DOS PROJECT-2

GOSSIP ALGORITHM & PUSH-SUM IMPLEMENTATION IN ELIXIR

Pradosa Patnaik (UFID : 12889584)

Supraba Muruganantham (UFID : 92159813)

Brief Description:

The aim of the project is to implement Gossip and Push-Sum algorithm in Elixir for different network topologies like FULL,2D, Imperfect 2D and Line and analyze their convergence time. The Gossip and Push-Sum algorithms are briefly described below.

Gossip algorithm:

Gossip is an information propagation algorithm designed for distributed systems. The aim of the algorithm is to send the message to every node in the network with the help of random neighbor selection.

- The Gossip protocol works by initiating the process from a single actor which forwards the message to the other actors. The point of convergence reached is when each of the actors listens to the message 10 times (arbitrary threshold).

Push-Sum Algorithm: Push-Sum is a variant of Aggregate calculation algorithm designed to calculate sum/average quantity for a distributed network. The algorithm works as follows,

- The Push Sum algorithm works by sending messages in the form of pairs(s,w) where s is the value of the actor number and w = 1 for each actor. The propagation converges when the s/w ration doesn't change when compared to a predefined value. (In our case 10^-10) for three consecutive times.

A glimpse of the network topologies: Network topology plays a vital role for the convergence time of the Gossip/Push-Sum algorithm. The topology determines who can be the neighbor of the current node. In this project we are considering four topologies, Full,2D, Imperfect 2D and Line.

Full Network: In this topology, every node can be a neighbor of all other nodes present in the network.

2D Grid: In this topology, nodes can only communicate to the neighbors, which are determined by placing the nodes in a 2D grid.

Imperfect 2D: In Imperfect 2D the node has an additional random neighbor along with the 2D grid neighbors.

Line: In line topology, the nodes can talk to adjacent two nodes on either side (Except for the end nodes which can only talk to one neighbor).

Expected Results:

The convergence time for Gossip and Push-Sum greatly depends on the network topology. In case of full network topology, the ideal convergence time should be in the order of O(logn) where n is the number of nodes present in the network. In case of Line topology, the average convergence time should be O(logn) but in worst case the convergence time can be O(n). In case of 2D and imperfect 2D the convergence time should remain in between full and Line topology time. Imperfect 2D should converge faster than 2D as it can communicate to an extra random neighbor. So, the ideal convergence time should come in the below order.

full < imperfect 2D < 2D < line

Implementation Details:

This project is implemented in Elixir using the Actor-Model feature of the language. For the simulation each actor is considered as a node and communicate with each other through message passing. Here we are using Genserver as the primary abstraction of a process which also has a capacity to maintain its state.

Project2.ex is the main file which takes three arguments from the command line NumNodes (Number of nodes in the network), Topology (full,2D, Imp2D, line) and Algorithm (gossip, push-sum) and starts the program Gossip/Push-Sum based on the command line argument and initializes a master process, which in turn spawns same number of process as the number of nodes passed from the command line (In case of 2D and Imperfect 2D the number of nodes are rounded to nearest prefect square.

In case of Gossip, the master process initiates the gossip by sending a message to one of the random process. Then the gossip algorithm continues until all nodes(processes) received the message at least once. To determine the convergence, we have set an arbitrary threshold of 10. Each process keeps the count of the number of time it heard the same message. As soon as the count reaches the threshold, the process makes it inactive and do not participate in the gossip anymore. Here we assume that the network converges, if every node in the network hears the message at least once.

In case of push-sum, after the nodes are created, the 's' and 'w' value of each process are initialized. Then the one of the actors starts the push-sum algorithm by updating its 's' and 'w' value to half and sending (s/2, w/2) pairwise to a random neighbor. The algorithm continues until one of the actor's ratio s/w stops changing more than 10^-10 in 3 consecutive rounds. In this case we are assuming the estimates have nearly been converged.

Interesting Observations:

Observations for Gossip:

- The convergence time for **Gossip** for different topologies did not follow the ideal convergence order ***(full<Imp2D<2D<line)*** for all network sizes.
- For small network sizes (below 400 nodes) the convergence order was found to be ***(full<Imp2D<2D<line)*** same as the ideal order.
- For large networks (size >400) the line topology performed better than 2d topology. So, for large network size the convergence order was found to be ***(full<Imp2D<line<2D).***
- Also, the for the same network size and topology, multiple runs of Gossip yields different convergence time.
- Overall, **Full** network always performed the best yielding lowest convergence time, whereas the worst convergence time was **2D** or **Line** depending on the network size.
- We could able to converge maximum 10000 actors with a reasonable convergence time of *530.562* seconds with full topology.

Observations for Push-Sum:

- The convergence time for **push-sum** for different topologies followed the ideal convergence order ***(full<Imp2D<2D<line)*** for all network sizes.
- The convergence time for the Line topology was the worst for all the network sizes.
- The S/W ratio after convergence was almost equal to the true average of the network.
- For large network size push-sum converges faster than Gossip for same topology.
- We were able to converge maximum 10000 actors with a reasonable convergence time of *90.002* seconds with full topology.

Maximum Values for Gossip:

Full – 10000, Imp 2D – 10000, Line – 4900, 2D – 4900

Maximum Values for Push Sum:

Full – 10000, Imp 2D – 10000, Line – 4900, 2D – 4900

How to Run:

The project is deployed as a mix project with name ***project2* .**
Compile the project from the *project2* directory.

```
$ mix escript.build
```

Run the project with the command

```
$  ./project2 49 full gossip
```

Or

```
$  ./project2 49 2D pus-sum
```

Possible values of topologies:

- Full
- 2d
- Imp2D
- line

Convergence assumption:

In case of gossip : Here we assume that the network converges, if every node in the network hears the message at least once **.**

In case of push-sum: In case of push-sum, our network converges if **one** of the actor's ratio s/w stops changing more than 10^-10 in 3 consecutive rounds. In this case we are assuming the estimates have nearly been converged.

Results:

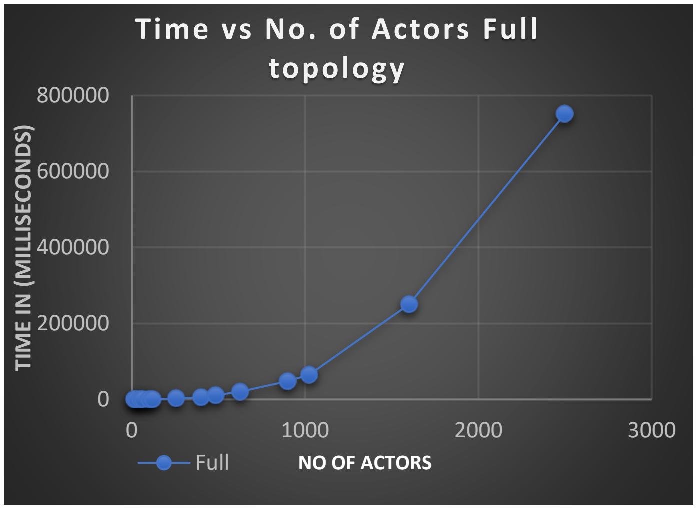
Gossip:

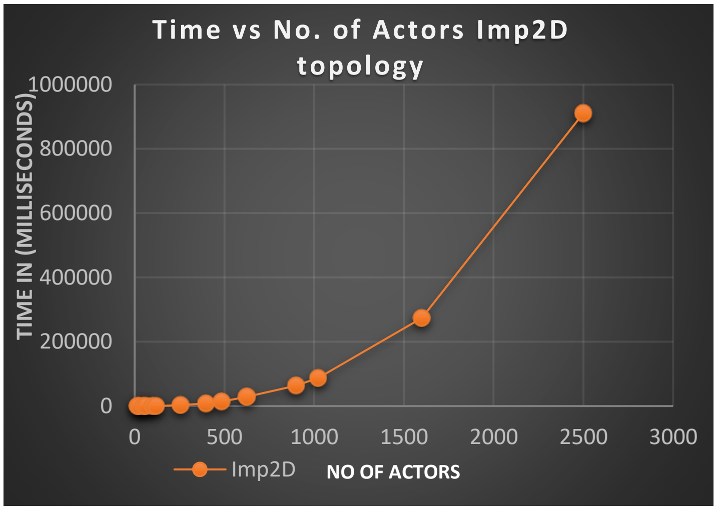
Below table represents the convergence time (in milliseconds) taken by different topologies for gossip algorithm.

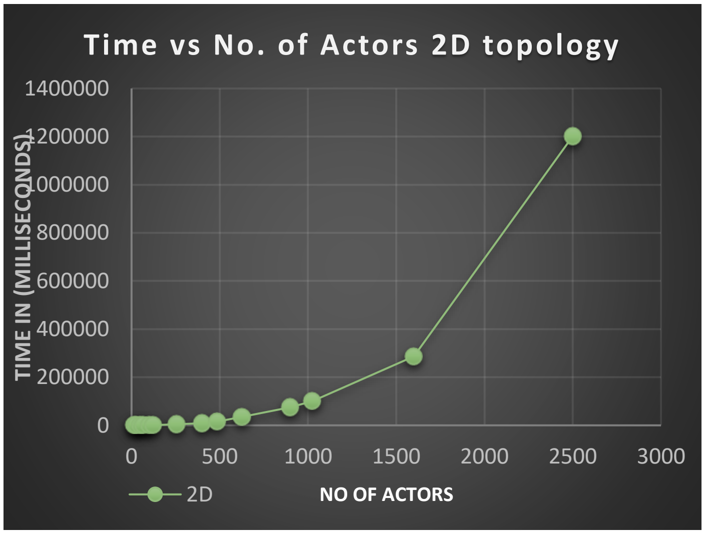
No of Actors	Full	Imp2D	2D	Line
16	11	3	13	17
25	20	24	29	37
49	58	68	92	124
64	95	119	164	233
100	360	399	491	528
121	379	553	764	793
256	2898	3124	3948	3654
400	5107	8518	9095	9696
484	10861	13696	15798	14297
625	20693	28743	34828	26939
900	47821	63516	75200	61873
1024	64735	87415	99775	80407
1600	250442	274182	286156	275328
2500	751362	911080	1201340	1112099

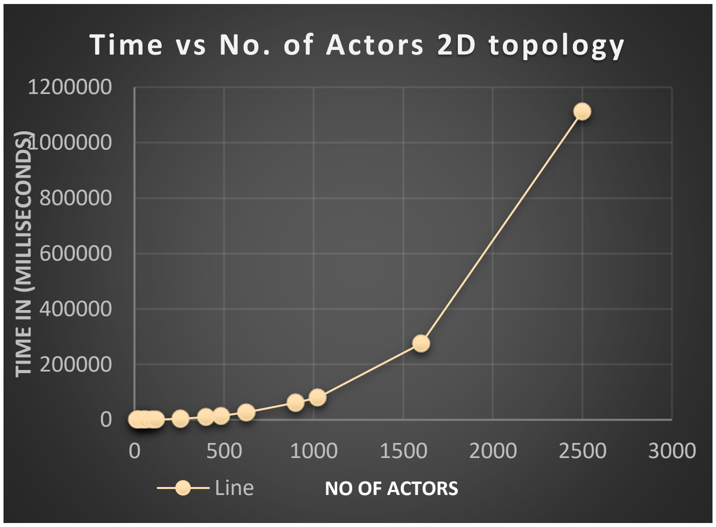
(Table1. Convergence time taken for each topology for different number of nodes(Gossip))

Individual Graphs for Time vs No. of Actors for each topology:

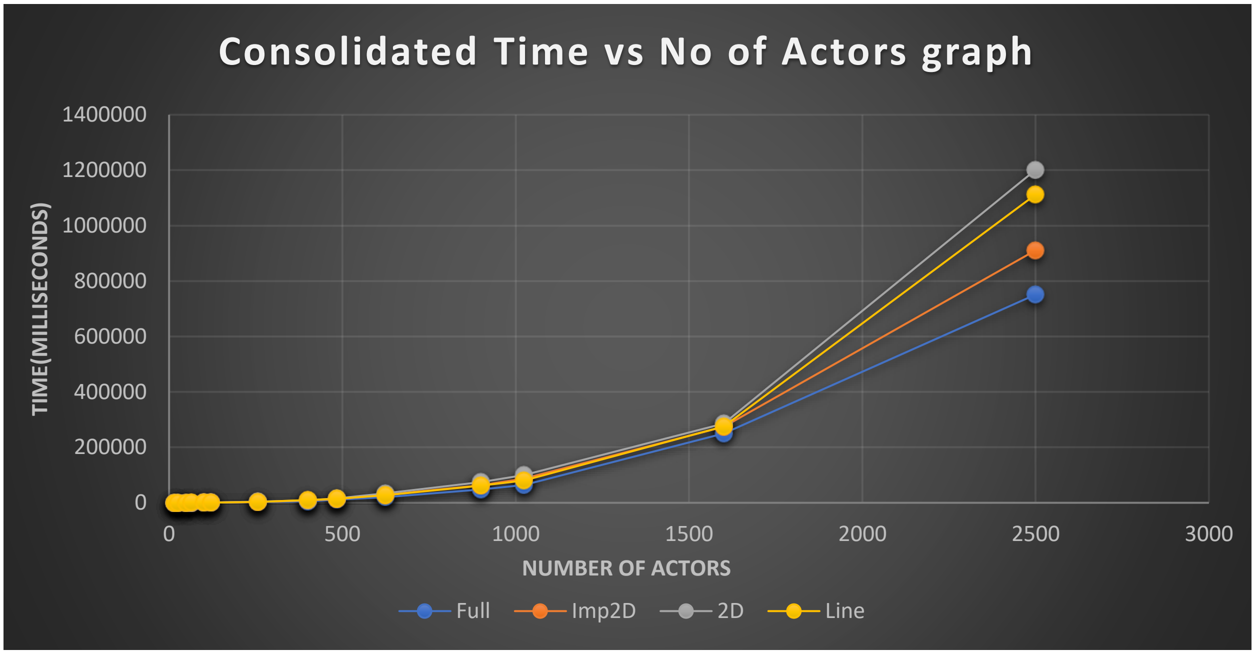








Consolidated Graph for Gossip:



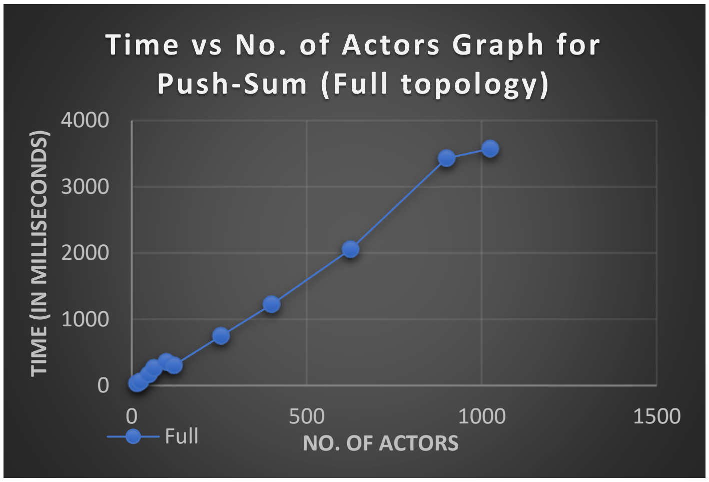
PUSH SUM:

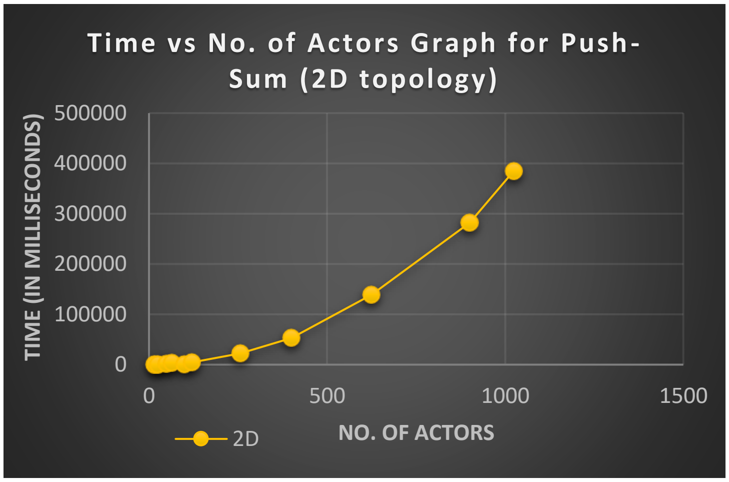
Below table represents the convergence time (in milliseconds) taken by different topologies for push-sum algorithm.

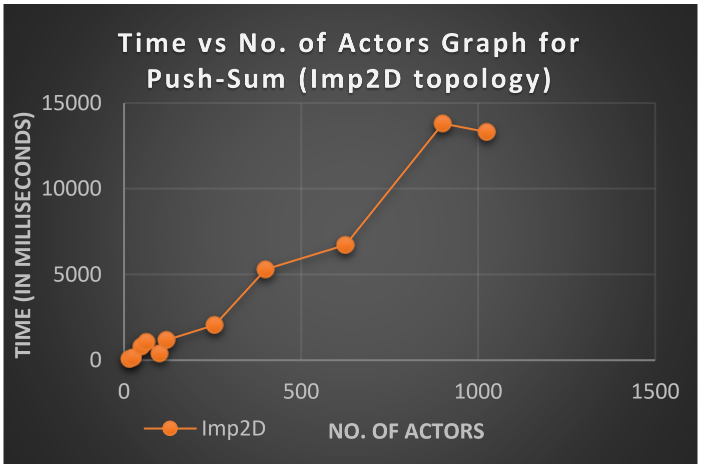
No of Actors	Full	Imp2D	2D	Line
16	35	81	96	497
25	63	144	244	1922
49	167	805	1417	20240
64	262	1075	3421	53803
100	360	399	491	57860
121	309	1182	4794	88144
256	752	2059	22670	146921
400	1227	5310	52729	272359
625	2058	6732	138966	1406275
900	3428	13806	281964	1464385
1024	3574	13306	385056	1915583

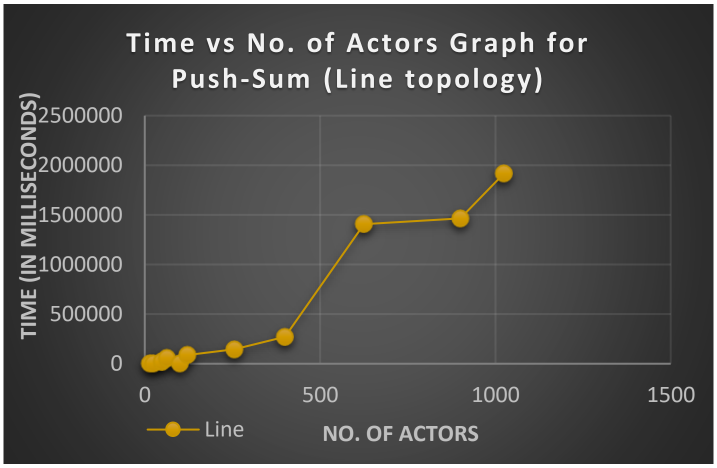
(Table2. Convergence time taken for each topology for different number of nodes (push-sum))

Individual Graphs for Time vs No. of Actors for each topology:

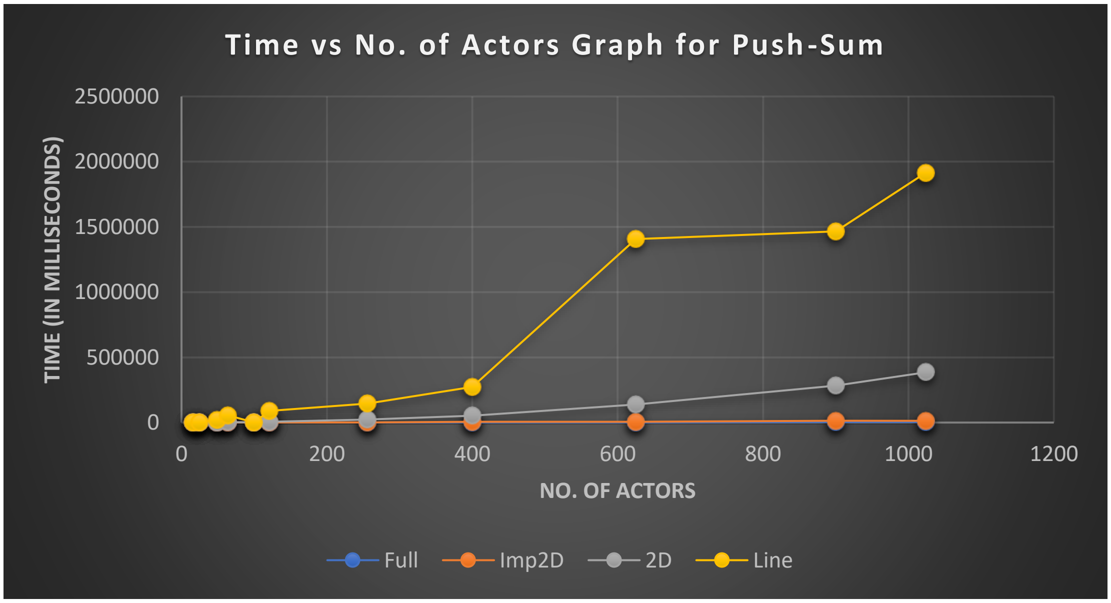








Consolidated Graph for Push-Sum:


Time vs No. of Actors Graph for Push-Sum
TIME (IN MILLISECONDS)
NO. OF ACTORS
Full
Imp2D
2D
Line