Review of "Inspector Joins"

http://lsirpeople.epfl.ch/nbonvin/courses/adms07/
Nicolas Bonvin

November 29, 2007

Summary In the traditional hash-based join algorithms there are two phases : partitioning and probing. During the first phase, the algorithm sees all the data. In this paper the authors introduced *inspector joins* which take advantage of this knowledge to optimize the second phase of the algorithm. They used this knowledge to improve the speed of a new type of cache-optimized join phase algorithm and to choose a join phase algorithm that is best suited to the data. They showed how theses statistics and specialized indexes are used to improve performance by reducing the CPU cache performance bottleneck.

Top 3 contributions

- *Introduction of inspector joins* : they take advantage of the information obtained during one pass of the join algorithm to improve the performance of a later pass by building the multifilters :
  - they minimize the number of cache misses without moving tuples around,
  - they exploit cache prefetching,
  - they choose the join phase algorithms according to theses informations about the data.
- *Specialized indexes* : these new indexes tackle the memory bandwidth sharing issue and are able to take advantage of nearly-sorted relations by enabling the novel cache-stationary optimization.
- *Concrete performance improvements on SMP machines* : usually performance decreases with the number of processors because they fight for resources (cache, ...). With this approach performance increases with the number of processors. The experimental results provided by this paper are well explained and look promising.

Most glaring problems

- *Special architecture needed* : they use a special kind of prefetch instruction which is only supported by 2 architectures (Itanium2, Pentium4). How about other architectures ?
- *General complexity* : the algorithm seems to be complex and introduces new data structures which add some maintenance costs.

Conclusion I like the way the "free" information provided by the inspection are used to choose a join phase algorithm corresponding the best to the given data. This approach is specific to a single query, works efficiently for any join query and predicate combinations and is used as an optimization technique direcly inside the join operator.