**ASSIGNMENT 16.2**

PROBLEM STATEMENTS:

1) **Pen down the limitations of MapReduce**

Solutions:

The limitations of MapReduce operations are:

(i) it’s based on disk based computing.

(ii) Suitable for a single pass computations - not iterative computations

(iii) Needs a sequence of MapReduce jobs to run iterative task.

(iv) Needs integration with several other frameworks/tools to solve bigdata usecases.

2) **What is RDD? Explain few features of RDD?**

Solutions:

RDD is the primary abstraction in spark and it is the core of Apache Spark.

Features of RDD:

(i) Resilient, i.e. fault-tolerant with the help of RDD lineage graph and so able to recompute missing or damaged partitions due to node failures.

(ii) Distributed with data residing on multiple nodes in a cluster.

(iii) Dataset is a collection of partitioned data with primitive values or values of values, e.g. tuples or other objects.

(iv) In-Memory, i.e. data inside RDD is stored in memory as much (size) and long (time) as possible.

3) **List down few RDD operations and explain each of them.**

Solutions:

Spark RDD consists of two following operations:

(i) Transformation.

(ii) Action.

So, the **Transformation** consists of:

(i) map: By applying a map() function, it returns a new rdd to each element of the assigned rdd.

Example:

val x= sc.parallelize(Array("b", "a", "c"))

val y= x.map(z => (z,1))

(ii) filter: By applying the filter() function, it returns a new rdd only if the condition satisfies.

Example:

valx= sc.parallelize(Array(1,2,3))

valy= x.filter(n => n%2 == 1)

**Action** consists of the following operations:

(i) reduce:

Aggregate all the elements of the RDD by applying a reduce() function pairwise to elements and partial results, and returns a result to the driver.

Example:

x = sc.parallelize([1,2,3,4])

y= x.reduce(lambda a,b: a+b)

Output:

x:[1,2,3,4]

y:10

(ii) max:

By applying the map() function, it returns the maximum item in the rdd.

Example:

valx= sc.parallelize(Array(2,4,1))

valy= x.max

Output:

x: [2, 4, 1]

y:4