COP5615 – Fall 2017

Project 2 – Gossip Simulator

Debarshi Mitra – 33813136
Aisharjya Sarkar – 44955999

What is Working:

Gossip Protocol:

We have used GenServer module of Elixir which initiates a main process. The main process then spawns number of actors as specified by the input parameter. All the actors are saved in a centralized list along with its pid. The topology and algorithm is also taken as user input. The main process initiates the gossip protocol by sending a rumor to any single random node. For each actor, the neighbor is calculated based on the input topology on the fly. Upon receiving the rumor, each transmits the rumor to any of its one neighbor, selected randomly, as obtained from the topology. Simultaneously, each actor transmits the rumor periodically to one of its random neighbors. In this manner, the rumor is spread through the network. Thus, in case of failure of one route, where all the neighbors of a node dies, the rumor will proceed through another path and reach other nodes in the network. We have maintained a convergence value at 75%. That is, when 75% of the nodes receive the rumor, the system shuts down.
The Gossip protocol is executed on four different network topologies as full, line, 2D and Imperfect 2D varying the number of nodes. The table below shows all the experiments that we were able to handle in acceptable time, given the number of nodes.

For Gossip, the full network topology gives the best results for all node size, followed by imperfect 2D and 2D.

Push-Sum Protocol:

The GenServer part is common for both gossip as well as push-sum in terms of main process initiation. Then the main process initiates the transmission process by forwarding an initial [s, w]

pair where s is set to 0. The value of s indicates the node id, in our case which starts from 0. Upon receiving, the actor adds up the s and w with its existing s and w, and then forwards s/2 and w/2 to any of its random neighbor. It keeps the other half to itself. The neighbors for an actor is build based on the input topology on the fly. In this manner, when the actor ratio does not change by more than 10^{-10} time for three consecutive times, the actor terminates. Further, once an actor terminates, it simply forwards a message unmodified to any random neighbor for the rest of the

time. Here too, we have maintained a convergence criteria of 75%. We performed push-sum for the four mentioned topologies.

For Pushsum, 2D gives the best results, followed by imperfect 2D and full, for all node size.

System Specification: RAM – 8GB, Processor – 4 Core, Virtual Machine – Ubuntu 16.04

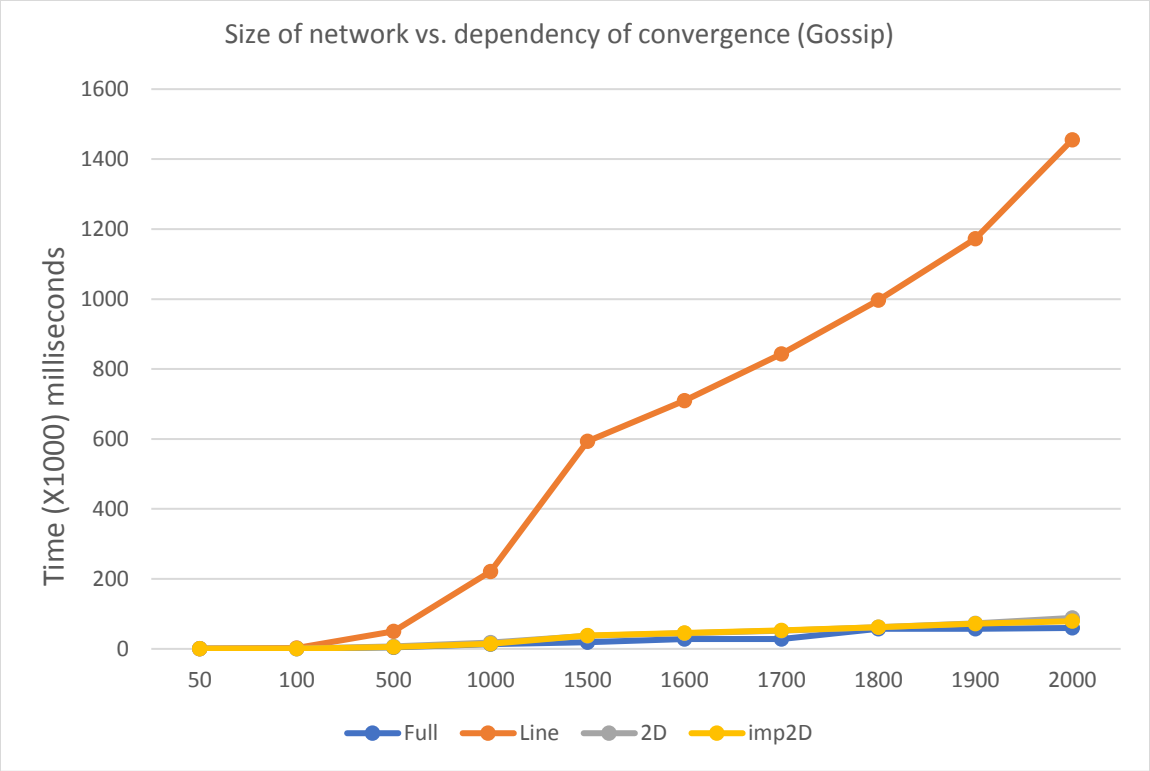
Largest network on which result was generated:

Algorithm	Full Network	Line	2D Grid	Imperfect 2D Grid
Gossip	2000	2000	2000	2000
Push-Sum	10000	3000	10000	10000

Gossip Algorithm:

Table below showing convergence time in milliseconds for Full, Line, 2D and Imperfect 2D Topology with respect to the number of nodes taken as input:

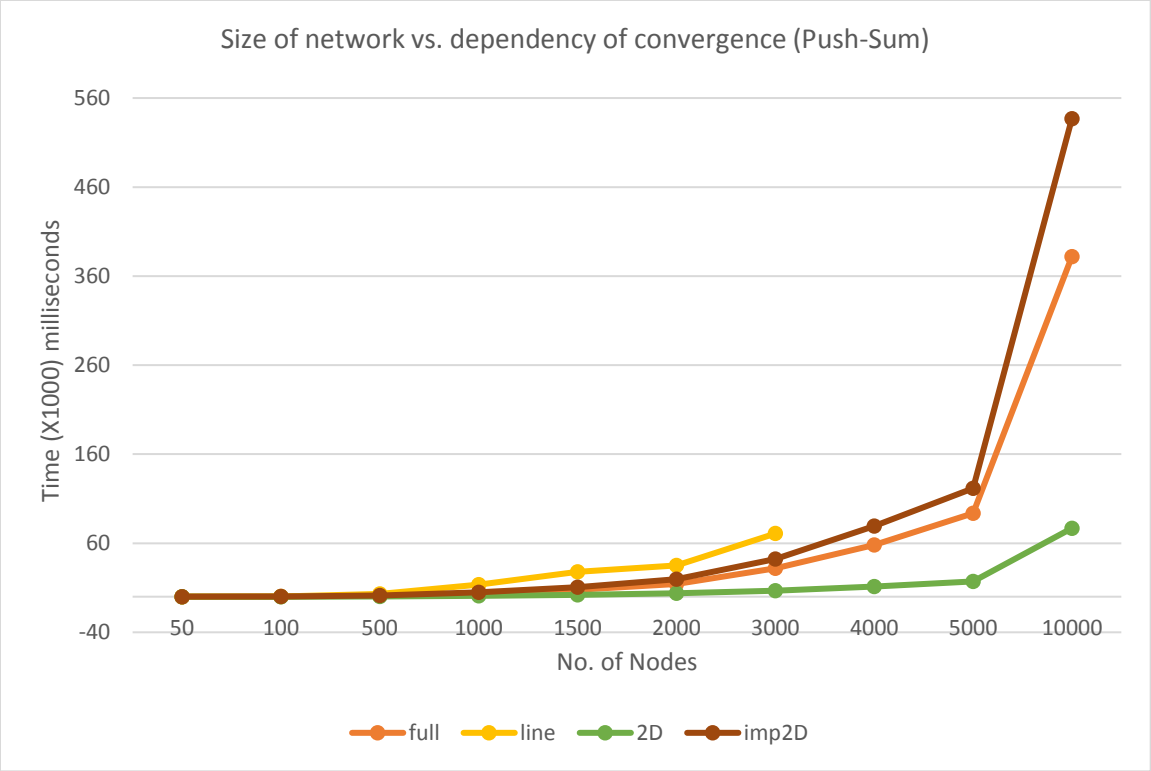
No. of Nodes	Full	Line	2D	Imp2D
50	476	619	522	527
100	882	2264	1092	883
500	4689	49973	7214	4801
1000	13711	220684	18406	14835
1500	18602	593225	37596	38339
1600	28093	709416	44597	45253
1700	28286	843280	52122	53143
1800	57333	996892	61789	62104
1900	57459	1172471	72988	72239
2000	59740	1455108	88499	79002



Push-sum Algorithm:

Table below showing convergence time in milliseconds for Full, Line, 2D and Imperfect 2D Topology with respect to the number of nodes taken as input:

No. of Nodes	full	line	2D	Imp2D
50	45	25	23	37
100	68	195	40	79
500	960	3073	348	1311
1000	3674	13689	843	4754
1500	7697	27939	1862	10540
2000	14200	34960	3881	19612
3000	31717	70846	6588	42439
4000	57920		11404	79115
5000	93772		17260	121653
10000	381927		76873	536831



Interesting Findings and Conclusion:

Gossip Algorithm:
1. Time taken for line topology convergence is maximum and really high compared to the other 3 topology.
2. The time taken to converge for line increases drastically when number of nodes are increased.
3. The full network topology convergence gives the best performance.

Push-sum Algorithm:
1. 2D topology takes the least time for convergence of all the topology.
2. Time taken for line topology convergence is maximum and quite high compared to the other 3 topology.