IMPLEMENTATION OF GOSSIP & PUSH-SUM ALGORITHMS WITH VARIOUS NETWORK TOPOLOGIES.

by

Ujwala Guttikonda (UFID - 57914323)

Nikhil Yerra (UFID - 95453265)

Course COP5615

Distributed Operating System Principles

Herbert Wertheim College of Engineering

UNIVERSITY OF FLORIDA

October 2022

Chapter 1

Introduction

In this study, we describe how the Asynchronous Gossip and Push-Sum algorithms performed when simulating various topologies. The convergence time is also measured for various network sizes, and we report the results for each topology. For Push-sum, this is when the node ratio frequently varies by less than 10 to the power -10, and for Gossip, it is when every node in the network has heard the rumor. The actor model in Erlang is entirely used in the source code, allowing us to simulate node communication and continuously track convergence.

1.1 Results

Below we present the result from running our simulations on all the topologies. The size of the network is gradually increased for each simulation, spanning from 8-4096 nodes in size.Furthermore, a simulation is done by initializing and carrying out the two algorithms on all the topologies for ten rounds. Then finally the average convergence time is calculated over these ten rounds. Another important thing to note is that the number of rounds needed for a Push-Sum node to converge was set to 5 as supposed to 3 (as stated in the project description). Figure 1.1 represents the results yielded from running the Gossip simulation and Figure 1.3 shows the results for the Push-Sum algorithm. Note that the convergence time is measured in the number of messages and not clock time. The reason for this is the apparent latency present when running a large network, which significantly affected the actual time for running our simulations.

1.2 Line

A group of nodes connected to one another in a straight line make up a basic line topology. When Gossip and Pushsum were used, the line topology had the worst performance. This outcome makes sense theoretically. As a result, the network's capacity to effectively spread a message is impacted because each node only has one or two connections. As a result, the message propagates linearly as each node waits for its messages to arrive before convergence occurs at the last node.

1.3 2D Grid

A 2D grid is a topology where all the nodes are randomized in a two-dimensional space in which the neighbors are connected.

A notable difference with the performance of the 2D grid is that its convergence time scales well with the size of the network. When there is a large number of nodes present in the network, the nodes are densely positioned in relation to one another, improving the convergence time significantly. Also, since each node is neighbors with only about 3-4 percent of the total nodes, it does not experience the loading effects described above.

On the other hand, it is not feasible to initialize the 2D grid for a small number of nodes. Due to the randomization of nodes positions on the grid, nodes may form sparse clusters where there are no connections. Even if the network is connected, there may be bottlenecks which significantly delay propagation.

1.4 Imperfect 3D Grid

The 3D grid depicts a three-dimensional space where nodes are connected in an orthogonal grid, or a cube.

In contrast to the Full network, the maximum number of neighbors per node is 6, which lessened the loading issues mentioned above. Furthermore, we saw that the Push-Sum and Gossip algorithms also performed moderately well.

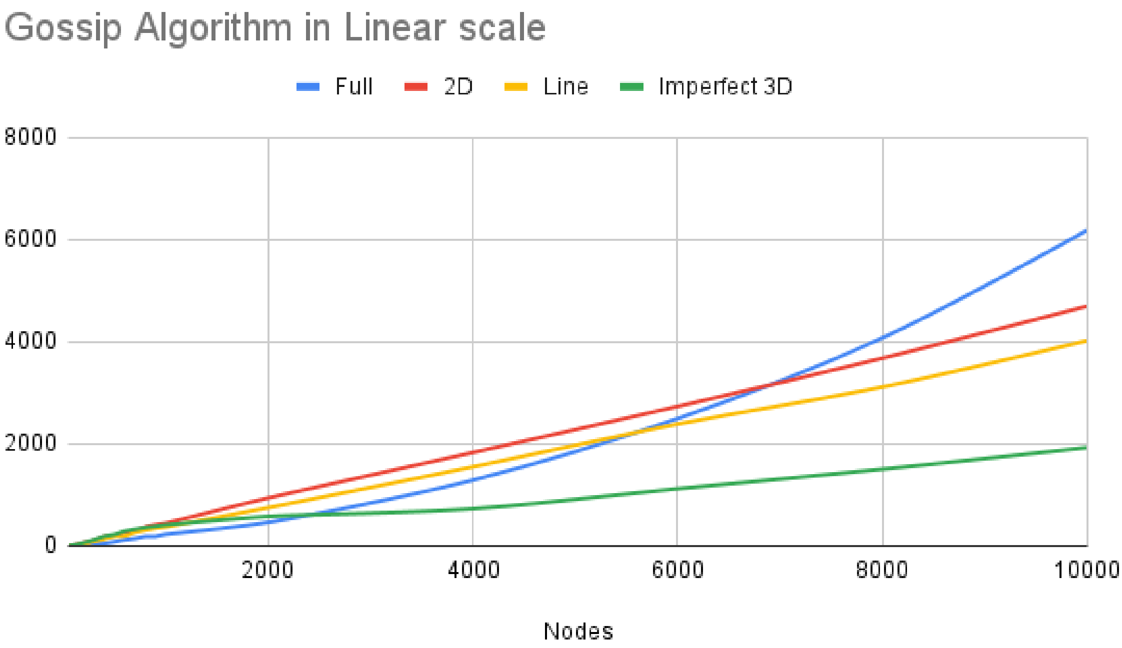

Figure 1.1: gossip algorithm in linear scale

1.5 Full

In a network, every other node is a neighbour of other node.

1.6 Interesting finds

- For the Gossip algorithm, Full topology beats hands down everyone else since every node is connected to every other node, the probability of receiving messages from other nodes increasing drastically. The blindspot (nodes that never converge) in the case of full topology is almost negligible. Therefore making it the best topology to spread the rumor.

In the case of PushSum, if we reduce the gradient of s/w (i.e. 'delta') 10 power -10 to 10 power -5, the topology convergence time is also reduced to half.

Line algorithm will perform poorly since the number of the neighbor is the smallest and thereby also increasing the chances of failure.

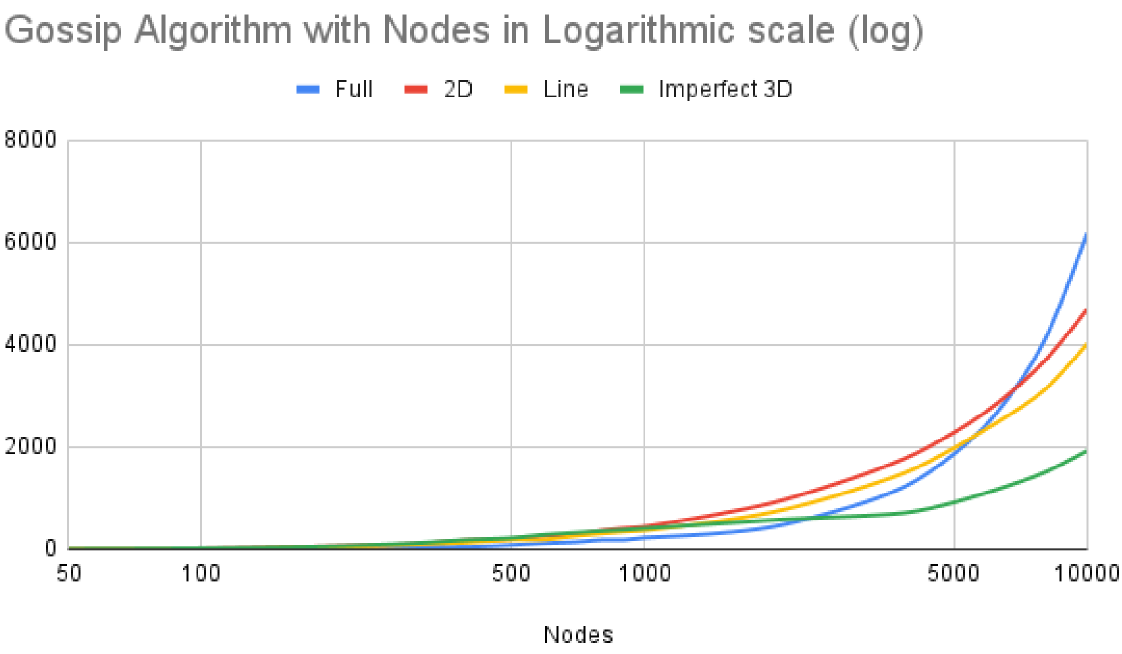

Figure 1.2: gossip algorithm with nodes in log scale

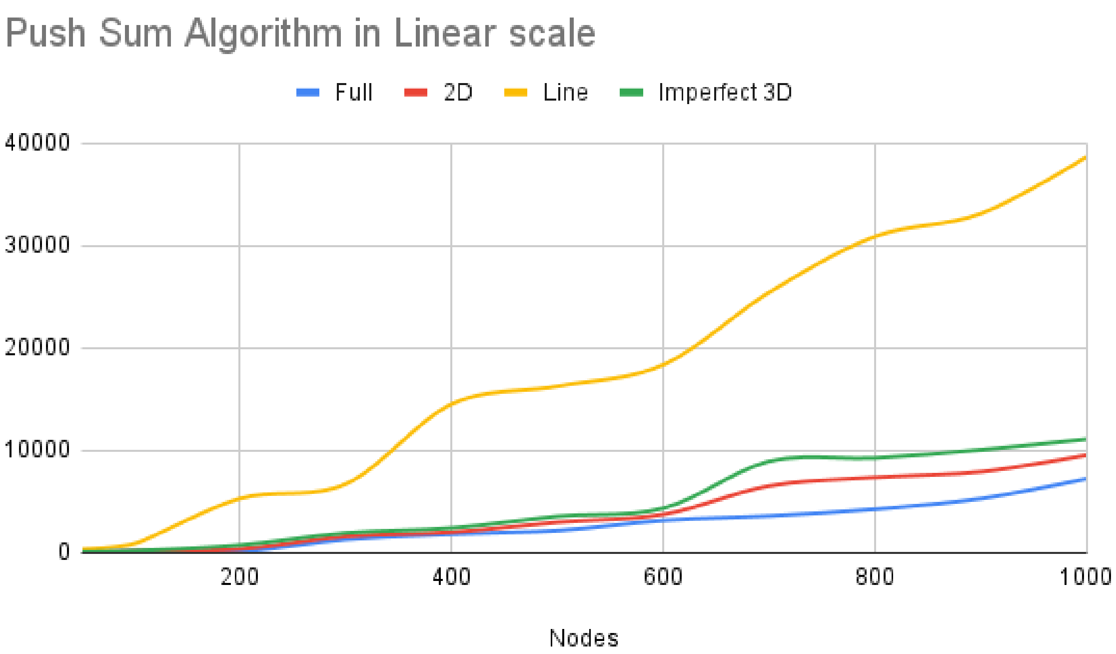

Figure 1.3: PushSum algorithm in linear scale

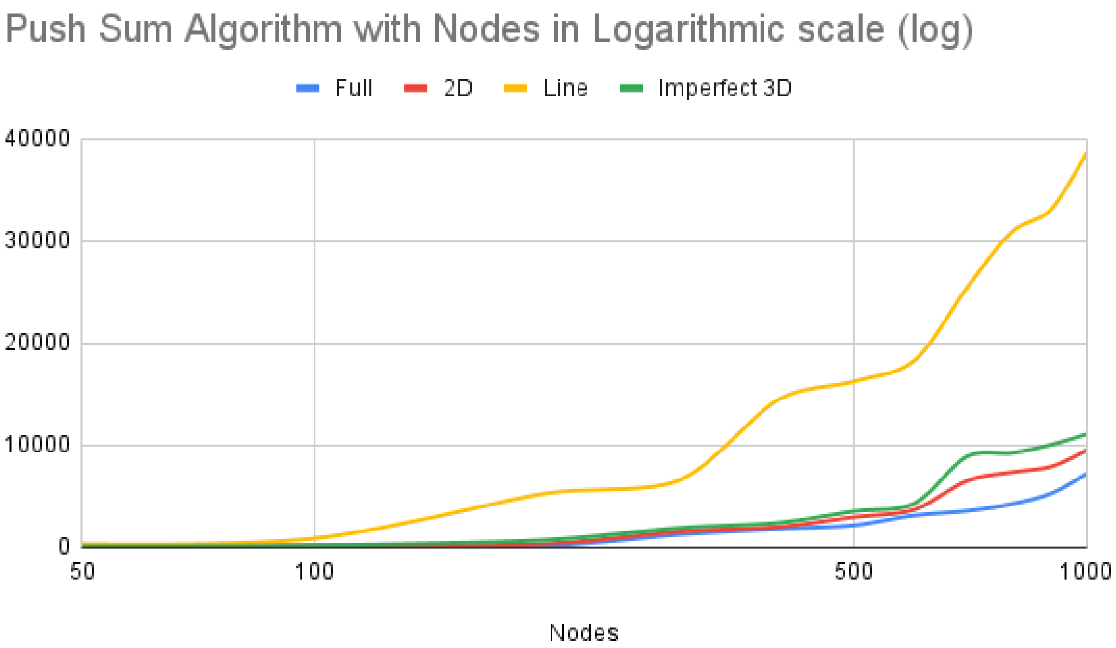

Figure 1.4: PushSum algorithm with nodes in log scale