

COP 5615: Distributed Operating Systems Principles

Project 2

Gossip and Pushsum Algorithm implementation based on Erlang Programming Language

Team members

* Rahul Vikram Porwal UFID: 44590947 Email: rahulporwal@ufl.edu

* Pujan Narendra Kothari UFID: 70615094 Email: p.kothari@ufl.edu

Introduction

Gossip is a rumour message which is spread among a group. In this Project, We will implement Gossip type algorithms i.e gossip algorithm and push sum algorithm. Using the actor based model, We determine the convergence of these algorithms and compare them.

Instructions to execute the project

- 1) Install the latest version of erlang.
- 2) Download the project1.zip and unzip it to any desired path.
- 3) Run command prompt at the file location with commands - erl
To run erlang environment
- 4) Compile the erlang file-
c(gossipmain2).
- 5) Then, To execute the Gossip or Pushsum Algorithm on Topology, Follow the command:
gossipmain2:start(<no. Of nodes>, <Type of Topology>,<Type of Algorithm>).

Legend-

Type of Topology and algorithm-

i) Full Topology - Insert '**full**' in <Type of Topology>.

Eg. gossipmain2:start(5,full,gossip).

ii) Line Topology- Insert '**line**' in <Type of Topology>.

Eg. gossipmain2:start(5,line,pushsum).

iii) 2D Topology - Insert '**top_2D**' in <Type of Topology>.

Eg. gossipmain2:start(7,top_2D,pushsum).

ivi) Imperfect 3D Topology - Insert 'imp3D' in <Type of Topology>.

Eg. gossipmain2:start(7,imp3D,gossip).

6) Hereafter, The supervisor will select a random leader and start the process of rumours via gossip or pushsum algorithm.

Note- Before every execution, Run flush(). Command in the erlang environment to empty all registered processes.

What is working

We were able to implement all topologies i.e Line Topology, Full Topology, 2D topology and Imperfect 3D topology with both gossip and push sum algorithms. Using the Gossip algorithm, The convergence of all the nodes is achieved when each node hears the rumour 5 times. Using Push Sum Algorithm, The convergence of the nodes is achieved when each node has a change in s/w ratio of less than 10^{-10} in 3 consecutive rounds.

Gossip Algorithm Implementation:

a) Full Topology-

```
C:\Users\kotha\IdeaProjects\Gossipsecond>erl
Eshell V13.0.4 (abort with ^G)
1> gossipmain2:start(100,full,gossip).

2> Rumour is : Rumour
2> Node: <0.118.0> ----- Count : 0
2> Rumour is : Rumour
2> Node Terminated: <0.118.0>
2> total time: 377.958400 milliseconds
2> Convergence Achieved
```

b) Line Topology -

```
1> c(gossipmain2).
{ok,gossipmain2}
2> gossipmain2:start(100,line,gossip).
Supervisor actor <0.86.0>
running gossip
created worker <0.87.0>
created worker <0.88.0>

3> Rumour message is.
3> Rumour
3> Node to send.....
3> <0.186.0>
3> Node Terminated: <0.187.0>
3> total time: 57603.174400 milliseconds
3> Convergence Achieved
```

c) 2D topology -

```
Eshell V13.0.4 (abort with ^G)
1> c(gossipmain2).
{ok,gossipmain2}
2> gossipmain2:start(100,top_2D,gossip).
```

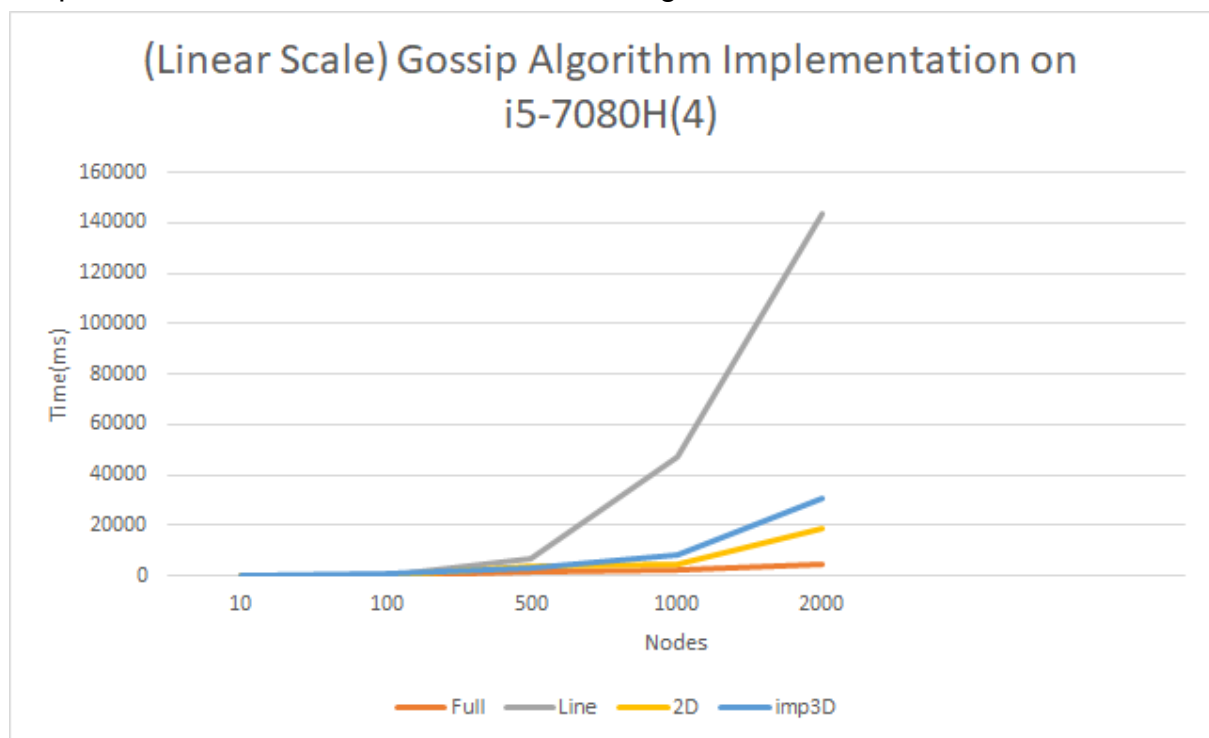
```
3> Node to send.....
3> <0.157.0>
3> Node Terminated: <0.97.0>
3> total time: 1102.336000 milliseconds
3> Convergence Achieved
```

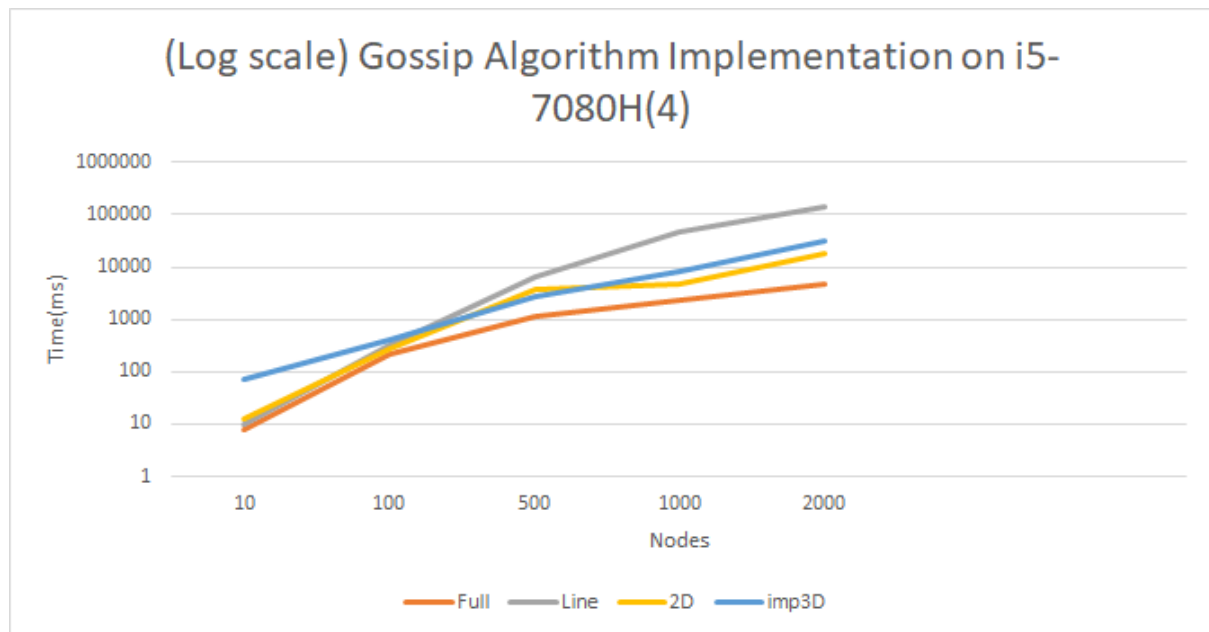
d) Imperfect 3D topology-

```
C:\Users\kotha\IdeaProjects\Gossipsecond>erl
Eshell V13.0.4 (abort with ^G)
1> gossipmain2:start(100,imp3D,gossip).
```

```
2> Node to send.....
2> <0.159.0>
2> Node Terminated: <0.84.0>
2> total time: 1409.228800 milliseconds
2> Convergence Achieved
```

Graph of Number of nodes v/s Time of Convergence





Pushsum Algorithm Implementation :-

a) Full Topology -

```
C:\Users\kotha\IdeaProjects\Gossipsecond>erl
Eshell V13.0.4 (abort with ^G)
1> gossipmain2:start(10,full,pushsum).
```

```
2> Node Terminated: <0.91.0>
2> total time: 1547.264000 milliseconds
2> Convergence Achieved
```

b)Line Topology -

```
C:\Users\kotha\IdeaProjects\Gossipsecond>erl
Eshell V13.0.4 (abort with ^G)
1> gossipmain2:start(10,line,pushsum).
```

```
2> Node to send.....
2> <0.87.0>
2> worker message received at node
2> <0.87.0>
2> Node Terminated: <0.87.0>
2> total time: 4801.331200 milliseconds
2> Convergence Achieved
```

c)2D Topology -

```
Eshell V13.0.4 (abort with ^G)
1> gossipmain2:start(10,top_2D,pushsum).
Supervisor actor <0.81.0>
Running 2D topology
running pushsum
2D topology node map
```

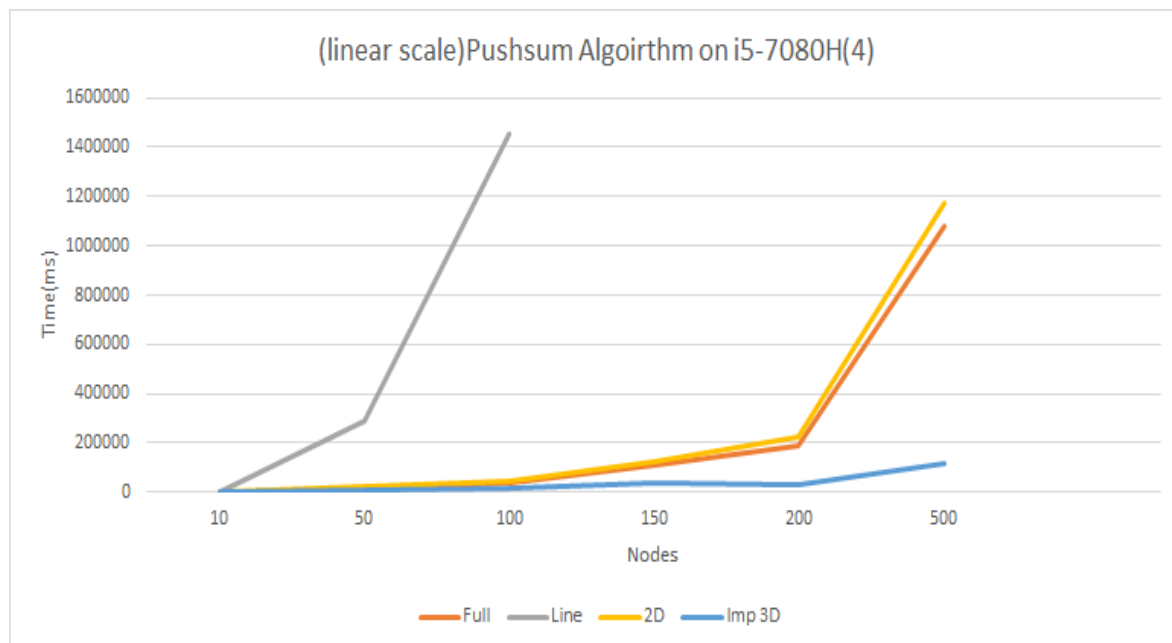
```
2> <0.89.0>
2> worker message received at node
2> <0.93.0>
2> Node Terminated: <0.93.0>
2> total time: 26755.174400 milliseconds
2> Convergence Achieved
```

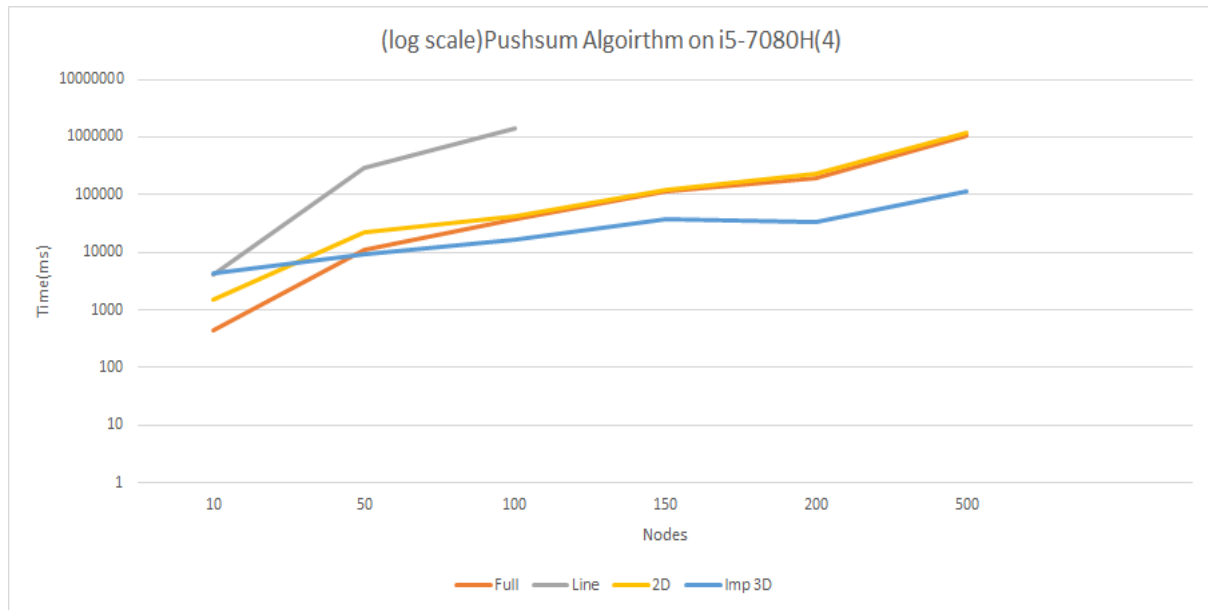
d)Imperfect 3D topology -

```
Eshell V13.0.4 (abort with ^G)
1> gossipmain2:start(10,imp3D,pushsum).
```

```
2> worker message received at node
2> <0.107.0>
2> worker message received at node
2> <0.95.0>
2> worker message received at node
2> <0.92.0>
2> Node Terminated: <0.92.0>
2> total time: 4808.704000 milliseconds
2> Convergence Achieved
```

Graph of Number of nodes v/s Time of convergence





Observations:

- When running Gossip, if we increase the rounds for each node to converge, the time to converge will also increase. Similarly, if we increase the threshold ratio of (S/W) , the time to converge for pushsum will also increase.
- Full topology has the highest number of neighbours for each node. So, for gossip algorithm we can observe that full topology converges the fastest. Line topology has the least number of neighbours for each node. So, it takes the longest time to converge.
- Line topology takes the longest time to converge even for Pushsum. And infact we could run it for a maximum of 100 nodes to get to convergence in a reasonable time. We could run it for a higher number of nodes and generate results in a reasonable time frame on a better configuration system. Another reason for the slow convergence lies in the way we ensure that the system does not go into a halt phase due to deleting nodes resulting in the breakage of the topology. In order to ensure that we do not have isolated nodes, we keep track of the neighbours of the nodes deleted and send a rumour to these neighbours randomly in order to ensure convergence of all the nodes and ultimately, the system.
- For pushsum, imperfect 3D topology converges faster than full topology. This is because of the random node that is introduced in

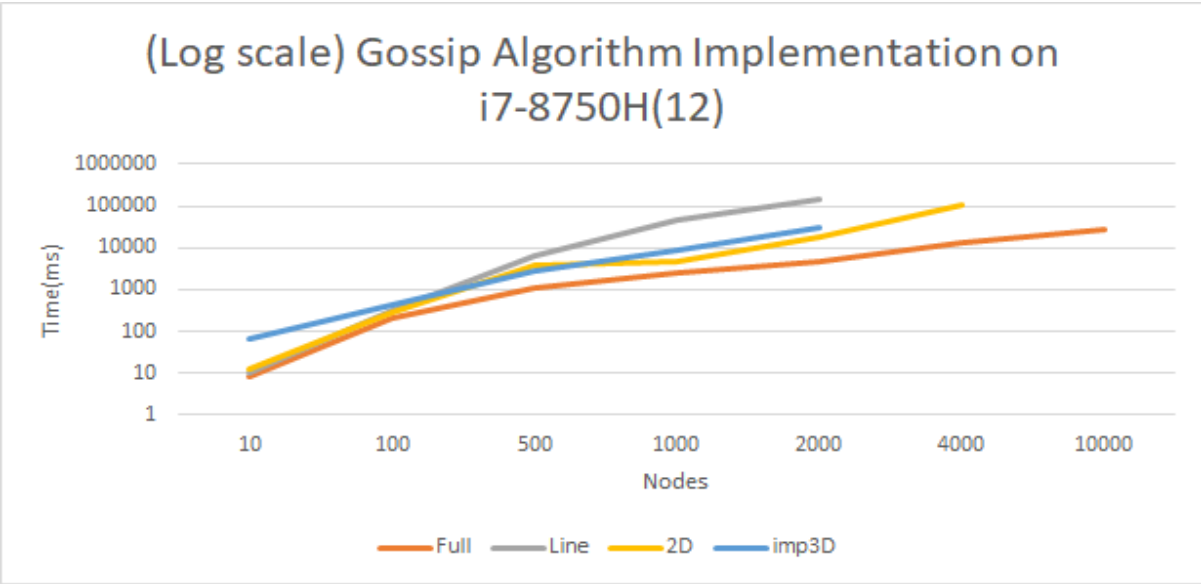
the imperfect topology, where apart from the usual neighbours that we select for each node, we also take an extra random node, which helps ensure faster convergence.

What is the largest network you managed to deal with for each type of topology and algorithm

The largest network We managed for each topology :-

Algorithm	Topology	Largest Network Tested
Gossip	Full	10000
Gossip	Line	2000
Gossip	2D	4000
Gossip	Imperfect 3D	2000
PushSum	Full	1000
PushSum	Line	200
PushSum	2D	500
PushSum	Imperfect 3D	1000

Graph of largest implemented algorithm network on Gossip-



Graph of largest implemented algorithm network on Pushsum-

