

### Lab 3: Map Reduce

- **AIM:**

To perform Map Reduce Algorithm by writing mapper & reducer algorithm using python and execute on Linux machine.

- **Application used:**

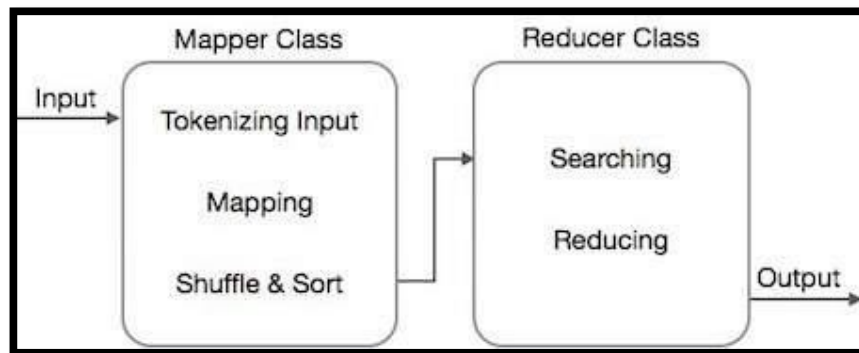
<https://cocalc.com/> >> LINUX TERMINAL

- **Theory:**

The MapReduce algorithm contains two important tasks, namely Map and Reduce.

- The map task is done by means of Mapper Class
- The reduce task is done by means of Reducer Class.

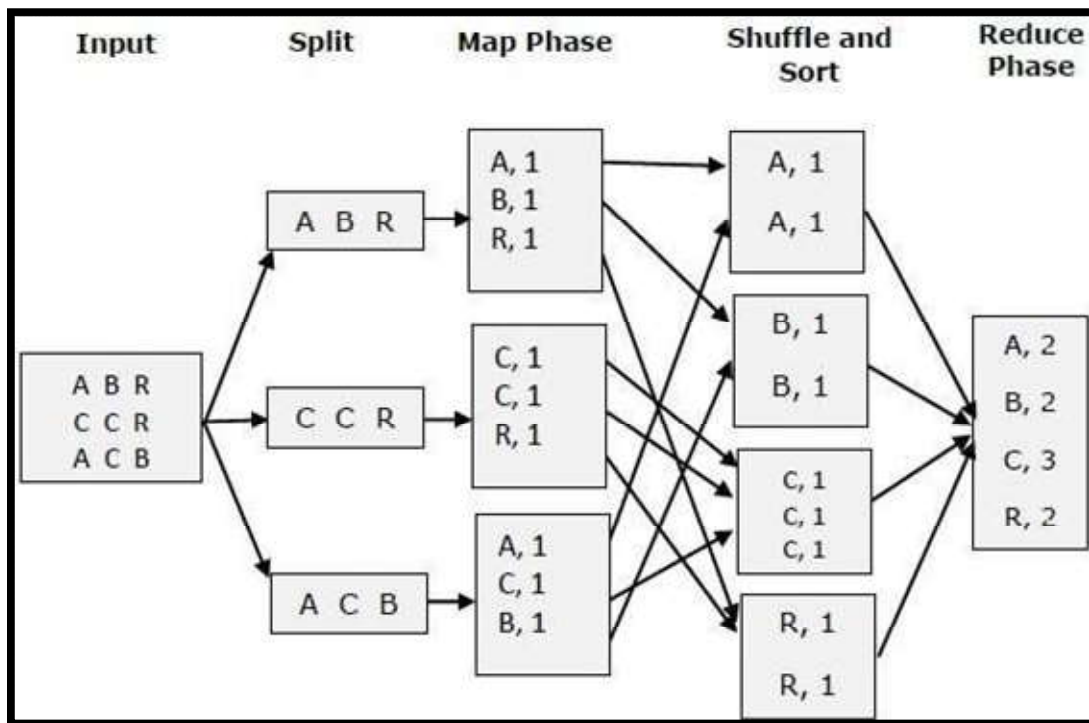
Mapper class takes the input, tokenizes it, maps and sorts it. The output of Mapper class is used as input by Reducer class, which in turn searches matching pairs and reduces them.



MapReduce implements various mathematical algorithms to divide a task into small parts and assign them to multiple systems. In technical terms, MapReduce algorithm helps in sending the Map & Reduce tasks to appropriate servers in a cluster.

These mathematical algorithms may include the following –

- Sorting
- Searching
- Indexing
- TF-IDF



- **Algorithm Steps: -**

1. using 'cat' command we will create a data.txt to store our raw data (i.e. list of words).

```
~$ cat >> labdata.txt
Apples
Oranges
Apples
Oranges
Peaches
Kiwi
Apples
Apples
Peaches
Oranges
```

2. We will then create a python file which will store our mapper class.

```
~$ cat >> mapper.py
import sys
for line in sys.stdin:
    line=line.strip()
    words=line.split()
    for word in words:
        print('%s \t %s'%(word,1))
```

3. We will then create a python file which will store Class Reducer.

```
~$ cat >> reducer.py
import sys
prev_word=None
prev_count=0

for line in sys.stdin:
    line=line.strip()
    word,count=line.split('\t')

    count=int(count)
    if prev_word==word:
        prev_count +=count
    else:
        if prev_word:
            print('%s\t%s'%(prev_word,prev_count))
            prev_count=count
            prev_word=word
        if prev_word==word:
            print('%s\t%s'%(prev_word,prev_count))
```

4. Next we will use the 'ls' command to check all the files we have created are saved in the system .

```
~$ ls
2023-02-26-file-BDALAB.term  labdata.txt  mapper.py  reducer.py
```

5. Now we will perform the mapper function by calling the Data.txt and mapper.py files in the terminal.

```
~$ cat labdata.txt | python mapper.py
Apples 1
Oranges 1
Apples 1
Oranges 1
Peaches 1
Kiwi 1
Apples 1
Apples 1
Peaches 1
Oranges 1
```

6. After our raw data has been mapped, we will call the reducer.py along with the previous two files.

```
~$ cat labdata.txt | python mapper.py | sort | python reducer.py
Apples 4
Kiwi 1
Oranges 3
Peaches 2
```

**Conclusion :-** We successfully executed Map Reduce Algorithm using on Linux Terminal.