*Spark*

**A C A D G I L D Page 1**

Session 3: Spark Core

Assignment 1*Spark*

**A C A D G I L D Page 2**

*Session 3: Spark Core*

*Assignment 1*

**Table of Contents**

1. Introduction ........................................................................................................................................ 3

2. Objective ............................................................................................................................................. 3

3. Prerequisites ....................................................................................................................................... 3

4. Associated Data Files ............................................................................................................................ 3

5. Problem Statement ............................................................................................................................... 3

6. Approximate Time to Complete Task ................................................................................................... 3 *Spark*

**A C A D G I L D Page 3**

**1. Introduction**

In this assignment you need to answer the given questions.

**2. Objective**

This assignment will help you to understand Big Data basics.

**3. Prerequisites**

None

**4. Associated Data Files**

N/A

**5. Problem Statement**

1. What is RDD?

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.

• 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.

2. Define Partitions.

Partitioning is nothing but dividing it into parts. If you talk about partitioning in distributed system, we can define it as the division of the large dataset and store them as multiple parts across the cluster.

Spark works on data locality principle. Worker nodes takes the data for processing that are nearer to them. By doing partitioning network I/O will be reduced so that data can be processed a lot faster.

In Spark, operations like co-group, groupBy, groupByKey and many more will need lots of I/O operations. In this scenario, if we apply partitioning, then we can reduce the number of I/O operations rapidly so that we can speed up the data processing.

To divide the data into partitions first we need to store it. Spark stores its data in form of RDDs.

scala> val someRDD = sc.parallelize(1 to 100, 4)

3. What operations does RDD support?

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.

4. What do you understand by Transformations in Spark?

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

Map,Filter,FlatMap,GroupBy etc.

5. Define Actions.

An action is one of the ways of sending data from Executer to the driver. Executors are agents that are responsible for executing a task. While the driver is a JVM process that coordinates workers and execution of the task. Some of the actions of Spark are:

reduce() take()

collect() takeSample()

count() takeOrdered()

first() countByKey()

foreach()

**6. Approximate Time to Complete Task**

30 min