Output for Connected Components:

[[nirali09@cssmpi1h java]$ mpirun -np 1 java ConnectedComponentsMPI 10componentgraph5000input.txt 5000 > output.txt

Number of connected components : 10

Total time elapsed = 232 msec

[nirali09@cssmpi1h java]$ mpirun -np 2 java ConnectedComponentsMPI 10componentgraph5000input.txt 5000 > output2.txt

Number of connected components : 10

Total time elapsed = 1006 msec

[[nirali09@cssmpi1h java]$ mpirun -np 3 java ConnectedComponentsMPI 10componentgraph5000input.txt 5000 > output3.txt

Number of connected components : 10

Total time elapsed = 798 msec

[nirali09@cssmpi1h java]$ mpirun -np 4 java ConnectedComponentsMPI 10componentgraph5000input.txt 5000 > output4.txt

Number of connected components : 10

Total time elapsed = 788 msec

output table:

|  |  |
| --- | --- |
| #nodes | Elapsed Time in msec |
| 1 | 232 |
| 2 | 1006 |
| 3 | 798 |
| 4 | 788 |

As sequentially program is running faster as there is no data send and receive overhead. Performance improvement is calculated between 2 and 4 nodes runtime.

Performance Improvement as compared to 2 nodes and 4 nodes:

1006/788 = 1.276