What Does MapReducer Program Consists of?

The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.

Minimally, applications specify the input/output locations and supply *map* and *reduce* functions via implementations of appropriate interfaces and/or abstract-classes. These, and other job parameters, comprise the job configuration. The Hadoop job client then submits the job (jar/executable etc.) and configuration to the JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

What is the relation Between the input and output od mapper, partitioner, and reducer?

The output of the mapper is given as the input for Reducer which processes and produces a new set of output, which will be stored in the HDFS. Reducer first processes the intermediate values for particular key generated by the map function and then generates the output (zero or more key-value pair).

The Hadoop reducer process the output of the mapper. After processing that data. It produces a new set of output. At last, HDFS Stores this output data,

The partitioner in MapReduce controls the partitioning of the key of the intermediate mapper output. A hash function key is used to derive the partition. According to the KEY-VALUE each mapper output is partitioned and records having the same key value go into the same partition, and then each partition is sent to the reducer. Partition phase takes place after map phase and before reduce phase.