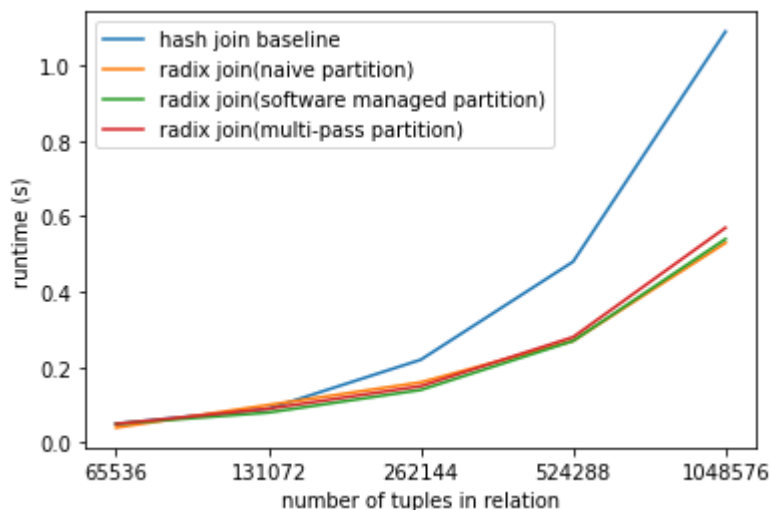


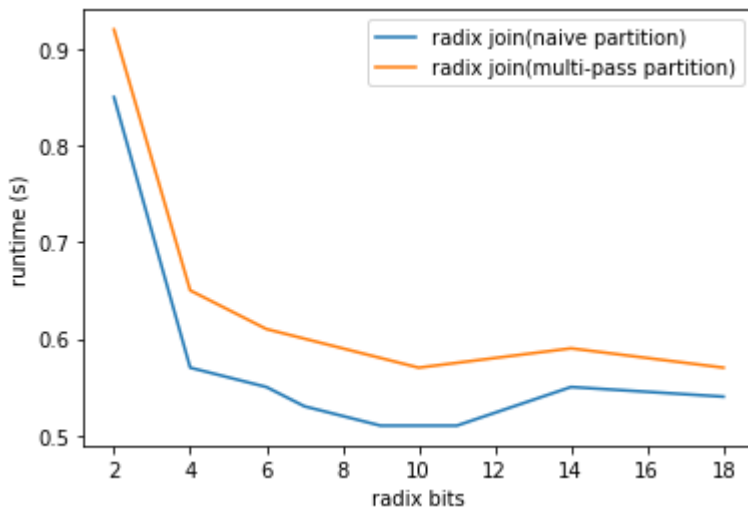
Assignment3 - Hardware optimized hash joins

Performance Analysis



- As number of tuples increasing, the run time increases linearly for the hash join baseline. Because it's just the naive loop which goes through the whole relation to build the hash table.
- Radix join is almost twice as fast as the hash join baseline.
- Software managed partition + non-temporal writes radix join is a little bit faster than the normal partition radix join and multi-pass partition radix join.

Partitioning Analysis



- Too many partitions might thrash the TLB cache, causing there is no more performance improvement.
- Using 9 or 10 radix bits for partition can achieve the best performance.
- Using the same number of radix bits, naive partition radix join is faster than multi-pass partition. Because for the naive partition radix join, we only copy each tuple in the relations once. But for the multi-pass partition, we copy each tuple once in each pass, there are total two pass in my experiment, thus each tuple in the relations is copied twice, which affects the performance of multi-pass partition radix join.