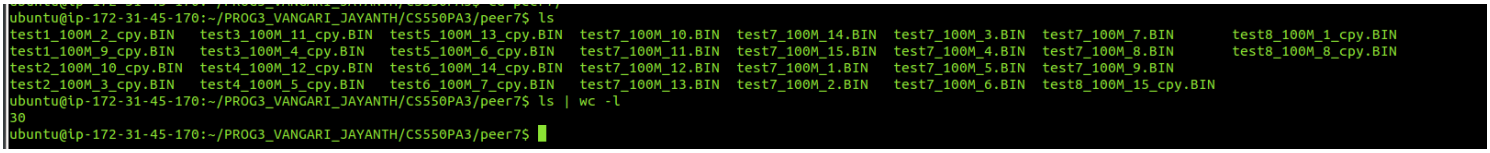
CS550 Advanced Operating Systems
Programming Assignment 3

Evaluation

submitted by:
Jayanth Vangari
A20337867

The evaluation of performance for the decentralized P2P system is done on Amazon ec2 m3.large instances essentially on binary files.



The Evaluation involves 2 experiments:

1. Evaluating register , search and download operations on 100kB files on 1 ,2 , 4 and 8 nodes
2. Evaluation throughput that can be achieved on file sizes ranging from 1kB, 10KB ,100KB ,1M , 10M , 100M and 1GB on 8 nodes.

Experiment 1 :

a. Registration :

single node: The time taken to register 10000 100KB files on a single node is **2127millisecs**.

Throughput (Op/sec)=4701

two nodes : The time taken to register 10000 100KB files on each of 2 nodes is :

node 1 - 4053 millisecs

node 2 - 4201 millisecs

Average time taken by a node to make 10000 registrations is = **4127 millisecs**

Client	Throughput (Ops/sec)
1	2467
2	2380

Aggregate Throughput (Ops/sec) = 2467+2380

=4847

four nodes : The time taken to register 10000 100KB files on each of four nodes is :

node 1 - 6157 millisecs

node 2 - 5877 millisecs

node 3 - 5531 millisecs

node 4 – 5999 millisecs

Average time taken by a node to make 10000 registrations is :

= **5891 millisecs**

Client	Throughput (Ops/sec)
1	1624
2	1701
3	1807
4	1666

Aggregate Throughput (Ops/sec) = 1624+1701+1801+1666
=6798

eight nodes : The time taken to register 10000 100KB files on each of eight nodes is :

node 1 - 7560 millisecs
node 2 - 7550 millisecs
node 3 - 7847 millisecs
node 4 – 7649 millisecs
node 5 - 6582 millisecs
node 6 – 7448 millisecs
node 7 - 7083 millisecs
node 8 – 7649 millisecs

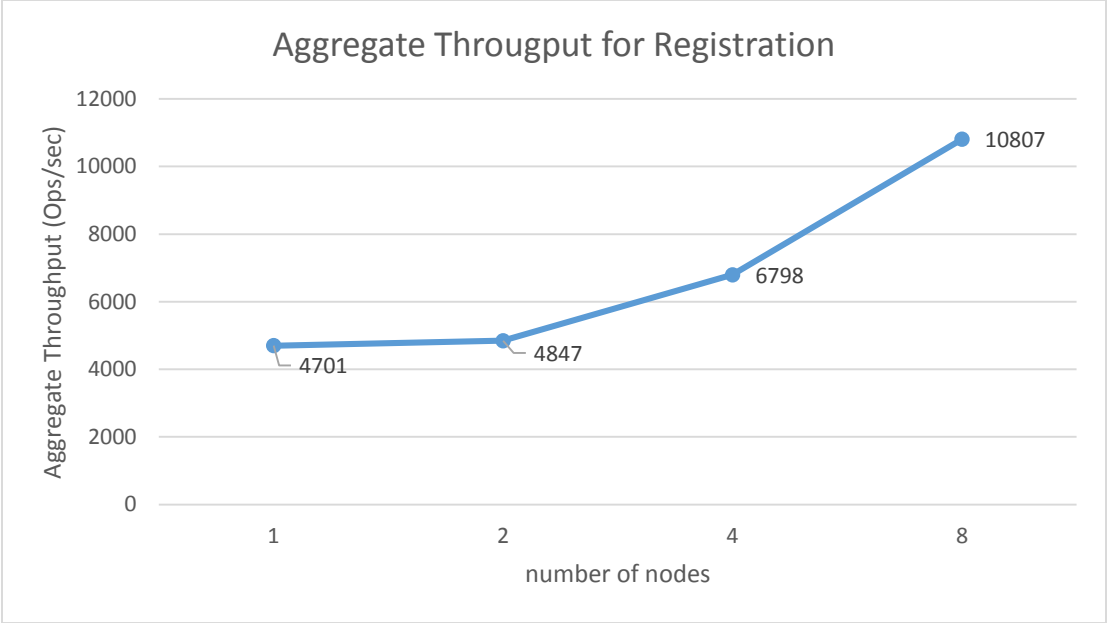
Average time taken by a node to make 10000 registrations is :

= **7421 millisecs**

a. **Computing aggregate Throughput in (registration) Operations per second for all 8 clients:**

Client	Throughput (Ops/sec)
1	1322
2	1324
3	1274
4	1307
5	1519
6	1342
7	1411
8	1308

Aggregate Throughput (Ops/sec) = 1322+1324+1274+1307+1519+1342+1411+1308
=10807



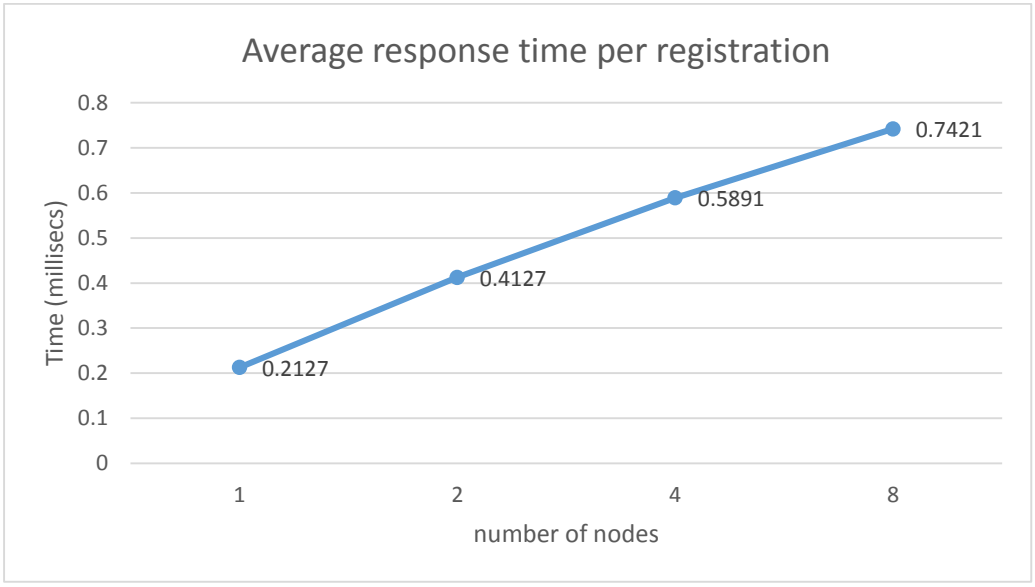
a. **Average response time per registration Operation :**

Average time taken for a single node per registration : 2127/10000
:0.2127 millisecs

Average time taken for two concurrent nodes per registration : 4127/10000
:0.4127 millisecs

Average time taken for four concurrent nodes per registration :5891/10000
: 0.5891 millisecs

Average time taken for eight concurrent nodes per registration :7421/10000
: 0.7421 millisecs



Explanation: The average response time for registration increases as the number of nodes in the System increases since , when a peer is registering its files , it does the registration at multiple nodes . I.e., the hash function directs the file distribution of registration evenly across all the peers so communicating to all the peers in the system during registration and all the clients during this concurrently increase the average response time. And also it is obvious that the aggregate throughput increases with number of concurrent clients.

b. Search :

single node: The time taken to search 10000 100KB files by a single node is **2248 millisecs**.

Aggregate Throughput (Ops/sec)=4448

two nodes : The time taken to search 10000 100KB files by each of 2 nodes is :

node 1 - 3080millisecs

node 2 - 3054 millisecs

Average time taken by a node to make 10000 searches is:

= **3067 millisecs**

Client	Throughput (Ops/sec)
1	3260
2	3274

Aggregate Throughput (Ops/sec) = 3260+3274

6534

four nodes : The time taken to search 10000 100KB files by each of four nodes is :

node 1 - 5076 millisecs

node 2 - 5115 millisecs

node 3 - 4941 millisecs

node 4 – 5054 millisecs

Average time taken by a node to make 10000 searches is :

= **5204 millisecs**

Client	Throughput (Ops/sec)
1	1970
2	1954
3	2023
4	1978

Aggregate Throughput (Ops/sec) =1970+1954+2023+1978

7925

eight nodes : The time taken to search 10000 100KB files by each of eight nodes is :

node 1 - 7879 millisecs
node 2 - 7754 millisecs
node 3 - 8056 millisecs
node 4 – 8492 millisecs
node 5 - 7158 millisecs
node 6 – 7726 millisecs
node 7 - 7364 millisecs
node 8 – 8397 millisecs

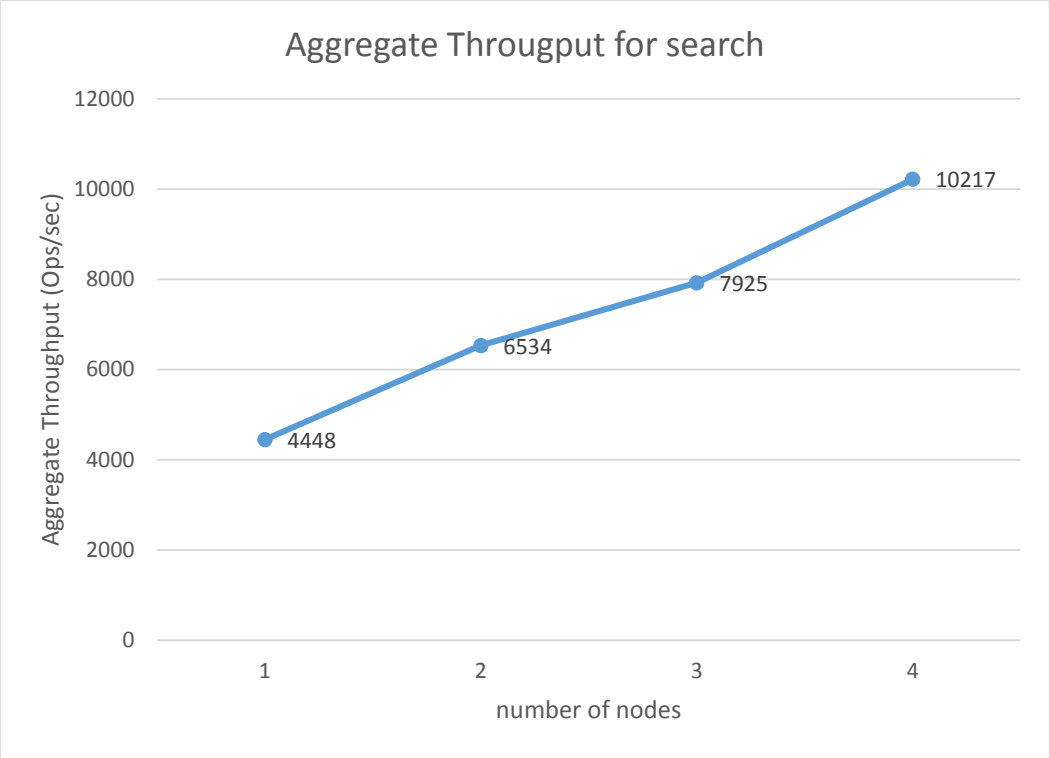
Average time taken by a node to make 10000 searches is :

= **7853.25 millisecs**

a. Computing aggregate Throughput in (search) Operations per second for all clients:

Client	Throughput (Ops/sec)
1	1269
2	1289
3	1246
4	1177
5	1397
6	1294
7	1357
8	1190

Aggregate Throughput (Ops/sec) = 1267+1289+1246+1177+1397+1294+1357+1190
=10217



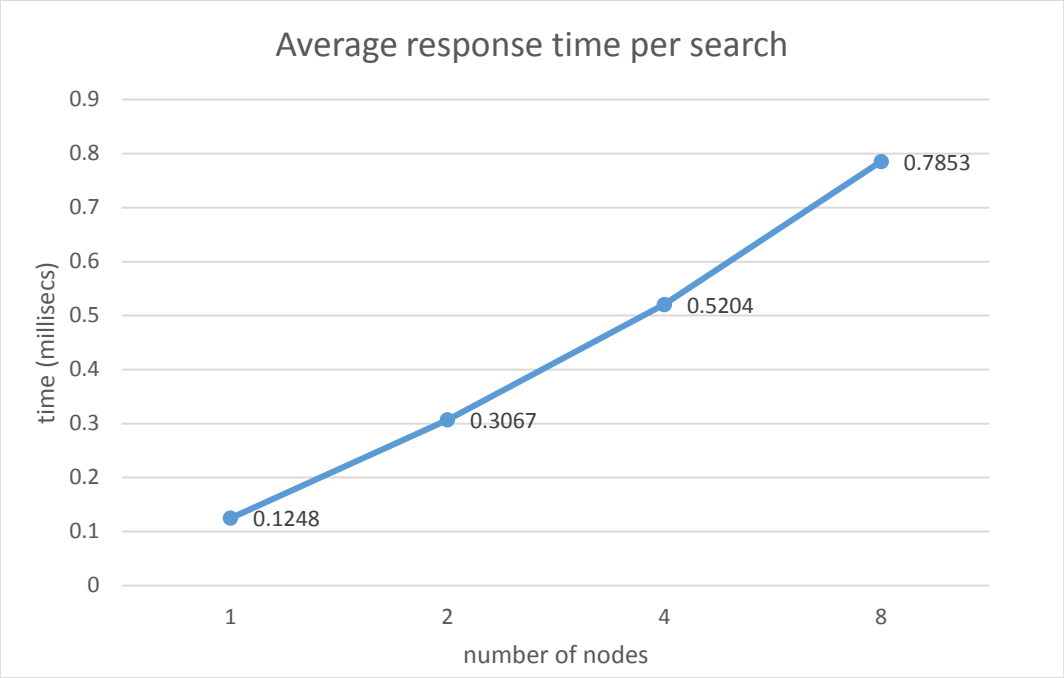
a. **Average response time per search Operation :**

Average time taken for a single node per search : 1248/10000
:0.1248 millisecs

Average time taken for two concurrent nodes per search : 3067/10000
:0.3067 millisecs

Average time taken for four concurrent nodes per search :5204/10000
: 0.5204 millisecs

Average time taken for eight concurrent nodes per search :7853.25/10000
:0.7853



Explanation: The search operation involves going over the network and doing a get operation on the remote node hash table with some latency. Increasing number of clients involves simultaneous communication on the network at a node .so gradual increase in the response time for the search operation. With number of concurrent clients doing the search operations. The aggregate Throughput increases.

c. Download:

single node: The time taken to download 10000 100KB files by a single node is **28595 millisecs**.

Aggregate Throughput (Ops/sec) =350 Ops/sec

two nodes : The time taken to download 10000 100KB files by each of 2 nodes is :

node 1 - 19597 millisecs

node 2 - 19689 millisecs

Average time taken by a node to make 10000 downloads is :

= **19643 millisecs**

Client	Throughput (Ops/sec)
1	510
2	507

Aggregate Throughput (Ops/sec) = 510+507

=1017 Ops/sec

four nodes : The time taken to download 10000 100KB files by each of four nodes is :

node 1 - 29577 millisecs

node 2 - 30979 millisecs
node 3 - 30657 millisecs
node 4 – 32535 millisecs

Average time taken by a node to make 10000 downloads is : = **30937 millisecs**

Client	Throughput (Ops/sec)
1	338
2	332
3	326
4	307

Aggregate Throughput (Ops/sec) = 338+332+326+307
=1303

eight nodes : The time taken to download 10000 100KB files by each of eight nodes is :

node 1 - 28505 millisecs
node 2 - 33434 millisecs
node 3 - 32612 millisecs
node 4 – 32132 millisecs
node 5 - 27137 millisecs
node 6 – 38933 millisecs
node 7 - 38259 millisecs
node 8 – 42527 millisecs

Average time taken by a node to make 10000 downloads is :

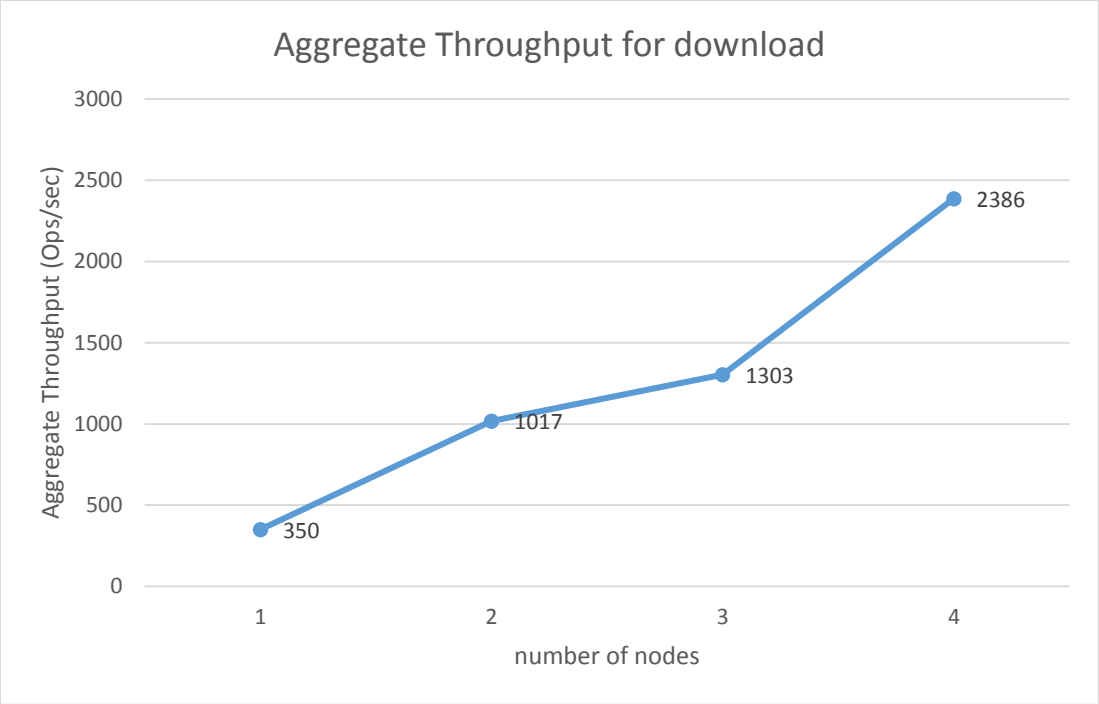
= **34226.12 millisecs**

a. **Computing aggregate Throughput in (download) Operations per second for all clients:**

Client	Throughput (Ops/sec)
1	350
2	299
3	306
4	310
5	368
6	256
7	261
8	236

Aggregate Throughput (Ops/sec) = 350+299+306+310+368+256+261+236

=2386



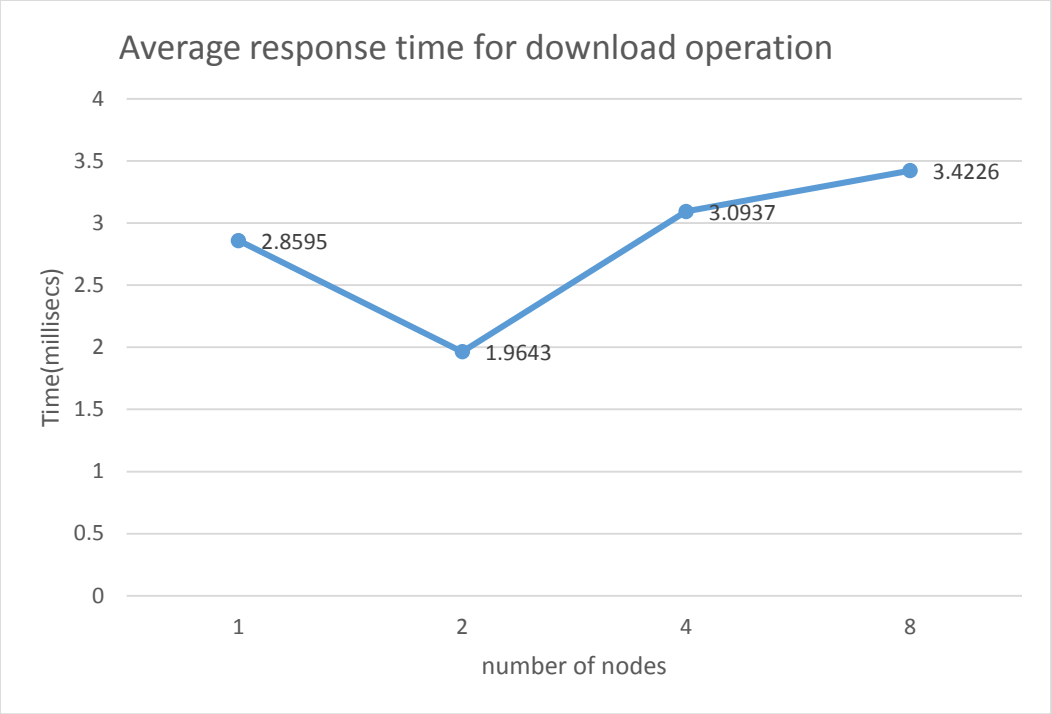
b. **Average response time per download Operation :**

Average time taken for a single node per download : 28595/10000
:2.8595 millisecs

Average time taken for two concurrent nodes per download : 19643/10000
:1.9643millisecs

Average time taken for four concurrent nodes per download :30937/10000
: 3.0937 millisecs

Average time taken for eight concurrent nodes per download :34226.12/10000
:3.4226



Explanation: The download operation on single node is local read. On the m3.large instance max throughput of local read is 90MB/s. so reading 10000 100KB files at max throughput would take 11.2 secs. The local read is happening at 35MB/s. Hence, 28.595 seconds for 10000 100KB reads. And as obvious as it seems, the response time increases with increase of number of nodes. Network bandwidth is utilized by nodes concurrently.

Experiment 2:

This experiment involves calculating throughput (downloads) for various file sizes. In this, we keep the number of files and servers fixed at 8 and run the experiment by varying the file size from 1KB to 1GB with different number of files for each file size.

1. **1KB , 10000 :**

   Throughput for file sizes of 10000 1KB for 8 clients.

   Below is the table giving time taken for each of the operations by each client for 10000 1kB files

Client	Register (ms)	Search (ms)	Download (ms)
1	18065	5140	400190
2	17662	5057	400360
3	18688	5723	400373
4	18229	5178	400222
5	17024	4854	400252
6	17003	4889	400211

7	17684	4824	400313
8	17753	5149	400287

Calculating Throughput (MB/S) for each of the clients downloading 1kB files.

Client	Throughput (MB/S)
1	0.025
2	0.0249
3	0.0247
4	0.0248
5	0.02498
6	0.0249
7	0.0247
8	0.02473

Aggregate Throughput (MB/S) = 0.025+0.0249+0.0247+0.0248+0.02498+0.0249+0.0247+0.0473 =0.221 MB/S

2. **10KB , 50000 :**

   Throughput for file sizes of 50000 10KB for 8 clients.

   Below is the table giving time taken for each of the operations by each client for 50000 10kB files

Client	Register (ms)	Search (ms)	Download (ms)
1	28635	27944	70073
2	26135	25713	64827
3	25378	25693	66297
4	25505	25480	64784
5	25388	26493	65383
6	26714	27304	67482
7	26781	25939	67087
8	26874	25764	66789

Calculating Throughput(MB/S) for each of the clients downloading 1kB files.

Client	Throughput (MB/s)
1	7.13
2	7.71
3	7.55
4	7.71
5	7.64
6	7.40
7	7.37
8	7.48

Aggregate Throughput (MB/S) =7.13+7.71+7.75+7.71+7.64+7.40+7.37+7.48
=60.19 MB/S

3. **100KB , 20000 :**

   Throughput for file sizes of 20000 100KB for 8 clients.

   Below is the table giving time taken for each of the operations by each client for 20000 100kB files

Client	Register (ms)	Search (ms)	Download (ms)
1	12072	10048	63856
2	11875	10414	65261
3	12526	11190	67271
4	12830	11548	66660
5	11375	10125	67704
6	11022	10043	65139
7	10818	10056	64702
8	11253	10634	68204

Calculating Throughput(MB/S) for each of the clients downloading 100kB files.

Client	Throughput (MB/s)
1	31.34
2	30.624
3	29.73
4	30
5	29.53
6	30.703
7	30.91
8	30.85

Aggregate Throughput (MB/S) = 31.34+31.624+29.73+30+29.53+30.70+30.91+30.85
=244.6 MB/S

4. **1MB , 7500 :**

   Throughput for file sizes of 7500 1MB for 8 clients.

   Below is the table giving time taken for each of the operations by each client for 7500 1MB files

Client	Register (ms)	Search (ms)	Download (ms)
1	4229	3202	65588
2	4410	3289	60601
3	4149	2814	61281
4	4127	2725	62751
5	3394	2619	56454
6	4080	2856	61039
7	4120	2842	61833
8	3769	3565	63181

Calculating Throughput (MB/S) for each of the clients downloading 1MB files.

Client	Throughput (MB/s)
1	114.50
2	123.76
3	122.38
4	119.5

5	133.45
6	122.9
7	121.2
8	118.7

Aggregate Throughput (MB/s) =114.50+123.76+122.38+119.5+133.45+122.9+121.2+118.
=975.5 MB/s

5. **10M , 1000:**

Throughput for file sizes of 1000 10MB for 8 clients.

Below is the table giving time taken for each of the operations by each client for 1000 10MB files

Client	Register (ms)	Search (ms)	Download (ms)
1	667	636	81024
2	672	622	83505
3	620	514	72739
4	606	560	79272
5	615	628	77423
6	685	632	82436
7	655	543	78256
8	662	613	81458

Calculating Throughput (MB/s) for each of the clients downloading 10MB files.

Client	Throughput (MB/S)
1	123.4
2	119.7
3	137.1
4	126.6
5	129.9
6	121.5
7	127.8
8	122.8

Aggregate Throughput (MB/S) = 123.4+119.7+137.1+126.6+129.9+121.5+127.8+122.8
=1008.8 MB/S

6. **100MB , 25 :**

Throughput for file sizes of 100MB 25 for 8 clients.

Below is the table giving time taken for each of the operations by each client for 25 100MB files

Client	Register (ms)	Search (ms)	Download (ms)
1	11	12	65709
2	11	12	62207
3	82	10	68982
4	12	11	69776
5	1 0	10	68277
6	10	10	64214
7	10	10	61359
8	10	10	60716

Calculating Throughput(MB/S) for each of the clients downloading 100MB files.

Client	Throughput (MB/S)
1	38.05
2	40.1
3	36.2
4	36.25
5	36.6
6	38.8
7	40.7
8	41.18

Aggregate Throughput (MB/S) = 38.05+40.1+36.2+36.25+36.6+38.8+40.7+41.18

=307.88 MB/S

7. **1GB , 3 :**

Throughput for file sizes of 3 1 GB for 8 clients.

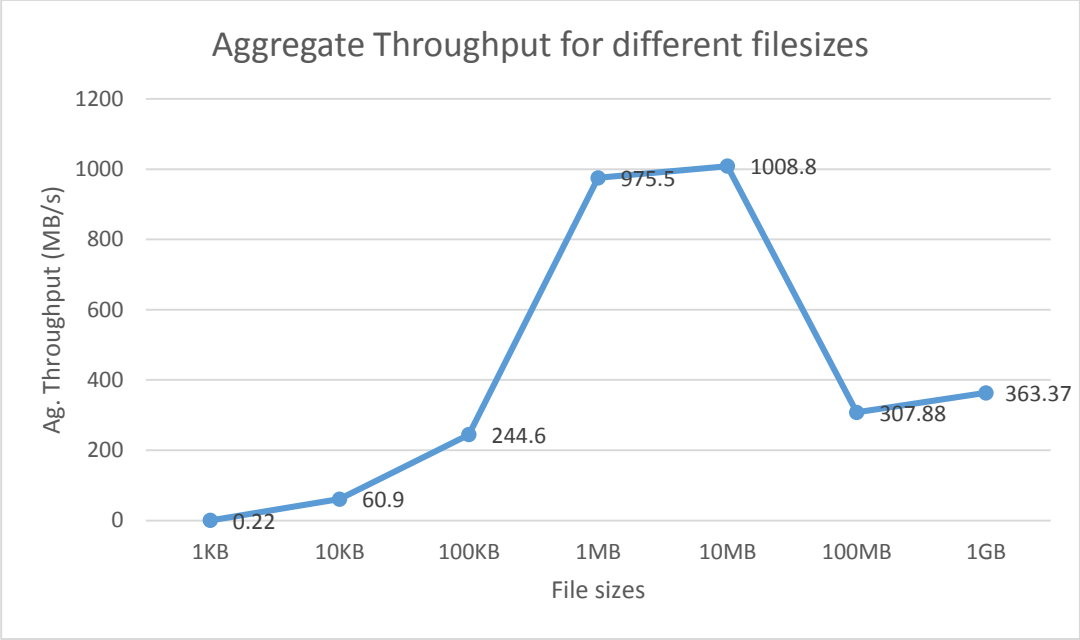
Below is the table giving time taken for each of the operations by each client for 3 1GB files

Client	Register (ms)	Search (ms)	Download (ms)
1	19	2	69164
2	24	3	65918
3	3	3	63386
4	3	2	71662
5	3	2	64564
6	3	3	63589
7	4	3	66863
8	3	2	63742

Calculating Throughput (MB/S) for each of the clients downloading 3 1GB files.

Client	Throughput (MB/S)
1	43.47
2	45.5
3	47.3
4	41.8
5	46.5
6	46.9
7	44.8
8	47.06

Aggregate Throughput (MBs/sec) =43.47+45.5+47.3+41.8+46.5+46.9+44.8+47.08
=363.37 MB/S



Explanation: The Aggregate Throughput for downloading 1KB files is considerably low as 8 nodes requesting and making downloads concurrently involve frequent switching among nodes rapidly and and choking up the network with exchange messages. Hence, lower throughput per node and lower aggregate Throughput for 1KB files.

The latency between 2 amazon instances when PINGed is 0.34ms.**(ping 172.31.36.25)** So max Throughput with a TCP window size of 64KB is 218MB/s When downloading 10KB file siz1es, since the socket buffer size is at 64KB the buffer is filled upto 64KB and then read from the buffer so , we see a reasonable throughput for download and increased aggregate throughput compared to 1KB files . Hence it's obvious the throughput gradually increases and Throughput for a node is observed to be maximum of 120MB/s at 100kB file sizes.

The sudden drop in Aggregate Throughput when downloading larger files of 100MB and 1GB should be due to when a 1GB file download is requested , downloading larger bursts of file block is not possible with the buffer size of 64KB and as 8 nodes are concurrently requesting 100MB and 1GB file downloads, The network is consumed with numerous huge file bocks. Hence the throughput observed is significantly lower compared to 1MB and 10MB file downloads.

3. Centralized vs Decentralized System:

Here for a Centralized system I just computed the Throughput in Operations/sec to verify the Performance of a System . 'cause having a centralized version of indexing server for the system would add load on the single Indexing server since all the nodes would contact the Central Server for register and search Operations , while in Distributed System each of the peers act as Indexing server. So, by changing the number of nodes we try to see what effect does the design of system have on register , search and download operations of files? and this can be understood and interpreted by measuring Throughput in (Op/s) of the System for 10000 operations each of register , search and downloads for 100kB files like in the experiment above.

a. Registration :

single node: The time taken to register 10000 100KB files on a single node is **319 millisecs**.

Aggregate Throughput (Ops/sec)=31347

two nodes : The time taken to register 10000 100KB files on each of 2 nodes is :

node 1 - 323 millisecs
node 2 - 329 millisecs

Client	Throughput (Ops/sec)
1	30959
2	30959

Aggregate Throughput (Ops/sec)=61918

four nodes : The time taken to register 10000 100KB files on each of four nodes is :

node 1 - 316 millisecs
node 2 - 319 millisecs
node 3 - 320 millisecs
node 4 – 317 millisecs

Client	Throughput (Ops/sec)
1	31645
2	31347
3	31250
4	31545

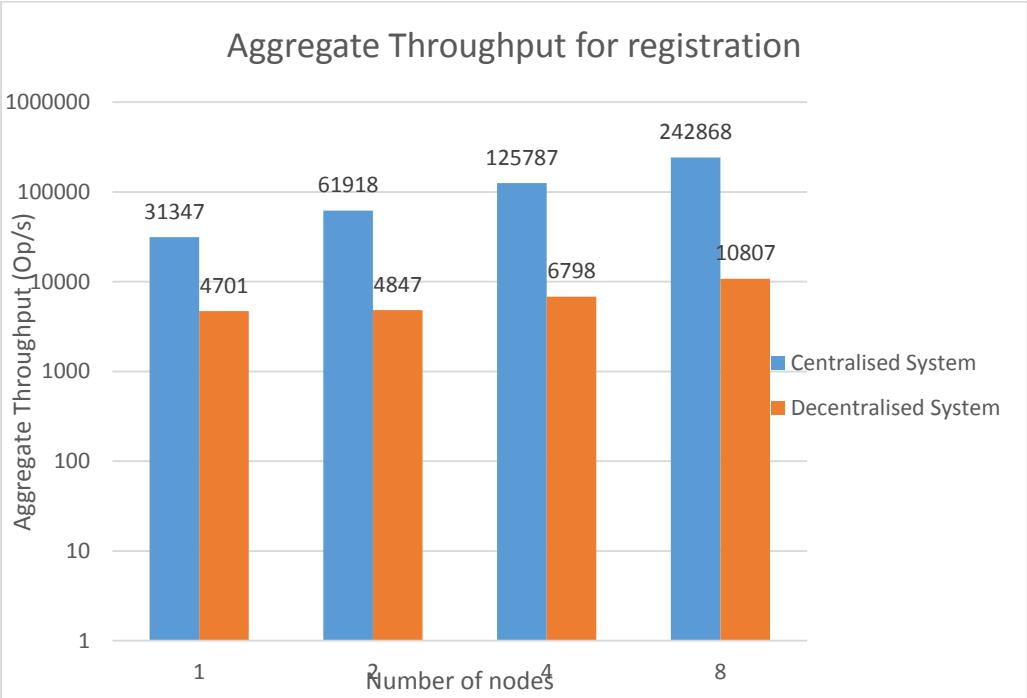
Aggregate Throughput (Ops/sec) =125787

eight nodes : The time taken to register 10000 100KB files on each of eight nodes is :

node 1 - 321 millisecs
node 2 – 314 millisecs
node 3 – 326 millisecs
node 4 – 319 millisecs
node 5 - 424 millisecs
node 6 – 324 millisecs
node 7 - 314 millisecs
node 8 – 318 millisecs

Client	Throughput (Ops/sec)
1	31259
2	31847
3	30674
4	31347
5	23584
6	30864
7	31847
8	31446

Aggregate Throughput (Ops/sec) =242868



Explanation : Comparing the aggregate throughput for centralized and decentralized It is observed that centralized registration has a higher magnitude aggregate throughput for registration operations compared to decentralized system for the scales of 1 to 8 nodes . In the centralized system there is only a single index server so when a node is registering files at indexing server is gonna communicate only with a single node all the time . Hence, larger throughputs for registration operations . In a decentralized system , a node when registering files is gonna communicate with all the other nodes and register files , this switching between nodes may add cost. And it can be understood that centralized system is efficiently able to handle the scales of 8 nodes , while this may not be the case when scales are increased more .

b. Search :

single node: The time taken to search 10000 100KB files by a single node is **4587 millisecs**.

Aggregate Throughput (Ops/sec)=2180

two nodes : The time taken to search 10000 100KB files by each of 2 nodes is :

node 1 - 4934 millisecs
node 2 - 4782 millisecs

Client	Throughput (Ops/sec)
1	2180
2	2091

Aggregate Throughput (Ops/sec) =4271

four nodes : The time taken to search 10000 100KB files by each of four nodes is :

node 1 - 5936 millisecs
node 2 - 4965 millisecs
node 3 - 4818 millisecs
node 4 – 4651 millisecs

Client	Throughput (Ops/sec)
1	1684
2	2014
3	2075
4	2150

Aggregate Throughput (Ops/sec) =7923

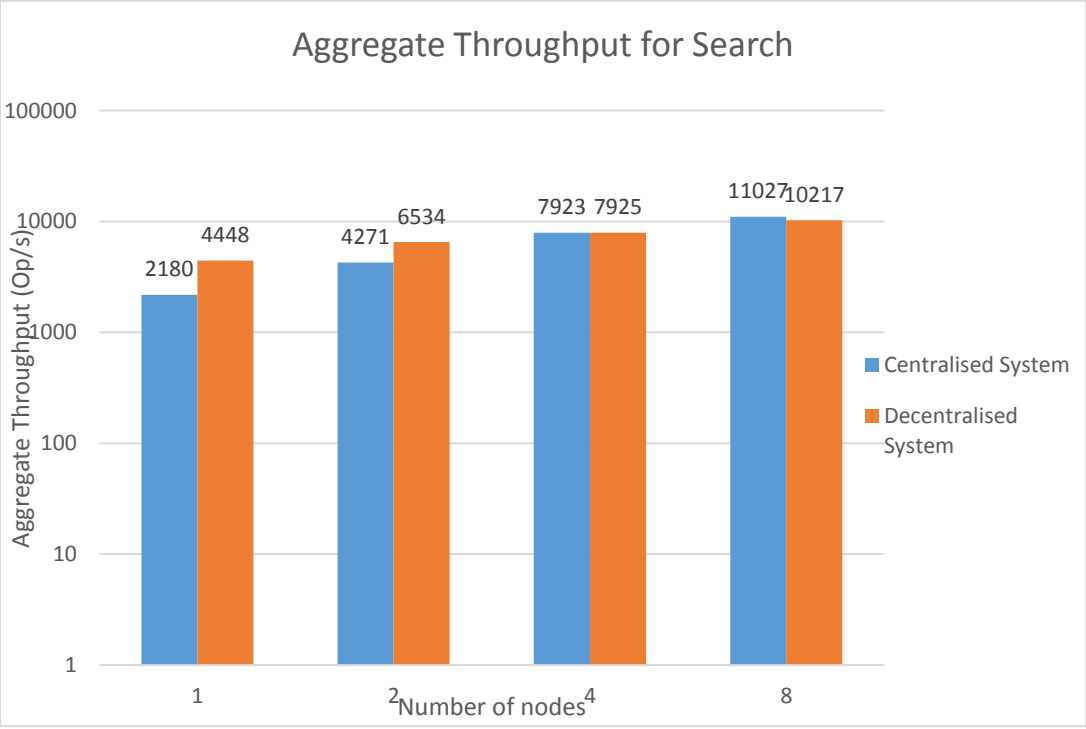
eight nodes : The time taken to search 10000 100KB files by each of eight nodes is :

node 1 - 7906 millisecs
node 2 - 7450 millisecs

node 3 - 6510 millisecs
node 4 – 7540 millisecs
node 5 - 6859millisecs
node 6 – 7776 millisecs
node 7 - 7384 millisecs
node 8 – 6853 millisecs

Client	Throughput (Ops/sec)
1	1264
2	1342
3	1536
4	1326
5	1457
6	1286
7	1354
8	1459

Aggregate Throughput (Ops/sec) =11027



Explanation: The aggregate Throughput of search operations for centralized and decentralized system increases gradually with increased scale of nodes from 1 to 8, as observed in the above experiments the increased scales of nodes increased aggregate throughput. At the present scale of 8 nodes as Centralised Indexing server still works efficiently the search operation throughput is almost at the same levels as decentralized system.

c. Download:

single node: The time taken to download 10000 100KB files by a single node is **6413 millisecs**.

Aggregate Throughput (Ops/sec) =1559

two nodes : The time taken to download 10000 100KB files by each of 2 nodes is :

node 1 - 17237 millisecs

node 2 - 17038 millisecs

Average time taken by a node to make 10000 downloads is :

= **19643 millisecs**

Client	Throughput (Ops/sec)
1	588
2	586

Aggregate Throughput (Ops/sec) =1174

four nodes : The time taken to download 10000 100KB files by each of four nodes is :

node 1 - 28859 millisecs

node 2 – 29605 millisecs

node 3 - 27605 millisecs

node 4 – 28046 millisecs

Client	Throughput (Ops/sec)
1	346
2	337
3	362
4	356

Aggregate Throughput (Ops/sec) =1401

eight nodes : The time taken to download 10000 100KB files by each of eight nodes is :

node 1 - 31071 millisecs

node 2 - 28744 millisecs

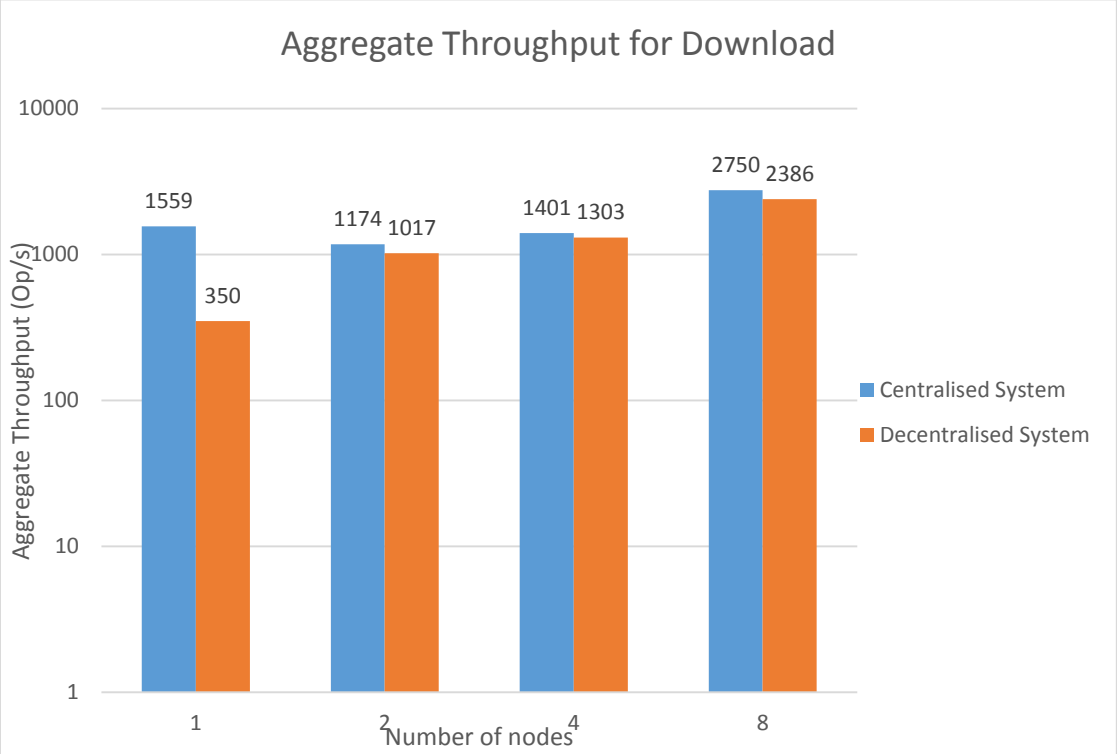
node 3 - 31465 millisecs

node 4 – 27985 millisecs

node 5 - 31481 millisecs
node 6 – 26218 millisecs
node 7 - 27625 millisecs
node 8 – 28641 millisecs

Client	Throughput (Ops/sec)
1	321
2	347
3	317
4	357
5	317
6	381
7	361
8	349

Aggregate Throughput (Ops/sec) = 2750



Explanation: The download operations for decentralized system at local node shows a relatively lower throughput compared to Centralized system , as the max local read throughput is observed to 90MB/s. Besides, that the aggregate throughput for download operations from remote

multiple nodes at various scales are pretty close. This aggregate throughput metric for download tells very little about the relative performance of the centralized system in contrast with Decentralized system.

Conclusion: The performance of Centralized system is slightly better compared to Decentralized system at the scale of 8 nodes as seen from the above experiments. So, for this experiment 8 nodes hasn't been a bottleneck for Centralized system. The advantages of Decentralized system shall be realized at more larger scales when having single indexing server for search and register operations should incur overwhelming load on the Indexing server when peers requesting are at larger scales. Even the Decentralized systems is observed to give a better performance at 8 node scale for register, search and download operations with only little increase of average response times for all the operations when nodes are increased from 1 to 8.