CS425 MP3 Report

Group 43: Siti Zhang (sitiz2) & Mingrui Yin (mingrui4)

Design

The simple distributed file system consists of the client component for user command and the server for one SDFS master and nine SDFS slaves. The slave saves the file name and version as the local metadata. The master stores the information of replicas and versions for file. All requests will go through the SDFS master.

Put: Put request will first check whether the local file exists. Then the request is sent to the SDFS master. And the master randomly chooses 4 members in its membership list as the replicas for the local file. Then the master will update its metadata and send the replicas information to the client, so the client can start putting localfiles in the SDFS. Besides, quorum is also used in determining whether the put is done. If we already put 3 files success, we will know put request is finished.

Get: Get request is sent to the SDFS master firstly. And it will get the replicas address and the versions of the sdfsfile. Then the client will get sdfsfile from replicas and we also use quorum, once we received the last version of sdfsfile from 3 replicas, the get request is finished.

Delete: Delete request also get the replicas and the version information of the sdfs file from the master. Then the master will remove the metadata of the sdfsfile and the slave will also delete the all versions of sdfsfile and remove the related metadata.

Ls: The master will send the replicas address about the sdfsfile to the client.

Store: It get the file names that stored in this process from its local metadata.

Get-versions: Get-versions request is similar to the get request. But the users can specify the last number of the versions they'd like to get. We will merger the different versions of file into one local file.

Replication: Since the SDFS is tolerant up to three machine failures at a time, we build 4 replicas for each file. We use active replication strategy. If one failure happens, the master will also check the number of the replicas. And if the number n is less than 4, it will select (4-n) members that alive in current membership list as the new replicas and start a re-replicate process. One alive replica will be chosen to copy the file to the new replicas. And all the time we have 4 replicas for each file.

Re-election: We use bully algorithm for re-election process. If master fails, all of the alive vms vote individually and choose the first vm in recent membership list as the master, then the vm who get the most votes will be re-elect as new master and copy the metadata from the old.

Useful of MP1

MP1 is helpful, since we can do a distributed grep of the log file in any machine to find out the useful information for debugging. And MP2 also plays an important role in failure detector.

Measurements

The measurements below shows in the following form:

a. re-replication time and bandwidth upon a failure (40 MB file).

Bandwidth is not stable but average is about 40MB/s, and re-replication time is about 1s.

b. times to insert, read, and update, file of size 25 MB, 500 MB.
Insert(put) and update time is similar, but read time is less.

c. time to store the entire English Wikipedia corpus into SDFS with 4 machines and 8 machines.
4 machines and 8 machines time are similar because the put progress is working simultaneous in different vms.

	1	2	3	4	5	Average	STD
re-replica	1.76s	0.85s	0.79s	1.08s	0.90s	1.08s	0.397
bandwidth	23.1MB/s	52.6MB/s	54.9MB/s	39.6MB/s	45.2MB/s	43.1MB/s	12.71
20M put	2.19s	2.05s	1.97s	2.23s	1.93s	2.08s	0.132
20M read	2.10s	2.04s	1.90s	2.10s	1.97s	2.02s	0.087
20M update	2.18s	2.00s	1.96s	2.21s	1.92s	2.05s	0.132
500M put	18.14s	17.11s	17.84s	17.49s	16.98s	17.51s	0.486
500M read	7.15s	7.93s	7.84s	6.99s	7.45s	7.47s	0.413
500M update	18.16s	17.85s	17.45s	17.02s	17.99s	17.69s	0.459
store wiki in4	47.76s	45.82s	46.95s	42.25s	43.14s	45.18s	2.39
store wiki in8	42.15s	49.48s	41.27s	46.19s	47.34s	45.29s	3.48

d. We also plot the time to perform get-versions as a function of num-versions for a 20M file.For get-versions request, it is obvious that the time will increase as the number of version increases.