Final Project 2023S SSW567-WS

SSW 567-WS
Prof. Andre Bondi
Part3-PerfTesting
Team: C

Performance Engineers
Arun Rao Nayineni
Dhruv Patel
Ruchitha Paithankar

Performance Testing:

GitHub Repository: https://github.com/ArunRao1997/SSW-567-Final-Project

Spreadsheet:
https://docs.google.com/spreadsheets/d/1N0yBNzZ9DUsn9lkimBLh68l4i6ReARdEUsQa7T1u2-s/edit?usp=sharing

Performance Testing:

Performance testing evaluates the speed, scalability, and stability of a Python application or system under various workloads and conditions. It identifies potential bottlenecks or issues that could impact performance. This testing is typically conducted during the development and testing phases to ensure the application or system can handle the anticipated workload and meet user performance requirements.

Report and Results:

An essential aspect of software testing is assessing code performance to determine its efficiency. Inefficient code might require refactoring. In this project, we simulated performance testing by measuring the time taken to parse 'n' lines, ranging from 100 to 10,000. Our code demonstrated a complexity of O(n).

We recorded two instances of our performance score. After measuring the initial performance, we refactored the code to minimize the number of mutants generated during mutation testing.

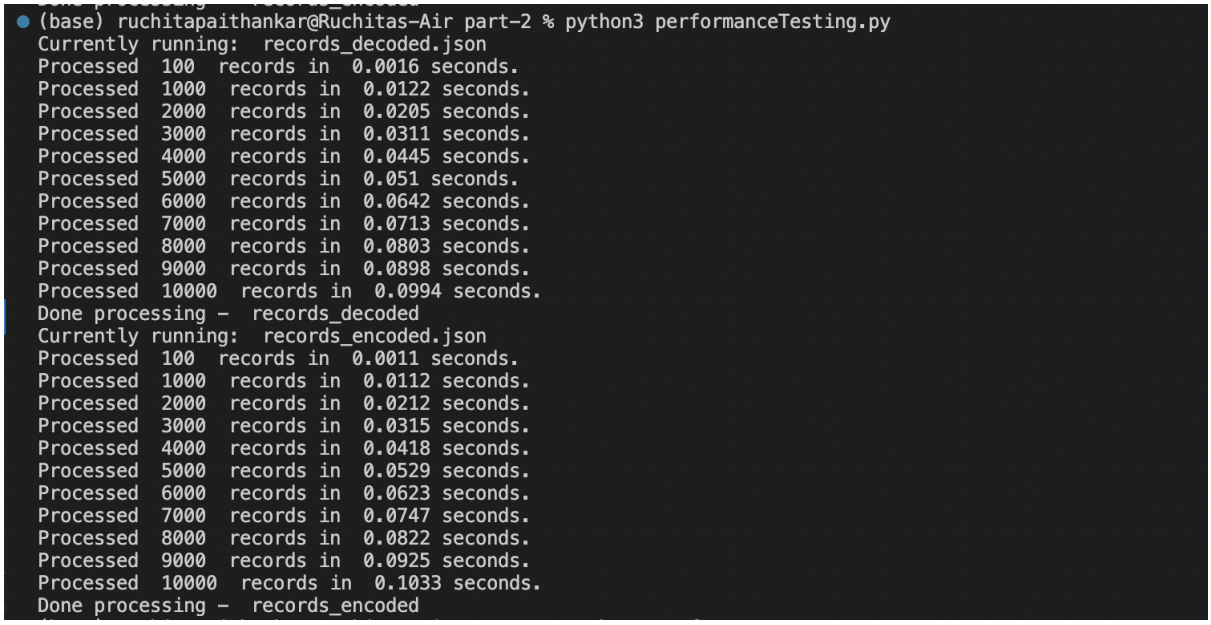

Figure 1: Screen Dump of Performance Test Performed Locally

To verify that the performance was not affected, we measured the performance again. We observed only a small increase in the time taken to complete each test, and the code maintained a complexity of O(n).

Figures:

Lines read (n)	Encode Performance (s)	Decode Performance (s)	Encode Performance Modified (s)	Decode Performance Modified (s)
100	0.00368379200	0.00323529200	0.003850708	0.003387208
1,000	0.03592770800	0.03185566700	0.037632041	0.033392291
2,000	0.07190883400	0.06399629200	0.074555916	0.065088083
3,000	0.10682275000	0.09675112500	0.112522625	0.098394542
4,000	0.14279737500	0.12927483300	0.14915825	0.130758042
5,000	0.17801850000	0.15994108400	0.187406583	0.16328225
6,000	0.21418037500	0.19589658300	0.224193333	0.19820675
7,000	0.24956683400	0.22731129100	0.265123459	0.226738083
8,000	0.28483079100	0.26147708400	0.298536125	0.2604685
9,000	0.32003737500	0.28104670800	0.33581875	0.284740125
10,000	0.35544920900	0.31050629100	0.373566042	0.316379333

Figure 1

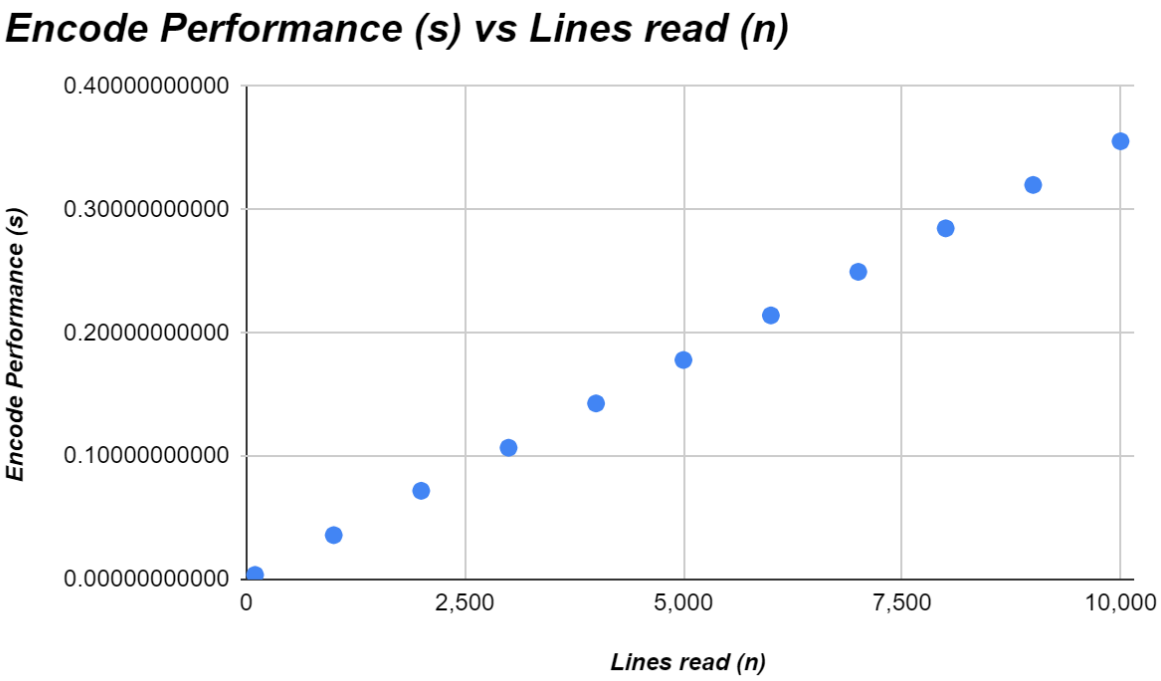
Encode Performance (s) vs Lines read (n)
0.40000000000
0.30000000000
0.20000000000
0.10000000000
0.00000000000
Encode Performance (s)
0
2,500
5,000
7,500
10,000
Lines read (n)

Figure 2

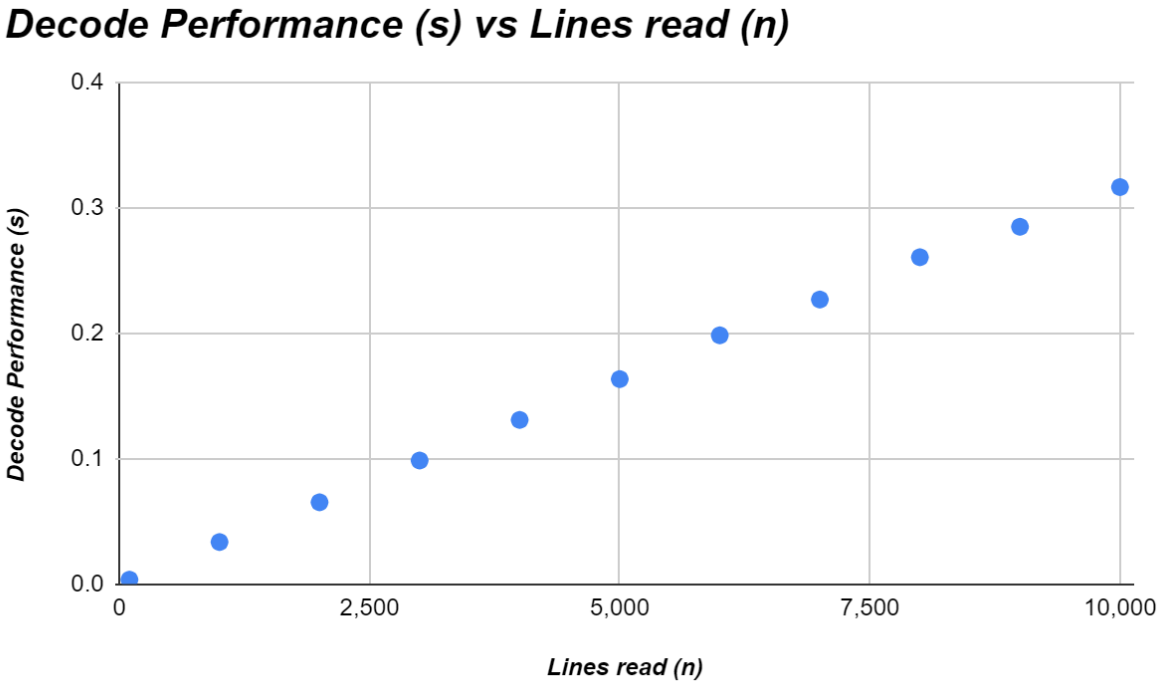
Decode Performance (s) vs Lines read (n)
0.4
0.3
0.2
0.1
0.0
Decode Performance (s)
0
2,500
5,000
7,500
10,000
Lines read (n)

Figure 3

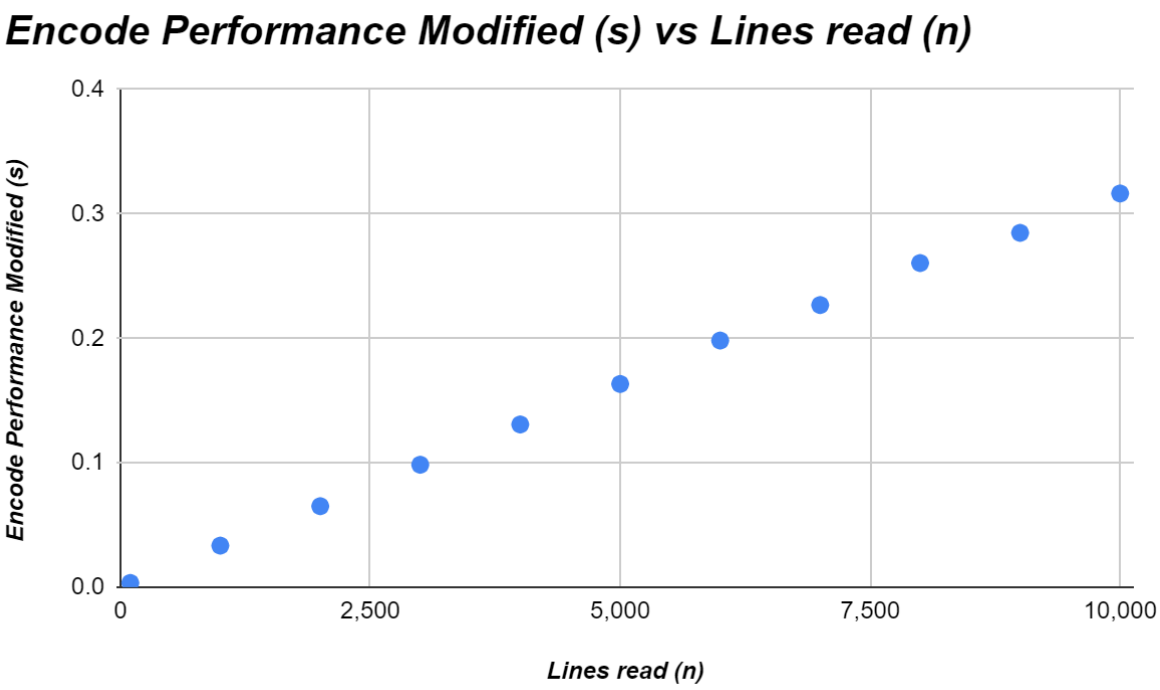

Figure 4

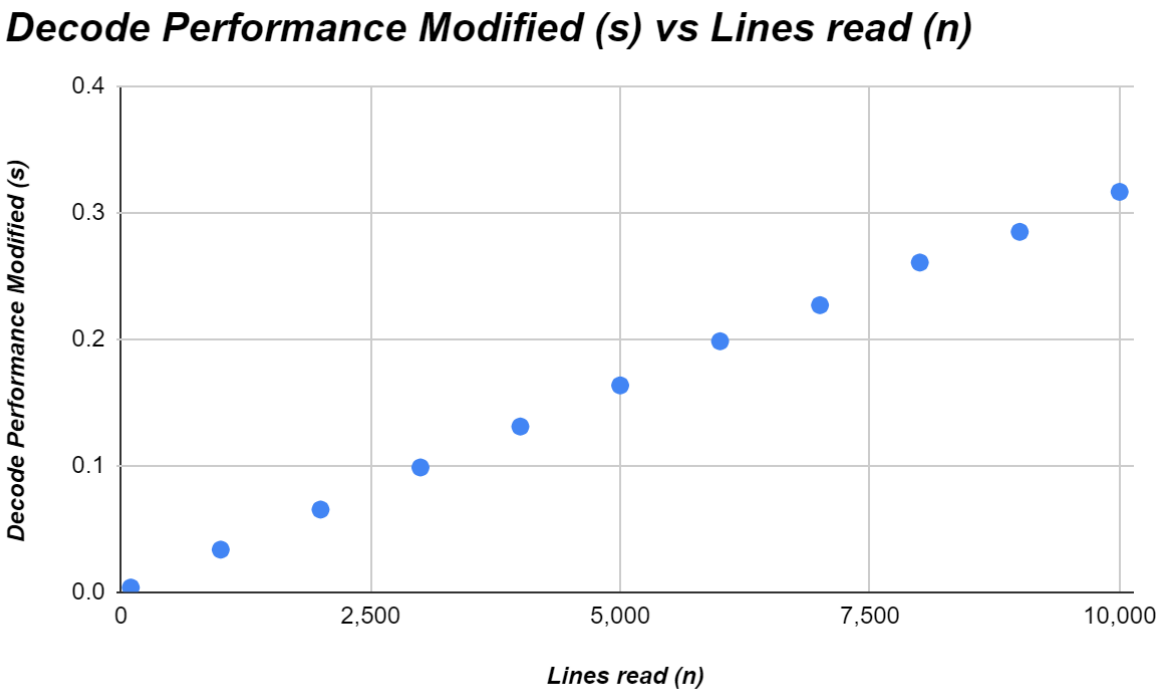

Figure 5