**COP5615 Project 2 Report: Gossip Algorithm**

**Group Members:**

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

**Topologies:**

1. Line Topology: In this topology, the nodes are lined up linearly and only the first and last nodes have one neighbor while the rest have 2 (index-1 and index+1). So max number of neighbors = 2 and min = 1
2. 2D Topology: In this topology, the nodes are formed like a grid in a 2-dimension plane. So max number of neighbors = 3 and min = 2 (as the corner nodes in the square have 2 neighbors only and the rest of the nodes have 3).
3. Imperfect 3D: In this topology, the nodes are formed like a grid in a 3-dimension plane, with 1 additional random neighbor i.e., 8+1 possible neighbors in total. So, max number of neighbors = 9 and min = 4.
4. Full: Full topology is where all nodes are spread out into the space and neighbors are chosen at random. Each node in this topology is a neighbor to one another.

**Gossip Simulation:**

* In this simulation, the main module takes the 4 inputs namely NumNodes, Topology, Algorithm and GossipsCount.
* On receiving these inputs, a supervisor is spawned which then creates multiple actors.
* Then a first actor is chosen by supervisor at random to initiate a gossip. This actor then sends the gossip message to its neighbors who are computed based on the topology input given.
* This process of spreading the gossip continues until all the processes send out specified number of gossips (given as an input GossipsCount).
* At the end, these processes after completing the spread (of number of gossips) print the convergence message and exit from the program.
* The final convergence time is given as an output which is the total time taken for all the processes to send out N gossip messages.

**Observations in Gossip simulation:**

* On executing the gossip algorithm, we observed that the Full topology converges the quickest followed by Imperfect3D, 2D and Line topologies.

i.e., best = Full, worst = Line

* The below graph shows how each of the topologies perform for the Gossip simulation:
* For 1000 nodes:

For 10000 nodes:

* We can clearly conclude from the above graphs that, the Full topology works the best and converges quicker than any other topology, followed by imp3d, 2d and the line topology is the slowest performing topology to converge.

**Note**: Two plots are shown here just to show the minor difference in the graph spread between 1000 and 10000 nodes

**PushSum Simulation:**

* In this simulation, the main module takes the 4 inputs namely NumNodes, Topology, Algorithm and GossipsCount, but the gossip count is not considered in the program
* On receiving these inputs, a supervisor is spawned which then creates multiple actors. Each actor has their own sum and weights assigned. Sum = i which is the index value of the node while weight = 1 commonly for all nodes.
* Then a first actor is chosen by supervisor at random to initiate a pushsum. This actor then sends the {sum/2,weight/2} pair message to its neighbors who are computed based on the topology input given. Then the actor halfs their sum and weight pairs too.
* This process of spreading the pushsum continues until the difference between any three consecutive iterations is less than 10-10. This is where the pushsum converges
* At the end, these processes after completing the spread, print the convergence message and exit from the program.
* The final convergence time is given as an output which is the total time taken for all the processes to send out the sum,weight pairs until difference between 3 consecutive iterations for each actor is 10-10.

**Observations for Pushsum:**

* On executing the pushsum algorithm, we observed that the Full topology converges the quickest followed by Imperfect3D, 2D and Line topologies.

i.e., best = Full, worst = Line

* The below graph shows how each of the topologies perform for the Gossip simulation:
* For 1000 nodes:
  + For 10000 nodes:

**Note**: Two plots are shown here just to show the minor difference in the graph spread between 1000 and 10000 nodes.