CS236 Project
Spring 2023
TA: Biqian Cheng (biqian.cheng@email.ucr.edu)

MapReduce is a programming model that simplifies data processing. MapReduce is the heart of Apache Hadoop which is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. While we are not directly covering MapReduce in class, this project will expose you to an important technology using ideas from the class (aggregation, grouping, ordering).

The term "MapReduce" refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data. The reduce job takes the output from a map job as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job. The example on the IBM website [1] gives you an idea. You will also find the official Hadoop MapReduce tutorial helpful [2].

A brief overview and introduction to MapReduce can be found in Episode 1 and Episode 2 of the following playlist by Anand Rajaraman of Stanford
https://www.youtube.com/playlist?list=PLLssT5z_DsK9JDLcT8T62VtzwyW9LNepV

Team and Setup:

For this project, you will work in teams of 2. Email the TA in your section your student ID and the names of both team members by May 3rd, 2023.

Some resources for setting up the Hadoop environment can be found here:
Windows – Installing Hadoop 3.2.1 in Windows 10 + basic word count example
MacOS - https://towardsdatascience.com/installing-hadoop-on-a-mac-ec01c67b003c
Many tutorials for Linux Unix are available.

Project Summary:

For this project, we will use MapReduce to find the *most profitable month* for hotel bookings over a period of four years. To do this, compute the total revenues recorded every month and arrange them in descending order. This can give us some interesting information like which months of the year show higher revenue for a particular market segment.
The problem is that the hotel company decided to change the way they keep their data, so there are two separate datasets with different schemas (but similar information) that we need to use.

The first dataset (***hotel-bookings***) provides booking information from 2015 to 2016. The second dataset (***customer-reservations***) provides information about the reservation details from 2017

to 2018. More detailed explanation of the two datasets is below (note: you may not need all the columns for each dataset).

The list of columns in the ***hotel-bookings*** dataset are:
hotel : This column specifies whether the reservation is for the resort hotel or the city hotel.
Booking_status : Value indicating if the booking was canceled (1) or not (0).
lead_time : Number of days that elapsed between the entering date of the booking into the PMS (reservation system) and the arrival date.
arrival_year : Year of arrival date
arrival_month : Month of arrival date with 12 categories: “January” to “December”
arrival_date_week_number : Week number of the arrival date
arrival_date_day_of_month : Day of the month of the arrival date
stays_in_weekend_nights : Number of weekend nights (Saturday or Sunday) the guest stayed for booked to stay at the hotel
stays_in_week_nights : Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel BO and BL/Calculated by counting the number of week nights
market_segment_type : Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
Country : Country of origin where customers belong to
avg_price_per_room: Calculated by dividing the sum of all lodging transactions by the total number of staying nights
email : Emails address of customers who made the bookings

The columns of the ***customer-reservations*** dataset are given below:
Booking_ID: unique identifier of each booking
stays_in_weekend_nights: Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
stays_in_week_nights: Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
lead_time: Number of days between the date of booking and the arrival date
arrival_year: Year of arrival date
arrival_month: Month of arrival date
arrival_date: Date of the month
market_segment_type: Market segment designation.
avg_price_per_room: Average price per day of the reservation; prices of the rooms are dynamic. (in euros)
booking_status: Flag indicating if the booking was canceled or not.

The goal of the project is to find out which month was most profitable from 2015-2018. To do that, arrange the months by revenue.

Formal Problem:

1. Find a way to combine the two given datasets (more efficient is better). Load them into the Hadoop Distributed File System, which will be fed into MapReduce Note that some attributes may be about the same reality but use a different format; for MapReduce to work they should be converted to the same format. (Sub-effective combination methods will affect the Mapper and Reducer performance later on)
2. Find the most popular month of the year for bookings. Using MapReduce to rank the month of the year for booking based on revenue.

Think About:

How many passes should you do? How much work should each pass do?

Due Date:

1. This project will be due on June 11, 2023, 11:59PM
2. Submissions to be made on Canvas/elearn – 1 submission per team.

Deliverables:

1. A zipped file containing:
    a. The jar file/ script to run your job
    b. A report/README file describing your project, including your student ID, email, and name (Read point 2 - below)
2. The report should contain:
    a. Each team member's detailed contribution.
    b. An overall description of how you chose to separate the problem into different MapReduce jobs, with reasoning.
        i. A description of each MapReduce job including:
            1. What the job does
            2. An estimate of runtime for the pass
        ii. A description of how you chose to do the join(s)
        iii. Code snippets
3. Prepare a 15 minute demo about your project. Scheduling a meeting with your TA through Calendly: https://calendly.com/bqcheng/cs236-project-demo-biqian

Note: Your final submission should include all files needed for the script to run (Do not include the two original datasets).

The jar file/ script should take as input three things:
A path to folder containing the Locations file
A path to folder containing the Recordings files
A path for output folder

Potential For Extra Credit:

Please feel free to try to increase the complexity of your project for extra credit. There are many ways that you can do this. Here are a few examples:

1. Good use of combiners.
2. A clever way to achieve faster execution time
3. Enriching the data, e.g., including top five popular booking in each season (if you have your own definition of the season ranges, please state them in your report). Seasons are not an attribute of the datasets. Therefore, instead of returning the most popular month of the year for bookings from the previous step, the corresponding season for it should be returned.
4. Find the most popular payment method (which refers to market segment in the given datasets) for each month. e.g. January, Offline; February, Online; (might not be true in real case).
5. Find more interesting insights from the datasets for instance, e.g. Using the attribute for country of the customer to find which months attract tourists from a particular country.

Whatever you try to do for extra credit, put them into your report/README.

Other:

Use of ChatGPT (or other similar tools) is not allowed for the project and will result in zero points for the project.

[1] https://www.ibm.com/topics/mapreduce
[2] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
[3] https://hadoop.apache.org/release/3.2.1.html
[4] https://www.oracle.com/java/technologies/downloads/#java8
[5] https://www.java.com/en/download/