Example Quote from Citations

Volker Markl
TU Berlin
VLDB Endow. Chair

adapters in [17]. He et al. investigated relational query processing on GPUs by implementing a complete set of relational operators for the GPU and a distinct set for the CPU in their custom-tailored database system GPUDB [19]. In general, all of those systems were

Heimel, Max, Michael Saecker, Holger Pirk, Stefan Manegold, and Volker Markl. "Hardware-oblivious parallelism for in-memory column-stores." **VLDB, 2013.**

Jens Teubner
TU Dortmund

2.5.2 Operator Placement

He and others early recognized the importance of operator placement and data movement [13]. To address the problem, they performed backtracking for sub-query plans and combined optimal sub plans to the final query plan [13]. From our experience in CoGaDB, this backtracking approach can be very time consuming. Thus, CoGaDB uses an iterative refinement optimizer that only considers query plans

Breß, Sebastian, Henning Funke, and Jens Teubner. "Robust query processing in co-processor-accelerated databases." **SIGMOD, 2016.**

Example Quote from Citations

Martin Kersten
Universiteit van Amsterdam
ACM Fellow

due to I/O starvation. In the case of joins, the input data is usually radix-partitioned and every partition processed in one thread [8]. This allows efficient lock-, synchronization- and replication-free parallel processing of the partitions if the data is distributed close to uniformly. For skewed data, radix-partitioning might fail to achieve the aspired perfor-

Pirk, Holger, Stefan Manegold, and Martin L. Kersten. "Accelerating Foreign-Key Joins using Asymmetric Memory Channels." **VLDB, 2011.**

Wolfgang Lehner
TU Dresden
ACM Fellow

within an OpenCL based column store database system. He et al. [5] computed all possible solutions for separate sub-plans below a given number of operators and combined the result for the full plan dividing the search space into much smaller problems. However, this is only applicable for tree

Karnagel, Tomas, Dirk Habich, and Wolfgang Lehner. "Local vs. Global Optimization: Operator Placement Strategies in Heterogeneous Environments." **EDBT/ICDT, 2015.**

Example Quote from Citations

Kevin Skadron
University of Virginia
IEEE Fellow, ACM Fellow

[12]. This is a culmination of much of the previous research performed on GPU-based data processing. Its authors design a database, called GDB, accessed through a plethora of individual operations. These operations are divided into operators, access methods, and primitives. The operators include ordering, grouping, and joining functionality. The access methods control how the data is located in the database, and includes scanning, trees, and hashing. Finally the primitives are a set of functional programming operations such as map, reduce, scatter, gather, and split. GDB has a number

Bakkum, Peter, and Kevin Skadron. "Accelerating SQL database operations on a GPU with CUDA." **GPGPU, 2010.**