**COP5615 Project 2 ReadMe: Gossip Algorithm**

**Group Members:**

* Venkata Ramana Pamarthy, UFID: 76176980
* Sreeram Aditya Potukuchi, UFID: 31395861

**What is working?**

* We have successfully implemented all 4 topologies namely Line, Full, 2D and Imperfect 3D across both Gossip and Pushsum algorithms.
* Achieved 100% convergence for all 4 topologies in Gossip and Pushsum simulation. (i.e., all nodes have received messages ‘N’ number of times, where N is the gossip limit taken as user input).
* In the Gossip simulation, out of the 4 topologies, Line topology was observed to be the slowest to converge and the Full topology was the quickest to converge.
* In the Pushsum simulation, out of the 4 topolgies, again Full topology was observed to be the best performing one, followed by Imperfect 3D and 2D, while Line topology was the slowest to converge.
* Pushsum was observed to be slower to converge than Gossip simulation for all topologies
* Able to scale the number of nodes to 10000 for each topology without any issues with the system limits for Gossip and 8000 for each topology in pushsum except for line, as it converges slowly.
* Also able to play with the Gossip Count (i.e., number of times an actor receives the gossip message) by taking input from the user during program execution.

**Largest Network for each Topology:**

* Based on our implementation and machine specifications (Dell 8GB ram, Intel i7 8th gen) , we observed the following largest networks for each topology and algorithm.

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Topology** | **Max Nodes** |
| Gossip | Line | **10000** |
| Gossip | 2D | **10000** |
| Gossip | Imperfect 3D | **10000** |
| Gossip | Full | **10000** |
| Pushsum | Line | **5000** |
| Pushsum | 2D | **8000** |
| Pushsum | Imperfect 3D | **8000** |
| Pushsum | Full | **8000** |

**Output for nodecount = 10000 in Gossip:**

**Shape

Description automatically generated with medium confidence**

**Output for nodecount = 8000 in pushsum**

**Shape

Description automatically generated with medium confidence**

**Steps to Execute the Program:**

The project is divided into two sub-projects: One for the main project and the other for bonus.

**For main project:**

* In the main project, we have three files, one for gossip simulation, one for pushsum simulation and the last one to compute neighbors based on topologies.
* To compile, run the following commands in command prompt

*c(main).*

*c(gossip\_main).*

*c(pushsum\_main).*

* To execute both gossip and pushsum algorithm, run the following commands

main:start(arg1, arg2, arg3, arg4)

where,

arg1 = number of nodes

arg2 = topology (one of *line*, *2d*, *imp3d*, *full*)

arg3 = algorithm (either *gossip* or *pushsum*)

arg4 = gossip count (number of messages each actor receives before converging)

For gossip:

Text

Description automatically generated

For pushsum:

* The program prints each process when it is converged and at the end prints convergence time in milliseconds.

For gossip:

Shape

Description automatically generated with low confidence

For pushsum:

Text

Description automatically generated

**For bonus project:**

* In the bonus project, we have three files, one for gossip simulation, one for pushsum simulation and the last one to compute neighbors based on topologies.
* To compile, run the following commands in command prompt

*c(main\_bonus).*

*c(gossip\_bonus).*

*c(pushsum\_bonus).*

* To execute both gossip and pushsum algorithm, run the following commands

main\_bonus:start(arg1, arg2, arg3, arg4, arg5)

where,

arg1 = number of nodes

arg2 = topology (one of *line*, *2d*, *imp3d*, *full*)

arg3 = algorithm (either *gossip* or *pushsum*)

arg4 = gossip count (number of messages each actor receives before converging)

arg5 = fault count (number of nodes to kill)

* The program prints each process when it is converged and at the end prints convergence time in milliseconds.
* We then request for the number of processes that haven’t converged to show that there has been a change in convergence when inducing fault

For gossip bonus:

Text

Description automatically generated

For pushsum bonus:

Text

Description automatically generated