# Session 16 Assignment 2

1. List down the limitations of mapreduce

It's based on disk based computing.

Suitable for single pass computations - not iterative computations.

Needs a sequence of MR jobs to run iterative tasks.

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

Apache Storm for stream data processing

Apache Mahout for machine learning

2. What is RDD? list down few features of RDD.

Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects.

Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster. RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.

Formally, an RDD is a read-only, partitioned collection of records. RDDs can be created through deterministic operations on either data on stable storage or other RDDs. RDD is a fault-tolerant collection of elements that can be operated on in parallel.

Features of RDD

○ 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.

○ Distributed with data residing on multiple nodes in a cluster.

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

3. List down few spark rdd operations and explain each of them

//Creating RDD from text file

val fileContent = sc.textFile("input.txt", 5)

The output of fileContent.getNumPartitions, will be 5

//Read the input file

sc.textFile("input.txt").flatMap(x => x.split(" "))

This will create a RDD of strings, where each element is a word

count - returns the number of elements in the RDD

countApprox(long timeout, double confidence) - Approximate version of

count() that returns a potentially incomplete result within a timeout, even if not all tasks have finished.

countByValue(scala.math.Ordering<T> ord) - Return the count of each unique value in this RDD as a local map of (value, count) pairs.