Reading Homework 13 (Due Tuesday 5/01, 3:20PM)

Due May 1, 2018 at 3:25pm **Points** 11 **Questions** 10 **Time Limit** None

Instructions

Assigned Reading:

-- Beginning through Section 3.1
of https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf (https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf) (https://www.journaldev.com/8848/mapreduce-algorithm-example)

--https://spark.apache.org/docs/latest/rdd-programming-guide.html (https://spark.apache.org/docs/latest/rdd-programming-guide.html) (http://spark.apache.org/docs/latest/programming-guide.html),)

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	3 minutes	11 out of 11

Score for this quiz: **11** out of 11
Submitted May 1, 2018 at 1:24pm
This attempt took 3 minutes.

The following 6 questions are based on the MapReduce paper that was in the assigned reading.

Assume that we want to create a parallel query execution engine for SQL queries using MapReduce. The input to each MapReduce job is one or more tables, split by tuple. Therefore, the key-value pairs input to the Map task is the following: The key is the value of the primary key attribute(s) of each tuple, and the value is the complete tuple (including the primary key).

Question 1 1 / 1 pts

Let's say that we want to implement a parallel version of the linear search described in section 12.3.1 of your textbook. E.g., we want to implement the following query:

SELECT * from X where att1 = 8;

In which part of a MapReduce job should we implement this linear search?

- It can only be done in the Map function
- It can only be done in the Reduce function
- It can only be done during the repartitioning step between Map and Reduce

Correct!

- [x] It can be done in either Map or Reduce, but less data will be sent over the network if done in Map
- It can be done in either Map or Reduce, but less data will be sent over the network if done in Reduce

Question 2

1 / 1 pts

Assume we have the query that we want to implement with a single MapReduce job:

SELECT att1, max(att2) FROM X GROUP BY att1;

In which phase is most natural to implement the GROUP BY part of the query?

- Map
- Reduce

Correct!

- [x] The repartitioning step between Map and Reduce

Question 3

1 / 1 pts

Same query as previous question.

In which phase is most natural to implement the max(att2) part of the query?

- Map
- Reduce

Correct!

- The repartitioning step between map and reduce

Question 4

2 / 2 pts

One way to do a join of two datasets in MapReduce is the following:

(1) Write a map function that goes through both tables that need to be joined. For each tuple from each table, it produces the following key-value pair: the key is the join attribute extracted from the tuple, and the value is the original tuple, annotated with the identifier of whichever table it came from.

(2) The repartitioning phase that happens between the Map and Reduce steps ensures that all key-value pairs with the same key end up being processed by the same Reduce worker. Therefore, since the join attribute is the key of all the key-value pairs produced by the Map function, all tuples with the same value of the join attribute will end up being processed by the same Reduce worker.

(3) The Reduce function then implements a function that joins tuples from the first table with tuples from the second table that have the same value of the join attribute.

Of the four parallel join algorithms described in Chapter 18.5.2 of your textbook, which ones can be implemented using the above steps in MapReduce (choose the best **TWO** options below)

Correct!

- [x] Partitioned Join
- [ ] Fragment-and-Replicate Join

Correct!

- [x] Partitioned Parallel Hash Join
- [ ] Parallel Nested-Loop Join

Question 5

1 / 1 pts

If

(1) The Map function extracts the attribute from each tuple that we want to sort a dataset/table by
(2) The key-value pairs produced by Map are not sorted
(3) The key-value pairs input to Reduce are sorted by key prior to execution of the Reduce function
(4) The Reduce function is just an identity function (output is the same as input)
(5) A **range partitioning** function is used to decide how to allocate which keys each Reduce worker receives (Section 4.1 of the paper, which was not in the assigned reading, says that while hash partitioning this intermediate key space is the default, any partitioning function can be used).

Which parallel sort algorithm from Section 18.5.1 of your textbook is being implemented?

Correct!

- (•) Range-Partitioning Sort
- ( ) Parallel External Sort-Merge
- ( ) None of the above. This solution does not sort the data.

Question 6

1 / 1 pts

Same question as before. (1) - (4) stay the same as the previous question, but (5) changes to the following:

(5) A **hash partitioning** function is used to decide how to allocate which keys each Reduce worker receives

Which parallel sort algorithm from Section 18.5.1 of your textbook is being implemented?

- Range-Partitioning Sort
- Parallel External Sort-Merge

Correct!

- (•) None of the above. This solution does not sort the data.

The following 4 questions are based on the reading from the Spark documentation.

Question 7

1 / 1 pts

SparkContext.___ lets you read a directory of containing many files, returning an RDD containing (filename, content) pairs.

Correct!

wholeTextFiles

orrect Answers

wholeTextFiles

Question 8

1 / 1 pts

"cogroup" is equivalent to the relational natural join operation.

- True
- False

Correct!

Question 9

1 / 1 pts

"flatMap" typically requires a shuffle (a repartitioning type operation similar to what is done between the Map and Reduce steps of a MapReduce job from the other reading), i.e., data needs to be redistributed across the partitions.

- True
- False

Correct!

Question 10

1 / 1 pts

accumulators are always computed eagerly, and require changing the lazy evaluation model of Spark.

- True
- False

Correct!

Quiz Score: **11** out of 11