Large Datasets for Scientific Applications (1TD268): A3, Part 2
Mei Wu

Section C – Concepts in Apache Spark and Distributed Computing

1. The print statement does not appear when she calls the function because she's performing a flatMap which is a transformation that creates a new RDD.
2. Immutable data in Spark partitions are advantageous because it frees the user from having to track/log the mutated changes and in the event of failures, recovery is easily attained because the initial state of an RDD is reused. Additionally, it allows the infrastructure to perform in parallel and efficiently without changing the data, rather using transformations to modify and actions to output results.
3. *.collect()* is an action that results Spark to save native object to memory in the respective language and calling this action will take up space in memory.
4. RDDs are resilient in the sense that a defined object of any format is immutable and any transformations or actions performed are done in a new RDD. In the event of a failure, the original RDD is rebuilt to continue the task.

Section D – Essay Question

Recommendations:

1. If her Spark architecture is based on unstructured RDDs, I would suggest to convert them DataFrames and SQL because in sticking with RDDs, she will have to manually write code to modify them.
2. It would be helpful to dynamically adjust resources based on the workload using a course-grained cluster manager, Apache Mesos.
3. If joins are frequently performed, bucketing would help to ensure that the data is well partitioned according to those values. It will help prevent a shuffle before a join resulting in an increase of speed in accessing the data.
4. Reduce the usage of UDFs because they are expensive tasks, forcing the representation of data as objects in the JVM and sometimes it does this multiple times per record in a query. Instead, try to use Structured APIs because they can perform transformations proficiently.