PROJECT REPORT

GOSSIP SIMULATOR

Team Members:
Arunima Agarwal (UFID: 3397-1331)
Karan Acharekar (UFID: 3868-9483)

Description:
The purpose of this project is to use Elixir and the actor model to build a gossip simulator which can be used both for group communication and aggregate computation and hence, determine the convergence of both the algorithms as mentioned. The input provided will consist of numNodes, topology and algorithm.

NumNodes is the number of nodes used to create the network for gossip simulation.

Topology is the topology to be built using the number of nodes. The four types of topology used is:

1) Full Topology
2) 2D Topology
3) Imperfect2D Topology
4) Line Topology

Algorithm is:

1) Gossip
2) Push-Sum

Convergence Assumptions:
For the convergence condition in both gossip and push-sum algorithm we have created a polling process that checks the state of each node that stops sending rumor and if the state remains same for some consecutive number of times then the network should converge as it is the stage that the nodes are not receiving any more messages and neither it can send any messages to other nodes.

Gossip Algorithm:

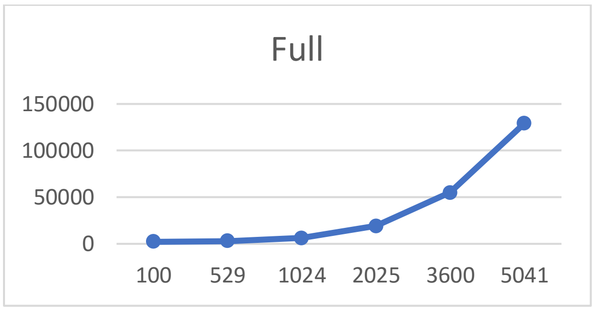
There are four topologies defined in the project full, 2D, imperfect2D and line. We create the neighbor list of each node per the requirement of the topology. We maintain the state of each node which consists of the neighbor list for each node. This neighbor list is created when the GenServer starts the actors and thus calls the build topology for that actor. In this build topology, the neighbor list for each actor is updated in the state.

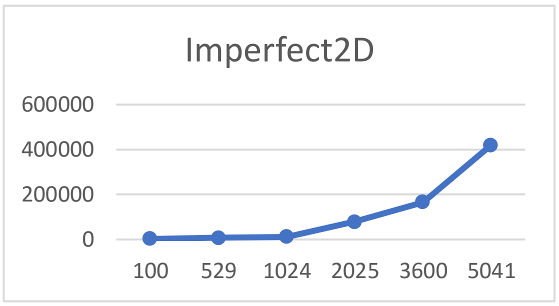
1. The main process send a rumor to any random node from the total nodes passed as the arguments.
2. The random process suppose node4 checks its neighbor list and select random neighbor and pass the rumor to it. This random neighbor check its neighbor list and pass the rumor and it goes on.
3. Each node maintains a rumor count, as soon as it hears the rumor it increases its count.
4. The node stops sending the rumor once the count reaches 10.
5. So, in each topology once the nodes stop sending i.e. kill their process after their count becomes 10 or the nodes has no neighbors left in the neighbor list, hence at this moment the network will converge and we will determine the convergence time.

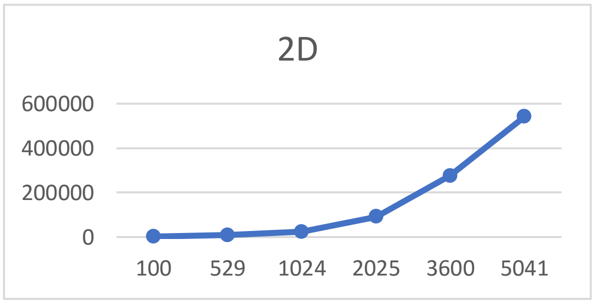
The table representing number of nodes and their convergence time is as below:

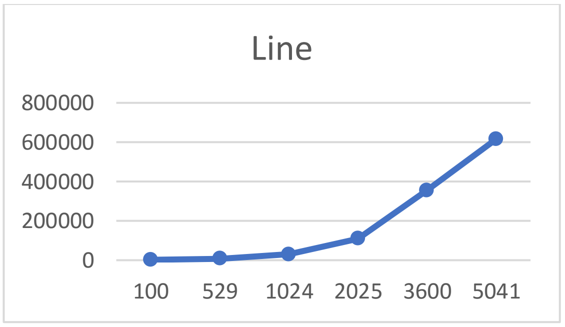
	100	**529**	**1024**	**2025**	**3600**	**5041**
Full	2093	3136	6346	19259	55011	129167
Imperfect2D	2211	6756	11402	78068	166429	419816
2D	2258	8206	23094	91286	275553	540535
Line	2268	8345	29530	109900	354842	614911

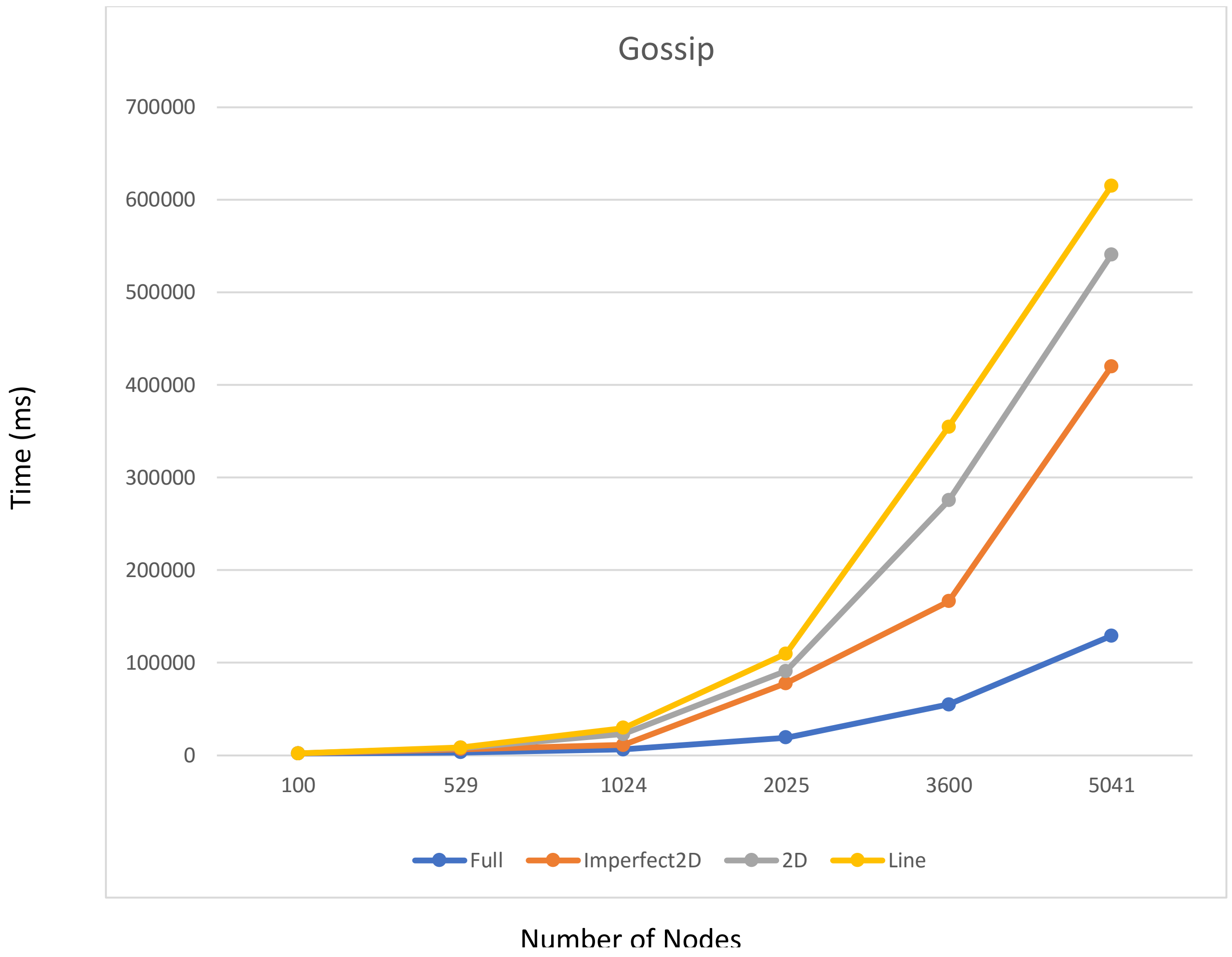
The Graph plotted with the above values is as follows:











Push-Sum Algorithm:

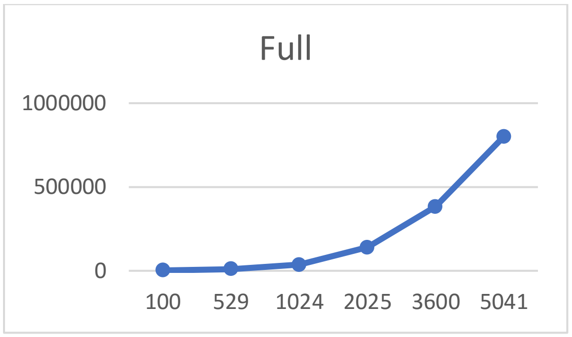
This also should work on the four topologies as defined above. This is used to compute the aggregate which is the ratio of sum/weight. Initially each actor will have weight of 1 and sum equals to its node number.

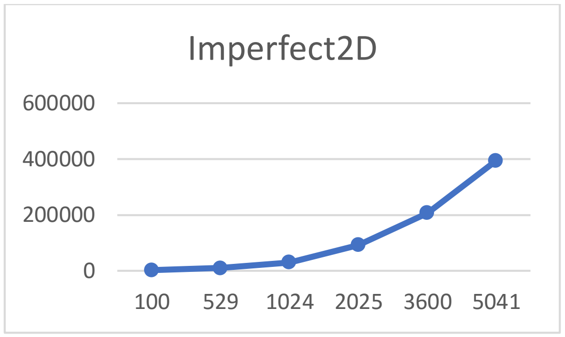
1. The main process starts one of the node to start the push-sum algorithm.
2. This node will have its sum and weight as initialized earlier. It selects some random neighbor from its neighbor list and sends half of its sum and weight to the next node and so on.
3. The sum estimate is the ratio of sum/weight which is calculate for each node and update it as every node sends the message.
4. If the ratio computed above did not change more than 10^-10 in 3 consecutive rounds, the node will terminate that stops its process. Hence, at this moment the network will converge.

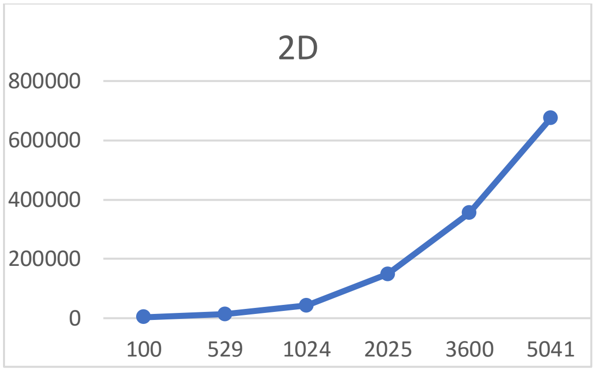
The table representing number of nodes and their convergence time is as below:

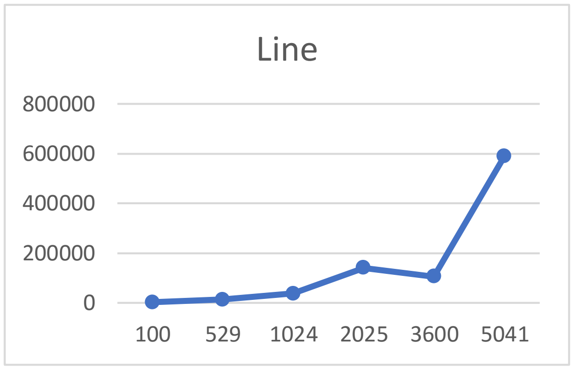
	100	529	1024	2025	3600	5041
Full	2346	10686	35441	139985	383034	800827
Imperfect2D	2453	9858	29846	92552	207627	393807
2D	2847	12765	42669	148558	354104	673807
Line	2534	12842	37164	140884	105899	590265

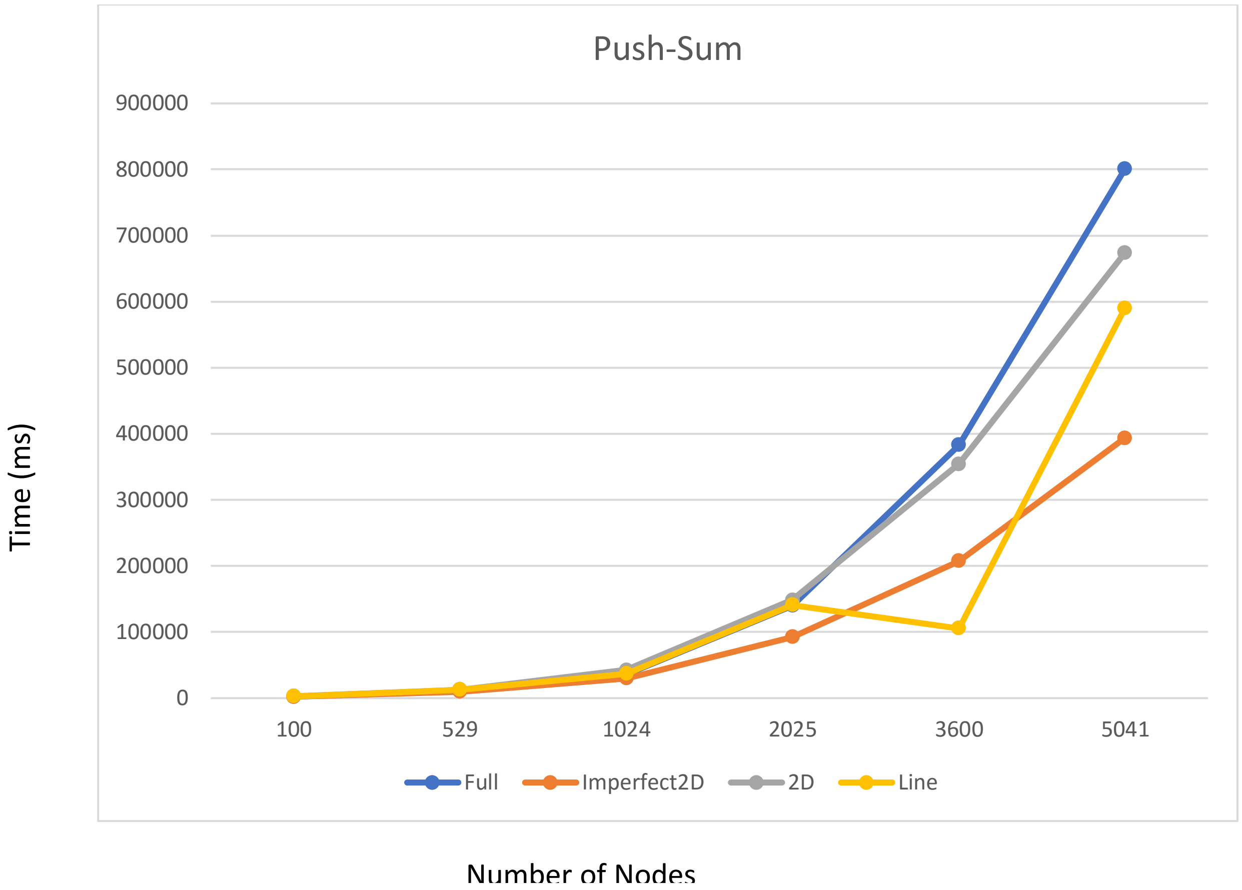
The Graph plotted with the above values is as follows:











Observations:

1) In the Gossip Algorithm, we have run the code for multiple number of actors i.e. nodes and have computed the time for each topology. It was observed that the full topology converges in less time as compared to other topologies while, the line takes the most time to converge. So, the order of each topology per the converging time is **full < imp2D < 2D < line**.
2) In the Push-Sum Algorithm, we have run the code for multiple number of actors i.e. nodes and have computed the time for each topology. It was observed that the imp2D converges in less time as compared to other topologies while, full takes the most time to converge. So, the order of each topology per the converging time is **imp2D < line < 2D < full.**
3) As we have observed from the two algorithms implemented the gossip algorithm is faster as compared to the Push-Sum algorithm. **E.g.** For node count 100 gossip takes less time to converge as compared to that of push-sum.