

## **Distributed Operating System Principle**

**Manish Yadav  
UF ID: 3836-6483  
m.yadav@ufl.edu**

**Vasu Negi  
UF ID: 8495-3933  
vasu.negi@ufl.edu**

Department of Computer and Information Science and Engineering  
University of Florida

## Table of Contents

- Table of Contents
- Build Process
- What is Working
  - Gossip
    - \* Normal Scale
    - \* Log Scale
  - PushSum
    - \* Linear Scale
    - \* Logarithmic Scale
- What is the largest network you managed to deal with for each type of topology and algorithm
  - Gossip 10k
    - \* Normal Scale 10k
    - \* Log Scale 10k
  - PushSum 10k
    - \* Linear Scale 10k
    - \* Logarithmic Scale 10k
- Interesting finds

## Build Process

- unzip the compressed file using `unzip filename.zip`
- `dotnet fsi --langversion:preview proj2.fsx nodeNum topology protocol` to run script where `nodeNum` is the number of nodes you want to run topology for. `topology` can have values in `[line, full, 2D, Imp2D]`. The protocol can have values either `gossip` or `push-sum`.

## What is Working

We able to run all `line`, `full`, `2D`, and `Imp2D` in any combination with `gossip` or `push-sum` protocol. The convergence in Gossip protocol is achieved when all the nodes have converged. A node is set to be converged when it listens to the message for the 10th time. After convergence, the node stops transmitting the message to its neighbor. Once the network is converged i.e. all nodes are converged, the total time for convergence is printed out.

## Gossip

### Normal Scale

- Line topology is the slowest to converge. This is due to the fact that it has only access to 2 neighbors (left and right node).
- Full topology is the fastest to converge in all possible scenarios. This is because it is connected to all the nodes and convergence is faster to achieve in this scenario

- As expected, 2D and imperfect 2D would achieve the convergence time in between the line and full with imperfect 2D performing slightly better or equal to 2D performance.

**Note:** All nodes weren't converging, on multiple runs we would notice that the convergence rate of topology would be 80-90%. This is because of the fact that the structure breaks while converging and some nodes are unreachable. In order to tackle this problem, we keep a track of all nodes that aren't converged, select one non converged node at random and keep sending messages till the time topology doesn't achieve the convergence.

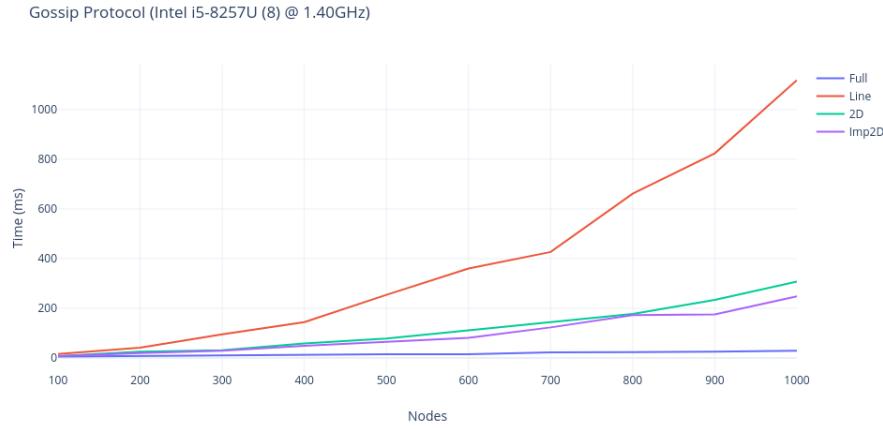


Figure 1: Gossip

## Log Scale

### PushSum

The PushSum network works by sending message  $s$  and  $w$  as parameter to an actor. The initial value of  $s$  is equal to the index of the actor and  $w = 1$ . The propagation stops when an actor's  $s/w$  ratio does not change for 3 times in a row (i.e stays within limit of  $10^{-10}$ )

**Note:** Unlike gossip algorithms, the push sum algorithm is able to converge in all topologies. We believe this is due to the fact we try to reduce the value  $s/w$  ratio till it does stop changing the ratio for consecutive three times. This allows in more messages being transmitted compared to the gossip algorithm allowing the network to converge. This also causes an increase in convergence time.

## Linear Scale

Gossip Protocol (Intel i5-8257U (8) @ 1.40GHz) - Log Scale

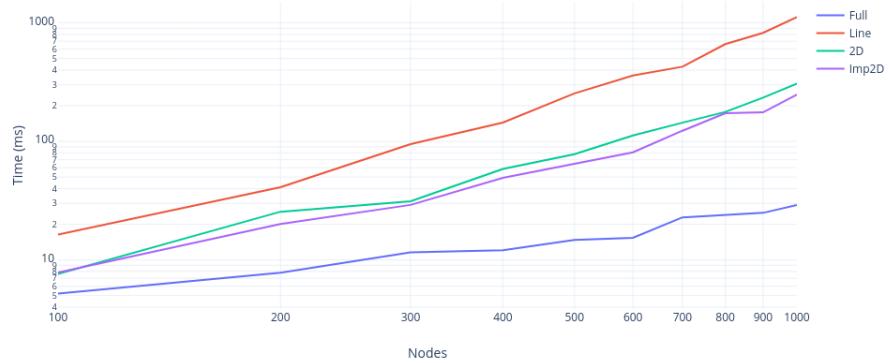


Figure 2: Gossip Log Scale

PushSum Protocol (Intel i5-8257U (8) @ 1.40GHz)

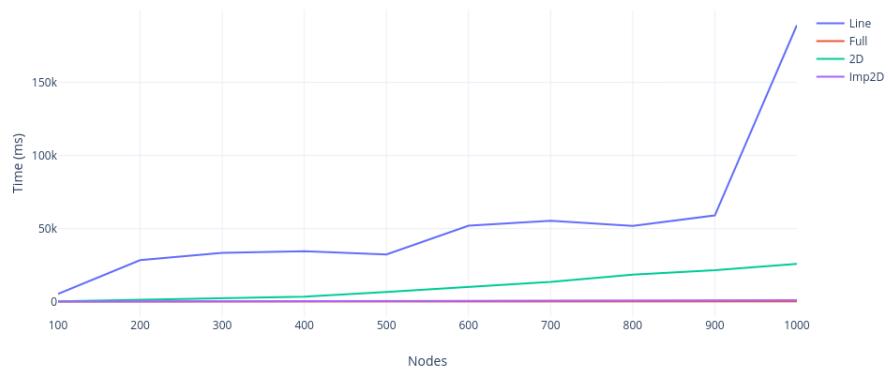


Figure 3: PushSum

PushSum Protocol (Intel i5-8257U (8) @ 1.40GHz) - Log Scale

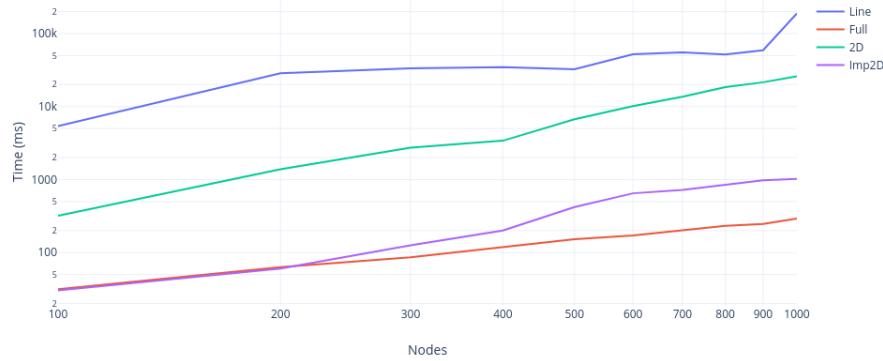


Figure 4: PushSum Log Scale

### Logarithmic Scale

#### What is the largest network you managed to deal with for each type of topology and algorithm

The largest network that we have managed to solve is 10k nodes for each topology and algorithm. We decided to limit it to 10k node so that the network could be compared. Notice that other topologies (full, 2D, and imp2D) could run on larger nodes within a reasonable time frame.

The following are the result of running the topologies in different combinations.

#### Gossip 10k

#### Normal Scale 10k

#### Log Scale 10k

#### PushSum 10k

#### Linear Scale 10k

#### Logarithmic Scale 10k

### 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

Gossip Protocol (Intel i5-8257U (8) @ 1.40GHz)

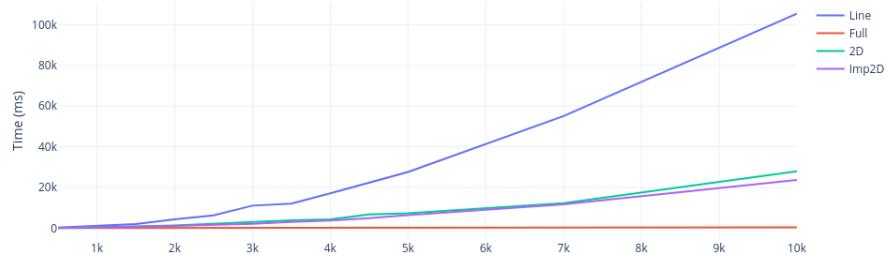


Figure 5: Gossip 10K

Gossip Protocol (Intel i5-8257U (8) @ 1.40GHz) - Log Scale

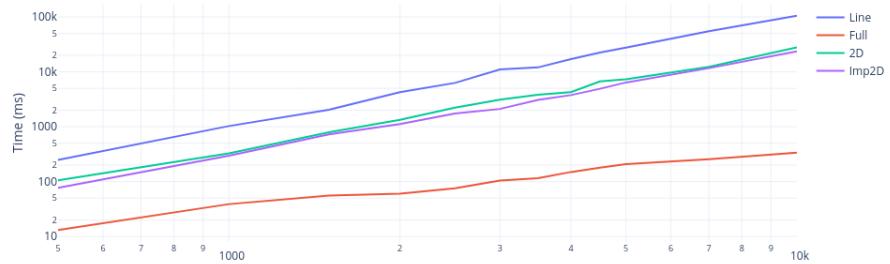


Figure 6: Gossip 10K Log Scale

PushSum Protocol (Intel i5-8257U (8) @ 1.40GHz)

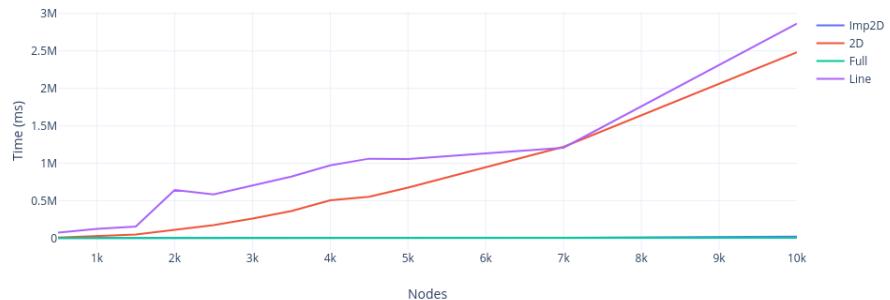


Figure 7: PushSum 10K



Figure 8: PushSum 10K Log Scale

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`)  $10^{-10}$  to  $10^{-5}$ , 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.