Assignment 3

Sahand Sabour - 山姆 - 2022380024

Part 1

Q1

The master does not keep a persistant record of chunk location information (i.e., which chunkservers have a replica for a given chunk). Instead, it asks each chunkserver about its chunks at the startup and whenever a chunkserver joins the cluster.

Q2

It removes the problem of keeping the master and chunkservers in sync as chunkservers join and leave the cluster, change names, fail, restart, etc. These are all common events that oftenly occur in a cluster with hundreds of servers. Moreover, it reduces the memory constraints for the master.

Part 2

Q1

Since by default, they store 3 replicas. We assume that we can fully utilize the bandwidth of each disk, and all chunks are stored across different server. First, assuming uniform distribution, we calculate the maximum number of chunks that could be stored on each server.

$$N = \frac{N_{chunk}}{N_{servers}} = \frac{\frac{Disks * Storage}{chunk\ size}}{N_{servers}} = \frac{\frac{10*10TB}{64MB}}{1000} \approx 1562$$

Accordingly, the paper suggests that the limit for the read rate peaks at an aggregate of 125 MB/s when the 1 Gbps link between the two switches is saturated, which satisfies our assumption. Therefore, the time for each disk to send out a chunk would be

$$T = \frac{64MB}{125MB/s} = 0.512s$$

Therefore, the minimum required time in this case would be $T * N = 799.744s$, which is roughly around 13 minutes and 19 seconds. It should be noted that this time was calculated by making extreme assumptions and neglecting disk failure, the initial read and final write time, etc.

Q2

The paper mentions that the limit for the read rate peaks at 12.5 MB/s per client when its 100 Mbps network interface gets saturated. The time to send out a chunk of data would be

$$T_{chunk} = \frac{chunk\ size}{bandwidth} = \frac{64MB}{12.5MB/s} = 5.12s$$

Therefore, as mentioned above in Q1, the minimum required time in this case would be $T * N = 7994.4s$, which is around 2 hours minutes, 13 minutes and 14 seconds.

Q3

If the mean time between failures (MTBF) is 10000 hours, since there are 8760 hours in a year, the number of server failures that are likely to occur in a year on this cluster would be

$$N_{failure} = \frac{N_{hours}}{MTBF_{server}} = \frac{8760\ \text{hours}}{\frac{10000\ \text{hours}}{1000}} = 876$$

, where $MTBF_{server} = 10$ hours indicates the MTBF for each server.

Q4

By comparing the obtained results, we can observe that the time required to recover a node failure (roughly 2 hours) is considerably less than server's MTBF. This suggests that on average, the cluster can indeed recover from a failure before another failure occurs twice, thus justifying storing three replicas.