CS425 MP3 Group35 Zhengliang Zhu (zz40) Mingren Feng (mingren3)

Design:

1)Master-peer structure: put(), get(), delete() all goes to master first, and master will return corresponding id and version for peer nodes to conduct these operations.

2)Consistency level: In our system, W = 4 (four replicas at any time) and R = 1, so that we have one extra replica even when three simultaneous failure.

3)Ordering: We use synchronous(blocking) TCP RPC calls for services on both master side and peer node side, and combined with version number to indicate the latest version at any time so that all messages are delivered in the same order. All history versions number and storage position of a SDFS file are stored by the master as a meta file and replicated to three other nodes every time it is modified.

4)Leader election are conducted spontaneously by each node with the help of MP2'S membership service.

MP1 is very useful for debugging MP2 since we can know the current states about the system, and can track the origin for bug from them.

Measurements:

(i) Re-replication time and bandwidth

We measure a 25MB file. The re-replication time is between killing one replica (Not master) and the master returning a message that the re-replication finished. The average time over 5 measurements is 211 ms, with standard deviation of 18 ms.

For bandwidth measurement, the bandwidth of failure detection is about 180 bps. The bandwidth of file transferring is about 140MB/s.

(ii) Times to insert, read, and update, file of size 25 MB, 500 MB

In our design, insert and update are equivalent as "write". We test 3 data points for 25MB and 500MB, each data point with 5 trails. The average and std time (ms) are shown below.

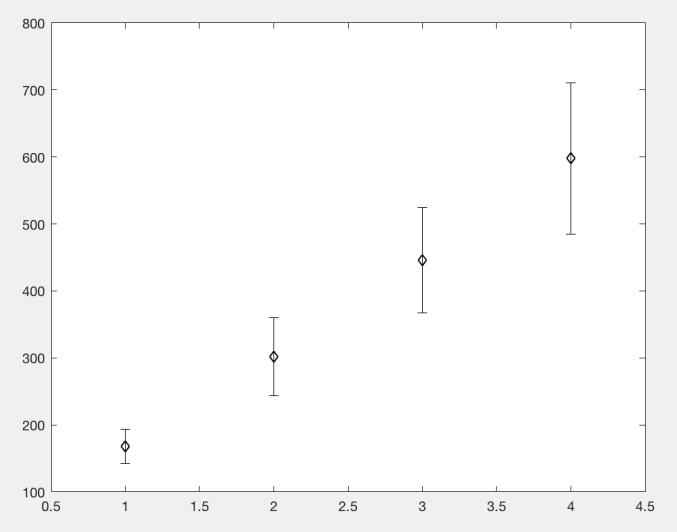
Read	25MB		500MB	
	Avg	std	avg	std
Data 1	168	26	1467	267
Data 2	165	24	1532	253
Data 3	172	22	1606	328

Write	25MB		500MB	
	Avg	std	avg	std
Data 1	503	98	4428	1212
Data 2	546	103	4546	1234
Data 3	537	113	4657	1304

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

We measure a 25MB file with 4 versions. The average and std time (ms) are shown below.

Num-versions	Avg	std
1	168	26
2	302	58
3	446	79
4	598	113



From the plot, the average time is about linearly-related to numbers of versions read because the major time of read is file transfer from replica to client.

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

The file size is about 1.35GB. When storing in 4 machines, the average time is 62s, std is 12s. When storing in 8 machines, the average time is 119s, std is 22s. Since we store the whole file in each replica, storing in 8 machines takes about twice of time than storing in 4 machines.