

COP5615: Distributed Operating System Principles

Project 2: GOSSIP ALGORITHM

Arnav Agarwal– UFID: 82177227

Sai Pande – UFID: 37696687

University of Florida

Computer Science and Information Science Engineering

Introduction

This project aims to study the convergence of two key algorithms, Gossip and Push-Sum, used in distributed systems for information propagation and sum computation, respectively. The algorithms are executed using asynchronous actors in Pony, with a focus on measuring their performance across different network topologies and varying numbers of nodes.

The goal of the project is to determine how long it takes for the algorithms to converge across different topologies and network sizes, which can significantly affect the speed of information dissemination or computation.

The main objectives include:

1. Implementing the Gossip and Push-Sum algorithms using Pony's actor model.
2. Experimenting with various topologies such as Full Network, 3D Grid, Line, and Imperfect 3D Grid.
3. Measuring convergence times across different network sizes for both algorithms.
4. Analyzing the results and drawing conclusions based on the experimental data.

Problem Statement

Write a program to simulate and analyze the convergence of Asynchronous Gossip algorithms in various network topologies. This project explores both information propagation and aggregate computation using the Gossip and Push-Sum algorithms respectively. The solution must be implemented using the Pony programming language, leveraging its actor model to achieve concurrency and simulate large-scale distributed systems.

1. The program should support two main algorithms:
 - 1.1 Gossip Algorithm for information propagation
 - 1.2 Push-Sum Algorithm for sum computation
2. It should also implement four different network topologies:
 - 2.1 Full Network
 - 2.2 3D Grid
 - 2.3 Line
 - 2.4 Imperfect 3D Grid
3. The user will provide three inputs to the program:
 - 3.1 numNodes: The total number of nodes (actors) in the network

3.2 topology: The network topology to be used (full, 3D, line, or imp3D)

3.3 algorithm: The algorithm to be executed (gossip or push-sum)

4. The program should simulate the chosen algorithm on the specified topology and report the convergence time. It should leverage multi-core machines using Pony's actor model to achieve high levels of concurrency and simulate large-scale networks efficiently.

Algorithms

3.1 Gossip Algorithm

The Gossip Algorithm is a widely-used approach for information dissemination in a network. It involves the following steps:

1. Starting: A participant (actor) is sent a rumor by the main process.
2. Step: Each actor selects a random neighbor and tells it the rumor.
3. Termination: Actors stop propagating the rumor after hearing it 10 times.

3.2 Push-Sum Algorithm

The Push-Sum Algorithm is used for distributed sum computation. It involves the following steps:

1. State: Each actor maintains two quantities, s (sum) and w (weight). Initially, $s = x_i = i$ (the actor's ID) and $w = 1$.
2. Starting: One actor is asked to start the process.
3. Receive: Upon receiving a pair (s, w) , the actor updates its own s and w values, then selects a random neighbor to send a message.
4. Send: Each actor sends half of its s and w values to a random neighbor.
5. Sum Estimate: The sum estimate at any given time is s/w .
6. Termination: An actor terminates if the ratio s/w does not change by more than 10^{-10} in three consecutive rounds.

Topologies

The network topology plays a critical role in determining the speed of convergence for both algorithms. The topologies tested in this project are:

1. Full Network: Each actor can communicate with every other actor in the network.
2. 3D Grid: Actors are arranged in a 3D grid, and each actor is connected to its six neighboring nodes (if available).
3. Line: Actors are arranged in a straight line, and each actor has only two neighbors (one on the left and one on the right).

4. Imperfect 3D Grid: Similar to the 3D grid but with an additional random neighbor for each actor.

Implementation Details

The program is implemented in Pony, utilizing its actor model for concurrency. The main components of the implementation are:

1. Main actor: Initializes the simulation based on user input.
2. Network actor: Creates and manages the network of nodes, implements different topologies.
3. Node actor: Represents individual nodes in the network, handles algorithm-specific logic.

Key features of the implementation:

1. Asynchronous communication between nodes using Pony's messaging system.
2. Random neighbor selection for both Gossip and Push-Sum algorithms.
3. Convergence detection and timing measurement.
4. Error handling and bounds checking for robustness.

How to interact with the program

The program is built to be interactive with the user. Once the program is compiled with `ponyc` command, the user can execute the program using the following command `./Project2 1000 line gossip`

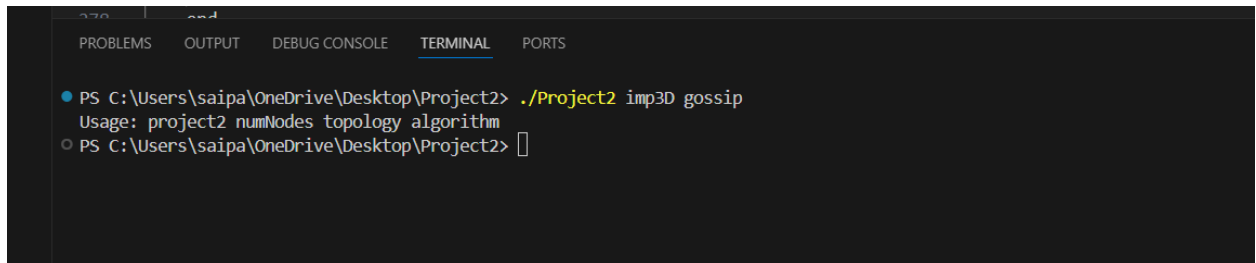
Where, Project2 is the .exe file name, number of nodes is 1000, topology is line and algorithm is gossip.

Working of the program

The program execution starts with the Main actor, which is the starting point of the program in Pony. The constructor named 'create', creates new instances of the actor. The values of number of nodes, topology and algorithm received from the command line argument are stored in an array named 'args'.

First, we check for the invalid scenarios such as:

1. The value of number of nodes or topology or algorithm is missing then the user gets the error mentioning the usage of the code



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\saipa\OneDrive\Desktop\Project2> ./Project2 imp3D gossip
Usage: project2 numNodes topology algorithm
○ PS C:\Users\saipa\OneDrive\Desktop\Project2> 
```

2. The value of number of nodes or topology or algorithm are incorrect, the user gets error “Invalid Input”



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\saipa\OneDrive\Desktop\Project2> ./Project2 1000 imp3D gos
Main Process started
Server requested for Topology imp3D
Server requested for randomization for actors: 1000
Invalid algorithm
gossip actor - Node(220)
○ PS C:\Users\saipa\OneDrive\Desktop\Project2> 
```

This Pony code implements a simulation of Gossip and Push-Sum algorithms across various network topologies. The Main actor initializes the simulation using user-provided arguments: number of nodes, topology type, and algorithm choice. The Network actor then creates the specified number of Node actors and arranges them according to the chosen topology (full, 3D grid, line, or imperfect 3D grid). The simulation begins by randomly selecting a starting node to initiate the chosen algorithm. In the Gossip algorithm, nodes spread a rumor to random neighbors until they've heard it 10 times, while in the Push-Sum algorithm, nodes exchange and update numerical values with neighbors until their ratio converges. Each Node actor operates independently, handling message passing and algorithm-specific logic, showcasing Pony's actor model for concurrency. The nodes communicate asynchronously, selecting random neighbors for interaction. The simulation tracks convergence time, considering the algorithm converged when the first node meets the convergence criteria. Robust error handling and bounds checking are implemented throughout the code to ensure reliability in various scenarios. Once the convergence criteria are met, the simulation terminates and reports the total convergence time. This implementation effectively demonstrates the use of actor-based concurrency to simulate large-scale distributed systems and analyze the performance of gossip-based algorithms in different network structures.

Gossip Algorithm:

Nodes	Full	Line	3D	Imp3D
500	1.34	0.61	1.84	7.86
2500	31.82	7.95	4.17	21.99
4500	177.70	14.33	4.36	32.63
6500	1323.77	26.68	10.50	39.15
8500	3773.81	33.40	7.40	65.99

Push-Sum Algorithm:

Nodes	Full	Line	3D	Imp3D
500	0.32	2.25	0.27	0.27
2500	1.14	11.74	0.57	0.54
4500	1.29	18.56	1.08	1.80
6500	4.88	24.04	2.48	2.26
8500	3.34	30.26	3.23	2.69

These tables show the average convergence time (in milliseconds) for different network sizes and topologies for both the Gossip and Push-Sum algorithms.

For example,

1. **./Project2 10000 line gossip**

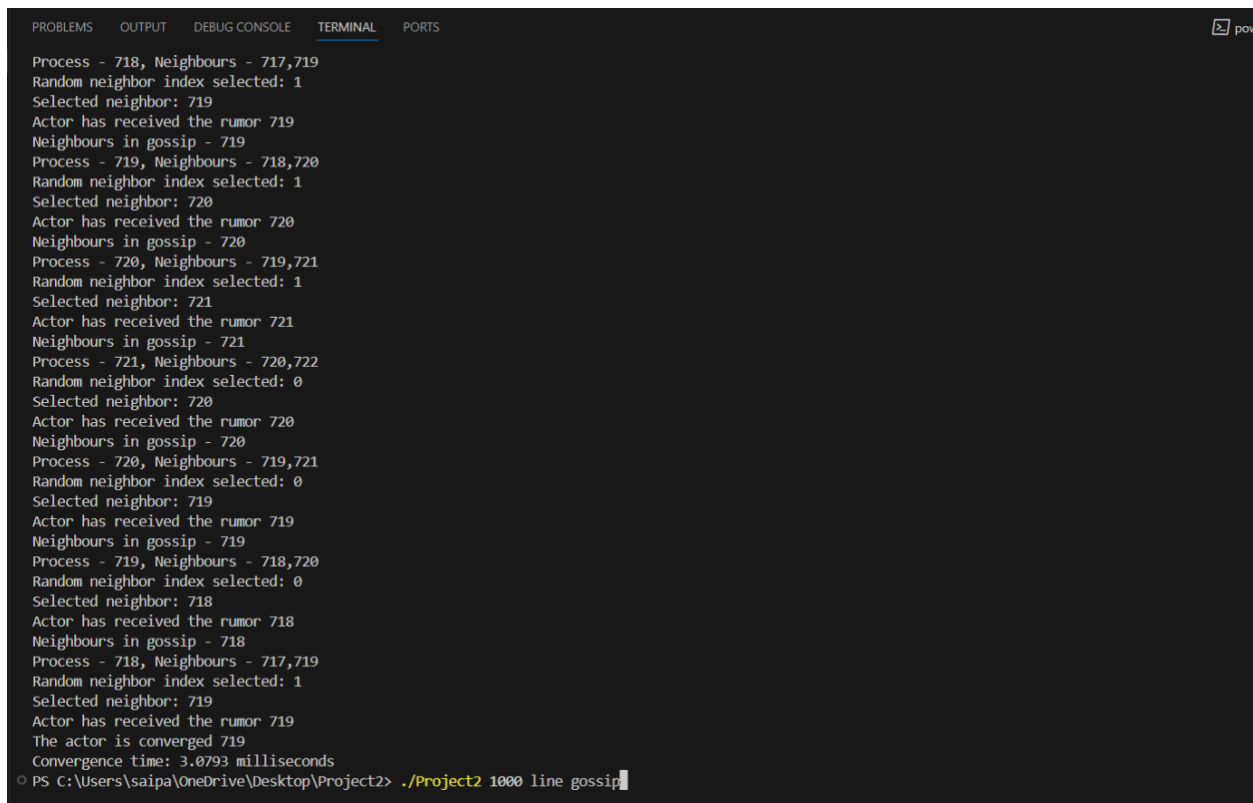
This command would simulate 10,000 nodes in a line topology using the Gossip algorithm. It would call `create_line_topology()` to set up the network, then `start_gossip()` on a random node. Nodes would spread rumors to their immediate neighbors (left and right) using

spread_rumor(). Random node selection is limited to these two neighbors. The simulation continues until a node hears the rumor 10 times, triggering convergence.

2. **./Project2 4000 3D push-sum**

This command simulates 4,000 nodes in a 3D grid topology using the Push-Sum algorithm. It calls create_3d_topology() to arrange nodes in a 3D grid. The simulation starts with start_push_sum() on a random node. Nodes exchange values with neighbors in the 3D grid using push_sum_step(). Random neighbors are selected from the six adjacent nodes in the 3D grid. The algorithm converges when a node's ratio doesn't change significantly for three consecutive rounds.

The Result of Executing Program for ./Project2 10000 line gossip



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Process - 718, Neighbours - 717,719
Random neighbor index selected: 1
Selected neighbor: 719
Actor has received the rumor 719
Neighbours in gossip - 719
Process - 719, Neighbours - 718,720
Random neighbor index selected: 1
Selected neighbor: 720
Actor has received the rumor 720
Neighbours in gossip - 720
Process - 720, Neighbours - 719,721
Random neighbor index selected: 1
Selected neighbor: 721
Actor has received the rumor 721
Neighbours in gossip - 721
Process - 721, Neighbours - 720,722
Random neighbor index selected: 0
Selected neighbor: 720
Actor has received the rumor 720
Neighbours in gossip - 720
Process - 720, Neighbours - 719,721
Random neighbor index selected: 0
Selected neighbor: 719
Actor has received the rumor 719
Neighbours in gossip - 719
Process - 719, Neighbours - 718,720
Random neighbor index selected: 0
Selected neighbor: 718
Actor has received the rumor 718
Neighbours in gossip - 718
Process - 718, Neighbours - 717,719
Random neighbor index selected: 1
Selected neighbor: 719
Actor has received the rumor 719
The actor is converged 719
Convergence time: 3.0793 milliseconds
PS C:\Users\saipa\OneDrive\Desktop\Project2> ./Project2 10000 line gossip
```

The convergence time of executing program for ./Project2 10000 line gossip is 3.0793 milliseconds.

The Result of Executing Program for ./Project2 4000 3D push-sum

```
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 1.03077e-09
S is 8.5187e+06
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 6.28347e-10
S is 1.09105e+07
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 3.83068e-10
S is 1.3974e+07
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 2.33513e-10
S is 1.78975e+07
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 1.42336e-10
S is 2.29227e+07
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 8.67715e-11
S is 2.93589e+07
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 5.28928e-11
S is 3.76022e+07
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 3.22871e-11
S is 4.816e+07
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 1.96394e-11
S is 6.16822e+07
The old ratio is: 193.438
The new ratio is: 193.438
Difference is: 1.19655e-11
Convergence achieved for 192
Convergence time: 23.1755 milliseconds
PS C:\Users\saipa\OneDrive\Desktop\Project2> █
```

The convergence time of executing program for ./Project2 4000 3D gossip is 23.1755 milliseconds.

Largest Executed Problem

The largest problem we were able to execute was

Topology	Algorithm	Nodes	Convergence Time (ms)
Line	Gossip	1,000,000	58,901.3
Full	Gossip	50,000	67,294.2
3D	Gossip	600,000	8,378.29
Imperfect 3D	Gossip	500,000	93,298.2
Line	Push-Sum	100,000	11,291.1
Full	Push-Sum	100,000	1,007.27
3D	Push-Sum	3,000,000	35,708.8
Imperfect 3D	Push-Sum	3,000,000	162156

GOSSIP

LINE - 1000000

```
Process - 110687, Neighbours - 110686,110688
Random neighbor index selected: 1
Selected neighbor: 110688
Actor has received the rumor 110688
Neighbours in gossip - 110688
Process - 110688, Neighbours - 110687,110689
Random neighbor index selected: 0
Selected neighbor: 110687
Actor has received the rumor 110687
The actor is converged 110687
Convergence time: 58901.3 milliseconds
```

FULL - 50000

```
Selected neighbor: 32785
Actor has received the rumor 32785
Neighbours in gossip - 32785
Random neighbor index selected: 47849
Selected neighbor: 47850
Actor has received the rumor 47850
Neighbours in gossip - 47850
Random neighbor index selected: 32473
Selected neighbor: 32473
Actor has received the rumor 32473
The actor is converged 32473
Convergence time: 67294.2 milliseconds
PS C:\Users\saipa\OneDrive\Desktop\Project2> █
```

3D - 600000

```
Selected neighbor: 443783
Actor has received the rumor 443783
Neighbours in gossip - 443783
Random neighbor index selected: 4
Selected neighbor: 436558
Actor has received the rumor 436558
Neighbours in gossip - 436558
Random neighbor index selected: 1
Selected neighbor: 436559
Actor has received the rumor 436559
The actor is converged 436559
Convergence time: 8378.29 milliseconds
█
```

Imperfect 3D - 500000

```
Neighbours in gossip - 88718
Random neighbor index selected: 1
Selected neighbor: 88719
Actor has received the rumor 88719
Neighbours in gossip - 88719
Random neighbor index selected: 2
Selected neighbor: 88799
Actor has received the rumor 88799
The actor is converged 88799
Convergence time: 93898.2 milliseconds
PS C:\Users\saipa\OneDrive\Desktop\Project2> █
```

PUSH SUM

Line- 300000

```
(base) arnav@Arnavs-MacBook-Pro project2 % ./project2 3000000 line push-sum
Main Process started
Server requested for Topology line
Server requested for randomization for actors: 3000000
gossip actor - Node(820279)
Convergence achieved for 0
Convergence time: 11291.1 milliseconds

(base) arnav@Arnavs-MacBook-Pro project2 % █
```

Full - 100000

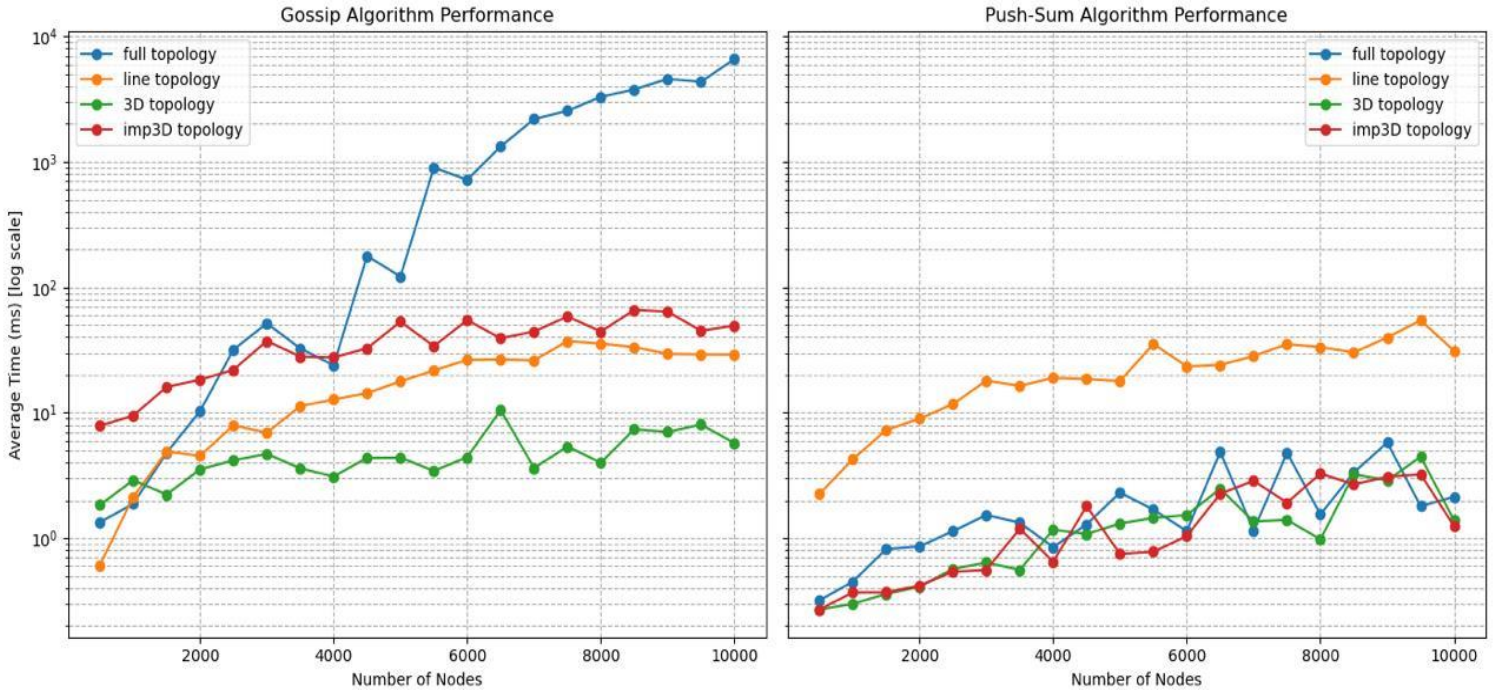
```
(base) arnav@Arnavs-MacBook-Pro project2 % ./project2 100000 full push-sum
Main Process started
Server requested for Topology full
Server requested for randomization for actors: 100000
gossip actor - Node(83133)
Convergence achieved for 5
Convergence time: 1007.27 milliseconds
(base) arnav@Arnavs-MacBook-Pro project2 % █
```

3D-3000000

```
(base) arnav@Arnavs-MacBook-Pro project2 % ./project2 3000000 3D push-sum
Main Process started
Server requested for Topology 3D
Server requested for randomization for actors: 3000000
gossip actor - Node(2180204)
Convergence achieved for 2180076
Convergence time: 35708.8 milliseconds
```

Imperfect 3D - 3000000

```
(base) arnav@Arnavs-MacBook-Pro project2 % ./project2 3000000 imp3D push-sum
Main Process started
Server requested for Topology imp3D
Server requested for randomization for actors: 3000000
gossip actor - Node(2995753)
Convergence achieved for 2995700
Convergence time: 162156 milliseconds
```



6.1 Gossip Algorithm

- Full Topology: The poor performance of the full network can be attributed to the high communication overhead. As every node is connected to every other node, the number of messages increases quadratically with the number of nodes, leading to slower convergence.
- 2 Line Topology: The linear arrangement limits communication to two neighbors, which reduces overhead but also slows down the propagation of rumors. This leads to a more predictable, but slower, convergence time.
- 3D Topology: The structured grid of the 3D topology provides an optimal balance between connectivity and communication overhead. This is why it shows the fastest convergence times.
- Imperfect 3D Topology: The additional random neighbor introduces some variability in convergence time, but it still performs better than the full topology due to the underlying grid structure.

6.2 Push-Sum Algorithm

1. Full Topology: The push-sum algorithm struggles with the full topology due to the high communication overhead. The need for each node to communicate with every other node introduces significant delays.

2. Line Topology: Push-sum performs worse in this topology because the algorithm requires stable communication to compute sums. The linear arrangement slows down convergence significantly.
3. 3D Topology: As with gossip, the 3D topology provides an efficient structure for message passing, allowing for faster and more consistent convergence.
4. Imperfect 3D Topology: The added randomness slightly increases the communication cost, but the performance remains close to that of the 3D grid.

Additional:

1. We also implemented a `run_experiments.sh` shell script to automate the running and averaging of the different topology and propagation methods with different network sizes and averaged them out. Make the `.sh` an executable by `chmod +x run_experiments.sh` and then `./run_experiments.sh` to automate.
2. The `graphing.py` plots the gossip and push-sum algos, 4 of them together on a logarithmic scale.

What is Working

3. Both Gossip and Push-Sum algorithms were implemented successfully.
4. Simulations were run across all four topologies: Full, Line, 3D Grid, and Imperfect 3D Grid.
5. Performance was measured and plotted for each algorithm and topology.
6. We averaged 10 runs for each configuration to ensure accurate results.

Conclusion

This project explored the performance of the Gossip and Push-Sum algorithms across various network topologies using Pony's actor model. Key conclusions include:

1. Full Network: Worst performance for both algorithms due to high communication overhead.
2. Line Topology: Predictable but slow convergence due to limited communication between nodes.
3. 3D Grid: The best performance overall, providing a good balance between communication efficiency and connectivity.
4. Imperfect 3D Grid: Introduces variability but performs significantly better than the full network.