# MAP REDUCE ASSIGNMENT REPORT

# Design:

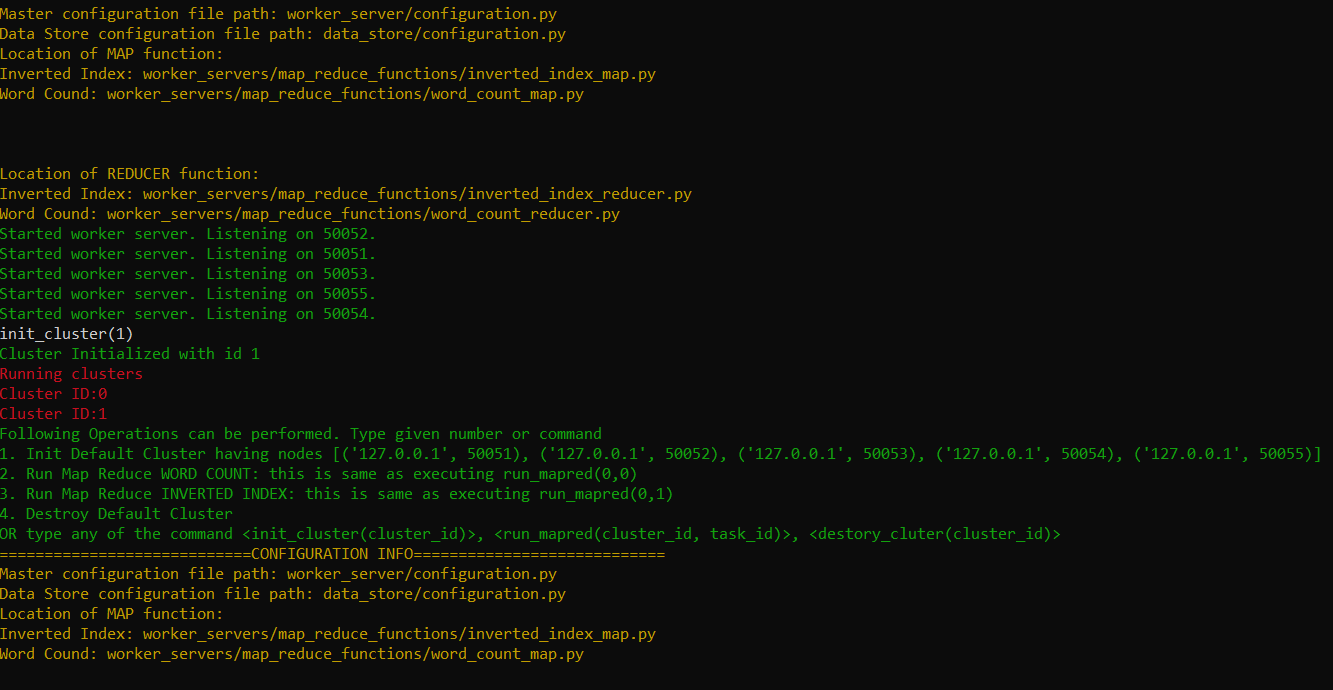
* Only master node communicates with store and other nodes
* Google RPC is used for all the communications among nodes
* Fault tolerance is not implemented
* The data is read in chunks and saved in form of files like: task0, task2. For mappers tasks it is stored in mapper folder and for reducer task it is store in reducer folder.
* All communication with the store in done using RPC calls, initial, intermediate and final data is save using store. The final output can be found in: `data\_store/final\_database.csv`
* Multiple clusters can be initialized using command init\_cluster(<cluster\_id>). For given cluster id the nodes address should be given in the `worker\_server/configuration`.worker\_list
* You can run map reduce on any on the clusters using command run\_mapred(<cluster\_id>, <task\_id>). Task\_id is 0 for word\_count and 1 for inverted\_index
* For the mappers and reducer nodes it is assumed that they cannot have all the data in memory, hence they read chunks from the respective folders which contain tasks for mapper and reducer.
* First all nodes are used as mappers and then all nodes are reused to be reducers.
* The map function can be modified at `rpc/worker\_servers/map\_reduce\_functions`
* For sorting and combining the data it is assumed that the data can be store in the memory
* For input commands either numeric values can be used or specific command can be given.
* Test cases for the word\_count and inverted\_index are given in `rpc/worker\_servers/test\_cases`
* Logs can be found in `rpc/map\_reduce\_logging`
* Following are the examples of the command:
  + run\_map\_red(0,0): runs map red on cluster 0, performs word\_count
  + run\_map\_red(1,0): runs map red on cluster 1, performs word\_count
  + init\_cluster(0): initializes cluster 0
  + init\_cluster(1): initializes cluster 1

## Configuration:

The application configuration can be found at: `rpc/worker\_servers/configuration`

|  |  |  |
| --- | --- | --- |
| S No. | Property | Explaination |
| 1. | configuration\_path | This is just the path of the configuration |
| 2. | worker\_list | List of list of tuples that contain addresses of nodes in cluster. The index of this list denotes the cluster\_id |
| 3. | word\_count\_path | Path of the file on which the word count will be executed |
| 4. | Inverted\_index\_path | Path of the directory on which the inverted\_index will be executed |
| 5. | mapper\_task\_path | Path where the input data is divided and saved for mapper to work on that data chunks. Assuming mapper node cannot have the whole data in memory that’s why the data has been divided in chunks. |
| 6. | reducer\_task\_path | Path where the data is divided in chunks for reducer to work on. Similar memory assumption for reducer node as mapper. |
|  |  |  |

Following is the screen shot running 2 clusters: cluster 0 and cluster 1



## Key-Value Store Configuration:

The configuration for the data store can be found at `rpc/data\_store/configuration.py`

At this path you can change the address where the data store server will be listening for the request and path to where initial, intermediate and final data is stored.

# Logging:

Logs are saved at location `rpc/map\_reduce\_logging/map\_reduce.log`