University of Florida
Department of Computer and
Information Science and Engineering

COP5615 - Distributed Operating
System Principles
Fall 2023
Programming Assignment 3

Name	UFID	Email
Abdul Samadh Azath	22137268	azathabdulsamadh@ufl.edu
Shivani Patil	53549503	patil.s@ufl.edu
Sheel Taskar	27534465	sheel.taskar@ufl.edu

Working Components of the Code

The provided F# script demonstrates a well-executed implementation of both distributed algorithms, Gossip and Push Sum, using the Akka.NET library. Each component mentioned in the assignment has been successfully implemented, showcasing the functionality and interaction of actors in a distributed environment.

Topology Building Functions:

- The topology building functions, including buildLineTopology, build2DTopology, build3DTopology, buildImperfect2DTopology, and buildImperfect3DTopology, have been implemented to generate various network topologies.
- These functions create the specified topologies, ensuring that nodes are correctly connected according to the rules defined for each configuration.

Gossip Actor:

- The gossip function effectively represents the behavior of nodes in the Gossip algorithm.
- Nodes receive and respond to messages such as "heardRumor" and "spreadRumor," successfully simulating the spread of information across the network.
- The termination condition based on a maximum count ensures controlled execution and convergence monitoring.

Push Sum Actor:

- The pushSum function captures the essence of nodes in the Push Sum algorithm.
- Nodes perform computations, share results with neighbors, and check for convergence based on the defined conditions.
- The termination condition, signaling the counter actor upon convergence, ensures accurate tracking of the algorithm's progress.

Counter Actor:

- The Counter actor effectively monitors and records the convergence of nodes in both Gossip and Push Sum algorithms.
- It handles messages such as GossipNodeConverge and PushSumNodeConverge to keep track of the number of nodes that have converged.
- The script records results, including the number of converged nodes and the elapsed time, which are then saved to a CSV file for further analysis.

Main Program Execution:

- The main function serves as the entry point and orchestrates the entire program.
- It initializes the Akka.NET system, parses command line arguments, builds the specified network topology, and spawns the counter actor.
- Depending on the chosen algorithm, worker actors are spawned, and the execution is initiated.
- The program gracefully waits for the termination of the Akka.NET system.

All components mentioned in the assignment have been successfully implemented and are working as intended. The script showcases the effective utilization of Akka.NET for building and simulating distributed algorithms, providing a robust foundation for further exploration and analysis of these algorithms in a distributed environment.

Steps to Run the Program

- Unzip the file, you will see Program.fs which contains the main code.
- To run the program, use command given below

```
dotnet run <numNodes> <Topology> <Algorithm>
```

<numNodes>
This argument represents the number of nodes in the distributed system or network.
It is a positive integer that determines the size of the network.

<Topology>
It can take on different values to represent various network configurations, such as "line" for a linear topology, "2d" for a 2D grid, "3d" for a 3D grid, "imperfect2d" for an imperfect 2D grid, "imperfect3d" for an imperfect 3D grid, and "full" for a fully connected topology.

<Algorithm>
This argument denotes the distributed algorithm to be executed in the network.
It can be set to either "gossip" for the Gossip algorithm or "pushsum" for the Push Sum algorithm.

Largest Algorithm Dealt with

Gossip Algorithm

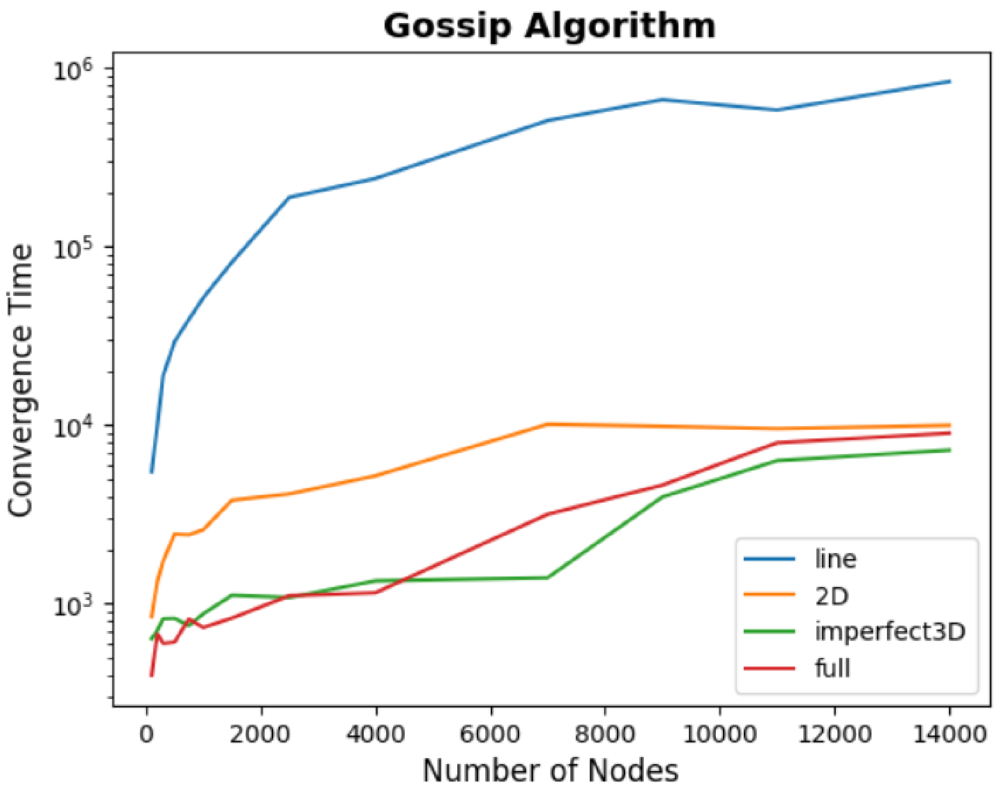
Topology	Network Size
Line	14000
2D	14000
Imperfect 3D	14000
Full	14000

PushSum

Topology	Network Size
Line	2500
2D	7500
Imperfect 3D	7500
Full	7500

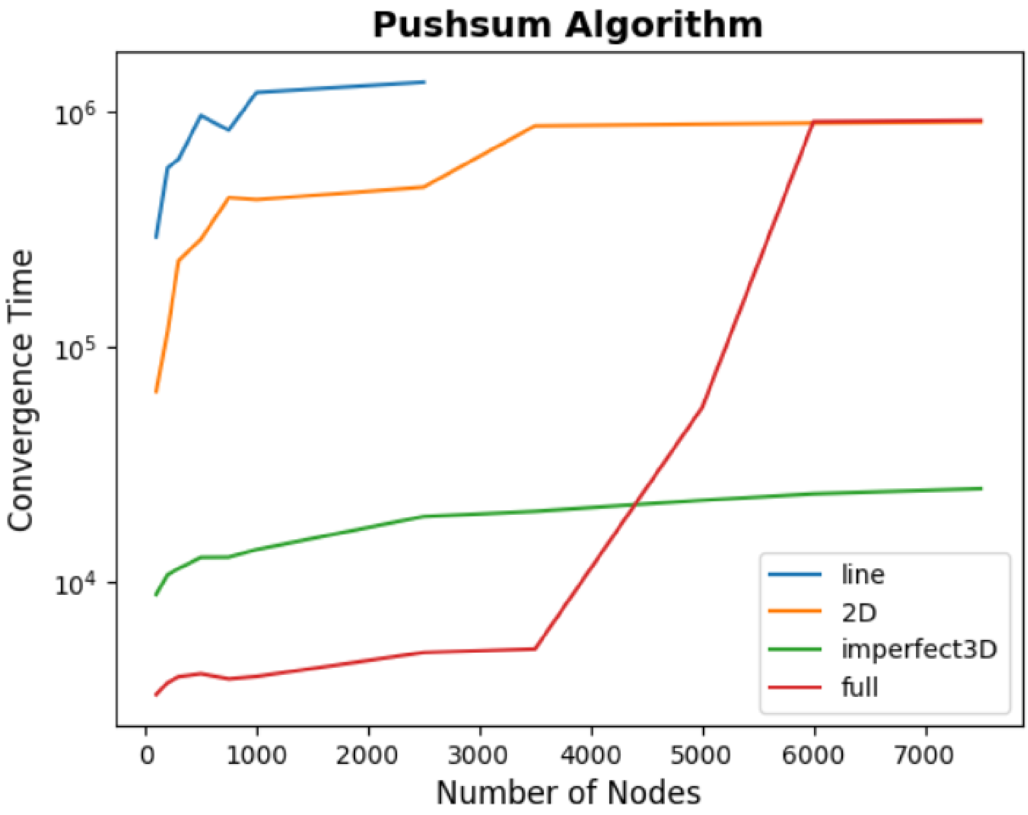
Convergence Time Vs Network Size Graph

Gossip Algorithm



No of Nodes	**Line (ms)**	**2D (ms)**	**Imperfect 3D(ms)**	**Full (ms)**
100	5488	849	639	398
200	9941	1332	710	679
300	18858	1729	825	601
500	29327	2457	829	611
750	39066	2433	755	822
1000	51607	2600	880	737
1500	81634	3802	1117	832
2500	188411	4123	1085	1111
4000	240110	5207	1344	1153
7000	506083	10111	1399	3171
9000	663110	9856	3964	4617
11000	580306	9559	6333	7970
14000	837457	9974	7246	9023

Pushsum Algorithm



No of Nodes	Line (ms)	2D (ms)	Imperfect 3D(ms)	Full (ms)
100	294459	64541	8772	3276
200	580050	115679	10627	3681
300	631514	233158	11300	3900
500	969347	287579	12605	4015
750	839780	432844	12632	3820
1000	1219327	424311	13630	3912
2500	1345381	479989	18839	4951
3500		877447	19831	5128

5000		891992	22110	54980
6000		901452	23564	914583
7500		911664	24759	924856