MP3 Report - Group#38 - yiteng3(Zhang) &maojunx2(Xu)

Project Design

We built this simple distributed file system with a master/slave architecture. The system selects the node with the highest id as the primary node in lexicographical order of hostnames. If the master node crashes, the system can re-select a new masterNode. Since this design is sufficiently tolerant to three simultaneous machine failures, the system uses **4** replicas, meaning that we store a file 4 times in total. If a replica crashes, master node would copy files in this crashed replica to other nodes. For consistency level, **W = 3**, **R = 2**, so that W+R>N.

The system mainly has "Get", "Put", "Delete" three basic functions ("Update" is similar as "Put") and also supports "get-versions filename version-num", "ls filename" and "store" commands.

When the console inputs "get sdfsfilename localfilename", the client will encapsulate the request in the "Message" format and send it to the master node, the master node searches which slave nodes in the system store nodes, and then returns the search results to the client. As a result, a "Get File" request is sent to the slave node, and the slave node sends the latest version of the file stored by itself to the client as a "Read Ack". Every time the client receives a "Read Ack", read_ack_num inreased by 1, when read_ack_num is equal to 2, the client query is successful, and the latest version of the file is stored.

When the console inputs "put localfilename sdfsfilename", the overall process of the system is similar to "Get". The main difference is that the client accepts the "Write Ack" returned by the slave node, and when write_ack_num is equal to 3, the client inserts successfully.

Command "get-versions filename version-num" is almost the same with GET. The only difference is that nodes storing the file return all recent version-num versions of the file.

When the console input "delete sdfsfilename", the master node will receive the client's "Delete File" request, and then directly forward the request to all slave nodes that store the file. After the slave node receives "Delete", it will be stored in the SDFS directory Delete the target file.

For the "ls filename" command, we simply check whether the vm is leader or not and if it's the leader, it can find where all of the replica of the file are stored and show them. While if it's not the leader, it will send request to the leader and get places the replicas are stored and show them.

For the “store” command, we first fetch storage information on the machine and display it.

Past MP USE

Use MP1 for debugging MP3
In MP3, system log each time a file operation is processed locally. We use MP1's grep function to query the operations on all machines. This extremely saved our debugging time.

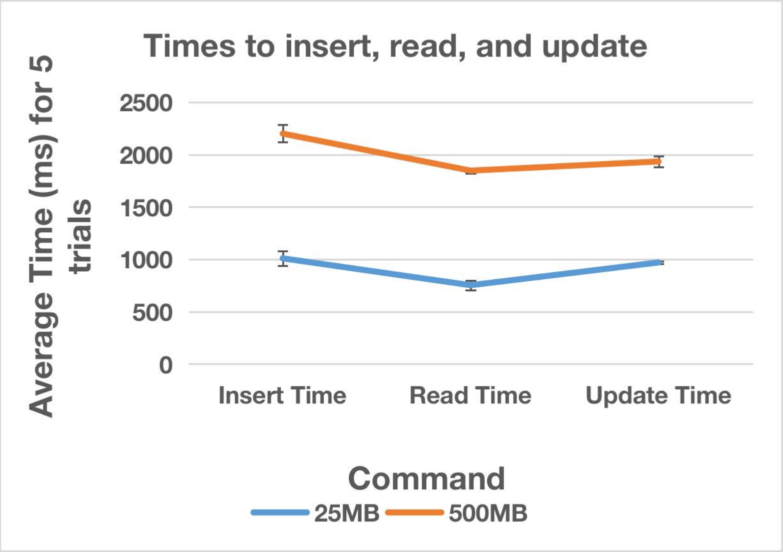
Use MP2 for debugging MP3
In MP3, master node need to maintain the list of all slave nodes, and the master node should be elected by all nodes through membership list.

Measurements

(i) re-replication time and bandwidth upon a failure (measure for a 40 MB file)
Time: 501.27 ms Bandwidth: 87.39 Mbps
Discussion: The process of re-replication includes the discovery of the leader, the leader sends a request to other replicas for backup, and the other replicas transfer files to the newly selected nodes. The main time-consuming of the whole process is the transfer of a file.

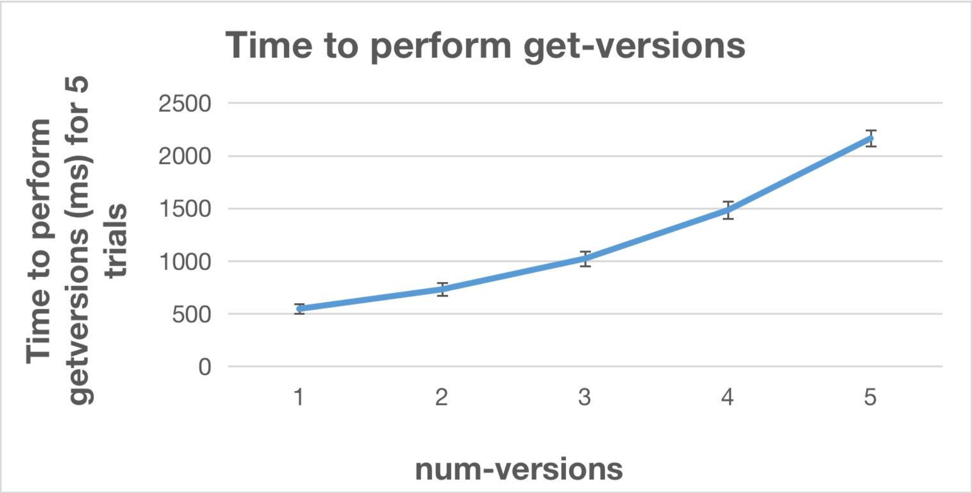
(ii) times to insert, read, and update, file of size 25 MB,500 MB (6 total data points), under no failure

File Size	Insert Time	Read Time	Update Time
25MB	**1008.34ms**	**752.06ms**	**969.13ms**
500MB	**2199.29ms**	**1845.75ms**	**1533.28ms**



Discussion: The time of Insert, Read and Update is approximately the ratio of W and R we set. Due to multi-threading, the ratio is not absolute.

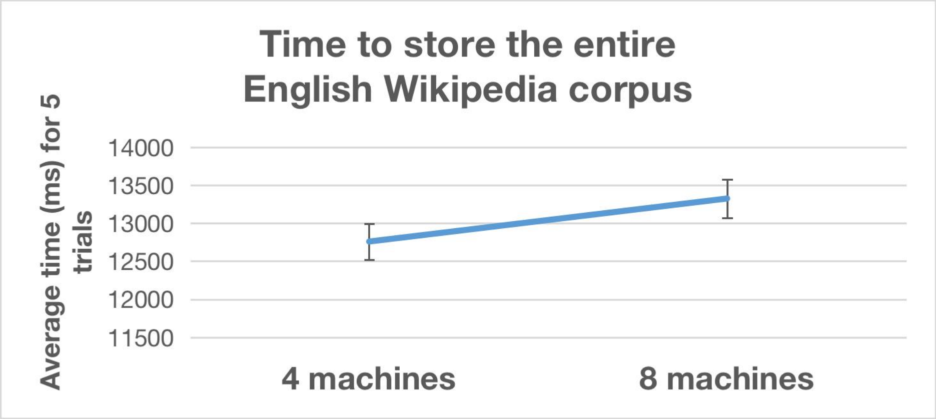
(iii) Plot the time to perform getversions as a function of num-versions



Discussion: As num-version increases, so does the time to perform getversions. But there is no absolute linear relationship between the two, because the system uses multiple threads to process requests. When the num-version is smaller, the rate of change of the slope of the time curve is a little smaller, which may be due to the smaller amount of data processed by the system and the smaller overall response delay. Transfer times are not entirely determined by file size.

(iv) time to store the entire English Wikipedia corpus into SDFS with 4 machines and 8 machines (not counting the master):

Time to store file into SDFS with 4 machines: 12759.67ms
Time to store file into SDFS with 8 machines: 13326.65ms



Discussion: It can be seen from the results that when large files are stored in the system, the number of machines in the system (when it is greater than or equal to 4) has no great impact on the storage speed. This is because the number of replicas in the system is fixed, and files will only be stored on a maximum of 4 nodes, so the time difference will only reflect some subtle points, such as the selection of storage nodes or the adjustment of storage structure, etc.