

DISTRIBUTED OPERATING SYSTEMS

PROJECT - 2

Team Members:

Daksh Gautam	11055891
Sarthak Chauhan	71835321

Algorithms Implemented

- Gossip
- Push-Sum

Topologies Implemented

- Full Network
- 3D Grid
- Random 2D Grid
- Sphere
- Line
- Imperfect Line

Convergence Criteria

Gossip

- We have decided to base our convergence on percentage of nodes achieving 10 messages. We set a threshold percentage at which we say the algorithm is converged and note the time to achieve that.
- This percentage is calculated by running the Gossip algorithm for each topology.
- In each topology at different number of nodes we note the percentage of nodes that receive the 10 messages.
- Out of these percentages we find the lowest percentage and set that as the threshold for that topology, as it will be achieved by all number of nodes for that topology.
- We then find the time at that percentage for each number of nodes for each topology.

Topology	Criteria
Full Network	When 95 percent of nodes achieve 10

	messages the algorithm is said to be converged the time is noted.
3D Grid	When 10 percent of nodes achieve 10 messages the algorithm is said to be converged the time is noted. Though different number of nodes showed convergence upto 35 percent.
Random 2D Grid	When 5 percent of nodes achieve 10 messages the algorithm is said to be converged the time is noted. Though increasing the number of nodes showed convergence upto 75 percent.
Sphere	When 10 percent of nodes achieve 10 messages the algorithm is said to be converged the time is noted. Though decreasing the number of nodes showed convergence upto 50 percent.
Line	When 10 percent of nodes achieve 10 messages the algorithm is said to be converged the time is noted.
Imperfect Line	When 10 percent of nodes achieve 10 messages the algorithm is said to be converged the time is noted.

-----Push Sum-----

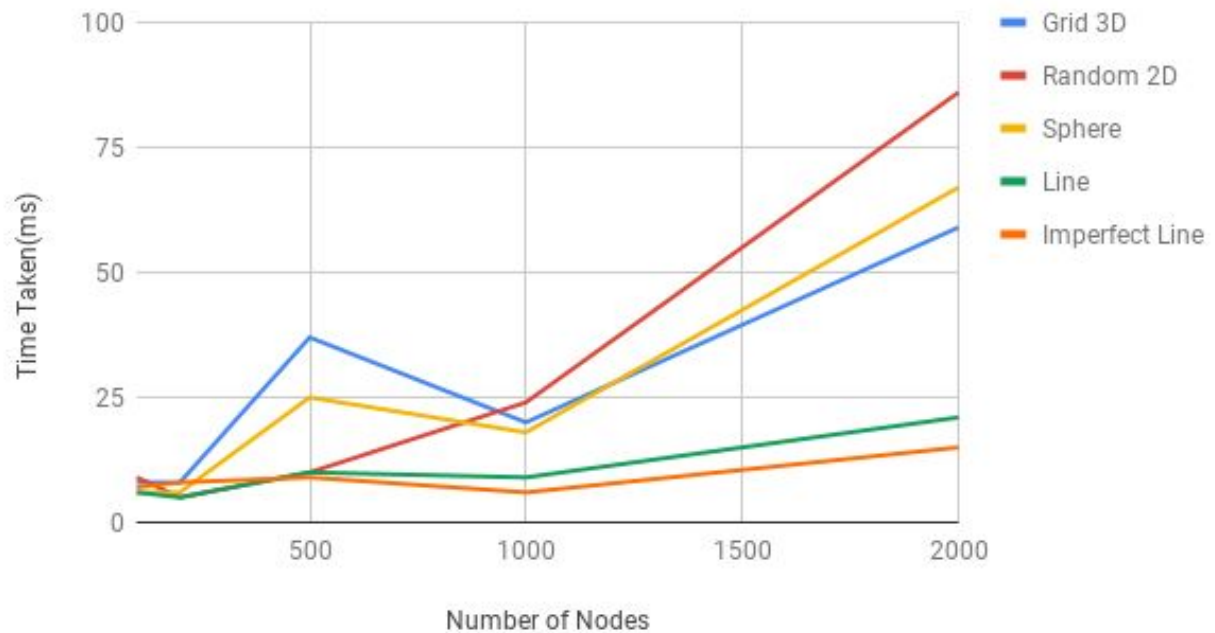
- We have decided to base our convergence on percentage of nodes achieving constant(difference less than 10^{-10}) ratio for s/w in 3 consecutive hits. We set a threshold percentage at which we say the algorithm is converged and note the time to achieve that for different number of nodes.
- This percentage is calculated by running the Push-Sum algorithm for each topology.
- In each topology at different number of nodes we note the percentage of nodes that achieve the constant ratio in 3 hits.
- Out of these percentages we find the lowest percentage and set that as the threshold for that topology, as this percentage will be achieved by all number of nodes for that topology.
- We then find the time at that percentage for each number of nodes.

Algorithm	Criteria
Full Network	When 95 percent of nodes achieve constant ratio the algorithm is said to be converged the time is noted.
3D Grid	When 15 percent of nodes achieve constant ratio the algorithm is said to be converged the time is noted. Though decreasing the number of nodes showed convergence upto 30 percent.
Random 2D Grid	When 10 percent of nodes achieve constant ratio the algorithm is said to be converged the time is noted. Though increasing the number of nodes showed convergence upto 78 percent.
Sphere	When 10 percent of nodes achieve constant ratio the algorithm is said to be converged the time is noted. Though decreasing the number of nodes showed convergence upto 40 percent.
Line	When 5 percent of nodes achieve constant ratio the algorithm is said to be converged the time is noted.
Imperfect Line	When 5 percent of nodes achieve constant ratio the algorithm is said to be converged the time is noted.

GRAPHS (Number of Nodes vs Time for Convergence)

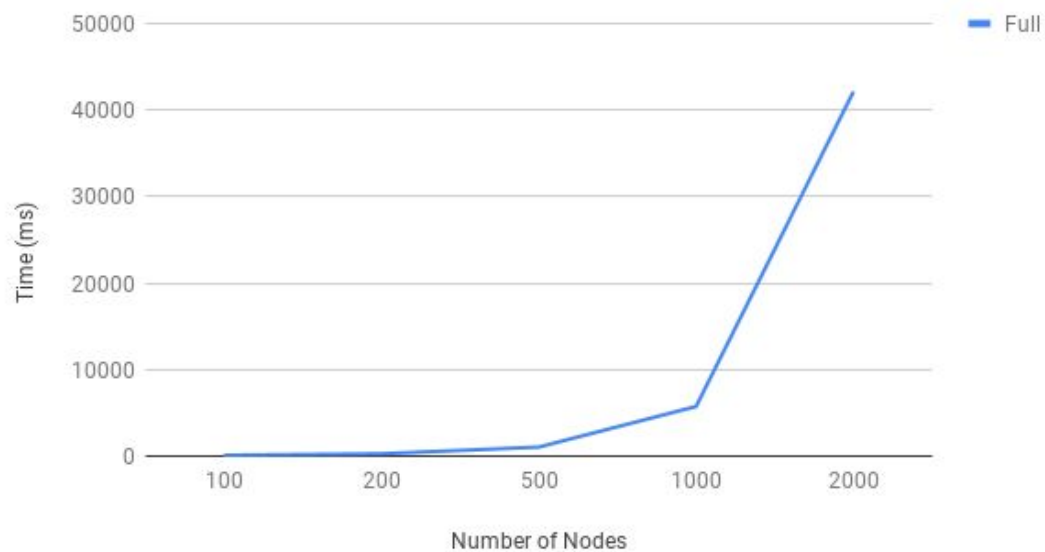
1. Gossip - Excluding Full Topology

Gossip - without Full



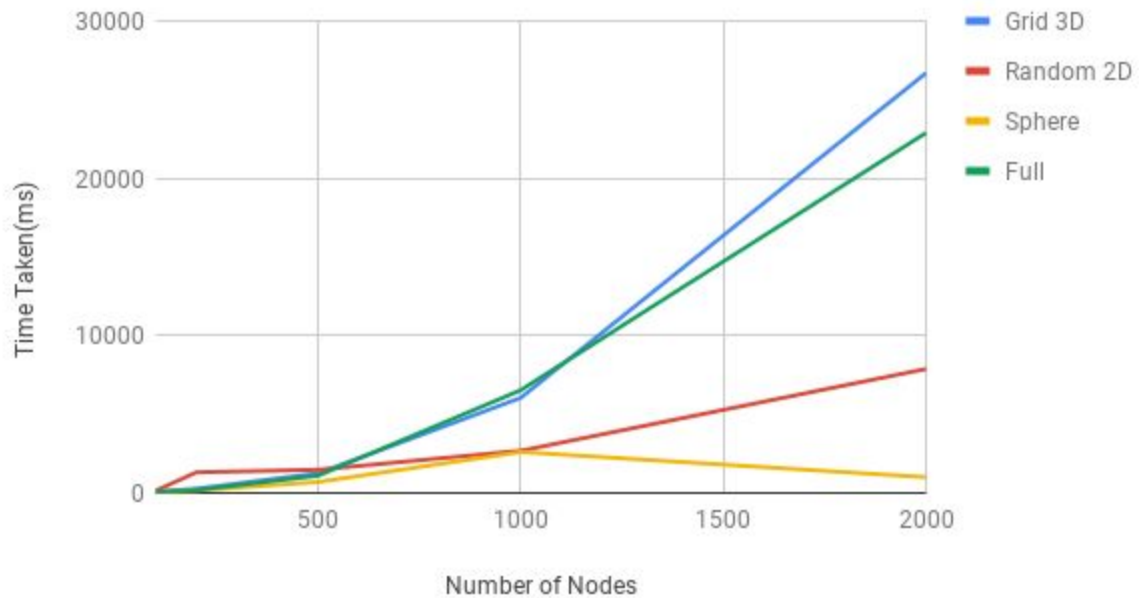
2. Gossip - Full Topology

Gossip - Full



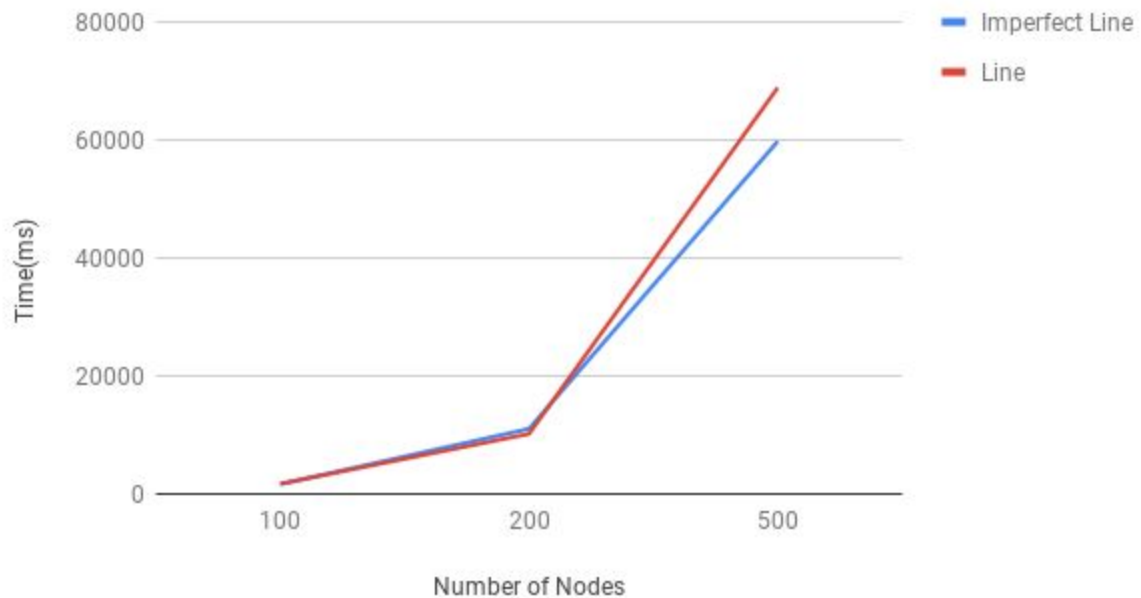
3. Push-Sum: Excluding Line and Imperfect Line

Push-Sum



4. Push-Sum: Line and Imperfect Line

Push-Sum: Imperfect Line, Line



Interesting Findings

1. In Gossip Algorithm and Push-Sum Algorithm in the Random 2D topology when we increase the number of nodes the convergence ratio starts to increase dramatically. This happens as we have fixed the distance to which we can find neighbours. As increasing the number of nodes leads to more number of nodes popping up in that distance the number of neighbours of each node increases and hence more convergence is achieved