Devon Timaeus
CSSE490 – Hadoop, Mohan

Lab 3

1.

TextOutputFormat takes the key and value pair for a record, and writes it as a string that delimits the two by a tab character. So, if the key were "hello" and the value was "world", then the output from the TextOutputFormat would be "hello\tworld".

2.

MultipleOutputFormat allows for writing to files that are determined by the Format, and not just auto-generated by the reducer. The format can be used to make different files for the different keys (or values) of the reducer's output. Furthermore, the OutputFormat can also be used to create custom output files that are named based off of both the output keys, and the files that the keys came from. As an example, if there is an input file of pricing data over multiple years, and we wanted to put all the output for prices for a given year into a file together, we could use a MultipleOutputFormat to create output files based on the year, and then put all the same year's data into the same file and split the output as opposed to all the output getting mixed together into a few output files based on reducer.

3.

In order to save network bandwidth, and speed up (slightly) the fetching of values from the map to the reduce, MapReduce jobs can be written to make use of the distributed cache. Basically, the idea is that whenever values outputted by the map are going to be used quite a bit, they can be copied to the local filesystem and used directly. In doing this, the network bandwidth is reduced, and the speed of the reading is increased. Suppose we wanted to have a loopkup table for the output of the map, then, anywhere that we are running a reduce job, we could have access to the output of a map. This also buys us the ability to make all the output of the maps available at each reducer, that being said, there are limitations on the size of the distributed cache file.