CS 301
High-Performance Computing

Lab 2 - CPU architecture, Triad, Performance

Rakshit Pandhi (202201426)
Kalp Shah (202201457)

February 12, 2025

Contents

1 Introduction

The performance of a code is characterized by key metrics such as floating-point operation rates (FLOPs/sec), memory bandwidth (GB/sec), and energy efficiency (Watts/FLOP). Hardware performance is evaluated using theoretical upper bounds and empirical measurements, which provide insights into achievable performance under realistic conditions. This lab focuses on analyzing these metrics to assess and optimize computational efficiency.

2 Problem A - Understanding your CPU architectures

2.1 Brief description of the problem

The goal of this exercise is to explore and understand the hardware architecture in a given Linux Computer.
1. Open the Terminal
2. Type the command lscpu
3. Several important parameters will be listed.

2.2 Hardware Details for LAB207 PCs

- Architecture: x86_64
- CPU op-mode(s): 32-bit, 64-bit
- Byte Order: Little Endian
- CPU(s): 12
- On-line CPU(s) list: 0-3
- Thread(s) per core: 2
- Core(s) per socket: 6
- Socket(s): 1
- NUMA node(s): 1
- Vendor ID: GenuineIntel
- CPU family: 6
- Model: 60
- Model name: Intel Core i5-12500
- Stepping: 3
- CPU MHz: 799.992
- CPU max MHz: 4.6G

- CPU min MHz: 800.0000
- BogoMIPS: 6584.55
- Virtualization: VT-x
- L1d cache: 288K
- L1i cache: 192K
- L2 cache: 7.5M
- L3 cache: 18M
- NUMA node0 CPU(s): 0-3

2.3 Hardware Details for HPC Cluster

- Architecture: x86_64
- CPU op-mode(s): 32-bit, 64-bit
- Byte Order: Little Endian
- CPU(s): 24
- On-line CPU(s) list: 0-23
- Thread(s) per core: 2
- Core(s) per socket: 6
- Socket(s): 2
- NUMA node(s): 2
- Vendor ID: GenuineIntel
- CPU family: 6
- Model: 63
- Model name: Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz
- Stepping: 2
- CPU MHz: 1419.75
- BogoMIPS: 4804.69
- Virtualization: VT-x
- L1d cache: 32K
- L1i cache: 32K

- L2 cache: 256K
- L3 cache: 15360K
- NUMA node0 CPU(s): 0-5,12-17
- NUMA node1 CPU(s): 6-11,18-23

3 Benchmarking

3.1 Brief description of the problem

The goal of this exercise is to explore the impact of the memory hierarchy on the computation speed of a CPU. For this, we will use the Stream Benchmark (copy, scale, sum, triad)

3.2 Vector Triad

The main logic is a[i]=b[i]+c[i]*d[i].

- In this, there are 4 total memory accesses, as there are 4 arrays, and 2 total FLOPs, one addition and another multiplication.

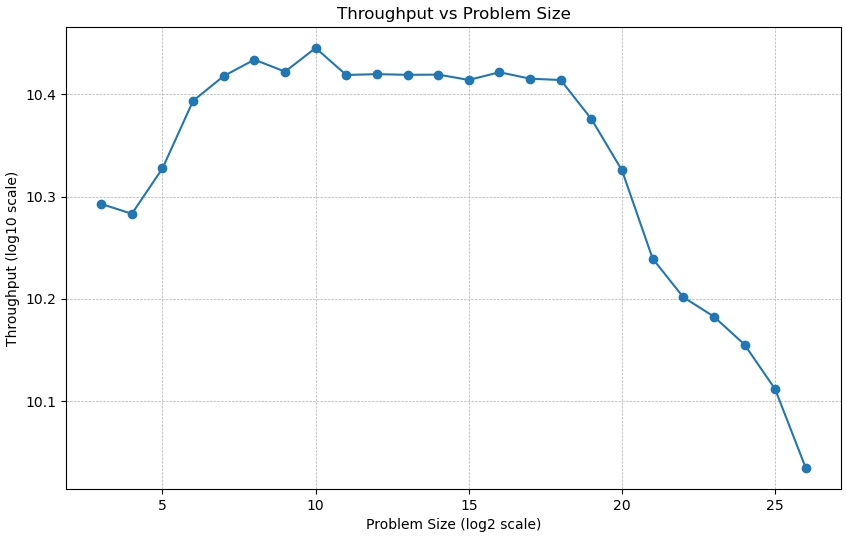

Figure 1: Screenshot from LAB207 PC

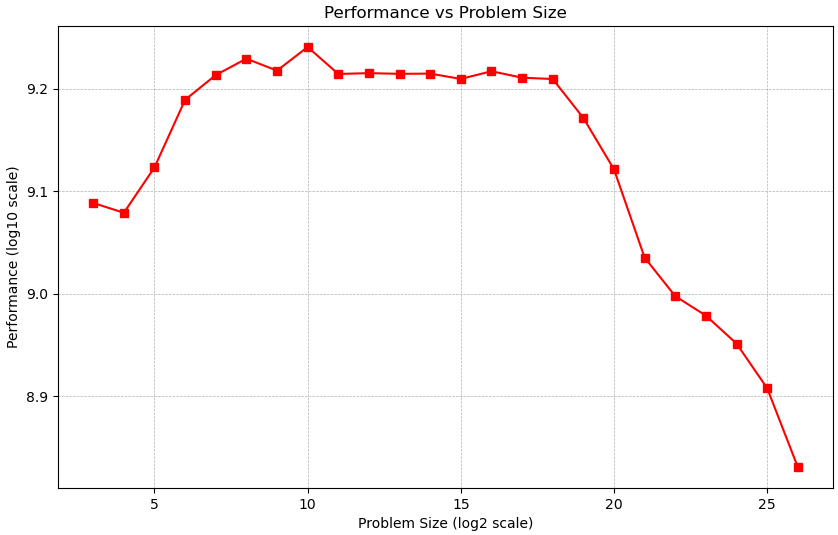

Figure 2: Screenshot from LAB207 PC

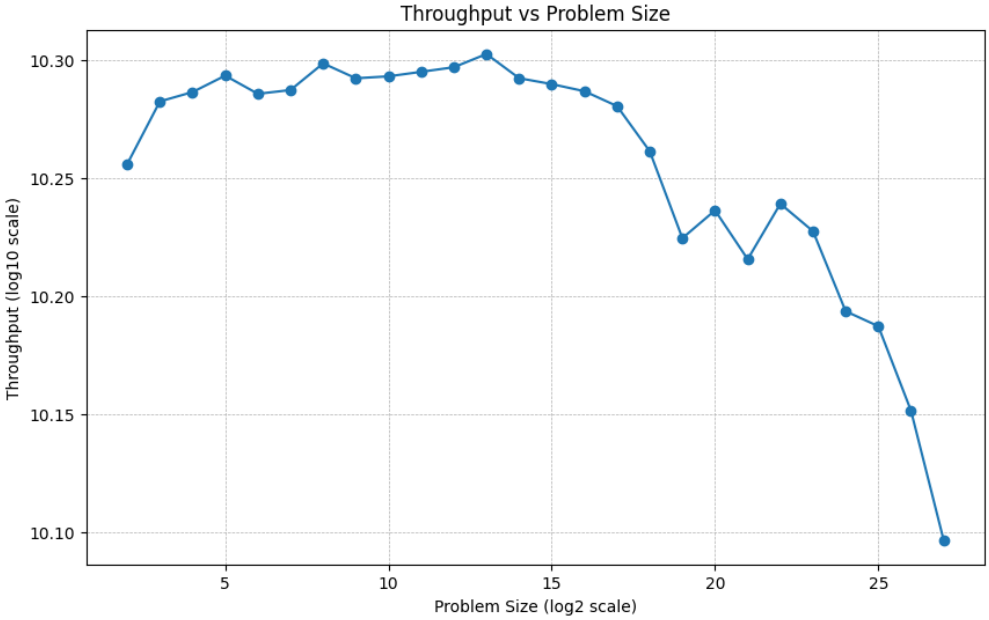

Figure 3: Screenshot from HPC Cluster

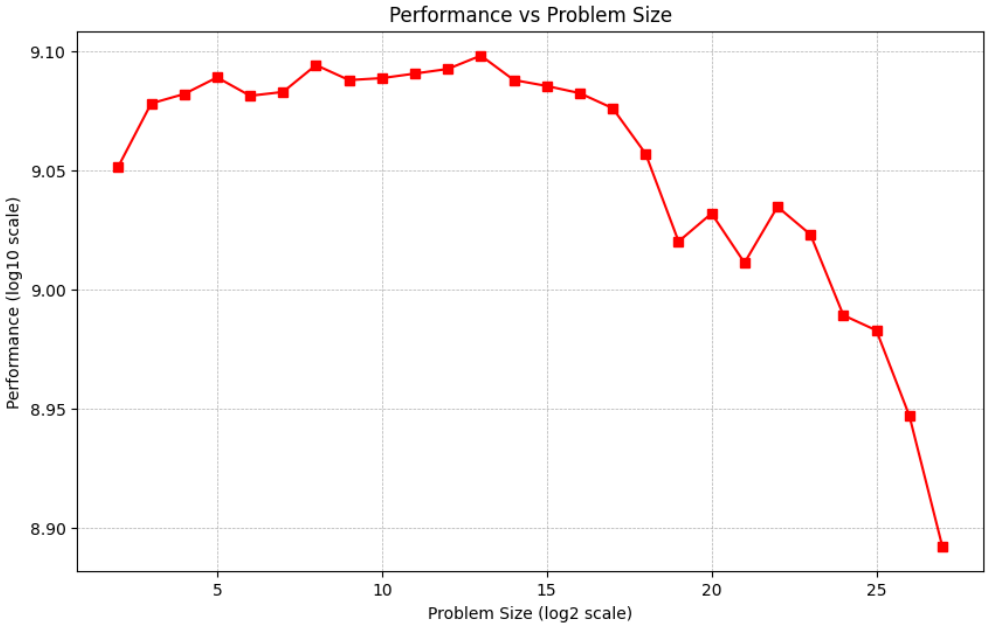

Figure 4: Screenshot from HPC Cluster

3.3 Sum

The main logic is a[i]=b[i]+c[i].

- In this, there are 3 total memory accesses, as there are 3 arrays, and 1 FLOP for addition.

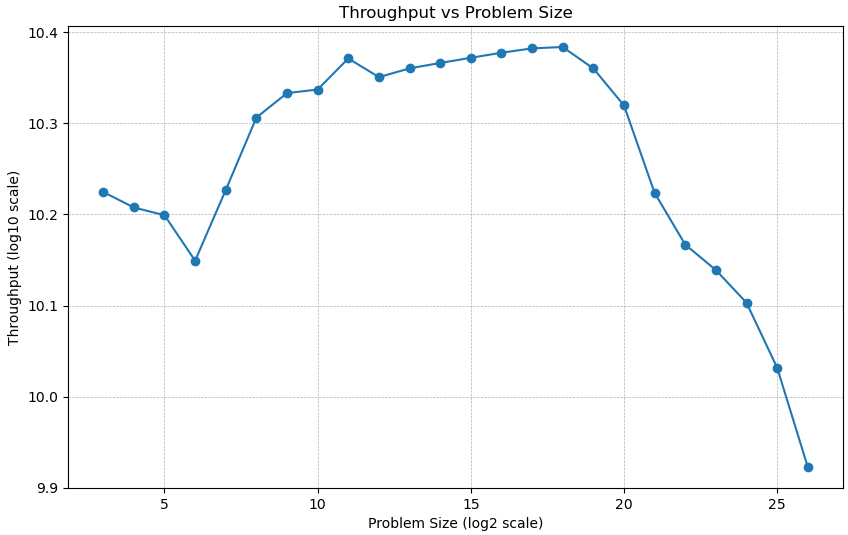

Figure 5: Screenshot from LAB207 PC

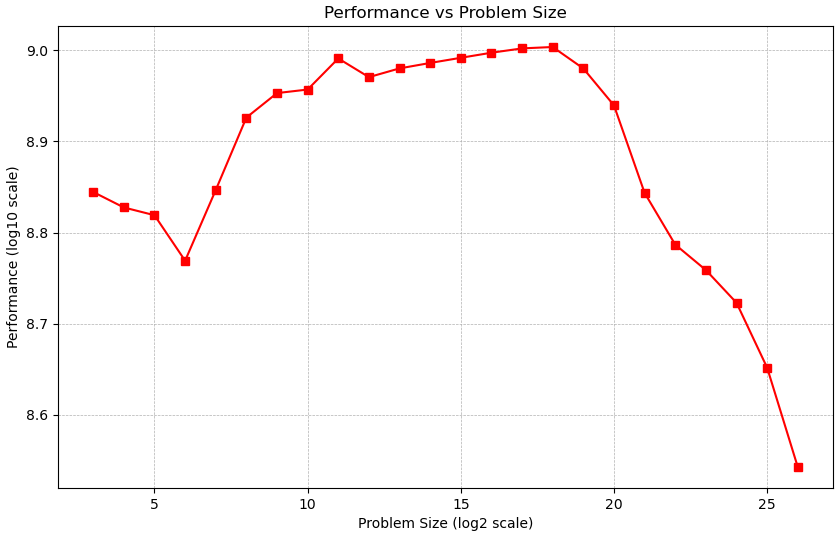

Figure 6: Screenshot from LAB207 PC

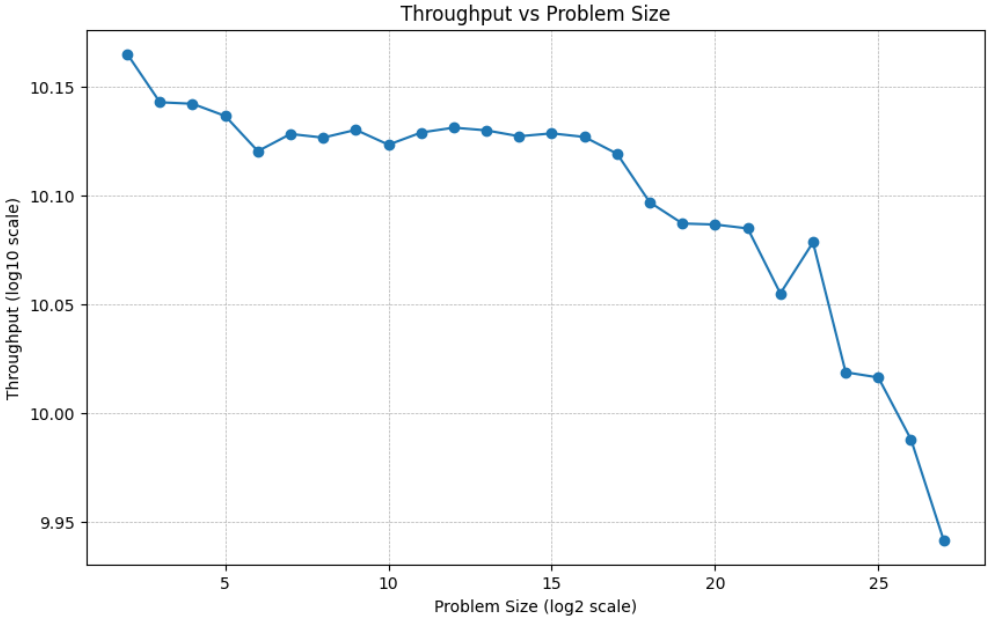

Figure 7: Screenshot from HPC Cluster

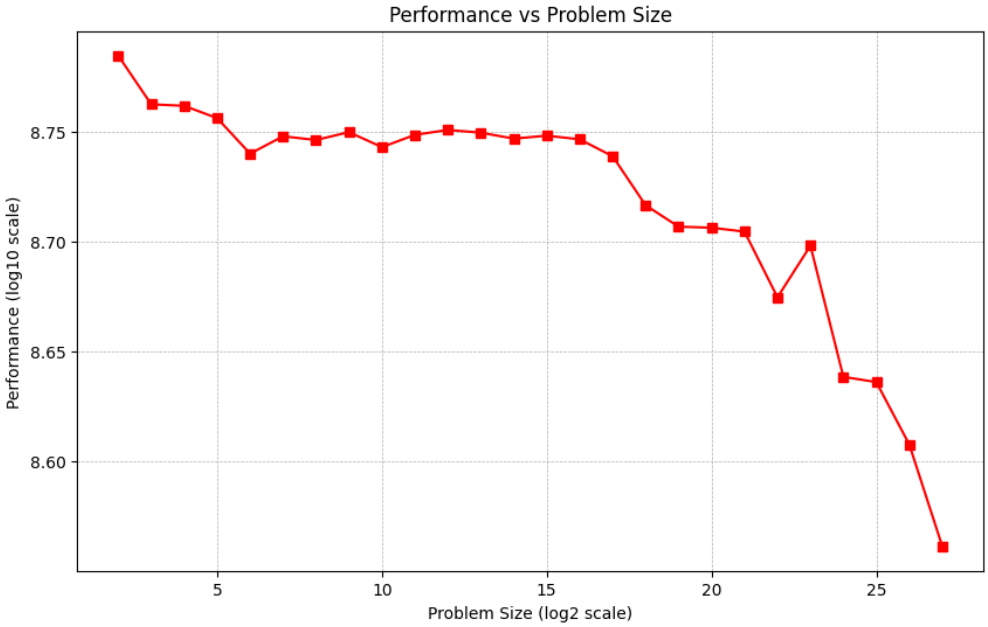

Figure 8: Screenshot from HPC Cluster

3.4 Scale

The main logic in the code is a[i]=5*c[i].

- In this, there are 2 total memory accesses, as there are 2 arrays, and 1 FLOP for multiplication.

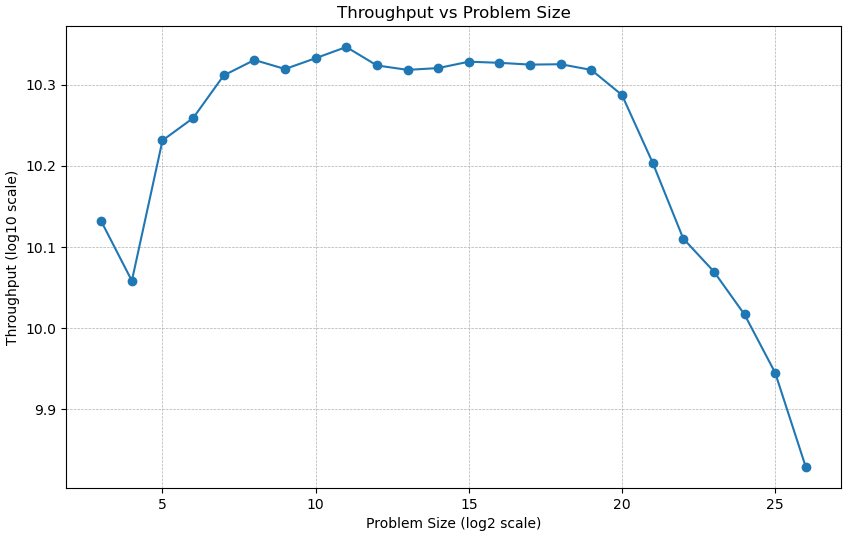

Figure 9: Screenshot from LAB207 PC

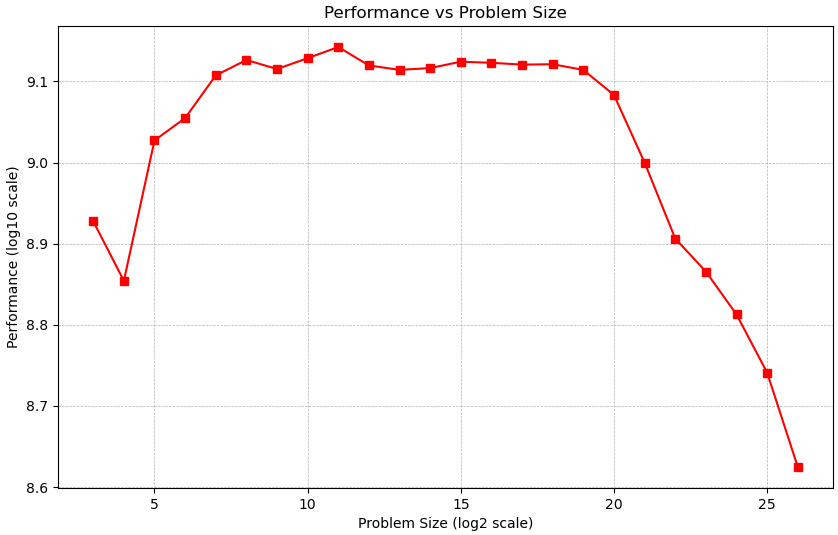

Figure 10: Screenshot from LAB207 PC

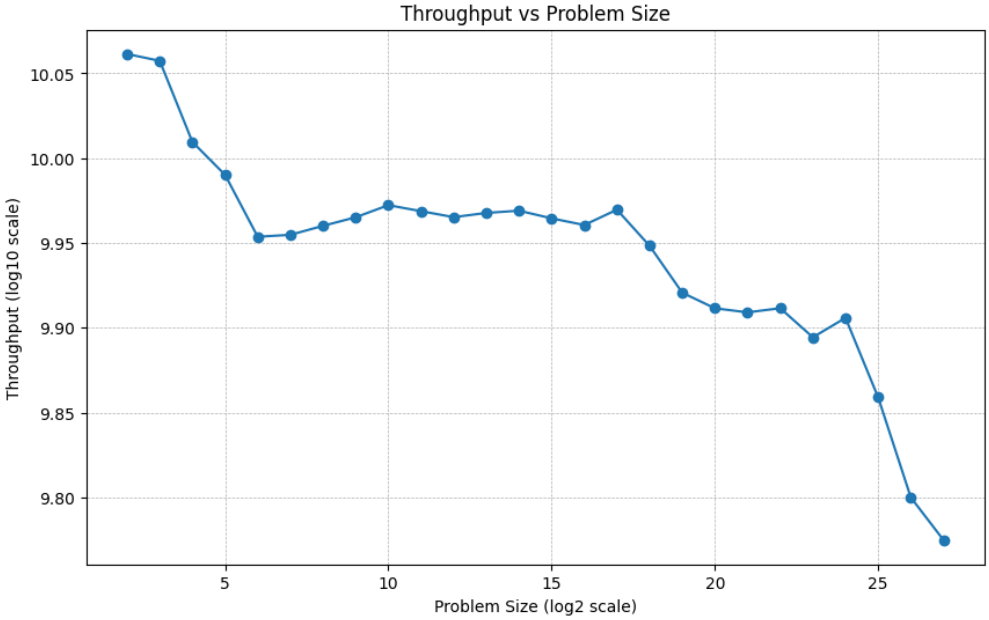

Figure 11: Screenshot from HPC Cluster

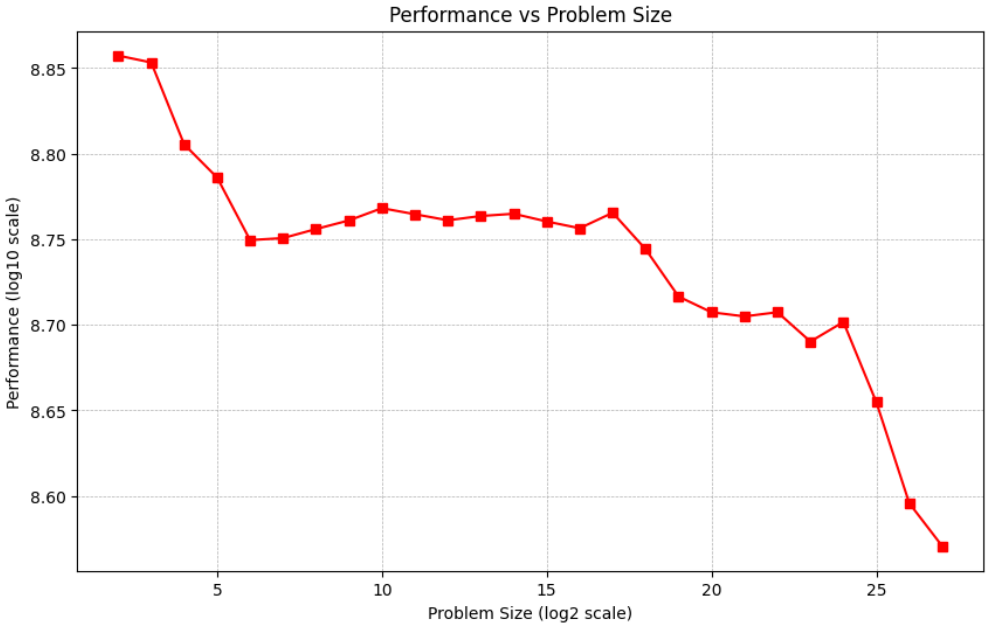

Figure 12: Screenshot from HPC Cluster

3.5 Copy

The main logic in the code is a[i]=b[i].

- In this, there are 2 total memory accesses, as there are 2 arrays, and 0 FLOP.

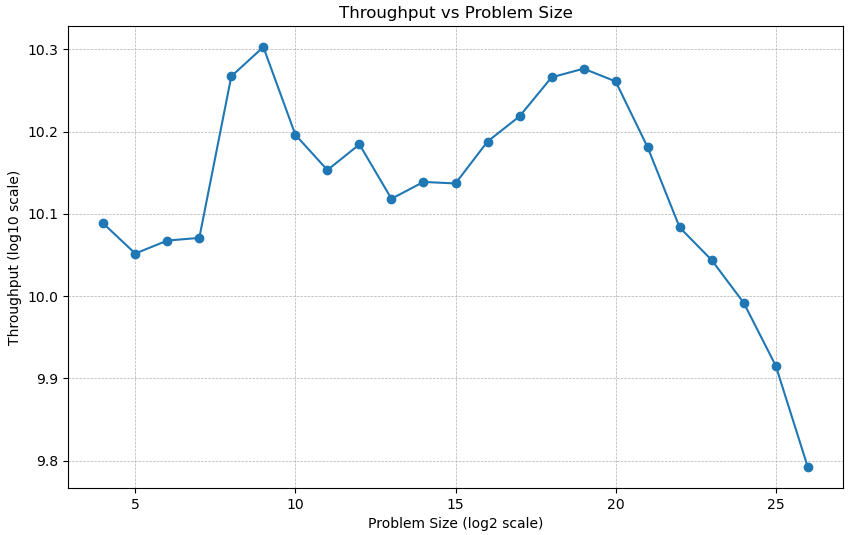

Figure 13: Screenshot from LAB207 PC

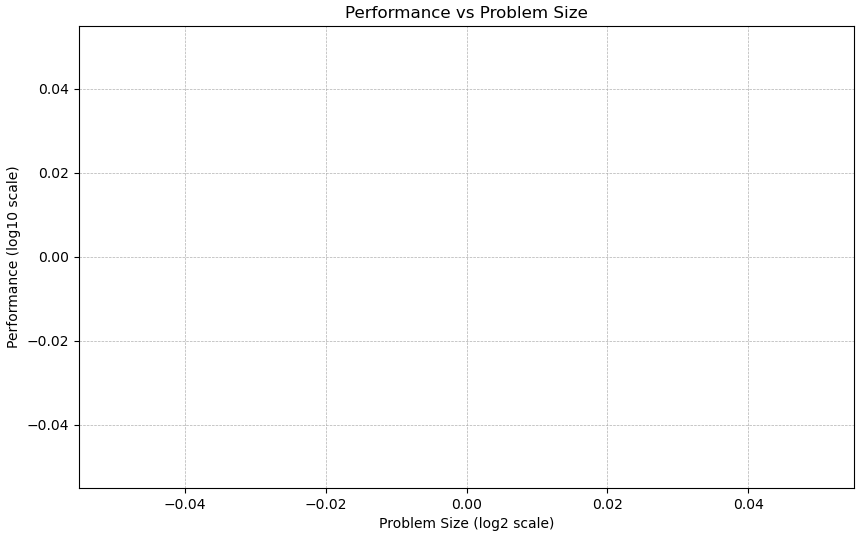

Figure 14: Screenshot from LAB207 PC

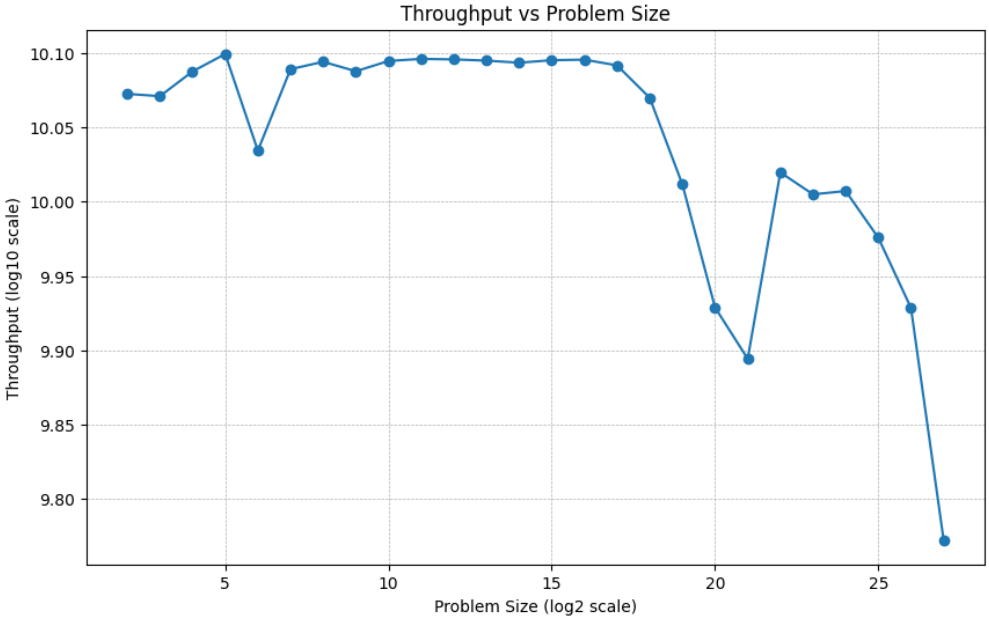

Figure 15: Screenshot from HPC Cluster

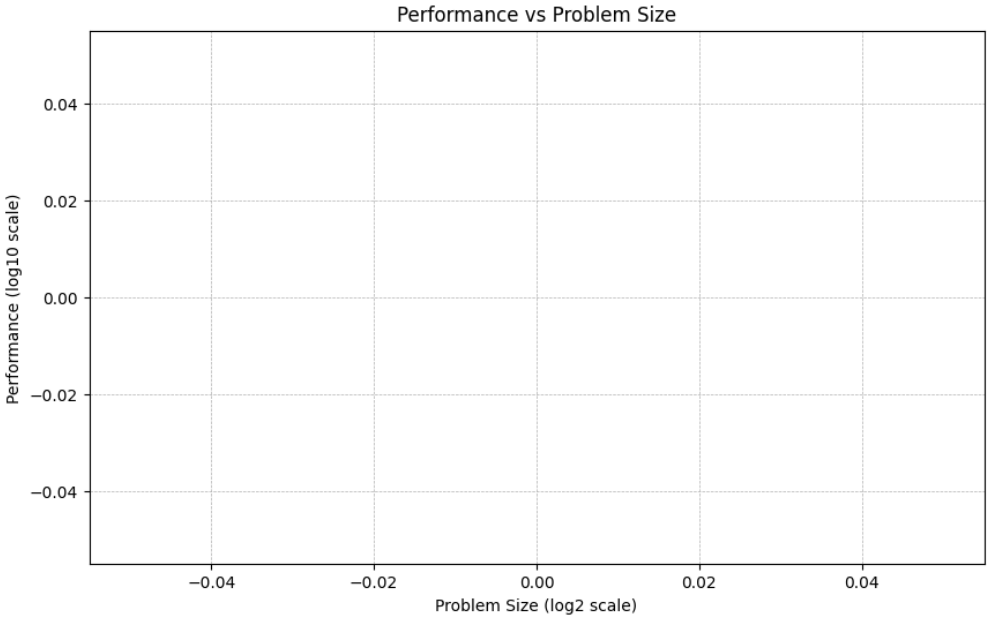

Figure 16: Screenshot from HPC Cluster

- In this code as there is no floating point operation being carried, hence the performance is ZERO therefore the plot.

4 Conclusions

- In this lab we compared the throughput and performance of a system using Low-level or Microbenchmarking technique.
- From the plots which we got we can see some sort of trend to be followed, which is that the L1,L2,L3 cache sizes for the LAB PCs is much higher than the HPC Cluster. Hence we can see that in the plots of the LAB ones the performance and throughput are increasing with some dips in between because at that problem size a particular cache got full and because of that a miss would take place,increasing the fetch cycle and reducing the performance.
- On the other hand the cache sizes for the HPC Cluster is very less compared to the LAB PC. L1 cache for the cluster is 32KB whereas it is 288KB. So for the plots for HPC cluster we are generally seeing a decreasing trend along with the dips for the same reason as above.
- Also for large problem sizes both the LAB PCs and HPC Cluster performs badly and the plot is very sharply decreasing at the ends.