CS 301
High-Performance Computing

Lab1 - CPU Architecture, Memory Hierarchy and Performance Measurement

Bhensdadia Vasu (202301416)
Darshit Ramani (202301404)

February 3, 2026

Contents

1 Introduction

In this lab, we will measure the CPU performance of both our Lab PC and the Cluster. We will also study the system architecture by implementing the vector triad code, copy, scale, and sum. These operations were discussed in our theory class, and now we will test them in practice to understand their impact on performance.

2 Hardware Details

2.1 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: 0-11
- Thread(s) per core: 2
- Core(s) per socket: 6
- Socket(s): 1
- NUMA node(s): 1
- Vendor ID: GenuineIntel
- CPU family: 6
- Model: 151
- Model name: 12th Gen Intel(R) Core(TM) i5-12500
- Stepping: 5
- CPU max MHz: 4600.0000
- CPU min MHz: 800.0000
- BogoMIPS: 5990.40
- Virtualization: VT-x
- L1d cache: 288K
- L1i cache: 192K
- L2 cache: 7.5M

- L3 cache: 18M
- NUMA node0 CPU(s): 0-11
- Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm epb invpcid_single tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid xsaveopt dtherm ida arat pln pts

2.2 Hardware Details for HPC Cluster (Node gics1)

- Architecture: x86_64
- CPU op-mode(s): 32-bit, 64-bit
- Byte Order: Little Endian
- CPU(s): 16
- On-line CPU(s) list: 0-15
- Thread(s) per core: 1
- Core(s) per socket: 8
- Socket(s): 2
- NUMA node(s): 2
- Vendor ID: GenuineIntel
- CPU family: 6
- Model: 63
- Model name: Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
- Stepping: 2
- CPU MHz: 1288.726
- BogoMIPS: 5204.73
- Virtualization: VT-x
- L1d cache: 32K
- L1i cache: 32K
- L2 cache: 256K

- L3 cache: 20480K
- NUMA node0 CPU(s): 0-7
- NUMA node1 CPU(s): 8-15

3 Problem Description

In this lab, we aim to measure the CPU performance of both the Lab PC and the Cluster by running the Stream Benchmark. This benchmark evaluates memory bandwidth using four operations: Copy, Scale, Sum, and Triad. These operations involve reading and writing large arrays, and their performance is limited by memory bandwidth rather than computational power.

To analyze performance, we measured throughput (GB/s) vs. problem size and FLOPs vs. problem size for each operation. The results were plotted separately for the Lab PC and the Cluster, producing a total of 16 graphs.

4 Benchmarking Methodology

We executed the operations on both the Lab PC and the Cluster, measuring:

- Execution time for varying problem sizes.
- Throughput (GB/s) and FLOPs for each operation.
- Differences in performance between the Lab PC and the Cluster.

For each case, we generated plots of:

- Throughput (Bytes/s) vs. Problem Size

$$Throughput = \frac{sizeof(double) * N * Total}{alg\_time}$$

  where, N is the number of vectors used in algorithm,

- FLOPs vs. Problem Size

$$FLOPs = \frac{M * Total}{alg\_time}$$

  where, M is the number of floating point operations in single operation.

5 Graphical Results

5.1 Vector Copy Operation : $a[i] = b[i]$

5.1.1 Throughput and FLOPs using HPC Cluster

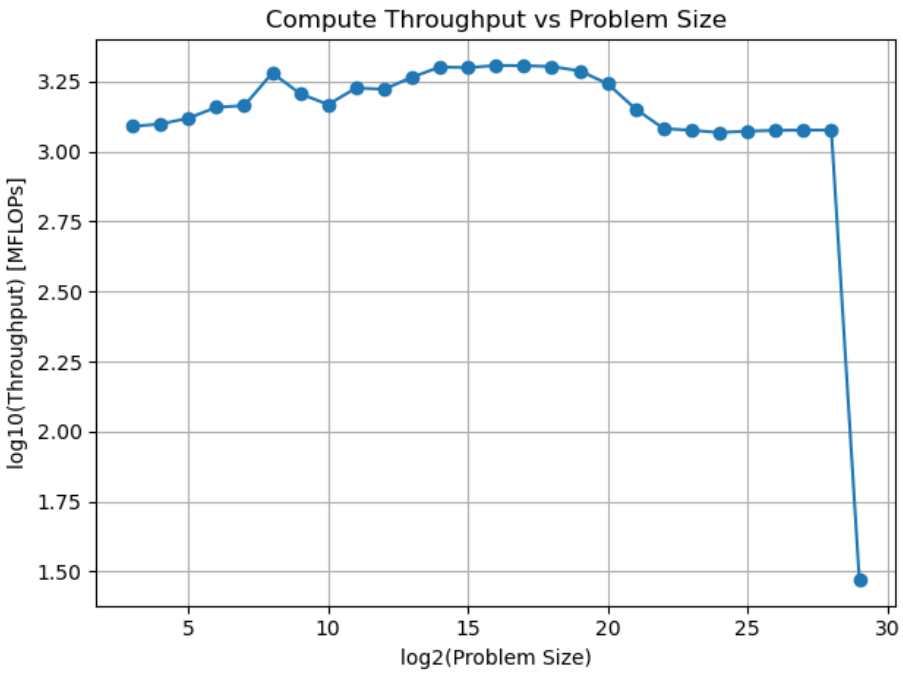

Figure 1: Throughput vs. Problem Size for the vector copy operation using HPC Cluster.

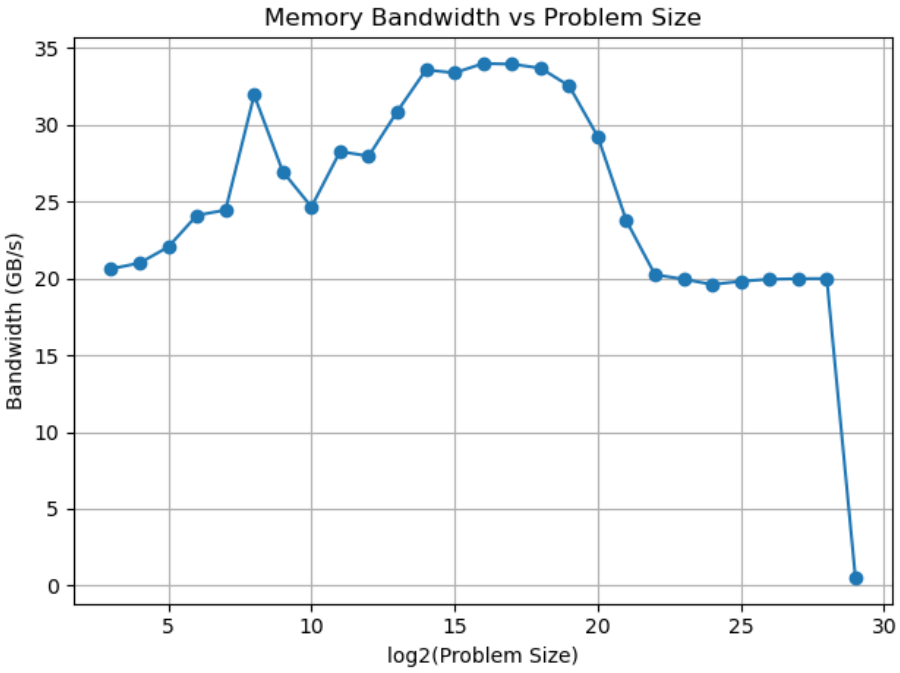

Figure 2: Bandwidth vs. Problem Size for the vector copy operation using HPC Cluster.

5.2 Vector Scaling Operation : $a[i] = k * b[i]$

5.2.1 Throughput and FLOPs using HPC Cluster

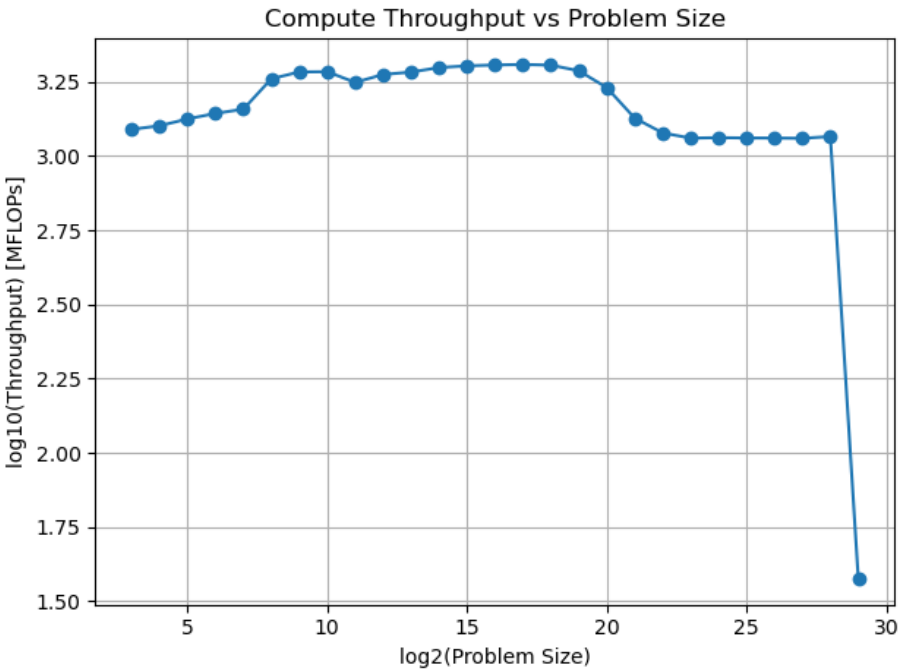

Figure 3: Throughput vs. Problem Size for the vector triad operation using HPC Cluster

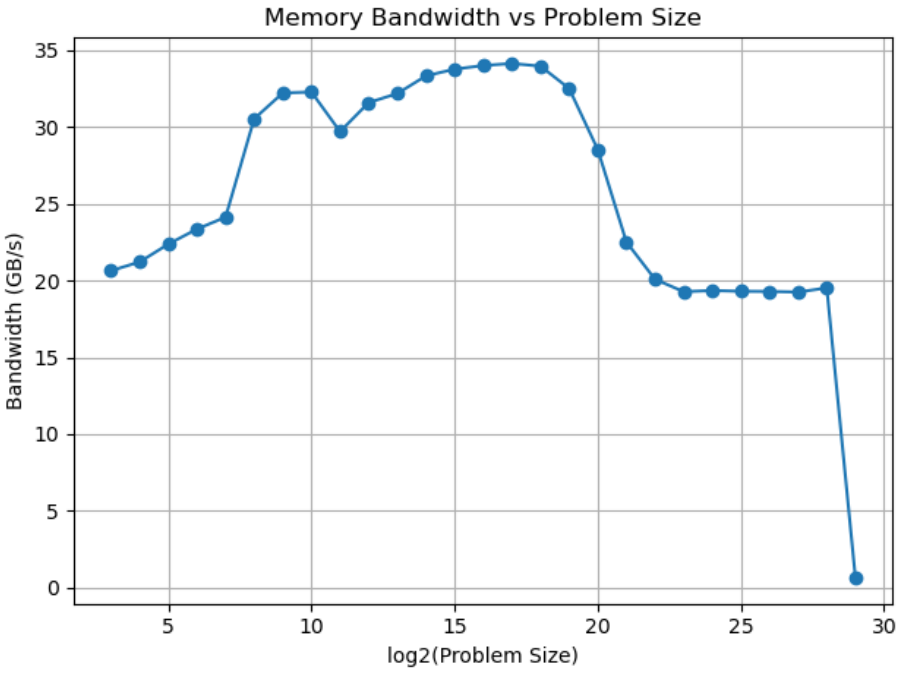

Figure 4: Bandwidth vs. Problem Size for the vector triad operation using HPC Cluster.

5.3 Vector Sum Operation : $a[i] = b[i] + c[i]$

5.3.1 Throughput and FLOPs using HPC Cluster

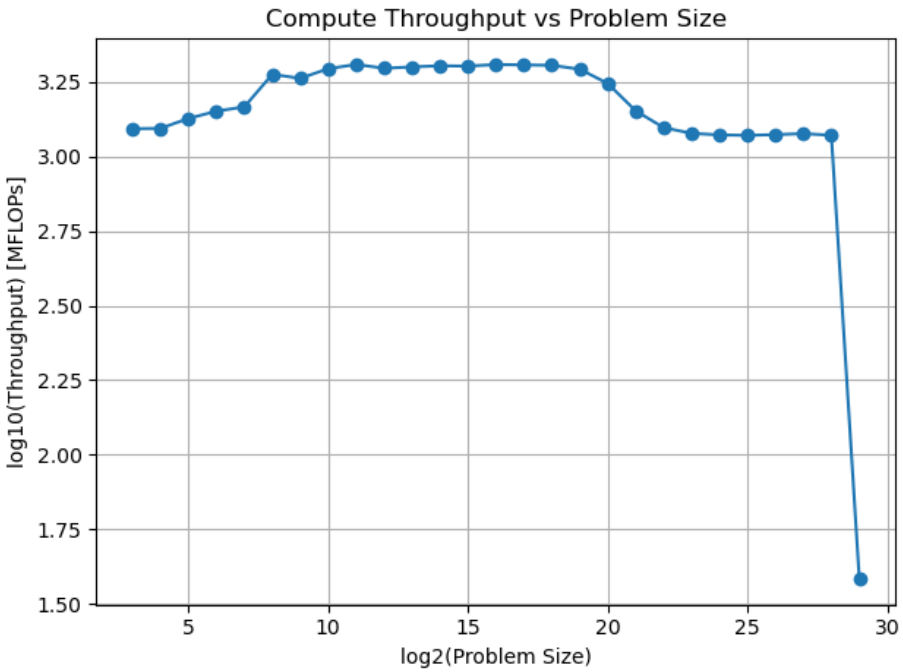

Figure 5: Throughput vs. Problem Size for the vector sum operation using HPC Cluster

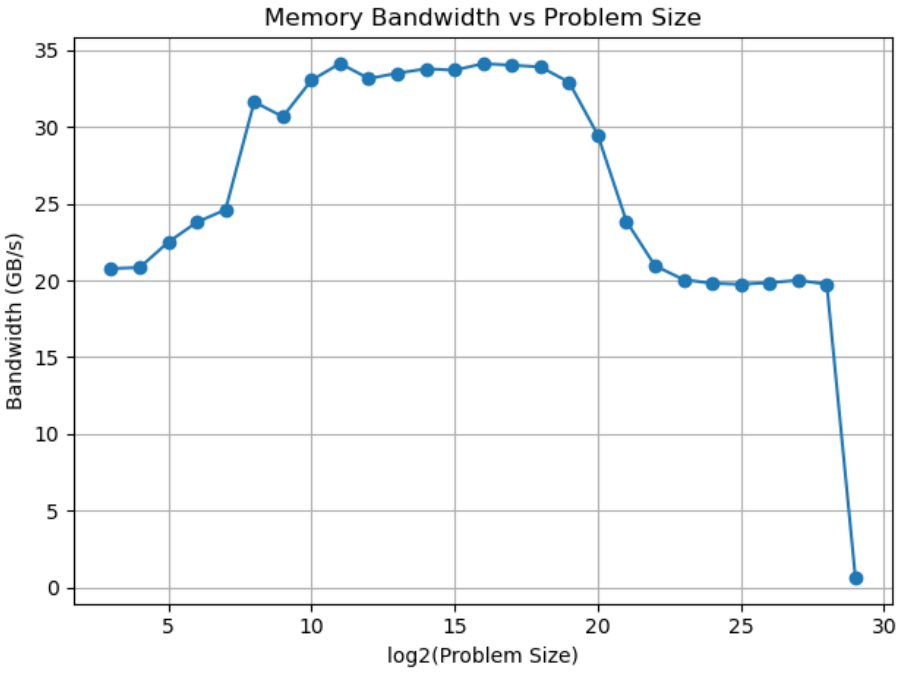

Figure 6: Bandwidth vs. Problem Size for the vector sum operation using HPC Cluster.

5.4 Vector Triad Operation : $a[i] = b[i] + c[i] * d[i]$

5.4.1 Throughput and FLOPs using HPC Cluster

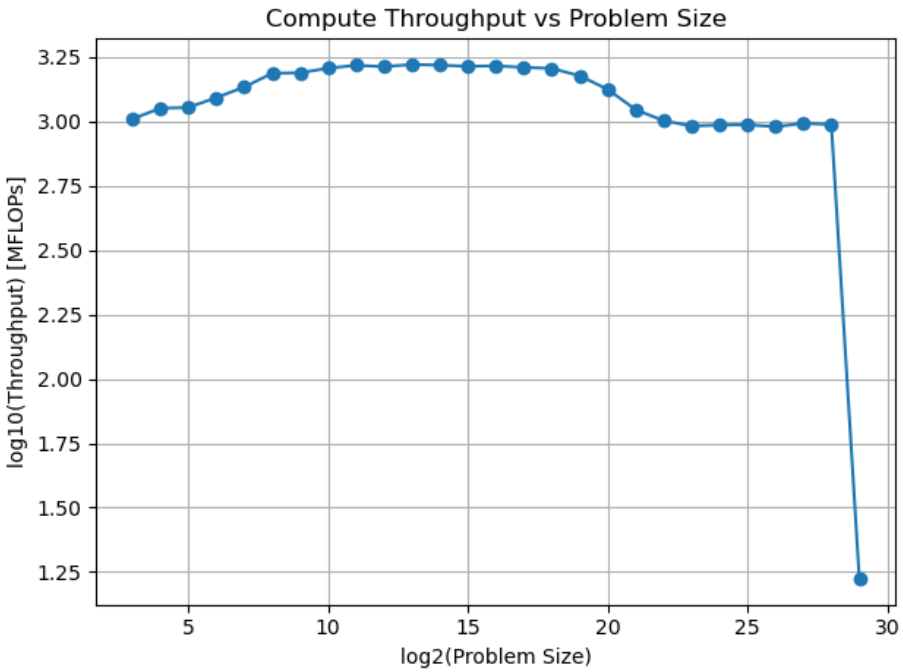

Figure 7: Throughput vs. Problem Size for the vector triad operation using Lab PC.

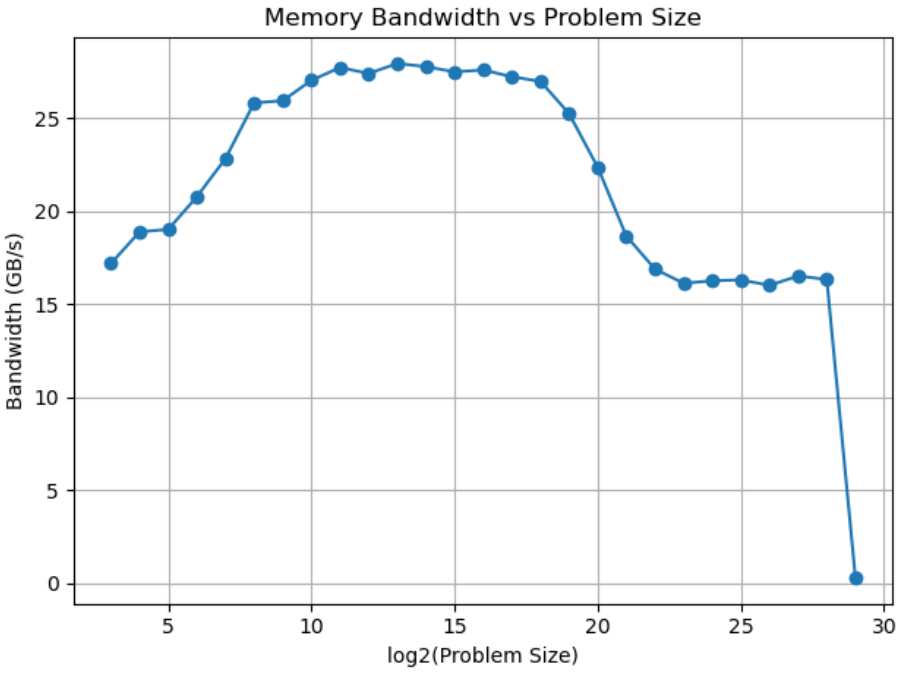

Figure 8: Bandwidth vs. Problem Size for the vector triad operation using Lab PC.

6 Conclusion

We tested all four operations on the Lab PC as well as HPC cluster. Performance was stable for small problem sizes but dropped as the size increased. This drop happens because memory bandwidth becomes a bottleneck. The system performs well in computing, but memory access slows it down. The actual performance is lower than the theoretical peak due to these memory limits. A sudden large drop in performance likely means the problem size exceeded a cache limit, forcing slower memory access.