Code Optimizations for Database Operations on Heterogenous Hardware

Arising from increasing hardware capabilities – especially increasing main memory – main-memory database systems are gaining importance. Due to the shift of the access bottleneck from the disk to the access of the main-memory, traditional assumptions fail, such as computations in main-memory can be obeyed in cost models. As a consequence, instead of improving the gathering of data from disk, we are now forced to improve the data processing itself and, thus, our algorithms (e.g., selections, joins, aggregations).

To reach good performance for our algorithms, we have to tune them to the underlying hardware. In this case, underlying hardware means that the algorithms have to exploit the capabilities of the used processor. In fact, these processors are not limited to the central processing unit (CPU), but in fact, current database research implies to use co-processors such as graphics processing units (GPUs) or Xeon Phis.

Apart from different (co-)processing devices, that may be available in our database system, there are several code optimizations that can be applied for a given database operator such as vectorization, parallelization, and loop unrolling. However, applying all applicable code optimization on a given algorithm will not lead to the optimal performance, because whether they improve an algorithms performance depends on the characteristics of the used processor and the workload (e.g., the selectivity).

Arising from the heterogeneity of modern (co-)processors and the huge amount of possible code optimizations, we can offer the following thesis topics for bachelors or masters:

**Master: A Survey on Code Optimizations for Database Operations**

In the high-performance community (HPC), code optimizations are well known and a huge variety has already been applied to several problems. However, the database community is not aware of most of these code optimizations and how to apply them for their database operations. As a consequence, this topic for a master thesis includes conducting an extensive survey on code optimizations that are used in the high-performance community as a basis. From this point, there are two directions that the discussion of the work could take: (1) you discuss and present how the characteristics of a given processing device may favor the found code optimizations, and (2) you discuss the applicability of the found code optimizations for the common database operations (selection, joins, aggregation, sorting).

**Bachelor/Master: Applying Code Optimizations on a given Database Operation**

Preliminary work on applying code optimizations on database operations has already been published in the community. Nevertheless, there is still a need for further investigations and implementations of code optimizations for database operations. Given a set of code optimizations and a given database operation, the task is to apply these code optimizations (and their combinations if possible) on the operation and conduct a performance study for them. The goal is to get a better understanding of the performance behavior of database operations depending on the applied code optimizations.