**Hadoop Streaming**

**Question 1: Select the facts about a combiner:**

It can significantly speed up the MapReduce job

Correct: Yes, because it can reduce the amount of records passed from mappers to reducers

Can be implemented in any language and specified in a ‘-combiner’ option in Hadoop Streaming command

Correct: Yes, a combiner is just a script or a program like a mapper and reducer are

It can be the same as a reducer in special cases

Correct: Yes, if the reducer implements commutative and associative function it can be used as combiner

**Question 2: How can a partitioner be implemented and what should be specified in ‘-partitioner’ option in a Hadoop Streaming command?**

Only in Java; specify java class in -partitions option

Correct: That's right, only Java classes are allowed as a partitioner

**Question 3: Select the correct statements about a partitioner:**

It is used to calculate a reducer index for each (key, value) pair

Correct: Yes, a partitioner determines a reducer for each record

Depends on a ‘key’ field (i.e. on the field the intermediate data is sorted) or on a subset of the ‘key’ fields

Correct: Yes, the main requirement is the same reducer for equal keys. So the reducer should depend on a 'key' field or on a part of it.

Standard ‘KeyFieldBasedPartitioner’ has similar options to the Unix ‘sort’ utility

Correct: Yes, look at the KeyFieldBasedPartitioner documentation

**Question 4: Select the correct statements about a comparator:**

Standard ‘KeyFieldBasedComparator’ has similar options to the Unix ‘sort’ utility

Correct: Yes, see KeyFieldBasesComparator documentation

**Question 5: In what cases should speculative execution (of mapper, for example) be turned off?**

If the mapper has a side-effect

Correct: Yes, if mapper has a side-effect (it updates a database, requests an outer service), speculative execution should be turned off to avoid double work

**Question 6: Select the facts about a speculative execution:**

Can be the reason of a KILLED tasks status

Correct: Yes, tasks (or more precisely, attempts) which were successful on one node were killed on the others

It can speed up the MapReduce job

Correct: Yes, it can speed up by running tasks on more productive nodes

**Question 7: Select the facts about a compression:**

A compression is a trade-off between CPU utilization, disk usage and ability of archives to be splitted by Hadoop

Correct: Yes, all of these are significant for choosing a compression format

A compression can be specified both for intermediate and for output data

Correct: Yes, see 'mapreduce.map.output.compress' 'mapreduce.output.fileoutputformat.compress' parameters here

Bzip2 format is splittable, i.e. one bzip2 archive can be processed by several mappers in parallel

Correct: Yes, bzip2 is splittable, because it is possible to determine compressed blocks bounds without reading the file header

**Question 8: What phase in MapReduce paradigm is better for aggregating records by key?**

Reduce

Correct: Yes, a reducer gets records sorted by key, so they are suitable for aggregation

**Question 9: What map and reduce functions should be used (in terms of Unix utilities) to select the only unique input records?**

map=’cat’, reduce=’uniq’

Correct: Yes, because 'uniq' on the sorted input records gives the required result

map=’uniq’, reduce=’uniq’

Correct: Yes, 'uniq' on the Map phase in some cases reduces the amount of records, 'uniq' on the Reduce phase gets the sorted records and solves the task

**Question 10: What map and reduce functions should be used (in terms of Unix utilities) to select only the repeated input records?**

map=’cat’, reduce=’uniq -d’

Correct: Yes, mappers pass all the records to reducers and then 'uniq -d' on the sorted records solves the task

**Question 11: In Hadoop Streaming mapper is run on:**

Stream of input records

Correct: Yes, a mapper reads input records one by another from stdin

**Question 12: What phase of MapReduce is more suitable for this code?**

**!/usr/bin/env python**

**import sys**

**import random**

**random.seed(100)**

**probability = float(sys.argv[1])**

**for line in sys.stdin:**

**if random.random() <= probability:**

**print line.strip()**

Map

Correct: Yes, it filters input records (makes a sample) without the requirement that input records are sorted

**Question 13: How can the Reduce phase in Hadoop Streaming be omitted?**

Set number of reducers to 0

Correct: Yes, that turns off the Reduce phase and the output of the Map phase becomes the output of the job