# Big Data Hadoop Training

Session 13 Assignment 1 Solution:

**1. What is RDD?**

**A**. **Resilient Distributed Dataset**

* RDD’s in spark are immutable distributed collection of objects.
* Each RDD is split into multiple partitions, which may be computed on different nodes of the cluster.
* The RDDs can contain any type of Python, Java or Scala objects, including user defined classes.
* They are not actual data, but they are Objects, which contain information about data residing on the cluster. The RDDs try to solve these problems by enabling fault tolerant, distributed In-memory computations.
* **Features:**
  + In Spark, data is stored in partitions of the RDDs and store in worker nodes (datanodes) which are computed in parallel across all the nodes.
  + RDDs load the data for us and are resilient, which means they can be recomputed.
  + RDDs perform two types of operations: **Transformations**, which creates a new dataset from the previous RDD and **Actions**, which return a value to the driver program after performing the computation on the dataset.
  + RDDs keeps track of Transformations and check them periodically. If a node fails, it can rebuild the lost RDD partition on the other nodes, in parallel
* **Properties:**
* List of partitions
* List of dependencies on parent RDDs
* Function to compute a partition given parents
* Optional list of preferred locations to compute each split (e.g. block locations for an HDFS file)
* Optional Partitioner for key-value RDDs (e.g. to say that the RDD is hash-partitioned)

**2. Define Partitions.**

**A.** A partition (aka split) is a logical chunk of a large distributed data set. Spark manages data using partitions that helps parallelize distributed data processing with minimal network traffic for sending data between executors.

By default, Spark tries to read data into an RDD from the nodes that are close to it. Since Spark usually accesses distributed partitioned data, to optimize transformation operations it creates partitions to hold the data chunks.

There is a one-to-one correspondence between how data is laid out in data storage like HDFS or Cassandra (it is partitioned for the same reasons).

Features:

* size
* number
* partitioning scheme
* node distribution
* repartitioning

By default, a partition is created for each HDFS partition, which by default is 64MB.

In general, smaller/more numerous partitions allow work to be distributed among more workers, but larger/fewer partitions allow work to be done in larger chunks, which may result in the work getting done more quickly as long as all workers are kept busy, due to reduced overhead. Increasing partitions count will make each partition to have less data (or not at all!)

**3. What operations does RDD support?**

**A.** RDDs perform two types of operations:

* **Transformations**, which creates a new dataset from the previous RDD.
* **Actions**, which return a value to the driver program after performing the computation on the dataset.

RDDs keeps track of Transformations and check them periodically. If a node fails, it can rebuild the lost RDD partition on the other nodes, in parallel

**4. What do you understand by Transformations in Spark?**

**A.** “Transformations” in Spark are functions applied on RDD, resulting in a new RDD. It does not execute until an action occurs. map() and filter() are examples of “transformations”, where the former applies the function assigned to it on each element of the RDD and results in another RDD. The filter() creates a new RDD by selecting elements from the current RDD that pass the function argument.

**5. Define Actions.**

**A.** Actions are the results of RDD computations or transformations. “Action” take back the data from the RDD to the local machine. After an action is performed, the data from RDD moves back to the local machine.

For example, ***fold***() is an action that implements the function passed again and again until only one value is left. Some other examples of actions include reduce, collect, first, and take.