Two emerging hardware trends will dominate the database system technology in the near future: increasing main memory capacities of several TB per server and massively parallel multi-core processing. Many algorithmic and control techniques in current database technology were devised for disk-based systems where I/O dominated the performance. In this work we take a new look at the well-known sort-merge join which, so far, has not been in the focus of research in scalable massively parallel multi-core data processing as it was deemed inferior to hash joins. We devise a suite of new massively parallel sort-merge (MPSM) join algorithms that are based on partial partition-based sorting. Contrary to classical sort-merge joins, our MPSM algorithms do not rely on a hard to parallelize final merge step to create one complete sort order. Rather they work on the independently created runs in parallel. This way our MPSM algorithms are NUMA-affine as all the sorting is carried out on local memory partitions. An extensive experimental evaluation on a modern 32-core machine with one TB of main memory proves the competitive performance of MPSM on large main memory databases with billions of objects. It scales (almost) linearly in the number of employed cores and clearly outperforms competing hash join proposals -- in particular it outperforms the "cutting-edge" Vectorwise parallel query engine by a factor of four.

Stichworte:
Parallel query processing, Joins, Main memory, Multi-core database systems

Jahr:
2012
Sprache:
en