# CS425 MP2 Report

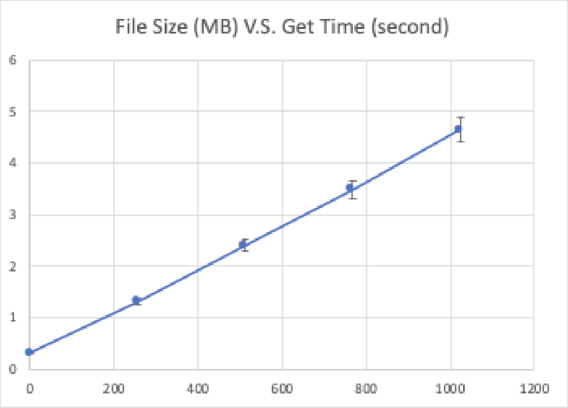
## Yifan Cao (yifanc7) and Enyi Jiang (enyij2)

***Design:*** We have the introducer serve as the master. And for each put/delete operation, the system will store **four** replicas on four servers (because at most three failures at the same time). The master server is responsible for **dealing with all the queries from any server.** To prevent concurrent writes/reads, the master will **process each query in arriving sequences** and save the uncompleted commands in a **queue**. The master will continue to process the next command after receiving the specific ACK (one get or four put/delete ACKs). The several commands procedure are explained as follows:

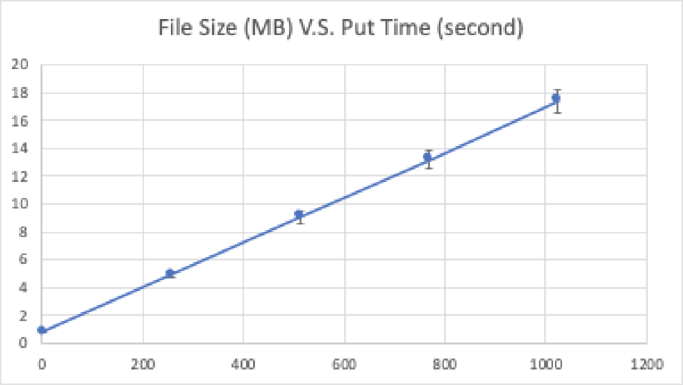
1. ***Put:*** First, the query server will forward the put query to the master, and the master will return the list of server IP addresses that stores the replicas to the query server. Then the query server will send the put request to each replica server. Each replica server will send ACK to the master after receiving the data.
2. ***Get***: The query server will forward the get query to the master, and the master will return one server IP address which has the sdfs replica. Then the query server will send the get request to the replica server, which will write the local file from the sdfs file and send the ACK after completion.
3. ***Delete***: First, the query server will forward the delete query to the master, and the master will return the list of server IP addresses that stores the replicas to the query server. Then the query server will send the delete request to each replica server. Each replica server will remove the sdfs file and send ACK to the master.
4. ***Store***: Print the data structure that stores all the sdfs file names on this server.
5. ***ls***: First, the query server will forward the ls query to the master, and the master will send back the list of servers that store the replicas of this sdfs file name.
6. ***Failure***: Master get the current sdfs files stored on the failed server and send a put request to a new replica server and after receiving the ACK from the new server, the master update this new file info.

***Plot:***

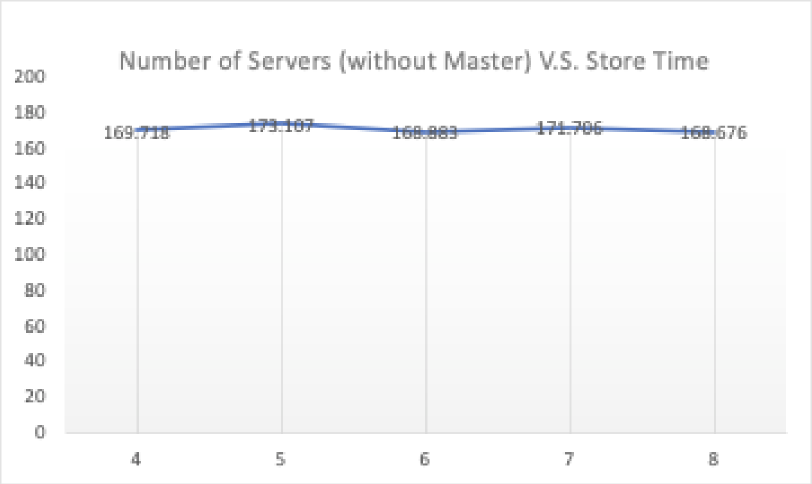
(i) time to get a file vs. file size (file size ranging from 1 MB to 1 GB): From intuition, the time to get a file should be proportional to the file size. Also, since we just need one ack to be received for the get command. Thus, we observe that it costs less time than put.



(ii) time to put a file vs. file size (file size ranging from 1 MB to 1 GB): We see that the put time is also proportional to the file size. However, in the put case, the master needs to receive four ACKs before ending this operation, we see the time for putting a file is roughly three to four times of the time for getting a file. The reason for not reaching four times is that we have other required execution time which completes only once for any specific command so we will get a factor smaller than expected four.



(iii) time to store the entire English Wikipedia corpus into SDFS with 4 machines and with 8 machines (not counting the master). Because we only store four replicas for each inserted file, no matter what number of servers we get (without master) in the range 4 to 8, we will only replicate the file to 4 of these servers. Thus, we can see that there are no significant differences between the store time for the entire English Wikipedia corpus. Though when the number of servers increases, the replicas will be stored less on each server because they are distributed evenly. But still, the total times of transmission remains the same.



**Report:** Write a report of less than 2 pages (12 pt font, typed only - no handwritten reports please!). Briefly describe your design (algorithm used and replication strategy), in less than 0.75 pages. For each of the following categories, measure and draw a plot that contains a line (or bar graph), with standard deviation bars: (i) time to get a file vs. file size (file size ranging from 1 MB to 1 GB), (ii) time to put a file vs. file size (file size ranging from 1 MB to 1 GB), (iii) time to store the entire English Wikipedia corpus into SDFS with 4 machines and with 8 machines (not counting the master): use the Wikipedia English (raw text) link at: http://www.cs.upc.edu/~nlp/wikicorpus/ .

For each plot, choose at least 5 values on x axis. For each data point (at least 5 readings each), and plot averages **and** standard deviations (and, if you can, confidence intervals). **Discuss your plots, don't just put them on paper,** i.e., discuss trends briefly, and whether they are what you expect or not (why or why not). (Measurement numbers don't lie, but we need to make sense of them!) Stay within page limit – for every line over the page limit you will lose 1 point!