16 – Vectorize VS. Compilation

1. Background
   A. Vectorwise
      i. Pre-compiles primitives which are basic operator on typed data
      ii. DBMS uses these primitives to evaluate query.
   B. Hyper
      i. Compile query by JIT compilation
      ii. Re-arrange query to data centric one which is equivalent to original query.
2. Comparation between two models
   A. Single threaded environment
      i. For aggregation and calculative query, Typer is better,
         (data-centric is good at calculative query because of few cache misses)
         but, for join and filter, Tectorwise is better.
         (vectorize is good at hiding cache misses)
      ii. Reasons of performance differences.
         1. Typer is good at calculative query because it can evaluate query with a few instructions.
         2. Tectorwise is good at join with small selectivity because its lower branch misprediction.
            Also, Tectorwse is good at join with large selectivity because it can amortize memory stalls, so that it can reduce cycle count.
   B. SIMD performance
      i. SIMD can improve performance than when it is evaluated in scalar implementation, especially on hashing and join
      ii. But, if data are not fit on CPU cache, its performance will fall down significantly.

3. Hybrid Model

   A. Vectorization and compilation is basically exclusive to each other, but there are hybrid models.

   B. Relaxed Operator Fusion

      i. Decompose pipelines into stages which operate on vectors of tuples.

      ii. Can solve fusion problem of vectorizing data-centric query plan.

      iii. It can improve query performance where needs hiding some cache miss latency.