Report.md

Project Report

Group Member

1. Ganeson Ravichandran, g.ravichandran@ufl.edu
2. Sai Pavan Kalyan Munaga, saipavank.munaga@ufl.edu

Implementation Details:

Gossip

```
1. Each Gossip Actor coninuously sends its message to choosen random worker.
2. When it receives a message from other worker, considers it as its secret mes
3. When any actor receives same secret for 10 times, It
```

Push Sum

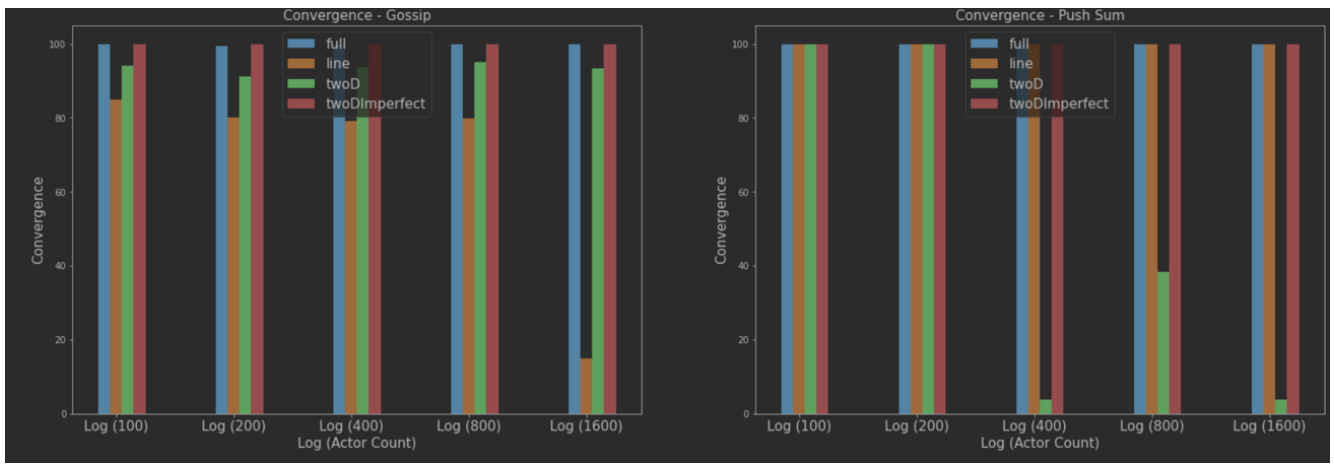
```
1. When a actor doesn't see any change in Sum / Weight beyond 10^-10 for consec
2. We terminate the actor, after all nodes in the network terminate.
```

This report is generated from running the script reports.sh and storing the output in reports.out. The result is then parsed through python notebook plots.ipynb to obtain the shown data frame and plots.

Plots

Convergence

	Gossip full Convergence [%]	line Convergence [%]	twoD Convergence [%]	twoDImperfect Convergence [%]	Push Sum full Convergence [%]	line Convergence [%]	twoD Convergence [%]	twoDImperfect Convergence [%]
100	100.0	85.00	94.000000	100.0	100.0	100.0	100.000000	100.0
200	99.5	80.00	91.326531	100.0	100.0	100.0	100.000000	100.0
400	100.0	79.25	93.750000	100.0	100.0	100.0	3.750000	100.0
800	100.0	79.75	95.025510	100.0	100.0	100.0	38.265306	100.0
1600	100.0	15.00	93.375000	100.0	100.0	100.0	3.750000	100.0

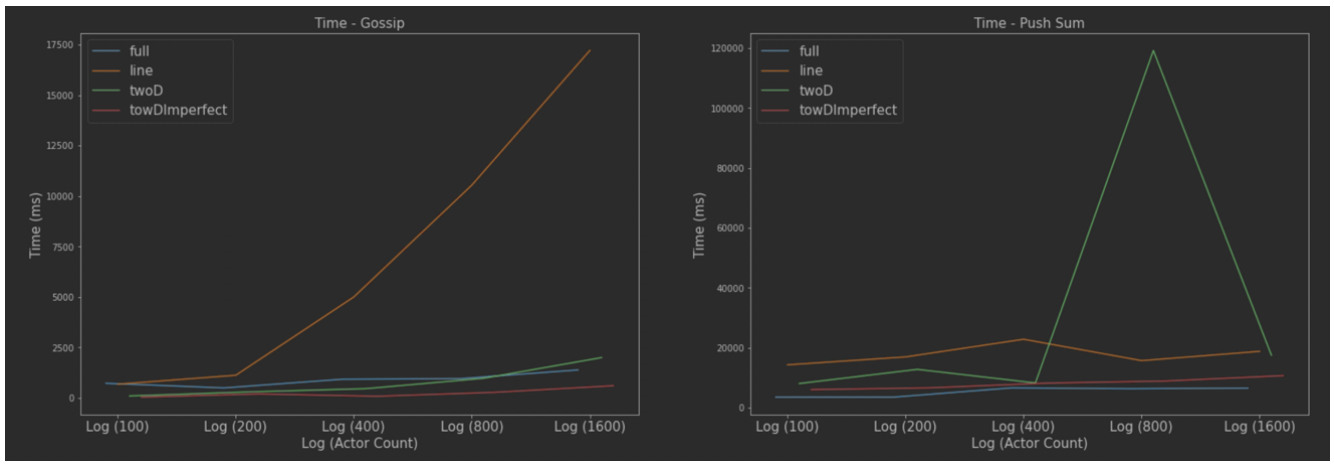


Observation

- In both algorithms, Full and twoD imperfect has full convergence. All actors receive the message or sum.
- This implies, random sause added to the topology, helps in better spread of message.
- Line performs the worst and this is expected considering the very restrictive constraint line has. It can choose one from at most 2 options
- The lesser the constraint in choosing neighbours, more options for choosing and better spread in lesser time. Thus, full performs the best.

Time

	Gossip full Time [ms]	line Time [ms]	twoD Time [ms]	twoDImperfect Time [ms]	Push Sum full Time [ms]	line Time [ms]	twoD Time [ms]	twoDImperfect Time [ms]
100	729	686	103	45	3530	14312	8075	6037
200	502	1128	301	202	3531	16948	12793	6637
400	930	5004	464	88	6620	22844	8262	8212
800	959	10539	977	284	6338	15711	119215	8911
1600	1391	17207	2003	610	6535	18809	17513	10705



Observation

- In both algorithms, Line take long time to converge. This is because of the constraint line has. It can send message to one of tow neighbouts.
- There is a bump in push sum, twoD which may appear misleading. But, if we correlate this with convergence, the convergence for 400, 1600 for twoD is very low. But for 800 it is high. the code was not able to receive any actors converging for continuous 5 seconds for 400 and 1600 - push sum and hence stopped with very low convergence. But for 800, more actors converged, thus there is a bump in time.
- The lesser the constraint in choosing neighbours, more options for choosing and better spread in lesser time. Thus, full followed by towDImperfect performs the best.