Machine Practice 3
CS425 Distributed System Networking
Prof. Indranil Gupta

Chen Zhu
Weiran Lin
Date: November 5, 2018

1 Objectives

- Implement a versioned Distributed File System which supports put, get, delete operations. It tolerates up to three machine failures once and quickly re-replicate files to other live machines.

2 System Design

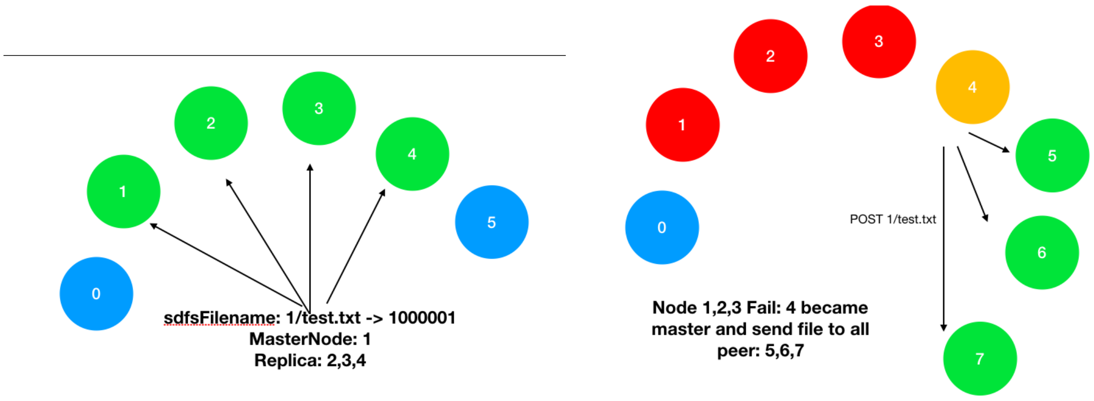

0
1
2
3
4
5
sdfsFilename: 1/test.txt -> 1000001
MasterNode: 1
Replica: 2,3,4
0
1
2
3
4
5
6
7
POST 1/test.txt
Node 1,2,3 Fail: 4 became master and send file to all peer: 5,6,7

Fig. 1A.

Fig. 1B.

Figure 1: Figure 1A shows the put algorithm of our system. Local server sends the local file to four machines according to the hash value of the sdfsfilename. The server responds to client until all replica machines acknowledge the message. Figure 1B shows the file re-replication procedure. When Master fails, the next live machine would know whether it becomes master of SDFS files and then update its masterFile map. The new master would then send SDFS files, to all its three peers. If slave fails, the master would send replication to all new peers and delete the replica on old peers.

Since SDFS can tolerate up to three machine failures at a time, we have four file replication in the system. We designate W as 4 and R as 1. The first live machine corresponding to a SDFS file is the master. It would present file re-replication when it finds replication machine fails.

2.1 Data Structure Design

Each machine maintains a sfile map and a masterFile map.

- **sfile**: It stores information of all files in SDFS including the sdfsfilename and all its timestamps. It lets server know what SDFS files are stored and all their versions.
- **masterFile**: It stores information of all files in SDFS including the sdfsfilename and whether the machine is the master of the SDFS file. It lets server know whether it is the master of SDFS files stored in it.

2.2 Design of Special Operations

- **get-versions sdfsfilename num-versions localfilename**: Local server sends GET HTTP request to all replication machines of the sdfsfilename. Replication machine would search its sfile map and find lattest num-versions versions according to the sdfsfilename.

2.2.1 Extra: Rejoin Strategy

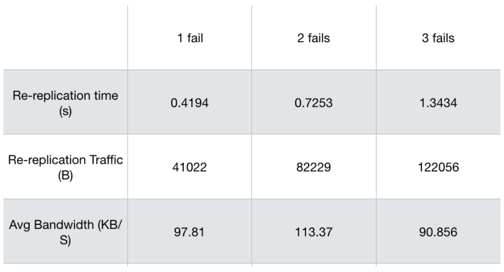
- **Master rejoins**: The old master would send DELETE HTTP request to the last peer to let the peer delete SDFS files which have new master. Then, the old master would update its masterFile map.
- **Slave rejoins**: The master would compare the old peer list and new peer list. For all SDFS files it masters, it will send these files to the new peerlist, and send DELETE HTTP request to the last peer in the old peer list on these files.

2.3 MP1 Usage

Each host will generate logs on HTTP requests status and re-replication requests and responds, and MP1 is used for grep log message from all hosts.

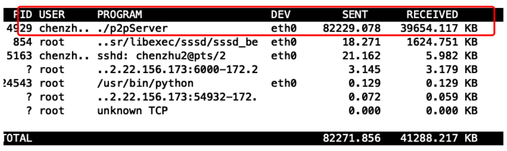
3 Performance Analysis

3.1 Bandwidth During Re-replication



	1 fail	2 fails	3 fails
Re-replication time (s)	0.4194	0.7253	1.3434
Re-replication Traffic (B)	41022	82229	122056
Avg Bandwidth (KB/S)	97.81	113.37	90.856

Fig. 2A. The Avg Bandwidth and total traffic when certain numbers of VMs fails. The replica file size is 40MB. We can see that the total traffic is the number of the VMs which fail simutaneously, which shows our design doesn't include any redundant traffics

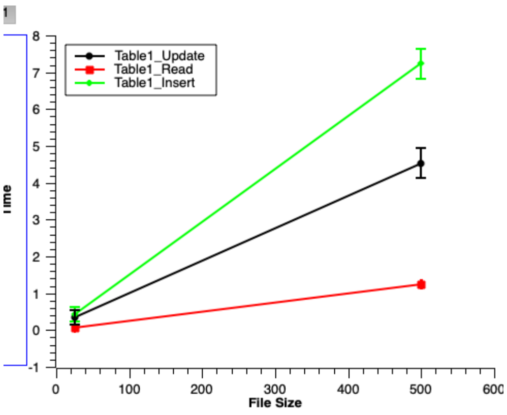


```
  PID USER     PROGRAM                      DEV        SENT      RECEIVED
 4929 chenzh.. ./p2pServer                  eth0   82229.078   39654.117 KB
  854 root     ..sr/libexec/sssd/sssd_be    eth0      18.271    1624.751 KB
 5163 chenzh.. sshd: chenzhu2@pts/2         eth0      21.162       5.982 KB
    ? root     ..2.22.156.173:6000-172.2               3.145       3.179 KB
24543 root     /usr/bin/python              eth0       0.129       0.129 KB
    ? root     ..2.22.156.173:54932-172.               0.072       0.059 KB
    ? root     unknown TCP                             0.000       0.000 KB

TOTAL                                              82271.856   41288.217 KB
```

Fig. 2B. The total traffic of 2 replications when 2 VM fail simutaneously. The bandwidth is measured using nethogs. We can see that 2 40M files are transmitted, thus causing the 82MB Traffic.

3.2 Times to get, put and update a file, and get version



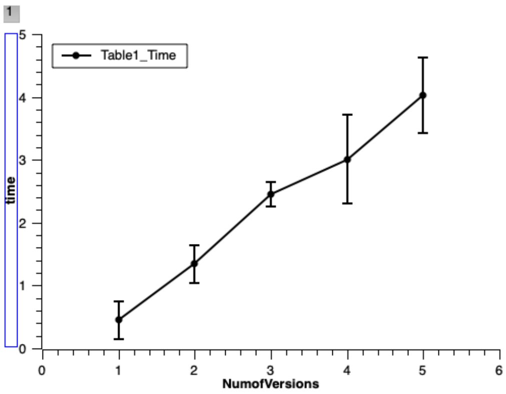


Fig. 2C. We can see that insert operation is slower than update operation. This is caused by TCP slow start algorithm. The read operation is the fastest because it responds to client once receives a response, while the write operation has to wait until all replicas response.

Fig. 2D. We can see that the time increases with the num-versions increase. The standard variation is high because the operation time depends on the size of SDFS file.

3.3 Times spent to send the entire wikipedia to VMs

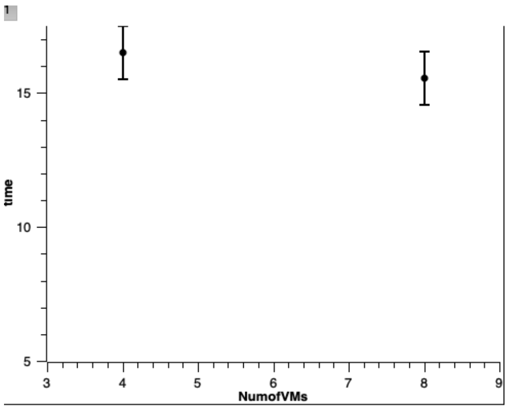


Fig. 2E.

Figure 2: We can see there is little time difference between 5 VMs and 9 VMs. The reason is that the number of replication is the same between these two systems. The standard deviation is high because the entire wikipedia is big, and the transmit process could be easily affected by network situation.