

Project 2

Suryansh Singh : 41403921

Piyush Johar : 01935-949

Follow the instructions to run the code

Extract the zip file

sbt compile;

sbt "run <args>"

Arguments will be: <number of nodes> <topology> <algorithm>

Example:

sbt "run 1000 Line Gossip"

sbt "run 1000 FullNetwork Gossip"

sbt "run 1000 3D Gossip"

sbt "run 5000 Imperfect3D Push Sum"

To run bonus file command is

sbt "run 1000 Line Gossip" : This will automatically take 10 percent.

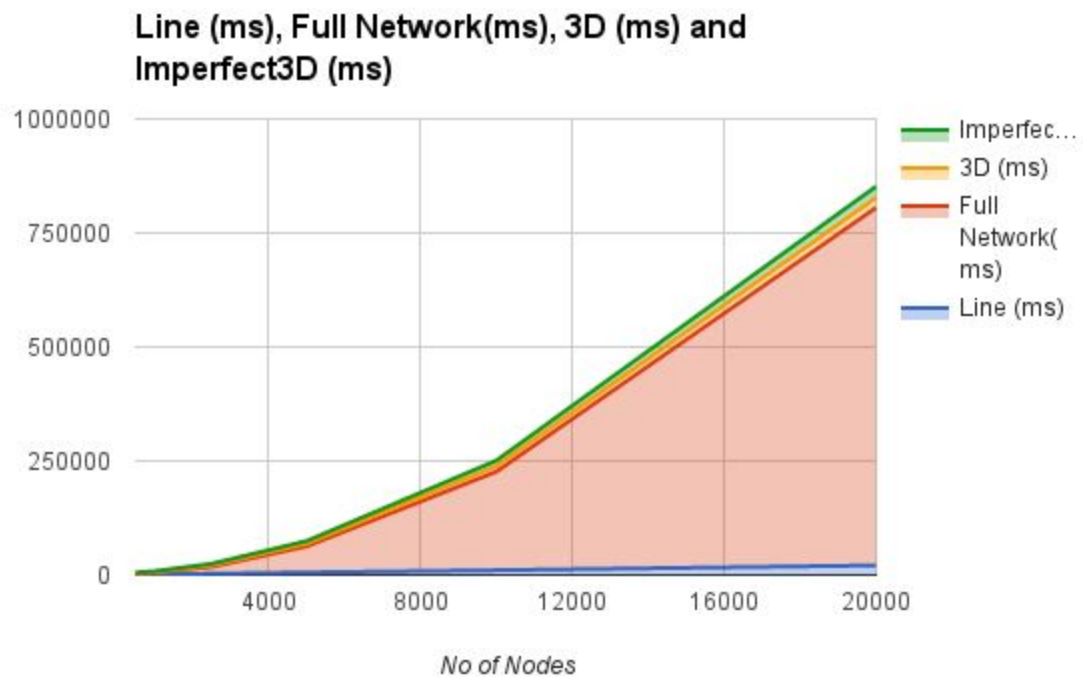
sbt "run 1000 Line Gossip 20" : This will take the specifies 4 argument as percentage.

Note: To find a random neighbour of the current actor we used an algorithm which takes $O(1)$ time.

Gossip Algorithm:**Unit of Time:** milliseconds

Termination condition for each actor: After receiving 10 gossips

No of Nodes	Line (ms)	Full Network(ms)	3D (ms)	Imperfect3D (ms)
500	1062	1669	1248	1362
1000	1447	3642	1605	1673
2500	2935	14943	3441	3456
5000	5665	57237	5806	5932
10000	10481	216248	12062	12278
20000	21035	784327	21952	24675



In Gossip implementation, an actor is terminated in 2 ways:

1. whenever the actor receives 10 gossips
2. whenever all the neighbours of an actor are unreachable

In our implementation, a situation arises when current actor's all neighbours are unreachable (node deadlock). In that case, this current actor is closed and the main system starts the Gossip process with the remaining actors again. This method makes sure that full system converges. For our Gossip implementation, the line performs better because the chances of occurrence of above mentioned situation are maximum in line topology. And in that case, our main system restarts the Gossip system with remaining actors.

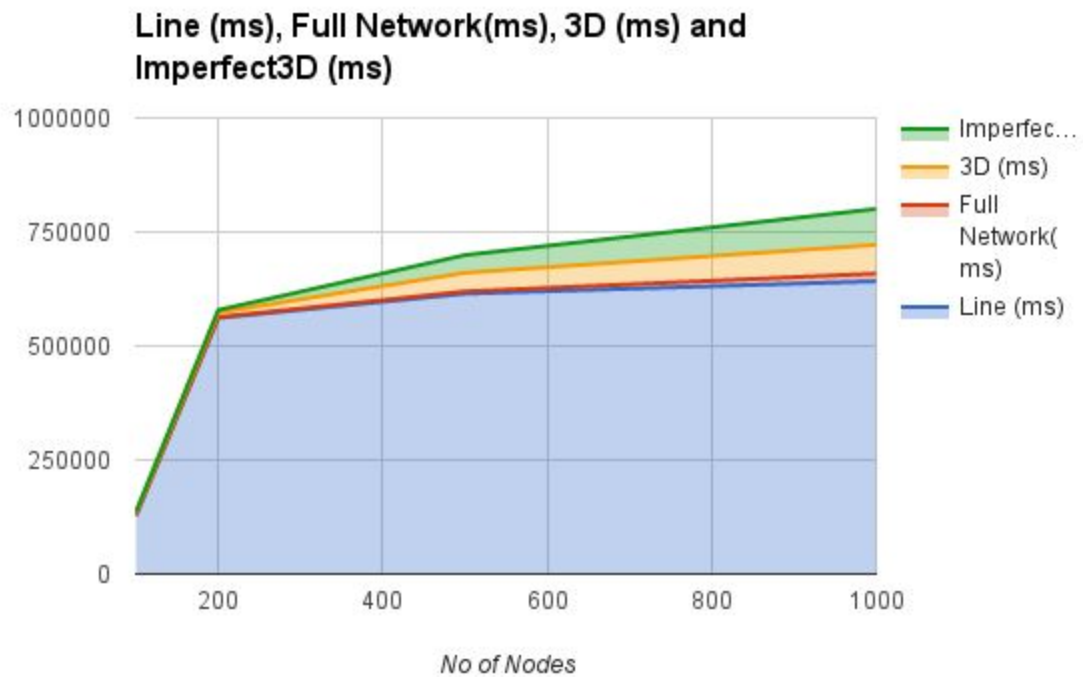
If we were not allowing the system to restart for above situation, then the current actor's termination condition would never be achieved (that is it will never receive 10 Gossips), and the system will continue to run forever, which would not be ideal. In this scenario, the Line topology would perform the worst, because the probability of above situation is maximum in Line. And the Line would continue to run forever.

Push Sum Algorithm:

Unit of Time: milliseconds

Termination condition for each actor: 3 consecutive rounds of push sums is less than 0.0000000001

No of Nodes	Line (ms)	Full Network(ms)	3D (ms)	Imperfect3D (ms)
100	128182	1138	4292	2878
200	561472	1893	9770	5632
500	614732	4789	40883	38919
1000	642123	16587	63048	78770



In Push sum implementation, an actor is terminated in 2 ways:

1. whenever the actor receives 10 gossips
2. whenever all the neighbours of an actor are unreachable

Observations:

Gossip Algorithm:

Line Topology

The line topology takes minimum time for our implementation as explained above. In real scenario, line would take maximum time (explained above)

3D

Time to propagate gossip is almost linear.

Imperfect 3D

As per our observation, Imperfect 3D takes more time than 3D grid topology but less than full network.

Full Network

Full network topology takes the maximum time in this simulation. In real life simulation, line would take maximum time. We have explained this above.

Push Sum Algorithm:

Line

Line Topology takes the highest amount of time to compute push sum in the network. But with increase in node size, the gain in time is decreasing.

3D Topology

3D takes less time than Line, but more than full network and imperfect 3D.

Imperfect 3D Topology

Imperfect takes less time than 3D but more than full network.

Full N/W Topology

It takes the minimum time of all the topologies.

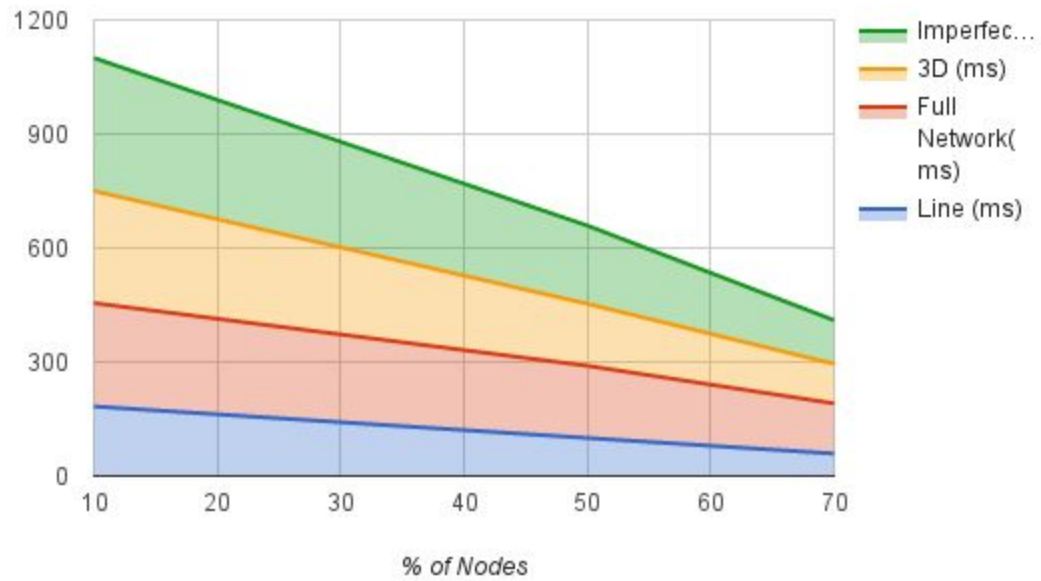
Bonus Question

1. For bonus part, we are taking another parameter from user, which is the percentage of nodes to kill.
2. We kill this percent of nodes from our actor system after 100ms from our system start. If our implementation of algorithm for any topology finishes before 100ms, our implementation will not kill any nodes. But in case the implementation goes beyond 100 ms(which will happen for most scenarios), our main system will close x percent of total nodes (except the current node). These nodes will be selected randomly from the current active nodes.
3. In case the user misses out passing this percentage parameter, our system assumes a default 10% and kills 10% of nodes after 100ms.
4. After closing these random nodes, a situation might arise where the current node may not have any active neighbour. Our implementation will then find a random actor and continue to propagate the message throughout the whole system. This is fault tolerance in our system.

Gossip Algorithm

% of Nodes	Line (ms)	Full Network(ms)	3D (ms)	Imperfect3D (ms)
10	183	272	295	350
50	100	190	163	206
70	59	132	104	115

Line (ms), Full Network(ms), 3D (ms) and Imperfect3D (ms)



Push Sum Algorithm

% of Nodes	Line (ms)	Full Network(ms)	3D (ms)	Imperfect3D (ms)
10	3923	1048	3955	3923
50	99	648	2161	206
70	65	465	216	125

Line (ms), Full Network(ms), 3D (ms) and Imperfect3D (ms)

