A Comparative Study of Implementation Techniques for Query Processing in Multicore Systems

Multicore systems and multithreaded processing are now the de facto standards of enterprise and personal computing. If used in an uninformed way, however, multithreaded processing might actually degrade performance. We present the facets of the memory access bottleneck as they manifest in multithreaded processing and show their impact on query evaluation. We present a system design based on partition parallelism, memory pooling, and data structures conducive to multithreaded processing. Based on this design, we present alternative implementations of the most common query processing algorithms, which we experimentally evaluate using multiple scenarios and hardware platforms. Our results show that the design and algorithms are indeed scalable across platforms, but the choice of optimal algorithm largely depends on the problem parameters and underlying hardware. However, our proposals are a good first step toward generic multithreaded parallelism.