Summary of Chapter 2.5 The Communication Cost Model

This chapter mentioned some methods to evaluate the performance of algorithms implemented on a computing cluster which is acyclic workflow. the bottleneck of this method is moving data among tasks, such as transporting the outputs of Map tasks to their proper Reduce tasks.

If an algorithm is implemented in the acyclic network, the output of these tasks could be input of the Map tasks. Such as standard MapReduce algorithm, the output in MapReduce jobs cascaded, and other general algorithms.

The communication cost of an algorithm is defined as the sum of all the communication cost produced during computing basing on this algorithm. This cost is considered as an important cost to measure the quality of the algorithm. we do not consider the amount of time it takes each task to execute when estimating the running time of an algorithm.

There are some reasons why communication cost is important. Firstly, the task in each node is simple. The complexity is linear in the scale of data input. Secondly, communication speed is lower than CPU speed. There is a competition in many cluster architectures, which also would enhance the communication cost. As a result, the compute node can run the tasks on a received input element after it takes to deliver that element.

Only input data size is considered because the output data size of one task should be as same as the input data size of following task, unless this output is the result. If the output is so larger that it is more than the input size, it is necessary to implement aggregation to reduce the output size, and normally, it is executed in reducer. In this case, the result will be sent to another collection to implement this aggregation. So that the communication cost is always proportional to the computation.

An example is used to explain how to calculate the communication cost in , and the side of is , and the size of is . The sum of the communication costs for all the map tasks is .

On the other side, wall-clock time which is the time that parallel algorithms to finish tasks. We can optimize communication cost by adjusting task distribution on different computing nodes. However, this algorithm is takes high wall-clock time.

Meanwhile, In the cluster-computing environment, some methods are mentioned to analyse the communication cost. A general theory is introduced.

Firstly, certain attributes need to be selected in the join with more than two relations and their values would be hashed and assigned to some number of buckets.

Secondly, select the number of buckets for each attribute, and use the product of these numbers k, as the number of reducers that will be used.

And then identify each of the k reducers with a vector of bucket numbers. These vectors have one component for each of the attributes selected in the first step.

At last, send tuples of each relation to all those reducers where it might find tuples to join with. Other components of the vector are unknown, so it must be sent to reducers for all vectors having any value in these unknown components.

Another example is provided in this section. and R, S and T have sizes , respectively. And is the probability that an R-tuple and S-tuple agree on B, and also the probability that An S-tuple and a T-tuple agree on C.

If we join and first, Using MapReduce algorithm of above sample, the communication is . And the probability of is . Before get the input, the communication cost is . In sum, the total communication cost of the algorithm is .

Similarly, if we join and first, the communication cost is .

There is another way, which is use a single MapReduce job that joins the three relations together. In this case, we need more than one reducer to finish aggregation tasks.