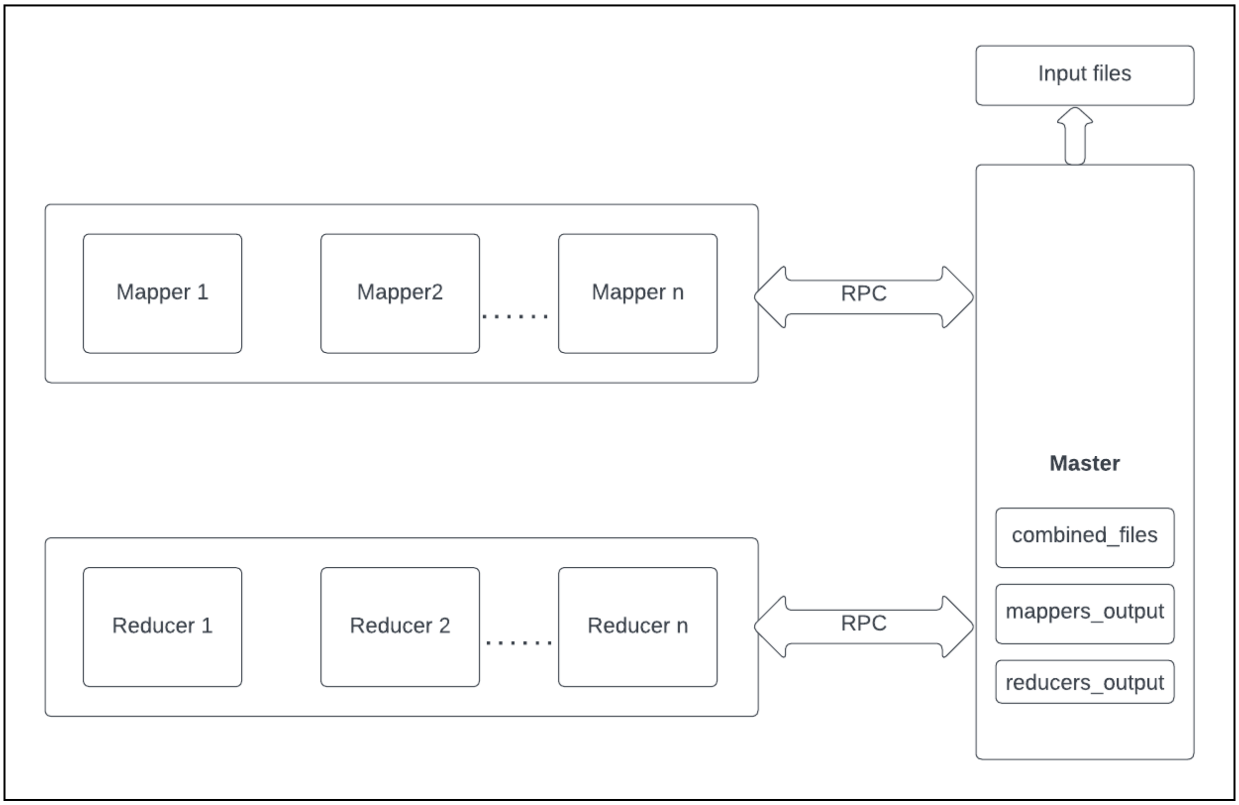
MapReduce

Rushikesh Pharate

Overview

The overall system comprises 3 types of processes, namely, Master, Mapper, and Reducer. There is also another process (database_server) that is used for performing the database operations by mapper and reducer. We can consider this database process as part of the master. The Master type has a single instance and coordinates the entire system. Mappers, Reducers have multiple instances and entire system communicates via RPC calls.. All types of processes are multi-threaded and the entire system is implemented in Python.



MapReduce Architecture

I. Master

In summary, the master process reads and splits the input, spawns mappers and reducers, monitors mappers and reducers, and sets up barriers.

Operations performed by the Master:

1. Read and Split Input:
    - Master reads all the text files stored in the ./inputFiles directory and divides the input equally in all the mappers
2. Spawns Mappers:
    - Spins up mapper processes (can be considered as different hosts) and send them their configuration like which byte offsets to read form the input, mapper_id, no_of_reducers, master_address, master_port etc.
3. Stand a Barrier:
    - After all the processes are up, the master brings up a barrier
    - Invokes status check thread corrsponsing to each mapper and periodically check the progress of each mapper
    -
4. Mapper Monitoring:
    - If a process fails or is not responding then the status checker thread changes that particular processes status to FAILED which will inturn result in respawning of that particular mapper
    - If the mapper is finished processing update its status to DONE which will in turn result in stopping the mapper process
5. Spawn Reducers:
    - Spins up reducer process with initial configuration like the one mentioned for mapper
6. Stand a Barrier again:
    - This barrier performs the same tasks as described in point 3 but for reducers
7. Reducer Monitoring:
    - This barrier performs the same tasks as described in point 4 but for reducers
8. Database Server:
    - Database server is master process responsible for answering mapper and reducer calls for data.

II. Mappers:

Mapper is a multithreaded process started by the Master (can be on a different host).
Mappers perform the below tasks

1) Once the process is up, the first thing it does is start an RPC server using the configurations received from the master.
2) After that, it send an request to master asking particular offsets of data.
3) Once the mapper receive sthe data it requested it performs its operations on it based on the type of mapper that is invoked [mapper_word_count.py or mapper_inverted_index.py]
4) While processing the data it maintains its progress in a mapper_progress variable and send this data to master whenever it received period request for progress.
5) After processing the data, it sends the its output to the master.
6) Once its task is finished, it waits till the Master sends an termination message and terminates itself.

III. Reducers

Reducer is a multithreaded process started by the Master (can be on a different host).
Reducers perform the below tasks

7) Once the process is up, the first thing it does is start an RPC server using the configurations received from the master.
8) After that, it send an request to master with its reducer_id and master sends back data to process.
9) Once the reducer receives the data it requested it performs its operations on it based on the type of reducer that is invoked [reducer_word_count.py or reducer_inverted_index.py]
10) While processing the data it maintains its progress in a reducer_progress variable and send this data to master whenever it received period request for progress.
11) After processing the data, it sends the its output to the master.
12) Once its task is finished, it waits till the Master sends an termination message and terminates itself

Fault Tolerance:

1. The status checker thread invoked by the master for each process maintains the state information of mappers and reducers.
2. There are 3 instances when the master will restart the process.
    - When the master is not able to communicate with the mapper/reducer - getting connection refused error
    - When the mapper/reducer is responding with the same progress status for multiple periodic checks
    - When the mapper/reducer is running for more than 3 minutes

Steps to Run MapReduce

- **Prerequisites**
    - You have the submission folder
    - Your base directory is MapReduce folder in the submission
    - The input text files need to be stored in the *./inputFiles* directory. There can be multiple input files.
    - The base directory should have an *userInput.json* file with the initial configuration for MapReduce
    - The base directory should have the below subdirectories
        - *./inputFiles*
        - *./mapperInput*
        - *./mappersOutput*
        - *./output*

- Open *userInput.json* file and change the *NO_OF_MAPERS / NO_OF_REDUCERS* fields as per your convenience.
- If you want to run Word Count on the given input then change "*MAPPER_TO_USE*" field to "*mapper_word_count.py*" and "*REDUCER_TO_USE*" field to "*reducer_word_count.py*"
- If you want to run Inverted Index on the given input then change "*MAPPER_TO_USE*" field to "*mapper_inverted_index.py*" and "*REDUCER_TO_USE*" field to "*reducer_inverted_index.py*"
- Make sure the ports that are mentioned in the userInput.json file are not used other processes
- *NO_OF_MAPERS* / *NO_OF_REDUCERS* should never exceed the *MAPPER_DETAILS* / *REDUCER_DETAILS* length
- Run *python3 master.py* command in the base directory
- Final output of the MapReduce job will be stored under *./output* folder in the *output.txt* file
- *log.txt* will be generated in the base directory and all the logs will be stored there

Limitations and Future Improvement

1) While failing mappers/reducers is handled in this program, if master fails then the program will crash. This can be prevented if we have a backup master.
2) The current implementation of status checker handles only a few failure states of the reducer and mapper, it can be modified to add more states
3) When the same process fails multiple times then there is high possibility that something is wrong and the job can be terminated. This is not handled in the current MapReduce implementation

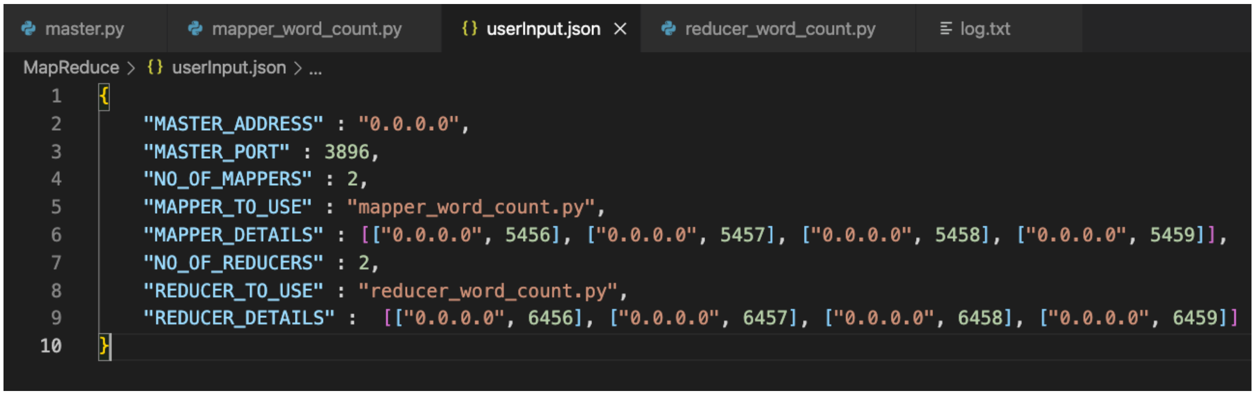
4) Right now, the directories required for MapReduce to run are hard coded. This can be made dynamic.

Outputs

1) Word Count

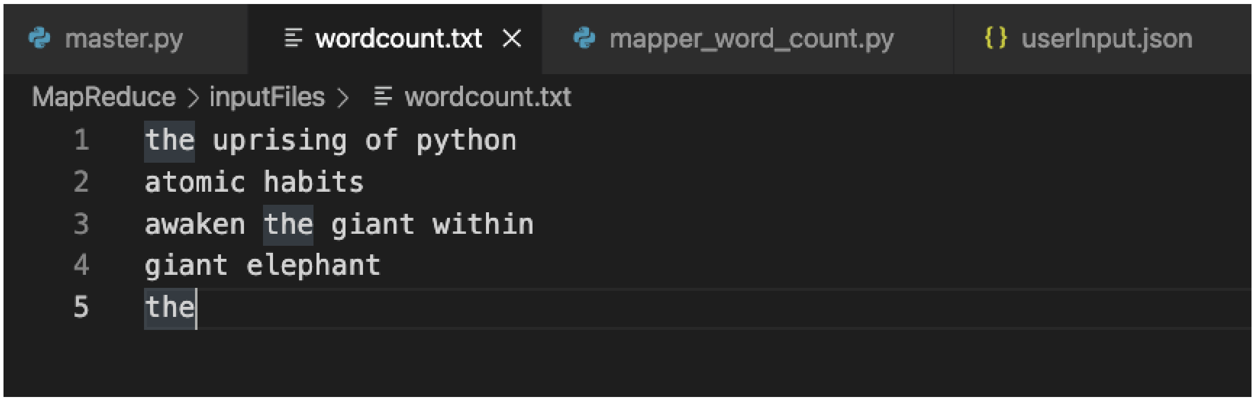
- Word Count counts the number of occurrences of all the words in one or many documents
- Each mapper generates a *(word, 1)* pair for each word and performs hashing / group by on it to get a *reducer_id.*
- Hashing/Group by allows mappers to assign all the occurenceof a given word to the same reducer which is crucial in MapReduce.
- After the mapper is finished processing reducers picks up the mapper's output and increment and store the no of occurrences of the word.

userInput.json



```
{
    "MASTER_ADDRESS" : "0.0.0.0",
    "MASTER_PORT" : 3896,
    "NO_OF_MAPPERS" : 2,
    "MAPPER_TO_USE" : "mapper_word_count.py",
    "MAPPER_DETAILS" : [["0.0.0.0", 5456], ["0.0.0.0", 5457], ["0.0.0.0", 5458], ["0.0.0.0", 5459]],
    "NO_OF_REDUCERS" : 2,
    "REDUCER_TO_USE" : "reducer_word_count.py",
    "REDUCER_DETAILS" :  [["0.0.0.0", 6456], ["0.0.0.0", 6457], ["0.0.0.0", 6458], ["0.0.0.0", 6459]]
}
```

./inputFiles/wordcount.txt



```
the uprising of python
atomic habits
awaken the giant within
giant elephant
the
```

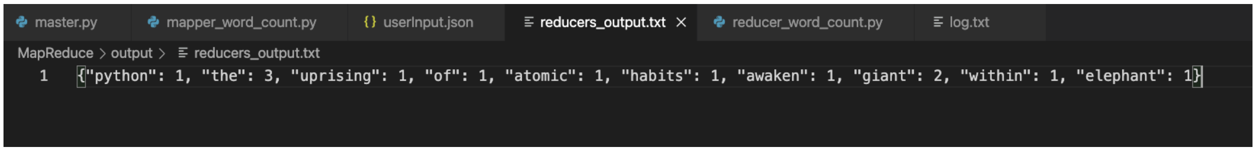
Logs:

Please note: Failing mapper and reducers was done on purpose to prove the system is fault tolerant.
If you want to test this, please uncomment the 2 lines in respective mappers/reducers code files that are responsible for raising an exception.

```
[2023-02-24 18:30:13] [INFO ] Master Node started on 0.0.0.0:3896
[2023-02-24 18:30:13] [INFO ] Process for handling data operations started
[2023-02-24 18:30:15] [INFO ] Total 3901940 bytes to process
[2023-02-24 18:30:15] [INFO ] Mapper 1 will process 1950977 bytes
[2023-02-24 18:30:15] [INFO ] Mapper 2 will process 1950963 bytes
[2023-02-24 18:30:15] [INFO ] Starting 2 mappers
[2023-02-24 18:30:15] [INFO ] Mapper 1 is up
[2023-02-24 18:30:15] [INFO ] Mapper 2 is up
[2023-02-24 18:30:18] [INFO ] Barrier is UP
[2023-02-24 18:30:18] [INFO ] Invoking status checker for 2 mappers
[2023-02-24 18:30:19] [INFO ] Mapper 1 progress: 0%
[2023-02-24 18:30:19] [INFO ] Mapper 2 progress: 0%
[2023-02-24 18:30:19] [ERROR] Mapper 1 is not responding - [Connection Refused]
[2023-02-24 18:30:19] [ERROR] Mapper 2 is not responding - [Connection Refused]
[2023-02-24 18:30:19] [INFO ] Terminating mapper 1
[2023-02-24 18:30:19] [INFO ] Terminating mapper 2
[2023-02-24 18:30:24] [INFO ] Starting mapper 1 again on 0.0.0.0:5456
[2023-02-24 18:30:27] [INFO ] Mapper 1 progress: 0%
[2023-02-24 18:30:28] [INFO ] Mapper 1 progress: 0%
[2023-02-24 18:30:29] [INFO ] Mapper 1 progress: 67%
[2023-02-24 18:30:32] [INFO ] Starting mapper 2 again on 0.0.0.0:5457
[2023-02-24 18:30:35] [INFO ] Mapper 2 progress: 0%
[2023-02-24 18:30:36] [INFO ] Mapper 2 progress: 0%
[2023-02-24 18:30:37] [INFO ] Mapper 2 progress: 62%
[2023-02-24 18:30:42] [INFO ] Mapper 1 progress: 100%
[2023-02-24 18:30:42] [INFO ] Mapper 1 finished processing
[2023-02-24 18:30:50] [INFO ] Mapper 2 progress: 100%
[2023-02-24 18:30:50] [INFO ] Mapper 2 finished processing
[2023-02-24 18:30:55] [INFO ] All the Mappers have Finished Processing and Barrier is down
[2023-02-24 18:30:55] [INFO ] Starting 2 reducers
[2023-02-24 18:30:55] [INFO ] Reducer 1 is up
[2023-02-24 18:30:55] [INFO ] Reducer 2 is up
[2023-02-24 18:30:59] [INFO ] Invoking status checker for 2 reducers
[2023-02-24 18:31:00] [INFO ] Reducer 1 progress: 0%
[2023-02-24 18:31:00] [INFO ] Reducer 2 progress: 0%
```

```
[2023-02-24 18:31:00] [ERROR] Reducer 1 is not responding - [Connection Refused]
[2023-02-24 18:31:00] [INFO ] Terminating reducer 1
[2023-02-24 18:31:01] [INFO ] Reducer 2 progress: 0%
[2023-02-24 18:31:02] [INFO ] Reducer 2 progress: 0%
[2023-02-24 18:31:03] [INFO ] Reducer 2 progress: 0%
[2023-02-24 18:31:04] [INFO ] Reducer 2 progress: 0%
[2023-02-24 18:31:05] [INFO ] Starting reducer 1 again on 0.0.0.0:6456
[2023-02-24 18:31:05] [INFO ] Reducer 2 progress: 0%
[2023-02-24 18:31:08] [INFO ] Reducer 1 progress: 0%
[2023-02-24 18:31:09] [INFO ] Reducer 1 progress: 0%
[2023-02-24 18:31:09] [ERROR] Reducer 1 is not responding - [Connection Refused]
[2023-02-24 18:31:09] [INFO ] Terminating reducer 1
[2023-02-24 18:31:13] [INFO ] Starting reducer 1 again on 0.0.0.0:6456
[2023-02-24 18:31:16] [INFO ] Reducer 1 progress: 0%
[2023-02-24 18:31:17] [INFO ] Reducer 1 progress: 0%
[2023-02-24 18:31:18] [INFO ] Reducer 1 progress: 0%
[2023-02-24 18:31:19] [INFO ] Reducer 1 progress: 0%
[2023-02-24 18:31:20] [INFO ] Reducer 1 progress: 0%
[2023-02-24 18:31:20] [INFO ] Reducer 2 progress: 100%
[2023-02-24 18:31:20] [INFO ] Reducer 2 finished processing
[2023-02-24 18:31:35] [INFO ] Reducer 1 progress: 100%
[2023-02-24 18:31:35] [INFO ] Reducer 1 finished processing
[2023-02-24 18:31:40] [INFO ] All the Reducers have Finished Processing
[2023-02-24 18:31:40] [INFO ] Database processes terminated
[2023-02-24 18:31:40] [INFO ] Run took: 86.80602598190308 seconds
[2023-02-24 18:31:40] [INFO ] Finished Task. Terminating
```

./output/reducers_output.txt



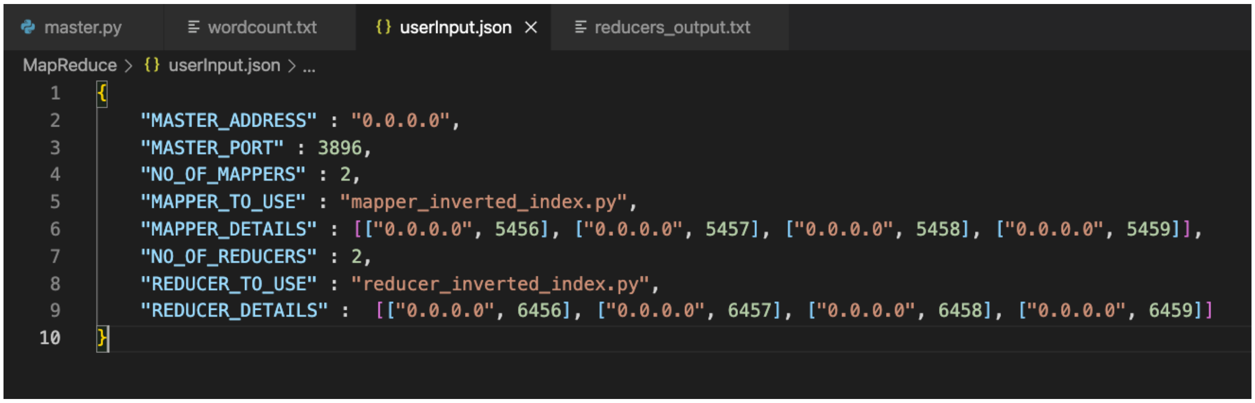
```
{"python": 1, "the": 3, "uprising": 1, "of": 1, "atomic": 1, "habits": 1, "awaken": 1, "giant": 2, "within": 1, "elephant": 1}
```

2) Inverted Index

- Inverted Index returns the character offset and file name for all the words in all of the given text files.
- Each mapper generates a *{word: [file-name, offset]}* for each word that is in range of the bytes assigned to that mapper
- After that it performs *hashing / group by* on that word to get an *reducer_id*
- *Hashing/Group by* allows mappers to assign all the occurence of a given word to the same reducer which is crucial in MapReduce.

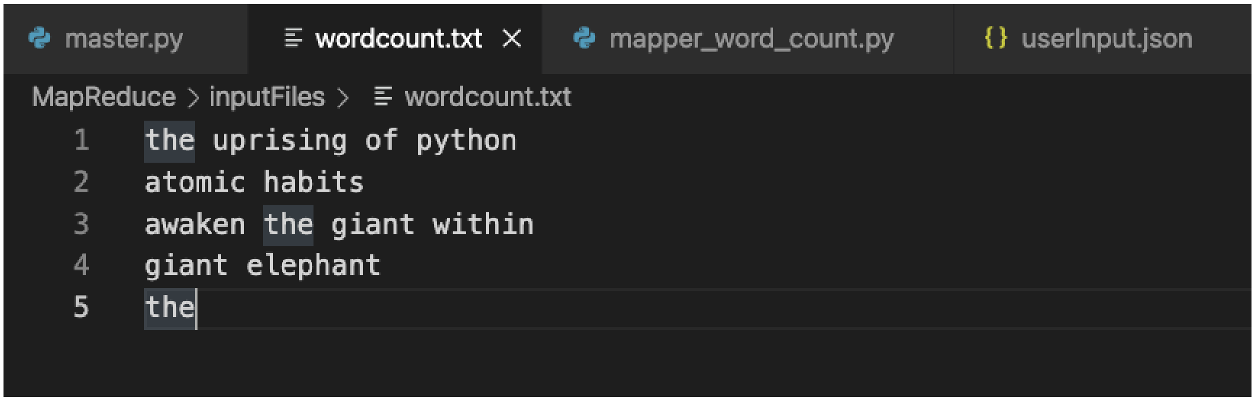
- After the mapper is finished processing reducers picks up the mapper's output and accumulates the words and their corresponding *[file-name, offset]* pairs

userInput.json

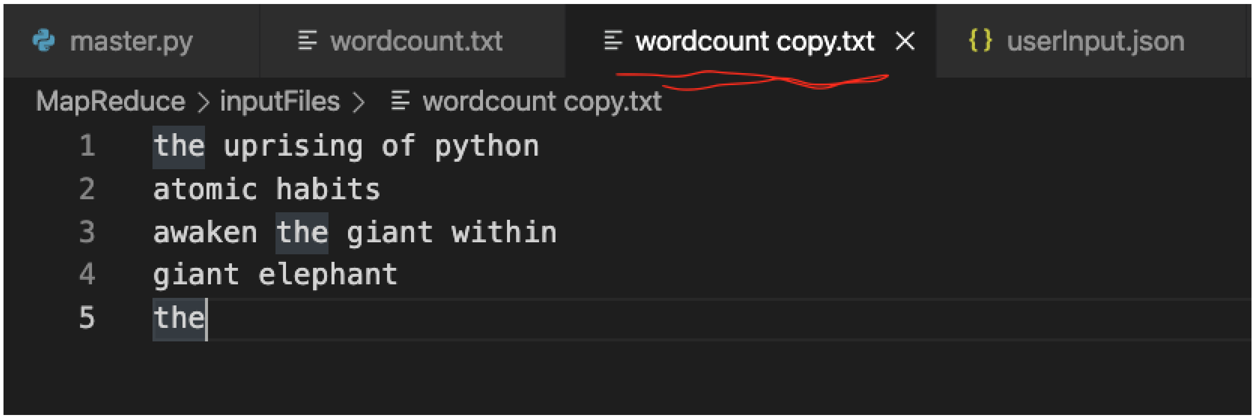


```
{
    "MASTER_ADDRESS" : "0.0.0.0",
    "MASTER_PORT" : 3896,
    "NO_OF_MAPPERS" : 2,
    "MAPPER_TO_USE" : "mapper_inverted_index.py",
    "MAPPER_DETAILS" : [["0.0.0.0", 5456], ["0.0.0.0", 5457], ["0.0.0.0", 5458], ["0.0.0.0", 5459]],
    "NO_OF_REDUCERS" : 2,
    "REDUCER_TO_USE" : "reducer_inverted_index.py",
    "REDUCER_DETAILS" :  [["0.0.0.0", 6456], ["0.0.0.0", 6457], ["0.0.0.0", 6458], ["0.0.0.0", 6459]]
}
```

./inputFiles/wordcount.txt
./inputFiles/wordcount copy.txt



```
the uprising of python
atomic habits
awaken the giant within
giant elephant
the
```



```
the uprising of python
atomic habits
awaken the giant within
giant elephant
the
```

Please note: I’m using multiple text files as inputs in this test case

Logs:

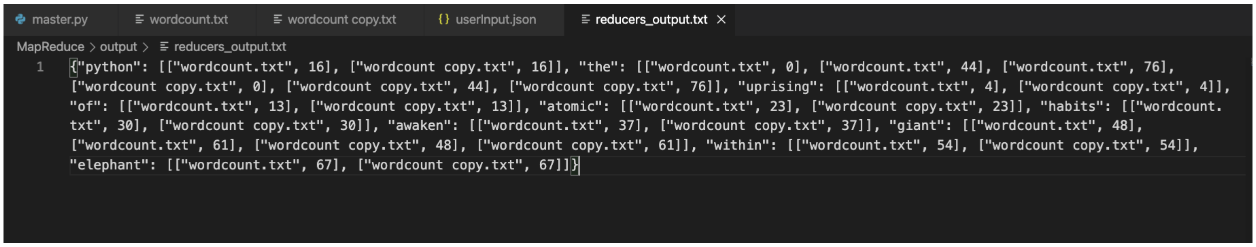
Please note: Failing mapper and reducers was done on purpose to prove the system is fault tolerant.
If you want to test this, please uncomment the 2 lines in respective mappers/reducers code files that are responsible for raising an exception.

```
[2023-02-24 19:05:23] [INFO ] Master Node started on 0.0.0.0:3896
[2023-02-24 19:05:23] [INFO ] Process for handling data operations started
[2023-02-24 19:05:25] [INFO ] Total 7338573 bytes to process
[2023-02-24 19:05:25] [INFO ] Mapper 1 will process 3669296 bytes
[2023-02-24 19:05:25] [INFO ] Mapper 2 will process 3669277 bytes
[2023-02-24 19:05:25] [INFO ] Starting 2 mappers
[2023-02-24 19:05:25] [INFO ] Mapper 1 is up
[2023-02-24 19:05:25] [INFO ] Mapper 2 is up
[2023-02-24 19:05:28] [INFO ] Barrier is UP
[2023-02-24 19:05:28] [INFO ] Invoking status checker for 2 mappers
[2023-02-24 19:05:29] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:05:29] [INFO ] Mapper 2 progress: 0%
[2023-02-24 19:05:29] [ERROR] Mapper 1 is not responding - [Connection Refused]
[2023-02-24 19:05:29] [INFO ] Terminating mapper 1
[2023-02-24 19:05:30] [INFO ] Mapper 2 progress: 49%
[2023-02-24 19:05:34] [INFO ] Starting mapper 1 again on 0.0.0.0:5456
[2023-02-24 19:05:37] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:05:38] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:05:38] [ERROR] Mapper 1 is not responding - [Connection Refused]
[2023-02-24 19:05:38] [INFO ] Terminating mapper 1
[2023-02-24 19:05:40] [INFO ] Mapper 2 progress: 98%
[2023-02-24 19:05:42] [INFO ] Starting mapper 1 again on 0.0.0.0:5456
[2023-02-24 19:05:45] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:05:46] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:05:46] [ERROR] Mapper 1 is not responding - [Connection Refused]
[2023-02-24 19:05:46] [INFO ] Terminating mapper 1
[2023-02-24 19:05:50] [INFO ] Starting mapper 1 again on 0.0.0.0:5456
[2023-02-24 19:05:53] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:05:53] [INFO ] Mapper 2 progress: 100%
[2023-02-24 19:05:53] [INFO ] Mapper 2 finished processing
[2023-02-24 19:05:54] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:05:54] [ERROR] Mapper 1 is not responding - [Connection Refused]
[2023-02-24 19:05:54] [INFO ] Terminating mapper 1
[2023-02-24 19:05:58] [INFO ] Starting mapper 1 again on 0.0.0.0:5456
```

```
[2023-02-24 19:06:01] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:06:02] [INFO ] Mapper 1 progress: 0%
[2023-02-24 19:06:03] [INFO ] Mapper 1 progress: 49%
[2023-02-24 19:06:13] [INFO ] Mapper 1 progress: 98%
[2023-02-24 19:06:23] [INFO ] Mapper 1 progress: 98%
[2023-02-24 19:06:36] [INFO ] Mapper 1 progress: 100%
[2023-02-24 19:06:36] [INFO ] Mapper 1 finished processing
[2023-02-24 19:06:41] [INFO ] All the Mappers have Finished Processing and Barrier is down
[2023-02-24 19:06:41] [INFO ] Starting 2 reducers
[2023-02-24 19:06:41] [INFO ] Reducer 1 is up
[2023-02-24 19:06:41] [INFO ] Reducer 2 is up
[2023-02-24 19:06:44] [INFO ] Invoking status checker for 2 reducers
[2023-02-24 19:06:45] [INFO ] Reducer 1 progress: 0%
[2023-02-24 19:06:45] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:06:45] [ERROR] Reducer 2 is not responding - [Connection Refused]
[2023-02-24 19:06:45] [INFO ] Terminating reducer 2
[2023-02-24 19:06:46] [INFO ] Reducer 1 progress: 0%
[2023-02-24 19:06:47] [INFO ] Reducer 1 progress: 0%
[2023-02-24 19:06:48] [INFO ] Reducer 1 progress: 0%
[2023-02-24 19:06:49] [INFO ] Reducer 1 progress: 0%
[2023-02-24 19:06:50] [INFO ] Starting reducer 2 again on 0.0.0.0:6457
[2023-02-24 19:06:50] [INFO ] Reducer 1 progress: 0%
[2023-02-24 19:06:51] [INFO ] Reducer 1 progress: 0%
[2023-02-24 19:06:52] [INFO ] Reducer 1 progress: 0%
[2023-02-24 19:06:53] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:06:54] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:06:54] [ERROR] Reducer 2 is not responding - [Connection Refused]
[2023-02-24 19:06:54] [INFO ] Terminating reducer 2
[2023-02-24 19:06:58] [INFO ] Starting reducer 2 again on 0.0.0.0:6457
[2023-02-24 19:07:01] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:02] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:03] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:04] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:05] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:06] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:07] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:07] [INFO ] Reducer 1 progress: 100%
[2023-02-24 19:07:07] [INFO ] Reducer 1 finished processing
[2023-02-24 19:07:08] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:09] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:10] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:11] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:12] [INFO ] Reducer 2 progress: 0%
[2023-02-24 19:07:13] [INFO ] Reducer 2 progress: 44%
```

```
[2023-02-24 19:07:28] [INFO ] Reducer 2 progress: 100%
[2023-02-24 19:07:28] [INFO ] Reducer 2 finished processing
[2023-02-24 19:07:33] [INFO ] All the Reducers have Finished Processing
[2023-02-24 19:07:33] [INFO ] Database processes terminated
[2023-02-24 19:07:33] [INFO ] Run took: 130.50399017333984 seconds
[2023-02-24 19:07:33] [INFO ] Finished Task. Terminating
```

./output/reducers_output.txt



```
{"python": [["wordcount.txt", 16], ["wordcount copy.txt", 16]], "the": [["wordcount.txt", 0], ["wordcount.txt", 44], ["wordcount.txt", 76],
["wordcount copy.txt", 0], ["wordcount copy.txt", 44], ["wordcount copy.txt", 76]], "uprising": [["wordcount.txt", 4], ["wordcount copy.txt", 4]],
"of": [["wordcount.txt", 13], ["wordcount copy.txt", 13]], "atomic": [["wordcount.txt", 23], ["wordcount copy.txt", 23]], "habits": [["wordcount.
txt", 30], ["wordcount copy.txt", 30]], "awaken": [["wordcount.txt", 37], ["wordcount copy.txt", 37]], "giant": [["wordcount.txt", 48],
["wordcount.txt", 61], ["wordcount copy.txt", 48], ["wordcount copy.txt", 61]], "within": [["wordcount.txt", 54], ["wordcount copy.txt", 54]],
"elephant": [["wordcount.txt", 67], ["wordcount copy.txt", 67]]}
```

As you can see, the output contains words with all their occurrences in multiple files along with the offset of that file