DISTRIBUTED OPERATING SYSTEM

PROJECT-2

**Group Members:**

Name: Shashank Bantakal UFID: 9592-6559

Name: Santhosh Suresh UFID: 5985-1919

**Implementation Details:**

This project implements Gossip & Push-Sum algorithms for line, full, Random2D, 3DTorus, Honeycomb and HoneycombRand topologies.

**Determination of Convergence Time:**

**GOSSIP ALGORITHM**

* In Gossip, a node stops transmitting once it has heard the gossip for 10 times.
* 3 different states are being maintained for each of the Node:-
* :oblivious, if the node has not received the gossip.
* :infected, if the node has received the gossip at-least once.
* :disconnected, if the node has received the gossip 10 times, it is marked disconnected from the network and will not further receive or send any messages.
* In this algorithm, the network is assumed to have converged if at-least 90% of the nodes have been marked as disconnected.

**PUSH SUM ALGORITHM**

* A network node is terminated when its s/w ratio did not change more than 10-10 in 3 consecutive rounds.
* Based on experiments with different node values for each topology, convergence percentage has been calculated and used in the application.

**Interesting Observations:**

* We expected network with full topology to converge soon for small (100-500) and medium sized (500-1000) network. But network with Line topology performed consistently better than other topologies for the above network size.
* 3DTorus, Random 2D and HoneycombRand show similar performance for smaller networks.
* 3DTorus is the most efficient of all the topologies for higher sized networks(>2000 nodes).
* Gossip network reached network convergence faster than Push-sum network.

**Graph of Convergence time vs size of the network:**

**GOSSIP ALGORITHM**

Here, Full topology takes the maximum time to converge followed by Line and Rand 2D for higher network size.

This could be due to the network structure wherein every node is connected to every other node in the network. Hence, for larger networks when neighbours are picked at random, the probability that a node is picked to send message is low when compared to a smaller network. This results in high convergence time.

The convergence time of topologies is as follows:

For Small Networks:

Line < Honeycomb < HoneycombRand < Rand 2D < 3D Torus < Full

For Large Networks:

3D Torus < HoneycombRand < Random 2D < Line < Honeycomb < Full

**PUSH SUM ALGORITHM**

Push-Sum follows a similar narrative as that of the gossip algorithm implementation.

3D Torus is the most efficient for larger networks and Full is the least efficient. This could be due to the fact that every node in 3D Torus has 6 neighbours, irrespective of the size of network.

The convergence time of topologies for high sized network in order are

3D Torus < HoneycombRand < Random 2D < Line < Honeycomb < Full

**BONUS - FAILURE NODES**

For the above experiment we maintained a constant number of node (500) and varied the percentage of failed nodes between 1 and 10.

We observed that for Full topology the convergence time increased with an increase in the number of failed nodes. Efficiency of other topologies didn’t change with increase in the number of failure nodes.

No network convergence was reached for line topology, as the network structure is such that message will only propagate if preceding and successive nodes are alive. Hence, in line topology failure of any nodes prevents further propagation of the message.