

# COP5615: Fall 2015 PROJECT 2

Team Members:

Aakriti V Agrawal	14331642	<a href="mailto:aakritiv.agrawal@ufl.edu">aakritiv.agrawal@ufl.edu</a>
Divya Awatramani	46599609	<a href="mailto:dawatram@ufl.edu">dawatram@ufl.edu</a>

The zipped folder includes a Gossip Algorithm for information propagation and Push Sum Algorithm for sum computation for four different topologies: Full, Line, 3D, Imperfect 3D.

### **Folder Structure:**

```
| + src
|   | + main
|     | + scala
|       | + GossipSimulation
|         | + GossipSimulation.scala
| + build.sbt
```

### **To run:**

```
> cd Aakriti_Divya_Project2/Project2
> sbt
> compile
> run numNodes topology algorithm
```

numNodes -> no of actors

topology -> full, line, 3D, Imp3D

algorithm -> Gossip or PushSum

### **What is the largest network we managed to deal with for each type of topology and algorithm?**

Gossip Algorithm:

- Full, 3D and Imperfect 3D we have managed to run the gossip simulation for 10234 nodes.
- Line network we managed to simulate the gossip between 2197 nodes

Push Sum:

- For all the topologies we were able to converge the system for 10684 nodes.

## **Working modules:**

- Push Sum algorithm for all the four mentioned topologies
  - Full
  - Line
  - 3D
  - Imperfect 3D
- Gossip Simulation for all the above mentioned topologies.

## **Implementation:**

### **Gossip Algorithm:**

- Master randomly selects a node and triggers it to start the gossip.
- The selected actor increases its gossip count and notifies master that he has received the gossip once.
- Same actor now randomly selects one of his neighbor and passes the gossip to it.
- It works on scheduled basis after the specified time (50 milliseconds) the actor keeps on spreading the gossip to his neighbors.
- If all the nodes have received the gossip at least once, then the master shuts down the system.
- When a single node receives the gossip 10 times then it notifies the master. The master checks whether at that moment if all the nodes are killed or not. If yes, then the system converges else it continue to spread the gossip.

### **Push Sum Algorithm:**

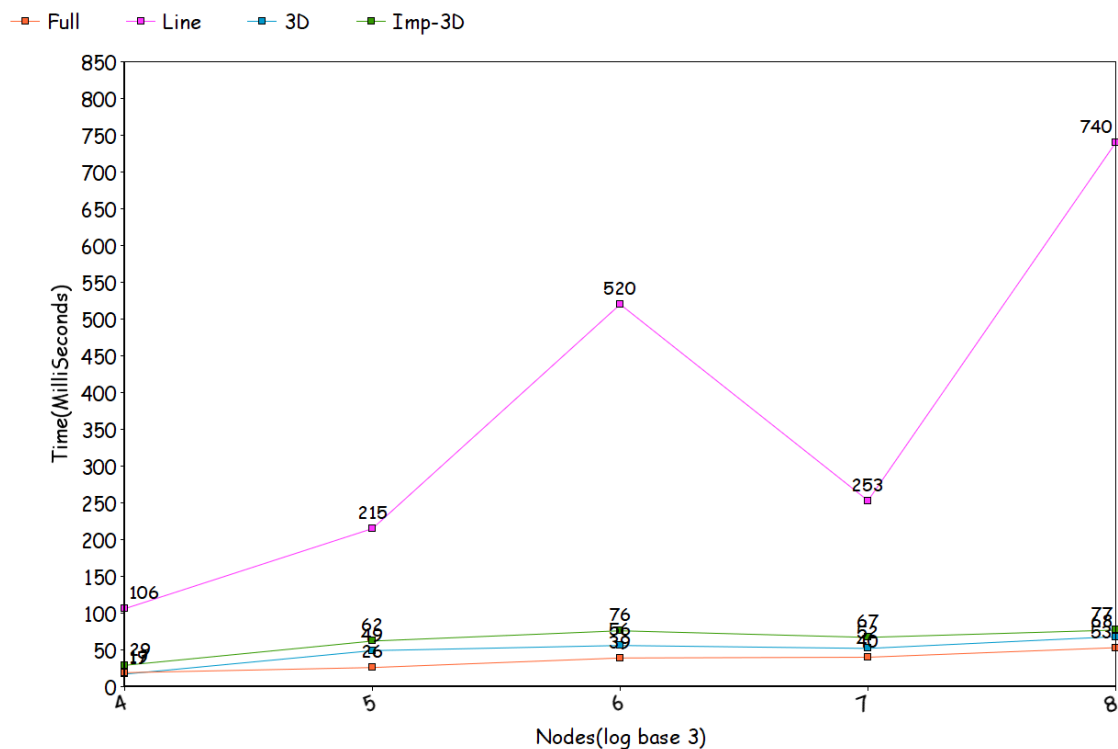
- All the nodes Sum and weight values are initialized with index value and 1.
- Master randomly selects one node and it adds the passed the sum, weight value pair into its respective values and then passes the half of it to one of his neighbors and keeps the remaining half with itself. It also maintains the S/W ratio
- As soon as the change in S/W value reaches  $1e-10$  more than thrice, then the system converges.

## Findings:

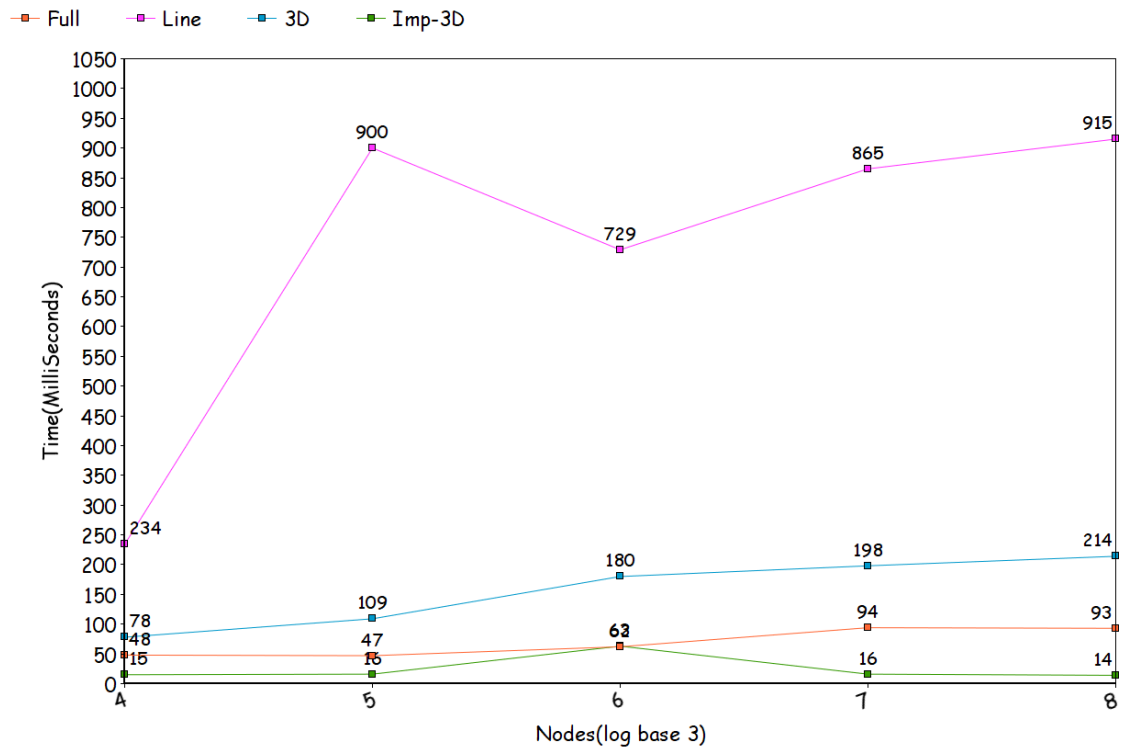
- Line Topology takes maximum time spreading the gossip as every node has at most two choices for selecting the next node to spread the rumor.
- Every time we run the algorithm for same topology and for the same number of nodes the convergence time varies as the nodes selected for spreading the gossip is random every time.
- If a node becomes idle in between before the convergence i.e. it has reached the maximum gossip count limit, than in that case the messages sent by his neighbors to him will be queued or lost and not propagated ahead hence the probability of spreading the gossip decreases. As that particular attempt of the neighbor to spread the gossip goes in vain.

## Graph for Dependency of convergence time of each topology as a function of the size of the network

### Gossip Algorithm:



## Push Sum Algorithm:



## Convergence Time:

### Gossip Algorithm:

NumNodes	Full	Line	Imperfect 3D	3D
64	19	106	17	29
125	26	215	49	62
216	39	760	56	76
343	40	761	52	67
512	53	891	68	77
729	44	1189	88	93
1000	62	3457	99	113
1331	78	28200	90	124
1728	113	32455	205	209
2197	102	45274	140	146
2744	144	80374	147	445
3375	1369	--	246	604
4096	190	--	223	212
4913	585	--	315	223

### Push Sum Algorithm:

NumNodes	Full	Line	3D	Imperfect 3D
64	48	234.00	78.00	15.00
125	47.00	1068.00	109.00	16.00
216	62.00	729.00	154.00	63.00
343	94	2696.00	298.00	16.00
512	93	965.00	485.00	14.00
729	125	13710.00	551.00	62.00
1000	184	15874.00	818.00	58.00
1331	193.00	13622.00	1246.00	265.00
1728	272.00	8880.00	1361.00	95.00
2197	199.00	9055.00	2600.00	382.00
2744	281.00	25328.00	3647.00	417.00
3375	1503.00	4753.00	4251.00	107.00
4096	367.00	19739.00	6729.00	161.00
4913	507.00	28198.00	9375.00	134.00
10000	835	163740	40434	209