Design

File Replication

For communication regarding file system query, we have used a separate udp port and used a tcp port for the actual transfer of files between nodes. Our system has one leader node that stores all file meta-data within the system and processes all put/get requests made from clients.

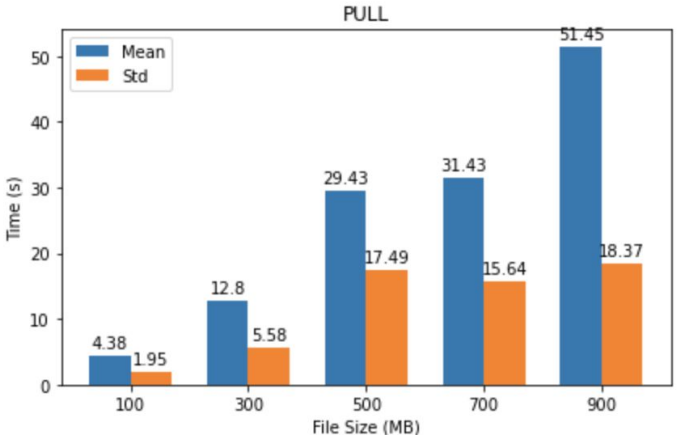
Put: When the request to put a file to DFS, the process sends a put request to the leader. And then, the leader assigns n (to tolerate up to n failures) addresses of alive nodes where requesting nodes then can replicate its file to. Finally, the leader makes the original owner of the file send the file to the assigned nodes to store its replicas.

Get: When the request to pull a file from DFS is made, the process sends a get request to the leader. Then, the leader picks one node who currently possesses the file, and commands the node to send the file to the requesting node.

Leader Election

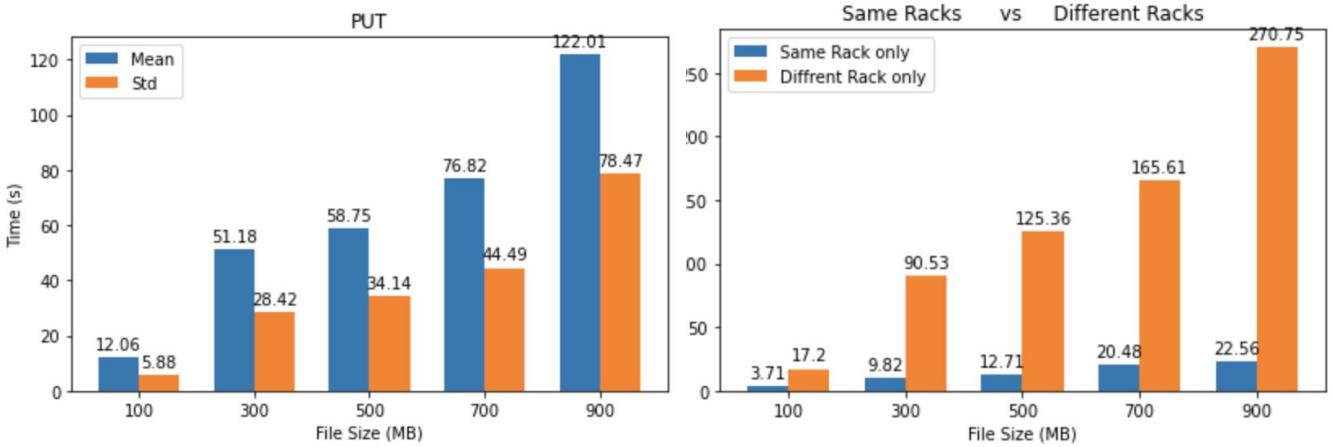
Whenever a leader failure is detected, the detector node initiates the Ring Election protocol, which will eventually pick the node with the highest service id as the new leader as the election message is passed around the ring. In order to solve multiple initiator problems, we have made each process to suppress the election message of any lower-id initiators. At the end of the leader election, the elected node requests other processes to send the list of its distributed files. Then, the new leader finds the missing distributed files (distributed files of ex-leader) and makes replicas of them.

(i) time to get a file vs. file size




```
100 MB | average:  4.38 seconds
(std:  1.95 sec)
300 MB | average:  12.8 seconds
(std:  5.58 sec)
500 MB | average:  29.43 seconds
(std:  17.49 sec)
700 MB | average:  31.43 seconds
(std:  15.64 sec)
900 MB | average:  51.45 seconds
(std:  18.37 sec)
```

(ii) time to put a file vs. file size



In general, the larger the file is, the longer it takes to transfer(put/get) a file among nodes. However, we believe that it also depends on which node the requesting node is getting/putting its file from/to . If the leader assigned the node from a different rack (have the different 3rd Ip Address block) to perform file query, the file transfer will take longer. But if the node from the same rack is assigned, the file transfer time will significantly reduce. We conclude this from the experiments comparing file transfer speed between one that consists of processes with the same rack(3rd IP Address block) and another that consists of processes with different rack(3rd IP Address block). <chart on the right>

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

In order to store the English Wikipedia corpus, we have had one node to put the file into the SDFS. Since the node replicates its file to n (to tolerate up to n failures) alive members within the system, the number of VMs within the system does not affect the time to store the corpus as long as the number of VM is above n. The only factor that matters was that which VMs were included. This is because systems tend to be faster if VMs are in the same rack as explained previously. Since, the file size was 1.3GB > 900MB, it took longer than any other files we tested before.