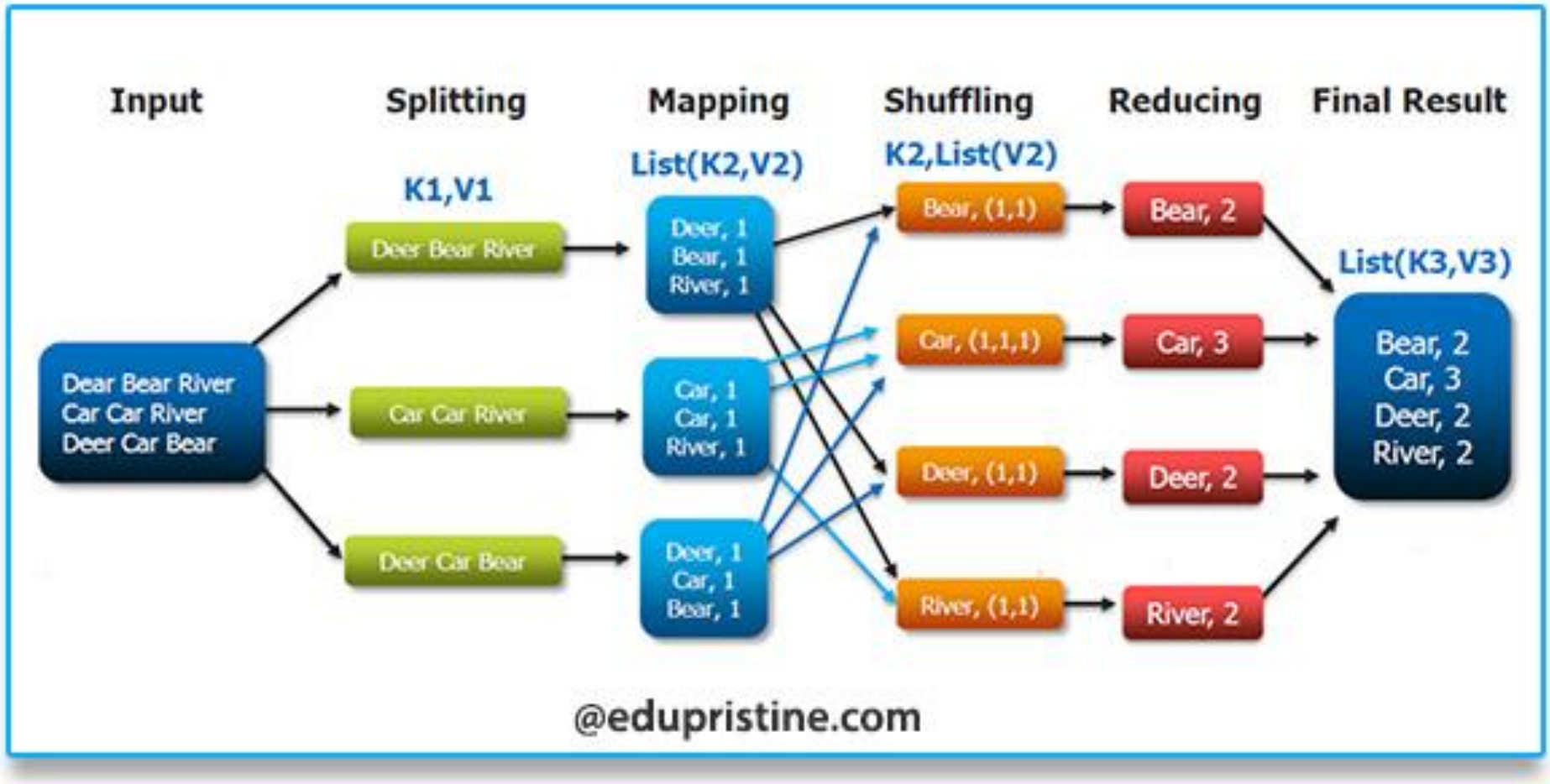
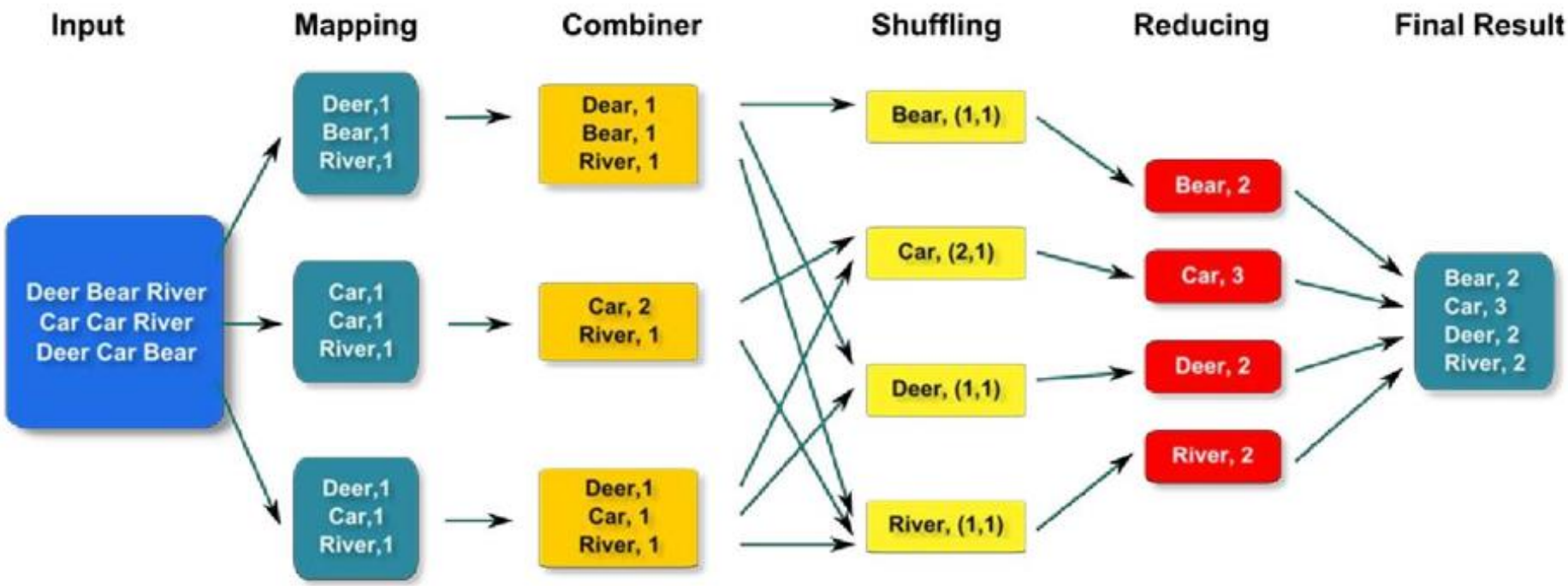
Big Data Analytics – Module 3 (MapReduce)

MapReduce Workflow – W/o Combiner



MapReduce Workflow – W/o Combiner



Combiner Functions

- Many MapReduce jobs are limited by the bandwidth available on the cluster, so it pays to minimize the data transferred between map and reduce tasks.
- Hadoop allows the user to specify a combiner function to be run on the map output, and the combiner function's output forms the input to the reduce function.
- A Combiner, also known as a semi-reducer, is an optional class that operates by accepting the inputs from the Map class and thereafter passing the output key-value pairs to the Reducer class. The main function of a Combiner is to summarize the map output records with the same key.
- Because the combiner function is an optimization, Hadoop does not provide a guarantee of how many times it will call it for a particular map output record, if at all. In other words, calling the combiner function zero, one, or many times should produce the same output from the reducer.

Combiner Functions

- The contract for the combiner function constrains the type of function that may be used.
- This is best illustrated with an example. Suppose that for the maximum temperature example, readings for the year 1950 were processed by two maps (because they were in different splits). Imagine the first map produced the output:

(1950,0)

(1950,20)

(1950,10)

And the second produced:

(1950,25)

(1950,15)

The reduce function would be called with a list of all values:

(1950,[0,20,10,25,15])

With output:

(1950,25)

Combiner Functions

- We could use a combiner function that, just like the reduce function, finds the maximum temperature for each map output.
- The reduce would then be called with:

(1950,[20,25])

and the reduce would produce the same output as before

- We may express the function calls on the temperature values in this case as follows:

max(0, 20, 10, 25, 15) = max(max(0, 20, 10), max(25, 15)) = max(20, 25) = 25

Specifying a combiner function

- The combiner function is defined using the Reducer class, and for this application, it is the same implementation as the reducer function in **MaxTemperatureReducer**
- The only change we need to make is to set the combiner class on the **Job**

Application to find the maximum temperature, using a combiner function for efficiency

```
public class MaxTemperatureWithCombiner {


  public static void main(String[] args) throws
    Exception { if (args.length != 2) {

      System.err.println("Usage: MaxTemperatureWithCombiner <input path> " +
          "<output path>");

      System.exit(-1);

    }


    Job job = new Job();
    job.setJarByClass(MaxTemperatureWithCombiner.class);
    job.setJobName("Max temperature");


    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
```

Application to find the maximum temperature, using a combiner function for efficiency

```
    job.setMapperClass(MaxTemperatureMapper.class);
    job.setCombinerClass(MaxTemperatureReducer.class);
    job.setReducerClass(MaxTemperatureReducer.class);


    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);


    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```