Final Exam

Report

Semester 20243

Subject	: **Distributed and Parallel System**
Study Program	: Informatics
Student Name	: mico martin
Student ID	: 001202300179
Class	: IT Class 2

1. Introduction

This report presents a performance comparison of three different data processing approaches (Sequential, Threading, and Multiprocessing) applied to the trip_duration column of the NYC Taxi Trip Duration dataset. Two key operations are performed:

- Filtering: Keeping only records where trip_duration > 1000 seconds.
- Sorting: Sorting the filtered data in ascending order.

Key Findings:

- Sequential processing consistently outperformed both threading and multiprocessing approaches
- Multiprocessing showed severe performance degradation with ~68-75 second execution times
- Threading showed modest overhead compared to sequential processing
- Performance scaling is linear for sequential and threading, but not for multiprocessing

2. System Specification

```
==================================================
SYSTEM SPECIFICATIONS
==================================================
Processor: Intel64 Family 6 Model 154 Stepping 4, GenuineIntel
System: Windows 10
Architecture: 64bit
CPU Cores: 10 physical, 12 logical
RAM: 15.73 GB
Python Version: 3.11.9
==================================================
```

3. Dataset Overview

Total Records: 1,458,644 rows
Target Column: trip_duration (seconds)
Filtering Criteria: trip_duration > 1000 seconds
Trip duration stats:

- count: 1,458,644
- mean: 959.49 seconds
- std: 5,237.43 seconds
- min: 1.00 second
- 25%: 397.00 seconds
- 50%: 662.00 seconds
- 75%: 1,075.00 seconds
- max: 3,526,282.00 seconds

4. Performance Analysis

```
================================================================================
DETAILED PERFORMANCE RESULTS
================================================================================

--- 25% of data ---
Sequential     : Time: 0.0318s, Results: 103,486, CPU Avg: 0.0%, CPU Max: 0.0%, Memory Avg: 76.5%
Threading      : Time: 0.0556s, Results: 103,486, CPU Avg: 0.0%, CPU Max: 0.0%, Memory Avg: 76.5%
Multiprocessing: Time: 68.2319s, Results: 103,486, CPU Avg: 20.6%, CPU Max: 96.0%, Memory Avg: 83.8%

--- 50% of data ---
Sequential     : Time: 0.0793s, Results: 206,976, CPU Avg: 4.8%, CPU Max: 9.5%, Memory Avg: 74.6%
Threading      : Time: 0.0980s, Results: 206,976, CPU Avg: 12.3%, CPU Max: 24.6%, Memory Avg: 74.7%
Multiprocessing: Time: 68.1709s, Results: 206,976, CPU Avg: 24.3%, CPU Max: 100.0%, Memory Avg: 82.9%

--- 75% of data ---
Sequential     : Time: 0.1200s, Results: 310,133, CPU Avg: 6.3%, CPU Max: 12.6%, Memory Avg: 73.2%
Threading      : Time: 0.1524s, Results: 310,133, CPU Avg: 7.0%, CPU Max: 15.4%, Memory Avg: 73.3%
Multiprocessing: Time: 66.8967s, Results: 310,133, CPU Avg: 19.9%, CPU Max: 100.0%, Memory Avg: 82.1%

--- 100% of data ---
Sequential     : Time: 0.1412s, Results: 413,577, CPU Avg: 6.9%, CPU Max: 13.8%, Memory Avg: 72.7%
Threading      : Time: 0.1936s, Results: 413,577, CPU Avg: 9.3%, CPU Max: 17.0%, Memory Avg: 72.7%
Multiprocessing: Time: 74.9463s, Results: 413,577, CPU Avg: 21.6%, CPU Max: 100.0%, Memory Avg: 83.1%
```

- Filtered Counts (trip_duration value that is greater (>) than 1000)

Data Size	Records	Filtered Row
25%	364,661	103,486
50%	729,322	206,976
75%	1,093,983	310,133
100%	1,458,644	413,557

- Execution Time (in second)

Data Size	Sequential	Threading	Multiprocessing
25%	0.0318	0.0576	62.2319
50%	0.0793	0.0980	68.1709
75%	0.1200	0.1524	66.8967
100%	0.1412	0.1936	74.9463

1. Sequential processing

Characteristics:

- Performance: Consistently fastest across all data sizes
- CPU Usage: Low to moderate (0-13.8% max)
- Memory Usage: Stable (~72-77% average)
- Scalability: Linear scaling with data size

Sequential processing demonstrates excellent performance for I/O-bound filtering operations. The execution time scales linearly with data size (approximately 2x time for 2x data), indicating optimal efficiency for this type of operation.

2. Threading Processing

Characteristics:

- Performance: 24-75% slower than sequential
- CPU Usage: Slightly higher than sequential (up to 24.6% max)
- Memory Usage: Similar to sequential (~72-77% average)
- Scalability: Linear scaling with modest overhead

Threading shows consistent overhead. For CPU-bound filtering operations, threading doesn't provide benefits and introduces context-switching overhead. However, the performance degradation is manageable and maintains linear scaling.

3. Multiprocessing

Characteristics:

- Performance: Severely degraded (530-2,146x slower)
- CPU Usage: High utilization (up to 100% max, 19-24.6% average)
- Memory Usage: Higher than other approaches (~82-84% average)
- Scalability: Non-linear, with high fixed overhead

Multiprocessing shows performance degradation due to:

- Process Creation Overhead: Significant time spent spawning processes
- Data Serialization: Large dataset serialization/deserialization between processes
- Inter-Process Communication: Overhead in result collection
- Memory Duplication: Each process maintains its own copy of data

Multiprocessing is non-linear, since it takes extra time and resources to starting and managing processes.

5. Conclusion

Sequential Processing is Optimal: For filtering operations on this dataset size, sequential processing provides the best performance with minimal resource overhead.

Threading Shows Modest Overhead: While not beneficial for this CPU-bound task, threading maintains reasonable performance with only 24-75% overhead.

Multiprocessing is Counterproductive: The overhead of process creation and data serialization completely negates any potential parallelization benefits.

Linear Scalability: Sequential and threading approaches scale linearly with data size, while multiprocessing performance is dominated by fixed overhead.

6. Appendix

https://github.com/mico-m31/final_dps