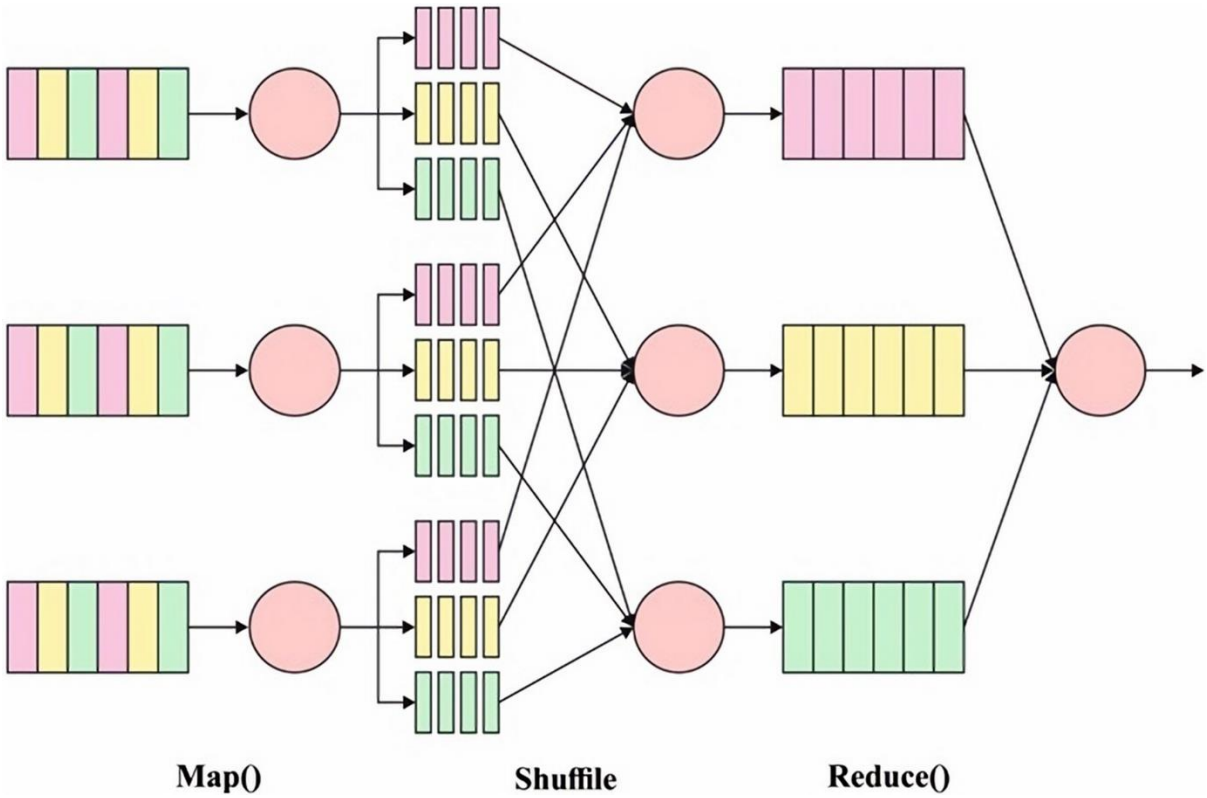
COMP3210/6210 - Big Data
Assignment 1
Semester 2, 2023
Macquarie University, School of Computing



ADITI ANAND

Student ID-47699752

Task 1:

In task 1, we were required to access the date and company values in the movie dataset to generate a series of pairs of year and company. The movie release year was extracted from date and top three production companies were extracted from company column. This pair was stored in a text file and a map reduce program was implemented to calculate the number of movies released by top three production company in each year.

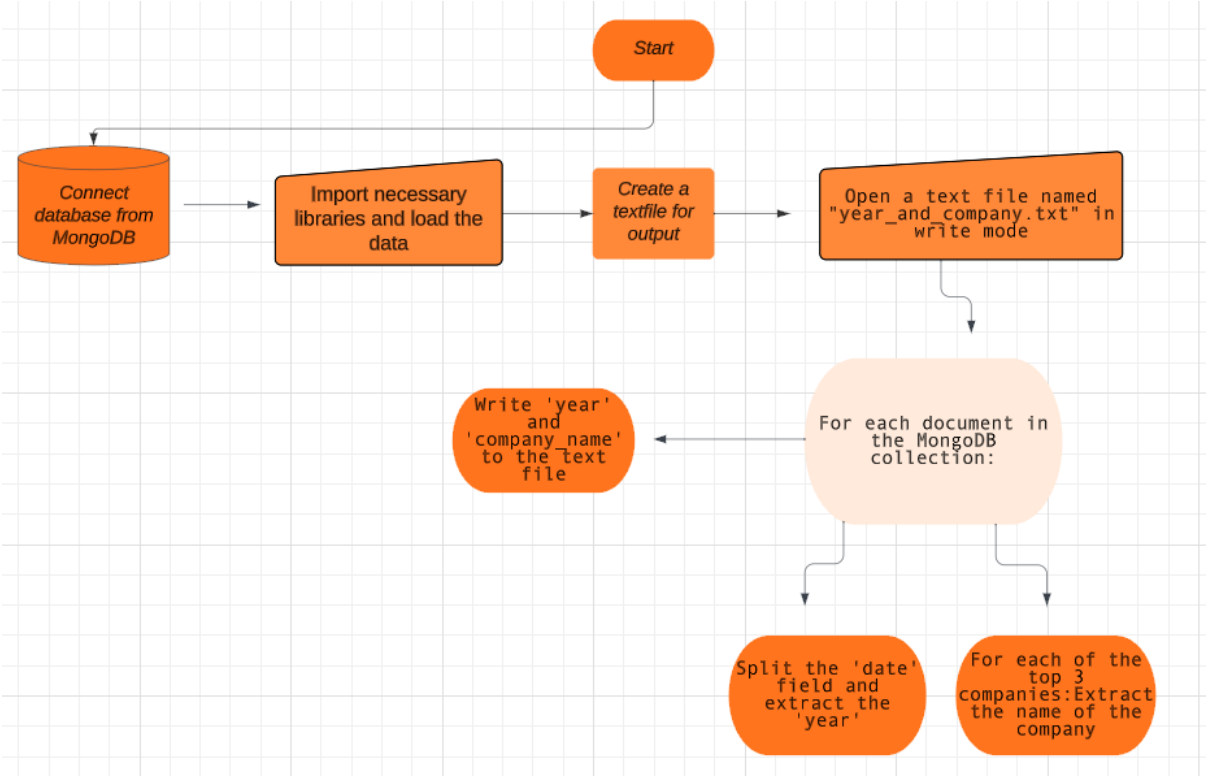
Step 1:

We implemented a python program using pymongo to create the pairs in a text file called “year_and_company.txt”

The Pseudocode for the following is:

- Start by importing the necessary libraries, including pymongo.
- Establish a connection to the MongoDB database and specify the database and collection to work with.
- Create a text file named "year_and_company.txt" for writing.
- Loop through each document in the MongoDB collection.
- For each document, split the date field to extract the year using split_date function.
- Extract the top 3 companies from the document.
- The names of the top 3 businesses are extracted using a loop through the list, and the year and name are then entered into a text file using write_to_file function.
- Finally, close the text file using close_file function.

The flowchart for the following is:



The output file named “year_and_company.txt” is as follows:

```
2009    Ingenious Film Partners
2009    Twentieth Century Fox Film Corporation
2009    Dune Entertainment
2007    Walt Disney Pictures
2007    Jerry Bruckheimer Films
2007    Second Mate Productions
2015    Columbia Pictures
2015    Danjaq
2015    B24
2012    Legendary Pictures
2012    Warner Bros.
2012    DC Entertainment
2012    Walt Disney Pictures
2007    Columbia Pictures
2007    Laura Ziskin Productions
2007    Marvel Enterprises
2010    Walt Disney Pictures
2010    Walt Disney Animation Studios
2015    Marvel Studios
2015    Prime Focus
2015    Revolution Sun Studios
2009    Warner Bros.
```

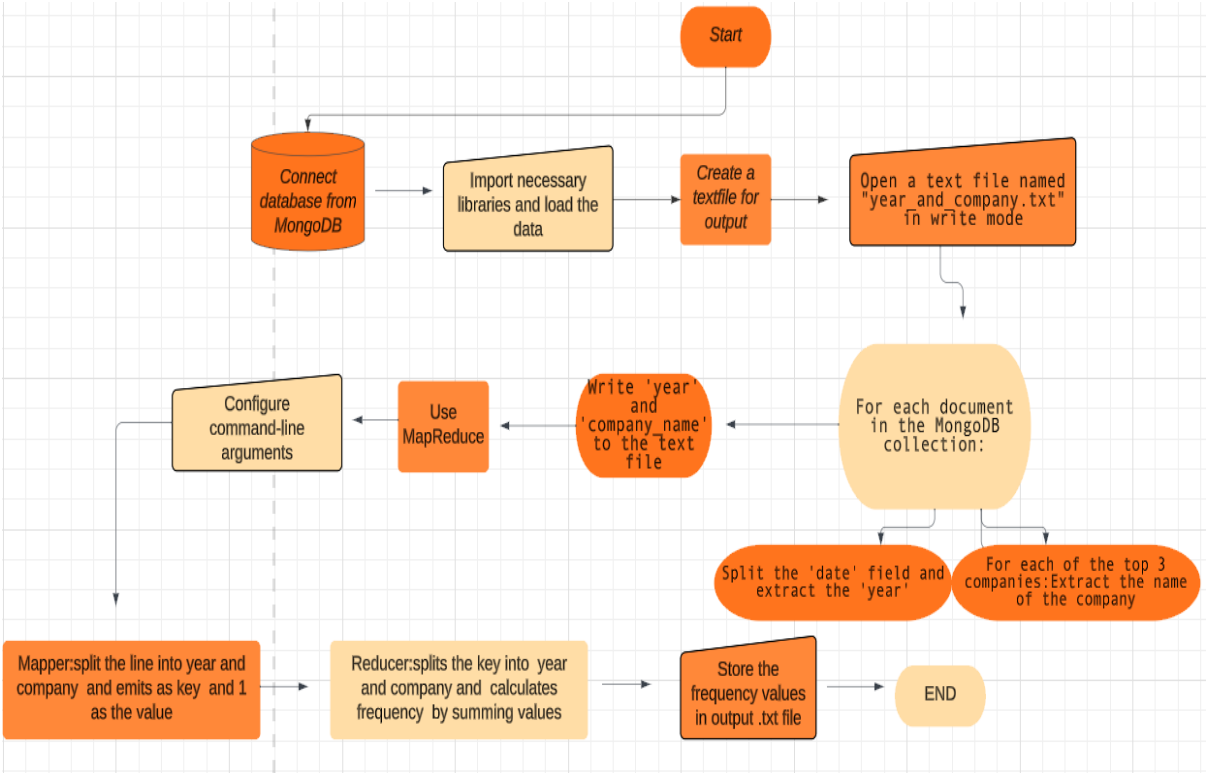
Step 2:

We implemented a MapReduce program to find the number of movies from each company in each year and the outputs were saved in text file called “task1_output”.

The pseudocode for the following is:

- Start by importing the necessary libraries, including MRJob.
- Define a MovieFrequency class that inherits from MRJob.
- The configure_args function is used to configure command-line arguments.
- The split_line function is used by the mapper function to split each input line into the year and company. The output is a key-value pair with the key being a tuple (year, company), and the value being 1.
- When processing key-value pairs, the reducer function divides the key into the year and the company using split_key, calculates the frequency, and outputs the result in the output format that has been selected.
- Finally, in the main entry point, we run the MovieFrequency MapReduce job.

The flowchart for the following is :



The output file named "task1_output.txt" is as follows:

```
"1916, Triangle Film Corporation"       1
"1916, Wark Producing Corp."    1
"1925, Metro-Goldwyn-Mayer (MGM)"       1
"1927, Paramount Pictures"      1
"1927, Universum Film (UFA)"    1
"1929, Metro-Goldwyn-Mayer (MGM)"       1
"1929, Nero Films"      1
"1930, The Caddo Company"       1
"1932, Paramount Pictures"      1
"1933, Paramount Pictures"      1
"1933, Warner Bros."    1
"1934, Columbia Pictures Corporation"   1
"1935, RKO Radio Pictures"      1
"1936, Charles Chaplin Productions"     1
"1936, United Artists"  1
"1936, Warner Bros."    1
"1937, Selznick International Pictures" 1
"1937, United Artists"  1
"1937, Walt Disney Productions" 1
"1938, Columbia Pictures"       1
"1938, Twentieth Century Fox Film Corporation"  1
"1939, Columbia Pictures"       1
"1939, Loew's Incorporated"     1
"1939, Metro-Goldwyn-Mayer (MGM)"       1
```

Task 2:

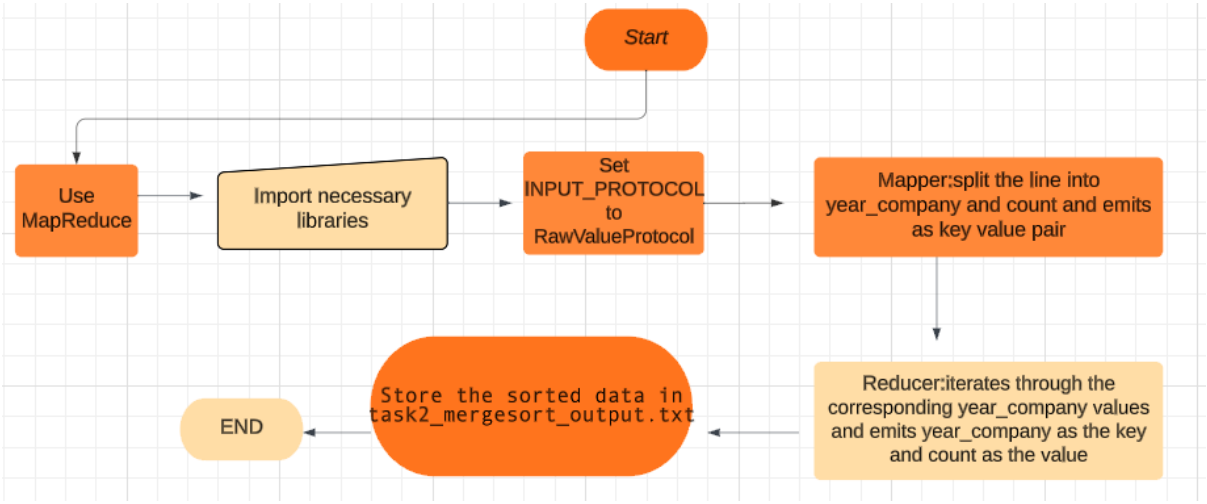
1. **Mergesort:**

We implemented the MapReduce program to employ the mergesort algorithm to sort the pairs of year and company in ascending order. The output data is stored in text file.

The Pseudocode is as follows:

- Import RawValueProtocol and MRJob
- Create a class called MergeSortMRJob and set the INPUT_PROTOCOL to RawValueProtocol.
- Establish a mapper function that reads each line of input, separates it into year_company and count, and outputs (int(count), year_company) as a key-value pair.
- Create a reducer function that takes a count and emits year_company as the key and count as the value after processing each distinct count by iterating over matching year_company values.
- Check if the script is being run as the main program. If it is, run the MergeSortMRJob.

The flowchart for Mergesort algorithm is:



The output file "'task2_mergesort_output.txt" is as follows:

```
"\"1916, Triangle Film Corporation\""   1
"\"1916, Wark Producing Corp.\""        1
"\"1925, Metro-Goldwyn-Mayer (MGM)\""   1
"\"1927, Paramount Pictures\""  1
"\"1927, Universum Film (UFA)\""        1
"\"1929, Metro-Goldwyn-Mayer (MGM)\""   1
"\"1929, Nero Films\""  1
"\"1930, The Caddo Company\""   1
"\"1932, Paramount Pictures\""  1
"\"1933, Paramount Pictures\""  1
"\"1933, Warner Bros.\""        1
"\"1934, Columbia Pictures Corporation\""       1
"\"1935, RKO Radio Pictures\""  1
"\"1936, Charles Chaplin Productions\"" 1
"\"1936, United Artists\""      1
"\"1936, Warner Bros.\""        1
"\"1937, Selznick International Pictures\""     1
"\"1937, United Artists\""      1
"\"1937, Walt Disney Productions\""     1
"\"1938, Columbia Pictures\""   1
"\"1938, Twentieth Century Fox Film Corporation\""      1
"\"1939, Columbia Pictures\""   1
"\"1939, Loew's Incorporated\"" 1
"\"1939, Metro-Goldwyn-Mayer (MGM)\""   1
"\"1939, Selznick International Pictures\""     1
```

2. **Bucketsort:**

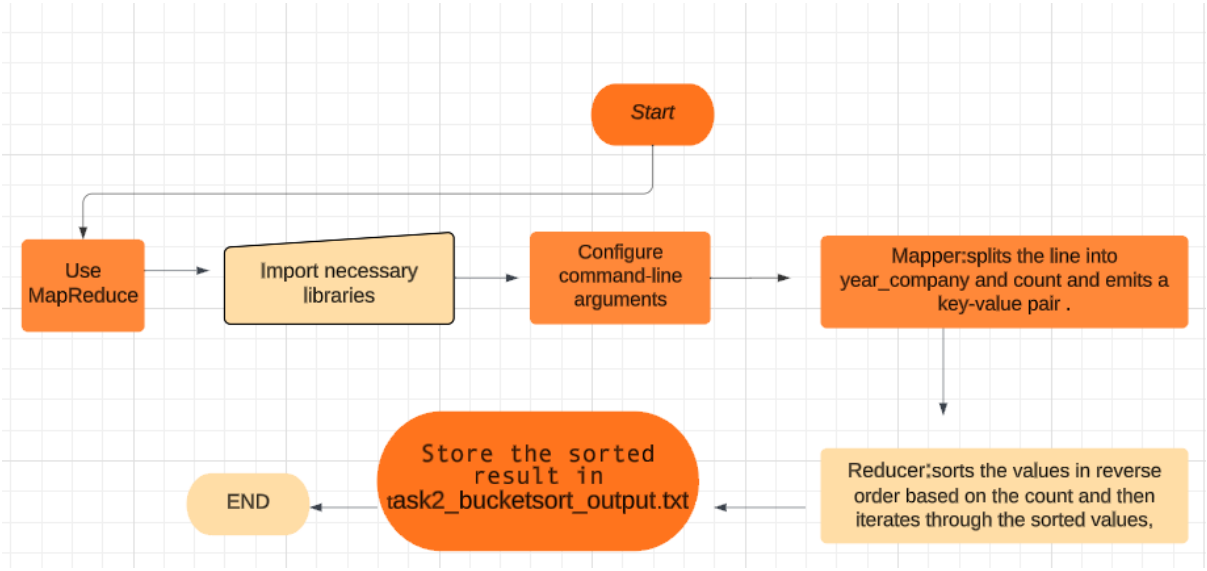
We implemented a MapReduce program to sort the pair of year and company in the descending order based on count of movies and output was stored in a text file.

The Pseudocode for the task is as follows:

- Configure Arguments: Create an argument on the command line that specifies how many top entries should be included in the output.
- Define a BucketSortMRJob class
- Mapper Function: Divide each input line into year_company and count for each input line. Emit a key-value pair with the values int(count), year_company, and None as the key.
- Reducer Function: Reverse-sort the values for each set of values according to the count. If the index is fewer than the desired number of top entries, iterate through the sorted values and output year_company as the key and count as the value.
- End: After processing all input data and producing the desired output based on the top entries, the code comes to an end.

The flowchart for bucketsort algorithm is:

“



The output file “task2_bucketsort_output.txt” is as follows:

```
"\"2009, Universal Pictures\""  15
"\"2006, Universal Pictures\""  15
"\"2005, Universal Pictures\""  15
"\"2002, Paramount Pictures\""  15
"\"2011, Columbia Pictures\""   14
"\"2003, Universal Pictures\""  14
"\"1999, Universal Pictures\""  14
"\"2010, Dune Entertainment\""  13
"\"2008, Universal Pictures\""  13
"\"2006, Twentieth Century Fox Film Corporation\""      13
"\"2004, Paramount Pictures\""  13
"\"2000, Warner Bros.\""        13
"\"2000, Universal Pictures\""  13
"\"2015, Universal Pictures\""  12
"\"2012, Relativity Media\""    12
"\"2011, Universal Pictures\""  12
"\"2010, Universal Pictures\""  12
"\"2010, Relativity Media\""    12
"\"2009, Columbia Pictures\""   12
"\"2002, Universal Pictures\""  12
```