Team Members:

1) Kiran Kumar Reddy Gorantla – 9383 3393

2) Yashaswi Karnati- 1993-0168.

Instructions to run the program:

1) Navigate to project folder

2) mix compile

3) mix escript.build

4) ./gossip_pushsum numnodes topology algorithm (Ex: ./gossip_pushsum 50 line gossip)

Topologies Implemented:

1) Line

2) Full

3) Random 2D Grid

4) 3D torus

5) Honeycomb

6) Random Honeycomb

Algorithms Implemented:

1) Gossip

2) Push Sum

Observations

1) Gossip

Line: This topology takes the largest time to converge out of all.

Full: This topology takes relatively less time to converge as every node is connected to every other node. Time taken to converge increases with increase in number of nodes as size of the network increases.

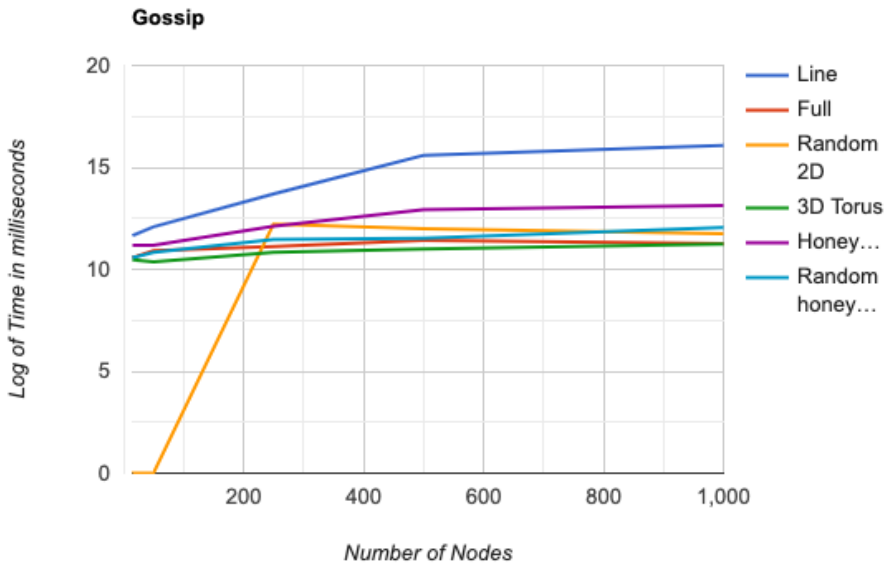
Random 2D Grid: Time taken by this topology to converge varies as the actors are connected randomly. Sometimes this topology may not converge for smaller networks. For each node, number of neighbors is directly proportional to size of network. So, time taken to converge gets better as number of nodes increases.

Torus: This topology takes relatively less time to converge. Time taken to converge increases with increase in number of nodes as size of the network increases.

HoneyComb: This topology outperforms line but full, 3D torus topologies converge faster.

Random Honeycomb: This topology takes relatively very less time when compared to honeycomb. Each node in this topology has an extra connection to random node in the entire network.

The below figure shows Log of Time taken to converge vs Size of network for different topologies.



2) Push Sum

Line: Unlike gossip, pushsum for line takes less time to converge. It also outperforms honeycomb for larger networks

Full: This topology takes relatively less time to converge as every node is connected to every other node. Time taken to converge increases with increase in number of nodes as size of the network increases.

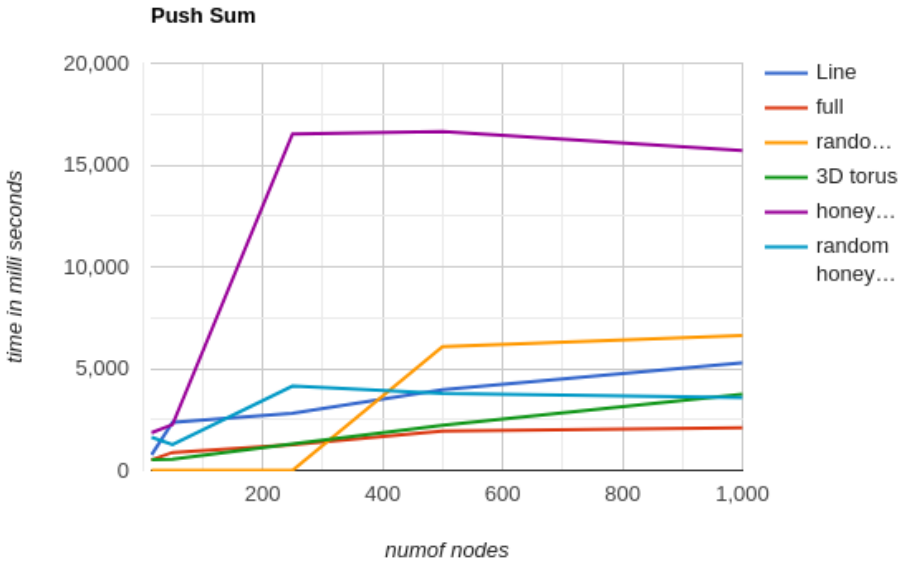
Random 2D grid: Time taken by this topology to converge varies as the actors are connected randomly. Sometimes this topology may not converge for smaller networks. For each node, number of neighbors is directly proportional to size of network. So, time taken to converge gets better as number of nodes increases.

Torus: This topology takes relatively less time to converge. Time taken to converge increases with increase in number of nodes as size of the network increases.

Honeycomb: This topology takes highest time to converge for larger networks.

Random Honeycomb: This topology takes relatively very less time when compared to honeycomb. Each node in this topology has an extra connection to random node in the entire network.

The below figure shows Time taken to converge vs Size of network for different topologies.


Push Sum
20,000
15,000
10,000
5,000
0
200
400
600
800
1,000
time in milli seconds
numof nodes
Line
full
rando…
3D torus
honey…
random honey…