**1. How are worker, executor and task related to each other?**

Master – Responsible for managing the resources across worker

Worker – Slave daemon in Spark responsible for computation

Workers contain executors for driving out the computations.

**Worker nodes are machines that run executors**

Host one or multiple Workers

One JVM (= 1 UNIX process) per Worker

Each Worker can spawn one or more Executors

**Executors run tasks**

Run in child JVM (= 1 UNIX process)

Execute one or more task using threads in a ThreadPool.

**Task**

Unit of work to execute, in an executor thread

Unlike MR, there is no “map” vs “reduce” task

Each task, either partitions its output for “shuffle”, or sends the output back to the driver

**2. What are the key features of Spark?**

•Allows Integration with Hadoop and files included in HDFS.

•Spark has an interactive language shell as it has an independent Scala (the language in which Spark is written) interpreter.

•Spark consists of RDD’s (Resilient Distributed Datasets), which can be cached across computing nodes in a cluster.

•Spark supports multiple analytic tools that are used for interactive query analysis , real-time analysis and graph processing

•Spark provides advanced analytic options like graph algorithms, machine learning, streaming data, etc

•It has built-in APIs in multiple languages like Java, Scala, Python and R

•It has good performance gains, as it helps run an application in the Hadoop cluster ten times faster on disk and 100 times faster in memory.

**3. What is Spark Driver?**

Spark Driver is the program that runs on the master node of the machine and declares transformations and actions on data RDDs. In simple terms, driver in Spark creates SparkContext, connected to a given Spark Master.

The driver also delivers the RDD graphs to Master, where the standalone cluster manager runs.

**4. What are the benefits of Spark over MapReduce?**

Due to the availability of in-memory processing, Spark implements the processing around 10-100x faster than Hadoop MapReduce. MapReduce makes use of persistence storage for any of the data processing tasks.

Unlike Hadoop, Spark provides in-built libraries to perform multiple tasks form the same core like batch processing, Steaming, Machine learning, Interactive SQL queries. However, Hadoop only supports batch processing.

Hadoop is highly disk-dependent whereas Spark promotes caching and in-memory data storage.

Spark is capable of performing computations multiple times on the same dataset. This is called iterative computation while there is no iterative computing implemented by Hadoop.

**5. What is Spark Executor?**

When SparkContext connect to a cluster manager, it acquires an Executor on nodes in the cluster. Executors are Spark processes that run computations and store the data on the worker node. The final tasks by SparkContext are transferred to executors for their execution.