HyPer: one DBMS for all

High-Performance OLTP AND OLAP on Brawny Servers AND Wimpy Devices

Tobias Mühlbauer, Florian Funke, Viktor Leis, Henrik Mühe, Wolf Rödiger, Alfons Kemper, Thomas Neumann

Technische Universität München
Munich, Germany
{muehlbau, finis, funke, leis, muehe, roediger, kemper, neumann}@in.tum.de

ABSTRACT

Ever increasing main memory capacities and processors with multiple cores have fostered the development of database systems that process and store data solely in main memory.

This talk presents HyPer [2], a high-performance hybrid OLTP&OLAP main memory database system. Unlike other main memory database systems, HyPer aims at providing highest performance for both, OLTP AND OLAP workloads on brawny AND wimpy systems. OLAP query processing is separated from mission-critical OLTP transaction processing using an efficient virtual memory (VM) snapshotting mechanism. Platform-independent high-performance OLTP and OLAP is achieved by efficiently compiling transactions and queries into efficient target machine code [8]. Even though the SQL-92 standard, a PL/SQL-like scripting language, and ACID-compliant transactions are supported, HyPer has a memory footprint of just a few megabytes. In particular, this talk highlights the following recent research efforts in the HyPer project:

Transaction processing. Transaction processing is dominated by index accesses. We specifically designed our default index, the *Adaptive Radix Tree (ART)* [3], for main-memory database systems and modern hardware. Its performance is only comparable to highly optimized hash table implementations but supports range scans while being very space efficient.

Intel's introduction of the Haswell microarchitecture added support for *Hardware Transactional Memory (HTM)*. HTM allows the execution of a section of code atomically and in isolation. HyPer uses this feature as a building block for processing transactions in parallel with high scalability and low overhead [4].

To handle long-running write transactions, HyPer was extended with a mechanism called *tentative execution* to allow for such transactions to be processed on a consistent snapshot of the database [5]. Side effects are applied to the actual database on commit.

Storage. Main memory is a precious resource when it comes to main memory database systems. In [1], we thus proposed the *compaction* of memory-resident databases by separating the hot working-set from the immutable cold data. We have shown that we can apply compression and further physical optimizations to the cold data without compromising OLTP performance.

Bulk loading of structured text data has become a major bottleneck in database systems. The delays it is causing are now ubiquitous as text formats are a preferred storage format for reasons of portability. With *Instant Loading* [7], we proposed a novel flat file loading approach that allows scalable bulk loading at the wire speed of SSDs and 10 GbE adapters.

Scale-out. To allow for the analysis of very large data sets, HyPer needs to be scaled out. In the course of building a distributed query engine, we developed a novel locality-sensitive data-shuffling scheme that improves network utilization of expensive *distributed database operators* [9]. Our scheme exploits locality in the data distribution and handles skewed inputs gracefully.

Further, *ScyPer* [6], a scaled-out version of our HyPer system was developed with the aim of maintaining the high transactional performance of a single HyPer instance (more than 100,000 TPC-C TX/s), while at the same time allowing elastic OLAP throughput on the transactional data.

1. REFERENCES

[1] F. Funke, A. Kemper, and T. Neumann. Compacting transactional data in hybrid OLTP&OLAP databases. *PVLDB*, 5(11):1424–1435, 2012.

[2] A. Kemper and T. Neumann. HyPer: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots. In *ICDE*, 2011.

[3] V. Leis, A. Kemper, and T. Neumann. The Adaptive Radix Tree: ARTful Indexing for Main-Memory Databases. In *ICDE*, pages 38–49, 2013.

[4] V. Leis, A. Kemper, and T. Neumann. Exploiting Hardware Transactional Memory in Main-Memory Databases. In *ICDE*, 2014.

[5] H. Mühe, A. Kemper, and T. Neumann. Executing Long-Running Transactions in Synchronization-Free Main Memory Database Systems. In *CIDR*, 2013.

[6] T. Mühlbauer, W. Rödiger, A. Reiser, A. Kemper, and T. Neumann. ScyPer: Elastic OLAP throughput on transactional data. In *DanaC*, 2013.

[7] T. Mühlbauer, W. Rödiger, R. Seilbeck, A. Reiser, A. Kemper, and T. Neumann. Instant Loading for Main Memory Databases. *PVLDB*, 6(14):1702–1713, 2013.

[8] T. Neumann. Efficiently compiling efficient query plans for modern hardware. *PVLDB*, 4(9), 2011.

[9] W. Rödiger, T. Mühlbauer, P. Unterbrunner, A. Reiser, A. Kemper, and T. Neumann. Locality-Sensitive Operators for Parallel Main-Memory Database Clusters. In *ICDE*, 2014.