Project 2

Chelsea Metcalf

October 1, 2015

1 ALGORITHM

1.1 Topology

Line: The actors are arranged in a line. Each actor has two neighbors, except for the first and last actor.
Full: An actor can send a message to any other actor.
3D: Actors are in a cube. If the given number of actors are not in a perfect cube, then the smallest cube size is found to fit in those nodes.
Imperfect 3D: Same as 3D except an actor can select a random node to send the message to.

1.2 Gossip

An actor receives a message and then sends it to a random neighbor. That actor will do the same. If an actor receives the message more than 10 times (gossipTermination), then that particular actor will terminate. If all the actors receive the message 10 times, then the whole system gets shutdown.

1.3 Push Sum

Once an actor(i) receives a message, it stores s(i) and w(i) values. It then selects any of its random neighbors(j) to send a message to. Once the message is received, actor(j) adds the received value to its corresponding value (s(i) + s(j)) and (w(i) + w(j)). This actor then again selects a random neighbor and sends half of the value of the above sum to that random actor. If an actor's ratio did not change for more than 10^{-10} in 3 consecutive rounds, then the actor terminates.

2 RESULTS

I measured and graphed the convergence by amount of ticks. Since the code is pretty efficient, the actual ms time doesn't vary that much. By using the tick count, I measured every time the worker comes back to the master, and

the master tells the worker to send the message. For example, when I ran my algorithm with gossipTermination set at 10, and 64 nodes, it took 24 ticks for line and 15 ticks for full. You can see the difference in the amount of ticks, but the time it took to run the program for each was about 100 ms.

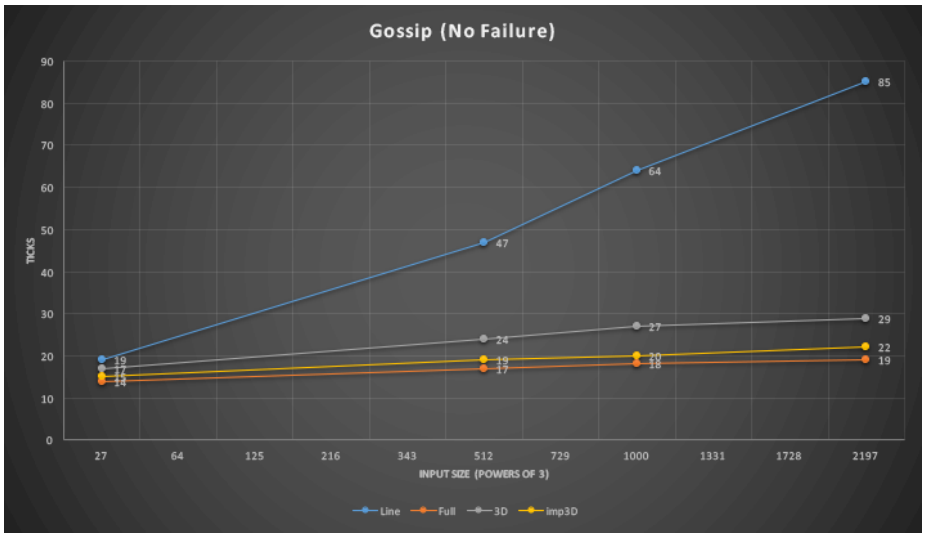

Figure 1: **Gossip - No Failure.** From this graph, it is shown that line has the slowest convergence time. In order from slowest to fastest: line → 3D → imperfect 3D → full

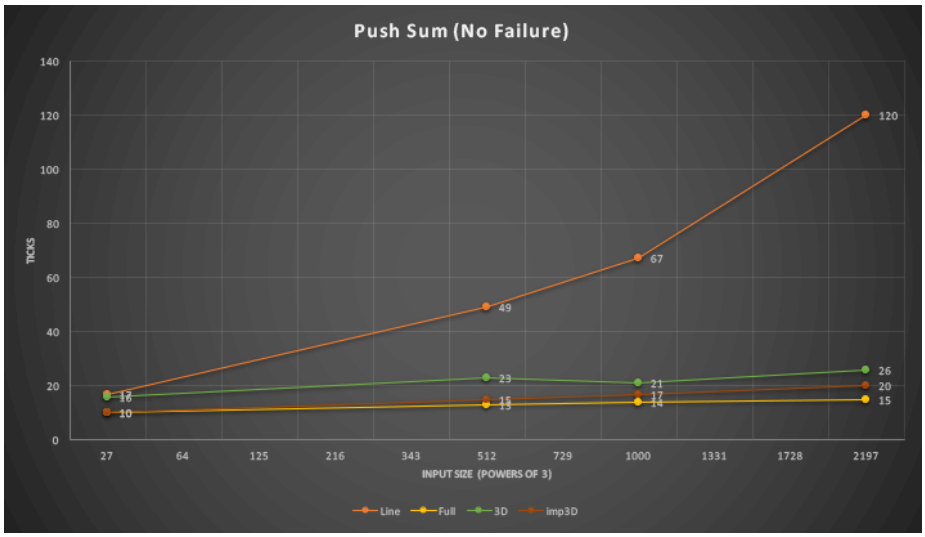

Figure 2: **Push Sum - No Failure.** From this graph, it is shown that line still has the slowest convergence time. Imperfect 3D and full are very fast and have a similar convergence time.

3 OBSERVATIONS

The type of topology made a big difference in convergence time. The line was slowest to converge. For example: We have nodes 0, 1, 2, 3, 4. Node 0 starts with a message and sends it to Node 1. Node 1 has a 50 percent chance to send the message to a node that hasn't heard the message yet. So, with every tick, there is only a 50 percent chance of making progress in the network.

In my graphs, I inputted values that were cubed (3^3, 8^3, ...). The reason for this is because 3D and imperfect 3D have no choice but to construct a cube from the lowest cube factor size. In order to test consistently, I used the same number of nodes for the line and full networks.

I have another project folder called "project2oldVersion" and you can run the jar and view the code for that implementation. It's the first way I implemented Gossip and Push Sum. I got around the same timing values for each topology. This was because I wasn't measuring it by ticks at first. Also, all my workers were talking to each other and not going back to the master. In my new code, all the workers go back to the master. The master acts like a coordinator.