Map-Reduce Working in Hadoop

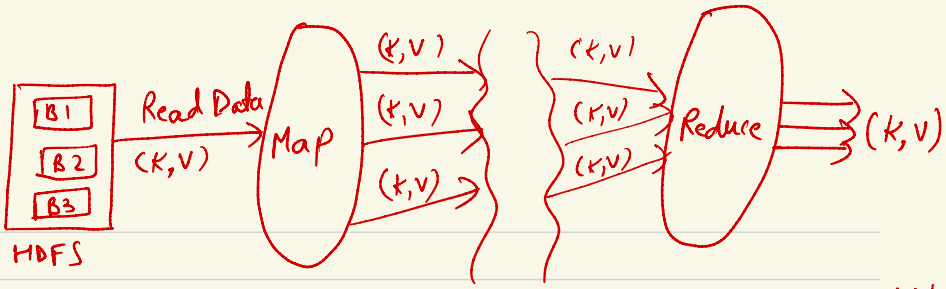
① HDFS

② YARN

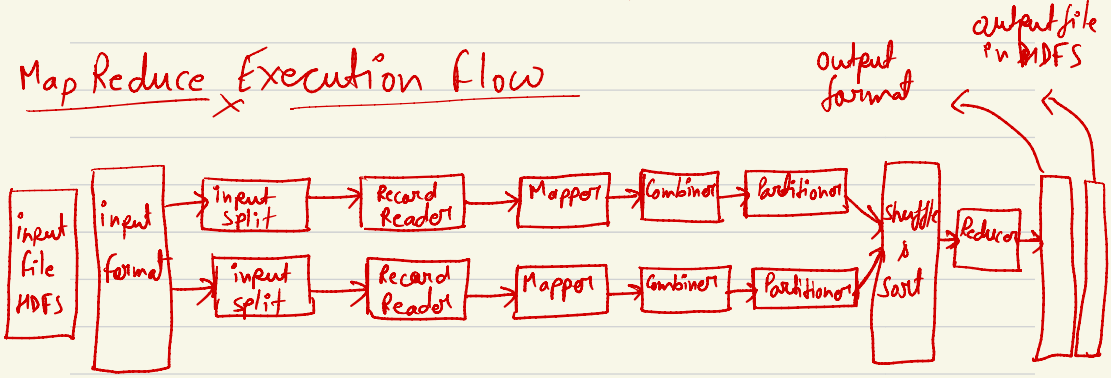
③ MAP-REDUCE

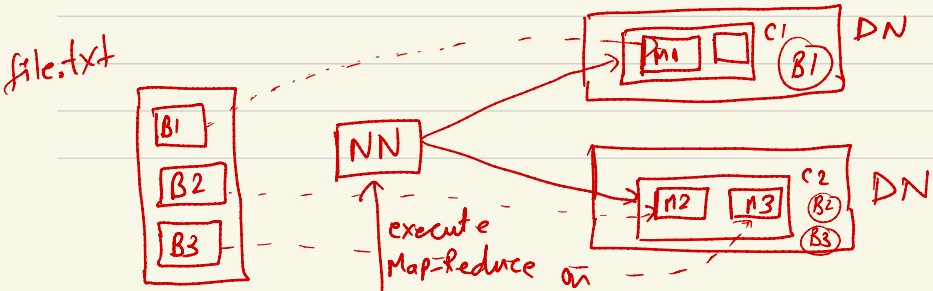
*) MAP → Reads Blocks/data and generate (Key, Value) output.

*) REDUCE → Reads (Key, Value) data generated by map phase and aggregates the result. Final result will be also in (Key, Value)



Map Reduce Execution Flow





HDFS

client

file.txt

word_count.py

def mapper ({K, V}): → coming from HDFS input file from HDFS

    ———

    ———

    ———

    return {K1, V1}

def Reducer ({K1, V1}):

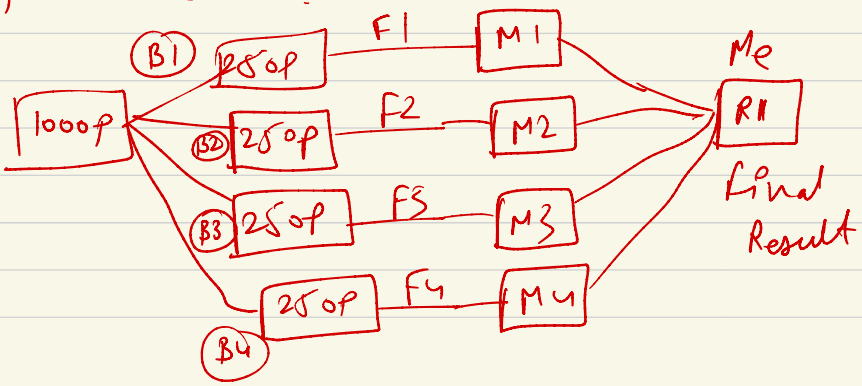
    ———

    ———

    print ({K2, V2}) → output file to HDFS

Book of 1000 pages.

q) Time for a human to calculate frequency of each word?



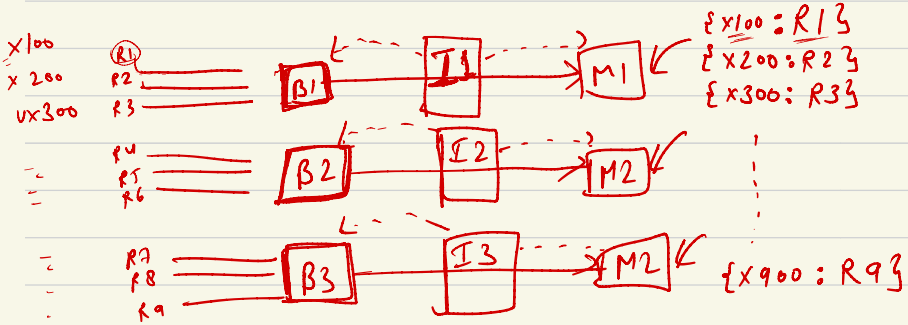
① Input Files: Simply the files or references for the data stored in HDFS

② Input Format: defines how these input files are read from source or how to serialize & deserialize them properly.

③ Input Splits: Logical representation of the data which will be processed by an individual mapper task. One map task is created for each input split.

*) Block: Actual unit where raw data will be stored physically.

*) Input Split: It is like a reference to the original Block of Data.

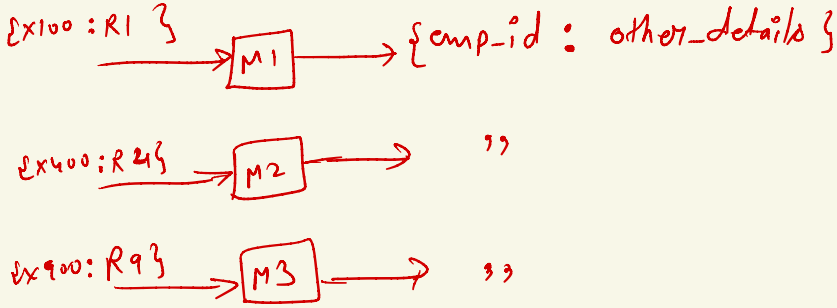


* Record Reader

↳ It communicates with the Input Splits in Hadoop Map Reduce and converts the data into key-value pairs.

* Mapper :

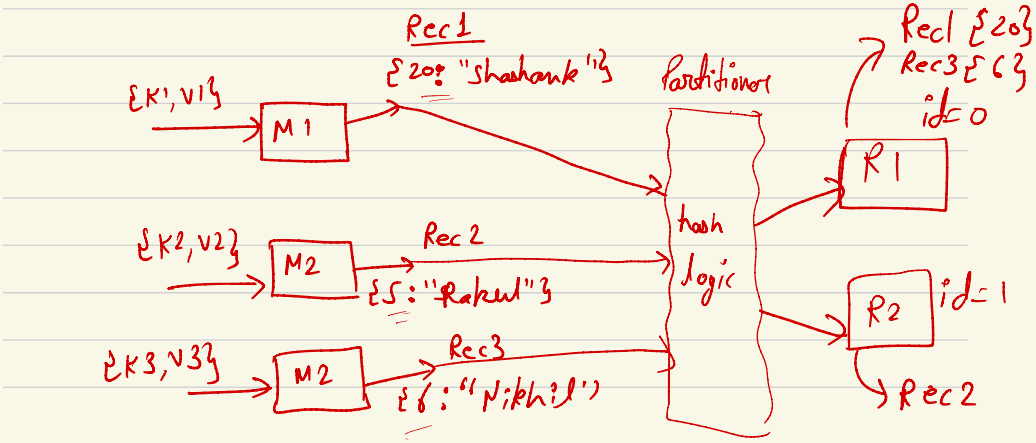
It processes each input record from record Reader and generates new key-value pair which is different from input key-value pair.



* Combiner

It is like a mini-reducer. It will aggregate results locally generated by each mapper task.

* Partitioner: It will decide in which reducer records of mapper task will go to.



Partitioner logic or hash function.

hash(Key) = Key % total_neum_of_reducers

Rec1
{20: "shashank"} ⇒ Key = 20
= hash(20)
= 20 % 2
= 0

Rec2
{5: "Rahul"} ⇒ Key = 5
= has(5)
= 5 % 2
= 1

x) **Shuffling & Sorting:**
It will arrang the data for each key on individual reducer.

x) **Reducer:** It will receive mapper task's data as input and will generate resultant output.

x) **Output Format:** In which format we want to write data in HDFS

*) Output Files: Generated reference for output data.

Word Count Example in Map-Reduce.

file.txt

```
cat bat mat cat rat
owl bat cat owl
rat rat rat cat
mat mat cat cat
```