**Session 16 Assignment 2**

* Limitations of MapReduce
* It is disk-based computing, which means it's very slow and not practical for real time jobs.
* Doesn't work well with iterative computation because it doesn't store any data in memory to be worked on. The developer must create multiple MapReduce jobs to accomplish one iterative process; just the number of IO operations will significantly contribute to slowness and burdensome on the network.
* The process of testing code is not convenient, as is also the process of using user-defined functions.
* Many Big Data problems don't fit the MapReduce format, which means other big data tools must be used to gather with it to solve most problems.
* Also, others professionals have to learn new (or something similar to what they already know) programming language to be able to use it; otherwise, they will have to use another tool altogether to solve their problems.
* What is RDD. Resilient Distributed Datasets---

It is Spark's fundamental primary data abstraction. It is a fault tolerant collection of data that is immutable, distributed, lazily evaluated, type inferred and cacheable.

* Features,
* Every RDD that is created is immutable. This means it can't be changed, so it's value is dependable at any instance of computation.
* Distributed which means it's broken up based on a logic (partitions) for the sake of computation in distributed environments thereby taking advantage of the benefits of distributed file systems (chiefly parallelism) .
* Because they are immutable, any transformation of an rdd creates a new RDD, which will be problematic in terms of memory use. To solve this problem, transformations on an RDD is lazily evaluated. which means instead of performing transformations, the transformations themselves are logged until an action is required. Then instead of creating several RDDs to complete the action, the transformations are evaluated in succession until the result is provided for the action.
* Type inferred has to do with the compiler's capability of determining the data types of Rdds.
* Cacheable means the whole data can be kept in memory for quicker access during computation.
* List down a few Spark RDD operations and explain each of them.
* Transformations---

Any operation that is done on an RDD to get another RDD as a result.

e.g. filter, union, etc.

* Filter for example, results in an RDD that contains only the elements that meet the filter argument.
* Union on the other hand takes an argument which is another RDD and results in a new RDD that contains elements from both RDDs.
* Actions

Any operation that returns a result to the driver program, or writes in a storage, and kicks off a computation.

e.g. count, first, collect, etc

* Count returns the number of elements in an RDD.
* First returns the first element in an RDD.
* Collect retrieves the complete list of elements from an RDD.