

Matrix Multiplication with One MapReduce Step

There often is more than one way to use MapReduce to solve a problem. You may wish to use only a single MapReduce pass to perform matrix multiplication $P = M N$. It is possible to do so if we put more work into the two functions. Start by using the Map function to create the sets of matrix elements that are needed to compute each element of the answer P . Notice that an element of M or N contributes to many elements of the result, so one input element will be turned into many key-value pairs. The keys will be pairs (i, k) , where i is a row of M and k is a column of N . Here is a synopsis of the Map and Reduce functions.

The Map Function: For each element m_{ij} of M , produce all the key-value pairs $(i, k), (M, j, m_{ij})$ for $k = 1, 2, \dots$, up to the number of columns of N . Similarly, for each element n_{jk} of N , produce all the key-value pairs $(i, k), (N, j, n_{jk})$ for $i = 1, 2, \dots$, up to the number of rows of M . As before, M and N are really bits to tell which of the two relations a value comes from.

The Reduce Function: Each key (i, k) will have an associated list with all the values (M, j, m_{ij}) and (N, j, n_{jk}) , for all possible values of j . The Reduce function needs to connect the two values on the list that have the same value of j , for each j . An easy way to do this step is to sort by j the values that begin with M and sort by j the values that begin with N , in separate lists. The j th values on each list must have their third components, m_{ij} and n_{jk} extracted and multiplied. Then, these products are summed and the result is paired with (i, k) in the output of the Reduce function.

You may notice that if a row of the matrix M or a column of the matrix N is so large that it will not fit in main memory, then the Reduce tasks will be forced to use an external sort to order the values associated with a given key (i, k) . However, in that case, the matrices themselves are so large, perhaps 10 20 elements, that it is unlikely we would attempt this calculation if the matrices were dense. If they are sparse, then we would expect many fewer values to be associated with any one key, and it would be feasible to do the sum of products in main memory.