COP5615- Distributed Operating System Principles Fall 2023

Programming Assignment #3

Gossip Algorithm

PA_27_Team

Team Members:

Name	**UFID**	**Email**
Rishika Reddy Alugubelli	28880035	r.alugubelli@ufl.edu
Dheekshita Neella	63258153	d.neella@ufl.edu
Aashritha Reddy Donapati	72422083	aa.donapati@ufl.edu
Vivek Reddy Gangula	52087488	gangula.v@ufl.edu

Table of Contents

1. How to run our program?

- Editor Used : Visual Studio Code
- Programming Language: F#
- Supported OS: Windows, MacOS
- Install Ionide for F# Plugin for Visual Studio Code
- Install .NET SDK from dotnet.microsoft.com/download
- Download the Team27.zip file and extract it.
- Open the program.fs file in VSCode.
- Open a terminal and go to the path of the folder.
- Install the AKKA package using the following command: dotnet add package Akka.FSharp
- For compilation, run the 'dotnet build' command in the terminal. The terminal displays a 'Build succeeded' message when the compilation is successful.
- For execution, run the following command in the terminal:

  donet run <numNodes> <topology> <algorithm>

  Possible values for algorithm: gossip or push-sum

  Possible values for topology: full or 2D or line or imp3D

2. What is working?

In our project, we have effectively implemented and demonstrated the Gossip and Push Sum algorithms, showcasing their functionality across diverse network topologies. Our emphasis was on verifying that these algorithms perform their designated tasks adeptly in various network structures, highlighting their adaptability and efficiency in distributed systems.

We implemented the Gossip algorithm to disseminate information, or a "rumor," around the network. Each node (actor) distributes the rumor to a neighbor at random. To prevent indefinite dissemination, we created a method in which actors stop transmitting the rumor after a specific number of times. This ensures that the algorithm converges, implying that the rumor has propagated widely enough across the network. Our solution clearly demonstrates how the Gossip algorithm disseminates information quickly and reliably, regardless of network structure.

Our Push Sum algorithm is intended for computing aggregate values over the network. Based on interactions with surrounding actors, each actor maintains and updates a sum ('s') and a weight ('w'). Convergence happens when the s/w ratio across the network stabilizes. We found the Push Sum algorithm to be particularly successful for distributed aggregate computation after testing it on several topologies, including the difficult Imperfect 3D Grid. This highlights its robustness and potential application in a wide range of diverse and dynamic network conditions.

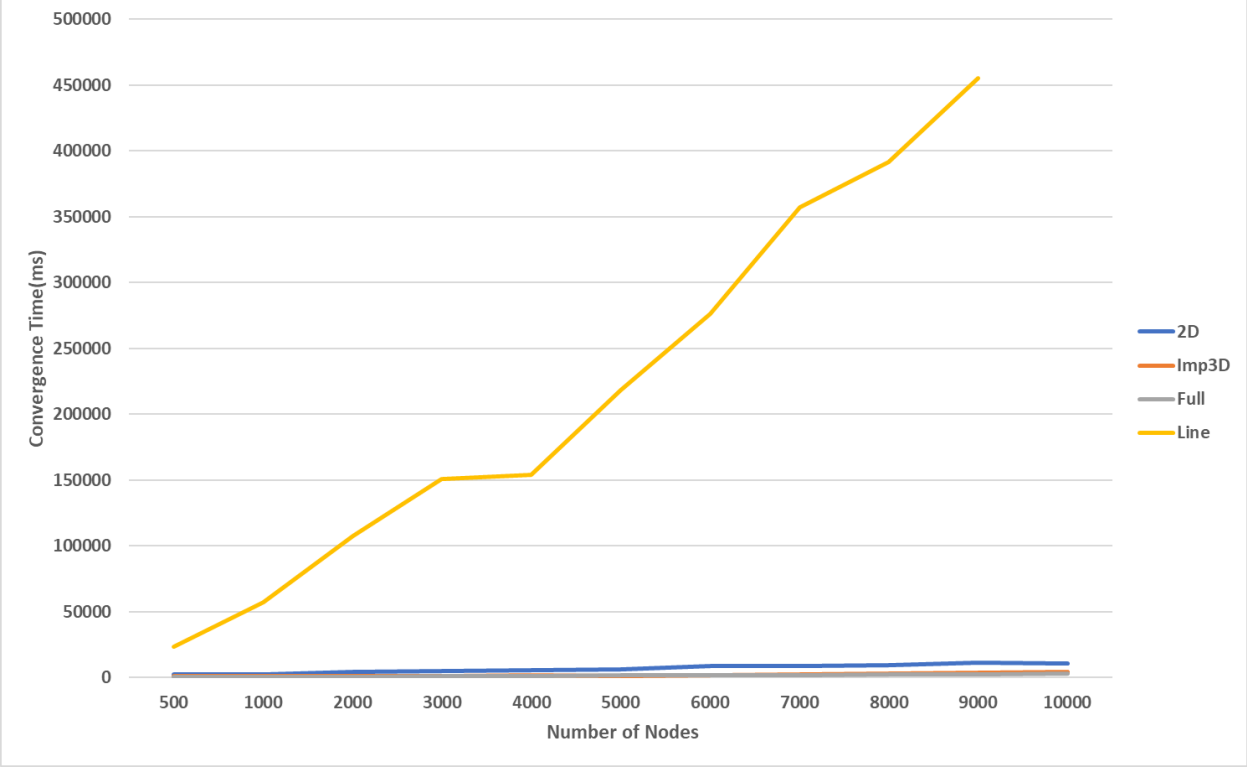
3. Gossip Algorithm

Largest network we managed to deal with for each topology, for Gossip Algorithm:

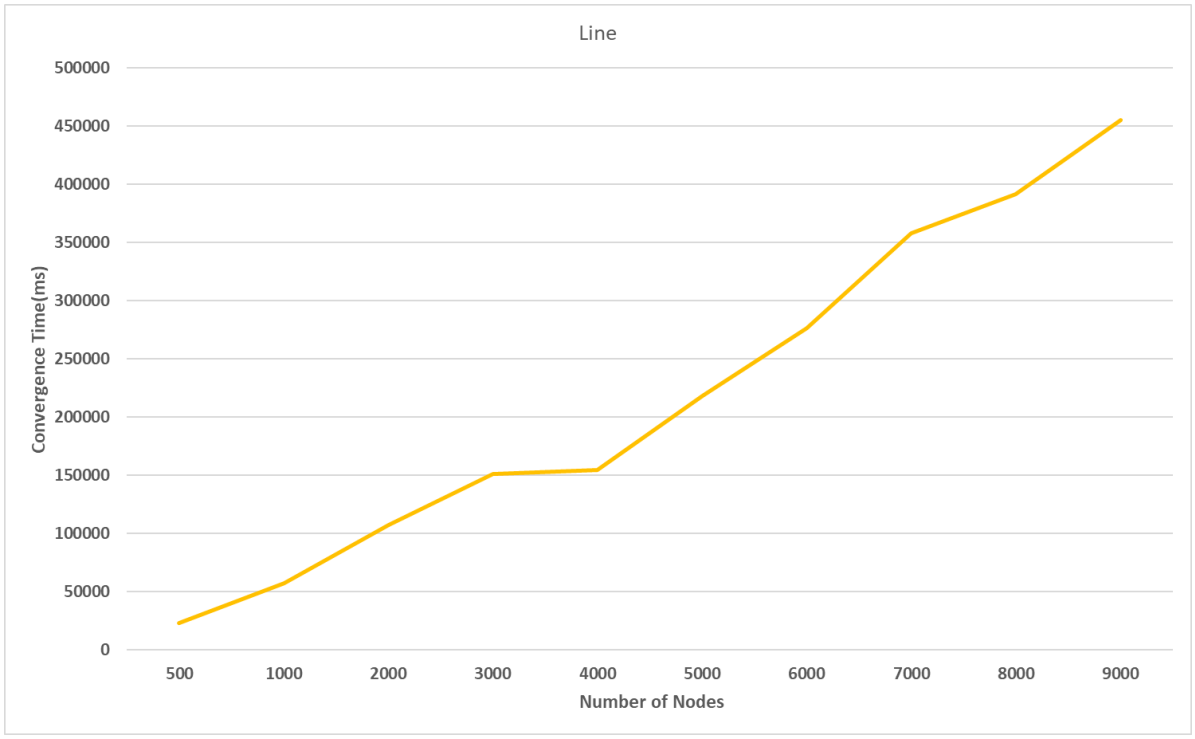
Topology	No. of Nodes
Line	10000
Full	10000
2D	10000
Imperfect3D	10000

Graphical Representation:

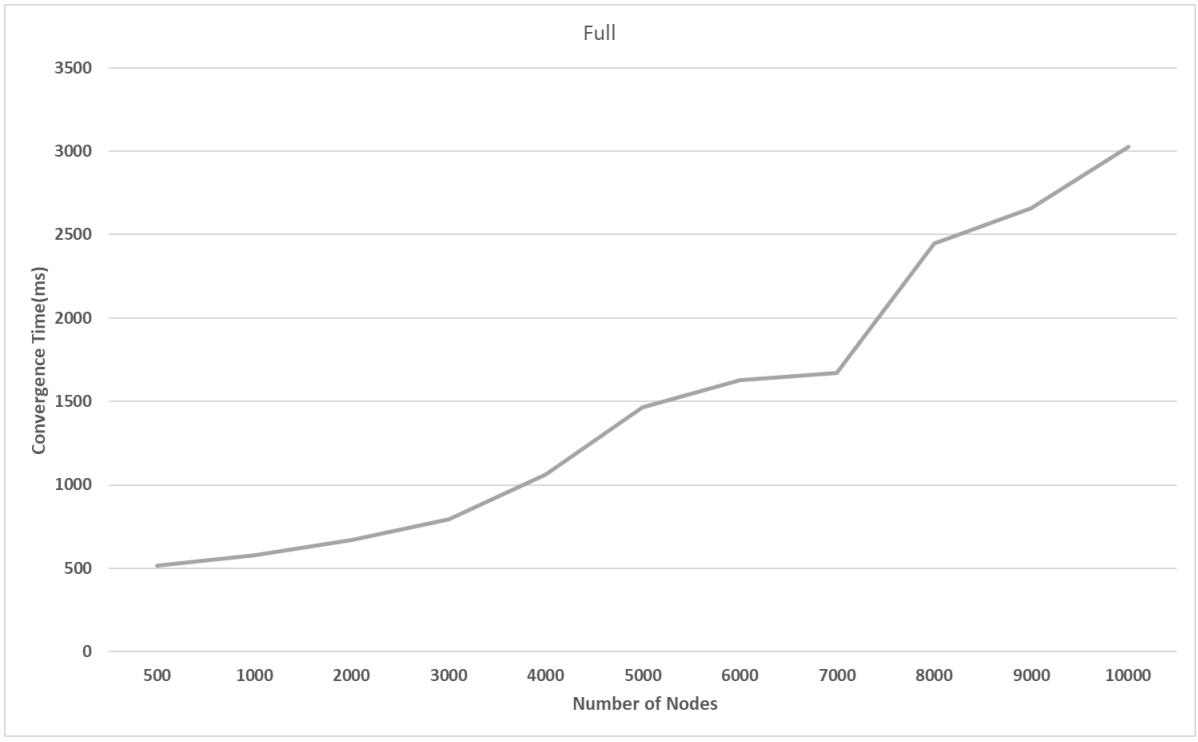
No. of nodes vs Convergence Time for all topologies:



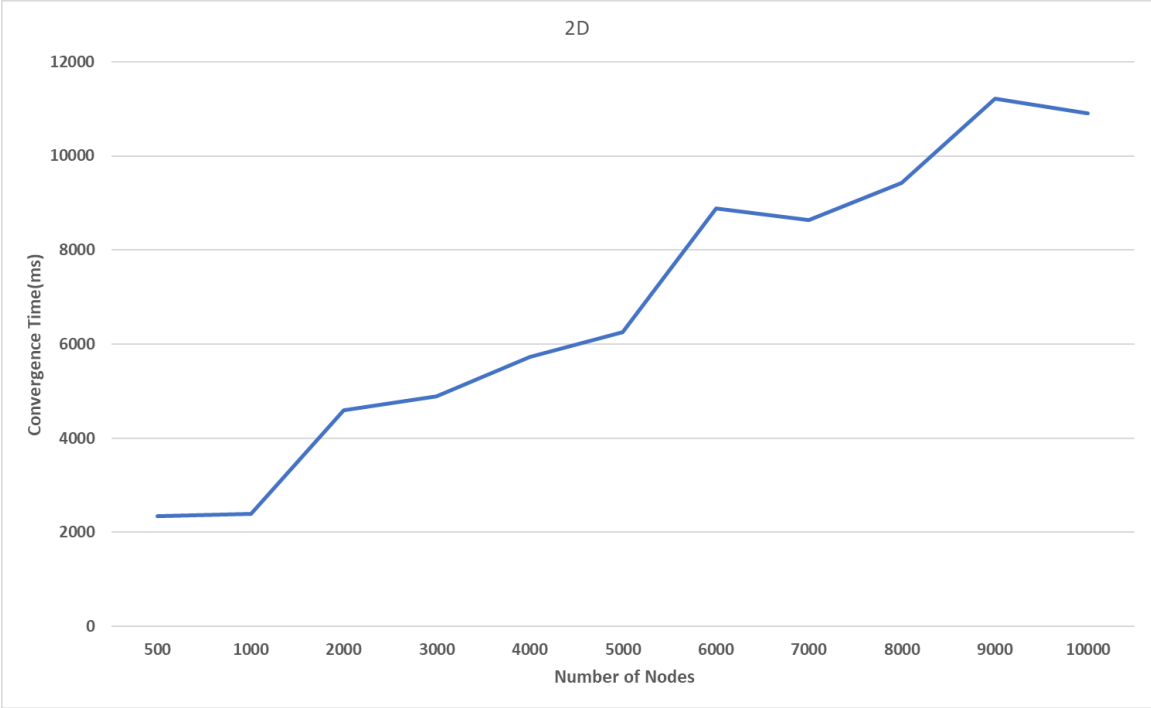
No. of nodes VS Convergence Time for Line Topology:



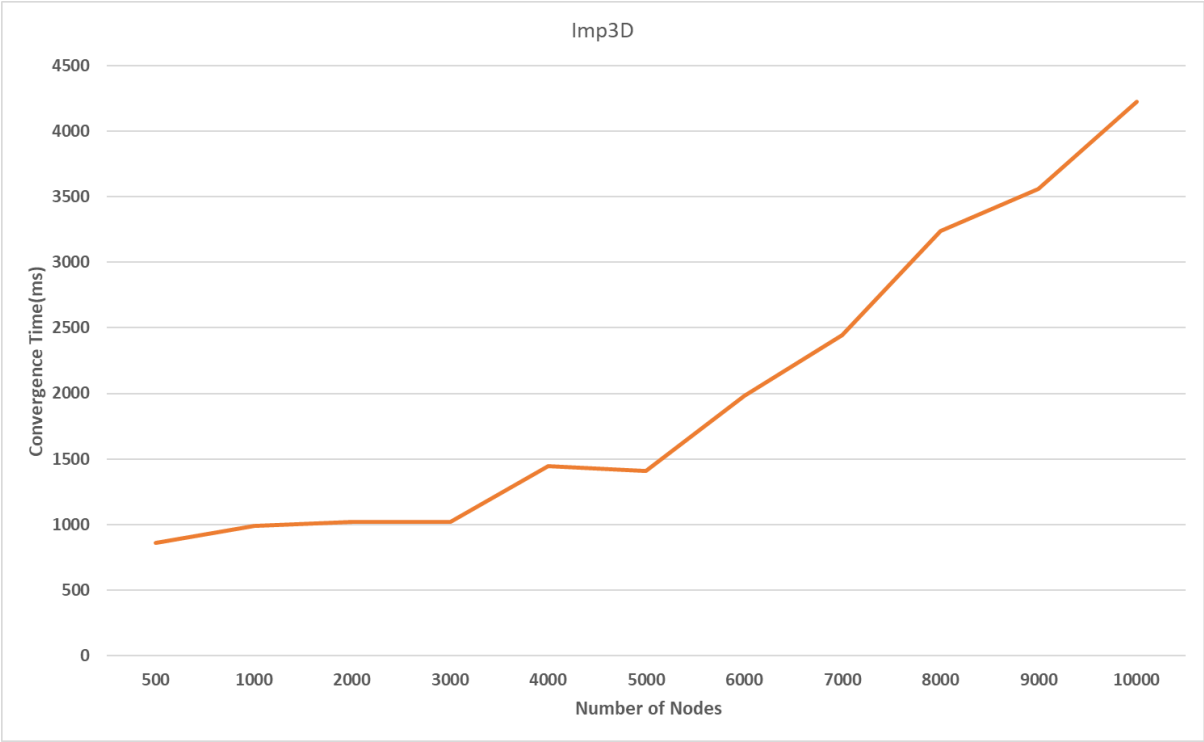
No. of Nodes VS Convergence Time for Full Topology:



No. of nodes VS Convergence Time for 2D Topology:



No. of nodes VS Convergence time for Imperfect 3D Topology:



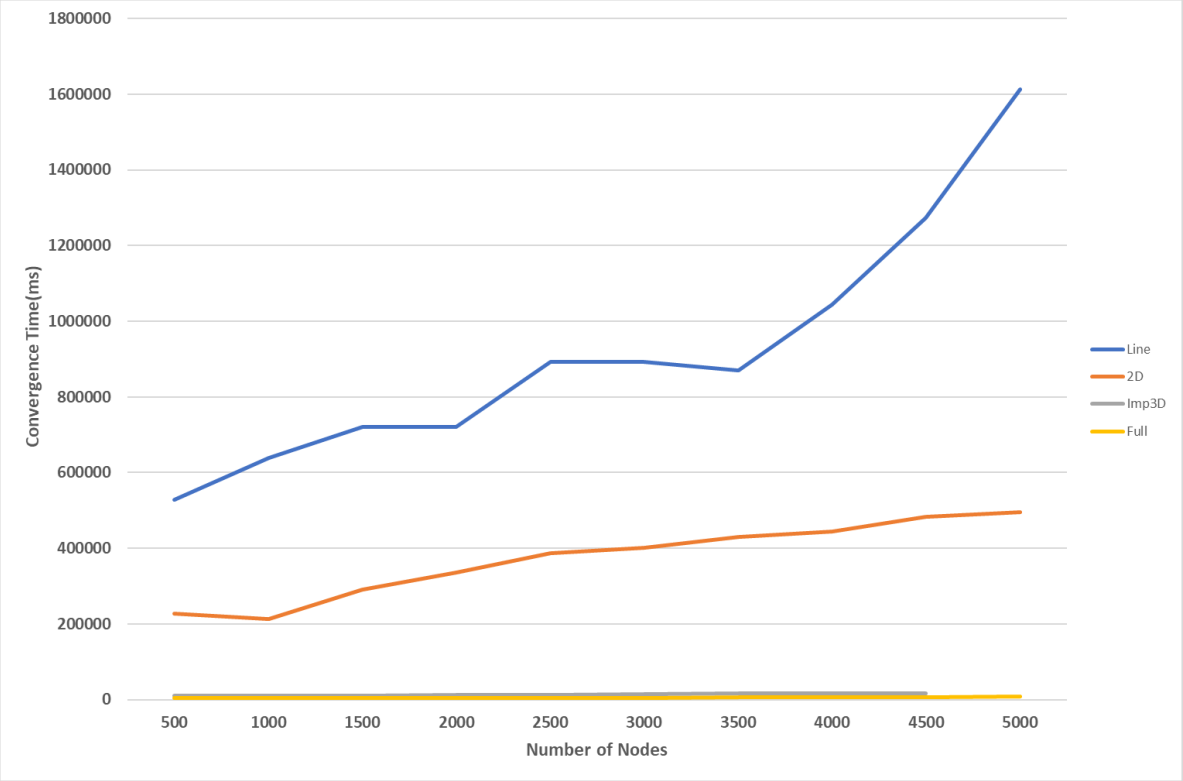
4. PushSum Algorithm

Largest network we managed to deal with for each topology, for Pushsum Algorithm:

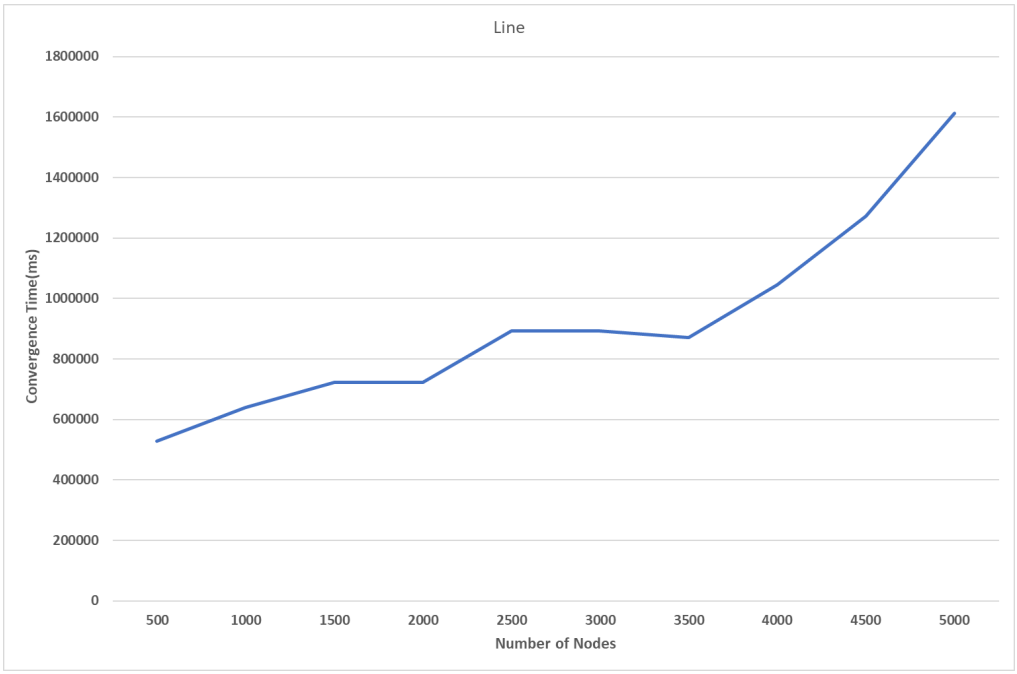
Topology	No. of nodes
Line	5000
Full	5000
2D	6000
Imperfect3D	10000

Graphical Representation:

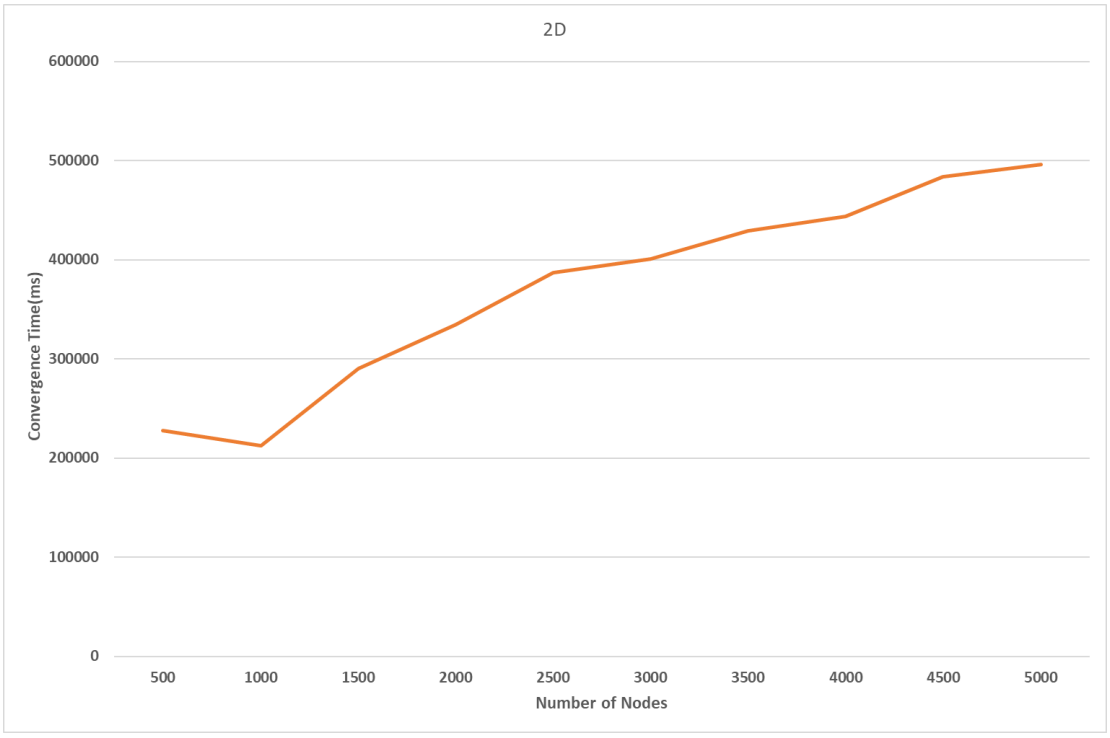
No. of nodes VS Convergence Time for all topologies:



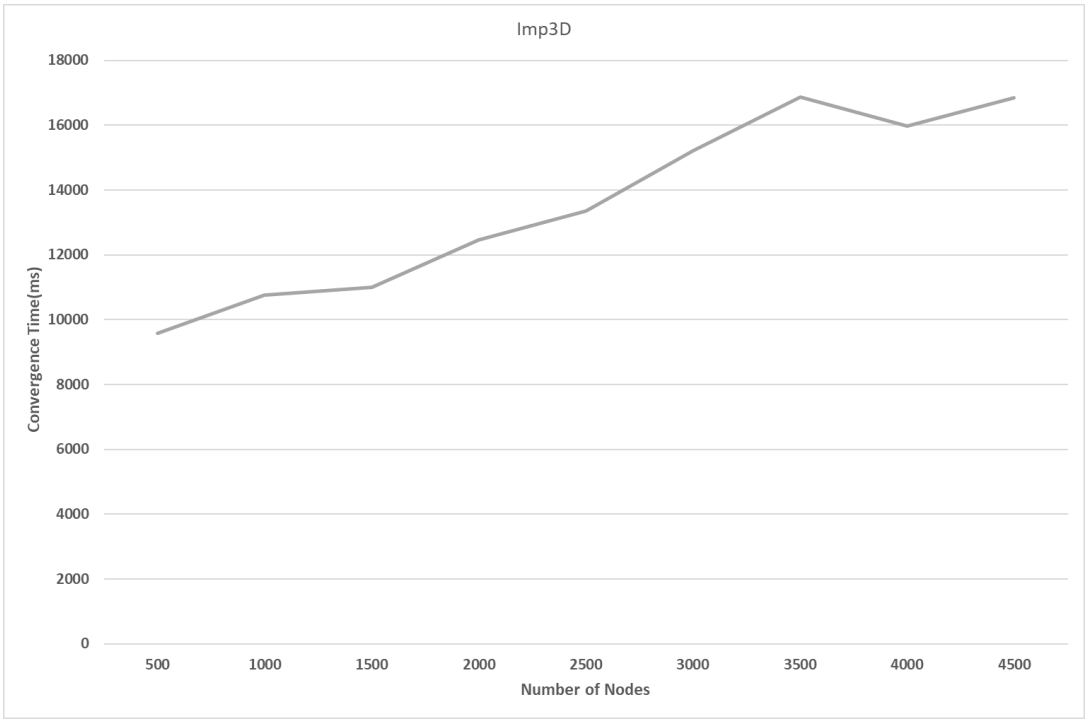
No. of nodes VS Convergence time for Line Topology:



No. of nodes VS Convergence time for 2D Topology:



No. of nodes VS Convergence time for Imperfect 3D Topology:



No. of nodes VS Convergence time for Full Topology: