Week3

James Quacinella

09/11/2015

Exercise 2.5.1

Question

What is the communication cost of each of the following algorithms, as a function of the size of the relations, matrices, or vectors to which they are applied?

(a) The matrix-vector multiplication algorithm of Section 2.3.2.
(b) The union algorithm of Section 2.3.6.
(c) The aggregation algorithm of Section 2.3.8.
(d) The matrix-multiplication algorithm of Section 2.3.10

Answer

(a) The fact that vector v cannot be stored in memory and is split up does not affect the total number of keys being created. As stated in the text: "From each matrix element m_{ij} it produces the key-value pair (i, m_{ij}, v_j)". The number of keys at the output is a lot smaller since the result is in some sense 'aggregating' data. So, the communication cost is simply the the number of key - value pairs needed, or $O(rc)$, with r being the number of rows of the matrix and c is the number of columns.

(b) The union algorithm produces a total t key-value tuples, where $t = r + s$. The reducer will be on the same order as the input. Therefore the communication is $O(t) = O(r + s)$.

(c) The mapper has an input size of r, the number of tuples in relation R. The reducer is harder to determine since it depends on how well the A attribute groups the data. Worse case, where each tuple is its own group, would mean that there is r tuples. The algorithm there is $O(r)$.

(d) The number of key value pairs generated from M will be the number of columns of N, c_N, times the number of rows of M, r_M. Similar logic for matrix N, we have $c_M r_N$. The reducer must produce $r_M c_N$ tuples as the final output. The order of the algorithm however, is $O(r_M c_N + c_M r_N + r_M * c_N)$. Since $c_M = r_N$ for matrix multiplication to work, this reduces to $O(2r_M c_N + {c_M}^2)$. For square matrices, this becomes $O(n^2)$, where n is the dimension of the matrices.

Question

Exercise 2.6.1 : Describe the graphs that model the following problems. (a) The multiplication of an n $\times$ n matrix by a vector of length n. (b) The natural join of R(A, B) and S(B, C), where A, B, and C have domains of sizes a, b, and c, respectively. (c) The grouping and aggregation on the relation R(A, B), where A is the grouping attribute and B is aggregated by the MAX operation. Assume A and B have domains of size a and b, respectively.

Answer

(a) The graph model of the problems lists out the set of inputs, outputs and a many to many relationship between the two. The set of outputs is the $nx1$ vector output, which are a set of n values. The set of inputs are the key-value tuples of the form (i, $m_{ij}v_j$) where i ranges from 0 to n, and j ranges from 0 to n as well. If the output is labled v, then the input (i, $m_{ij}v_j$) maps to v_i.

(b) The set of inputs are all possible R tuples with all the possible S tuples. The outputs are all possible triples, with components from the domains of A, B, and C in that order. Each output is mapped from two inputs. For output (a,b,c), the two inputs (a,b) and (b,c) map to it.

(c) The set of inputs are the tuples in the R relation, (a, b). The outputs are of the form $(a, MAX(b))$, where $MAX(b)$ is the maximum value of the b values that have corresponding key of a. Each output is mapped to the inputs that have the same corrssponding key value, since the a component is how we aggregate.