Report: Project 2 – Gossip Protocol in Scala (With Failure)

Students: Chiranjib Sur (UFID : 2531 4396)

Pramit Dutta (UFID : 7513 2433)

Course: COP 5615: Distributed Operating Systems Principles

The Gossip protocol, in this case, has been simulated with failure of node scheme where each now and then, there are failures in the nodes either in the form of communication or the physical node. There are some interesting facts that can be seen from the plotted graphs for the various network connection schemes of the protocol. The implementation has been tested for convergence of the message passing for a change in the number of nodes and for different protocol topology. They are for full network, line topology, 3D topology and improper 3D topology. The graphs plotted are mainly based on the time analysis for the convergence of the message passing between the nodes where there is a restriction that one node can transmit the message to its immediate neighbors for a limited number of times. However the simulation is also based on two different messages. One is based on just simple message and the second is based on computing the numerical value for the average of all the integers that are present in each individual nodes of the topology. The time analysis is also considered for this kind of difference in the purpose of the message sending and a comparative study has been made.

However the details of the analysis is discussed beneath the graph where it is plotted for each analysis and the details of the description of the analysis has been provided.

It is worthy to mention that there are various kind of observation being made for this failure model. Being built on the top of the non-failure model, this portion shares the same kind difficulties like the

Largest networks dealt with: Maximum number of nodes for which the Gossip Algorithm converged: (We tried with integers ranging from 10000 to 100000, increasing 10000 at a time, and recorded these floors convergences)

Gossip Algorithm:

Line topology- 60000 nodes (with 50% converge)

Full – 30000 nodes

3DGrid – 20000 nodes

Imperfect 3DGrid- 25000 nodes

Largest networks dealt with: Maximum number of nodes for which the PushSum Algorithm converged: (We tried with integers ranging from 10000 to 100000, increasing 10000 at a time and recorded these floors of convergences)

PushSum Algorithm:

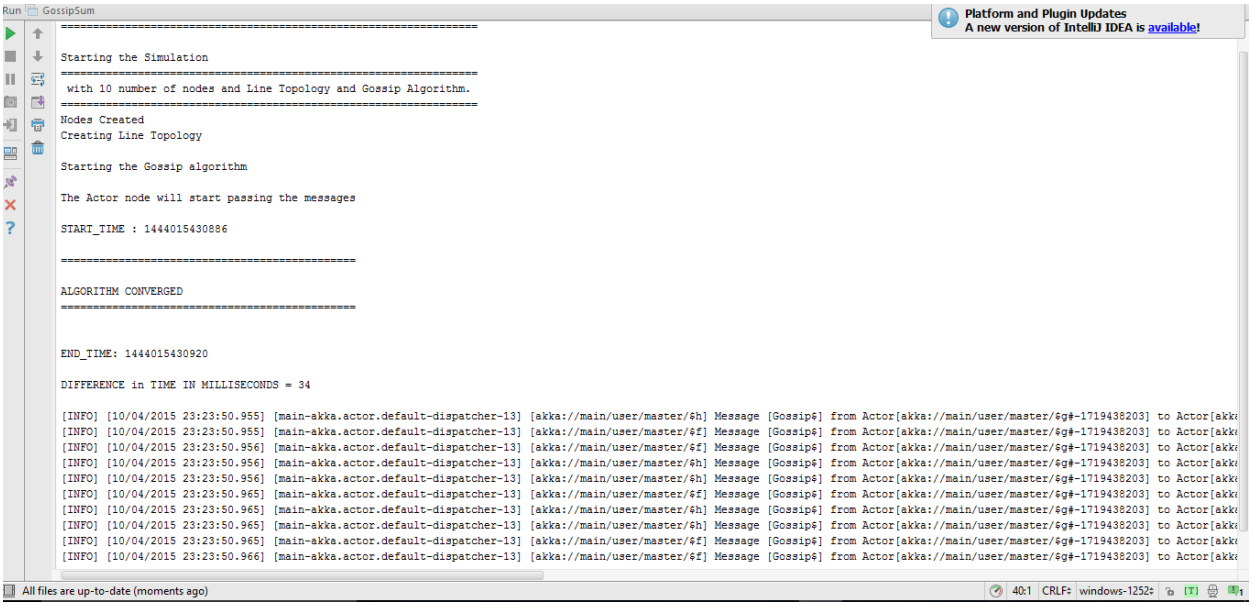
Line topology- 60000 nodes (with 50% converge)

Full – 30000 nodes

3DGrid – 20000 nodes

Imperfect 3DGrid- 25000 nodes

Snap Shots of the Program output



Results & Discussion:

In the following figure (Fig 1) we have plotted the time for convergence for the different protocols for the message passing system. It can be easily seen that the full topology performs worst though it has the best connectivity. This is due to communication overhead it has faced and there may be much traffic and loss of data. Whereas the line protocol has communication scarcity and is unable to process the spread throughout the network. Whereas the 3d and the improper 3d has been able to gather much momentum due to more number of connectivity throughout the system network. It can also be seen that for low number of nodes, the difference in time for convergence is much less and as the number of nodes increases, the difference increases. The difference in time of convergence between a 3D and Imperfect 3D topology remains almost the same but between a Full topology and a 3D/Imperfect 3D the difference in convergence time grows as the number of node increases, because the full topology has to pass message to all every possible node unlike the 3D/Imperfect3D topology.

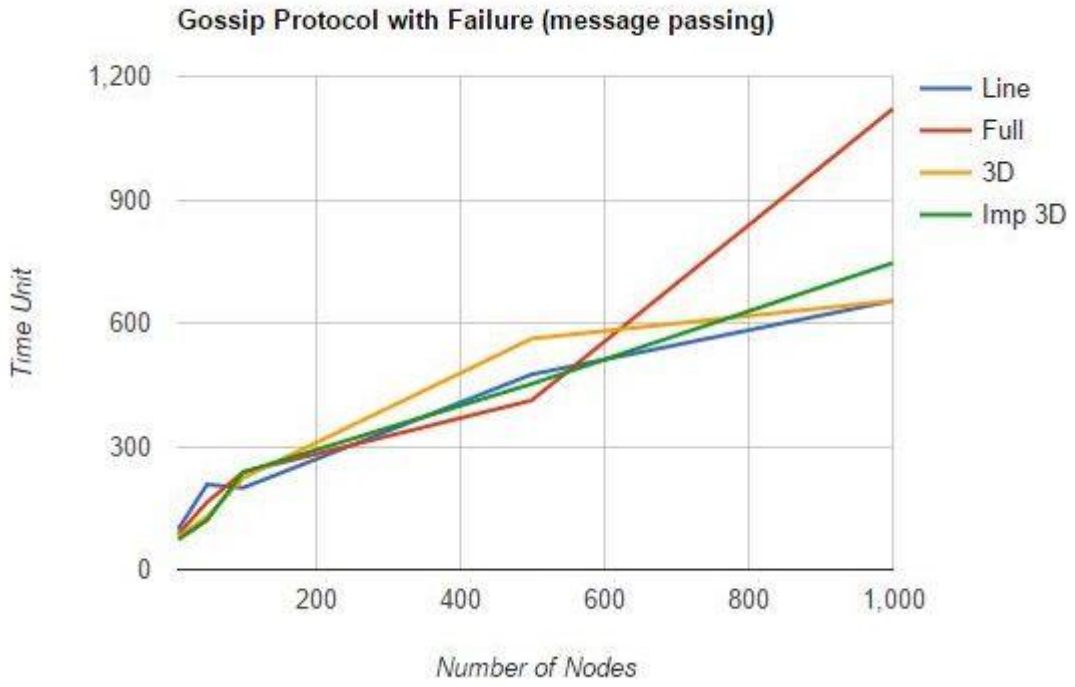


Fig 1: Plot for Different Topologies: Time vs Number of Nodes (Gossip)

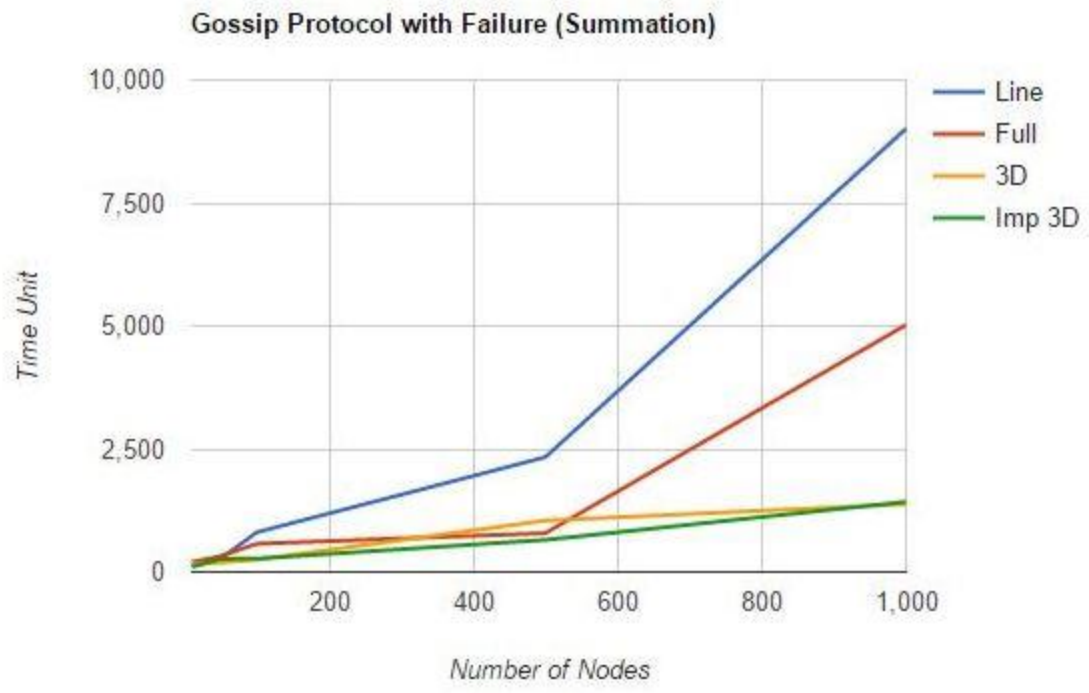


Fig 2: Plot for Different Topologies: Time vs Number of Nodes (PushSum)

Difficulty Faced:

While propagating the messages we faced an issue when we tried to pass message from one node to just another node. The Algorithms were not converging in that. When we increased the number of nodes a particular node can send messages to, the messages propagated fast and the algorithms converged. The challenge in the case of failure model implementation was also to converge the algorithm after the network failed.

- ➔ Failure without system call is difficult
- ➔ Regenerate after failure was not known
- ➔ Failure not only happens due to system call but also due to communication overhead
- ➔ Line topology fail a lot due to line breakage

Findings:

We found out that the line topology has a lower rate of convergence when the number of nodes increase that is because line topology has the slowest rate of message passing among all the topologies we implemented. 3D and the Imperfect 3D topology seems to perform better over here in case of aggregating using the Gossip protocol.

References : http://opendatastructures.org/ods-java/12_2_AdjacencyLists_Graph_a.html

http://alvinalexander.com/scala/iterating-scala-lists-foreach-for-comprehension