Fault-Tolerant Storage System

Fangzheng Guo
Zihang Zeng

1. Introduction to metrics

In testing, we measured the system's performance with following metrics:
1. Number of messages exchanged in the whole process.
2. Number of bytes transferred in the whole process.
3. The average response time per client for search and obtain request.

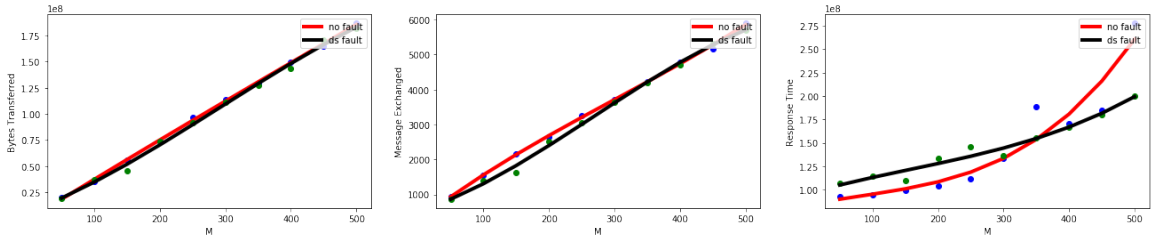
2. Arguments

M: number of files in the system after initialization.
N: total number of requests
F: request frequency in the system
C: number of clients making concurrent requests
We measured the system with different M, N, F, and C based on the rule of control variates method. We tested the system with directory server failure, storage nodes failure and no fault.

3. Experiment result and discussion

3.1 The relationship between M and performance (N = 100, F = 10/s, C = 1)

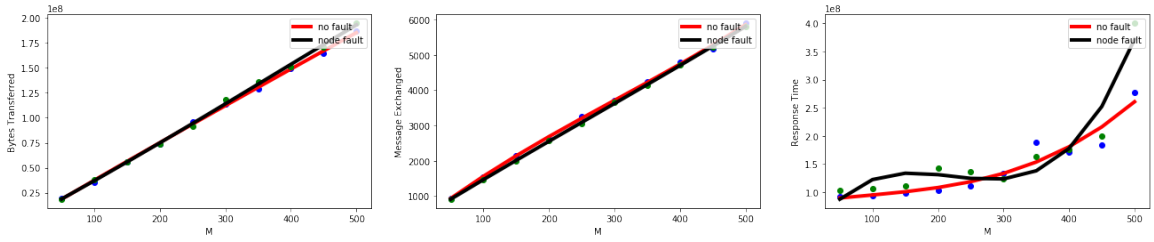
When the number of files increases, there will be more message exchanged and bytes transferred during the file-initialization process. Since we let all the files evenly distribute in the nodes, the increment of those 2 metrics is nearly linear. When there is a directory server failed, there will be less message exchanged since we cannot synchronize 2 servers. As a result, there will also be less bytes transferred.
We can see an increasing trend on response time while M increase. The reason is since some manipulations on system include searching through local files to find a match. When there are more files, the searching will take longer time.



Pic 1. System performance with different M (no fault vs directory server failure)

We also set up 2 nodes failure in the system to test the fault-tolerance performance. There should be less message exchanged totally in the system with failure, since every time a new file is added to the system, the node which is connected to the client needs to send less replicas. However, since we test the system

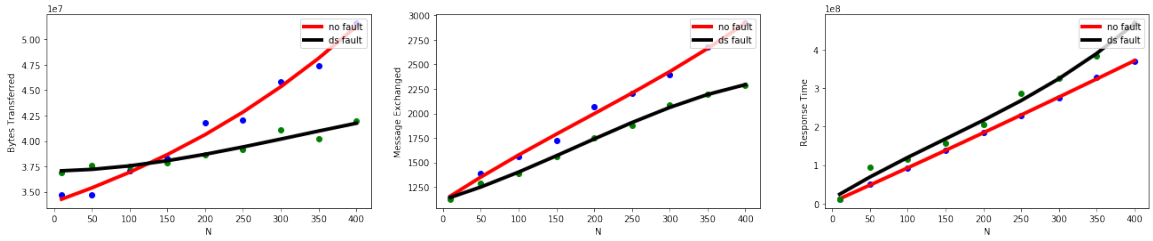
with a small number of requests, this is not obviously shown on data. We can infer that when there happens some nodes failure after initialization, there are more messages exchanged to let the client connected to a new node and also let the server delete the failed nodes from its node list. The file synchronization will influence the average response time. The system with nodes failure should have a lower average response time because the node needs to send less replicas to others after a 'add file' operation is finished so it could respond to next request earlier. However, since we only have 100 requests, this is not obviously shown on data.



Pic 2. System performance with different M (no fault vs nodes failure)

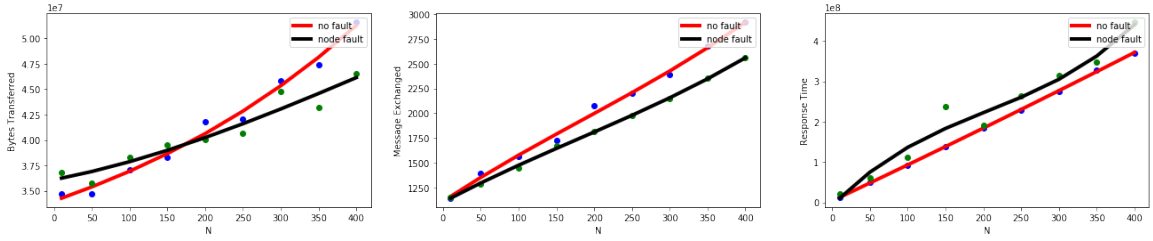
3.2 The relationship between N and performance (M = 100, F = 10/s, C = 1)

When there are more requests, there will be clearly more message exchanged in the system. Moreover, following the discussion, the increment of it will be nearly linear. There will be exactly less messages exchanged in the system with failure since we cannot synchronize 2 servers, which also leads to less bytes transferred. Since there are only 1 active client (C = 1), the response time is proportional to number of requests.



Pic 3. System performance with different N (no fault vs directory server failure)

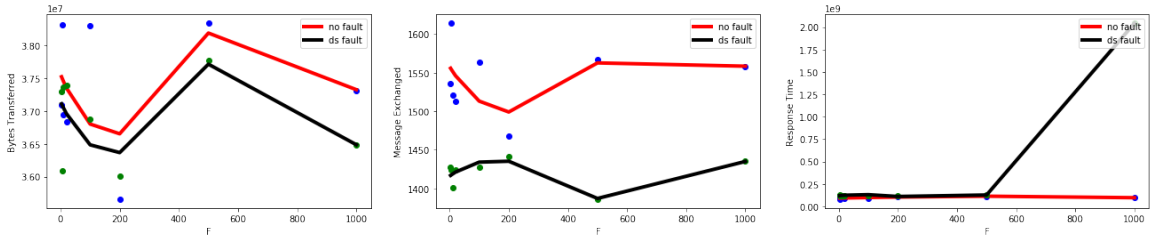
When there exists nodes failure in the system, following discussion above, there is likely to be less message exchanged in the system. Less nodes means less replicas to send, when N is big, the response time will decrease. However, considering the node reconnection and fault detection process, the average response time is actually longer on data.



Pic 4. System performance with different N (no fault vs nodes failure)

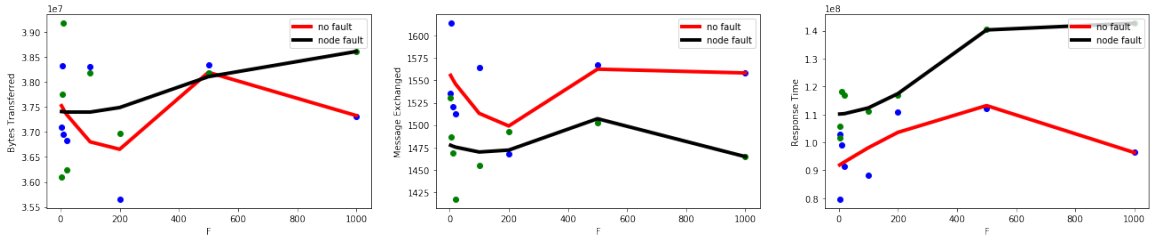
3.3 The relationship between F and performance (M = 100, N = 100, C = 1)

F is a main factor of average response time. From the experiment result, we can clearly find out that the system with directory failure is unable to handle intensive requests. When both directory server is on, if the main server is too busy to accept new message, the sender will send this message to the back-up server again. This will lead to more message exchanged in the system, but the response time is stable.



Pic 5. System performance with different F (no fault vs directory server failure)

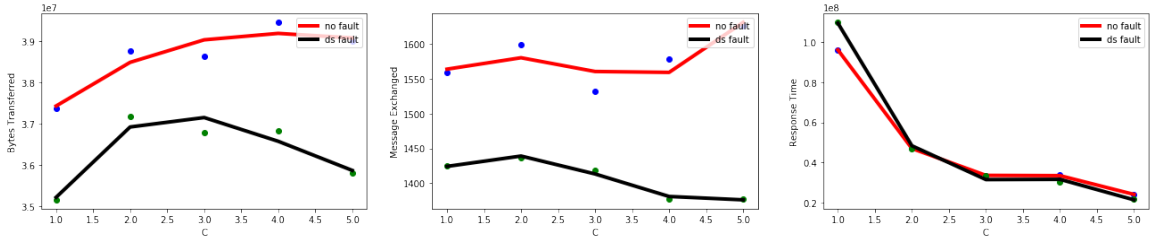
When there is less nodes in the system, the average response time is longer than normal system. This is because the reconnection and failure detection take too much time. We can expect a shorter average response time when N is big. Intuitively, F is not relevant to the number of nodes, because the client will connect to a single storage node.



Pic 6. System performance with different F (no fault vs nodes failure)

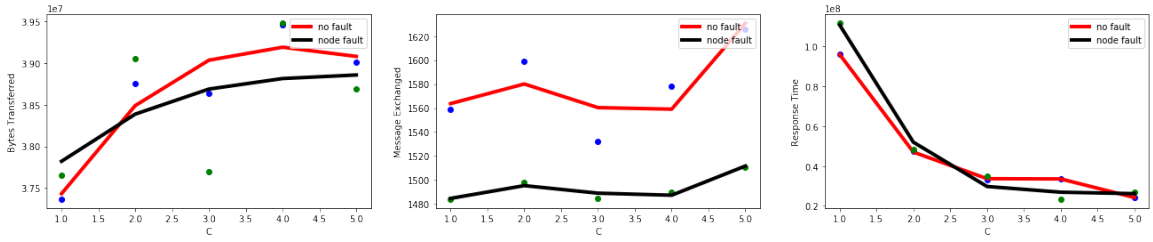
3.4 The relationship between C and performance (M = 100, N = 100, F = 10/s)

When there are multiple clients connected to the system, more nodes will be redeemed, so that the response time will decrease.



Pic 7. System performance with different C (no fault vs directory server failure)

When there is less node in the system, we can expect a shorter average response time when the number of clients is larger than the number of nodes. This is because when there exist more than 1 clients connecting to the same node, there will be fewer total messages exchanged. For example, if client A and client B add new file P and new file Q to node 1 separately, node 1 will only need to send copies to other nodes. But if they are connected to node1 and node 2 separately, node 1 and node 2 should send these files to each other. So, 2 messages is no longer needed.



Pic 8. System performance with different C (no fault vs nodes failure)

4. Stress Test

We test the system's performance with significantly high frequency and large number of requests. The result shows that the system can handle at most 1400 requests with a frequency of 1000 time/s by one single client. The main reason for system down is the storage node is asked to read and write at the same time, which is impossible. Moreover, the socket will automatically close during the system running.