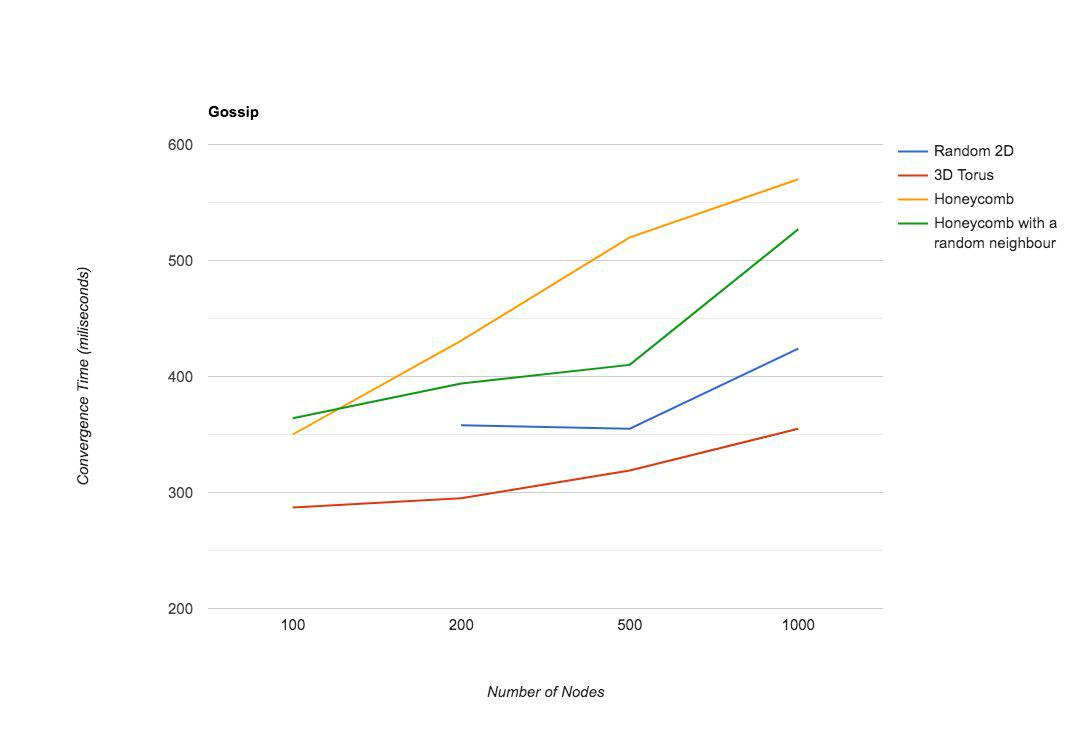
* We have implemented both ​**Push-sum**​and ​**Gossip** ​algorithms with all topologies mentioned in the project which are :
  + Full Network
  + Line
  + Random 2D Grid
  + 3D torus Grid
  + Honeycomb
  + Honeycomb with random a neighbour
* **Instructions for running the code :**
  + For linux/Mac
    - Go the project directory
    - mix escript.build (Optional)
    - ./proj2 100 line push-sum
    - First command line argument is the number of nodes
    - Second command line argument is the type of topology you choose
    - Third command line argument is the name of the algorithm you choose
    - Abstract command: ./proj2 <numOfnode> <topology> <algorithm>
  + For windows
    - Go the project directory
    - mix escript.build (Optional)
    - escript .\proj2 100 line push-sum
    - First command line argument is the number of nodes
    - Second command line argument is the type of topology you choose
    - Third command line argument is the name of the algorithm you choose
    - Abstract command: .\proj2 <numOfnode> <topology> <algorithm>

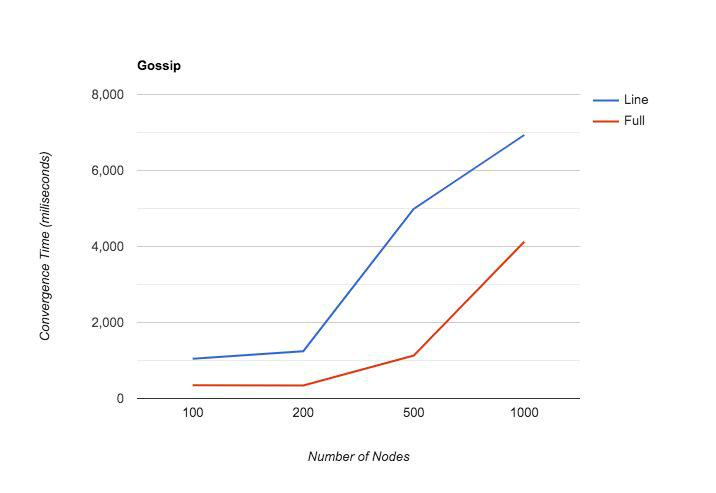


* **Largest values for each topology are :**

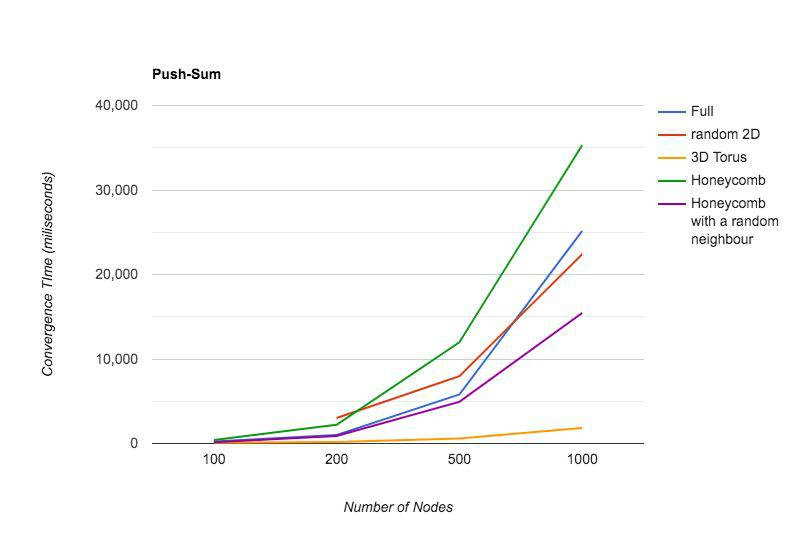
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Topology** | | **Number of Node** | | |
|  |  |  |  |  |  |
| Full Network | | |  |  |  |
|  |  |  |  |  |  |
|  | Line | |  |  |  |
|  |  |  |  |  |  |
| Random 2D | | |  |  |  |
|  |  |  |  |  |  |
| 3D torus Grid | | |  |  |  |
|  |  |  |  |  |  |
| Honeycomb | | | 50000 | |  |
|  |  |  |  |  |  |
| Honeycomb with a random neighbour | | | 50000 | |  |
|  |  |  |  |  |  |

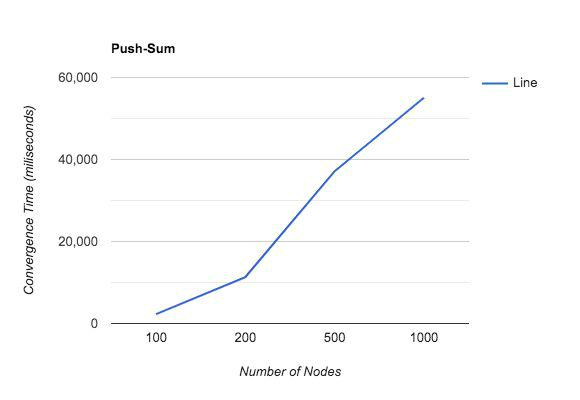
**●** **Gossip algorithm graphs**





**● Push-Sum Graphs**





* **Interesting findings :**
  + Time taken by ​**Line** ​topology​​was maximum in both algorithms.
  + In case of Honeycomb and Honeycomb with random neighbour​**,** **Honeycomb with random neighbour** ​took less time in both thealgorithms.
  + Random 2D because of random coordinate allocation to processes and specific distance (0.1) constraint, it mostly does not even start for less number of nodes. It happens because less number of nodes are sparsed in a comparatively large area, so the starting process might not assigned any neighbours.
  + As the number of nodes increases, change in time taken is minimum for 3D torus.
  + From all the topologies, 3D torus took less time and is most efficient topology as far as time is concerned.