Performance Testing

Testing Environment

We have used plain (direct) I/O requests and REST API for performing the I/O operations.

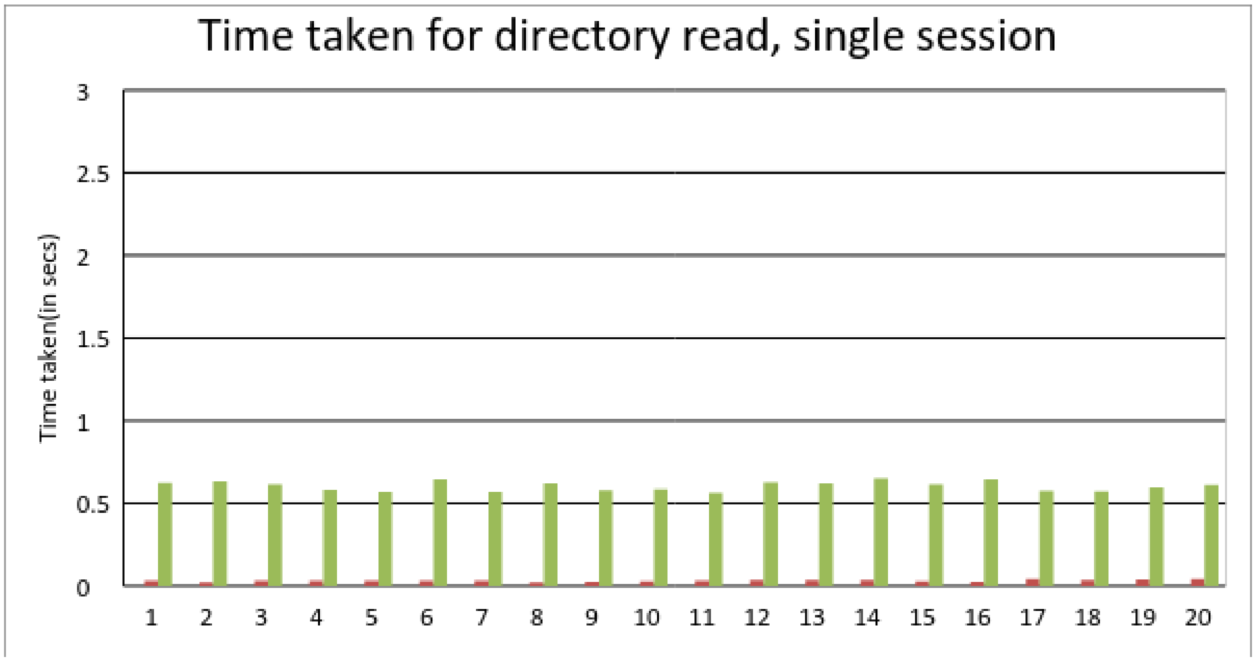
- In the first section, performance was tested by using a shell script for list directory (metadata related), file upload, file download and file delete.

Single Session, Single File (Plain Requests)

1) list metadata(directory) information

File size(kB)	Ceph(secs)	HDFS(secs)
1	0.030439138	0.621695
2	0.022926807	0.633479
4	0.031247139	0.61423
8	0.030158043	0.5811
16	0.031104803	0.570597
32	0.029989958	0.646783
64	0.030119896	0.568131
128	0.022655964	0.62036
256	0.026463985	0.579783
512	0.026609182	0.586086

1024	0.030353785	0.562637
2048	0.033744097	0.6247
4096	0.034879923	0.618775
8192	0.034982204	0.652192
16384	0.029280901	0.613326
32768	0.024727106	0.645273
65536	0.040113926	0.575957
131072	0.034832001	0.571772
262144	0.036827087	0.594731
524288	0.041426897	0.610098



Observation:

We can see that time taken for directory Read , is lower for Ceph than HDFS.

Conclusion:

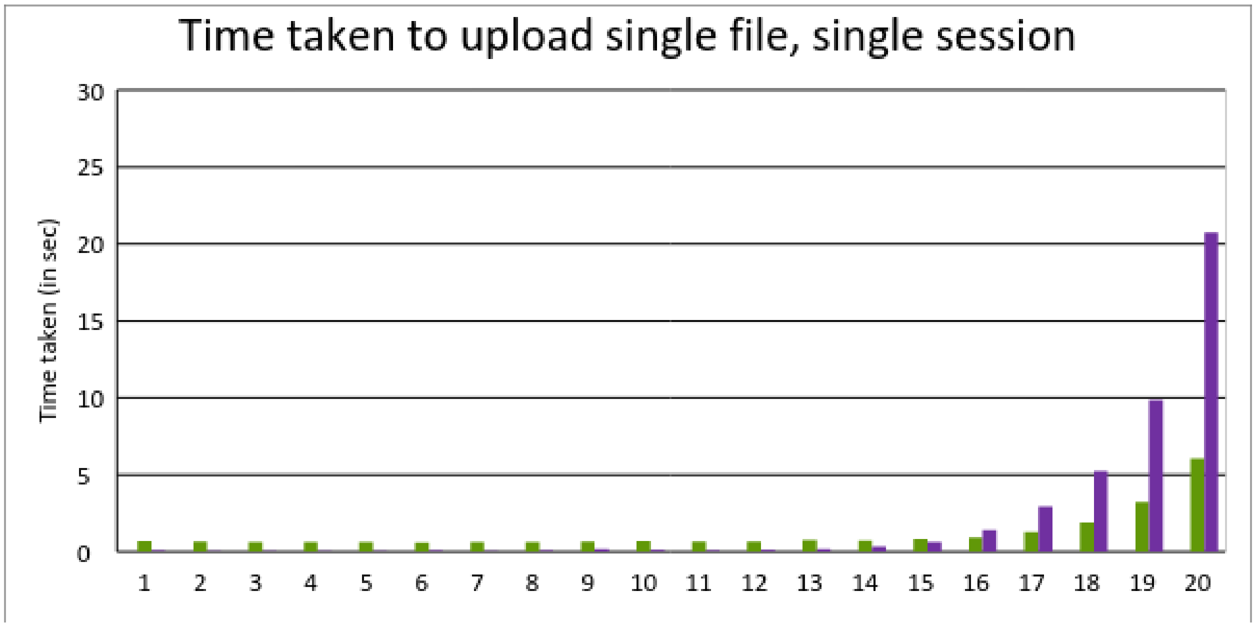
1. We can closely observe that time taken to read the directory information from the name node is higher in HDFS as it follows the master-slave configuration with name node maintaining the metadata information and the data nodes responsible for actual data storage.
2. For CEPH, due to its decentralized architecture where we have the capability to retrieve the metadata information directly from the storage nodes without much interference from the monitor node makes it suitable to retrieve the metadata listing much faster than HDFS.

2) File write:

File size(kB)	HDFS(secs)	Ceph(secs)
1	0.7012341	0.068618
2	0.6606490 6	0.058163

4	0.62941313	0.058169
8	0.62559509	0.058507
16	0.64083385	0.058276
32	0.60640597	0.075765
64	0.64042187	0.06056
128	0.63828206	0.067631
256	0.65622902	0.167546
512	0.68719292	0.100602
1024	0.64419818	0.096773
2048	0.67238688	0.126431
4096	0.76715612	0.197406
8192	0.709934	0.350301
16384	0.83229399	0.630633
32768	0.94150615	1.389107
65536	1.28578997	2.942095
131072	1.91591597	5.249471
262144	3.23681712	9.851026

524288	6.0388648 5	20.73971



Observation:

We can see that time taken to upload smaller files is almost the same for all the three file systems, but as the file size increases,

1. As file size increases, time taken by CEPH to upload the file increases proportionally.
2. For HDFS, the time taken also increases, but the rise is very less when compared to other file system studied

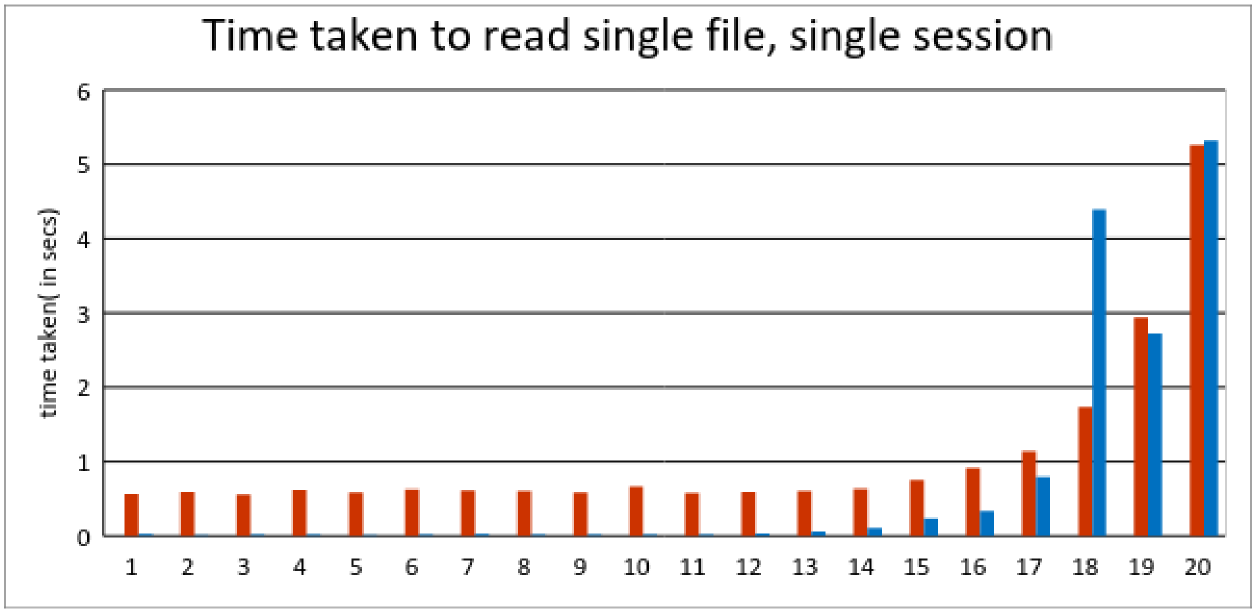
Conclusion:

For smaller file sizes, the upload performance is good for CEPH. HDFS is good for both smaller as well as large files.

3) File read:

File size(kB)	HDFS(secs)	Ceph(secs)
1	0.569065	0.025014
2	0.593658	0.013478

4	0.560347	0.019754
8	0.621998	0.01939
16	0.584578	0.017114
32	0.62745	0.023065
64	0.615296	0.026645
128	0.612468	0.021862
256	0.586649	0.023433
512	0.661648	0.022926
1024	0.577532	0.024306
2048	0.594815	0.038356
4096	0.609595	0.066167
8192	0.637466	0.110662
16384	0.756646	0.236698
32768	0.915312	0.337004
65536	1.142332	0.794628
131072	1.737675	4.39064
262144	2.940764	2.72098
524288	5.254994	5.323936



Observation:

For downloading a file, CEPH takes lesser time than HDFS for smaller file sizes. A marked variation happens for file sizes of range 65536 to 131072 kb where HDFS takes lesser time than CEPH. For very large file sizes both of them behave similar.

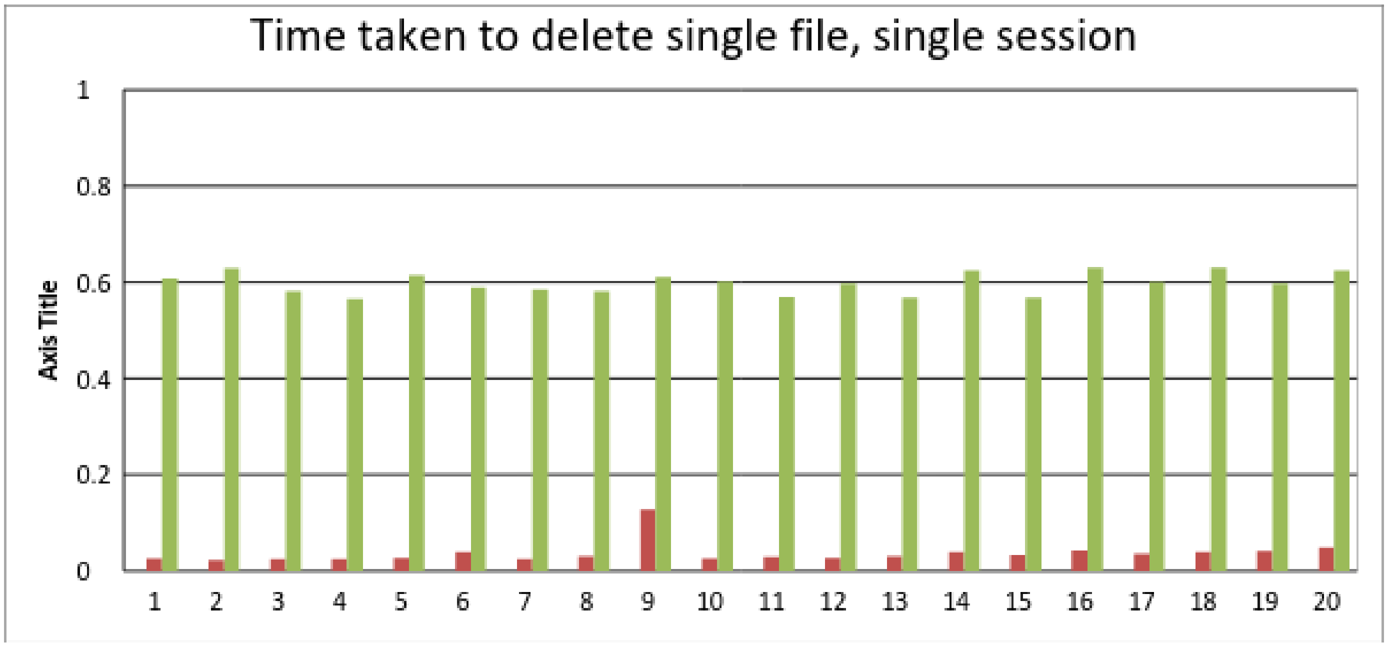
Conclusion:

CEPH is better for small file and very large file reads whereas HDFS is optimized for all large file reads.

4) File delete:

File size(kB)	Ceph(in secs)	HDFS(in secs)
1	0.025914907	0.608009
2	0.022119045	0.629491
4	0.02415204	0.581484
8	0.025264025	0.565785
16	0.026772022	0.614449

32	0.038611889	0.589655
64	0.024303913	0.584699
128	0.030338049	0.581294
256	0.127332926	0.611708
512	0.025959015	0.602025
1024	0.029842854	0.569584
2048	0.027185917	0.597247
4096	0.03115797	0.568546
8192	0.038819075	0.624
16384	0.032590866	0.568005
32768	0.042558908	0.630474
65536	0.035859108	0.600949
131072	0.038727999	0.631062
262144	0.040961981	0.597217
524288	0.048100948	0.624726



Observation:

We can observe that the time taken a delete a particular file size is very less in CEPH than in HDFS.

Conclusion:
CEPH is better than HDFS.

Multiple Clients (Plain Requests)

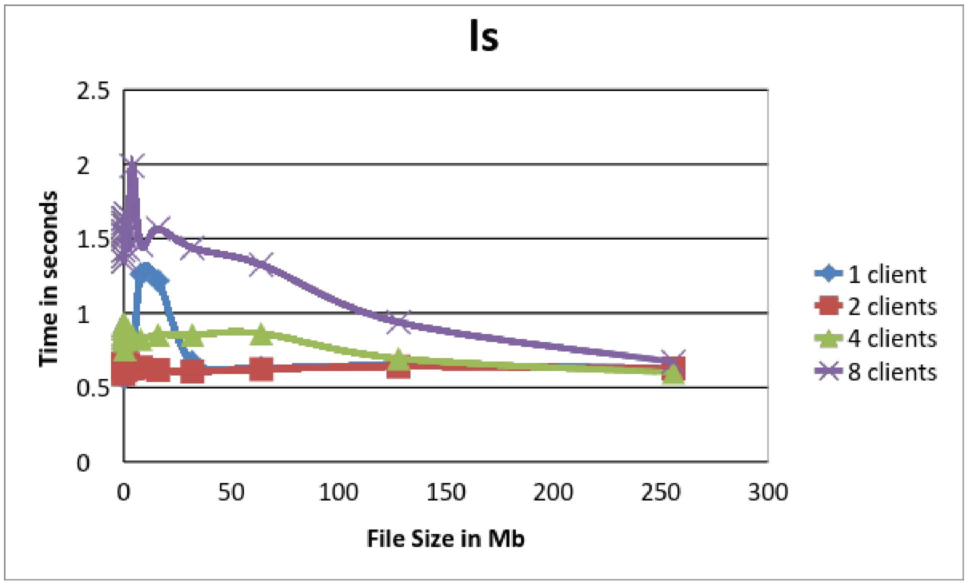
<u>For HDFS:</u>

The various observations when multiple clients are accessing the HDFS are as bellow

1) List metadata (directory) information

File Size in Mb	1 client	2 clients	4 clients	8 clients
0.000977	0.600777	0.654138	0.8515	1.421399
0.001953	0.571241	0.591674	0.89576	1.34838
0.003906	0.569517	0.640296	0.91799	1.635256
0.007813	0.626926	0.641653	0.81506	1.525592

0.015625	0.615592	0.637852	0.83608	1.549345
0.03125	0.623727	0.624925	0.90034	1.383236
0.0625	0.647424	0.661784	0.8674	1.584671
0.125	0.666871	0.626756	0.93614	1.607485
0.25	0.584478	0.647927	0.9051	1.636284
0.5	0.662124	0.630518	0.75745	1.509815
1	0.568956	0.617971	0.8485	1.668658
2	0.582986	0.657454	0.8804	1.430114
4	0.6014	0.628196	0.84949	1.989053
8	1.262238	0.637123	0.82515	1.460835
16	1.217832	0.616647	0.85506	1.567164
32	0.671184	0.605459	0.85401	1.437927
64	0.634377	0.622376	0.86397	1.325465
128	0.650494	0.642142	0.69525	0.938765
256	0.63255	0.629568	0.60039	0.673008



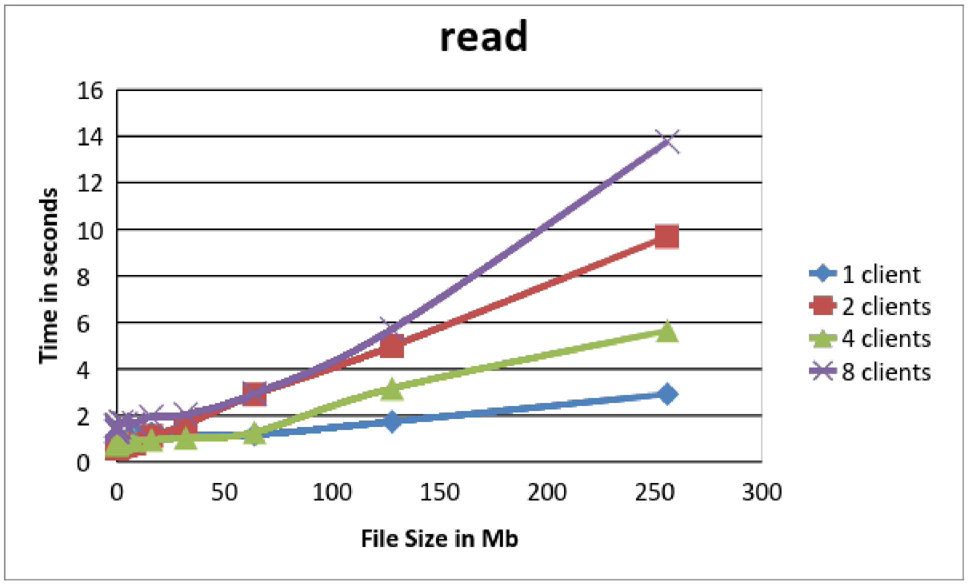
Observation:

AS the number of clients increase the time taken for listing the becomes more.

2) File read:

File Size in Mb	1 client	2 clients	4 clients	8 clients
0.000977	0.58329	0.672886	0.758896	1.3026
0.001953	0.65039	0.622535	0.767725	1.5475
0.003906	0.614073	0.639238	0.857472	1.3098
0.007813	0.590367	0.608878	0.836111	1.3597
0.015625	0.575828	0.615906	0.825718	1.5481
0.03125	0.572705	0.64	0.822672	1.5945
0.0625	0.582931	0.65423	0.814995	1.5424

0.125	0.585655	0.607203	0.917359	1.5737
0.25	0.568542	0.63125	0.872781	1.4902
0.5	0.656248	0.643218	0.822846	1.5018
1	0.66417	0.692242	0.891481	1.4856
2	0.656164	0.705911	0.856015	1.7252
4	0.668908	0.713848	0.947503	1.6457
8	1.266261	0.785131	0.949838	1.6766
16	1.23332	1.113188	0.962888	1.9428
32	1.188862	1.604311	1.04874	2.0539
64	1.17242	2.901361	1.280097	2.9406
128	1.737513	4.975512	3.167772	5.7133
256	2.917595	9.69025	5.647539	13.772



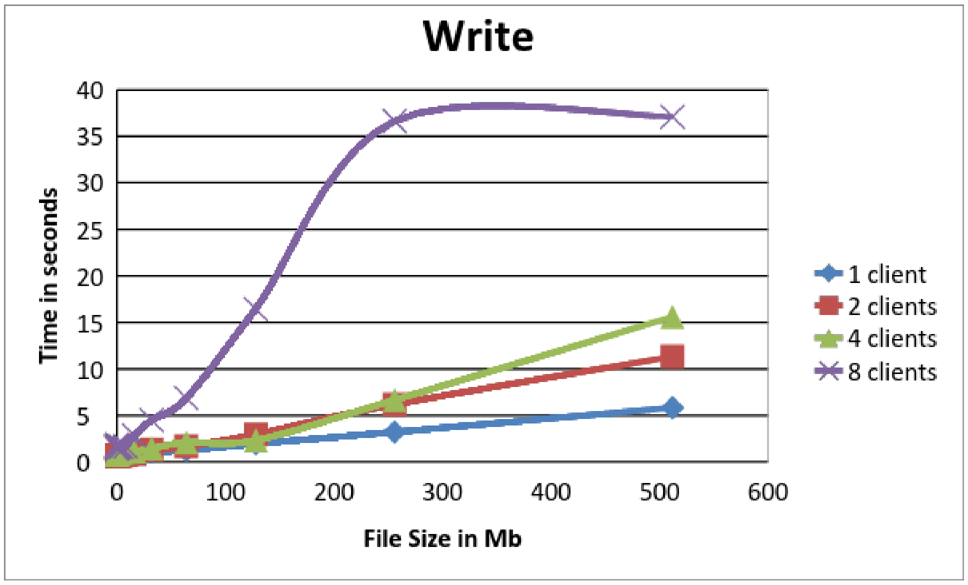
Observation:

The read operation shows a clear increase in time as the file size and the number of clients increase.

3) File Write:

File Size in Mb	1 client	2 clients	4 clients	8 clients
0.000977	0.673553	0.724548	0.9311	1.523847
0.001953	0.675397	0.649199	0.87902	1.792038
0.003906	0.700122	0.65645	0.93624	1.649609
0.007813	0.687632	0.66719	0.93939	1.690589
0.015625	0.673512	0.705704	0.91132	1.462985
0.03125	0.667065	0.667974	0.85923	1.733203

0.0625	0.677372	0.656215	0.83838	1.630992
0.125	0.714849	0.715467	0.97039	1.73066
0.25	0.697404	0.757779	0.8493	1.834049
0.5	0.717998	0.693744	0.8441	1.698942
1	0.714956	0.701214	0.90402	1.63025
2	0.673405	0.703921	1.0128	1.703535
4	0.669926	0.738912	1.02171	1.623825
8	1.475506	0.772136	1.05529	1.679601
16	1.524426	0.904022	1.11396	2.863503
32	1.100767	1.308258	1.4683	4.5
64	1.281053	1.69987	2.09748	6.882023
128	1.961082	2.981537	2.33453	16.45
256	3.225565	6.193077	6.66596	36.64
512	5.836868	11.3291	15.6069	37.09721



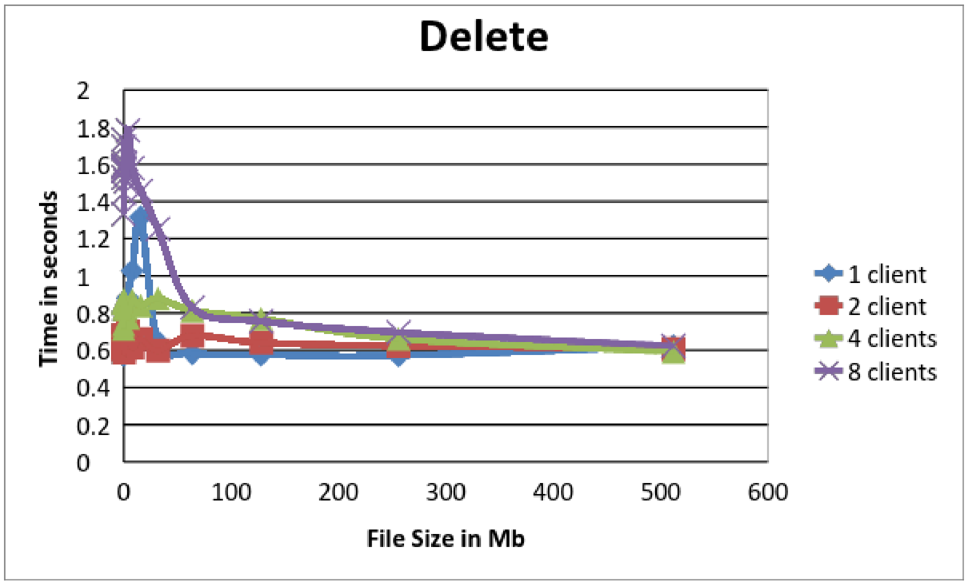
Observations:

The Write performance is similar to the read performance

4) File Delete:

File Size in Mb	1 client	2 client	4 clients	8 clients
0.000977	0.611938	0.627842	0.71514	1.332575
0.001953	0.653364	0.619949	0.85564	1.566943
0.003906	0.572968	0.613556	0.86179	1.528454
0.007813	0.616891	0.677027	0.85295	1.693983
0.015625	0.607594	0.683351	0.82451	1.592833
0.03125	0.627308	0.606248	0.73888	1.586265
0.0625	0.598775	0.618905	0.86842	1.731415
0.125	0.629693	0.653256	0.86998	1.444577

0.25	0.642055	0.656002	0.86927	1.563909
0.5	0.669133	0.58954	0.86259	1.574417
1	0.582857	0.632597	0.82224	1.605547
2	0.587434	0.673313	0.8269	1.508008
4	0.88085	0.699988	0.77595	1.781385
8	1.027435	0.613426	0.87021	1.579004
16	1.316716	0.659884	0.83965	1.456171
32	0.645797	0.598726	0.87826	1.252418
64	0.584006	0.67821	0.81393	0.82937
128	0.5758	0.641824	0.77048	0.755931
256	0.57209	0.623695	0.66172	0.69641
512	0.622918	0.608256	0.5928	0.624159



Observation:

The Delete operation performs similarly for all the clients.

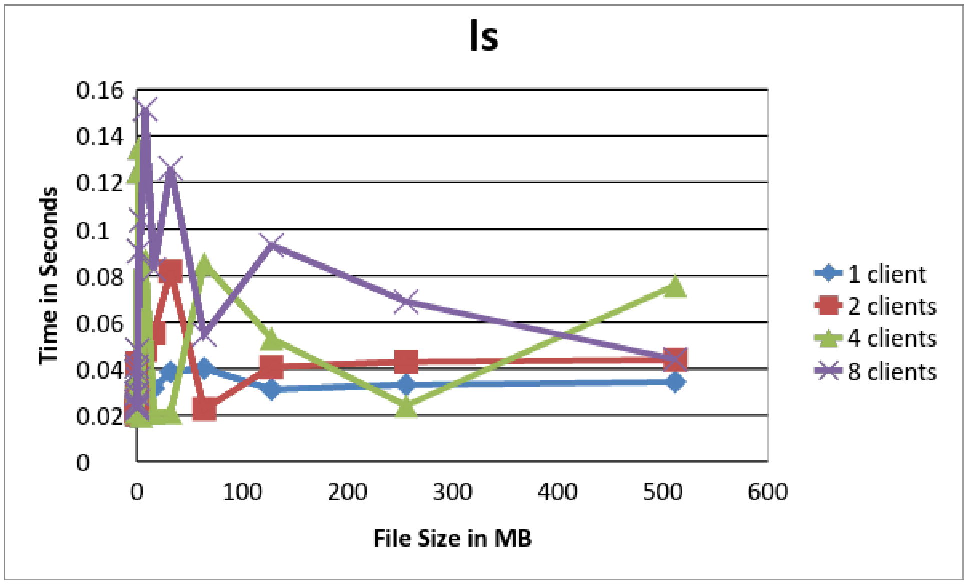
For Ceph:

The various observations when multiple clients are accessing the Ceph are as bellow

1) List metadata (directory) information

File Size in Mb	1 client	2 clients	4 clients	8 clients
0.000977	0.019615	0.028018	0.022734	0.025755
0.001953	0.023552	0.021972	0.021243	0.023202
0.003906	0.030965	0.021414	0.021993	0.029378
0.007813	0.022834	0.020369	0.022326	0.024531
0.015625	0.025054	0.022284	0.021524	0.038886
0.03125	0.021702	0.022505	0.031786	0.029081

0.0625	0.026923	0.021407	0.023266	0.041034
0.125	0.040031	0.021722	0.030233	0.023752
0.25	0.02619	0.022147	0.036238	0.0228
0.5	0.038558	0.04231	0.021802	0.048014
1	0.031819	0.022144	0.124583	0.035659
2	0.033324	0.023034	0.134875	0.090487
4	0.023924	0.030679	0.019796	0.103773
8	0.034183	0.048147	0.086815	0.151525
16	0.031617	0.055262	0.021006	0.082687
32	0.03862	0.081974	0.021138	0.126087
64	0.040149	0.022641	0.085148	0.05476
128	0.031046	0.04081	0.052999	0.093076
256	0.033035	0.042882	0.024144	0.068821
512	0.034327	0.043839	0.075837	0.043993



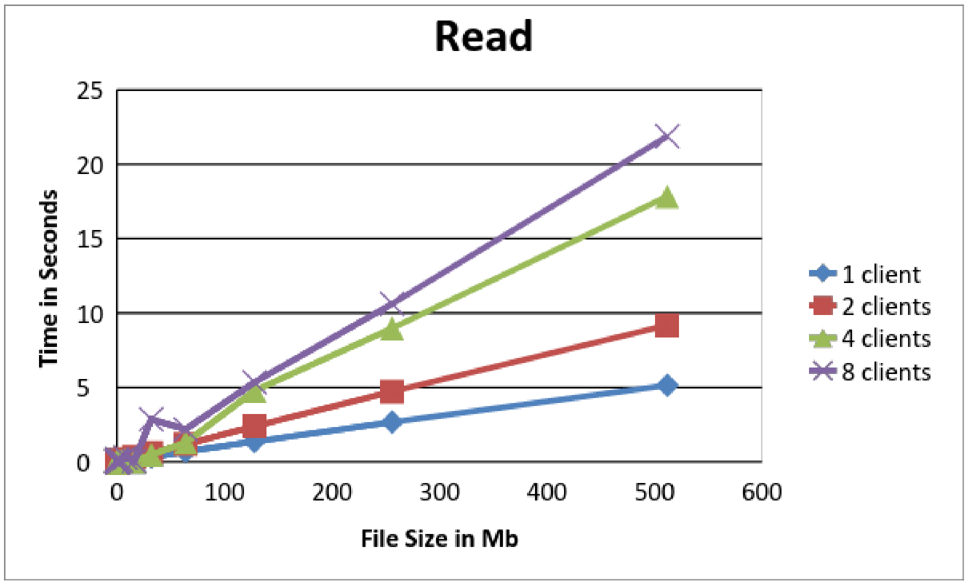
Observations:

The time taken to list is random and the listing is not effected as the number of clients increase .

2) File Read:

File Size in Mb	1 client	2 clients	4 clients	8 clients
0.000977	0.025894	0.016878	0.020488	0.056948
0.001953	0.025473	0.020761	0.014685	0.175339
0.003906	0.026008	0.018763	0.017772	0.040403
0.007813	0.016227	0.022159	0.017926	0.04931
0.015625	0.032091	0.014285	0.013377	0.111867
0.03125	0.023339	0.017975	0.013814	0.068568
0.0625	0.018016	0.021346	0.018251	0.03504
0.125	0.025033	0.022306	0.019616	0.052967
0.25	0.020908	0.09877	0.022637	0.014827

0.5	0.027354	0.035281	0.019789	0.016934
1	0.030499	0.038424	0.046403	0.030413
2	0.038721	0.08838	0.064546	0.043912
4	0.059258	0.087707	0.110798	0.109717
8	0.105949	0.176656	0.156042	0.294537
16	0.193616	0.336191	0.016702	0.077592
32	0.348164	0.604975	0.492142	2.869871
64	0.708077	1.193124	1.306616	2.198757
128	1.376493	2.402326	4.788165	5.35966
256	2.677865	4.735985	8.979768	10.62023
512	5.149752	9.186259	17.84417	21.8842



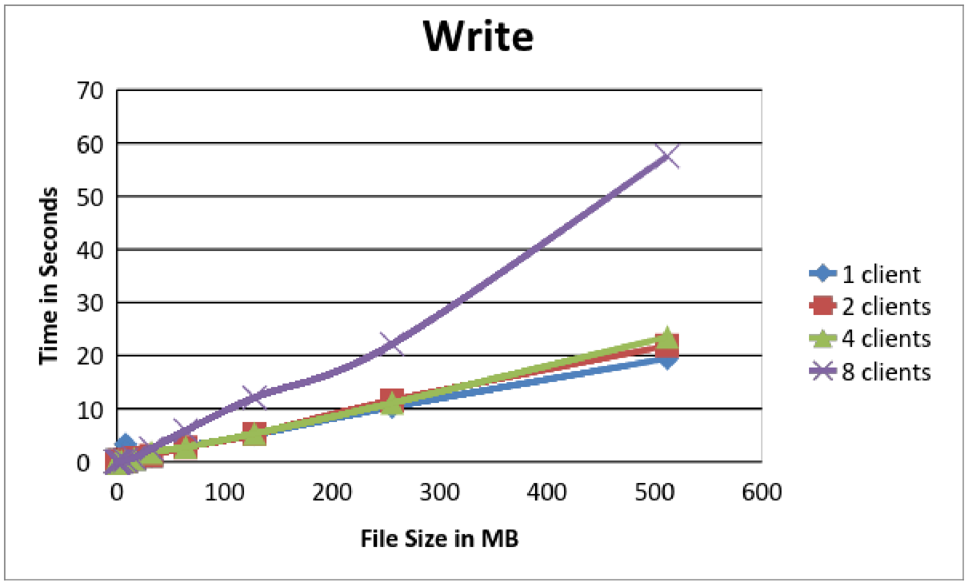
Observations:

The time taken to read increseses with the file size and the number of clients accesing it .

3)File Write:

File Size in Mb	1 client	2 clients	4 clients	8 clients
0.000977	0.0776	0.046478	0.049863	0.044761
0.001953	0.075397	0.038411	0.034644	0.095422
0.003906	0.073162	0.034867	0.036696	0.050096
0.007813	0.058678	0.054279	0.036803	0.048994
0.015625	0.081856	0.036206	0.061203	0.061672
0.03125	0.060851	0.035864	0.052256	0.062801
0.0625	0.069415	0.043567	0.070175	0.070995
0.125	0.073903	0.040728	0.040739	0.080121
0.25	0.074466	0.110525	0.048942	0.054404
0.5	0.086323	0.071054	0.05235	0.087591
1	0.096122	0.073903	0.069645	0.079854
2	0.188868	0.256674	0.103509	0.126251
4	0.190121	0.176375	0.182996	0.171943
8	3.353298	0.305459	0.326456	0.333525

16	0.629268	0.727529	0.60467	0.830246
32	1.180894	1.219415	1.833794	2.508908
64	2.870618	2.787135	2.839756	5.912279
128	5.157545	5.234787	5.358406	12.11382
256	10.36301	11.57316	11.10402	22.16121
512	19.47695	21.87123	23.52994	57.50725



Observations:

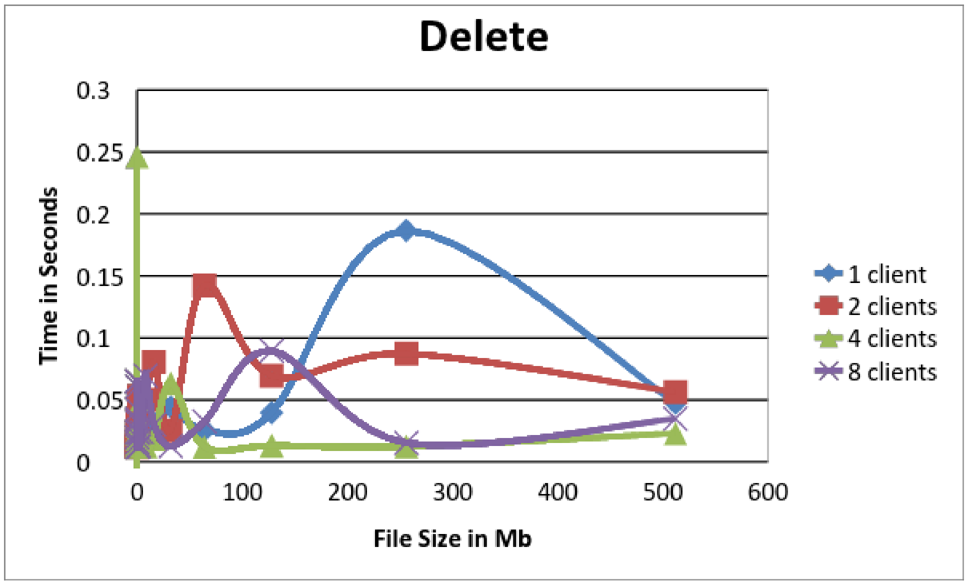
The write operation performs similar to the write operation.

4) File Delete:

File Size in Mb	1 client	2 clients	4 clients	8 clients
0.000977	0.023335	0.024432	0.014358	0.019946

0.001953	0.033974	0.012119	0.013307	0.024062
0.003906	0.025731	0.012623	0.012529	0.034646
0.007813	0.029386	0.012414	0.246083	0.065406
0.015625	0.033978	0.013086	0.024855	0.064635
0.03125	0.0275	0.024445	0.029372	0.060809
0.0625	0.029583	0.015876	0.023235	0.032256
0.125	0.036653	0.011981	0.022716	0.042243
0.25	0.027465	0.034715	0.019023	0.015896
0.5	0.035054	0.024564	0.018654	0.012321
1	0.037189	0.016979	0.013379	0.02349
2	0.024581	0.053154	0.02978	0.012707
4	0.02493	0.040667	0.050308	0.024213
8	0.02485	0.050279	0.012631	0.069523
16	0.031327	0.080644	0.018637	0.030218
32	0.044405	0.025345	0.063485	0.012989
64	0.025767	0.142567	0.012211	0.032663

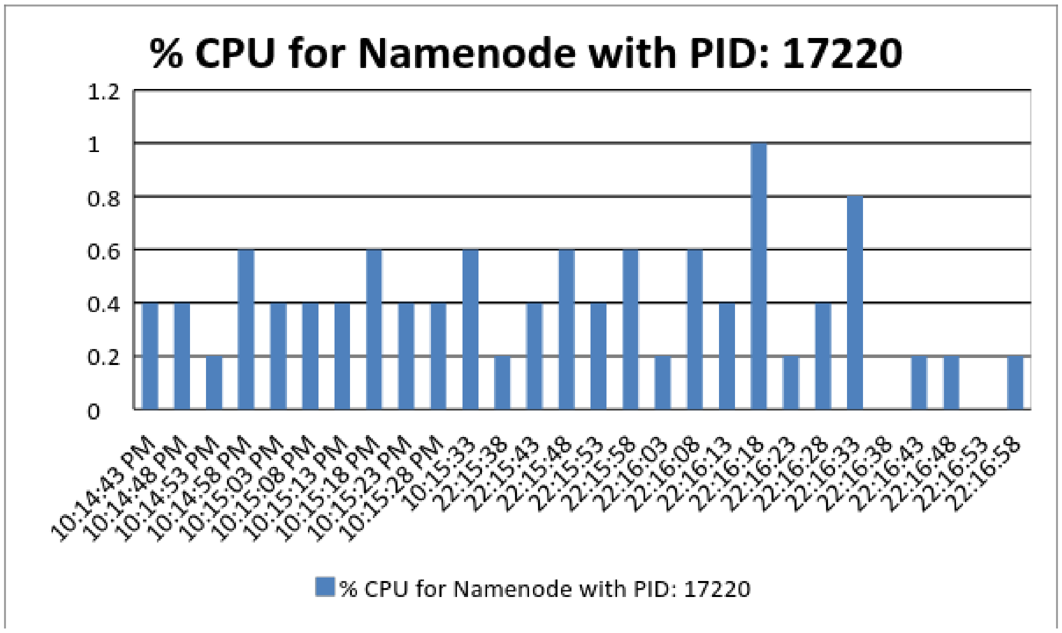
128	0.039795	0.069528	0.013329	0.089627
256	0.18655	0.087276	0.012684	0.015359
512	0.047578	0.056437	0.023062	0.034906



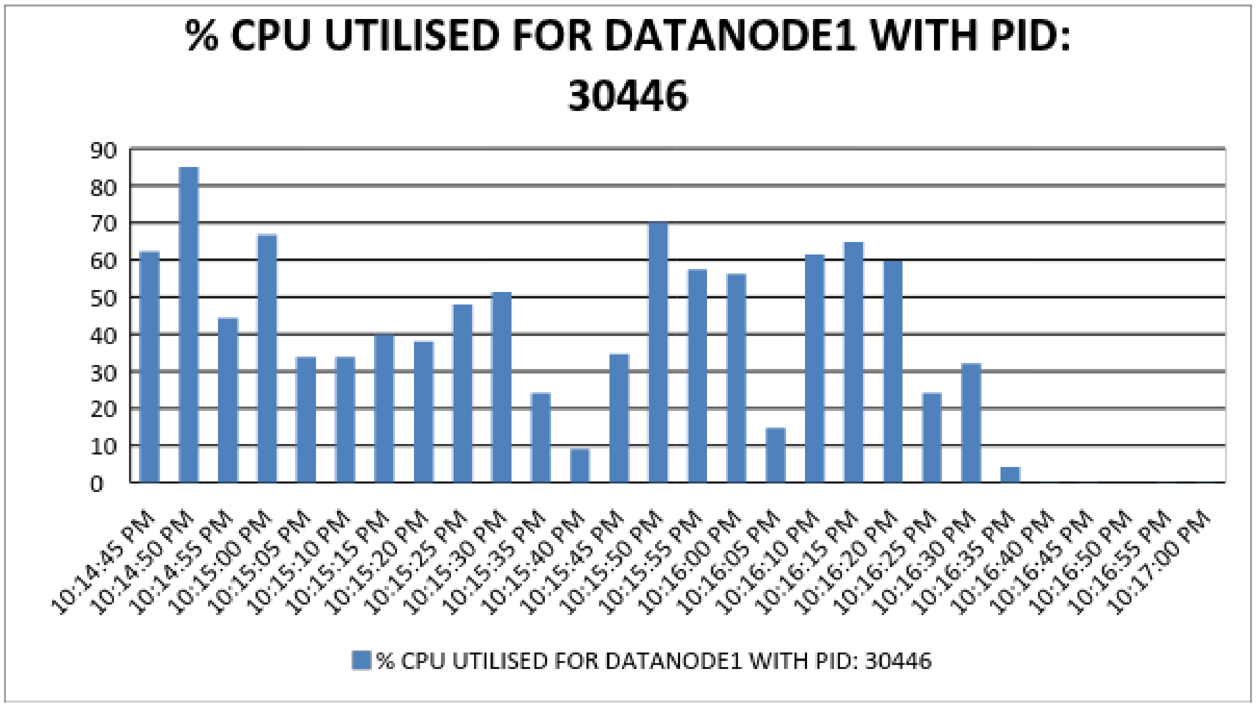
Observations:
The delete file as the same random behavior as the ls operations .The increase in the number of clients doesn't change the behavior of the delete
Conclusion:
Ceph and HDFS has similar kind of behavior with multiple clients.

CPU Utilization for HDFS

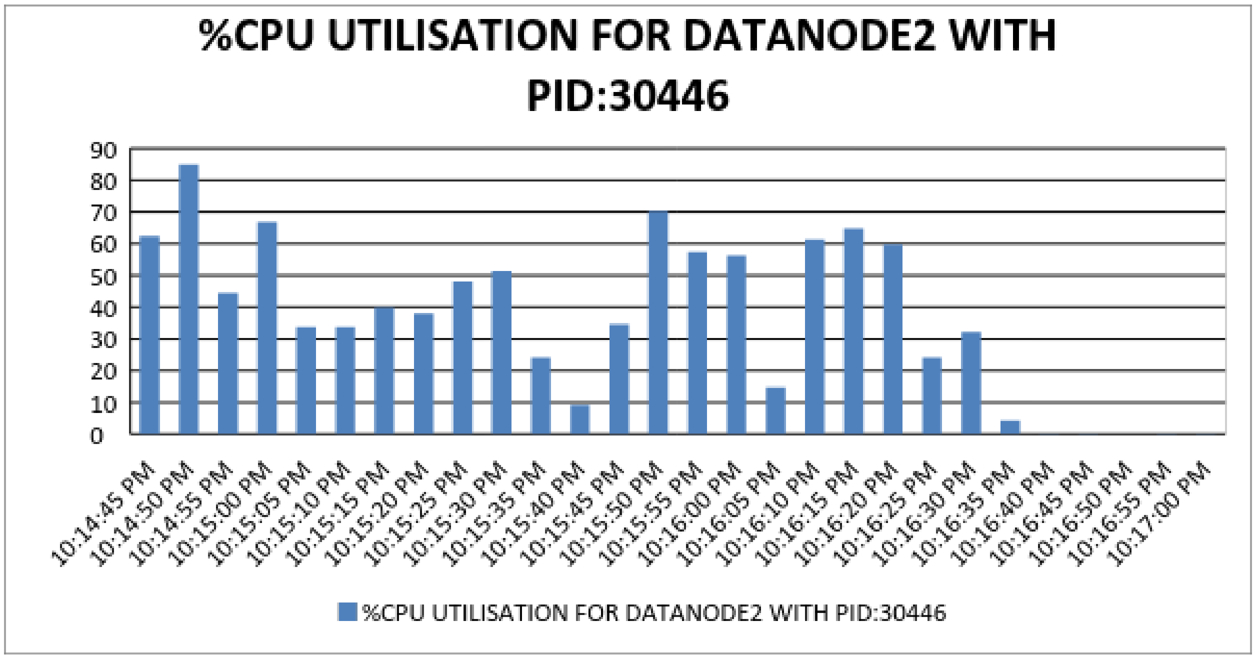
Name node



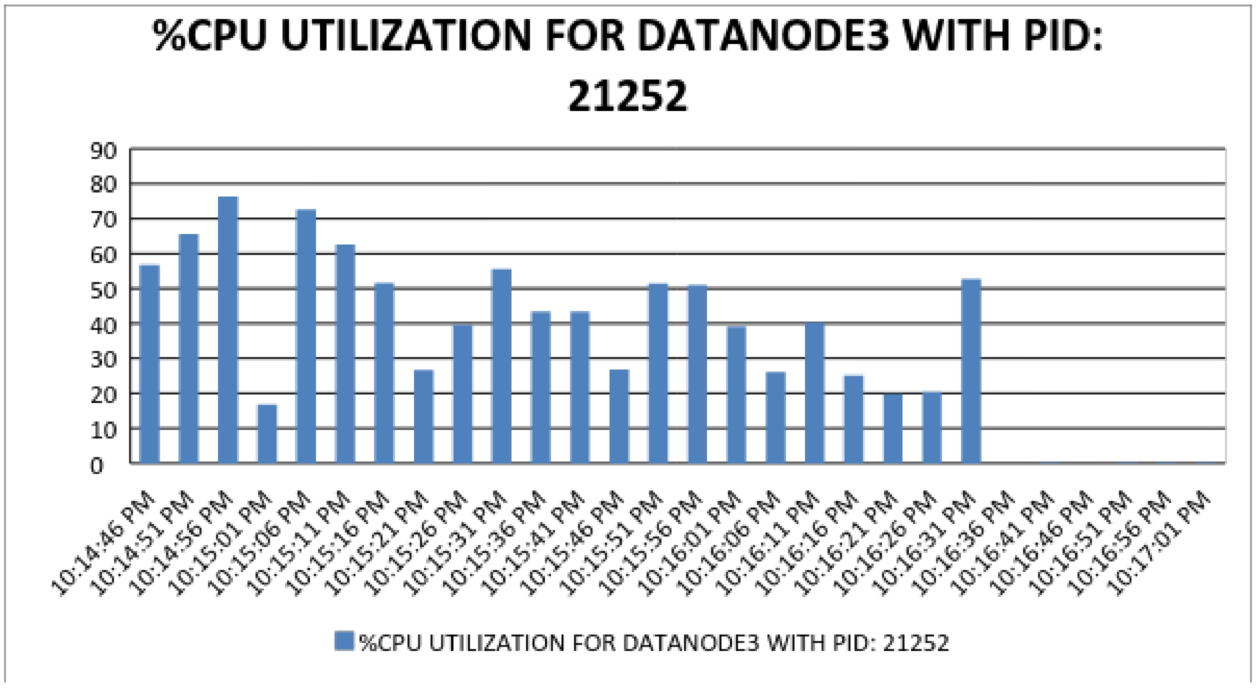
Slave 1(data node):



Slave 2(data node):



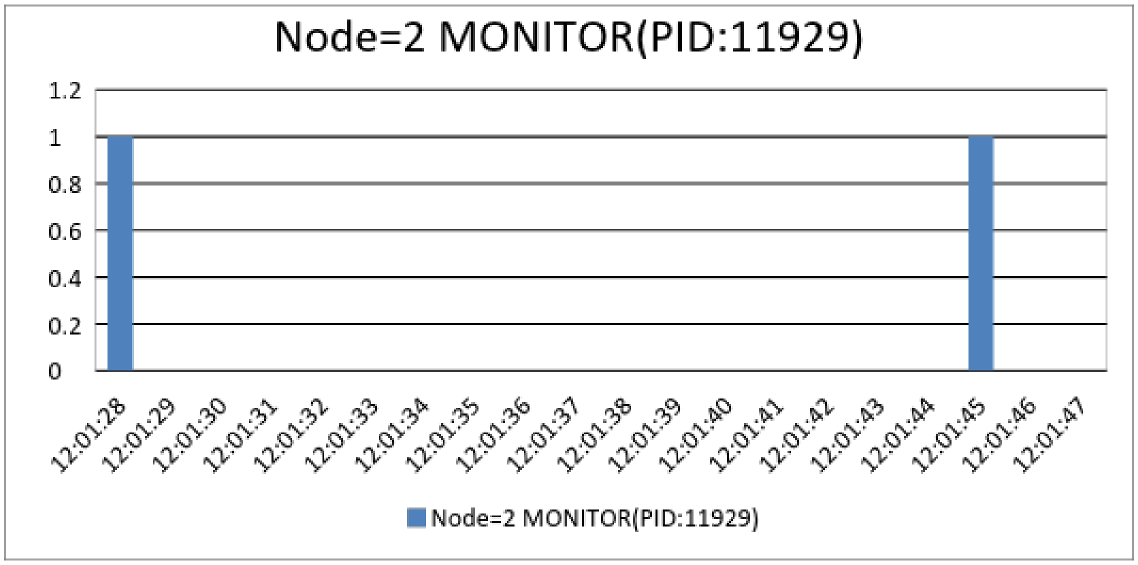
Slave 3(data node):

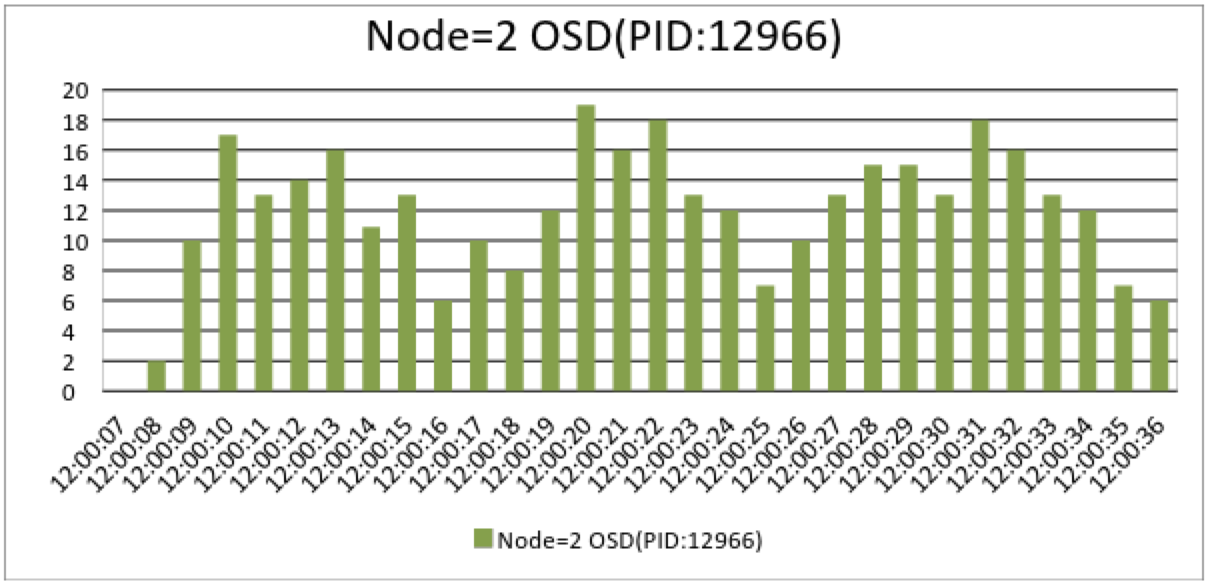


CPU UTILIZATION FOR CEPHFS:

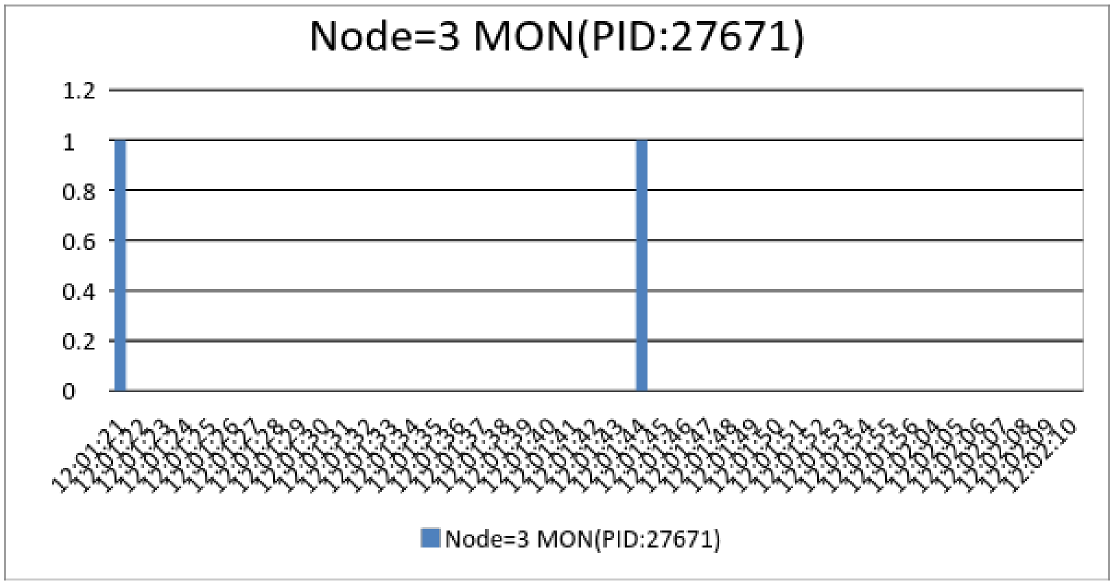
Node 1: admin node

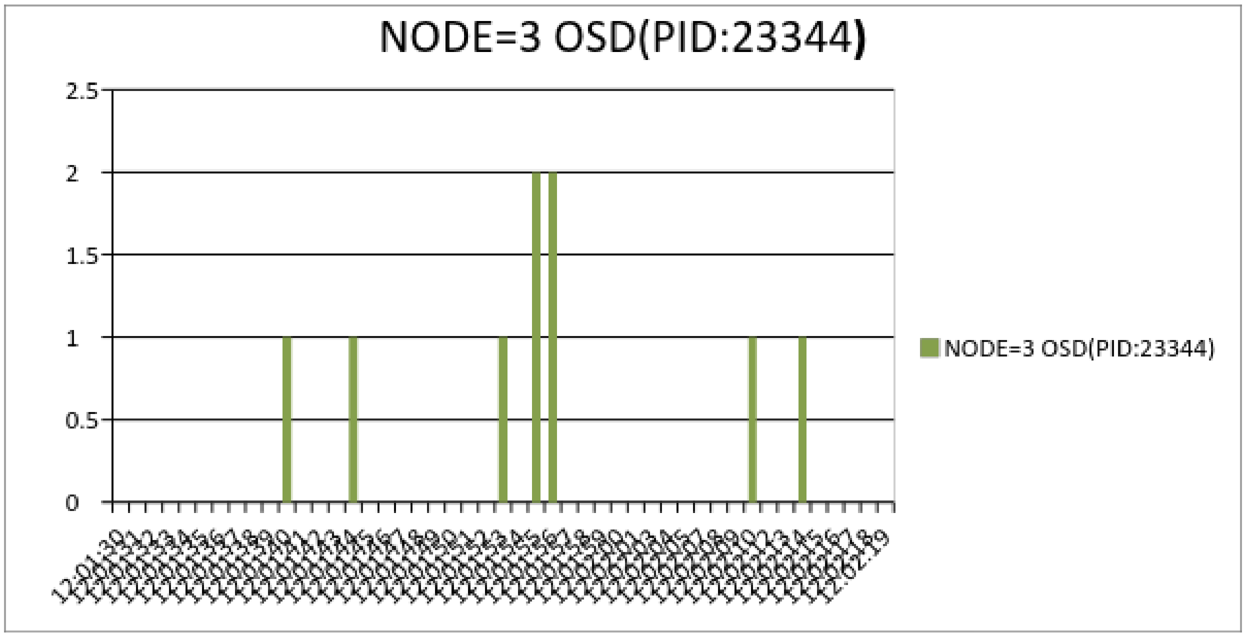
Node 2 (1 monitor, 1 OSD in multi-node cluster):

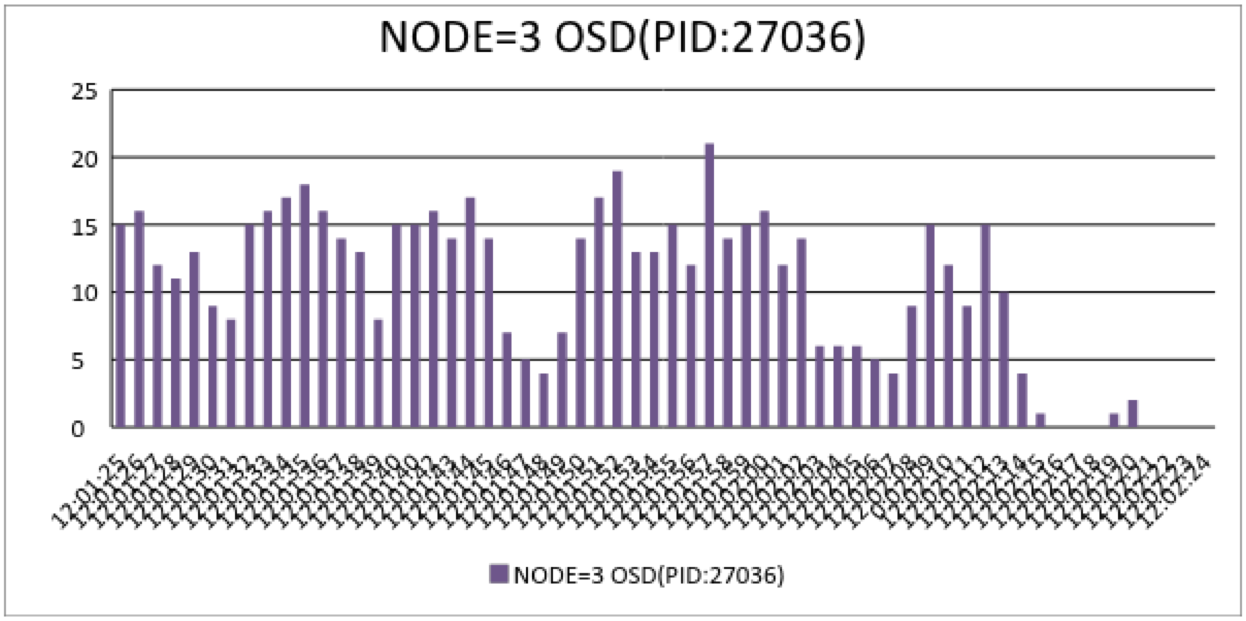




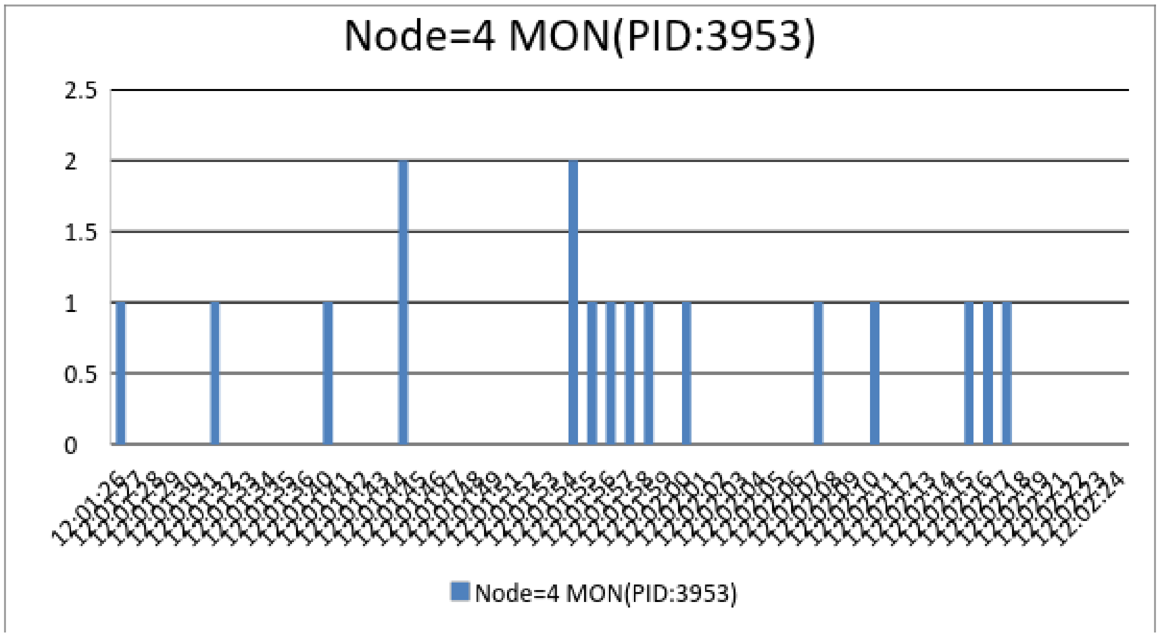
Node 3 (1 monitor, 2 OSD's in cluster):

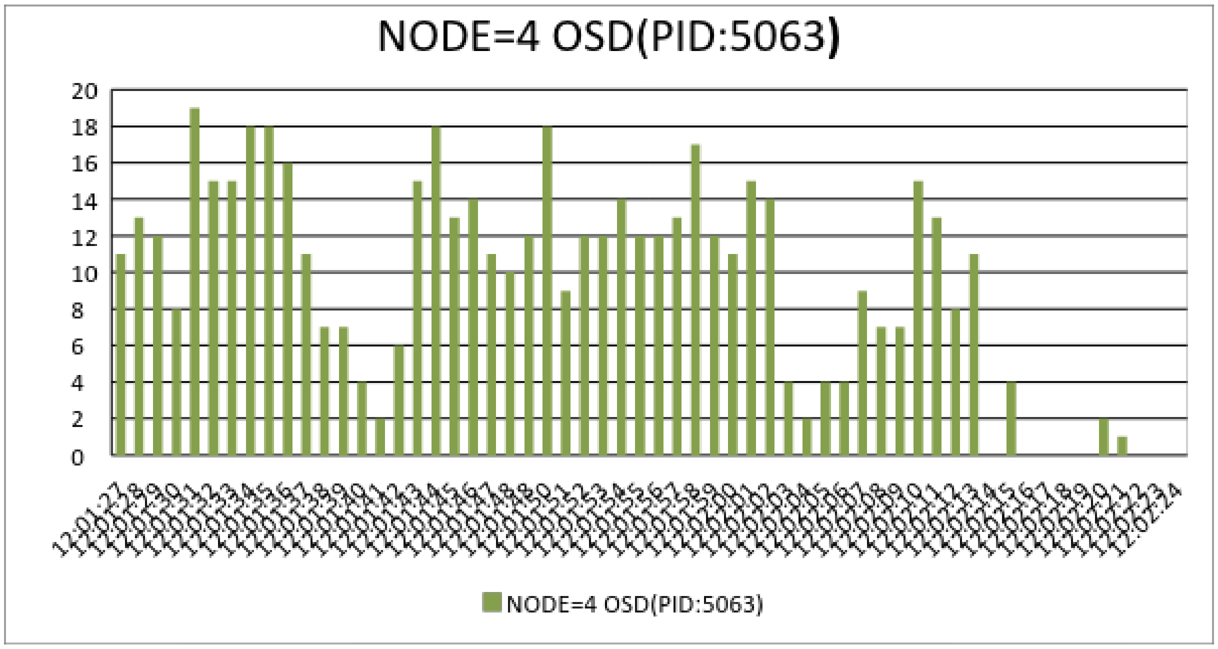

Node=3 MON(PID:27671)
1.2
1
0.8
0.6
0.4
0.2
0
12:01:21
12:01:22
12:01:23
12:01:24
12:01:25
12:01:26
12:01:27
12:01:28
12:01:29
12:01:30
12:01:31
12:01:32
12:01:33
12:01:34
12:01:35
12:01:36
12:01:37
12:01:38
12:01:39
12:01:40
12:01:41
12:01:42
12:01:43
12:01:44
12:01:45
12:01:46
12:01:47
12:01:48
12:01:49
12:01:50
12:01:51
12:01:52
12:01:53
12:01:54
12:01:55
12:01:56
12:02:04
12:02:05
12:02:06
12:02:07
12:02:08
12:02:09
12:02:10
🟦 Node=3 MON(PID:27671)


NODE=3 OSD(PID:23344)
2.5
2
1.5
1
0.5
0
12:01:30
12:02:19
🟩 NODE=3 OSD(PID:23344)



Node 4 (1 monitor, 1 OSD in cluster):





Observation for CPU utilization:

For CEPHFS:

1. CEPHFS has 2- 3 processes running on each node in the mulit-node cluster.
2. From Node 2, 3 and 4 we can see that not much of CPU activity on the monitor processes.
3. There is significant CPU usage visible in the one of the processes on the OSD nodes.
4. CEPHFS monitor retrieves the latest copy of cluster map which gives information on the CEPHFS monitor, OSD and metadata server. It does not provide any info on object locations/storage.
5. This is computed on the fly at OSD daemons using the CRUSH algorithm which reflects the high CPU usage at the Object Storage Devices (OSD).

For HDFS:

1. HDFS has one process running on the namenode and one process on each datanode.
2. The CPU utilization for the process on the namenode is minimal.
3. We can easily observe that when the data is written on the datanodes, there is considerable increase in the CPU utilization.

Disk I/O UTILIZATION:

Test scenario:

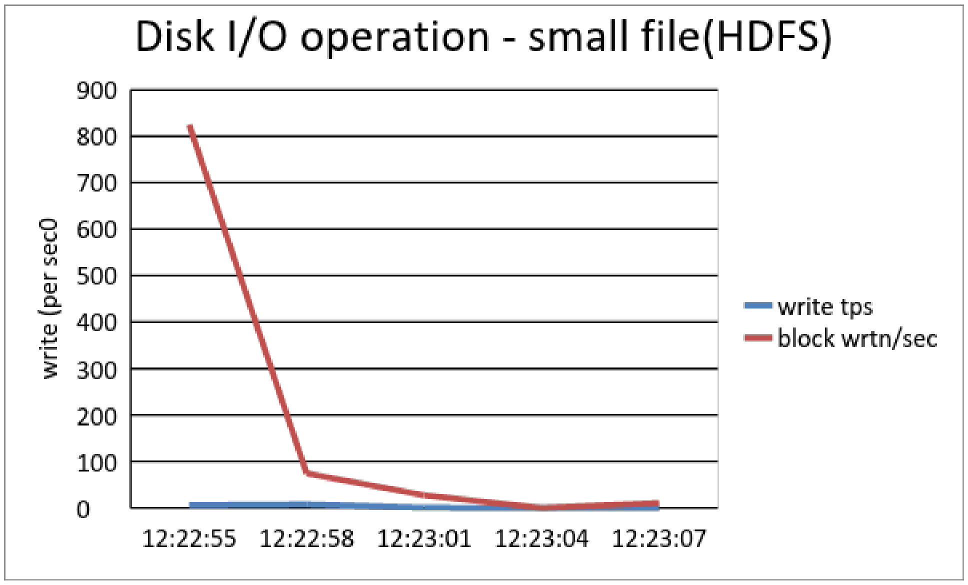
The test was carried out using the concept in mounted/installed file systems:

Check (blks/s)/(r+w/s) < 16Kb (~32 blocks), the I/O is predominantly random.

If the ratio is > 128Kb (~256 blocks), it is predominantly sequential. This perspective helps analyze file system I/O latency.

1. For HDFS (small file size):

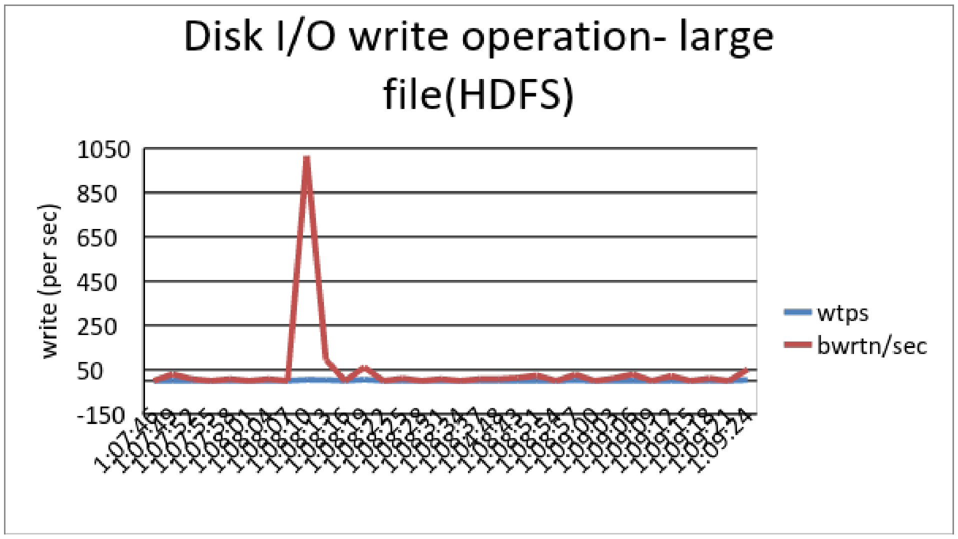
12:22:52	**write tps**	**block wrtn/sec**
12:22:55	6	818.67
12:22:58	7.67	66.67
12:23:01	0.67	26.67
12:23:04	0	0
12:23:07	0	10.67



For HDFS (large file size):

1:07:43 AM	**wtps**	**bwrtn/sec**
1:07:46	0	0
1:07:49	0.67	29.33
1:07:52	0.67	8
1:07:55	0	0
1:07:58	0.67	8
1:08:01	0	0
1:08:04	0.67	8
1:08:07	0	0
1:08:10	4.67	1010.67
1:08:13	3.67	90.67
1:08:16	0	0
1:08:19	5.33	56
1:08:22	0	0
1:08:25	0.67	10.67
1:08:28	0	0
1:08:31	0.67	8
1:08:34	0	0
1:08:37	0.67	8
1:04:48	0.67	8
1:08:43	0.33	13.33
1:08:51	0.67	24
1:08:54	0	0
1:08:57	2	26.67
1:09:00	0	0

1:09:03	0.67	10.67
1:09:06	0.67	29.33
1:09:09	0	0
1:09:12	1	21.33
1:09:15	0	0
1:09:18	0.67	10.67
1:09:21	0	0
1:09:24	3.67	48



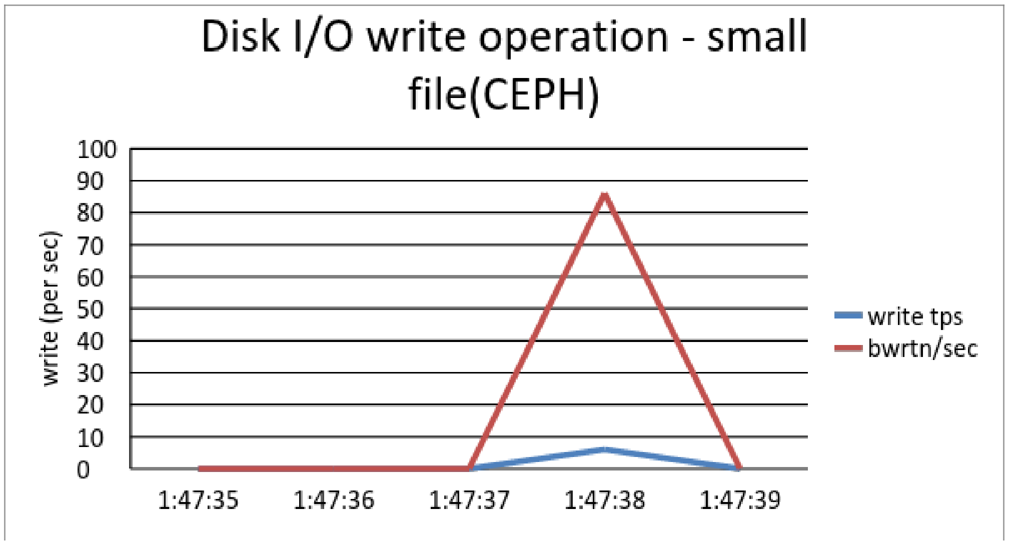
Conclusion:

1. Ratio computed for the maximum peak: (818.67*512)/6 >128 kB, -> sequential disk access followed for small files and is much faster.
2. For large file sizes, ratio computer for the maximum peak = (blks/s)/(w/s) = 1.1Mbytes > 128 kb pattern is sequential disk access in this case for large file writes. (less seek activity and rotational delay) . Overall HDFS is faster at disk access for small as well as large files.

2. For CEPHFS (small file size):

1:47:34	write tps	bwrtn/sec

1:47:35	0	0
1:47:36	0	0
1:47:37	0	0
1:47:38	6	80
1:47:39	0	0

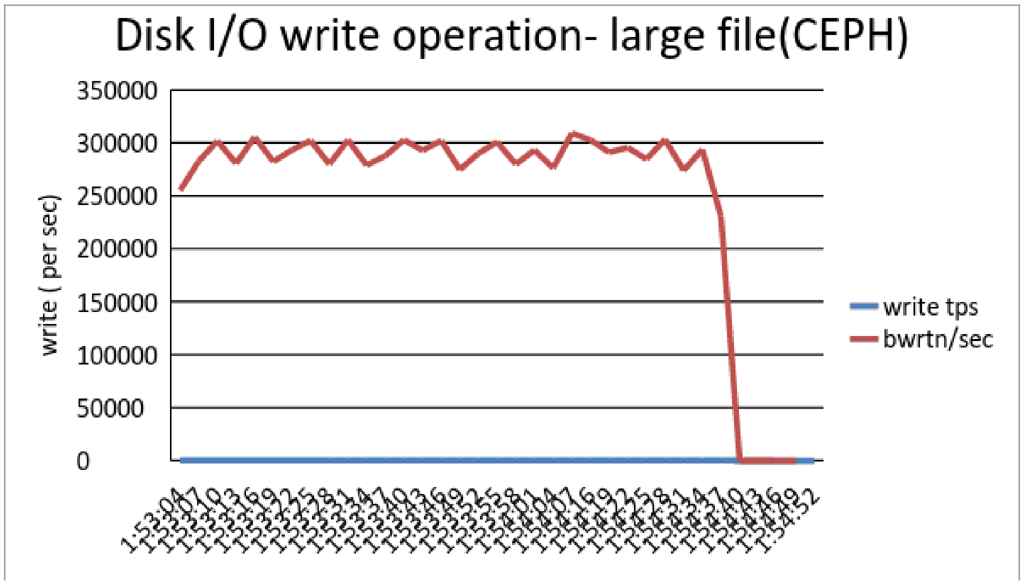


For CEPHFS (large file size):

	write tps	bwrtn/sec
1:53:04	267	254845.3
1:53:07	279.33	282432
1:53:10	305	301554.7
1:53:13	278	280293.3
-1:53:16	304	305205.3
1:53:19	282.33	282026.7

1:53:22	292.67	293021.3
1:53:25	299.33	301778.7
1:53:28	277.67	279616
1:53:31	299	302453.3
1:53:34	281	278640
1:53:37	284	287456
1:53:40	295.33	302421.3
1:53:43	289.67	292600
1:53:46	294.67	301738.7
1:53:49	271.33	274504
1:53:52	287	289869.3
1:53:55	294.67	300397.3
1:53:58	277	279616
1:54:01	286	292864
-1:54:04	274.67	275901.3
1:54:07	204.33	308925.3
1:54:16	298	302453.3
1:54:19	284	290816
1:54:22	288.33	295205.3
1:54:25	281.67	284325.3
1:54:28	296	302765.3
1:54:31	268	273776
1:54:34	287	293549.3
1:54:37	228.67	230224
1:54:40	0	0

1:54:43	2.67	53.33
1:54:46	0	0
1:54:49	0	0
1:54:52	0	



Conclusion:

1. Ratio computed for the max peak activity (small file): (blks/s)/(w/s) = (80*512/6)= 6.26kb. with 6 transfers per second. I/O access mainly random for small files.
2. For large files in CEPHFS, ratio computed for max peak value: (305205.3*512/304) = 514 kb, for large file upload too, behaving similar to HDFS and involves lesser seek time and rotational delays for mounted disk access. CEPHFS is better for large file sizes disk accesses.

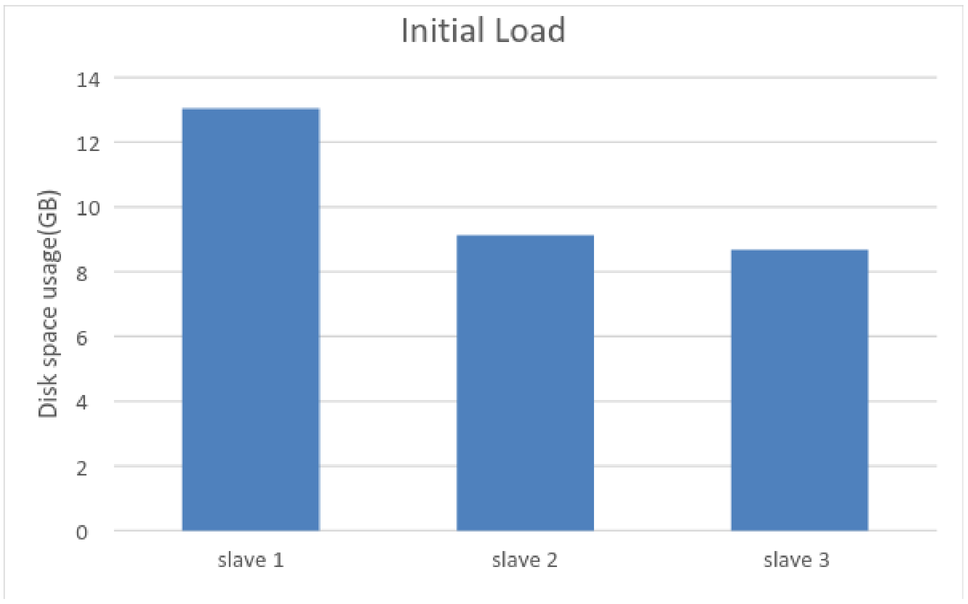
FAULT TOLERANCE (LOAD BALANCING ASPECT):

1. For HDFS:

1. Initial phase: total = 30.84 GB (no change/balancing changes)

Data Node	usage
slave 1	13.04

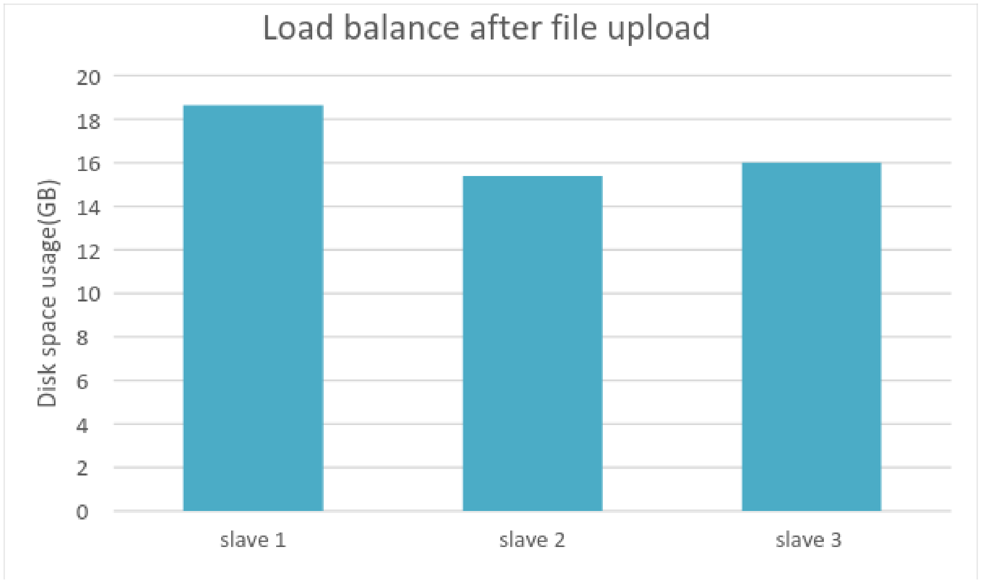
slave 2	9.12
slave 3	8.68



2. Upload 10 GB of data, 20 GB Data added (replication factor = 2)

   Load balance at the data nodes:

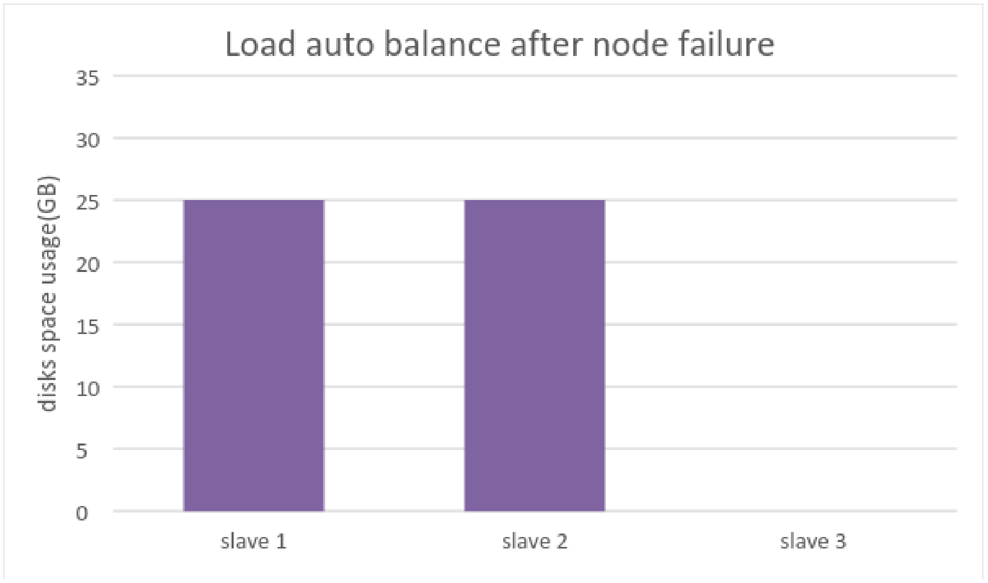
Data Node	**usage**
slave 1	18.65
slave 2	15.4
slave 3	16.02



3. kill/crash a data node:(slave node 3)

Auto balancing of load under node failure conditions:

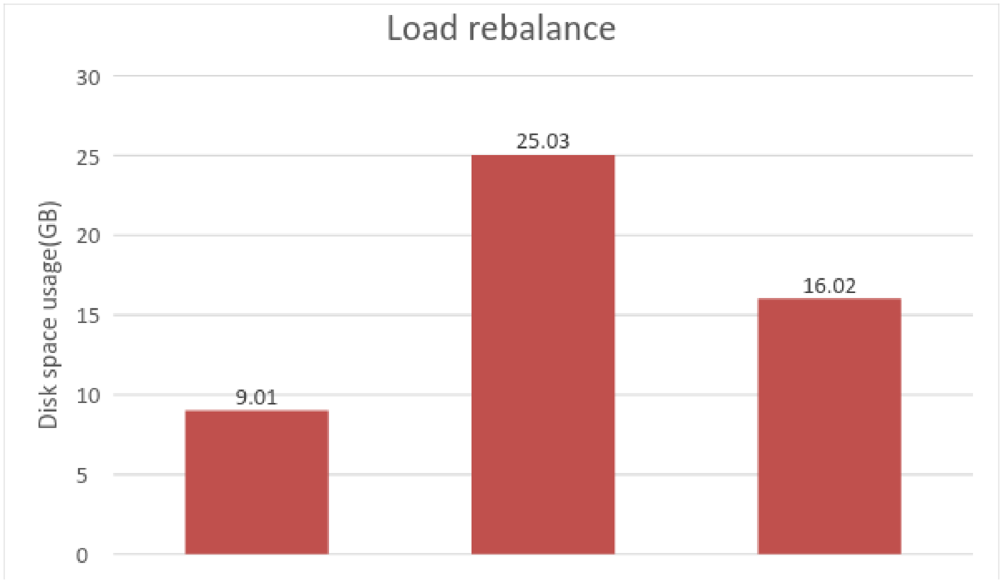
Data Node	**usage**
slave 1	25.03
slave 2	25.03
slave 3	0



4. rebooting node:

Load auto re-balancing:

Data Node	**usage**
slave 1	9.01
slave 2	25.03
slave 3	16.02



Conclusion:

For HDFS:

HDFS handles fault tolerance and load balancing through the use of heartbeat messages.

Heartbeats from a DataNode also carry information about total storage capacity, fraction of storage in use, and the number of data transfers currently in progress.

These statistics are used for the NameNode's block allocation and load balancing decisions.

The default heartbeat interval is three seconds.

FOR CEPHFS:

Test scenario:

1) disconnected node MATTERN at approximately 12:20:46 - monitor clustered registered this disconnect immediately
2) At 12:23:24 the clustered registered that the OSD on MATTERN was not accessible
3) Recovery started at 12:30:12.071824
4) Recovery completed by 12:33:58.274204

Intermediate testing when node down:

Tested write of 10gb file while node was down and recovery was pending
Tested read (cat) (successfully) of 4gb file while node was down and recovery was pending

```
### SCHIPER ###
# ceph -s PERIODIC CLUSTER STATUS check #
```

```
ceph@schiper:/ceph-mnt/mycephfs$ date; ceph -s
Wed May  7 12:20:28 CDT 2014
   cluster 218fa31a-d75b-4006-b069-6c8fa83af754
    health HEALTH_OK
    monmap e1: 3 mons at
{chandy=10.176.68.180:6789/0,lamport=10.176.68.205:6789/0,mattern=10.176.68.137:
                                                                    6789/0}, election epoch 28, quorum 0,1,2
chandy,lamport,mattern
    mdsmap e6: 1/1/1 up {0=lamport=up:active}
    osdmap e39: 3 osds: 3 up, 3 in
     pgmap v43385: 208 pgs, 5 pools, 22105 MB data, 5562 objects
          255 GB used, 649 GB / 953 GB avail
              208 active+clean

ceph@schiper:/ceph-mnt/mycephfs$ date; ceph -s
Wed May  7 12:20:46 CDT 2014
^[[A^[[B    cluster 218fa31a-d75b-4006-b069-6c8fa83af754
    health HEALTH_WARN 1 mons down, quorum 1,2 chandy,lamport
    monmap e1: 3 mons at
{chandy=10.176.68.180:6789/0,lamport=10.176.68.205:6789/0,mattern=10.176.68.137:
                                                                    6789/0}, election epoch 30, quorum 1,2
chandy,lamport
    mdsmap e6: 1/1/1 up {0=lamport=up:active}
    osdmap e39: 3 osds: 3 up, 3 in
     pgmap v43386: 208 pgs, 5 pools, 22105 MB data, 5562 objects
          255 GB used, 649 GB / 953 GB avail
              208 active+clean

ceph@schiper:/ceph-mnt/mycephfs$ date; ceph -s
Wed May  7 12:21:01 CDT 2014
   cluster 218fa31a-d75b-4006-b069-6c8fa83af754
    health HEALTH_WARN 1 mons down, quorum 1,2 chandy,lamport
    monmap e1: 3 mons at
{chandy=10.176.68.180:6789/0,lamport=10.176.68.205:6789/0,mattern=10.176.68.137:
                                                                    6789/0}, election epoch 30, quorum 1,2
chandy,lamport
    mdsmap e6: 1/1/1 up {0=lamport=up:active}
    osdmap e39: 3 osds: 3 up, 3 in
     pgmap v43387: 208 pgs, 5 pools, 22105 MB data, 5562 objects
          255 GB used, 649 GB / 953 GB avail
              208 active+clean

ceph@schiper:/ceph-mnt/mycephfs$ date; ceph -s
Wed May  7 12:21:16 CDT 2014
   cluster 218fa31a-d75b-4006-b069-6c8fa83af754
```

```
    health HEALTH_WARN 1 mons down, quorum 1,2 chandy,lamport
    monmap e1: 3 mons at
{chandy=10.176.68.180:6789/0,lamport=10.176.68.205:6789/0,mattern=10.176.68.137:
                                                              6789/0}, election epoch 30, quorum 1,2
chandy,lamport
    mdsmap e6: 1/1/1 up {0=lamport=up:active}
    osdmap e39: 3 osds: 3 up, 3 in
     pgmap v43387: 208 pgs, 5 pools, 22105 MB data, 5562 objects
          255 GB used, 649 GB / 953 GB avail
              208 active+clean
```

==

```
ceph@schiper:/ceph-mnt/mycephfs$ date; ceph -s
Wed May  7 12:23:41 CDT 2014
^[[A2014-05-07 12:23:44.213232 7f055876b700  0 -- :/1006526 >> 10.176.68.137:6789/0
pipe(0x7f055400e400 sd                                                                      =3 :0 s=1
pgs=0 cs=0 l=1 c=0x7f055400e660).fault
   cluster 218fa31a-d75b-4006-b069-6c8fa83af754
    health HEALTH_WARN 129 pgs degraded; 129 pgs stuck unclean; recovery 3302/11126 objects
degraded (29.                                                                   678%); 1/3 in osds are
down; 1 mons down, quorum 1,2 chandy,lamport
    monmap e1: 3 mons at
{chandy=10.176.68.180:6789/0,lamport=10.176.68.205:6789/0,mattern=10.176.68.137:
                                                              6789/0}, election epoch 30, quorum 1,2
chandy,lamport
    mdsmap e6: 1/1/1 up {0=lamport=up:active}
    osdmap e41: 3 osds: 2 up, 3 in
     pgmap v43394: 208 pgs, 5 pools, 22105 MB data, 5563 objects
          255 GB used, 649 GB / 953 GB avail
          3302/11126 objects degraded (29.678%)
               79 active+clean
              129 active+degraded
```

==

1)**reconnect** MATTERN at approximately
2)read 4gb file (single) at approximately 13:12:56 (time cat single)

```
real    11m19.463s
user    0m0.000s
sys     0m14.556s
```

Tested read (cat) (successfully) of 4gb file after node was initially connected, recognized immediately by the monitoring and OSD clusters.

### SCHIPER ###
# ceph -s PERIODIC CLUSTER STATUS check #

```
ceph@schiper:/ceph-mnt/mycephfs$

ceph@schiper:/ceph-mnt/mycephfs$ date; ceph -s
Wed May  7 13:06:31 CDT 2014
2014-05-07 13:06:32.057202 7ff5acfb4700  0 -- :/1007846 >> 10.176.68.137:6789/0
pipe(0x7ff5a800e400 sd=3 :0 s=1 pgs=0 cs=0 l=1 c=0x7ff5a800e660).fault
    cluster 218fa31a-d75b-4006-b069-6c8fa83af754
     health HEALTH_WARN 1 mons down, quorum 1,2 chandy,lamport
     monmap e1: 3 mons at
{chandy=10.176.68.180:6789/0,lamport=10.176.68.205:6789/0,mattern=10.176.68.137:6789/0},
election epoch 30, quorum 1,2 chandy,lamport
     mdsmap e6: 1/1/1 up {0=lamport=up:active}
     osdmap e43: 3 osds: 2 up, 2 in
      pgmap v43692: 208 pgs, 5 pools, 22105 MB data, 5562 objects
            196 GB used, 404 GB / 633 GB avail
                 208 active+clean

ceph@schiper:/ceph-mnt/mycephfs$ date; ceph -s
Wed May  7 13:08:17 CDT 2014
    cluster 218fa31a-d75b-4006-b069-6c8fa83af754
     health HEALTH_OK
     monmap e1: 3 mons at
{chandy=10.176.68.180:6789/0,lamport=10.176.68.205:6789/0,mattern=10.176.68.137:6789/0},
election epoch 32, quorum 0,1,2 chandy,lamport,mattern
     mdsmap e6: 1/1/1 up {0=lamport=up:active}
     osdmap e45: 3 osds: 3 up, 3 in
      pgmap v43697: 208 pgs, 5 pools, 22105 MB data, 5562 objects
            267 GB used, 637 GB / 953 GB avail
                 208 active+clean


ceph@schiper:/ceph-mnt/mycephfs$ date; ceph -s
Wed May  7 13:08:24 CDT 2014
    cluster 218fa31a-d75b-4006-b069-6c8fa83af754
     health HEALTH_OK
     monmap e1: 3 mons at
{chandy=10.176.68.180:6789/0,lamport=10.176.68.205:6789/0,mattern=10.176.68.137:6789/0},
election epoch 32, quorum 0,1,2 chandy,lamport,mattern
     mdsmap e6: 1/1/1 up {0=lamport=up:active}
     osdmap e45: 3 osds: 3 up, 3 in
      pgmap v43697: 208 pgs, 5 pools, 22105 MB data, 5562 objects
            267 GB used, 637 GB / 953 GB avail
                 208 active+clean
```

===