DOS Project 2

Surya Sudharshan (UF ID:5019-3163)

Problem Statement:

Gossip type algorithms can be used both for group communication and for aggregate computation. The goal of this project is to determine the convergence of such algorithms through a simulator based on actors written in Elixir. Since actors in Elixir are fully asynchronous, the particular type of Gossip implemented is the so called Asynchronous Gossip.

Gossip Algorithm:

The Gossip algorithm involves the following:

• **Starting**: A participant(actor) it told/sent a rumour(fact) by the main process • Step: Each actor selects a random neighbour and tells it the rumour

• **Termination**: Each actor keeps track of rumour and how many times it has heard the rumour. It stops transmitting once it has heard the rumour 10 times (10 is arbitrary, you can play with other numbers or other stopping criterion).

Push-Sum Algorithm involves the following:

• **State**: Each actor Ai maintains two quantities: s and w. Initially, s = xi = i (that is actor number i has value i, play with other distribution if you so desire) and w = 1

• **Starting**: Ask one of the actors to start from the main process.

• **Receive**: Messages sent and received are pairs of the form (s,w). Upon receive, an actor should add received pair to its own corresponding values. Upon receive, each actor selects a random neighbour and sends it a message.

• **Send**: When sending a message to another actor, half of s and w is kept by the sending actor and half is placed in the message.

• **Sum estimate**: At any given moment of time, the sum estimate is s w where s and w are the current values of an actor.

• **Termination**: If an actors ratio s w did not change more than 10−10 in 3 consecutive rounds the actor terminates.

Topologies:

• **Full Network** Every actor is a neighbour of all other actors. That is, every actor can talk directly to any other actor.

• **3D Grid**: Actors form a 3D grid. The actors can only talk to the grid neighbours.

• **Random 2D Grid**: Actors are randomly position at x, y coordinates on a [0-1.0]X[0-1.0] square. Two actors are connected if they are within .1 distance to other actors.

• **Sphere**: Actors are arranged in a sphere. That is, each actor has 4 neighbours (similar to the 2D grid) but both directions are closed to form circles.

• **Line**: Actors are arranged in a line. Each actor has only 2 neighbours (one left and one right, unless you are the first or last actor).

• **Imperfect Line**: Line arrangement but one random other neighbour is selected from the list of all actors.

Expected Results:

From the description of the Network topologies whether it be Gossip or Push-Sum the order in which the time taken to converge from less to higher time is given by . The time taken to converge is calculated by the difference in system times when the process is initiated until all the actors terminate. This is in the order of milliseconds.

Actual Results:

2D Topology

No. of Nodes	Convergence Time(ms)
50	2891
100	2031
150	2047
250	3984
1000	54313

Full Topology

No. of Nodes	Convergence Time(ms)
50	4047
100	6531
150	8531
250	13762
1000	59188

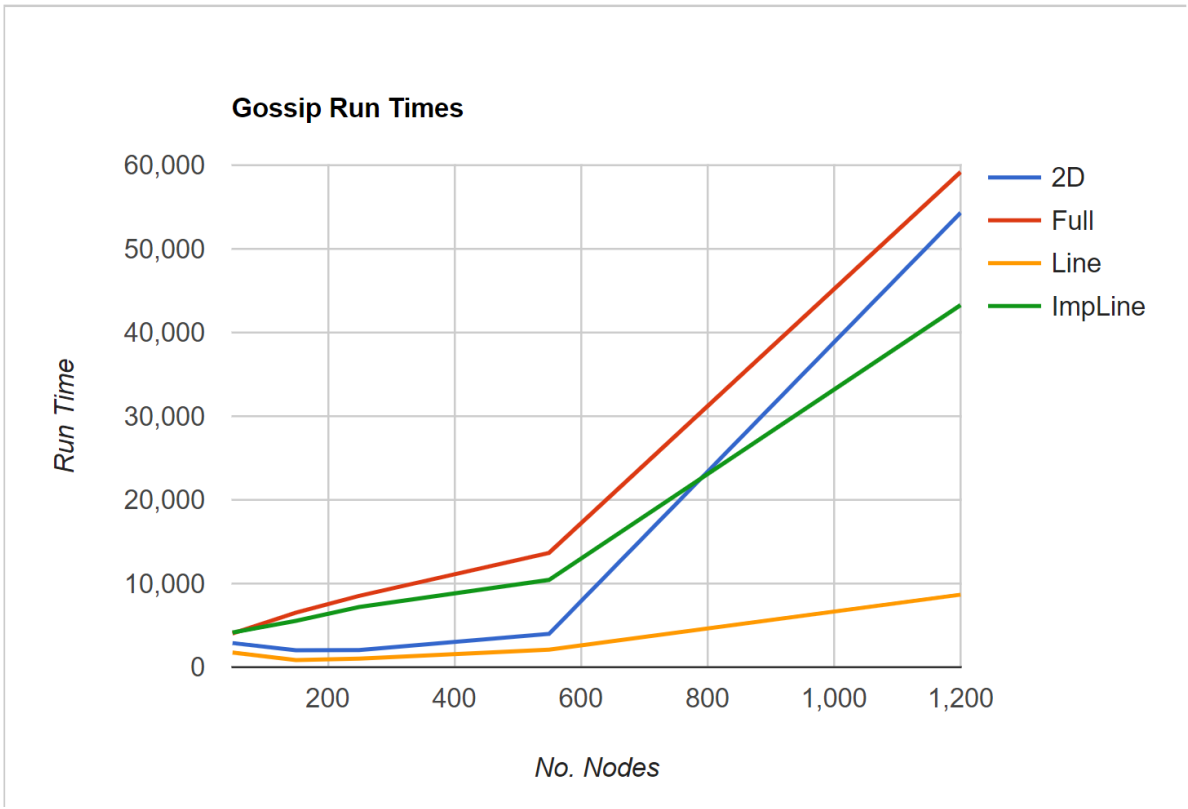
Line Topology

No. of Nodes	Convergence Time(ms)
50	1750
100	844
150	1031
250	2094

1000	**8672**

ImpLine Topology

No. of Nodes	Convergence Time(ms)
50	**4140**
100	**5531**
150	**7188**
250	**10437**
1000	**43281**



Push-Sum topology:

2D Topology

No. of Nodes	Convergence Time(ms)
50	719
100	56250
150	108797
250	227656
1000	500000*

FullTopology

No. of Nodes	Convergence Time(ms)
50	203
100	575
150	1141
250	3500
1000	78094

Line Topology

No. of Nodes	Convergence Time(ms)
50	4359
100	34000
150	86172
250	106031
1000	500000*

ImpLine Topology

No. of Nodes	Convergence Time(ms)
50	234
100	594
150	1187
250	3078
1000	58906

Interesting Observations:

After several rounds of testing and observation, what caught our observation was the convergence of Line topology was faster when compared to rand2D topology after a certain number of nodes, say around 150 or so.

Apart from this, the order of convergence can be obtained as:

T(Impline)<T(Full)<T(Line)<T(rand2D)

The kind of convergence issues that emerged while implementing this project have inculcated a good understanding of the gossip and push-sum protocols. Line topology has been the most difficult one to converge as the number of neighbors is limited for an actor and the message propagation is not that frequent as it is with the other networks