सुस्वागतम्

நல்வரவு

ସୁସ୍ୱାଗତମ

సుస్వాగతం

સુસ્વાગતમ

সুস্বাগতম

ಸುಸ್ವಾಗತ

സുസ്വാഗതം

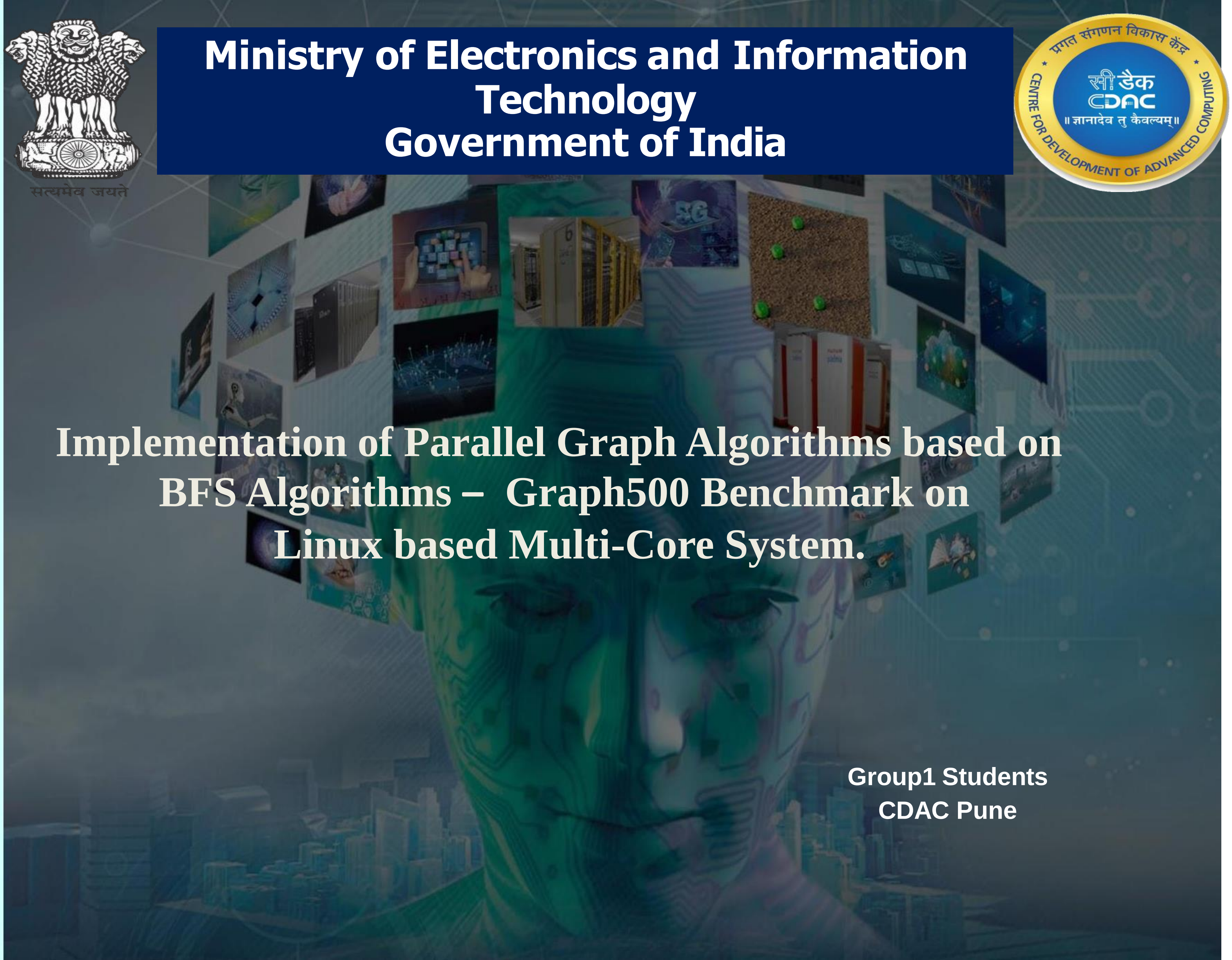
সুস্বাগতম

ਸੁਆਗਤਮ

सुस्वागतम्

خوش آمدید

Ministry of Electronics and Information Technology
Government of India



Implementation of Parallel Graph Algorithms based on BFS Algorithms – Graph500 Benchmark on Linux based Multi-Core System.

Group1 Students
CDAC Pune

CONTENTS

- □ Introduction
- □ Objective
- □ Methodology & Techniques
- □ Implementation
- □ Result
- □ Conclusion
- □ Future Scope
- □ References

INTRODUCTION

High-performance computing (HPC), graph exploration algorithms play a vital role in modeling and processing large datasets, often represented as graphs.

These algorithms find applications in a wide array of scientific fields, including genomics, astrophysics, artificial intelligence, data mining, national security, and information analytics.

Graph search algorithms, particularly Breadth-First Search **(BFS)** and Depth First Search **(DFS)**, serve as foundational components for a myriad of graph-based applications.

OBJECTIVE

- Study and Analyze Existing Parallel BFS Algorithms on Multi-Core Systems
- Performance Analysis on Message Passing Cluster
- Implement Algorithmic Improvements
- Performance Demonstration on PARAM Shavak

METHODOLOGY

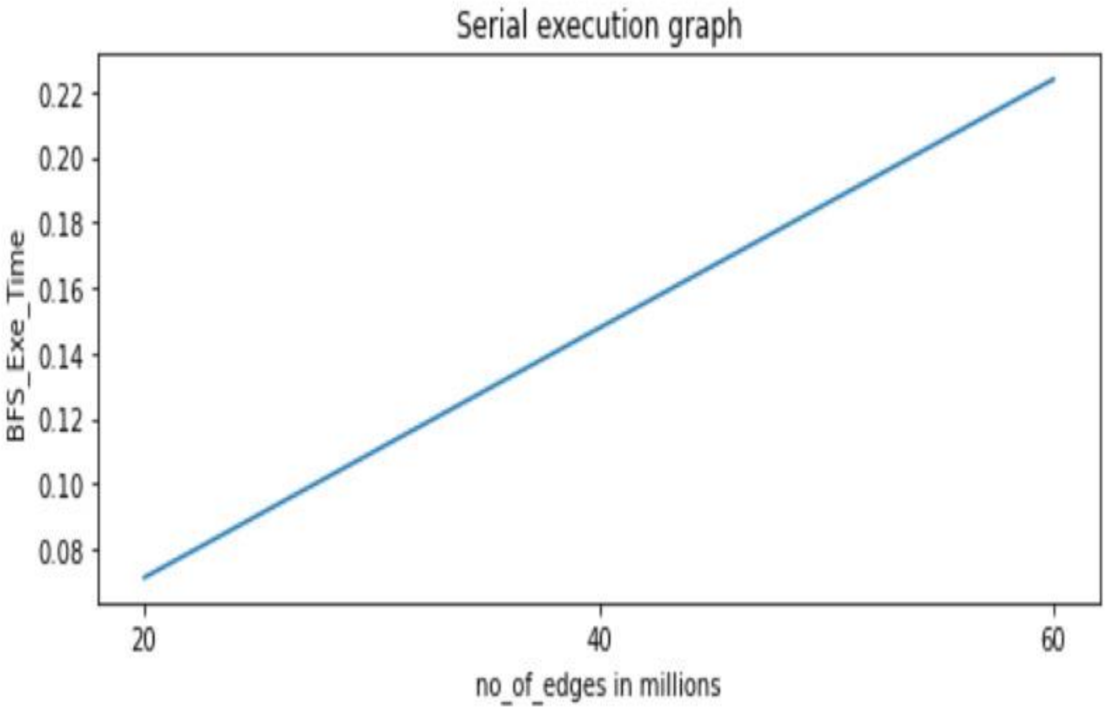
- Analysis of Existing Implementations
- Implementation of serial & parallel code
- Performance Analysis on Message Passing Cluster
- Benchmarking
- Performance Visualization

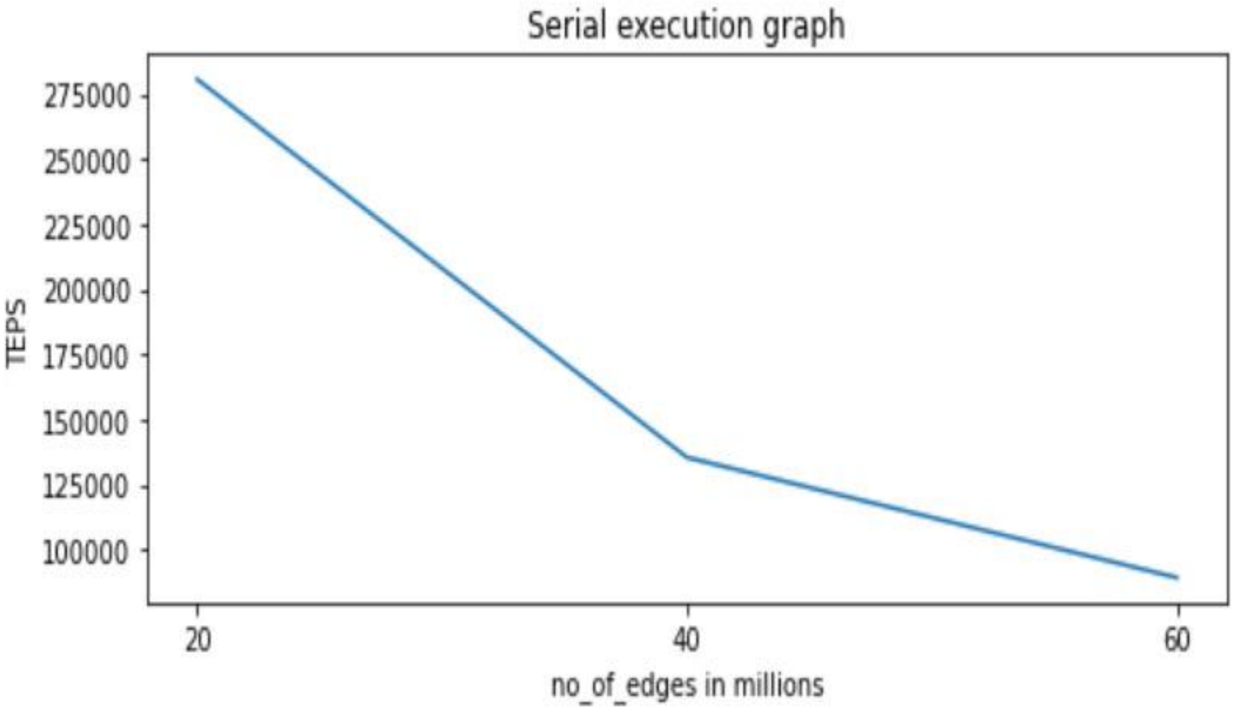
TECHNIQUES

- Parallelization Techniques
- Graph Partitioning
- Message Passing
- Benchmarking Tools
- Visualization Tools
- Code Optimization

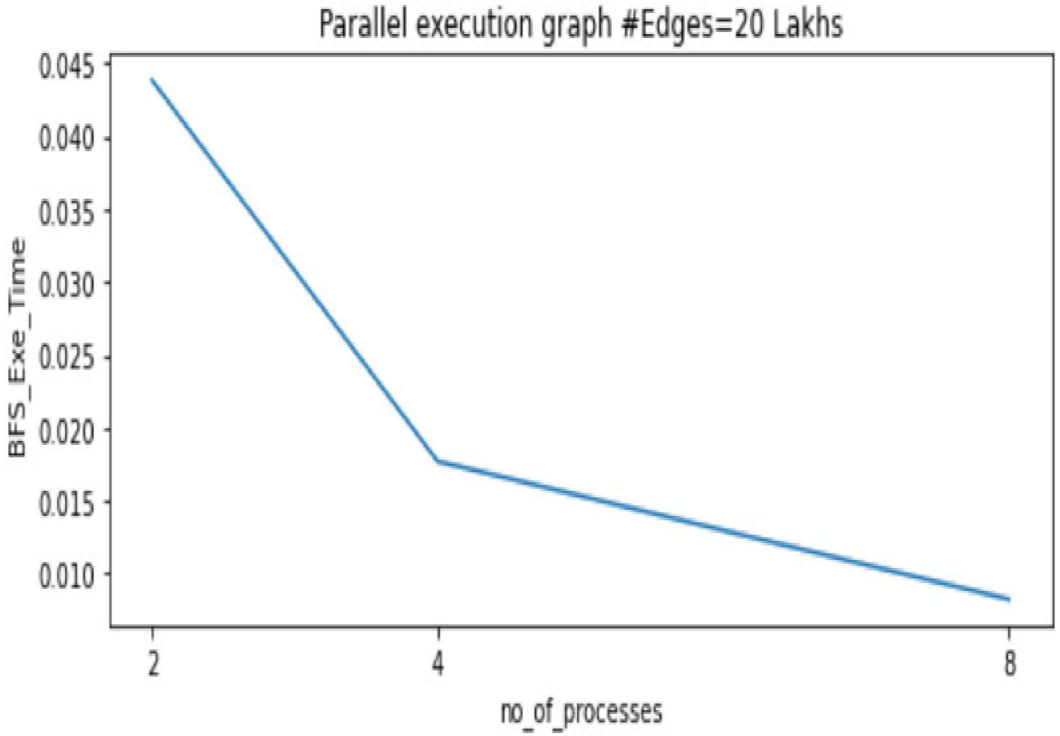
RESULT

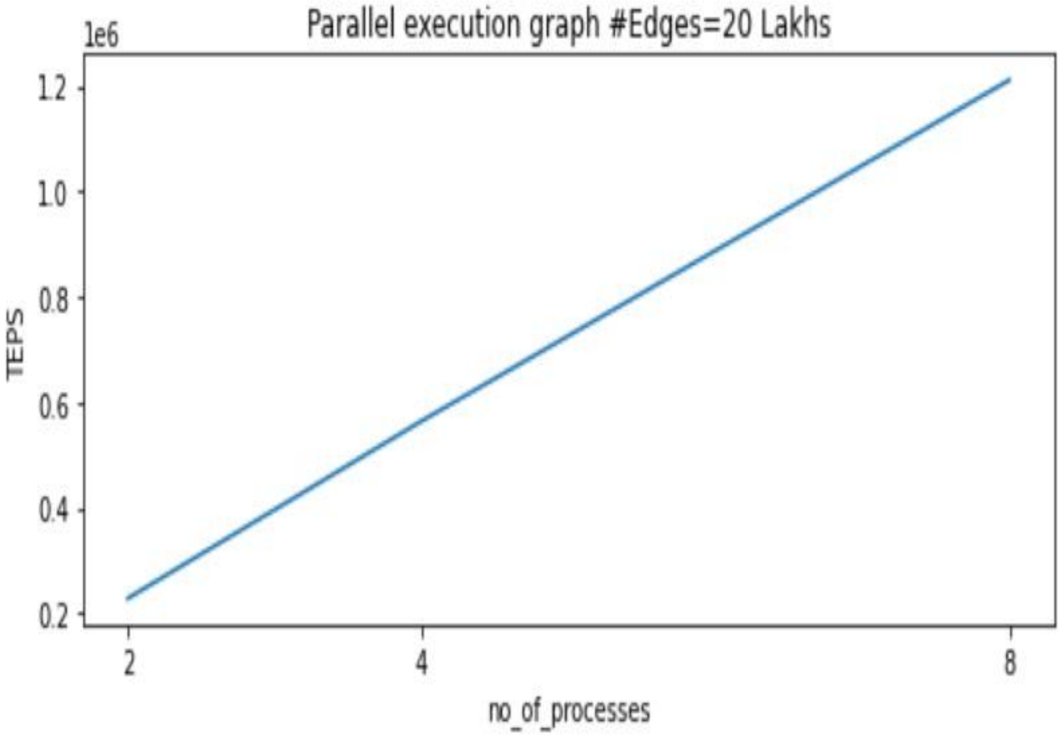
Serial Implementation
No. of constant vertices (N) = 10000



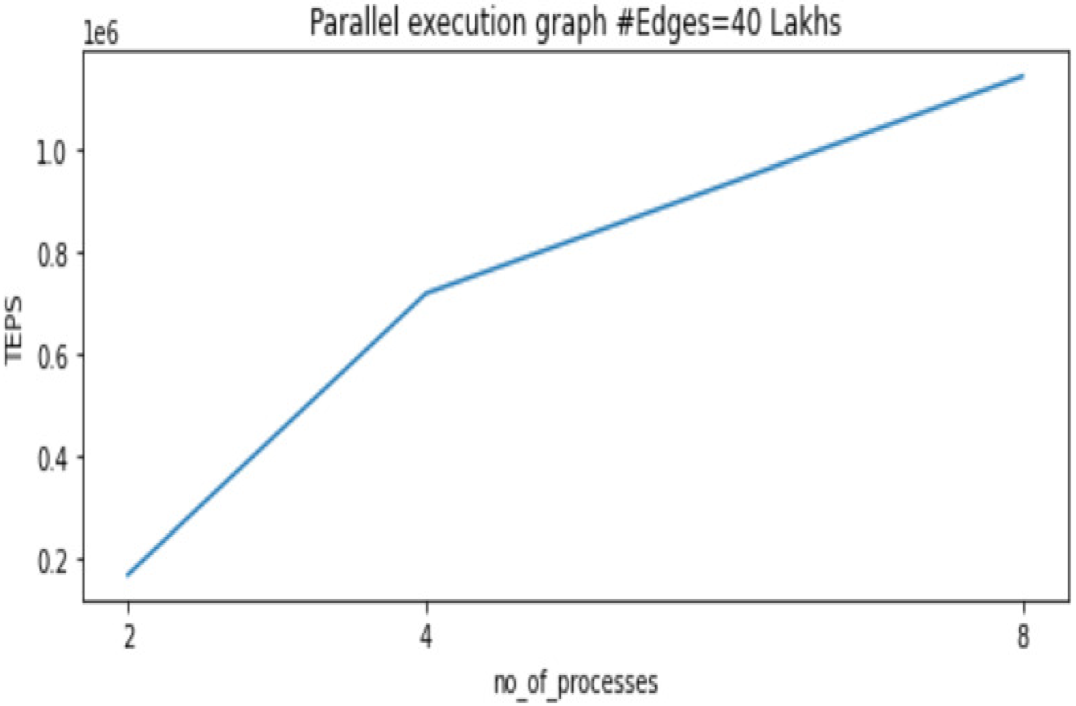


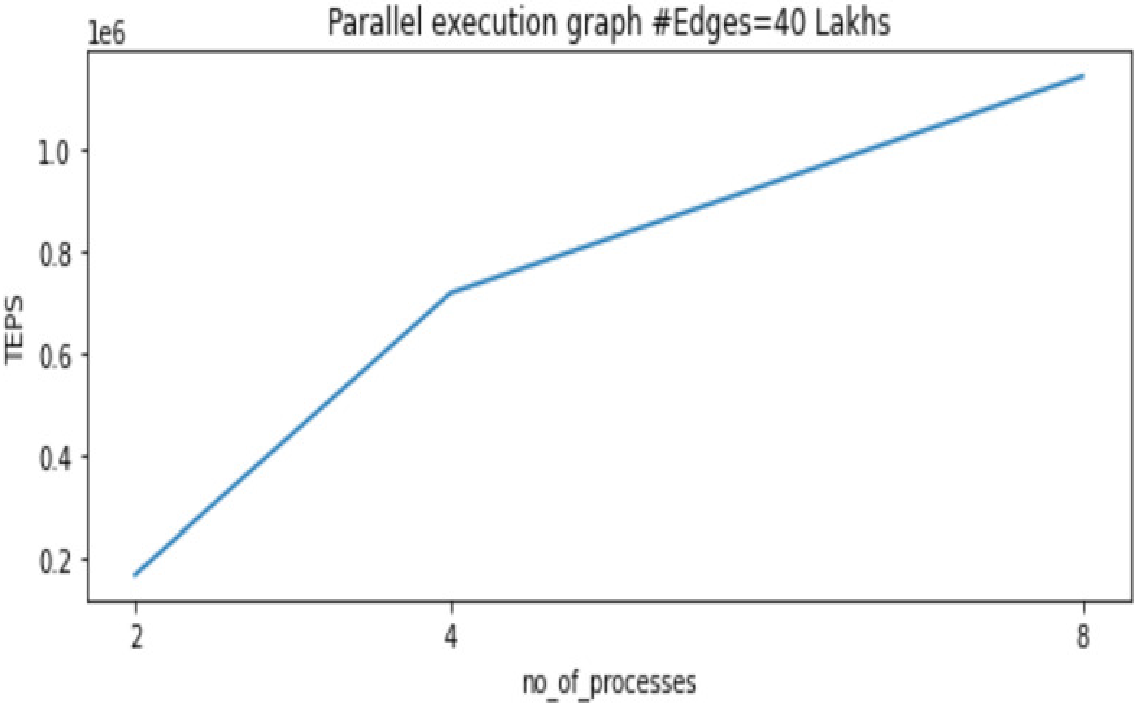
Parallel Implementation
No. of constant vertices (N) = 10000



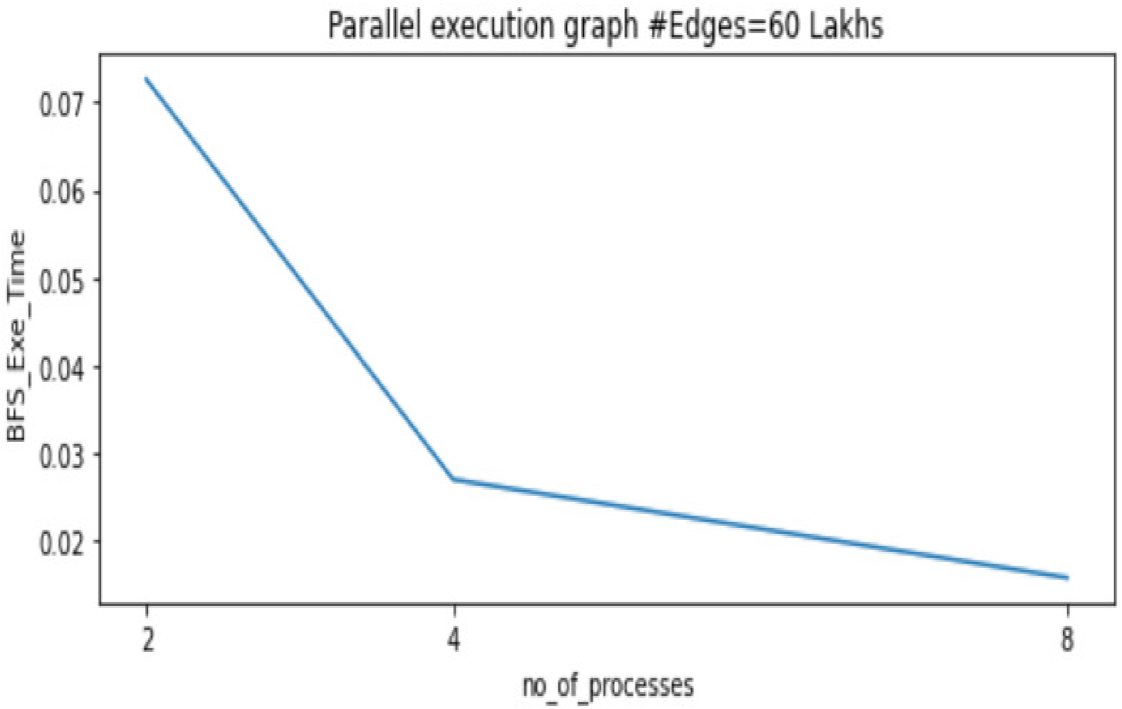


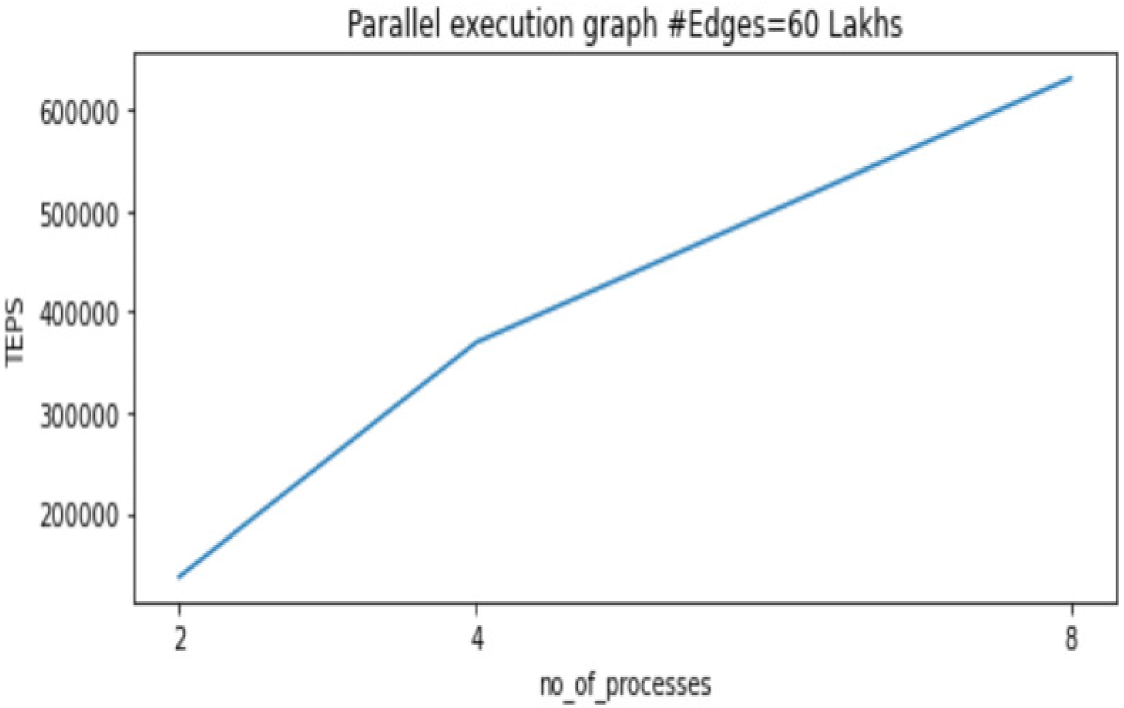
No. of constant vertices (N) = 10000



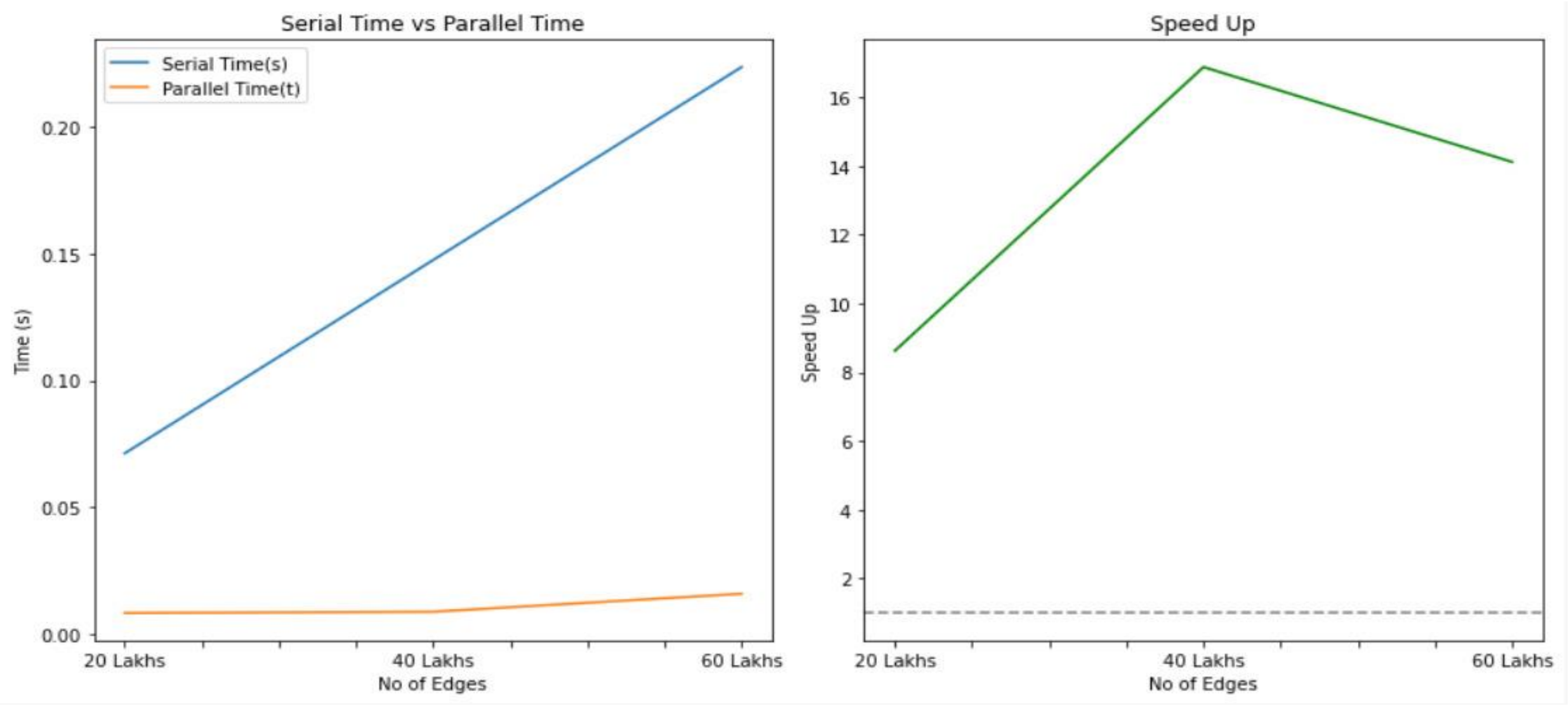


No. of constant vertices (N) = 10000





SPEEDUP


Serial Time vs Parallel Time
Serial Time(s)
Parallel Time(t)
Time (s)
0.00
0.05
0.10
0.15
0.20
20 Lakhs
40 Lakhs
60 Lakhs
No of Edges
Speed Up
Speed Up
2
4
6
8
10
12
14
16
20 Lakhs
40 Lakhs
60 Lakhs
No of Edges

CONCLUSION

Existing Open-Source Implementations: The project thoroughly examined existing open-source implementations of parallel BFS algorithms tailored for Multi-Core Systems. These implementations serve as the foundation for large-scale graph processing.

Performance Analysis: An essential aspect of the research involved analyzing the performance of these BFS algorithms on a Message Passing Cluster. This investigation aimed to assess how well these algorithms scale and utilize CPU resources for large-scale graph processing tasks.

Benchmarking: Extensive benchmarking tests were conducted on various computing systems, including PARAM Shavak supercomputers. These tests provided valuable insights into the speed and efficiency of graph traversal using the Graph500 benchmark.

FUTURE SCOPE

Advanced GPU Utilization: Future work can delve deeper into GPU acceleration techniques, exploring more efficient ways to leverage the computational power of GPUs in BFS algorithms.

Scalability: As data sets continue to grow in size and complexity, further research can focus on optimizing the scalability of BFS algorithms to handle even larger graphs on distributed computing environments.

Hybrid Models: Combining BFS with other graph algorithms, such as graph coloring, can lead to innovative hybrid approaches that offer improved performance and efficiency.

Real-World Applications: Applying optimized BFS algorithms to real-world applications, such as social network analysis or genomics, can demonstrate the practical significance of these research efforts.Top of Form

REFERENCES

1) ToyotaroSuzumura Tokyo Institute of Technology IBM Research – Tokyo / JST CRESTHighly. Scalable Graph Search for the Graph500 Benchmark.

2) Author – AdityaNongmeikapam Course – 633 Parallel Algorithms Instructor – Russ Miller, PARALLEL BREADTH-FIRST SEARCH USING MPI

3) HuiweiLv · Guangming Tan · Mingyu Chen · Ninghui Sun, Understanding Parallelism in Graph Traversal on Multi-core Clusters

Thank You !