## 2CS702 - Big Data Analytics

## Practical 6

**Aim**: **- To analyse the impact of different numbers of mappers and reducers.**

**Author: Darshil Maru 20BCE514**

**Guide: Dr. Purnima Gandhi**

**Introduction of Mappers and Reducers:-**

**Mappers**

**Hadoop Mapper** is a function or task used to process all input records from a file and generate the output that works as input for Reducer.

No. of Mapper= {(total data size)/ (input split size)}

1) By default, in Hadoop, the input size split is 128 MB. However, we can change the same
On reducing the size, the number of mappers will increase
**set mapreduce.input.fileinputformat.split.maxsize=100000;**
For example, for block size 100 MB and input of 1gb, 10 mappers will be used

2) We can also set the number of mappers as in the driver code
**Job.setNumMapTask(5) which sets 5 number of mappers**

3) While executing the job
**hadoop jar /hirw-starterkit/mapreduce/stocks/MaxClosePrice-1.0.jar com.hirw.maxcloseprice.MaxClosePrice -D mapred.reduce.tasks=10 /user/hirw/input/stocks output/mapreduce/stocks**

The right level of parallelism is 10-100 mappers per node; if the mappers are relatively small, then, maybe 300 mappers per node.
If the number of mappers is too much or too less, it can slow the system down.

**Reducers**

Reducer in Hadoop MapReduce reduces a set of intermediate values which share a key to a smaller set of values. In MapReduce job execution flow, Reducer takes a set of an intermediate key-value pair produced by the mapper as the input.
As given on the website of Hadoop the ideal number of Reducers are (.95 or 1.75) * (nodes * number of mappers)
1) We can also set the number of reducers as

**Job.setNumReduceTask(5) which sets 5 number of reducers**

2) Using command line

```
jar word_count.jar com.home.wc.WordCount /input /output \ -D
mapred.reduce.tasks
= 20
```

which sets the number of the reducers to 20

Suppose there are 100 reduce slots available in the cluster.

When we consider .95 as the factor, there will be 95 reducers and it means that no task will be waiting. This is to be done when all the tasks take equal number of times.

When we take 1.75 as the factor, 100 tasks will start simultaneously, while 75 will be in the queue. This will allow better load balancing as it would prevent bottleneck of the jobs.

## Conclusion

In this practical we learned about the various ways we can change the number of mappers and reducers and how it impacts the job in general. It is essential to select the right number of mappers and reducers while running jobs that have Tbs of data.