Student Conference 2019
Topics

Gunter Saake, Jacob Krüger
Database Operation Tuning  
(David Broneske)

A current trend in database systems is to tune algorithms at a very fine granularity. Current code optimizations are controversially discussed, but a clear applicability of them is missing. Consequently, discuss the applicability of a subset of available code optimizations on selected database algorithms.

- Bogdan Raducanu, Peter Boncz, Marcin Zukowski: Micro Adaptivity in Vectorwise
- Jingren Zhou, Kenneth A. Ross: Implementing Database Operations Using SIMD Instructions
Database Operations on Modern Processing Devices  
(David Broneske)

Tuning database operations to the underlying hardware is a hot topic with the increasing usage of co-processors. There are numerous publications involving different algorithms and processing devices. Create a survey regarding database operations on different processing devices.

- Rene Müller, Jens Teubner, Gustavo Alonso: Data Processing on FPGAs
- Thomas Willhalm, Yazan Boshmaf, Hasso Plattner, Nicolae Popovici, Alexander Zeier, Jan Schaffner: SIMD-Scan: Ultra Fast in-Memory Table Scan using on-Chip Vector Processing Units
Evolution of column-oriented RDBMS operations
(Bala Gurumurthy)

Current trend in RDBMS is moving towards close-to-metal re-implementation of typical DBMS operations for underlying hardware. With the availability of newer features (like multi-core, SIMD) as well as device architectures (GPU, FPGAs) in the hardware landscape researches are done in tuning the operations to adapt to the hardware. In this work, we would survey the evolution of DBMS operations with reference points for the newer hardware availabilities. The work, in the end, provides a view on the hardware landscape with changes being applied to the DBMS operations and also the areas of dense and sparse researches.

- GPU-Accelerated Database Systems: Survey and Open Challenges - Sebastian Breß
- Accelerating SQL database operations with CUDA - Peter Bakkum
- Relational co-processing in graphics processors - Bin Sheng He
- Implementing Database Operations Using SIMD Instructions - J Zhou
GPU Cache management techniques for data processing environment  

(Bala Gurumurthy)

Due to limited cache space in a GPU, not all the input data can be processed and stored in GPU. As an alternative, hot input data buffers are proposed to be stored in a GPU for further processing without transfer overhead. In this work, we will look into the issue of caching in GPU and list the possible alternatives for caching in a GPU. Since column cannot be directly stored within a GPU, we look for alternative representation of data that is still sufficient for performing database operations over them (like bitmap, position list etc.) Overall, the work presents the state of the art techniques in intermediate representation for storing column in a GPU as well as the buffer management techniques used for caching in GPU.

• Waste Not.. Efficient Co-Processing of Relational Data - Holger Pirk
• In-cache query co-processing on coupled CPU-GPU architectures - Jiong He
• Efficient Data Management for GPU Databases - Peter Bakkum
• Techniques for Caches in GPUs - Guenther Schindler
Paving the way from game theory to cooperative DB components

(Gabriel Campero Durand)

Research in economy and game theory is ripe with models that seek to understand how agents compete for resources and how, through market design, they can be encouraged to collaborate, converging to optimal allocations for the group. In data management research there have been some attempts to adopt these models, for example in creating marketplaces for data fragmentation in the Mariposa Stream Processing System. However, this is not widely adopted. With the development of agent-based machine learning solutions for data management, it is possible that these techniques will gain relevance. In this topic we aim to start with a quick review on economic and game theory concepts, followed by a careful collection and discussion of related work. We conclude by proposing, based on discussions, potential applications in storage engine management, query processing and cloud computing.

Multi-agent deep reinforcement learning and databases (Gabriel Campero Durand)

The success of single agent deep reinforcement learning naturally creates interest in evolving to multiagent solutions, like DeepMind's AlphaStar. These are specially interesting since they address realistic use cases, where agents are not in entire control of a system. In this conference topic we will categorize the state of the art in the field, highlighting challenges and potentials in some approaches. In addition, we take a deep dive into one mature approach. We conclude by considering the feasibility of applying such approach to a database task.

• Database Query Optimization with Deep Reinforcement Learning: https://www.youtube.com/watch?v=Rw3ewEXOKC8

Learning from demonstrations with deep reinforcement learning (Gabriel Campero Durand)

Though reinforcement learning is a useful online method, it is often infeasible to train agents by interacting with a real-world system. Moreover, simulated environments are costly to produce. Thus, training agents in an offline manner, by using traces from an expert interacting with the system, is particularly compelling for practitioners. In this student conference topic we study in detail such approach. We consider how it has been used (or proposed to be used) in recent data management cases, and we list existing frameworks for off-the-shelf learning from demonstrations.

• Hester, Todd, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan et al. "Deep q-learning from demonstrations." In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.


Machine learning on networks and Graph-based recommenders  
(Gabriel Campero Durand)

Graph databases are a special kind of general data management system optimized for network-oriented analytical queries and storage. They are mainly developed to support a specific representation of a graph, namely property graphs. However, recent trends require further features from these databases, either to support novel data representations (embeddings) or highly efficient feature engineering processes. In this seminar topic we aim to study some of these trends, by considering one of two applications: machine learning on networks, or graph-based recommenders. For the chosen domain we describe carefully the domain, we take a detailed look at a given example study, and we outline the implications for system development.


Interests in Systematic Software Reuse
(Jacob Krüger)

Systematic software reuse in terms of software product lines is often only introduced after a larger set of different variants has evolved. For varying reasons, including cost reduction, faster development, or improved management, these variants are merged and integrated into a platform (reverse engineering). While there are several case studies that report on the migration processes and experiences, we still need a detailed analysis of the actual industrial motivations that lead to the adoption of product lines. To this end, we aim to analyze various topics on the adoption of software product lines at different venues and in different years. Topics may include the motivation and costs for extracting features, the evolution of software, or synchronizing independent variants at SPLC, ICSE, or VaMoS.

- A defined selection of topics, venues, and years can be defined to scope the extent of the analysis
Automated Test Refactoring

(Jacob Krüger)

Software is regularly updated or refactored, for example, to remove errors, introduce new features, or migrate towards a new technology. However, any change in the productive software also means that corresponding test cases may break or are not sufficient anymore. The purpose of this survey is to identify and summarize existing techniques on automated test case refactoring, meaning techniques that track code changes and support developers in maintaining the test cases for these artifacts.


• Arie van Deursen, Leon Moonen, Alex van den Bergh, and Gerard Kok. 2002. Extreme Programming Perspectives. Chapter Refactoring Test Code
How do We Forget?  

(Jacob Krüger)

Understanding a program is an essential activity in software engineering and the research area of program comprehension is extensively investigated. However, most studies are concerned with recovering understanding of a program and how to improve code design for this purpose. Such processes resemble learning of artifacts. In contrast, the process of forgetting in software engineering is rarely investigated. With this project, we aim to provide an overview on existing studies that are concerned with forgetting in software engineering and what factors affect developers' memory.

Cloud-based Protein Identification

(Roman Zoun)

Mass spectrometers are devices to digitize real world samples with growing success on the market. The technology sequences proteins to identify protein biomarkers of biological environments, such as oceans, humans, or microbial communities which are used in the research fields proteomics, metaproteomics and metabolomics. These biomarkers are similar to a fingerprint and can be used to identify the sample data. Due to the fast quality upgrades of the mass spectrometer, they produce ever-increasing amounts of data, resulting in terabytes of output data by a single machine. The analysis step, so called protein identification, is used to bring insights into the sample data. The protein identification is now a big data problem.

Task: Find protein identification solutions which use big data technology and map them to the big data landscape.
