Alonso, Gustavo; Andrzejewski, Witold; He, Bingsheng; Fröning, Holger; Sattler, Kai-Uwe; Seeger, Bernhard; Sitaridi, Evangelia; Teich, Jürgen; Zukowski, Marcin:

Database accelerators

DOI:	10.22032/dbt.42361
URN:	urn:nbn:de:gbv:ilm1-2020200360

Original published in:	Database architectures for modern hardware : report from Dagstuhl Seminar 18251 / Boncz, Peter A.; Gräfe, Götz; He, Bingsheng; Sattler, Kai-Uwe. - Wadern : Schloss Dagstuhl. - 8 (2018), 6, p. 67-68. (Dagstuhl Reports : Dokumentationen zu Dagstuhl-Seminaren und Dagstuhl-Perspektiven-Workshops / Schloss Dagstuhl, Leibniz-Zentrum für Informatik. - Wadern : Schloss Dagstuhl. - 8 (2018), 6, p. 63-76. - DOI: 10.4230/DagRep.8.6.63)
ISSN:	2192-5283
Original published:	2019-01-17
[Visited:	2020-03-12]

3 Working groups

3.1 Database Accelerators

Gustavo Alonso (ETH Zürich, CH), Witold Andrzejewski (Poznan University of Technology, PL), Bingsheng He (National University of Singapore, SG), Holger Fröning (Universität Heidelberg – Mannheim, DE), Kai-Uwe Sattler (TU Ilmenau, DE), Bernhard Seeger (Universität Marburg, DE), Evangelia Sitaridi (Amazon.com, Inc. – Palo Alto, US), Jürgen Teich (Universität Erlangen-Nürnberg, DE), and Marcin Zukowski (Snowflake Computing Inc. – San Mateo, US)

Hardware-based acceleration technologies provide great opportunities for speeding up database processing. GPUs (optionally with Tensor Cores), iGPUs, FPGA, TPUs, intelligent network devices, memory and disks are only some examples of suitable approaches. Based on an analysis of available technologies we discussed in the working group use cases for database accelerators from the level of individual operators and algorithms up to the level of complex tasks such as query planning and optimization. Particularly, we investigated the following three questions: How to exploit tensor cores for query operators, e.g. for joins? How to speed up (by batch processing) scalar functions, parsing/deserialization of strings/CSV/JSON, as well as the transposition of batches of records? How to exploit accelerators for cardinality estimation and query planning? How to abstract the execution on heterogeneous resources with the help of a task dependency model?

There are a few interesting points to note in the discussion. First, it is still quite difficult to obtain the most efficient implementation for a given problem on a target architecture, although the problem has been studied to some extent in the form of paper publication or open source. Second, hardware architectures are evolving, and even the state-of-the-art implementations can become inefficient in future architectures. Third, one of the consequences from the first and second points is that, it is rather challenging and tedious to have a fair and complete benchmark on different implementations for a given problem across different architectures.

As results of the group's discussion we propose a public repository for implementations of database tasks using different accelerator technologies which forms the basis for programming contests and at the same time allows for a performance comparison of different implementations. As a second result we discussed a survey to be prepared that covers the state of the art of implementing fundamental database operations such as joins, aggregations, sorting, and advanced scans for different accelerators, so that the community can be aware of the state-of-the-art work that has been done, and identify the challenges and opportunities for improving the performance of those operations.

Finally, we discussed future research directions with respect to hardware acceleration for database tasks both on premise (intelligent memory and storage controllers, memory filters, as well as gather operations with static and dynamic strides) as well as for cloud environments (intelligent storage, virtualization of accelerators). The work items are to be defined, since the scope spans across many relevant areas. It can be the topic of a future Dagstuhl seminar.

As a side project, we had quite intensive discussions on how to exploit the Tensor cores for different data processing operations beyond deep learning. The recent and rapid development of deep learning systems and applications have driven tremendous efforts in tensor accelerator units. One example is from NVIDIA's tensor core and the other example is Google's TPU. Those tensor hardware units can typically demonstrate superb tensor computation performance. In this study, we show how common database operations can be implemented from those sensor operations. We will implement our proposal on NVIDIA Volta architecture, and demonstrate its performance and tradeoff. We expect that there will be some tradeoff in such mappings since they may be too restrictive in implementing with tensor operations.

References

1 Bingsheng He. *Data Management Systems on Future Hardware: Challenges and Opportunities*. Proc. 33rd Int. Conference on Data Engineering (ICDE), p. 1609, 2017.

2 Sebastian Breß, Max Heimel, Norbert Siegmund, Ladjed Bellatreche, and Gunter Saake. *GPU-Accelerated Database Systems: Survey and Open Challenges*, Trans. Large-Scale Data and Knowledge-Centered Systems, Vol. 15, pp. 1–35, 2014.