**What is Action in Spark?**

Actions are RDD’s operation, that value returns back to the spar driver programs, which kick off a job to execute on a cluster. Transformation’s output is an input of Actions. reduce, collect, takeSample, take, first, saveAsTextfile, saveAsSequenceFile, countByKey, foreach are common actions in Apache spark.

###### What is Map and flatMap in Spark?

The map is a specific line or row to process that data. In FlatMap each input item can be mapped to multiple output items (so the function should return a Seq rather than a single item). So most frequently used to return Array elements.

##### What are broadcast variables?

Broadcast variables let programmer keep a read-only variable cached on each machine, rather than shipping a copy of it with tasks. Spark supports 2 types of shared variables called broadcast variables (like Hadoop distributed cache) and accumulators (like Hadoop counters). Broadcast variables stored as Array Buffers, which sends read-only values to work nodes.

###### What is Transformations?

The transformations are the functions that are applied on an RDD (resilient distributed data set). The transformation results in another RDD. A transformation is not executed until an action follows.  
The example of transformations are:

1. map() – applies the function passed to it on each element of RDD resulting in a new RDD.
2. filter() – creates a new RDD by picking the elements from the current RDD which pass the function argument.

###### Q37) What are Actions?

An action brings back the data from the RDD to the local machine. Execution of an action results in all the previously created transformation. The example of actions are:

* reduce() – executes the function passed again and again until only one value is left. The function should take two argument and return one value.
* take() – take all the values back to the local node form RDD.

###### What are Accumulators in Spark?

Spark of-line debuggers called accumulators. Spark accumulators are similar to Hadoop counters, to count the number of events and what’s happening during job you can use accumulators. Only the driver program can read an accumulator value, not the tasks.

### **Define Actions in Spark.**

In Spark, an action helps in bringing back data from an RDD to the local machine. They are RDD operations giving non-RDD values. The reduce() function is an action that is implemented again and again until only one value if left. The take() action takes all the values from an RDD to the local node.

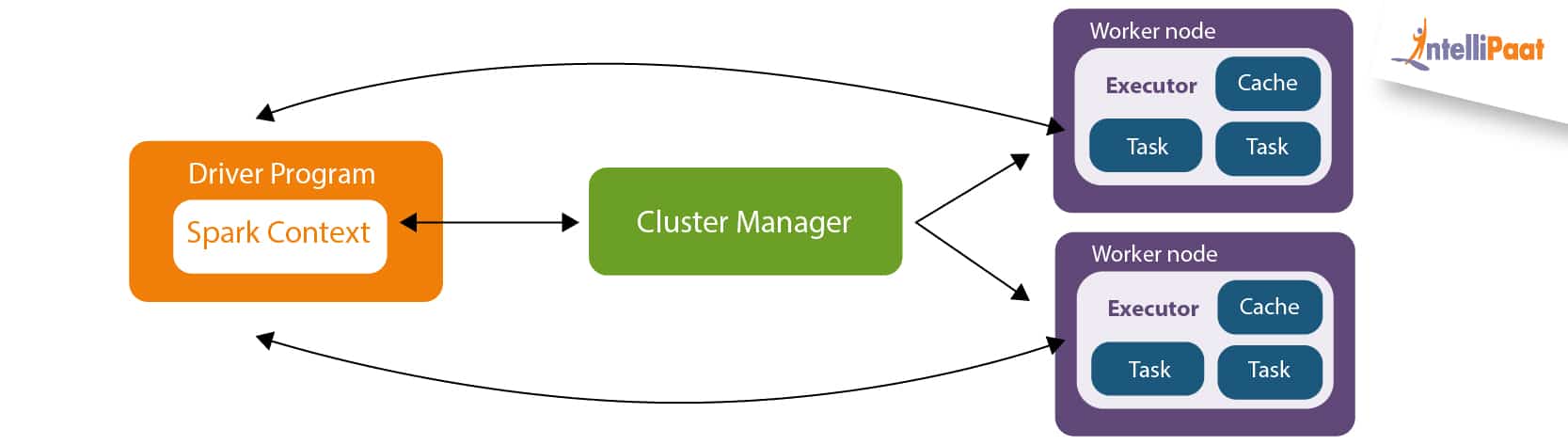
### **Name the types of Cluster Managers in Spark.**

The Spark framework supports three major types of Cluster Managers.

* **Standalone:** A basic Cluster Manager to set up a cluster
* **Apache Mesos:** A generalized/commonly-used Cluster Manager, running Hadoop MapReduce and other applications
* **YARN:** A Cluster Manager responsible for resource management in Hadoop

## Working of the Apache Spark Architecture

The basic Apache Spark architecture is shown in the figure below:

  
Driver Program in the Apache Spark architecture calls the main program of an application and creates SparkContext. A SparkContext consists of all the basic functionalities. Spark Driver contains various other components such as DAG Scheduler, Task Scheduler, Backend Scheduler, and Block Manager, which are responsible for translating the user-written code into jobs that are actually executed on the cluster.

Spark Driver and SparkContext collectively watch over the job execution within the cluster. Spark Driver works with the Cluster Manager to manage various other jobs. Cluster Manager does the resource allocating work. And then, the job is split into multiple smaller tasks which are further distributed to worker nodes.

Whenever an RDD is created in the SparkContext, it can be distributed across many worker nodes and can also be cached there.

Worker nodes execute the tasks assigned by the Cluster Manager and return it back to the Spark Context.

An executor is responsible for the execution of these tasks. The lifetime of executors is the same as that of the Spark Application. If we want to increase the performance of the system, we can increase the number of workers so that the jobs can be divided into more logical portions.