# CS 301
# High-Performance Computing

## Lab5 - Performance Analysis of Parallel Computing Implementations

Pratham Patel (202201485)
Ramkumar Patel (202201509)

# Contents

**Abstract**

This report presents an analysis of parallel computing performance using different core counts on a cluster and a lab PC. The results show that speedup varies with problem size and number of cores, with an observed decrease in efficiency for small problem sizes due to parallel overhead. The findings provide insight into optimal parallelization strategies.

# 1 Introduction

High-Performance Computing (HPC) leverages parallel processing to solve complex problems efficiently. The speedup achieved by using multiple cores is a crucial factor in determining the effectiveness of parallelization. However, performance is influenced by various factors, including problem size, computational overhead, and hardware configuration. This experiment investigates the speedup characteristics on a computing cluster and a lab PC by analyzing different core counts and problem sizes.

# 2 Methodology

The experiment involves solving a computational problem using different numbers of cores on two systems:

- **Cluster:** Multi-node system with higher processing power.
- **Lab PC:** Single-node system with limited resources.

The execution times for different problem sizes were recorded and used to calculate speedup:

$$\text{Speedup} = \frac{T_{serial}}{T_{parallel}} \tag{1}$$

where $T_{serial}$ is the execution time for a single core and $T_{parallel}$ is the execution time using multiple cores.

## 3 Results and Discussion

Figures 1 and 2 illustrate the speedup for different core counts on the cluster and lab PC, respectively.

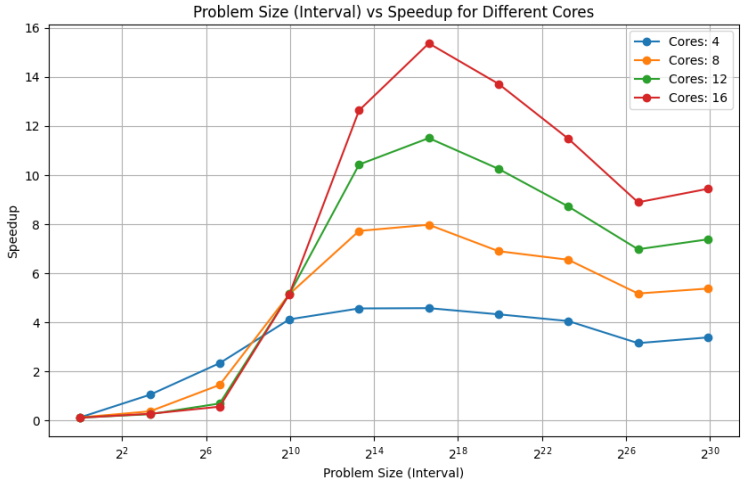


Figure 1: Speedup vs. Problem Size for Different Cores on Cluster

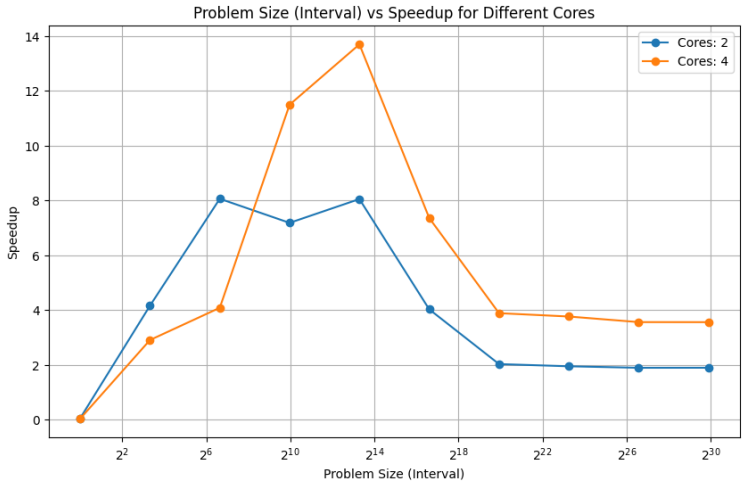


Figure 2: Speedup vs. Problem Size for Different Cores on Lab PC

The results indicate that for large problem sizes, increasing the number of cores significantly improves performance. However, for small problem sizes, the speedup decreases when using more cores. This can be attributed to parallel overhead, including:

- Increased communication and synchronization between cores.
- Load imbalance due to insufficient work per core.
- Cache contention and memory latency.

Figures 3 and 4 illustrate the time taken for different core counts on the cluster and lab PC, respectively.

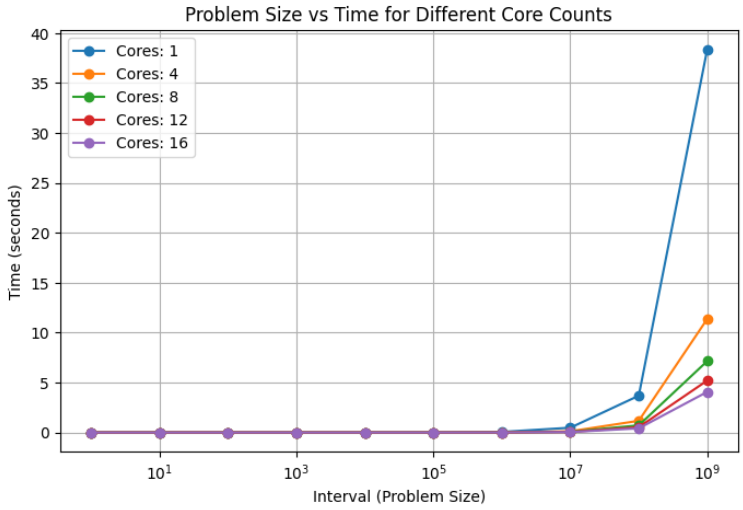


Figure 3: Problem Size vs. Time for Different Cores on Cluster

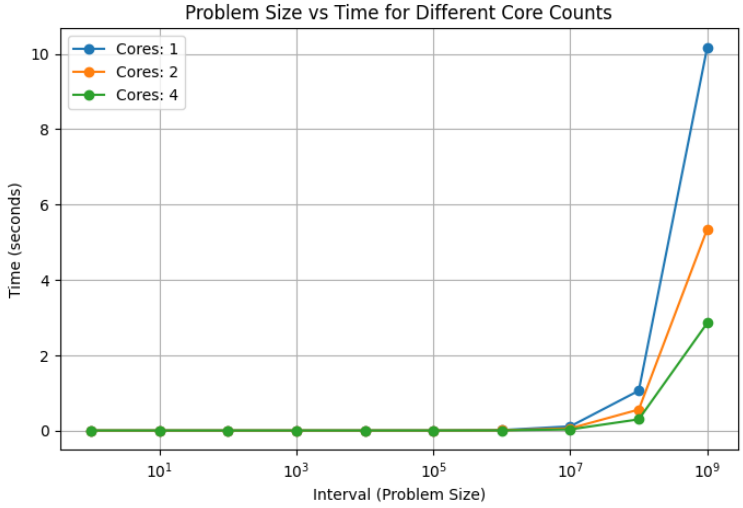


Figure 4: Problem Size vs. Time for Different Cores on Lab PC

On the cluster, higher core counts lead to greater speedup due to superior interconnects and optimized scheduling. However, diminishing returns are observed as problem size decreases. On the lab PC, the speedup is limited due to hardware constraints and non-optimized parallel execution.

# 4 Conclusion

This experiment demonstrates that while parallelization can greatly improve computational efficiency, it is crucial to consider problem size when determining the optimal number of cores. For small problem sizes, single-core execution may outperform parallel execution due to reduced overhead. Future work can explore adaptive parallelization techniques to mitigate these limitations.