Assignment: MapReduce Methodology for Analyzing Student Grades

Overview

This assignment explores the MapReduce methodology for analyzing student grades using the ”UniversityCoursesGrades” dataset. The dataset is provided in the file `coursegrades.txt`, where each line follows the format:

(Year, Course Name, Student Grade, University Name)

Each value is separated by a comma.

Tasks

Note: For each task, after implementing the MapReduce process, provide a concrete example using a sample dataset.

Task 1: Average Grade per Course

Find the average grade for each course and determine which course has the highest average.

Mapper:

- Generates key-value pairs: `(Course Name, Grade)` e.g., (Distributed, 100)

Reducer:

- Receives `(Course Name, (G1, G2, ...))`, e.g., (Distributed, (100, 55, 85))
- Computes the average grade for each course
- Outputs: `(Course Name, Average Grade)`, e.g., (Distributed, 80)

Example Execution:
Input File (`coursegrades.txt`):

```
2023, Distributed , 55, GIU
2023, Distributed , 85, GIU
2023, Security , 60, GIU
2023, Security , 40, GIU
```

Expected Output:

```
Distributed , 70
Security , 50
```

Task 2: Average Grade per University

Find the average grade for each university and determine which university has the highest average.

Mapper:

- Generates key-value pairs: `(University Name, Grade)` e.g., (GIU, 200), (GIU, 60)

Reducer:

- Receives `(University Name, (G1, G2, ...))`, e.g., (GIU, (200, 60, 70))
- Computes the average grade for each university
- Outputs: `(University Name, Average Grade)`, e.g., (GIU, 110)

Example Execution:
Input File (`coursegrades.txt`):

```
2024, Security, 90, GIU
2024, Security, 80, GIU
2024, Distributed System, 70, GUC
```

Expected Output:

```
GIU, 85
GUC, 70
```

Bonus Task: Top 3 Highest Grades per Year

Identify the top 3 highest student grades recorded for each academic year.

Mapper:

- Generates key-value pairs: `(Year, Grade)` e.g., (2025, 80)

Reducer:

- Receives `(Year, [G1, G2, ...])`, e.g., (2025, [80, 60, 50, 100, ...])
- Sorts the grades in descending order
- Extracts the top 3 highest grades
- Outputs: `(Year, [Top 3 Grades])`, e.g., (2025, [100, 80, 60])

Example Execution:
Input File (`coursegrades.txt`):

```
2025, Distributed, 80, Harvard
2025, Security, 60, MIT
2025, AI, 50, Stanford
2025, Networks, 100, Oxford
2025, Systems, 90, Cambridge
```

Expected Output:

```
2025, [100, 90, 80]
```

Deliverables:

1. **Your code must be submitted via the Google Form link:**
   `https://forms.gle/ue3RWVR83edUZhmFA`
   (No emails will be accepted)
   Note: Only the team leader should submit the project.
2. **Submit the notebook** that contains both tasks and the obtained output files in a **zipped file**.
3. **Submit a PDF file** containing:
   - Screenshots of the code
   - Screenshots of the output files
   - A brief description of how you solved the tasks
4. **Make sure to comment on every step while coding.**
5. **Feel free to use any other Python packages.**

Important Notes:

1. **Plagiarism is not tolerated, and copied work will result in a score of 0.**
2. **Groups must have the same TA** (different groups with the same TA are allowed). Each team can have up to **4 members**.

Deadline:

The deadline is **14th March 2025 at 11:59 PM**.