Efficient Query Processing on Modern Hardware

Thomas Neumann
Lehrstuhl für Informatik III: Datenbanksysteme
Fakultät für Informatik
Technische Universität München

ABSTRACT

Most database systems translate a given query into an expression in a (physical) algebra, and then start evaluating this algebraic expression to produce the query result. The traditional way to execute these algebraic plans is the iterator model: Every physical algebraic operator conceptually produces a tuple stream from its input, and allows for iterating over this tuple stream. This is a very nice and simple interface, and allows for easy combination of arbitrary operators,but it clearly comes from a time when query processing was dominated by I/O and CPU consumption was less important: The iterator interface causes thousands of expensive function calls, degrades the branch prediction of modern CPUs, and ofter results in poor code locality and complex book-keeping.

On modern hardware query processing can be improved considerably by processing tuples in a data centric, and not an operator centric, way. Data is processed such that it can be kept in CPU registers as long as possible. Operator boundaries are blurred to achieve this goal. In combination with an code compilation framework this results in query code that rivals the speed of hand-written code. When using these techniques in the HyPer DBMS, TPC-H Query 1 for example can single-threaded aggregated the scale factor 1GB data set in about 68ms on commodity hardware.