Project 2 – Gossip Simulator

Implementation Details:

For Gossip algorithm we have assumed that the convergence will occur only when all the actors have received the rumour 10 times. In case if there happens to be a non-converged actor whose all neighbours are already converged, we terminate that actor and mark it as converged because without any neighbours the actor won't be able to receive any rumours and converge by itself.

For push-sum algorithm, we have assumed that the convergence occurs for an actor if its sum to weight ratios does not change more than 10^-10 for 3 consecutive rounds. Once all the actors are converged, we declare that the push sum system has converged successfully.

We will be simulating these two protocols for the following four topologies.

1. **Full Network –** Each actor in the system is a neighbour of all the other actors. Thus, each actor can randomly select any other actor to send a rumour.
2. **Line –** All the actors are arranged in a line. Thus, each actor can randomly select either previous or next actor to send a rumour.
3. **3D –** All the actors are stored in a 3D grid. Thus, each actor can randomly select an actor from its at most 6 neighbours to send a rumour.
4. **Imperfect 3D –** In this topology, all the actors are arranged in same structure as 3D topology. The only difference is that along with the 6 neighbours, each actor can select 1 more random actor as its neighbour. Thus, each actor can select a random actor from its at most 7 neighbours to send a rumour.

In our implementation, we have stored all the actors in a 3D array irrespective of the topology. For the given number of nodes, first we determine the nearest perfect cube number by rounding up the node count. Suppose we get the perfect cube as N, then for each topology we will have 3D array as following:

1. **full –** [1, 1, N]
2. **line –** [1, 1, N]
3. **3D –** [$N^{1/3}$, $N^{1/3}$, $N^{1/3}$]
4. **imp3D –** [$N^{1/3}$, $N^{1/3}$, $N^{1/3}$]

For full and line topologies, we essentially have a 1D array which is wrapped in a 3D array. We have done this to have a common data structure instance where all the actor references are stored so that we can have a generic logic to find the neighbours of an actor.

Once all the actors are successfully initialized, we randomly select an actor and send it a rumour or a push-sum message in case of push-sum algorithm. We have also used the scheduler functionality from akk.net using which we wake up all the actors after a fix interval and tell them to send a rumour or push-sum message to their neighbours only if they have already received the rumour.

Gossip Protocol:

After simulating gossip protocol for each of the topologies, followings were our findings regarding the actor count and the corresponding time taken to converge the algorithm.

1. Full

Actors Count	Time (ms)
27	12818
125	14359
512	16442
729	16459
1000	17465

2. Line

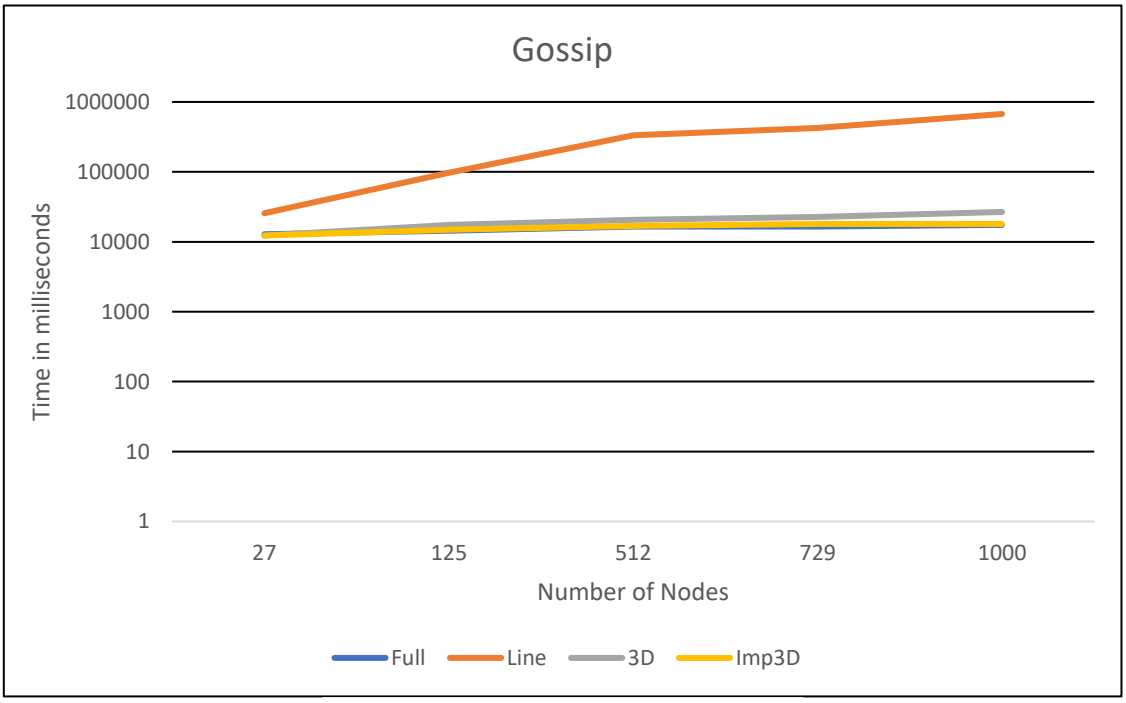
Actors Count	Time (ms)
27	25622
125	97375
512	334386
729	426984
1000	672201

3. 3D

Actors Count	Time (ms)
27	12310
125	17409
512	20512
729	22592
1000	26722

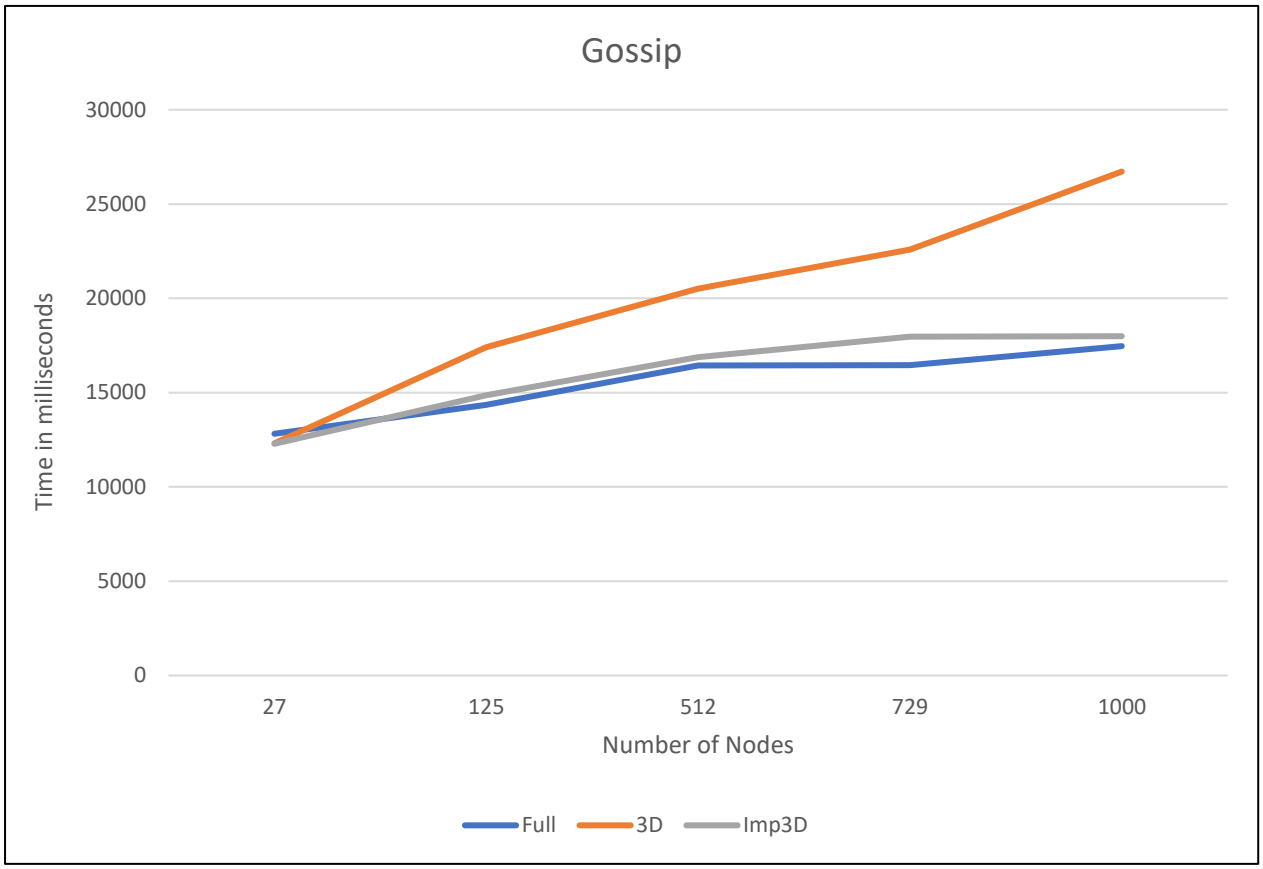
4. Imperfect 3D

Actors Count	Time (ms)
27	12287
125	14862
512	16886
729	17972
1000	17992



Logarithmic scale (Base 10) y axis

Following is a chart without line topology so that we can see the difference between rest of the topologies.



Push-Sum Protocol:

After simulating the push-sum protocol for each of the topologies, followings were our findings regarding the actor count and the corresponding time taken to converge the algorithm.

1. Full

Actors Count	Time (ms)
27	9859
125	11341
512	11770
729	12328
1000	13181

2. Line

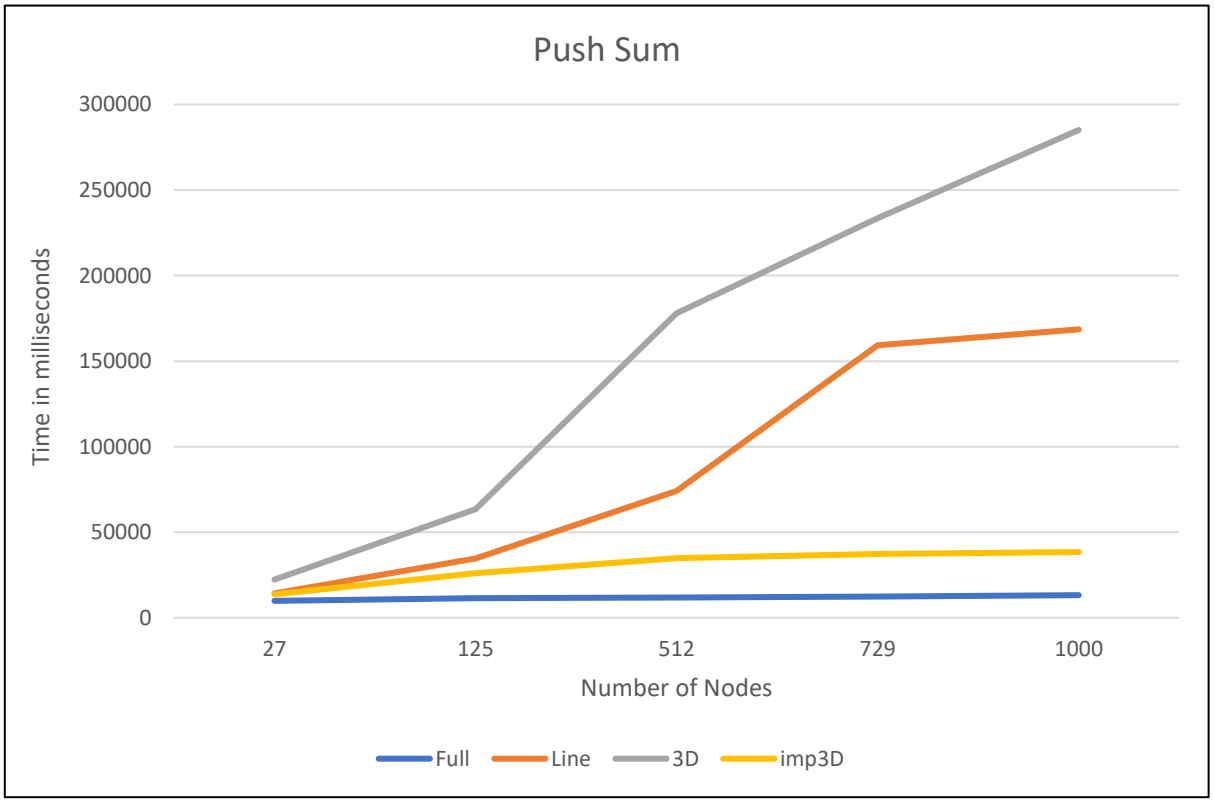
Actors Count	Time (ms)
27	14229
125	34707
512	74032
729	159207
1000	168551

3. 3D

Actors Count	Time (ms)
27	22285
125	63459
512	177940
729	233496
1000	285089

4. Imperfect 3D

Actors Count	Time (ms)
27	13670
125	26096
512	34808
729	37305
1000	38392



Interesting findings:

1. For gossip protocol, the topologies in the order of fastest to slowest convergence time are full, 3D, imp3D and line. Compared to full, 3D and imp3D, the line topology took significantly more time to converge.
2. From this we can infer that, the time required to converged to the protocol depends on the number of neighbours available to each actor. In full topology, each actor is the neighbour of all other actors in the system, thus it has the fastest convergence time.
3. The convergence time for imperfect 3D topology is relatively better than the 3D topology. This is because in case of 3D topology each neighbour has 1 more random neighbour along with the existing 6 neighbours.
4. In case of push-sum protocol, the topologies in the order of fastest to lowest convergence time are full, imperfect 3D, line, and 3D.
5. Here, the line topology is taking less time to converge than the 3D topology which is little bit interesting. After debugging we found out that, for 3D topology, each actor has at most 6 neighbours. Some of these neighbours are from different dimension and the sum value present in them is significantly different from what the actor currently has. Because of this it took relatively more push messages to converge the ratio of sum to weight.