**Apache Spark and Hadoop.**

**1.Real-time analysis**

**Apache Spark**– It can process real time data i.e. data coming from the real-time event streams at the rate of millions of events per second, e.g. Twitter data for instance or Facebook sharing/posting. Spark’s strength is the ability to process live streams efficiently.

**Hadoop MapReduce** – MapReduce fails when it comes to real-time data processing as it was designed to perform batch processing on voluminous amounts of data.

**2.Speed:**

**Apache Spark**– Spark is lightning fast cluster computing tool. Apache Spark runs applications up to 100x faster in memory and 10x faster on disk than Hadoop. Because of reducing the number of read/write cycle to disk and storing intermediate data in-memory Spark makes it possible.

**Hadoop MapReduce**– MapReduce reads and writes from disk, as a result, it slows down the processing speed.

**3.Fault tolerance**

**Apache Spark**– Spark is fault-tolerant. As a result, there is no need to restart the application from scratch in case of any failure.

**Hadoop MapReduce**– Like Apache Spark, MapReduce is also fault-tolerant, so there is no need to restart the application from scratch in case of any failure.

**4.Security**

**Apache Spark**– Spark is little less secure in comparison to MapReduce because it supports the only authentication through shared secret password authentication.

**Hadoop MapReduce**– Apache Hadoop MapReduce is more secure because of Kerberos and it also supports Access Control Lists (ACLs) which are a traditional file permission model.

**5.Easy to Manage**

**Apache Spark**– Spark is capable of performing batch, interactive and Machine Learning and Streaming all in the same cluster. As a result makes it a complete data analytics engine. Thus, no need to manage different component for each need. Installing Spark on a cluster will be enough to handle all the requirements.

**Hadoop MapReduce**– As MapReduce only provides the batch engine. Hence, we are dependent on different engines. For example- Storm, Giraph, Impala, etc. for other requirements. So, it is very difficult to manage many components.

**6. Difficulty:**

**Apache Spark**– Spark is easy to program as it has tons of high-level operators with RDD – Resilient Distributed Dataset.

**Hadoop MapReduce**– In MapReduce, developers need to hand code each and every operation which makes it very difficult to work.