CS3.304 - AOS Assignment 5 - Report

About:

Py-SPARK is a powerful open-source distributed computing system that provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. In this assignment, we utilized Py-SPARK to analyze the House Pricing dataset. The dataset contains information on property sales in England and Wales, including transaction unique identifiers, prices, and countries.

Objective:

IIn this report, I provide a comparison of the time taken for three distinct tasks using different numbers of CPU cores. The findings are presented in three sections, each corresponding to the performance with 2, 4, and 6 cores. Additionally, visual representations have been included to illustrate the performance trends for each task under varying core configurations. This comparison aims to offer insights into the efficiency of parallel processing across different core counts, facilitating a comprehensive understanding of the computational performance.

How to Run the code:

To execute the Py-SPARK programs on Abacus, follow the instructions provided:

Login to Abacus:

```
ssh cs3304.24@abacus.iiit.ac.in
```

Create Virtual Environment and Install Py-SPARK:

```
pip install pyspark
```

Copy Files to Abacus:

```
scp <source_file_path> <user_id>@abacus.iiit.ac.in:<destination_path>
```

Requesting an Interactive Job:

```
sint3 -c 2/4/6
```

Execute Python Program:

```
python3 2023201024_q1/q2/q3.py
```

Relinquish the Job:

```
exit
```

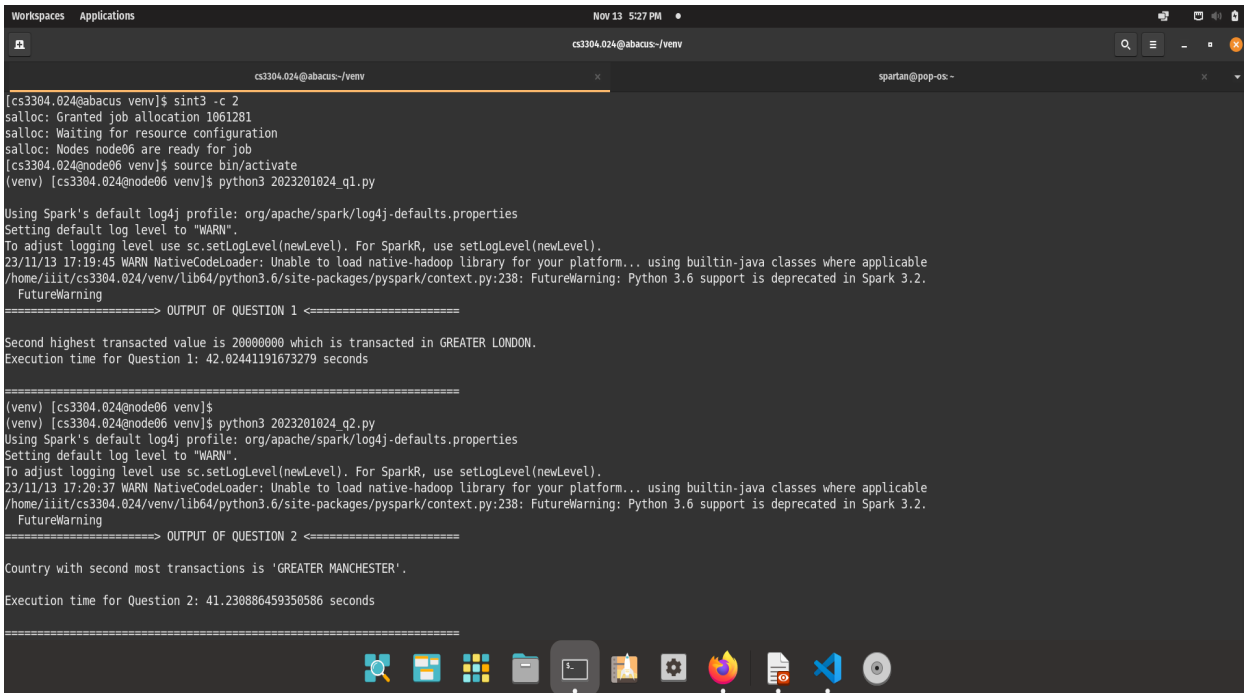
Comparison:

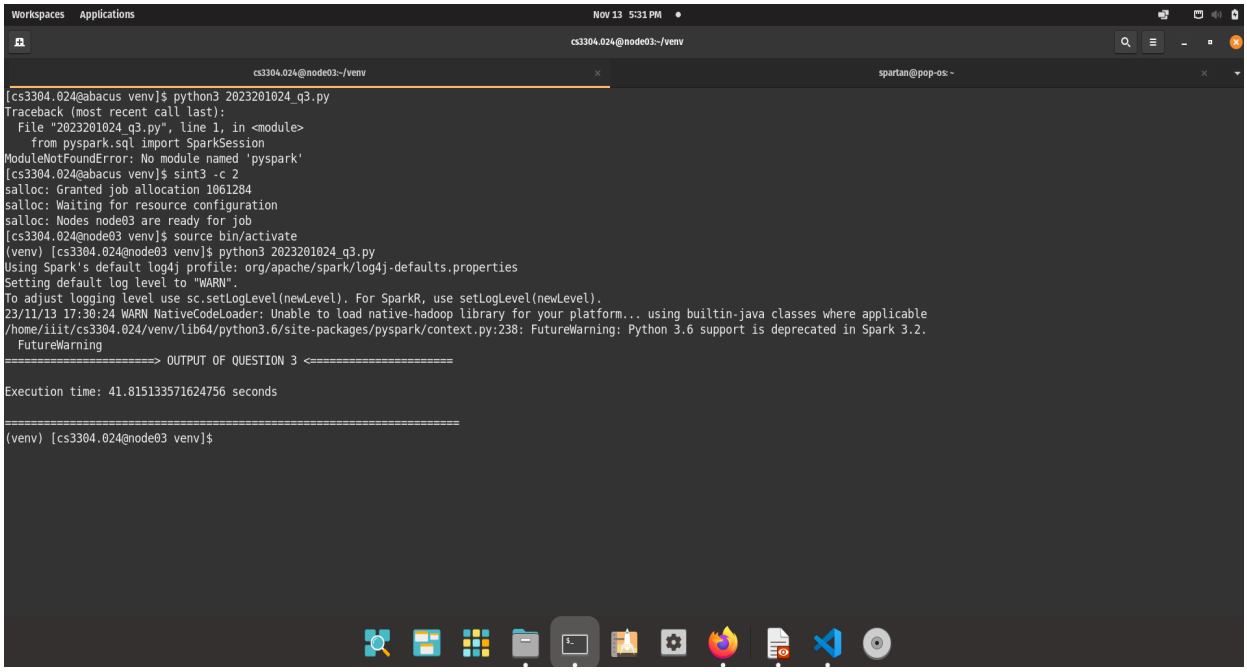
---

2 Cores:

Time taken using 2 cores are as follows

- Question 1: 42.02 seconds
- Question 2: 41.23 seconds
- Question 3: 41.81 seconds



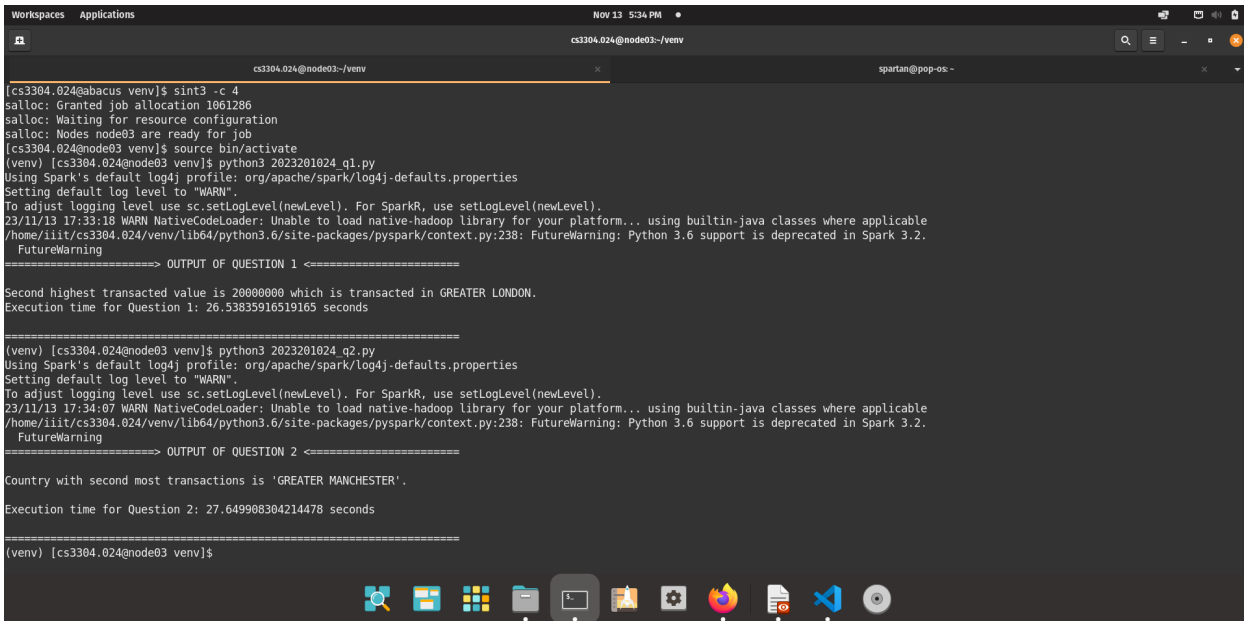


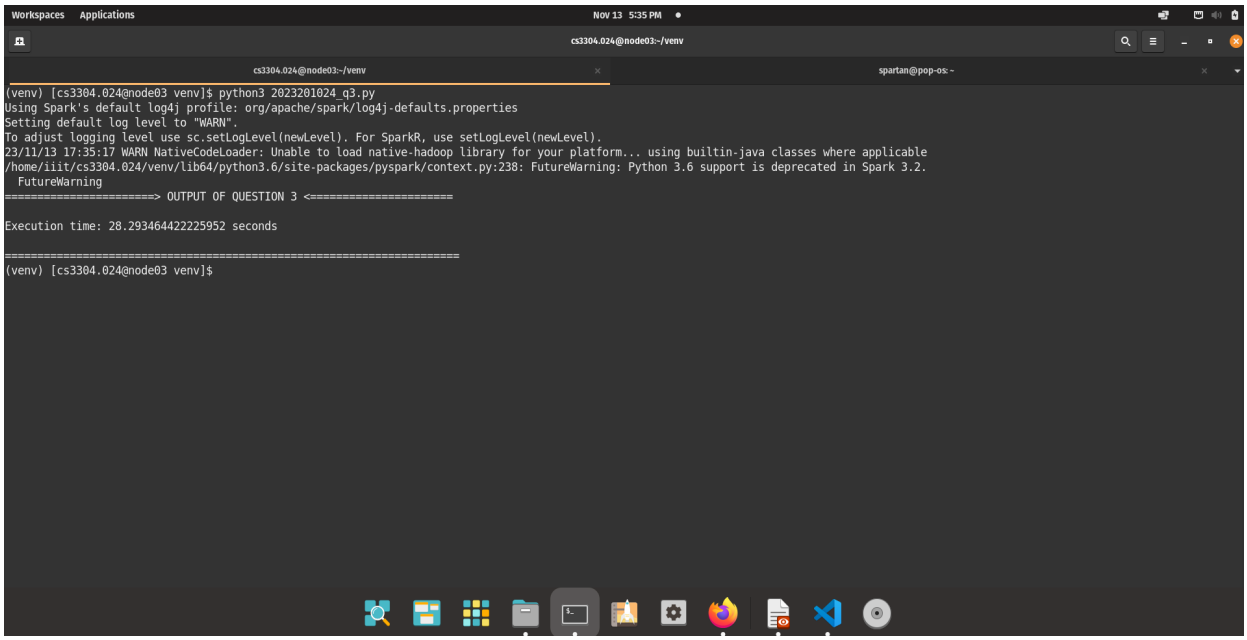
---

4 Cores:

Time taken using 2 cores are as follows

- Question 1: 26.53 seconds
- Question 2: 27.64 seconds
- Question 3: 28.29 seconds





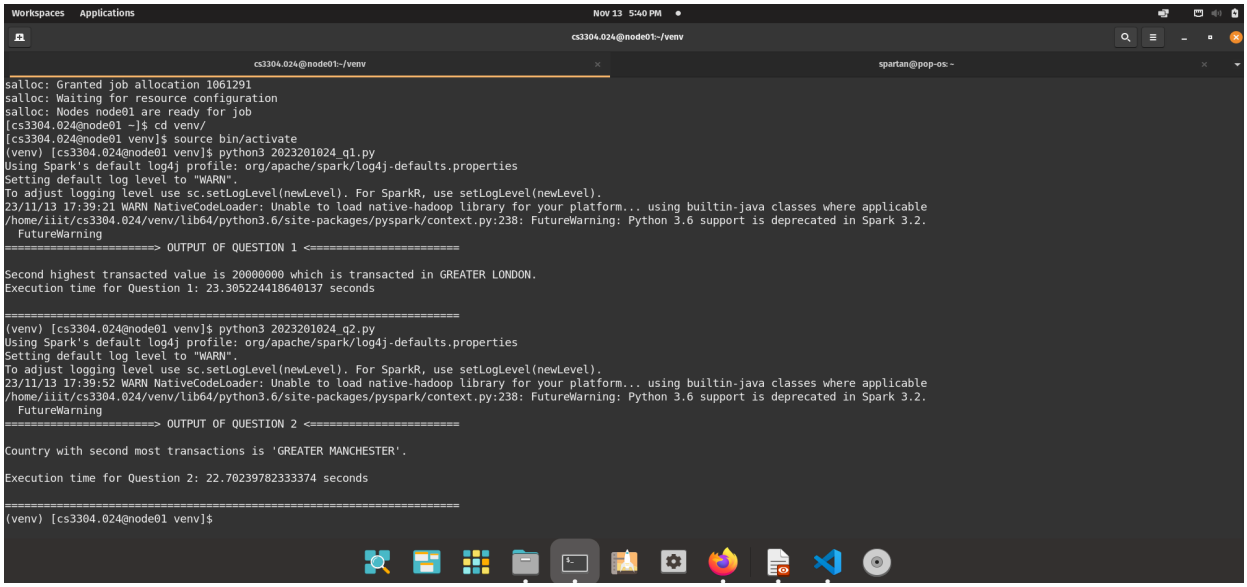
---

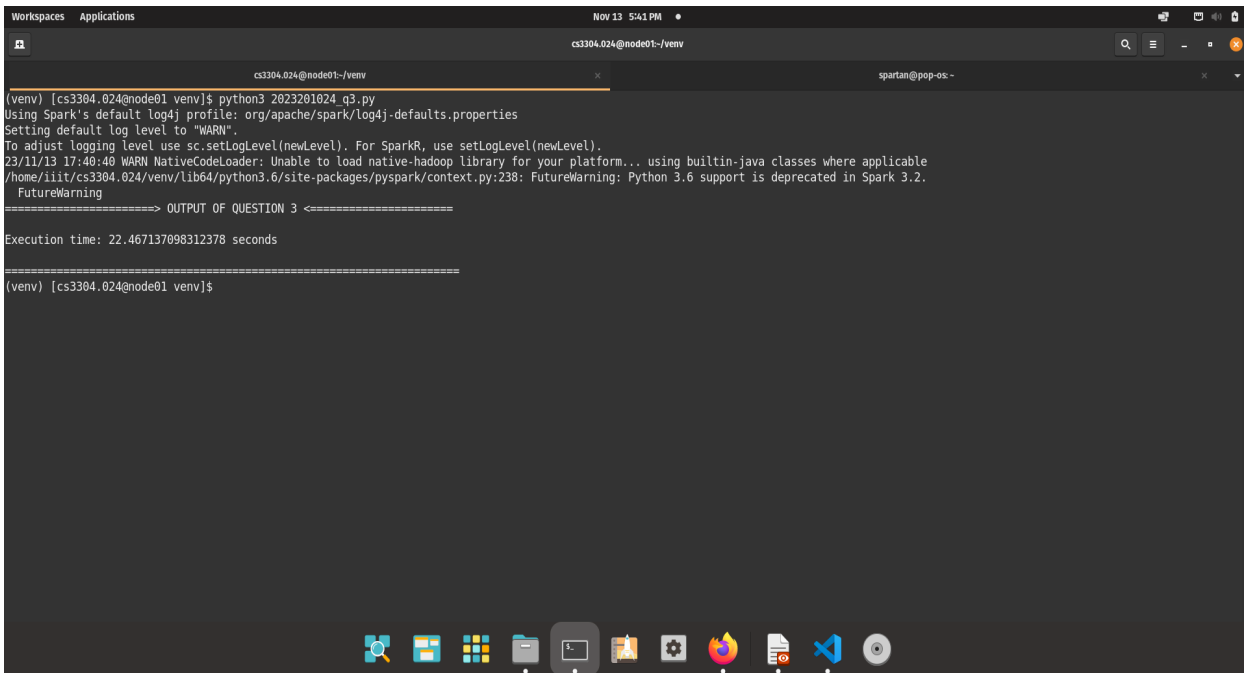
6 Cores:

Time taken using 2 cores are as follows

- Question 1: 23.30 seconds

- Question 2: 22.70 seconds
- Question 3: 22.46 seconds





Results:

Table:

Cores	Question 1	Question 2	Question 3
2	42.02s	41.23s	41.81s
4	26.53s	27.64s	28.29s
6	23.30s	22.70s	22.46s

Observations:

- As the number of cores increases from 2 to 6, there is a noticeable decrease in the time taken for all three questions.
- Question 3 consistently shows a faster execution compared to Question 1 and Question 2 across different core configurations.
- The reduction in time is more prominent when moving from 2 to 4 cores than from 4 to 6 cores, indicating diminishing returns with additional cores.

These observations suggest that parallel processing with a higher number of cores contributes to improved efficiency, particularly noticeable in the transition from 2 to 4 cores. However, the gains become less significant when moving from 4 to 6 cores. The choice of the number of cores should be considered in light of achieving an optimal balance between computational power and resource utilization.