Enterprise Platform and Integration Concepts
Research Group of Prof. Dr. Hasso Plattner

Master's Project, Summer Term 2019

Parallelization and Query Plan Optimization for the TPC-DS Benchmark

Description

For relational databases, standardized benchmarks are the method of choice for comparing throughputs of competing database systems or of different versions of a system. One of these benchmarks is the TPC-DS benchmark, which simulates typical queries as seen in decision support systems. Compared to the popular TPC-H benchmark, TPC-DS poses more challenges to the database system as queries are more complex and the input data is skewed.

In this project, we will take the TPC-DS benchmark as a yardstick for improving our own open-source database system Hyrise. While extending the existing C++ code base of Hyrise, the focus will be (1) on improving the scalability of the system, i.e., using available CPU cores as efficiently as possible, and (2) on optimizing the query plan generation so that more efficient execution paths are chosen by the database engine. With a flexible benchmark framework already in place in Hyrise, it will be a matter of days before we can look at first performance numbers. From there, we can track our improvements and will have measurable impact early in the project.

The results will be integrated into the main code base of Hyrise to improve the overall project. After this project, participants will also have the opportunity to dive deeper into identified issues as part of a Master's thesis.

Goals

Several goals can be addressed independently during the course of this project, depending on the number of participants, interests, and progress during the project:

1. Implementing the TPC-DS benchmark (generating data, parameterizing queries)
2. Implementing SQL features required to execute TPC-DS queries
3. Analyzing generated query plans generated and improving the optimizer component of the database
4. Identifying control flow inefficiencies and improving the database operators to a point where they are entirely memory-bound
5. Improving the scheduler and the placement of data on different NUMA nodes in order to increase the TPC-DS throughput in systems with many cores

Existing Infrastructure

We will provide a productive work environment and appropriate infrastructure to get you up to speed quickly. This includes

- a database system that can execute most queries out of the box,
- a benchmark framework, which automatically executes queries in parallel, tracks their execution time and reports the results,
- scripts for comparing multiple benchmark runs, e.g., for tracking the improvement made by a commit; as well as scripts that plot the throughput with varying number of CPUs or the improvement over time, and
- a code base with a high degree of test coverage (>90%) and a CI server that enforces code quality for all pull requests

Learning Goals

During this project, you will gain insights into the inner mechanics of a relational database and how complex queries are executed efficiently. Furthermore, you will get a better understanding of programming multi-threaded software both on a single processor and on systems with up to 16 CPUs and 480 logical cores. As you will have to work with existing database components such as the query optimizer and the scheduler, you will learn to familiarize yourself and contribute to a large existing C++ code base.

Prerequisites

Prior understanding of the fundamentals of databases (e.g., through attendance of the lecture 'Datenbanksysteme', the online lecture 'In-Memory Data Management', or the seminar 'Develop your own Database') as well as knowledge of C++ is expected.

Contact

You are welcome to contact one of us via mail or visit us in the villa.

Matthias Uflacker, Markus Dreseler, Jan Koßmann