Question 1

Correct Note de 1,00 sur 1,00

Select the correct statements
(Bareme: -25% per answer incorrectly selected. No point nor penalty per statement left unselected).

a. monrdd.flatmap(f) contains as many items as monrdd

b. A Spark stage in which each partition depends on a single parent partition will be **computed on the same node as its parent**, without exchanging data through the network : there is **no need for shuffle** in this case. ✔

c. monrdd.map(f) contains as many items as monrdd ✔

d. monrdd.persist() is an action that **stores immediately** monrdd **in memory** or **on disk**, depending on parameters specified inside the parentheses.

e. Suppose we write in the pyspark shell:
rdd2 = monrdd.map(f)
if the first item in rdd2 is 0 and function f is lambda x : 3/x, there is no reason why this instruction should raise an error message at this point of the program (i.e., after typing and sending the instruction).

✔ True since the transformation is evaluated only when triggered by an action, so not at this point of the program.

f. df = sc.textFile('aa.txt') returns a **Dataframe** named **df.**

Votre réponse est correcte.

Les réponses correctes sont :
monrdd.map(f) contains as many items as monrdd, Suppose we write in the pyspark shell:
rdd2 = monrdd.map(f)
if the first item in rdd2 is 0 and function f is lambda x : 3/x, there is no reason why this instruction should raise an error message at this point of the program (i.e., after typing and sending the instruction).,

A Spark stage in which each partition depends on a single parent partition will be **computed on the same node as its parent**, without exchanging data through the network : there is **no need for shuffle** in this case.

Question 2

Partiellement correct Note de 0,75 sur 1,00

Select the correct statements (or at least according to our lectures).

(-30% per incorrect selection, no penalty for an item not selected but no point either).

- [ ] a. Dataframe are strictly better than RDDs for structured data.
- [x] b. RDDs are not a completely obsolete API and are still a reasonable choice for unstructured data, but not for structured data. ✔
- [ ] c. It makes sense to evaluate SQL queries on an arbitrary RDD.
- [x] d. orderBy(), groupBy(), select()... are methodes supported by Dataframes but not RDDs. ✔
- [x] e. A Dataframe can be converted to an RDD and vice-versa. ✔
- [ ] f. What makes **RDD** faster than **Dataframe** is that **RDD** stores data in **column** format, and therefore **compresses** data more efficiently.

g. Aggregates can be computed faster on unstructured data like RDDs (rather than on Dataframes) because then we do not have to parse columns.

Votre réponse est partiellement correcte.

Vous en avez sélectionné correctement 3.
Les réponses correctes sont :
RDDs are not a completely obsolete API and are still a reasonable choice for unstructured data, but not for structured data.,

Dataframe are strictly better than RDDs for structured data.,

A Dataframe can be converted to an RDD and vice-versa.,

orderBy(), groupBy(), select()... are methodes supported by Dataframes but not RDDs.

Question 3

Correct Note de 1,00 sur 1,00

Which of the following structures store data as named columns ? (bareme: all-or-nothing)

- [x] a. Dataframe
- [ ] b. RDD
- [x] c. Dataset

Votre réponse est correcte.

Les réponses correctes sont :
Dataframe,

Dataset

Question 4

Incorrect Note de 0,00 sur 1,00

Select all correct statements (bareme: all-or-nothing).

- [ ] a. In Spark, operations can process write-only shared variables (shared between workers). The current state of these variables is essentially stored on the workers (rather than the driver) in practice.
- [x] b. In Spark, operations can process efficiently read-only shared variables (shared between workers). The current state of these variables is essentially stored on the workers (rather than the driver) in practice. ✔
- [ ] c. In Spark, operations can process efficiently read-only shared variables (shared between workers). The current state of these variables is essentially stored on the driver (rather than the workers) in practice.
- [ ] d. In Spark, operations can process write-only shared variables (shared between workers). The current state of these variables is essentially stored on the driver (rather than the workers) in practice.

- e. A priori,
  counter =...
  rdd.foreach(x => counter += f(x,counter)) looks like a wrong use of the shared variable counter

Votre réponse est incorrecte.

Les réponses correctes sont :
In Spark, operations can process write-only shared variables (shared between workers). The current state of these variables is essentially stored on the driver (rather than the workers) in practice., In Spark, operations can process efficiently read-only shared variables (shared between workers). The current state of these variables is essentially stored on the workers (rather than the driver) in practice., A priori,
counter =...
rdd.foreach(x => counter += f(x,counter)) looks like a wrong use of the shared variable counter

Question 5

Correct Note de 1,00 sur 1,00

How does spark call an operation that takes as input an rdd and returns something which is not an rdd, generally returning this value to the "driver".

(answer using a single world, in singular and lowercase)

Réponse :

action ✔

La réponse correcte est : action