

Master Project:

In-Memory Data Management – To Parallelize or not to Parallelize: that is NOT the Question for Real-Time Analytics; the Question is How Much.

General Background

Traditional enterprise systems use two separate database systems, one for Online Transaction Processing (OLTP) and another for Online Analytical Processing (OLAP). This separation was introduced because of performance reasons. With the current hardware (80 CPU cores / 2TB main memory per machine; possibly multiple servers) and software developments, a common database approach for both, OLTP and OLAP, becomes feasible using In-Memory Column Databases [P09, P11, PZ12].

The Challenge of Scheduling for Parallel Query Execution

While one might argue whether parallel query execution is required for transactional query processing, analytical queries on large data sets need to run in parallel to achieve acceptable performance for real-time analytics. With 80 cores and counting per computing node, sequential processing of queries is not an option, as we cannot expect single queries to fully utilize all available resources throughout execution. Several challenges arise if we allow intra-query parallelism, e.g., what if we need to run two queries in parallel? A naïve approach would be to divide the cores by two and run each query on half the cores. However, due to architectural restrictions (e.g., data partitioning, memory channels, bandwidth restrictions), operator implementations (e.g., sequential code, sync points) and data characteristics (e.g., table sizes, number of distinct values) most queries / operators will most likely show sub-linear scaling curves. And what happens if one query finishes earlier, or if further queries arrive during execution?

Much research has been conducted in the area of parallel query execution in the area of disk-based database management systems (e.g., [RM95], [WC04]). However, we see three main changes that do not allow us to simply adopt previous work to In-Memory Databases:

- *More processing units*: We have much more parallel processing units in one computing node (many-core CPUs) and see a trend for further growth
- *Faster execution time*: Queries are executed much faster in an In-Memory Database (rather seconds compared to minutes or hours); hence the overhead of running a complex scheduling algorithm easily accounts for a significant portion of total execution time of a query
- *No disk*: Much of the previous work has been focused on avoiding the disk I/O bottleneck. Disk is not a bottleneck for complex, read-intensive queries on In-Memory databases anymore

Due to these changes, existing work on parallel query processing has to be revisited for In-Memory databases and potentially adapted.

Envisioned Content of Master Project

The envisioned outcome of the master project is a demonstration that showcases the benefits of an optimized scheduling for parallel queries on in-memory databases. This includes the following tasks:

- Evaluation and Optimization of scheduling algorithms for parallel queries in the context of in-memory databases
- Implementation of scheduling algorithms in a database prototype
- Optimization of database operators for parallel execution
- Implementation of a demo application that demonstrates the potential of the prototype

During the project, we will have the opportunity to test our implementation with enterprise class hardware (256GB RAM, 32 cores), as well as relevant query workloads of enterprise applications.

Expected requirements for project participants

- Solid programming skills; especially C++; ideally experience in parallel programming
- Knowledge of In-Memory Data Management (e.g., Participation in one of the corresponding lectures offered by the EPIC chair)

References

- [P09] Hasso Plattner: A Common Database Approach for OLTP and OLAP Using an In-Memory Column Database, Proceedings of the 35th SIGMOD International Conference on Management of Data, 2009
- [P11] Hasso Plattner: SanssouciDB: An In-Memory Database for Processing Enterprise Workloads, BTW, 2011
- [PZ12] Hasso Plattner and Zeier, A.: In-Memory Data Management – Technology and Applications, Springer, 2012
- [RM95] Erhard Rahm and Robert Marek: Dynamic Multi-Resource Load Balancing in Parallel Database Systems, Proceedings of the 21th International Conference on Very Large Data Bases, 1995
- [WC04] Jun Wu and Jian-Jia Chen and Chih-wen Hsueh and Tei-Wie Kuo: Scheduling of Query Execution Plans in Symmetric Multiprocessor Database Systems, Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS), 2004

Contact

Enterprise Platform and Integration Concepts Group, HPI, Potsdam

- Prof. Hasso Plattner, office-epic@hpi.uni-potsdam.de
- Dr. Jürgen Müller, juegen.mueller@hpi.uni-potsdam.de, +49 (331) 5509 – 1319
- Johannes Wust, Johannes.wust@hpi.uni-potsdam.de, +49 (331) 5509 – 1310