- Test script : **test.sh**
- Test log files are placed in **test_log** folder in the source code
- **time** command of Linux is used to calculate the performance
- Buffer size is set to a maximum of **4096** bytes for every read / write
- Every access is repeated **200** times
- For multiple clients case, we ran tests with **2 or 3** clients accessing simultaneously.

Test Case 1 : Relay Test Case

compute-0-0 <-----> compute-0-1 <-----> compute-0-2 <------> compute-0-3

Here we try to search a file from compute-0-0 and the file is served by compute-0-3. This is to test that the request are relayed properly.

Relay			
File Size	Requestor	Provider	Time
1k	compute-0-0	compute-0-3	34.53
2k	compute-0-0	compute-0-1, 0-2	0.91
4k	compute-0-0	compute-0-1	9.03
8k	compute-0-0	compute-0-3	17.44
Multiple requests			
8k	Compute-0-0, 0-1, 0-2	compute-0-3	17.438 / 17.458

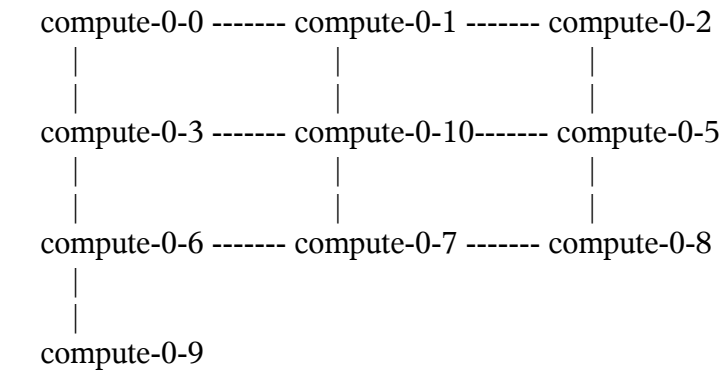
Test Case 2 : Star Test Case with 4 nodes

compute-0-0 - Peers: compute-0-1, compute-0-2, compute-0-3
compute-0-1 - Peers: compute-0-0, compute-0-2, compute-0-3
compute-0-2 - Peers: compute-0-0, compute-0-1, compute-0-3
compute-0-3 - Peers: compute-0-0, compute-0-1, compute-0-2

Star-4			
File Size	Requestor	Provider	Time
1k	compute-0-0	compute-0-3	34.28
2k	compute-0-0	compute-0-1, 0-2	1.32
4k	compute-0-0	compute-0-1	9.81
8k	compute-0-0	compute-0-3	17.63
Multiple requests			
8k	Compute-0-0, 0-1, 0-2	compute-0-3	17.711 / 17.685

Test Case 3 : Mesh Test Case with 10 nodes

10 Nodes : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-8, compute-0-9

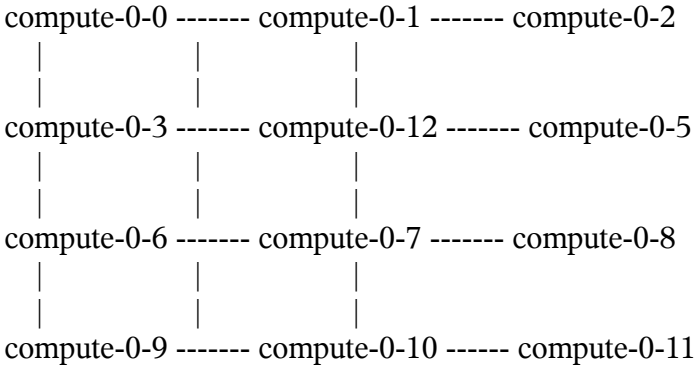


compute-0-0 - Peers : compute-0-1, compute-0-3
compute-0-1 - Peers : compute-0-0, compute-0-10, compute-0-2
compute-0-2 - Peers : compute-0-1, compute-0-5
compute-0-3 - Peers : compute-0-0, compute-0-10, compute-0-6
compute-0-10 - Peers : compute-0-1, compute-0-5, compute-0-7, compute-0-3
compute-0-5 - Peers : compute-0-2, compute-0-10, compute-0-8
compute-0-6 - Peers : compute-0-3, compute-0-7, compute-0-9
compute-0-7 - Peers : compute-0-6, compute-0-10, compute-0-8
compute-0-8 - Peers : compute-0-5, compute-0-7
compute-0-9 - Peers : compute-0-6

Mesh-10			
File Size	Requestor	Provider	Time
1k	compute-0-1	compute-0-3	34.08
2k	compute-0-3	compute-0-10	1.76
4k	compute-0-7	compute-0-1	9.56
8k	compute-0-0	compute-0-9	17.35
Multiple requests			
8k	compute-0-0, 0-1	compute-0-9	17.612 / 17.975

Test Case 4 : Mesh Test Case with 12 nodes

10 Nodes : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-8, compute-0-9, compute-0-11, compute-0-12



compute-0-0 - Peers : compute-0-1, compute-0-3
compute-0-1 - Peers : compute-0-0, compute-0-12, compute-0-2
compute-0-2 - Peers : compute-0-1, compute-0-5
compute-0-3 - Peers : compute-0-0, compute-0-12, compute-0-6
compute-0-12 - Peers : compute-0-1, compute-0-5, compute-0-7, compute-0-3
compute-0-5 - Peers : compute-0-2, compute-0-12, compute-0-8
compute-0-6 - Peers : compute-0-3, compute-0-7, compute-0-9
compute-0-7 - Peers : compute-0-6, compute-0-12, compute-0-8, compute-0-10
compute-0-8 - Peers : compute-0-5, compute-0-7, compute-0-11
compute-0-9 - Peers : compute-0-6, compute-0-10
compute-0-10 - Peers : compute-0-9, compute-0-7, compute-0-11
compute-0-11 - Peers : compute-0-8, compute-0-10

Mesh-12			
File Size	Requestor	Provider	Time
1k	compute-0-1	compute-0-3	34.14
2k	compute-0-3	compute-0-10	7.44
4k	compute-0-0	compute-0-1	9.75
8k	compute-0-0	compute-0-9	18.03
Multiple requests			
8k	compute-0-0, 0-1	compute-0-3	17.752 / 17.891

Test Case 5 : Star Test Case with 10 nodes

10 Nodes : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-8, compute-0-9

compute-0-0 - Peers : compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-8, compute-0-9
compute-0-1 - Peers : compute-0-0, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-8, compute-0-9
compute-0-2 - Peers : compute-0-0, compute-0-1, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-8, compute-0-9
compute-0-3 - Peers : compute-0-0, compute-0-1, compute-0-2, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-8, compute-0-9
compute-0-10 - Peers : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-5, compute-0-6, compute-0-7, compute-0-8, compute-0-9
compute-0-5 - Peers : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-6, compute-0-7, compute-0-8, compute-0-9
compute-0-6 - Peers : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-7, compute-0-8, compute-0-9
compute-0-7 - Peers : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-8, compute-0-9
compute-0-8 - Peers : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-9
compute-0-9 - Peers : compute-0-0, compute-0-1, compute-0-2, compute-0-3, compute-0-10, compute-0-5, compute-0-6, compute-0-7, compute-0-8

Star-10			
File Size	Requestor	Provider	Time
1k	compute-0-0	compute-0-3	36.28
2k	compute-0-0	compute-0-10	3.66
4k	compute-0-0	compute-0-1	11.86
8k	compute-0-0	compute-0-9	19.87
Multiple requests			
8k	compute-0-0, 0-1	compute-0-3	19.911 / 19.885

Analysis :

In PA2 number of RPC calls made is very high compared to number of calls made in PA1.

Number of RPC calls made to fetch the file remains the same in PA1 and PA2. This is dependent on the size of the file. In each call, 4096 bytes of data is read and transferred.

In PA1, one RPC call was made in the whole system to get the location of the file. All subsequent calls were related to actual file transfer. In PA2, no of RPC calls to get the file location is comparatively higher than PA1. On that note, time taken to find the file location is relatively higher than PA1.

In spite of having a MAXTTL of 4 and a broadcast to all once the file is found, number of search calls exchanged is quite high .

However we do not depend on one node to give us the file location. Index server is distributed and index server is no more a bottle neck. With our caching feature that we have in our project, time taken to search a file should get reduced as time progresses.