Ankur Bagchi - 05549066
Saravanan Setty - 33299221

1. Introduction

To run the project, please use the command:

```
sbt "run <no of nodes> <topology> <algorithm> <optional inputs>"
```

Topologies: line, full, 3D, imp-3D
Algorithms: push-sum, gossip

Optional Inputs:

wf: Wildfires the message, sending it to all neighbors instead of 1. This only works for push-sum.
nb: No backsies, will not send the message back to the previous sender unless it is the only active neighbor left.

eg:

```
sbt "run 40 line gossip"
sbt "run 30 imp-3D push-sum wf nb"
sbt "run 20 full push-sum nb"
```

2. Topologies

The line topology connects the nodes in a straight line.

The full topology connects every node to every other node.

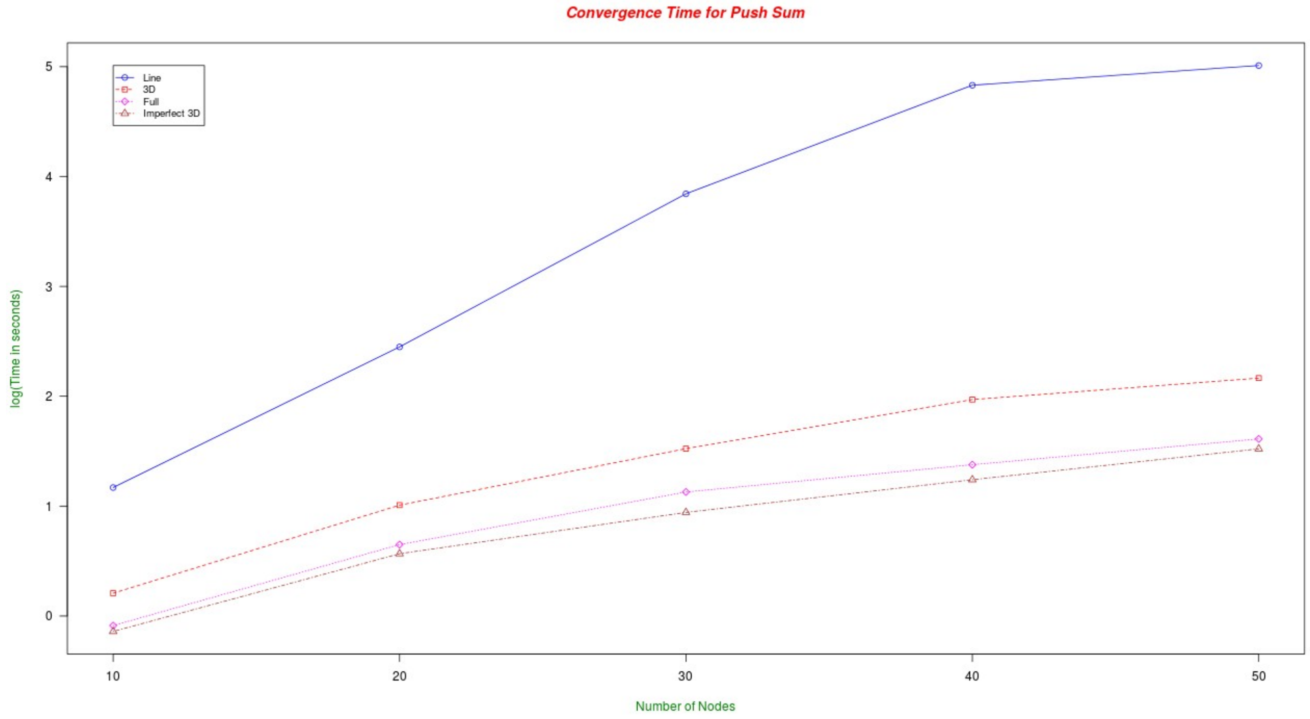
The 3D topology has been implemented as a cuboidal lattice with each member having upto 6 neighbors. The dimensions are taken as cuberoots of the provided number of nodes. The lattice is not filled completely if nodes < dimensional volume.

The imperfect 3D topology further connects each node to a random node that is not its neighbor, if such a node is available.

3. Observations

3.1 Push-Sum

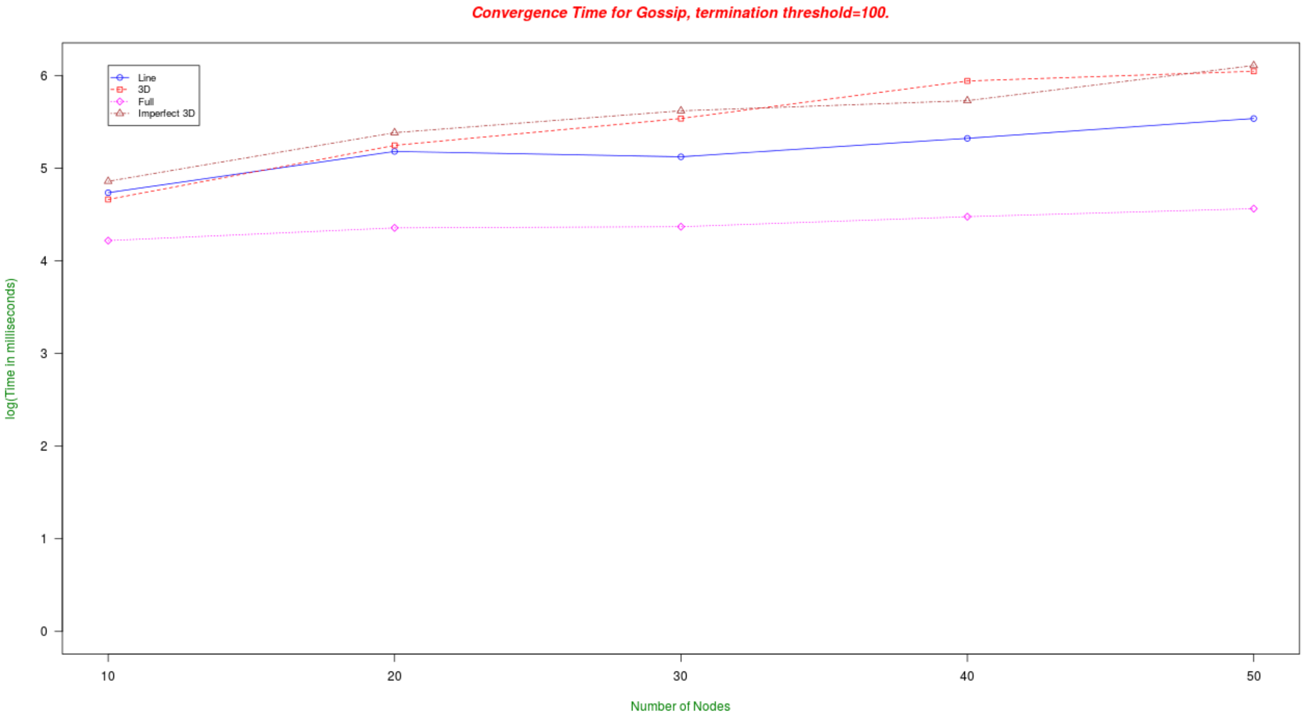
As seen on the graph below, the line topology takes the most time to converge, increasing exponentially as the number of nodes increase. All three of the other topologies take linear time to converge as nodes increase.



Interestingly, the imperfect 3D grid is the most efficient, even more so than the full topology, converging in linear time. Of the two remaining, the full topology is more efficient than the pure 3D grid.

3.2 Gossip

With a node dying at 10 messages, the algorithm converges very quickly and thus time taken is sensitive to background processes running on the system. Hence, we use 100 as the number in the graph below, plotting the time taken for each topology to converge.



Full topology here seems to be the most efficient, with the other topologies taking more time as number of nodes increase. Line levels out while 3D and imperfect 3D grids seem to be the least efficient at convergence.