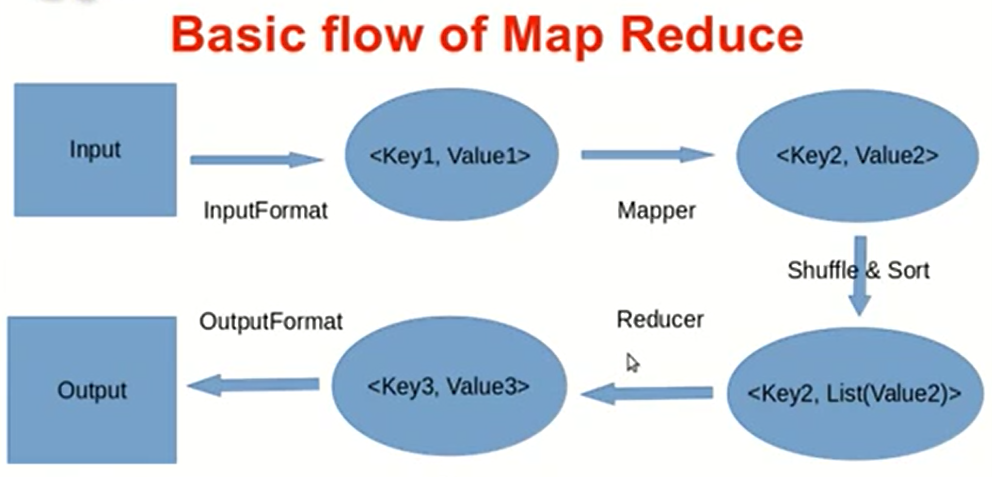
Mapper :

will extract data from input file with help of input format (TextInputFormat -default)

Reducer :

will aggregate the data from mapper and send to output Format (TextOutputFormat-Default)



Input format: internally call record reader to read the input file and split the n- no of records

MAP, Reduce 🡪 Functional Programming it will be enable distributed parallel processing

From HDFS (blocks) to read the data and send it for process (MAP)

MApper 🡪 will run parallelly depends on input split ( first phase of processing) process the data

This where user can write custom business logic ( all are complex logic specified)

i/p (Key , value) to the Mapper

Key will be byte offset, key will be deposition ( first key starts as 0 as key and Value will be complete line

,2nd row 🡪Key for second line will no. Of character in the first line)

MAP function will get one key and one value as input at a time once map function will process will get next key and value provided by frame work

**Input to reducer (All the values corresponding to same key goes to same reducer)**

Suppose 1000 of Mappers the key Datafiar(word) will be available in 3 - node

1 mapper 🡪 DataFlair is word as key and count is 100 as value

2nd mapper 🡪 DataFlair is word as key and count is 50 as value

3rd mapper 🡪 DataFlair is word as key and count is 500 as value

**Here each key is DataFlair so it will go to same reducer this reason we keep the key as word**

In reducer Key is single value and value is collection (counts of words)

O/p from reducer is final value again in the Key, value format

Output of mapper is called intermediate o/p stored on local disk (? Since its intermediate o/p so no need to store in HDFS)

Until unless all the mapper finish ,reducer phase won’t start ( all the mapper completed then only reducer phase will start)

o/p of mapper node ,Shuffling of data from mapper node to reducer node ( this is physical movement happen)

is shuffling will start all the mapper are finish or as soon as just one mapper finish ?

**as soon as first mapper finish shuffling will start i.e data will transfer to mapper node to reducer node**

**Framework will merge all the intermediate o/p which will get from all the mapper and give to reducer**

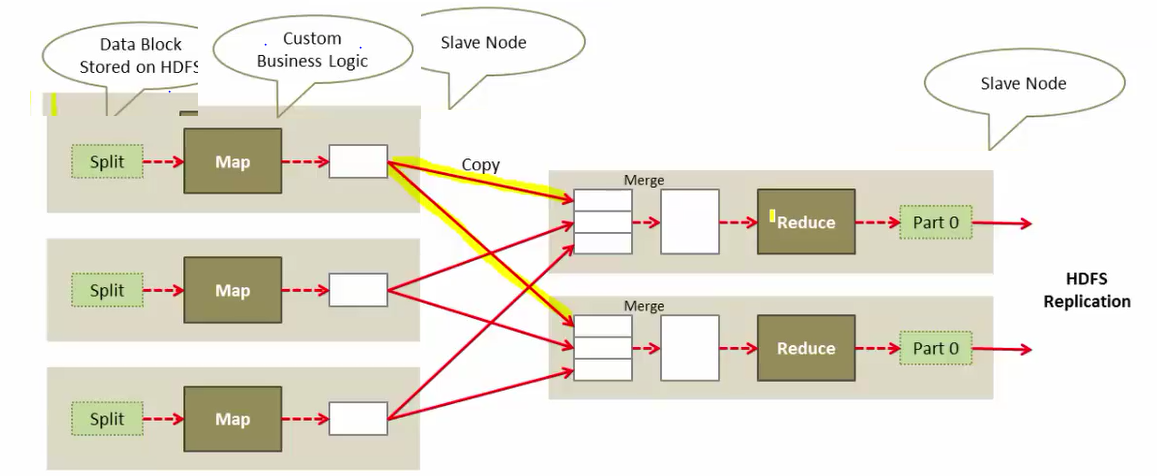
**In Reducer Phase where user can mention custom business logic usually aggregation and summation in the reducer phase**

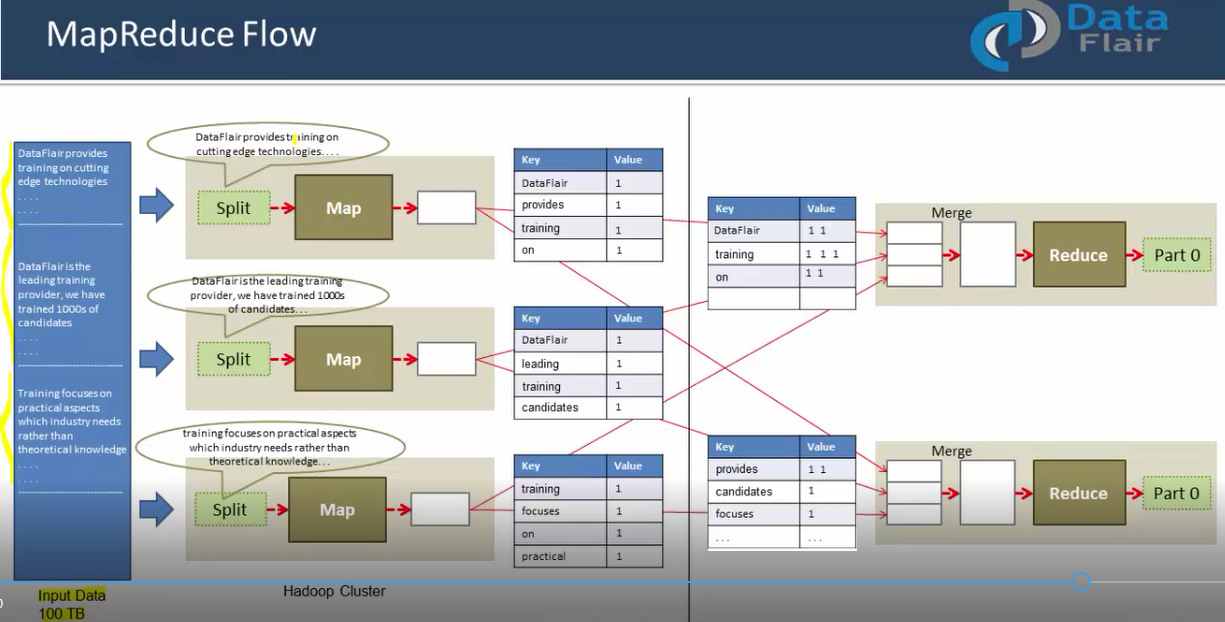
**o/p of reducer is final o/p will stored on the HDFS .**

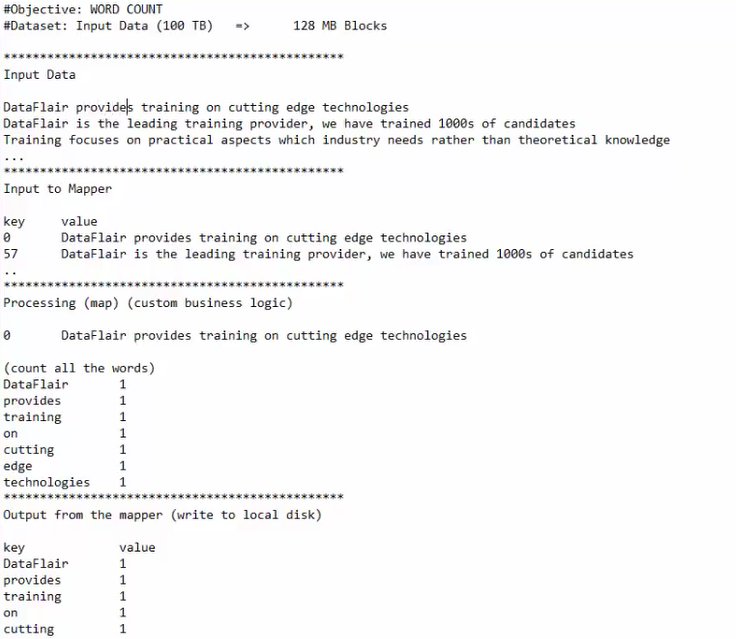
**as soon as data will stored in HDFS , data will be replicated**

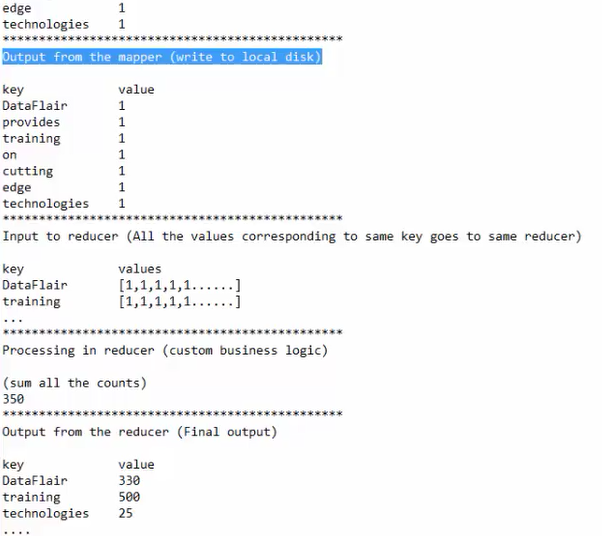
**all white boxes data on the local disk**

**o/p of Each mapper will send to each reducer that is responsible for framework**

****

****

****

****

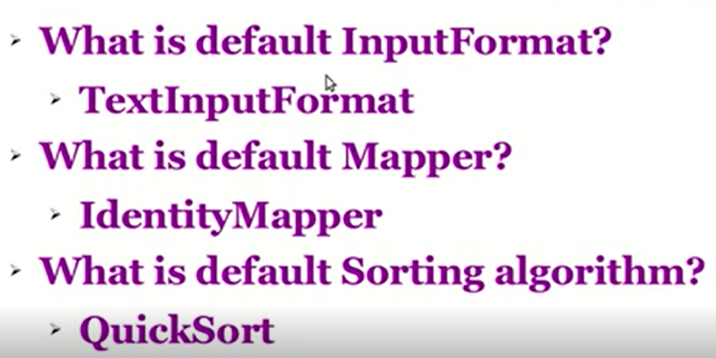
I have 1 GB data and block size is 64 MB so it will split the 16 blocks by default

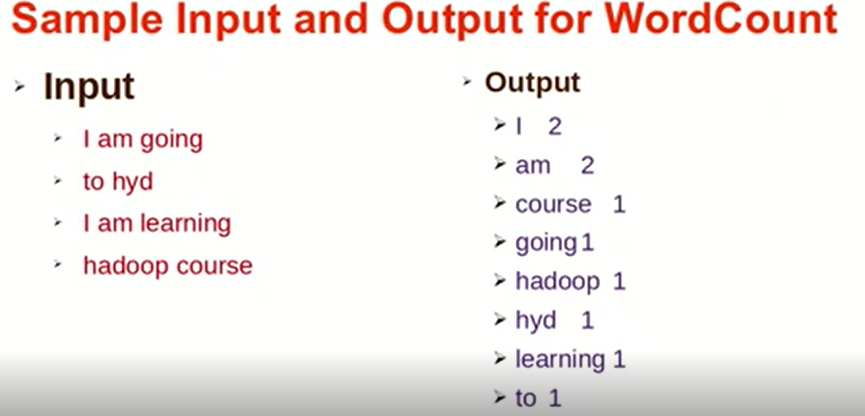
**No.of blocks = No.of Split** , so16 split🡪 16 Mapper are executing parallel

Record reader 🡪Record 🡪 will pass to Mapper 🡪 Shuffle & sort 🡪 Duplicate key exists it will come us list of values 🡪 Reducer 🡪 it will reduce the record 🡪 record writer 🡪 write data into output file

**1 Mapper can process one split at any point of time**







Steps:

1.Mapper 🡺 map()

2. 1. Read the file line by line

2.2 Split the line into words

2.3 assign Count(1) to each word

3 Reducer 🡪reduce()

3.1 sum the list of values

3.2 assign the sum to corresponding word

For implementing above steps we require the below java class

Wordcountjob

WordcountMapper

wordCountReducer

