

# Word Count Performance Analysis using Multi-threading

High Performance Computing – Task 1

**Team:**

Seif Eldin Mohamed | Mohamed Essam | Mohamed Medhat | Saeed Waled Saeed

**Under Supervision Of:** Dr. Hossam Reda Mohamed



# Problem Description

🎯 **Objective:** Compare Serial and Parallel execution.

📄 **Task:** Count the total number of words in a text file.

📊 **Metrics used:**

- Speedup
- Relative Speedup
- Efficiency
- Scalability

# Solution Overview

## Serial Version

- Read full text file.
- Count all words in one process.
- Measure execution time.

## Parallel Version


- Split the text into equal chunks.
- Each thread counts words in its chunk.
- Merge all partial results.
- Measure total time.



# Implementation Steps

 Upload text file to Colab ( `wordcount_sample_2MB.txt` )

 Read file → split text into words

 Implement `count_words()` function using `Counter()`

 **Implement:**

- `run_serial(text)`
- `run_parallel(text, num_threads)`

 **Measure:**

- `T_serial`, `T_p1`, `T_pN`

 **Compute:**






- $\text{Speedup} = T_{\text{serial}} / T_{\text{pN}}$
- $\text{Relative Speedup} = T_{\text{p1}} / T_{\text{pN}}$
- $\text{Efficiency} = \text{Speedup} / N$

# Results and Metrics

Metric	Value
T_serial	0.1948 s
T_p1	0.1841 s
T_pN (2 threads)	0.1259 s
Speedup	1.55
Relative Speedup	1.46
Efficiency	0.77



# Analysis of Results

-  **Speedup > 1** → Parallel version is faster.
-  **Relative Speedup  $\approx 1.46$**  → good parallel scaling.
-  **Efficiency = 0.77** → 77% resource utilization.
-  Overhead due to thread creation and merging.
-  Small data size limits speedup.



# Conclusion & Future Work

- ✔ Parallelization improved performance (**1.55× faster**).
- 👍 Efficiency shows good use of 2 threads.
- 📈 For larger datasets, speedup is expected to increase.

## 🧪 Future work:

- Try more threads.
- Test on bigger files.
- Compare Multi-threading vs Multiprocessing.



# Key Takeaways

- 💡 Simple task → noticeable performance gain.
- ⚡ Threading is effective for I/O-heavy workloads.
- ★ Important metrics: Speedup, Efficiency, Scalability.
- 🧠 Understanding parallel overhead is crucial in HPC.



# Questions?

Thank You.