BIG DATA

Zoljargal Batbaatar

I. Task1.1

Summary:

In task 1.1, I used the pymongo library to connect to a MongoDB database. I created a new text file – task1_1_output.txt for storing output and queries the collection to retrieve "Artist", "Year", and "Sales" data fields. Then I extracted values from each document and created a triplet in the format <artist, year, sales>, and added it to the text file. Some artist names consist of several names separated by comma, during the iteration, program was taking only first word of artist name, assuming following words as a year and sales, so I replaced commas in the artist name with dashes.

Pseudocode:

Import the pymongo library

Establish a connection to the MongoDB database

Select the database and collection from which to retrieve data

Create a new text file for storing the output

Query the collection to retrieve the desired data fields

Iterate over the query results

  Extract the values of "Artist", "Year", and "Sales" from each document

  Replace any commas in the artist name with dashes

  Create a triplet in the format <artist, year, sales>

  Write the triplet to the text file

Close the text file and the MongoDB connection

II. Task1.2

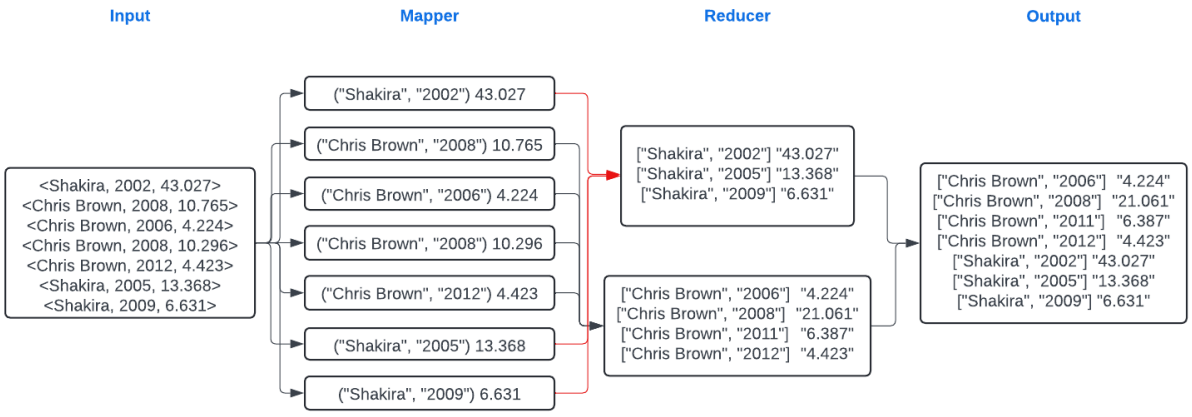
Summary:

In task 1.2, I used MapReduce program to calculate the total sales for each artist in each year from task1_1_output.txt file. The mapper function extracts artist, year, and sales data from input lines and emits key-value pairs. The reducer function sums up the sales values for each artist in each year.

Pseudocode:

```
Import the MRJob library for MapReduce jobs

Import the MRStep library for defining MapReduce steps

Define a class MRTotalSales that inherits from MRJob

    Define a mapper function that takes a key and a line as input

        Split the line into fields

        Extract the artist, year, and sales from the fields

        Try to convert the sales to a float, setting it to 0.0 if the conversion fails

        Emit a key-value pair with the artist and year as the key and sales as the value

    Define a reducer function that takes a key and a list of values as input

        Sum up the sales for each artist in each year

        Emit the total sales for each artist in each year as a string

If the script is run as the main program

    Run the MRTotalSales job
```

Flowchart 1.2:



III. Task 2.1

Summary:

In this task I used Mapreduce to identify the top-selling artist for each year from task1_2_output.txt file. The mapper function extracts artist, year, and sales data from each line, emitting key-value pairs with the year as the key and a tuple containing the artist and sales as the value. The first reducer function then finds the top-selling artist for each year based on sales, emitting key-value pairs with None as the key and a tuple containing the year, top-selling artist, and sales. Finally, the second reducer function sorts the results by year in descending order and emits key-value pairs with the year as the key and a list containing the top-selling artist and sales.

Pseudocode:

Import the MRJob library for MapReduce jobs

Import the MRStep library for defining MapReduce steps

Define a class MRTopSellingArtist that inherits from MRJob

    Define a mapper function that takes a key and a line as input

        Split the line into fields

        Extract the artist, year, and sales from the fields

        Convert sales to a float, removing quotes around the values

Emit a key-value pair with the year as the key and a tuple containing the artist and sales as the value

Define a reducer function that takes a year and a list of tuples as input

Find the top-selling artist for the year based on sales

Emit a key-value pair with None as the key and a tuple containing the year, top-selling artist, and sales as the value

Define another reducer function that takes None and a list of tuples as input:

Sort the list of tuples by year in descending order

Emit key-value pairs with the year as the key and a list containing the top-selling artist and sales as the value

Define the steps for the MapReduce job, consisting of two MRSteps

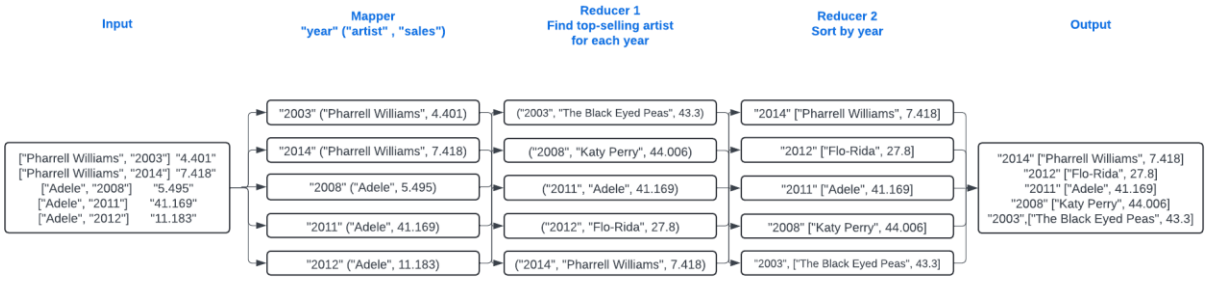
The first step uses the mapper and the reducer_top_artist function to find the top-selling artist for each year

The second step uses the reducer_sort_by_year function to sort the results by year in descending order

If the script is run as the main program

Run the MRTopSellingArtist job

Flowchart 2.1:



IV. Task 2.2

Summary:

In this task, I used MapReduce to identify the top 5 selling artists from task1_2_output.txt file. The mapper function extracts artist names and sales values from each line using regular

expressions and emits key-value pairs with the artist as the key and the sales as the value. The first reducer function sums up the sales for each artist across all time periods and emits key-value pairs with None as the key and a tuple containing the artist and total sales. The second reducer function sorts the artists by total sales in descending order and emits the top 5 artists along with their total sales.

Pseudocode:

Import the MRJob library for MapReduce jobs

Import the MRStep library for defining MapReduce steps

Import the re library for regular expressions

Define a class MRTopSellingArtists that inherits from MRJob

    Define a mapper function that takes a key and a line as input

        Split the line into fields

        Extract the artist and sales from the fields

        Extract the sales value using a regular expression

        Emit a key-value pair with the artist as the key and the sales as the value

    Define a reducer function that takes an artist and a list of sales as input

        Sum up the sales for each artist across all time periods

        Emit a key-value pair with None as the key and a tuple containing the artist and the total sales

    Define another reducer function that takes None and a list of tuples as input

        Sort the list of tuples by total sales in descending order

        Emit the top 5 artists with their total sales

    Define the steps for the MapReduce job, consisting of two MRSteps:

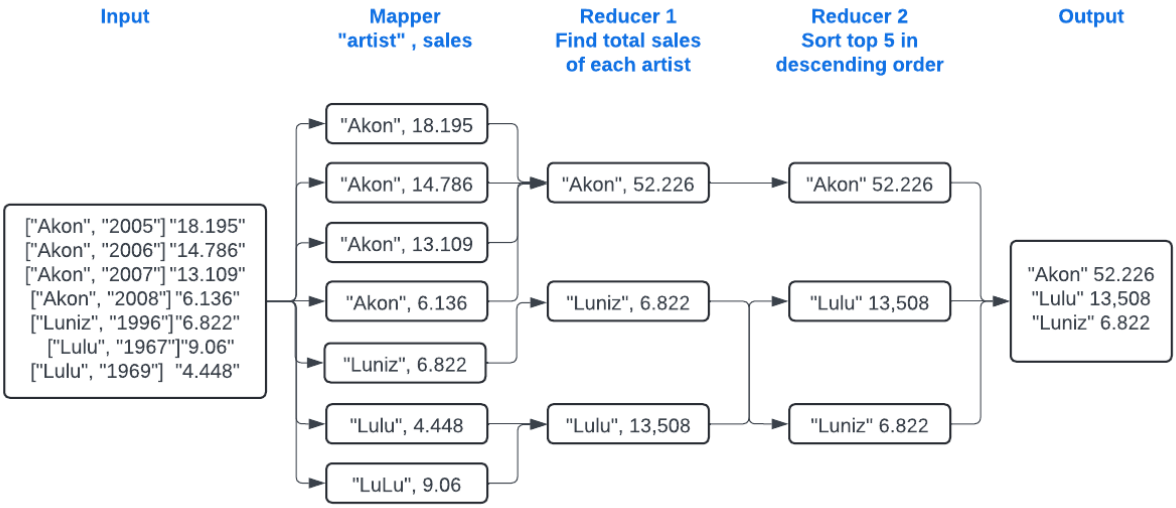
        The first step uses the mapper and the reducer_sum_sales function to sum up the sales for each artist

        The second step uses the reducer_sort_by_sales function to sort the results by total sales and emit the top 5 artists

If the script is run as the main program

Run the MRTopSellingArtists job

Flowchart 2.2:



V. Task 2.3

Summary:

In this task I used MapReduce to find the top-selling artists for each decade from task1_2_output.txt file. The mapper extracts the artist, year, and sales from the input data, calculates the decade for each year, and emits key-value pairs with the decade range and artist as the key, and the sales as the value. The first reducer sums up the sales for each artist within each decade, while the second reducer sorts the artists by total sales in descending order within each decade and emits the top 3 artists. The final reducer sorts the decades in descending order and emits the top artists for each decade.

Pseudocode:

Import the MRJob library for MapReduce jobs

Import the MRStep library for defining MapReduce steps

Import the re library for regular expressions

Define a class MRTopSellingDecade that inherits from MRJob

Define a mapper function that takes a key and a line as input

Split the line into fields

Extract the artist and sales from the fields

Extract the sales value using a regular expression

Calculate the decade for the current year

Emit a key-value pair with the decade range and artist as the key, and the sales as the value

Define a reducer function sum_sales that takes a decade and a list of sales as input

Sum up the sales for each artist within each decade

Emit a key-value pair with the decade as the key and a tuple containing the total sales and artist as the value

Define a reducer function sort_sales that takes a decade and a list of (sales, artist) tuples as input

Sort the artists by total sales in descending order within each decade

Emit the top 3 artists with their total sales for each decade

Define a reducer function decade_sort that takes None and a list of (decade, (total_sales, artist)) tuples as input

Sort the decades in descending order

Emit the top artists for each decade

Define the steps for the MapReduce job, consisting of three MRSteps:

The first step uses the mapper and the reducer_sum_sales function to sum up the sales for each artist within each decade

The second step uses the reducer_sort_sales function to sort the artists by total sales within each decade

The third step uses the reducer_decade_sort function to sort the decades and emit the top artists for each decade

If the script is run as the main program

Run the MRTopSellingDecade job

Flowchart 2.3: