Machine Problem 3 Report

1. Design

We design file distributed system upon our MP2 and MP1 distributed system structure. This system can realize basic functions including file transition, file replication, failure detection, file re-replication, and leader selection.

a) **Master(leader)**: master node contains all the file storage information in SDFS system. Whenever a node tries to perform command related to access files, it needs to send the request to master and ask it for the nodes information in SDFS system, then it can perform the command according to the node list. Besides, nodes need to send the file updates to master whenever they have a file update action.

b) **File acquisition:** file can be obtained by 'get' command and 'get version' command. 'get' command obtain the latest version of the file in SDFS system. 'get version' can get k versions of the files that store in SDFS system.

c) **File transition**: file can be put to 4 machines for replication in SDFS system.

d) **File deletion:** All versions and replicas of the file would be deleted in the SDFS system. When a node first join or rejoin the system, it would not have former SDFS files.

e) **Node fail:** when a node fail, if it is not a master, master would delete the failed node information and perform re-replication request to other node. It the failed node is master, new master election would performed and re-replication would be done after the new master is elected.

f) **File version:** Every file has up to 5 versions in the SDFS system, if a new version is put in the system and there are already 5 versions of the file, the system would delete the oldest version.

g) **Show file list:** 'store' command can show the SDFS file list on each node, and 'ls file' command can show the nodes that store the file in SDFS system.

h) **Master(leader) election:** if the master node fails, then the system would elect the node with the smallest ID to be the new master. And all nodes would their file information to new master to renew the master info list. And master node would go through its master list to see which file needs to re-replica, then it would tell the machines that have the remain replica to perform 'put' command to the other machines, so that the number of replicas of file can be satisfied.

With the experience earned from MP1, We can debug the file command in MP3 more easily.

2. Measurement

(i) Re-replication time and bandwidth upon a failure (40 MB)

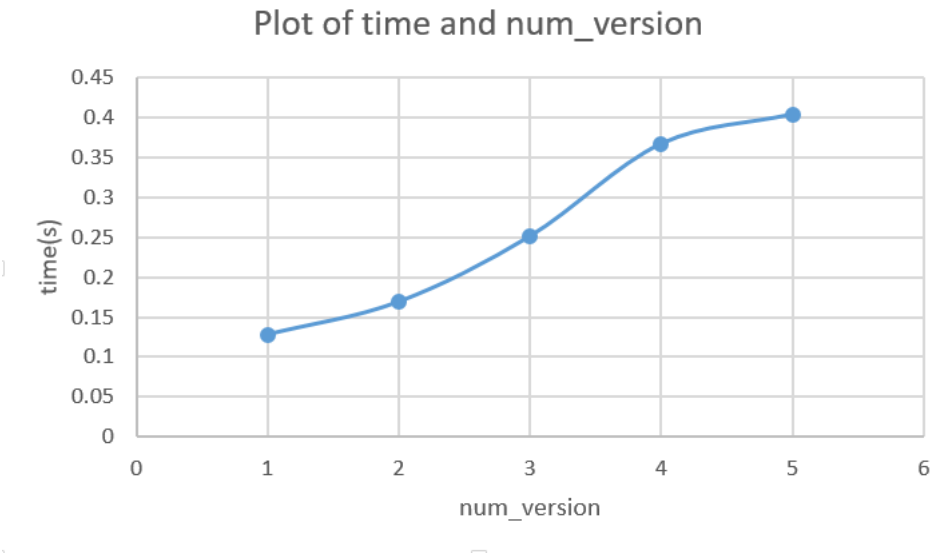
- Re-replication time: 0.8 second
- Bandwidth: 40MB/0.8s = 50MB/s

(ii) Times to insert, read, and update, file of size 25 MB, 500 MB, under no failure;

Average Time(second)	Insert	Read	Update
20MB file	0.558	0.102	0.502
500MB file	7.375	2.143	7.293

(iii) Plot the time to perform get-versions as a function of num-versions

We plot time of 1-5 num-versions.



The time cost of file is approximately linearly related to the number of versions of getting file. When we get more than one version of file, we append all the versions to the end of the local file that we use to store the content, so the time cost and num_version would be linearly related.

(iv) Time to store the entire English Wikipedia corpus into SDFS with 4 machines and 8 machines

We use seconds as time measurement.

Trial	1	2	3	4	5	Average	Standard Deviation
4 machines	21.667	14.528	14.79	15.612	13.518	16.023	2.900
8 machines	13.334	12.927	14.574	13.214	12.322	13.274	0.738

We use 4 replicas for each file in SDFS, so the procedure of storing the file into SDFS with 4 machines and 8 machines is similar. So the time cost of 4 machines and 8 machines is similar as well. The difference between the time cost may due to network delay.