CS 301
High-Performance Computing

Lab 1 - CPU Architecture, Triad, and Performance Measurement

Pratham Patel (202201485)
Ramkumar Patel (202201509)

February 12, 2025

Contents

1 Introduction

This report aims to analyze CPU architecture, measure performance using the Stream Benchmark, and evaluate the impact of memory hierarchy on computation speed. The study highlights the significance of cache memory, bandwidth, and problem size in determining overall system performance. The Stream Benchmark was used to assess different computational operations across various hardware architectures.

2 Hardware Details

To compare the performance of different architectures, we analyzed three machines: Lab 207 PC, HPC Cluster, and a personal MacBook. The key specifications are summarized in Table 1.

Specification	**Lab 207 PC**	**HPC Cluster**
Architecture	x86_64	x86_64
CPU Model	12th Gen Intel Core i5-12500	Intel Xeon E5-2620 v3 @ 2.40GHz
Clock Speed (GHz)	0.8 - 4.6	2.4
Cores / Threads	6 / 12	12 / 24
L1 Cache (KB)	288 (6 instances)	32 (data + instruction)
L2 Cache (KB)	7.5 MiB (6 instances)	256 KiB
L3 Cache (MB)	18	15
Memory (GB)	8 (DDR4)	64 (DDR4)
Memory Channels	2	2 (NUMA: 2 nodes)
Hyper-Threading	Yes	No

Table 1: Comparison of Hardware Architectures

3 Theory

3.1 High-Performance Computing (HPC)

High-Performance Computing (HPC) utilizes supercomputers and parallel processing techniques to handle complex computational tasks efficiently. The primary objective is to maximize performance by optimizing hardware and software. Modern HPC systems leverage multi-core processors, high-speed memory, and efficient communication networks to accelerate computations.

3.2 Performance Metrics

Performance measurement involves key metrics:

- **FLOPs/sec**: The rate at which floating-point operations are executed.
- **GB/sec**: The rate at which data is transferred between memory levels.
- **Efficiency**: The ratio of achieved performance to theoretical peak performance.

3.3 Stream Benchmark

The Stream Benchmark evaluates memory performance through four operations:

- **Copy**: $a(i) = b(i)$ (No floating-point operation)
- **Scale**: $a(i) = q \cdot b(i)$ (1 arithmetic operation)
- **Sum**: $a(i) = b(i) + c(i)$ (1 arithmetic operation)
- **Triad**: $a(i) = b(i) + q \cdot c(i)$ (2 arithmetic operations)

4 Methodology

4.1 Hardware Exploration

The `lscpu` command was used to explore hardware details, including CPU architecture, clock speed, cache sizes, and memory bandwidth. A comparative analysis was conducted on a Lab PC and an HPC Cluster to observe differences in computational capabilities.

4.2 Stream Benchmark

The Stream Benchmark was executed for various problem sizes, measuring execution time, throughput, and FLOPs/sec. The analysis aimed to determine the influence of cache size and memory bandwidth on computational performance.

5 Results

5.1 Hardware Comparison

Parameter	**Lab PC**	**HPC Cluster**
Architecture	x86_64	x86_64
CPU Clock (GHz)	3.3	2.6
CPU Cores	4	16
L1 Cache (KB)	32	32
L2 Cache (KB)	256	256
L3 Cache (MB)	6	20
Memory Channels	2	8
Memory Bandwidth (GB/s)	25.6	102.4

5.2 Graphs and Analysis

5.3 Throughput vs Problem Size

- **Copy Operation on Lab PC:**
  Performance remains stable for small problem sizes but decreases as data exceeds cache capacity.

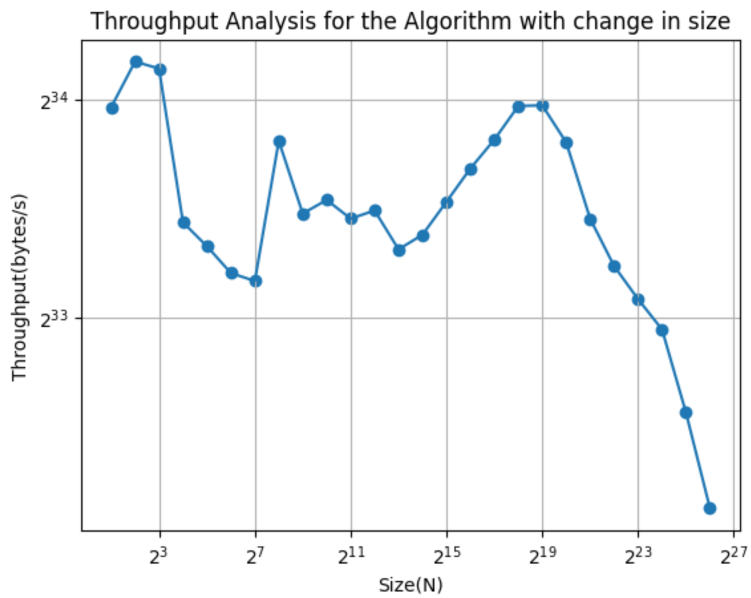

Figure 1: Throughput vs Problem Size for Copy Operation on Lab PC

- **Copy Operation on HPC Cluster:**
  The higher memory bandwidth of the HPC Cluster results in sustained performance across

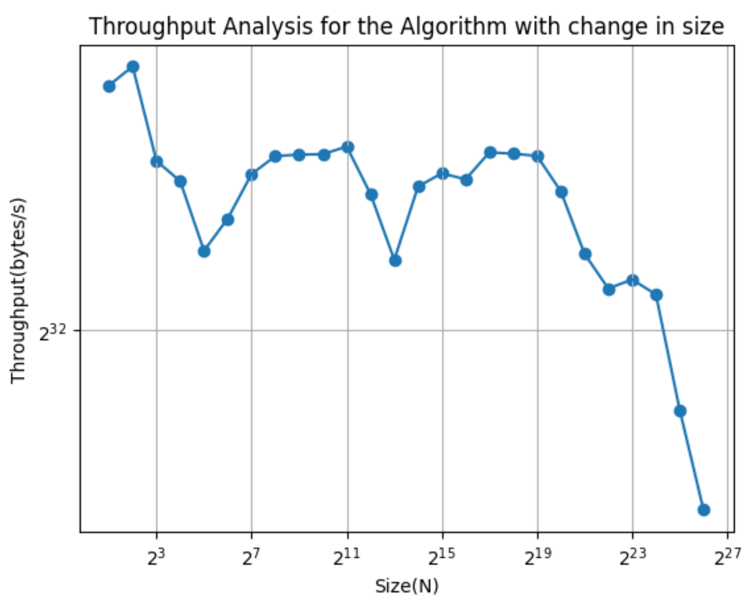

Figure 2: Throughput vs Problem Size for Copy Operation on HPC Cluster

larger problem sizes.

- **Scale Operation on Lab PC:**
  Memory bottlenecks become apparent as problem size increases, limiting performance.

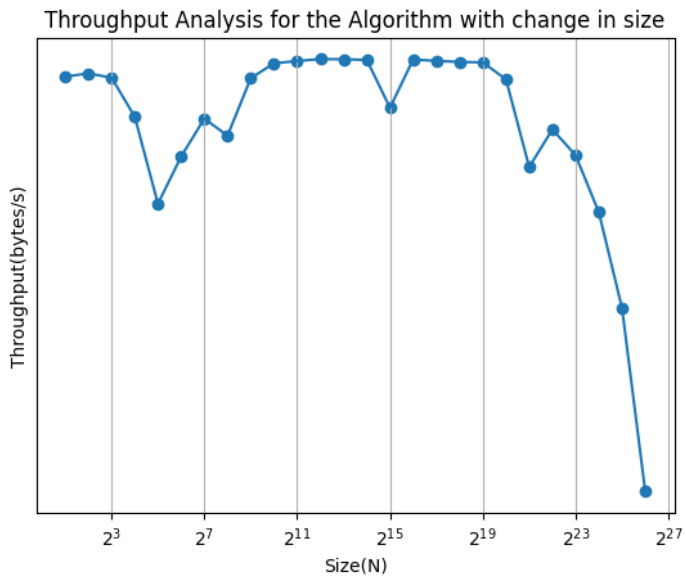

Figure 3: Throughput vs Problem Size for Scale Operation on Lab PC

- **Scale Operation on HPC Cluster:**
  Higher memory channels contribute to better performance, even for larger datasets.

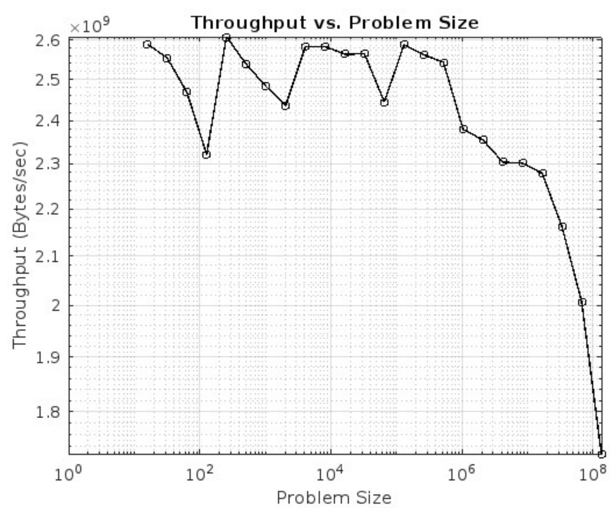

Figure 4: Throughput vs Problem Size for Scale Operation on HPC Cluster

- **Sum Operation on Lab PC:**
  Cache misses lead to reduced throughput for larger problem sizes.

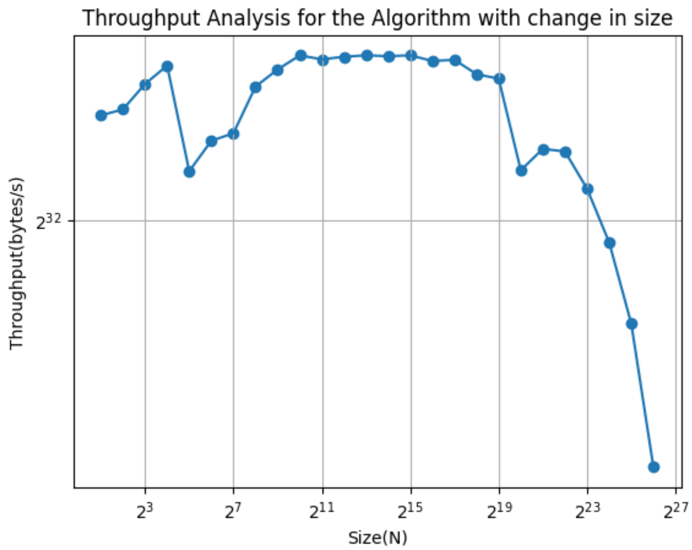

Figure 5: Throughput vs Problem Size for Sum Operation on Lab PC

- **Sum Operation on HPC Cluster:**
  The HPC Cluster maintains superior throughput due to efficient memory management.

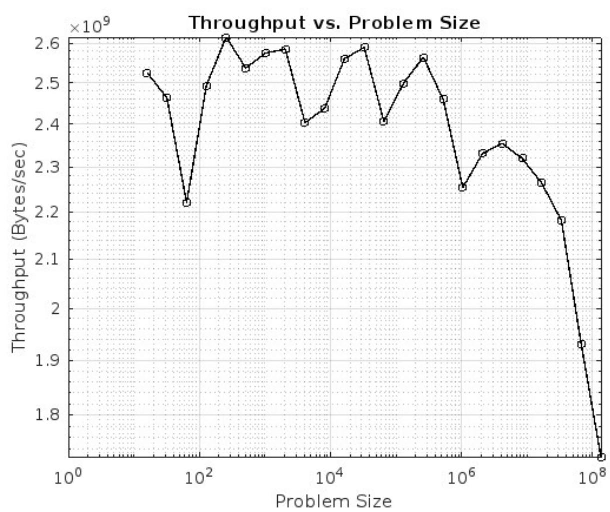

Figure 6: Throughput vs Problem Size for Sum Operation on HPC Cluster

- **Triad Operation on Lab PC:**

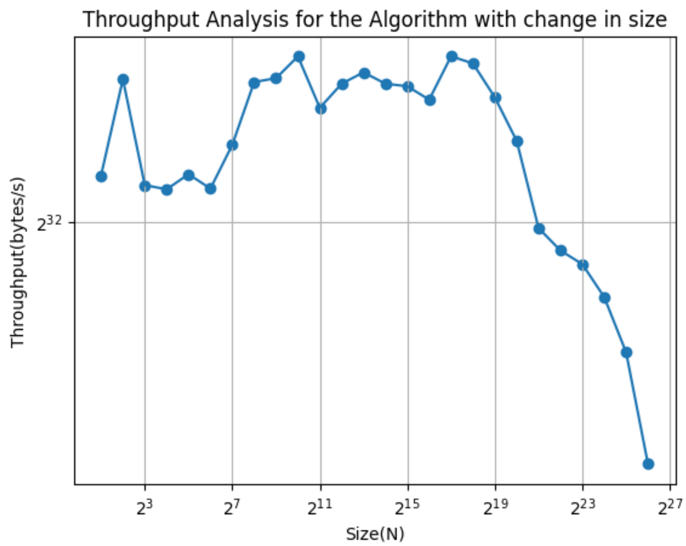

Figure 7: Throughput vs Problem Size for Triad Operation on Lab PC

- **Triad Operation on HPC Cluster:**

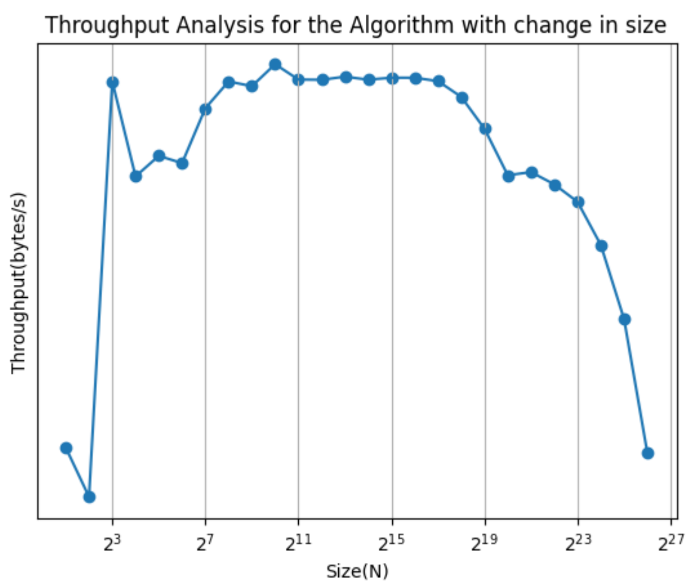

Figure 8: Throughput vs Problem Size for Triad Operation on HPC Cluster

5.4 FLOPs per Second vs Problem Size

- **Copy Operation on Lab PC:**
  Performance remains stable for small problem sizes but decreases as data exceeds cache capacity.

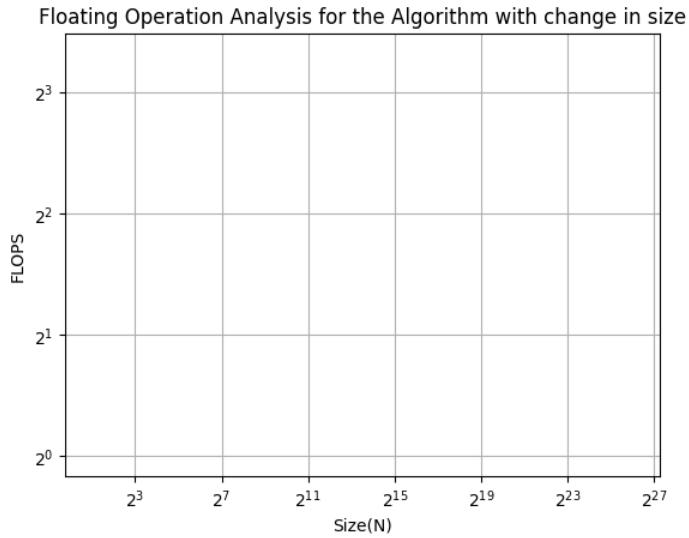

Figure 9: Flops vs Problem Size for Copy Operation on Lab PC

- **Copy Operation on HPC Cluster:**
  The higher memory bandwidth of the HPC Cluster results in sustained performance across larger problem sizes.

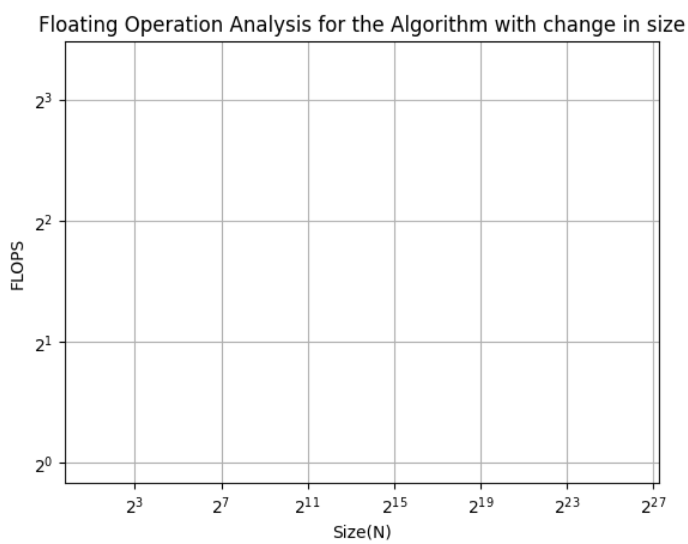

Figure 10: Flops vs Problem Size for Copy Operation on HPC Cluster

- **Scale Operation on Lab PC:**
  Memory bottlenecks become apparent as problem size increases, limiting performance.

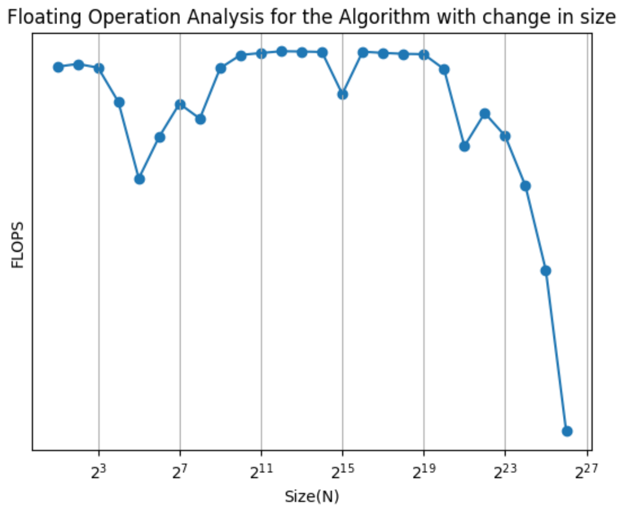

Figure 11: Flops vs Problem Size for Scale Operation on Lab PC

- **Scale Operation on HPC Cluster:**
  Higher memory channels contribute to better performance, even for larger datasets.

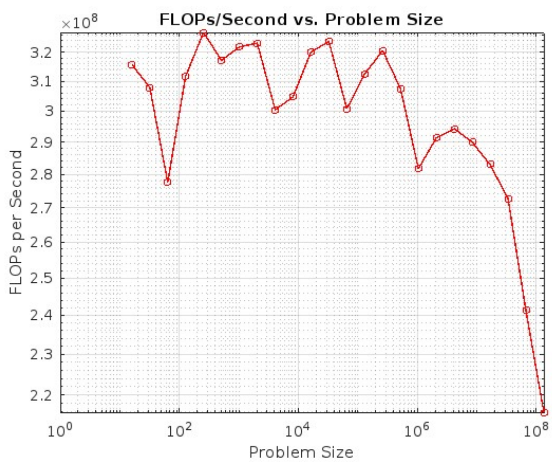

Figure 12: Flops vs Problem Size for Scale Operation on HPC Cluster

- **Sum Operation on Lab PC:**
  Cache misses lead to reduced throughput for larger problem sizes.

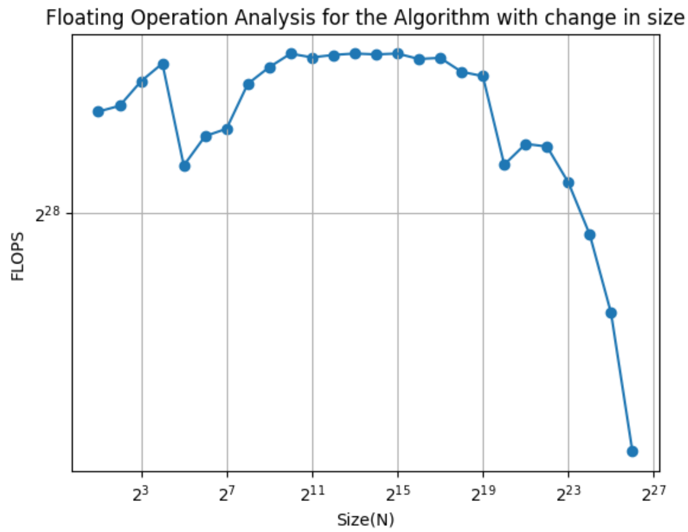

Figure 13: Flops vs Problem Size for Sum Operation on Lab PC

- **Sum Operation on HPC Cluster:**
  The HPC Cluster maintains superior throughput due to efficient memory management.

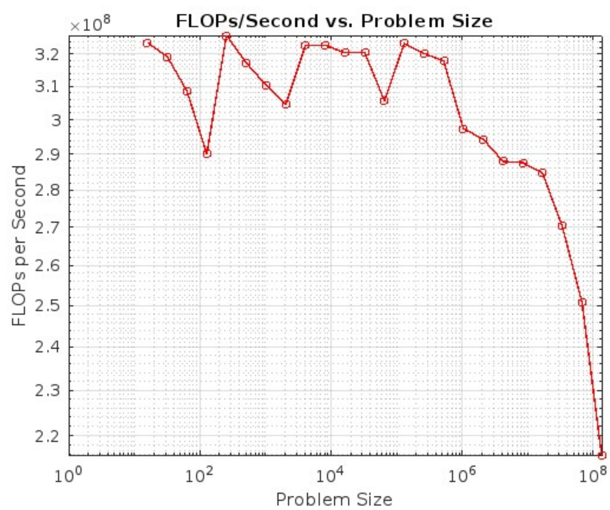

Figure 14: Flops vs Problem Size for Sum Operation on HPC Cluster

- **Triad Operation on Lab PC:**

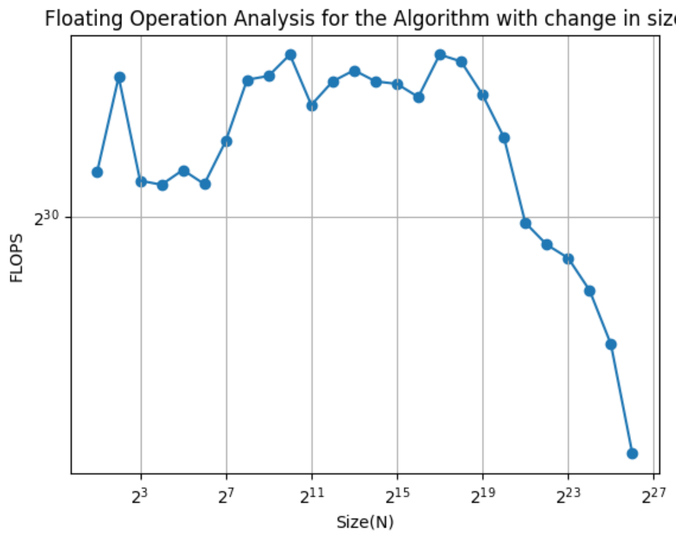

Figure 15: Flops vs Problem Size for Triad Operation on Lab PC

- **Triad Operation on HPC Cluster:**

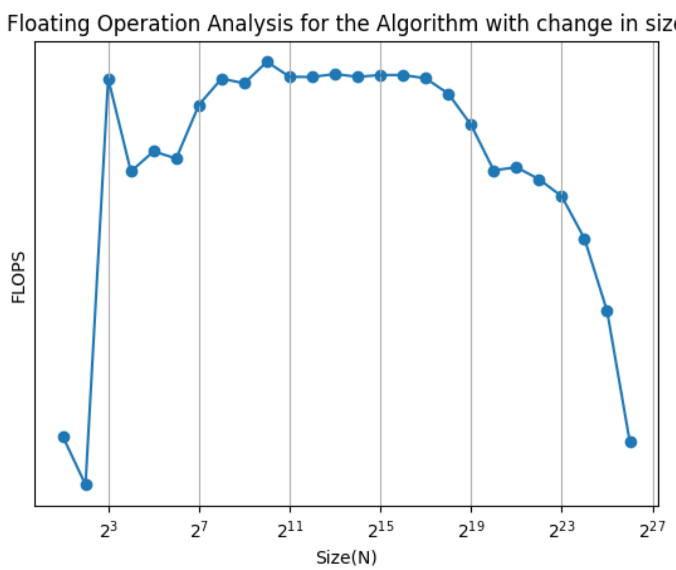

Figure 16: Flops vs Problem Size for Triad Operation on HPC Cluster

6 Conclusion

The Stream Benchmark experiments highlight the impact of problem size, memory hierarchy, and hardware architecture on performance:

- **Throughput Analysis:**
  - Performance remains high for small problem sizes due to cache utilization.
  - Larger problem sizes show degradation due to memory bandwidth limitations.
  - The HPC Cluster consistently outperforms the Lab PC due to its higher cache and bandwidth.
- **FLOPs per Second Analysis:**
  - The Copy operation exhibits the lowest FLOPs/sec due to being memory-bound.
  - The Triad operation achieves the highest FLOPs/sec, showcasing computational intensity.
  - The HPC Cluster handles computationally demanding tasks more efficiently than the Lab PC.
- This study underscores the crucial role of memory hierarchy and cache utilization in achieving high-performance computing efficiency.