Matrix Multiplication

A. Algorithm

Task 1: Map

Here all the input are read and key-value pairs are formed. Value for 'A' matrix will have row index and for 'B' matrix, coloumn index.

Eg A,2,4,17 -> [<A,2>,<4,17>]
B,5,8,31 -> [<B,8>,<5,31>]

Task 2: GroupBy

All the previous key value pairs are then grouped together by same key and are sorted on the basis of their index and all the list of A's rows and B's coloumns are formed.

Eg [<A,2>,{17,54,18,22}] where A[2][0]=17, A[2][1]=54, A[2][2]=18 and so on.

Task 3: FlatMapToPair

For every single tuple, all combinations of the each index is made and the previous list is made value.

Eg [<A,2>,{17,54,18,22}] -> [<2,0>,{17,54,18,22}]
[<2,1>,{17,54,18,22}]
[<2,2>,{17,54,18,22}]
[<2,3>,{17,54,18,22}] and so on. Similarly for B.

Task 4: ReduceByKey

Now the two tuple having same key i.e. one from A matrix and another from B will reduce. The ouput will be the product of both values for each i and the result will also be a list.

Eg [<2,0>,<{1,2,3,4},{1,2,3,4}>] = [<2,0>,{1,4,9,16}]

Task 5: Map

The list elements are then summed to form the final value for each C[i][j].

Eg [<2,0>,{1,4,9,16}] = [<2,0>,30]

Task 6: Sort

All the values are then sorted for the final ouput.

B. Computational And Communication Cost.

If we are considering matrix A[m x n] and matrix B[n x p] and k clusters.

Computational Cost :-

The total computation cost will be the sum total of all the computations at each mapper and reducer. The final reducer will have the computation of m*n*p which will be greater than computation at other mapper and reducer. Hence final computation cost will be of order **m*n*p**.

Communication Cost:-

Task Number	Computation cost
Task 1	m*n+n*p
Task 2	m+p
Task 3	2*m*p
Task 4	m*p
Task 5	m*p

Total : (m*n+n*p)+(m+p)+(2*m*p)+(m*p)+(m*p)