Group Number:59 Group Member: Jialin Liu (jialin2), Zijun Cai (zijunc2)

MP3 REPORT

DESIGN

Message communication

- Node: (1) Process cluster internal UDP messages (Heartbeat / Membership Change Relay); (2) Process command line message (Join/Leave/Membership Query); (3) Process Client request: SDFS file interaction; (4) Process internal file transmission: Communications between masters, file related API calls / data transfer between nodes.

Rules about Master

- First 3 joined machines are the masters of this cluster. The first one is primary master. Primary master handles all the things itself. Backup masters just backup the FileLocation List of primary master. If primary master fails, survivor master with smallest node id will upgrade itself as the new primary master. If the number of backup master is less than two, primary master will assign a random datanode to be the new backup master and copy FileLocation List to it. The Primary master maintains a FileLocation mapping (<Filename : <Node ID : Timestamp>>) about files storage information, and it pushes the update in this list to backup master regularly.

File Distribution

- Each file is stored in **4 random machines**. This design is to prevent a corner case when the master saves file to itself and another data node and crash after sending “write op success” to client and before writing to third datanode. When deleting a failed node: Primary master checks FileLocation List, deletes all the information related to this node, and re-replicate all the missing replica to another random node.

Put (=Upload/Update) Process

- Client requests for putting a file to the cache zone on the master node. If it is a write, master sends the file from the cached zone to 4 chosen datanodes. If it is an update, master checks FileLocation List where this file stores at and write to them. When master detects this file has been transmitted to 3 data nodes, it responds to client that this file has been saved to SDFS.

Get(=Download) Process

- Client sends request for getting a file to the master. Master checks FileLocation List where this file stores at. If this file does not exist, it will respond with “Not available” message and close connection. Otherwise, master compares all the timestamp in the FileLocation List, and asks the node with latest timestamp to transfer the file back to client through master.

Delete Process

- Client sends request for deleting a file to the master. Master checks FileLocation List where this file stores at. If this file does not exist, master will respond with “Not available” message and close connection. Else, after receiving successful delete messages, master will notify client.

MP1’s USAGE

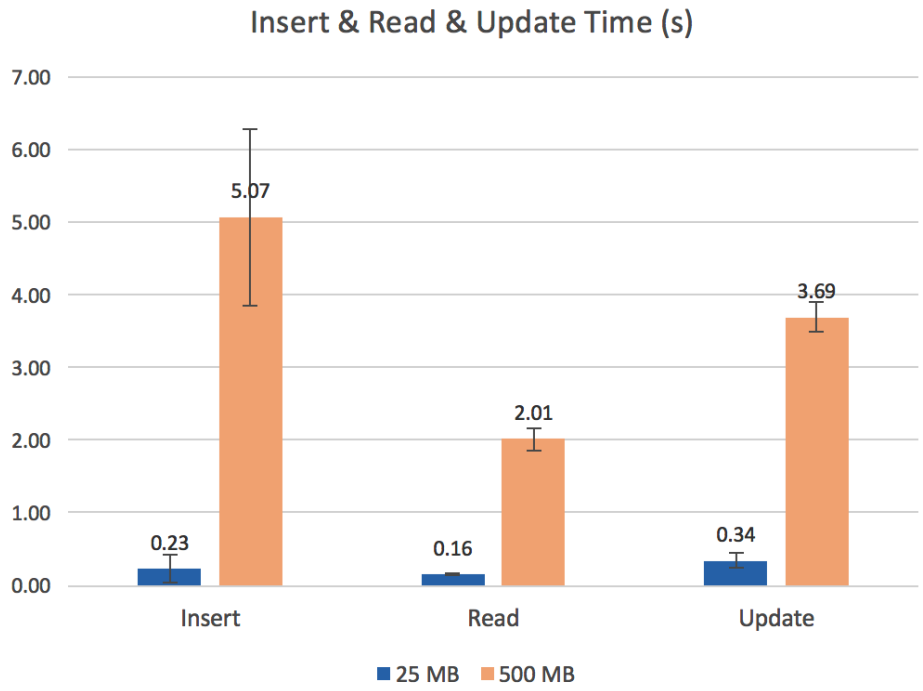
Dgrep helps fetching special information and locate errors by targeting at special keywords like “join” or “store”. We use dgrep to eliminate many hidden bugs.

MEASUREMENTS (Based on 5 trials each)

1. For a 40MB file: Avg Re-replication time = 0.165s (Stdev = 0.051), Avg Bandwidth upon a failure = 8.044 Mb/s (Stdev = 0.140)
2. Time to insert/read/update files with size of 25/500 MB under no failure:

	Insert(s)	Read(s)	Update(s)

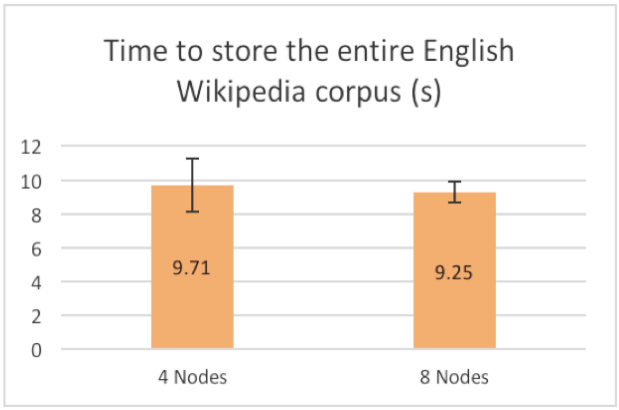
	25 MB	500 MB	25 MB	500 MB	25 MB	500 MB
Avg	0.23	5.07	0.16	2.01	0.34	3.69
Stdev	0.20	1.22	0.02	0.16	0.11	0.21



As we can see, time for inserting, reading and updating data with small size, like 25MB, varies little. But for data with bigger size like 500MB, we can easily recognize that time for reading is 2.5X faster than inserting. That is because when inserting file, client doesn't transfer file to data nodes directly. Master should store the file first in its cache zone, then transfer the file to data nodes. But when reading file, client can get data from a data node directly.

3. Average time to detect write-write conflicts for two consecutive writes within 1 minute to the same file = 1.399 (ms) (Stdev = 0.102).
4. Time to store the entire English Wikipedia corpus into SDFS with 4 machines and 8 machines (not counting the master)

	4 Nodes (s)	8 Nodes (s)
Avg	9.71	9.25
Stdev	1.56	0.63



As we use 4 replica storage strategy, file insertion is not related to the number to nodes in the cluster when cluster size >= 4, so the performance of 4 nodes and 8 nodes are almost the same.