Homework

Week 11

Dr. Shirley Moore

Oscar Galindo

* Exercise 2.2.1: Suppose we execute the word-count MapReduce program described in this section on a large repository such as a copy of the Web. We shall use 100 Map tasks and some number of Reduce tasks.

1. Suppose we do not use a combiner at the Map tasks. Do you expect there to be significant skew in the times taken by the various reducers to process their value list? Why or why not?

Yes, I do expect that to happen since words are most likely not uniformly distributed throughout the internet. Instead, words like “like”, “a”, “the” etc. Are likely very repeated so if we had, say, 26 reducers the distribution will probably fall under one bucket assuming the grouping occurs based on the hashing of only the first letter.

1. If we combine the reducers into a small number of Reduce tasks, say 10 tasks, at random, do you expect the skew to be significant? What if we instead combine the reducers into 10,000 Reduce tasks?

For the entire internet the “lesser” buckets/reduce tasks the better because we could balance the load on all the reducers. As mentioned in the chapter, having too many reducer operations could potentially have a bigger skew since we would, likely, cause a reduce task to handle millions of entries while the other takes less time.

1. Suppose we do use a combiner at the 100 Map tasks. Do you expect skew to be significant? Why or why not?

I think the skew would not be as significant as in the case where we have way too many reduce tasks since we reflect more parallelism when we have map tasks with combiners.

* Exercise 2.5.1: 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?
  1. The matrix-vector multiplication algorithm of Section 2.3.2.

Input to the algorithm is matrix M and vector V hence communication cost is O(M/p+v/p), considering that we must communicate “p” pieces of the resulting stripes that are obtained in the algorithm.

* 1. The union algorithm of Section 2.3.6

O(r\*s), r and s being the sizes of the RDDs that will be joined by first mapping projections and then eliminating repeats

* 1. The aggregation algorithm of Section2.3.8

Map task: O(a\*b\*c) where a,b,c are the possible “descriptions” of the attributes A, B, C.

Reduce task: O(a\*b) where a,b are the possible “descriptions” of the attributes A, B

Algorithm: O(a\*b\*c)

* 1. The matrix multiplication algorithm of Section 2.3.10

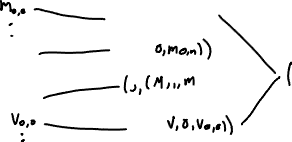
Map task: O(m3+n3) assuming m and n are the size of a single dimension of their respective arrays. O(m3+n3) is the time complexity since we take every mij element in the range of columns that equal in magnitude to the size of the rows in N. And for N we take all the elements Nijin the range 1…rows(N).

Reduce task: takes all created entries and performs additions based on the attribute “j” of the created tuples with the form ((i,k),(M,j, mij))

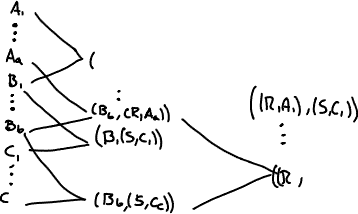
Algorithm: O(m3+n3)

* Exercise 2.6.1: Describe the graphs that model the following problems.

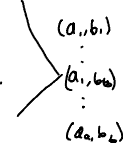
* 1. The multiplication of an n × n matrix by a vector of length n.



* 1. 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.



* 1. 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.



* Exercise with Spark

Attempted to run:

Graphical user interface, text, application

Description automatically generated

Got:

Text, timeline

Description automatically generated

In general, when I executed code in Spark I received many times the message that cache was not enough and it said it also failed in erasing some file that was supposed to be present. For this specific error I am not sure what was the error.