We define a data processing pipeline to be compliant when it respects data source protection policies.

Goal: Create frameworks that provide **compliant data processing by design.**

Problems and Challenges:
• Support multiple workloads
• Implement data policies should become simple
• Assess the impact of anonymization techniques
• Users should still get data of the highest disclosable quality

Advisor: Rudi Poepsel Lemaitre
E-mail: r.poepsellemaitre@tu-berlin.de
Office: EN 739
6. Questions from the Audience
Follow-up Questions or Concerns

For further questions or concerns feel free to contact Juan Soto via email.

His email address is juan.soto@tu-berlin.de.