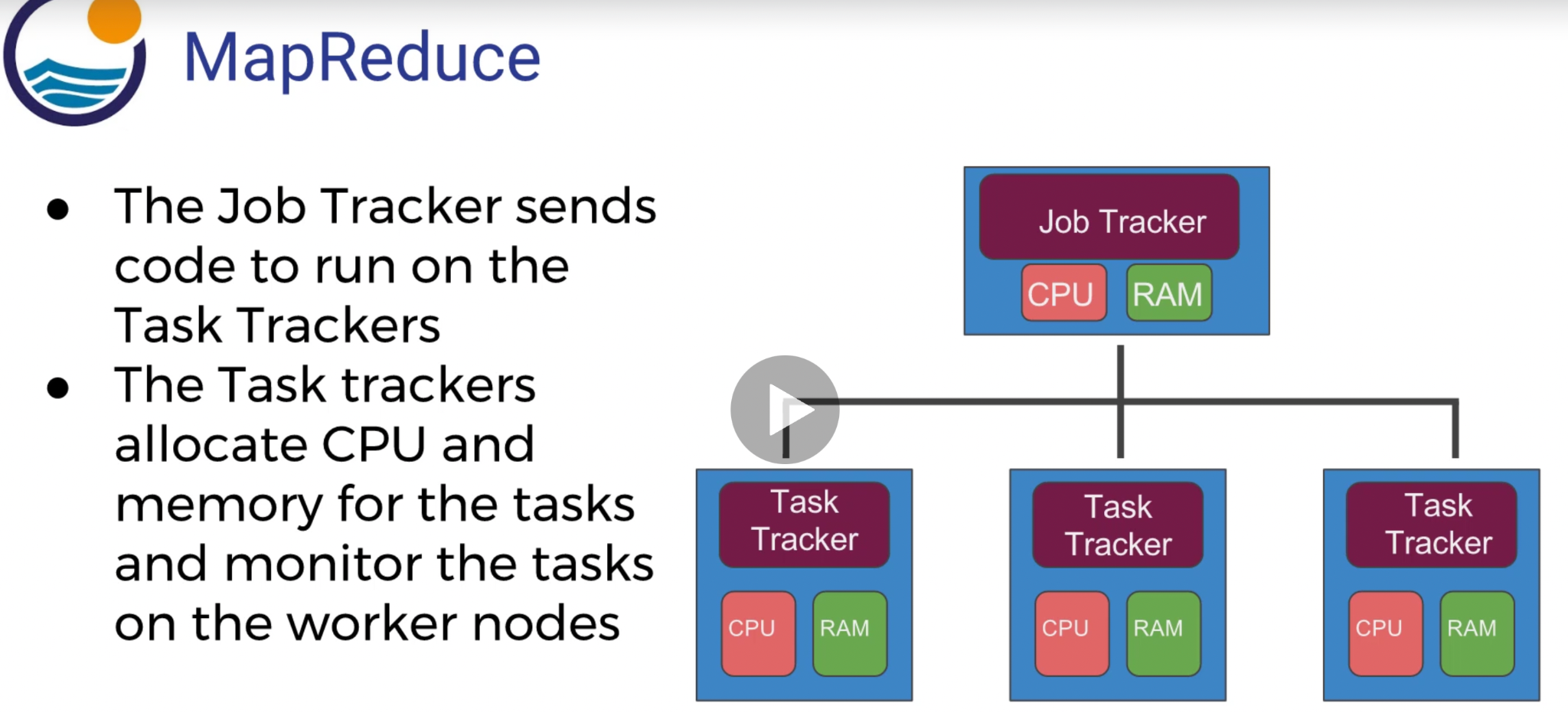
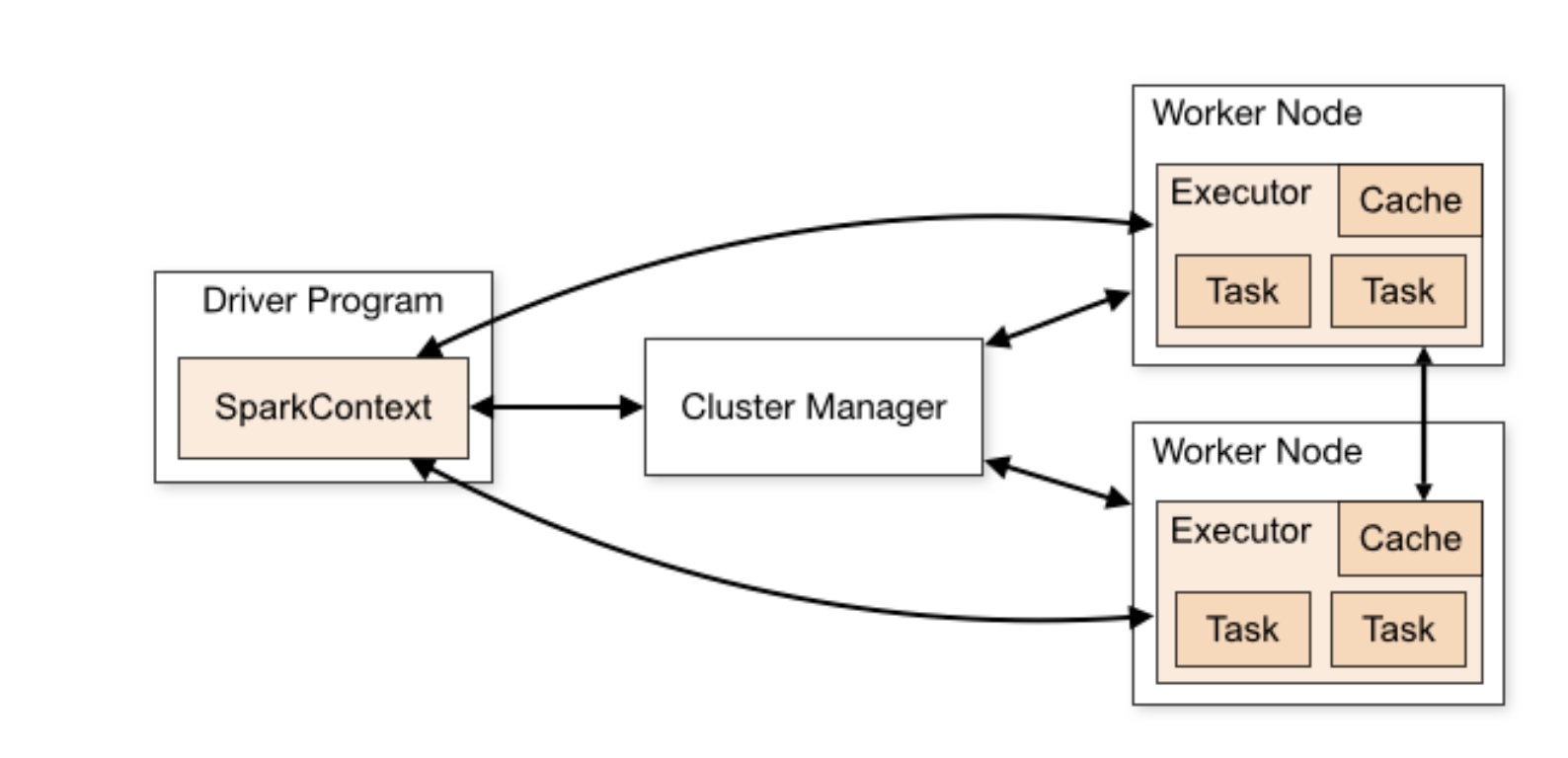
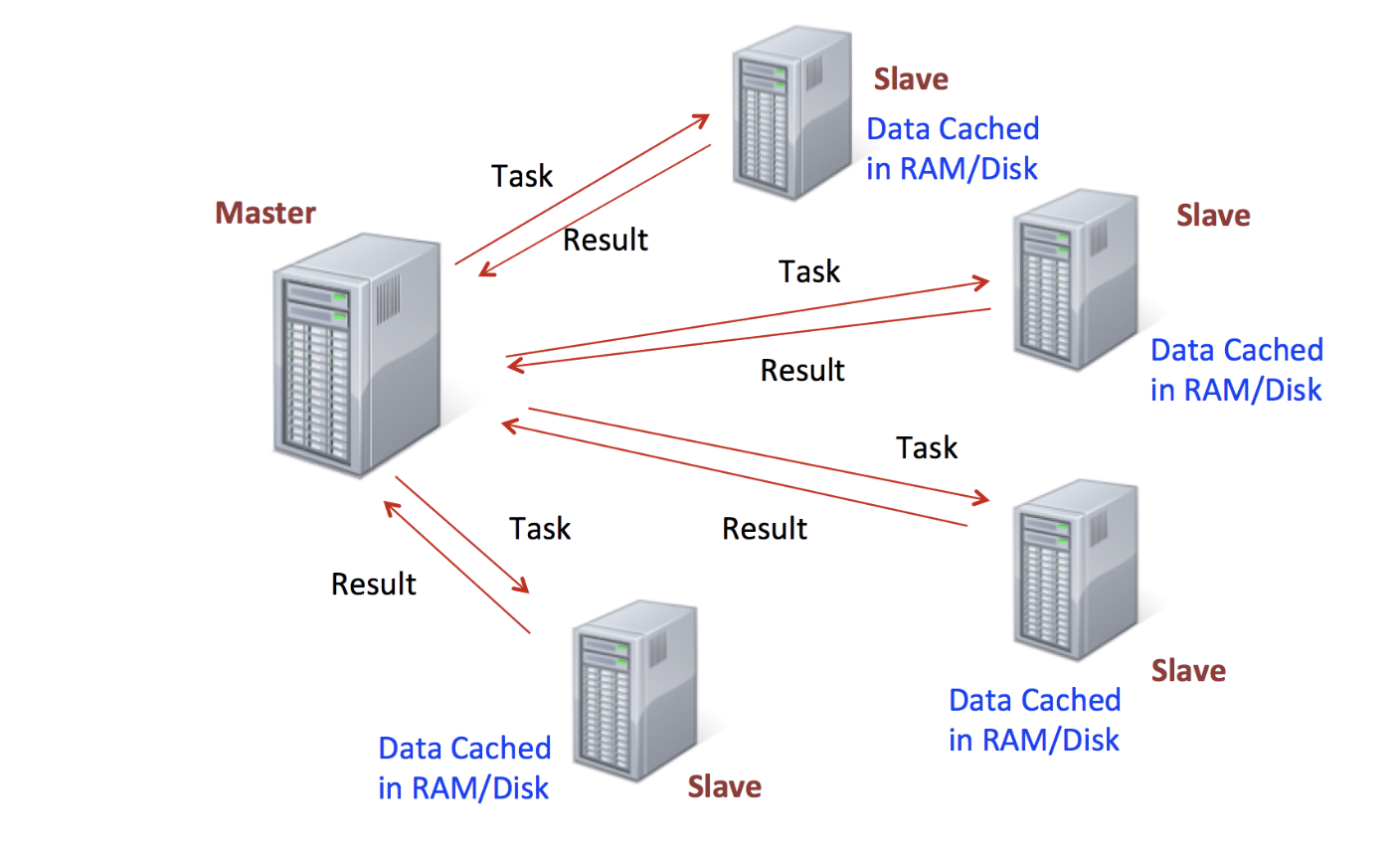
**Slide Link :**

<https://docs.google.com/presentation/d/1u5FT9oG2lkSP6BKS9xYcHAd-MP17L3KSQFsX44nAzAY/edit#slide=id.g1db6bb72b7_0_382>

1. Used for very large data. Which numpy and pandas cannot handle
2. Spark works with a distributed system which is very advantageous than having a single powerful machine.
3. Small data upto 32-64 gb can be handled on a RAM i.e. when an excel file is opened, RAM is used.
4. Further, SQL can be used to store data in hard drive and access in ram
5. Further, distributed systems can be used to access large data which cannot be handled on a single system. Also has an advantage that if one machine fails, others remain
6. Distributed system uses fault tolerance i.e. stores files on multiple machines i.e. if one fails, it has other
7. HDFS or Hadoop distributed file systems are used to store large data using multiple systems and Map Reduce is used to do computation on that distributed large data set.
8. Hadoop also stores files on multiple machines using HDFS Hadoop distributed file system and uses map reduce to work with them. HDFS uses blocks of size 128mb, replicated 3 times. Has fault tolerance.
9. Map reduce consists of job trackers and Task trackers.

* Spark is an open source project on Apache. It is an alternative to Map reduce i.e. way to distribute tasks on this data.
* Spark can use data, stored in different formats like Cassandra, AWS S3, HDFS
* Spark is 100 times faster than map reduce. MapReduce keeps data in disk after each map and reduces operation. Whereas, Spark keeps it in memory i.e. like RAM and not on disk.
* Spark can also spill over disk if memory is full. But, keeping things on RAM makes it faster.
* Spark is based on the RDDResilient Distributed Dataset. RDD has 4 main features: Distributed collection of Data, Fault - Tolerant, Parallel operation, Ability to use many data sources. Its architecture looks similar to HDFS except that memory is cached.

* RDDs are immutable, laily evaluated and cacheable
* There are 2 types of spark operations: Transformations and Actions
* Transformations are a recipe to follow. Actions are actually performing the recipe and returning the results.
* In syntax we need to write the transformation and call the action to perform that. It is like confirmation that we really want to perform that transformation.
* Spark is for clusters of machines and usually linux systems are used for clusters.