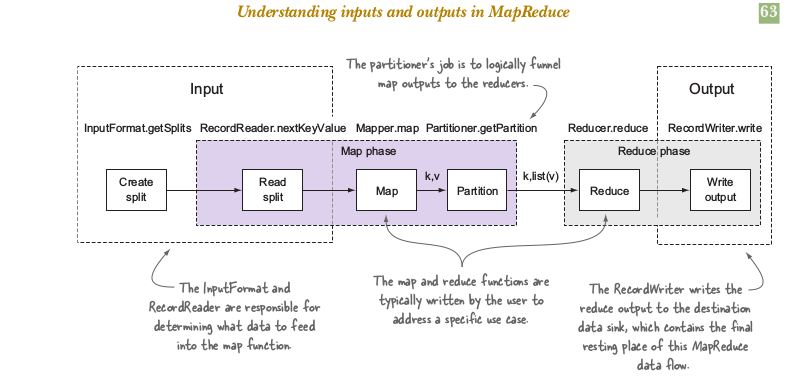
**Understanding input and output of MapReduce.**

****

The above picture is the snapshot of input and output flow of MapReduce. The two classes that support data input in MapReduce are InputFormat and RecordReader . The InputFormat class is consulted to determine how the input data should be partitioned for the map tasks, and the RecordReader performs the reading of data from the inputs.

InputFormat is an interface. The class extends the InputFormat must fullfill three contracts:

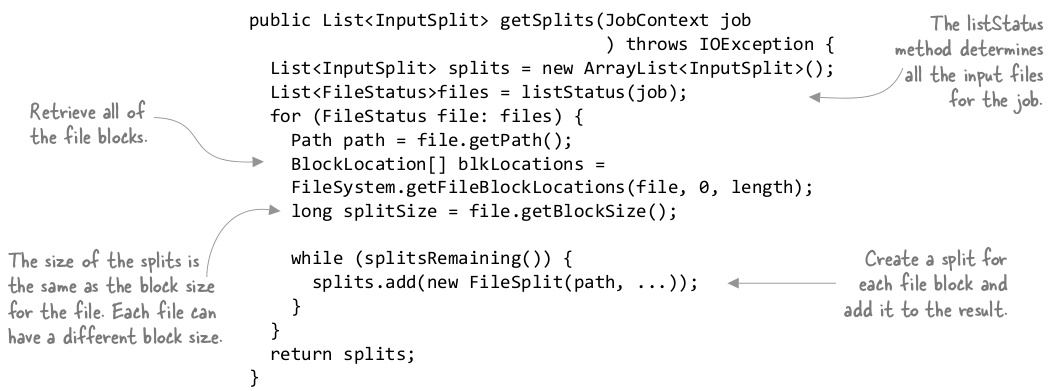
1. They describe type information for map input keys and values.

2. They specify how the data input should be partitioned

3. They indicate the RecordReader instance that should read the data from the source.

There is class provided by Hadoop-Core and Hadoop-mapreduce-client-core that implements InputFormat ie FileInputFormat (its an abstract class). It provide one abstract method getRecordReader that return the custom Recordreader class.

Arguably, the most crucial contract is that of determining how to divide the input data. In MapReduce nomenclature, these divisions are referred to as input splits. The input splits directly impact the map parallelism, because each split is processed by a single map task. Working with an InputFormat that’s unable to create multiple input splits over a single data source (such as a file) will result in a slow map phase, because the file will be processed sequentially.

The split is done on the block basis of each file in each node. In general, if you have n nodes, the HDFS will distribute the file over all these n nodes. If you start a job, there will be n mappers by default. Thanks to Hadoop, the mapper on a machine will process the part of the data that is stored on this node. I think this is called Rack awareness.

This split-ed blocked data is input for the RecordReader in “initialize” method, that method internally open the FSDataInputStream to read the data block or use the Reader class to read the block line by line.

**RecordReader**

You’ll create and use the RecordReader class in the map tasks to read data from an input split and to provide each record in the form of a key/value pair for use by mappers. A task is commonly created for each input split, and each task has a single RecordReader that’s responsible for reading the data for that input split.

