CS425 MP2 Group 44 Failure detector

Shichu Zhu(szhu28), Fan Shi(fanshi2)

SDFS Design

The distributed file system build on the structure of our membership list as well as failure detection protocol. Chord's distributed hash table is the main inspiration of our design. We will apply a hash function to distributed system file names to get a file key and map to a particular node on the ring. Therefore, we utilize per file master to be responsible to performing Write and Read. Since three nodes may fail at once, the file system will store four replicas for each file. For file write, the client will transfer the file to the file master and then send another rpc call to the file master starting file replication. The write will not return until all the replicas have received the file. Here we ensure strong consistency which will allow us to have better read performance. The read operation will send a grpc call to file master and pull the file. The delete operation is easy to implement which we send a rpc call to each client and each replica (including master) will delete the file from its local file system. We create a memory table in each process to keep track of each file (different versions) in the current node. Implement store and get-versions method we will use get the information from the memory table.

To ensure the total ordering for each file, we add a lock to each master. In this way, each file master can only execute one file putting per time. Based on our design, we will wait until the file has been replicated in all the replicas. Therefore, allowing parallel put operation will lower down both writing job. Under this observation, we add the mutex lock for each put operation for file master.

We also deal with node failure. In our design, each node in the system will check if he need to react to the falling node get new files from other nodes. Since we should always 4 replicas for each file, 4 node need to get files from each file master. In put command we use push method, but in failure replication we use pull operation to get the files.

We also use different names systems in distributed system and local file system. We will convert the distributed file name so that the sdfs name support special character such as '\'.

For command get-versions, we simply get all the versions for the file from master.

Measurement :

For each measurement, we will get totally 5 data points and calculate the mean and standard deviation.

1. Replication time for 40 MB file and bandwidth used

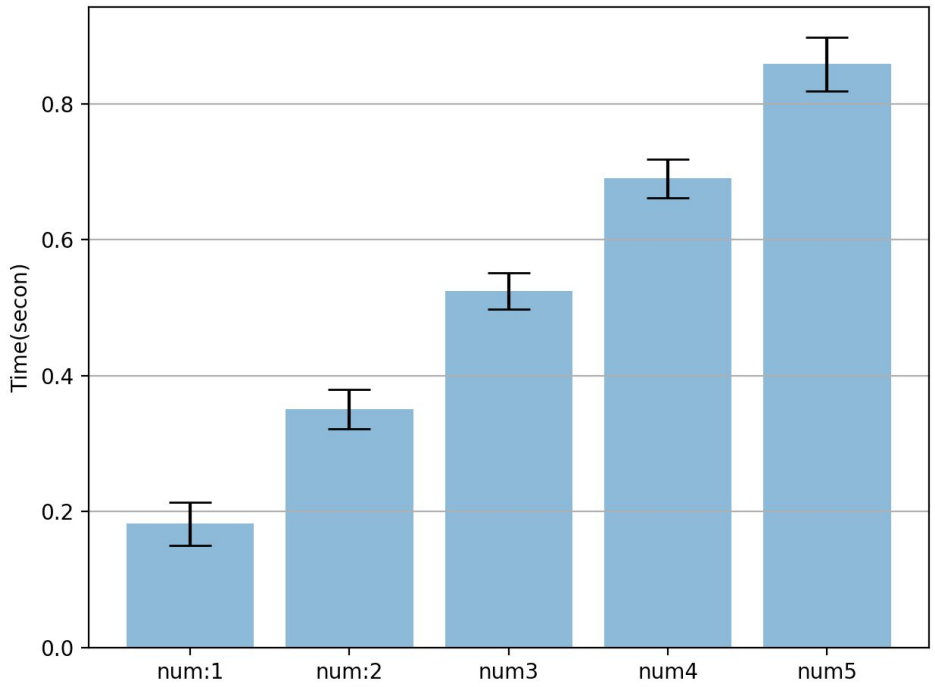
In the measurement, we use 5 nodes and prepare 5 files (distributed into 5 master) and then fail one master

Replication time (mean): 3.68s

Standard Deviation: 0.44

2. Time to insert, delete, update file.

3. Get version plot for different num versions



4. Time to read English Wikipedia corpus into SDFS