COP5615 Project 2 -Gossip Simulator

Budigi Sreelalitha
UFID:36717336

Subhasree Jayachandran
UFID: 66222414

Overview of the Project:

Gossip:

The gossip algorithms converge after all nodes have heard a rumor 10 times or when a node has previously communicated the rumor 1000 times to its neighbors. When a node detects that it has heard the rumor more than 10 times, it declares itself convergent, and it also converges when it has delivered the rumor to more than 1000 nodes. When a node notices that all its neighbors have converged, it proclaims itself converged since no more rumor flow can be expected.

The Following are the topologies that are implemented:

Line Topology: Here the gossip is spread across actors who are placed in linear order forming a line. An actor at a position x has atmost 2 neighbors at positions x-1 and x+1. Maximum no of neighbors is 2 and minimum is 1.

Full Topology: In this Topology an Actor has all other Actors as neighbors hence the convergence time is the maximum in this topology.

3D Topology: In this Actors are placed in a 3-Dimensional Cube, Maximum no of neighbors is 8 and minimum no of neighbors is 3.

Imperfect 3D Topology: This is like the 3D topology but each actor one more random neighbor.

Push- Sum:

Push-sum convergence is achieved by presuming that a node's ratio has not changed by more than 10^ -10 in three consecutive rounds. When all the nodes have reached convergence, the process will come to an end.

If we implement the Push-Sum method according to the document's specifications, just one node will converge and the network will come to a standstill, while the other nodes will not. We've taken two approaches to fix this problem:

1) When a converged node receives a message, it sends the identical s and w values to its neighbors but does not keep track of the received s and w values. If all a Node's neighbors are converged, the Node proclaims itself to be converged. We can ensure that all nodes are converged in this fashion.

2) When a Node is ready to converge, it broadcasts its s and w to all of its non-converged neighbors. We may claim that all the nodes have converged in this fashion. We also made certain that a converged node never receives a message.

The first implementation, out of the two, yielded superior results in terms of runtime and s/w ratio. When compared to the second version, the first method is extremely scalable in terms of convergence time.

Project Explantation:

Input:

1. **nodeCount** – Number of Nodes(Actors) participating (Value is any positive Integer)
2. **topology** – Topology used (Line, Full, 3D, Imperfect 3D)
3. **algorithm** – Algorithm Used (Gossip or Push-Sum)

Command to run:

```
dotnet fsi Gossip.fsx nodeCount topology algorithm
```

Implementation Details for Gossip:

1. The given topology is built by constructing a neighbor list for each actor in the start accordingly.
2. The Algorithm is then started by selecting a random Actor to spread the rumour message.
3. An Actor upon receiving an external rumor will increase the *rumourheardtime* by 1, At the same time continues to send the rumor message to its neighbors and this number of times it sends the rumor to its neighbor is handled by *spread.*
4. Each Actor keeps track of the number of times it has received the rumour message. It stop propagating the message to its neighbors when this count is beyond the rumourreceivelimit.
5. Each Actor also keeps track of the no of times it has sent the rumour message , if this is more than rumoursendlimit (1000 in our project) , then it converges.
6. When the neighbors of an actor has converged, then the actor also converges since no more gossip propagation is required.

Implementation Details for Push-sum:

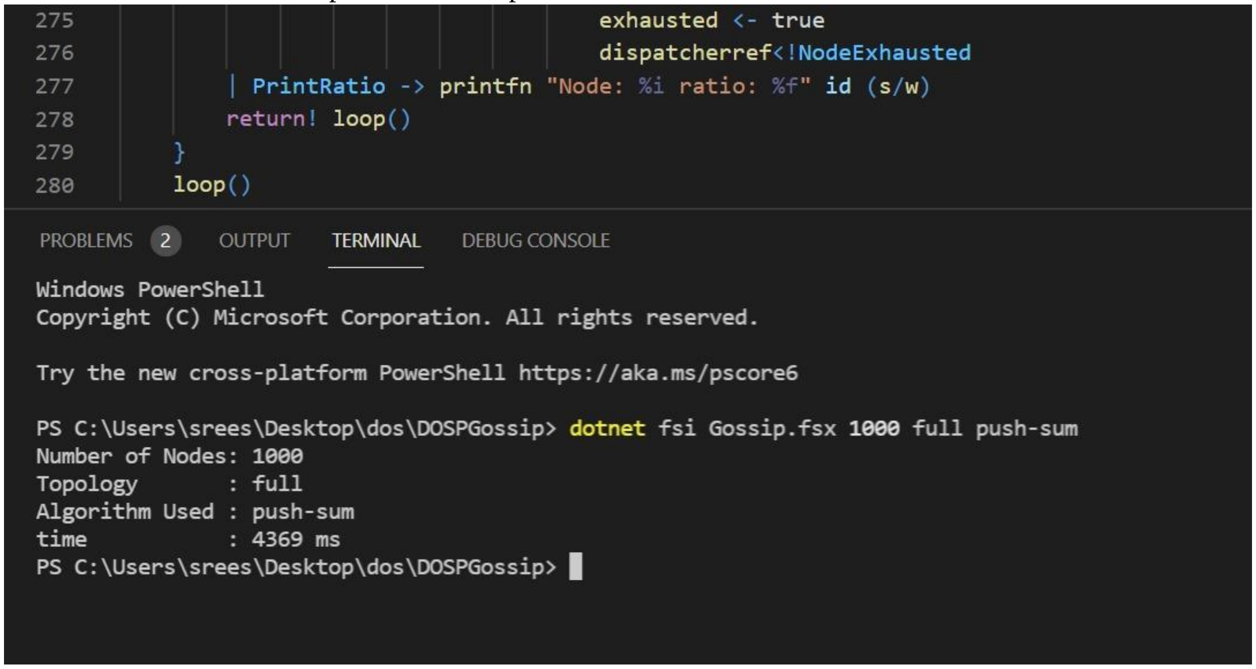
1. Each Actor maintains two variables say, s and w. Initially s = i (i is the Actor Number) and w =1.
2. An Actor starts in the beginning, a message is sent and received in pair of (s,w).
3. Upon receiving an actor keeps half of s and w and sends out the remaining half in message to its neighbors depending on the topology.
4. At a given time sum estimate s/w is computed from current values of s and w of an actor.
5. The actor terminates if the s/w did not change more than 10^-10 in the last 3 consecutive times.

Output:

The output displays the time the algorithm takes to converge for the given topology with given number of nodes in milliseconds.

Sample input/Output:

Command: dotnet fsi Gossip.fsx 1000 full push-sum

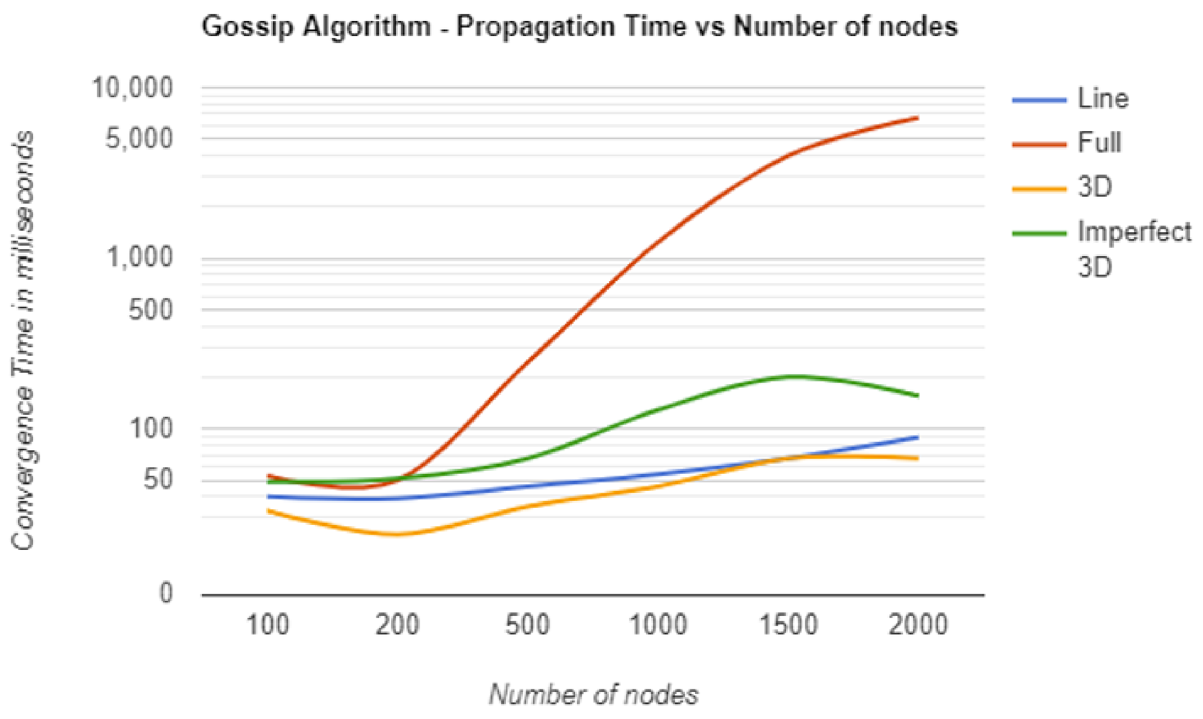


```
                                                exhausted <- true
                                                dispatcherref<!NodeExhausted
                | PrintRatio -> printfn "Node: %i ratio: %f" id (s/w)
                return! loop()
            }
            loop()
```

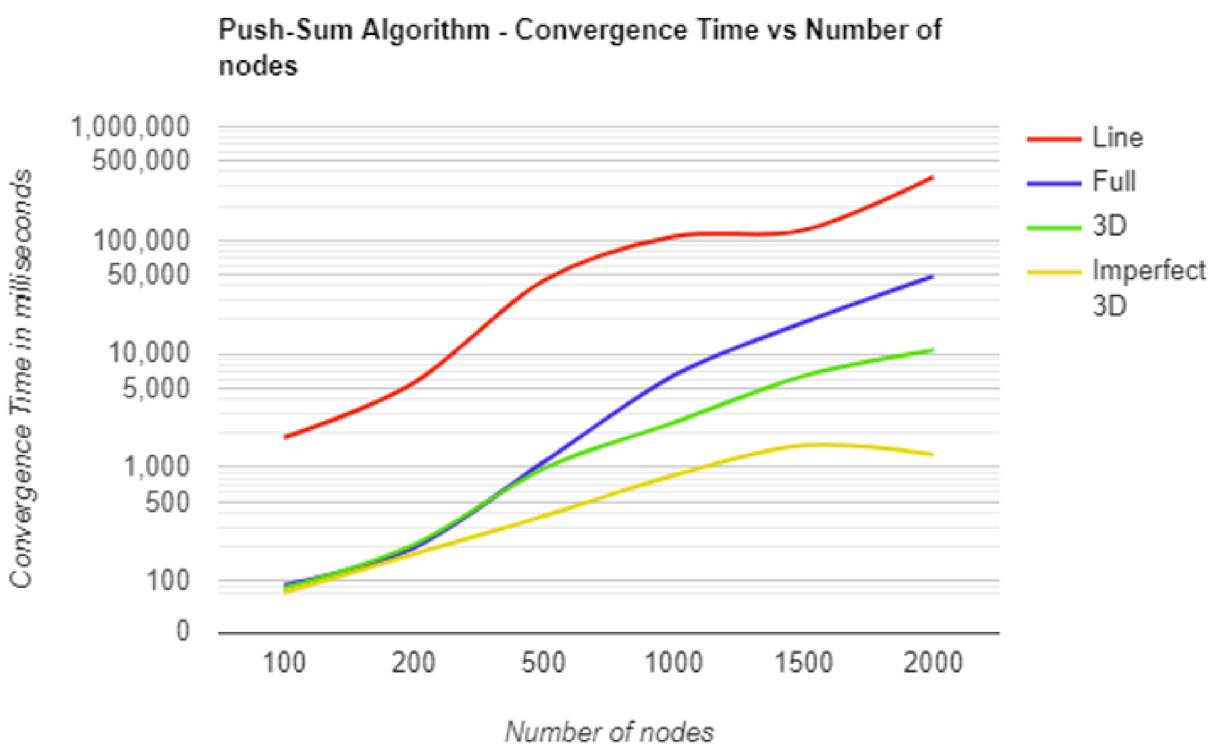
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\srees\Desktop\dos\DOSPGossip> dotnet fsi Gossip.fsx 1000 full push-sum
Number of Nodes: 1000
Topology       : full
Algorithm Used : push-sum
time           : 4369 ms
PS C:\Users\srees\Desktop\dos\DOSPGossip>
```

Results:

Gossip Algorithm (Logarithmic Scale) plot :



Push-Sum (Logarithmic Scale) plot :



Interesting Findings:

Gossip Algorithm:

- *When the rumour has started to propagate among n nodes at a time T from a node,*
  - *In case of a line topology n+2 nodes could have received the rumour at time T+1.*
  - *In case of a Full Topology 2n nodes could have received the rumour at time T+1*
  - *The Imperfect 3D and 3D topology progation time will fall in between and Imperfect 3D has an edge over 3D.*
  - *Hence the Expected Propagation time is* Full < Imperfect 3D < 3D < Line
- *Our project has achieved Line~3D<Imperfect 3D < Full, the reason is because since we are simulating more nodes on a quad core machine where we cannot guarantee that all actors get same amount of execution time and cannot guarantee parallelism of all nodes.*
- *Here as the number of nodes increases the convergence time increases exponentially.*

Push Sum:

- *Push sum takes much longer time converge than gossip since each node has to send the s and w value as well to the network.*
- *If there more number of neighbors, then the network is denser and hence the values are sent uniformly throughout the network resulting in a faster convergence.*
- *Hence full topology converges first, followed by Imperfect 3D and then 3D topology.*
- *In line topology the network is weakly connected and not dense hence take more time than the other topologies to converge.*
- *The result we have got is as expected, this is because we can guarantee that all the nodes get equal amount of computational time because only few nodes perform the actions at any point of time*

What works:
Converges of nodes when connected in line, full, Imperfect3D, 3D topology for the gossip and Push-sum Algorithms.

Largest Network Managed for Each Algorithm – Topology

Algorithm	**Topology**	**Max Nodes**
Gossip	Line	10000
	3D	55000
	Imperfect 3D	55000
	Full	10000
Push sum	Line	2000
	3D	8000
	Imperfect 3D	8500
	Full	10000