19th May - Task2

**Problem:**

Explore about the java and Scala Future objects used for concurrency of partitions in executors.

**Walkthrough:**

We know that spark used rdd to partition a collection of data and parallelly processes it. But it does not signify that the individual tasks does not depend on each other and are independent.

Using future objects we can do parallel programming where a unit of work is allowed to run separately from the primary application thread. When the work is complete, it notifies the main thread about the completion or failure of the worker thread.

**Scala Future Objects:**

A Future gives you a simple way to run a job inside your spark application concurrently.

Scala support Thread pools, Thread pools is nothing but tarting a thread required us to allocate a memory region for its call stack and context switch from one Thread to another. Which consumes much more time than work in the concurrent task. For this reason, most concurrency frameworks have facilities that maintain a set of Threads in a waiting state and start running when concurrently executable work tasks become available, and we call this as Thread Pools.

In Scala, by importing the packages mentioned as:

* import scala.concurrent.ExecutionContext.Implicits.global
* import scala.concurrent.Future
* import scala.Success
* import scala.Failure

**Snapshots:**