Distributed and Parallel Databases (2023) 41:267–271
https://doi.org/10.1007/s10619-023-07435-z

# Introduction to the special issue on self-managing and hardware-optimized database systems 2022

**Constantinos Costa**[1,2] · **Ilia Petrov**[3]

Accepted: 23 May 2023 / Published online: 3 June 2023

Data management systems have evolved in terms of functionality, performance characteristics, complexity, and variety during the last 40 years. Particularly, the relational database management systems and the big data systems (e.g., Key-Value stores, Document stores, Graph stores and Graph Computation Systems, Spark, MapReduce/Hadoop, or Data Stream Processing Systems) have evolved with novel additions and extensions. However, the systems administration and tasks have become highly complex and expensive, especially given the simultaneous and rapid hardware evolution in processors, memory, storage, or networking. These developments present new open problems and challenges to data management systems as well as new opportunities.

The SMDB (International Workshop on Self-Managing Database Systems) and HardBD&Active (Joint International Workshop on Big Data Management on Emerging Hardware and Data Management on Virtualized Active Systems) workshops organized in conjunction with the IEEE ICDE (International Conference on Data Engineering) offered two distinct platforms for examining the above system-related challenges from different perspectives. The SMDB workshop looks into developing autonomic or self-* features in database and data management systems to tackle complex administrative tasks, while the HardBD&Active workshop focuses on harnessing hardware technologies to enhance efficiency and performance of data processing and management tasks. As a result of these workshops, we are delighted to present the third special issue of DAPD titled “Self-Managing and Hardware-Optimized Database Systems 2022,” which showcases the best contributions from the SMDB 2021/2022 and HardBD&Active 2021/2022 workshops.

✉ Constantinos Costa
costa.c@cs.pitt.edu

✉ Ilia Petrov
ilia.petrov@reutlingen-university.de

[1] University of Pittsburgh, Pittsburgh, USA

[2] Rinnoco Ltd, Limassol, Cyprus

[3] Reutlingen University, Reutlingen, Germany

## 1 In the following, we provide a brief overview of the 9 papers in this special issue

The paper by Wenyan Lu et al. on accelerating SQL processing utilizing a database offloading engine proposes a hardware-software co-designed system called Database Offloading Engine (DOE) that effectively offloads SQL operations, improving DBMS performance on CPU-accelerator heterogeneous systems. It contains a hardware accelerator architecture, Conflux, for effective SQL operation offloading, and a software DOE programming platform called DP2 for seamless integration and harnessing of computing power. DOE can partition relational operators and map them to multiple kernels in parallel. Experiment results show that DOE achieves over 60\% DRAM bandwidth utilization and more than 100x and 10x performance improvement compared to PostgreSQL and MonetDB, respectively. respectively.

The paper by Juwei Shi et al. on analyzing the performance of data parallel DAG Workflows for large-scale data analytics proposes a new cost model called Bottleneck Oriented Estimation (BOE) to accurately estimate the allocation of preemptable resources based on a predicted execution time in Directed Acyclic Graph (DAG) workflows. The model identifies bottlenecks to predict task execution time and uses a state-based approach to iteratively estimate overall execution plans. To handle the skewness of various jobs, the model is refined with order statistics theory. The BOE model outperforms state-of-the-art models by a factor of five, and the refined skew-aware model has an average prediction error under 3% when estimating execution time for 51 DAG workflows with HiBench and TPC-H workloads.

The paper by Muhammad Attahir Jibril et al. on handling the updates adaptively for graph HTAP proposes an adaptive update handling approach for graph databases in Hybrid Transactional/Analytical Processing (HTAP) workloads that benefit from GPU accelerators. To fully utilize GPU processing, dedicated graph representations are needed, but make the in-place updates difficult. The paper discusses and evaluates strategies for propagating transactional updates from an update-friendly table storage to a GPU-optimized sparse matrix format for analytics.

The paper by Alexander Baumstar et al. on compiling the queries adaptively for graph databases presents an adaptive approach for improving query compilation to efficient machine code in graph databases. Particularly, generating and optimizing machine code can become an issue for short-running queries or those that produce early results quickly. The proposed approach integrates graph query interpretation and compilation, with query execution starting immediately using the interpreter while code generation runs in the background. When the code generation is complete, the execution switches to the compiled code, which helps to hide compilation times and reduce latency. Evaluation results demonstrate the effectiveness of this approach.

The paper by Bala Gurumurthy et al. on atomic synchronization for sort-based group-by operations on GPUs discusses the use of GPUs to accelerate relational database operations, specifically the group-by operation. Hash-based approaches are currently preferred over sort-based approaches, but the latter can potentially exploit the parallelism and the memory architecture found in modern GPUs to minimize the overhead of atomics. The paper proposes different variants using atomics, which can almost entirely mitigate the overhead. The proposed approach achieves a 3x speed-up over existing sort-based approaches and up to 2x speed-up over hash-based approaches.

The paper by Lambros Odysseos et al. on combining system and machine learning to tune distributed data stream applications shows how ML applications processing continuous data streams over distributed stream processing engines (DSPEs) require tuning of system configurations and hyperparameters to optimize application throughput and accuracy. This paper presents a comprehensive study of the interactions between system configuration and hyperparameter tuning, revealing unexpected impacts on both application and ML model performance. The results suggest the need for new combined tuning approaches, and open new research directions in self-managing distributed stream processing systems.

The paper by Juliana Hildebrandt et al. on exploiting SIMD for lightweight integer compression discusses the use of integer compression in columnar database systems to reduce memory usage and improve query processing speed. The paper focuses on the use of Single Instruction Multiple Data (SIMD) extensions in modern CPUs to minimize the computational effort required for compression and decompression. While scalar compression algorithms typically compress a block of N consecutive integers, SIMDified implementations scale the block size to k * N, where k is the number of elements that can be simultaneously processed in an SIMD register. This approach can improve performance but may lead to a degradation of compression ratio compared to scalar variants. The paper proposes a novel SIMD concept called BOUNCE, which compresses k different blocks of size N concurrently within SIMD registers while guaranteeing the same compression ratio as scalar variants. The proposed SIMD concept is shown to work well on various Intel CPUs and may offer a new generalized SIMD concept to optimize other algorithms.

The paper by Hamza Mustafa et al. on a new GPU-based algorithm for local trajectory clustering discusses the challenge of clustering spatio-temporal data in the form of trajectories, and proposes a GPU algorithm called GTraclus to address scalability challenges. The location-based sensors like GPS have made it possible to collect large amounts of trajectory data, which can be used for applications such as intelligent transportation systems and urban planning. However, the volume of data and the concept drift present in city traffic pose scalability challenges. The proposed GTraclus algorithm uses the Minimum Description Length (MDL) principle to simplify trajectories into line segments and a parallelized trajectory partitioning algorithm for local trajectory clustering. The experimental results show that GTraclus has up to 24X faster execution time on average against the multicore CPU version of the Traclus algorithm, MC-Traclus.

The paper by Harish Kumar Harihara Subramanian et al. on supporting DBMS GPU-based operations with out-of-the-box libraries describes the ongoing research in the database community regarding the use of GPUs for query processing. The heterogeneity in GPU architectures and varying capabilities make it challenging to develop optimal operator implementations for specific device generations involving manual tuning. The paper investigates various general-purpose libraries that are portable and easy to use for arbitrary GPUs for testing their production readiness for database operations. The authors develop a framework that allows a user to easily plug-in new libraries and custom code for query execution using a simple task model. The paper experimentally evaluates different GPU libraries' performance and observe significant diversity among them.

Finally, we would like to thank all the authors who have contributed their papers for this issue. We own our sincere gratitude to the reviewers who contributed to assembling such a high-quality special issue. We are also indebted to the DAPD Journal Editors, editorial office, and the publishing and production teams for their assistance in preparation and publication of this issue.

### 1.1 Lead guest editors

Constantinos Costa, University of Pittsburgh & Rinnoco Ltd, costa.c@cs.pitt.edu
Ilia Petrov, Reutlingen University, Germany, ilia.petrov@reutlingen-university.de

### 1.2 Guest editors

Herodotos Herodotou, Cyprus University of Technology, herodotos.herodotou@cut.ac.cy
Yingjun Wu, Singularity Data Inc., yingjunwu@singularity-data.com
Bailu Ding, Microsoft Research, bailu.ding@microsoft.com
Panos K. Chrysanthis, University of Pittsburgh, panos@cs.pitt.edu
Meichun Hsu, Oracle Corporation, meichun.hsu@oracle.com.
Shimin Chen, Chinese Academy of Sciences, chensm@ict.ac.cn
Mohammad Sadoghi, UC Davis, msadoghi@ucdavis.edu

### 1.3 Reviewers

Bo Tang, (Southern University of Science and Technology)
Deepak Majeti, (Ahana)
Eduardo Cunha de Almeida, (Federal University of Paraná)
George Pallis, (University of Cyprus)
Ilia Petrov, (Reutlingen University)
Jeff LeFevre, (University of California, Santa Cruz)
Jiaheng Lu, (University of Helsinki)
John Paparrizos, (University of Chicago)
Khuzaima Daudjee, (University of Waterloo)
Siddhartha Sahu, (University of Waterloo)
Manos Athanassoulis, (Boston University)
Matthias Sax, (Confluent Inc.)
Meike Klettke, (Universität Regensburg)
Mohammad Sadoghi, (University of California, Davis)
Peiquan Jin, (University of Science and Technology of China)
Qiong Luo, (Hong Kong University of Science and Technology (HKUST))
Sang-won Lee, (Sungkyunkwan University)
Shimin Chen, (Chinese Academy of Sciences)
Thamir Qadah, (Umm Al-Qura University)
Tianzheng Wang, (Simon Fraser University)

Uta Störl, (University of Hagen)
Wolfgang Lehner, (Technische Universität Dresden)
Xiaodong Zhang, (The Ohio State University)
Xinyu Chen, (National University of Singapore)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.