Note: The input for testing taken by us **0.125 million files per 100 MB**.

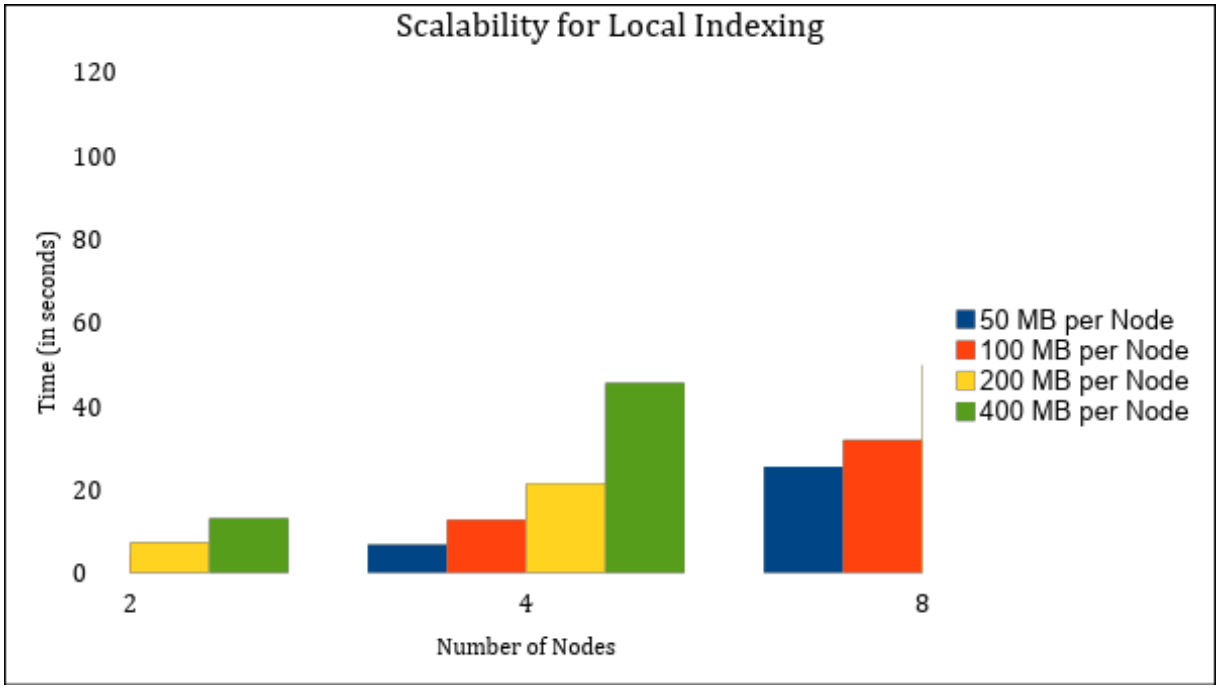
Local Indexing:

Number of Nodes vs. Input Size per Node

	50 MB per Node	100 MB per Node	200 MB per Node	400 MB per Node
1	29.724526	61.524275	87.795834	656.170232
2	27.996064	54.318103	110.266355	690.347058
4	27.104765	54.358388	102.194582	663.294917
8	26.763031	57.974508	113.047895	676.528335

Time (in seconds)

As the number of input files are very large, so time taken for doing I/O to read each file is very large and is included in Local Indexing Time Calculations., and that's why Local Indexing is taking so much time, for example on input of 400 MB on each node means there are 0.5 million files on each node.

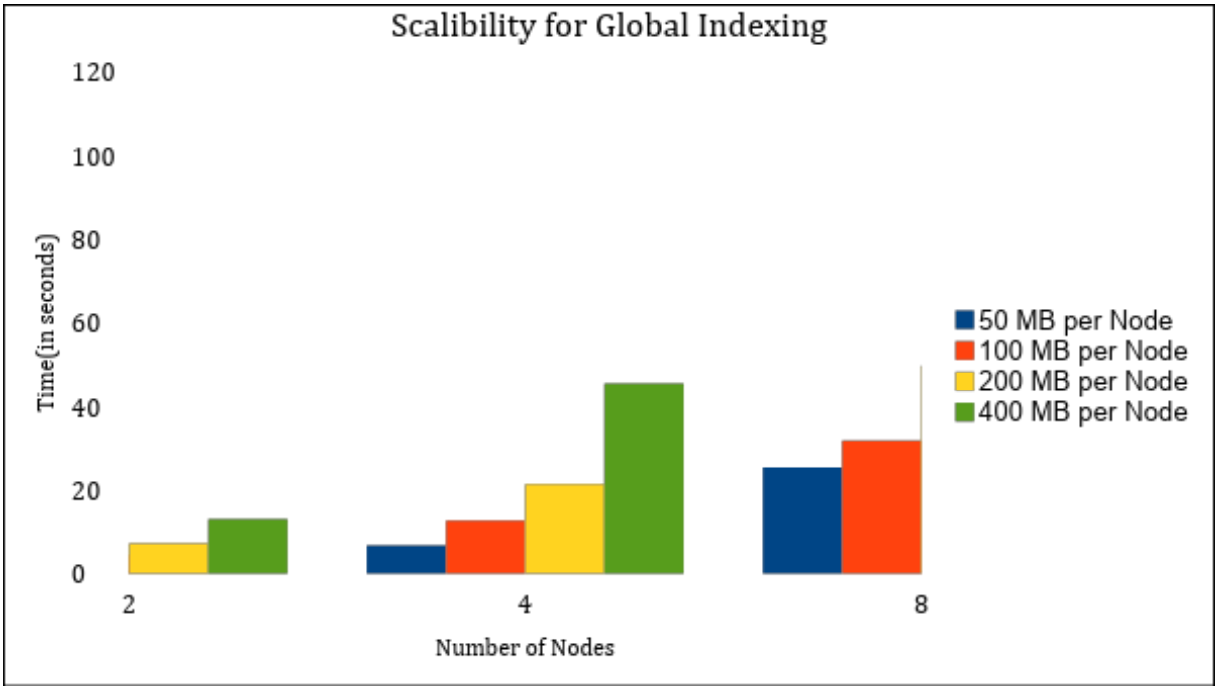


Global Indexing:

After the creation of Local Indedxing, due to efficient communication and design, Global Indexing on input as large as 3.2 GB takes only 107.57769 seconds.

Number of Nodes vs. Input Size per Node

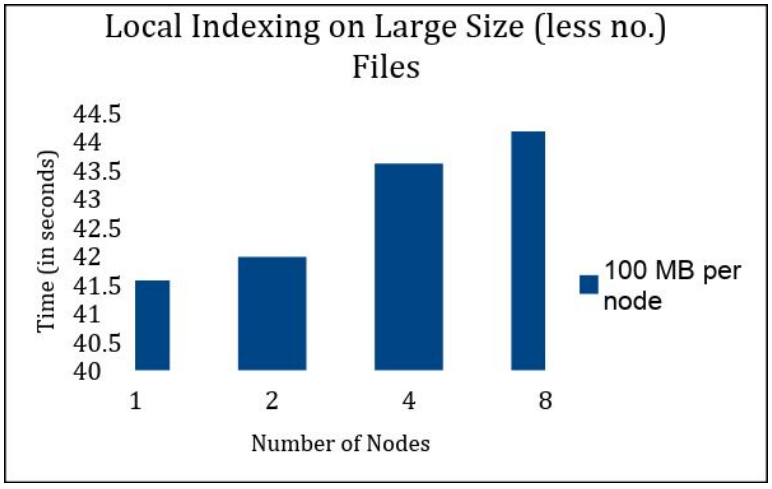
	50 MB per Node	100 MB per Node	200 MB per Node	400 MB per Node
2	1.214269	4.462831	7.319010	13.114381
4	6.841083	12.701781	21.381423	45.485292
8	25.449537	31.824969	49.695247	107.57769



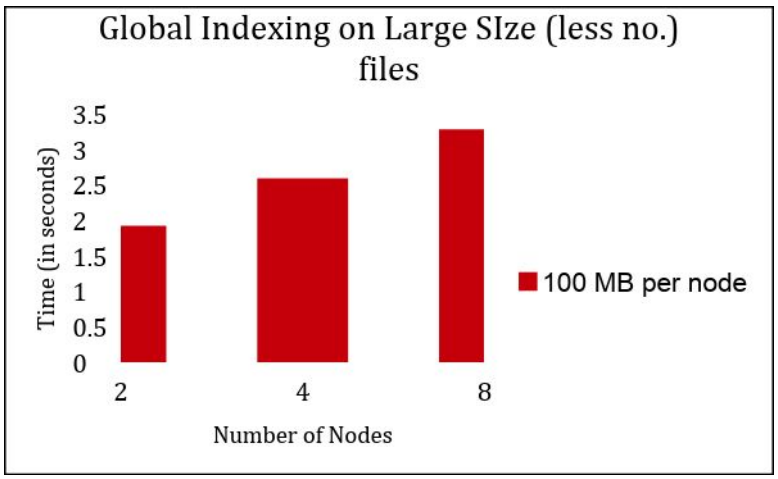
Time (in seconds)

We also ran our code on less no of files but with large size, and results are as follows:

Local Indexing:



Global Indexing:


Global Indexing on Large SIze (less no.) files
Time (in seconds)
3.5
3
2.5
2
1.5
1
0.5
0
2
4
8
Number of Nodes
100 MB per node