CS425 DS MP3 - Simple Distributed File System

Contributors: Shubham Singhal & Nirupam K N (ss77, nirupam2)@illinois.edu

1. Design & Implementation:

We are storing total 4 replicas of a file in the distributed file system. Files are not sharded in the system. Here we have chosen W+R = 5 where W=4 and R=1.

PUT: put <localfilename> <sdfsfilename>

A file is put into 4 nodes based on the neighbourship list from the local file path to sdfs file path.

GET: get <sdfsfilename> <localfilename>

Sdfs file is copied from one of the replicas and stored in the local file path.

DELETE: delete <sdfsfilename>

Deletes sdfs file from all the replicas in a distributed file system.

LIST: ls <sdfsfilename>

List all the nodes where this file is currently being stored

STORE: List all sdfs files currently being stored at this node.

Here sdfsfilename is an arbitrary string while localfilename is the Unix-style local file system name.

Update: Is similar to PUT, since updates are comprehensive, i.e., they send the entire file, not just a block of it.

Re-replication: Since each file has 4 replicas. When a node fails, node with highest replica-id of that file will take the lead to re-replicate the file in any of the servers which are currently up and running in the system. This leader election is per file basis and this leader is not a global leader.

When two nodes tries to put into the same sdfs filename, a warning prompt is popped up to the second Put initiator. If the user doesn't give any input (Say Y/N): By default the Put command is ignored.

Reads and writes occur between peer nodes directly without the involvement of the master. We do not perform any caching of the data, as all the nodes have information about the filetable. Membership list populated from MP2 is used as reliable list of nodes that are alive in order to choose the suitable substitute for replication and also for the node to get the files.

2) Using MP1 for debugging MP3:

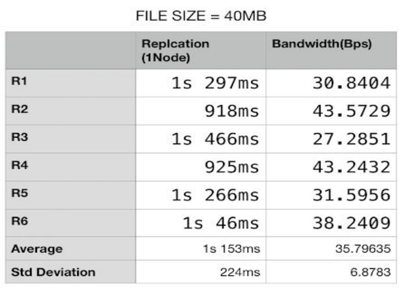
MP1 was used regularly to catch errors while debugging and log the debug/failure messages.

3) Performance:

(i) Re-replication time and bandwidth upon a failure

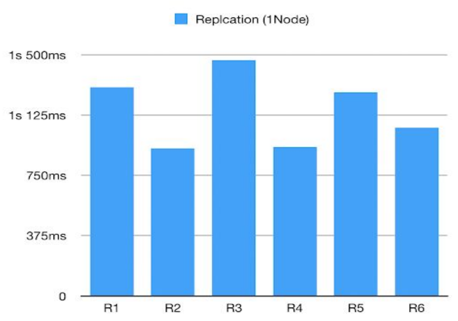
Avg Bandwidth usage on failure: 35.79635 Bps Average Time: 1s 153ms Standard Deviation: 224ms

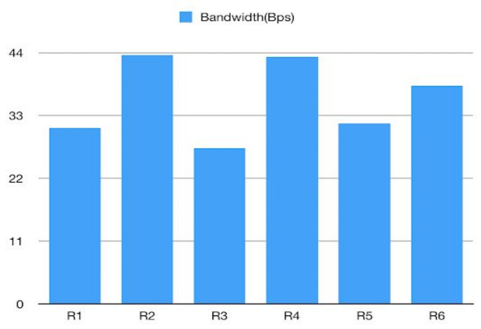
The above mentioned values are re-replication time and bandwidth of 1 file of size 40 MB to only one node.



FILE SIZE = 40MB

	Replcation (1Node)	Bandwidth(Bps)
R1	1s 297ms	30.8404
R2	918ms	43.5729
R3	1s 466ms	27.2851
R4	925ms	43.2432
R5	1s 266ms	31.5956
R6	1s 46ms	38.2409
Average	1s 153ms	35.79635
Std Deviation	224ms	6.8783

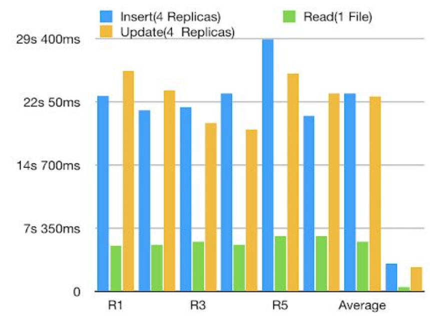


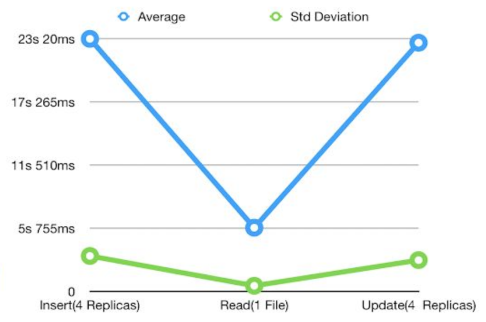


(ii) Times to insert, read, and update, file of size 25 MB, 500 MB (6 total data points), under no failure:

FILE SIZE = 500 MB

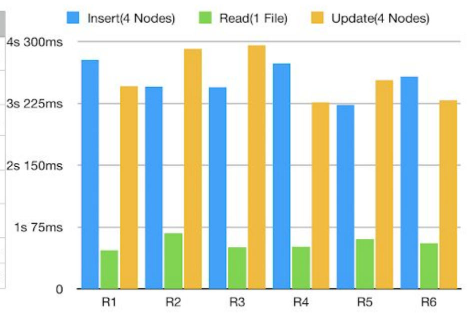
	Insert(4 Replicas)	Read(1 File)	Update(4 Replicas)
R1	22s 739ms	5s 297ms	25s 679ms
R2	21s 97ms	5s 453ms	23s 385ms
R3	21s 461ms	5s 808ms	19s 602ms
R4	23s 37ms	5s 417ms	18s 853ms
R5	29s 330ms	6s 434ms	25s 374ms
R6	20s 452ms	6s 479ms	23s 37ms
Average	23s 20ms	5s 815ms	22s 665ms
Std Deviation	3s 244ms	526ms	2s 863ms

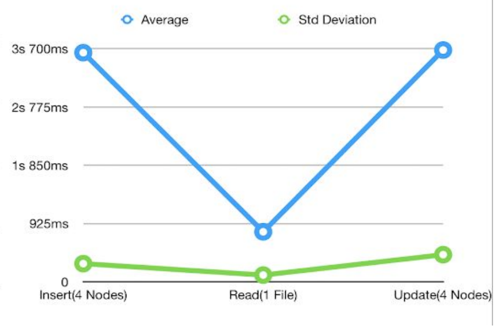




FILE SIZE = 25MB

	Insert(4 Nodes)	Read(1 File)	Update(4 Nodes)
R1	3s 977ms	673ms	3s 521ms
R2	3s 518ms	971ms	4s 170ms
R3	3s 507ms	723ms	4s 233ms
R4	3s 918ms	738ms	3s 243ms
R5	3s 202ms	862ms	3s 627ms
R6	3s 692ms	798ms	3s 276ms
Average	3s 636ms	794ms	3s 678ms
Std Deviation	289ms	108ms	431ms



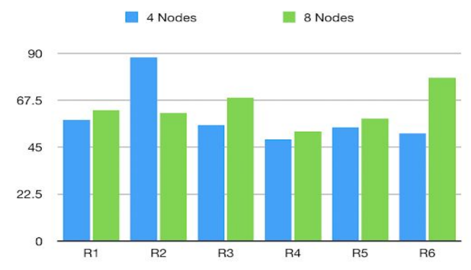


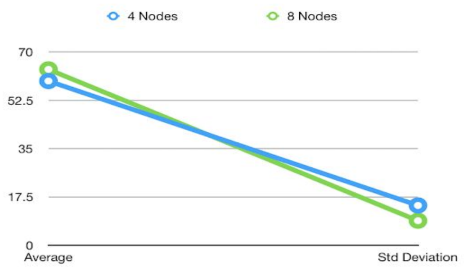
Statistics for inserts and updates are almost similar, since they perform effectively the same operations.
Time to read the file is in milliseconds as we are reading from the first replica which is up and running instead of reading from all the 4 replicas (as Read Quorum is 1)

(iii) Time to store the entire English Wikipedia corpus into SDFS with 4 machines and 8 machines (not counting the master)

Table 1

	4 Nodes	8 Nodes
R1	58.17974	62.78568
R2	88.15857	61.34589
R3	55.55088	68.71009
R4	48.77385	52.50971
R5	54.58983	58.64981
R6	51.67458	78.27893
Average	59.48791	63.71335
Std Deviation	14.41567	8.88538





Statistics show that the average and standard deviation is more or less similar, because the number of replicas is constant and it doesn't depend on the size of system