**The distributed filesystem**

A name node takes care of reading from and writing to a set of data nodes running on different computers. The application programmer needn’t concern himself with this.

**The mapping and reducing**

The mapper takes in each fixed-length key-value record from the input file, and outputs any number of similarly key-value records. The application programmer only needs to implement the *map* function to send a bunch of records to the output collector.

The outputted records are sorted by key.

The reducer takes in a key and all the values associated with it, and outputs some sort of result obtained from the values of that key. The application programmer only needs to implement the *reduce* function to send the output of the reducer given a key and the list of values for that key, to the collector.

**Running custom mappers and reducers**

After setting up the system as described in *System Maintain Documents.docx*, simply put (with your corresponding mapper and reducer classes of course)

MAPPER=ha.testing.wordcount.Mapper

REDUCER=ha.testing.wordcount.Reducer

at the end of your configuration file before starting JobClient and sending over the job configuration to the master node.