Sharding and Performance Troubleshooting

Premise 1

The new intern is responsible for writing a reporting job that will run at night on your systems during off-peak hours (it takes hours to complete!). He thought it would be a good idea to test if it works in production in the middle of your business day! Now your CPU is spiking and your prod system is sluggish. How to deal with it?

Ans:- For this case :

>> We have to first we have to see there is any dump that was taken prior to this mishap.

>> As normally this production is in cluster mode,

>> Stop the load balancer

>> We have to make the primary server to secondary and a secondary server to primary.

>> In the new primary upload the dump that was previously taken.

>>After the previous primary is made secondary, kill the instant.

>> This will remove all the excess load on cpu and performance will increase.

Premise 2

You have noticed a drastic performance issues with your MongoDB deployment. Queries are slow, responsiveness is generally slow, and updates take quite some time as well.

You have tried several things to boost performance. You started reading off secondaries, reworked some of your queries, and even expanded to a sharded cluster. Nothing has worked and performance has remained the same. Last thing you can do is to fix everything that is wrong with this deployment. If you cannot fix it, or take a long time to do it, identify what the problem is and come up with a plan of action to fix it.

The long-running queries include:

· finding users by screen name:

db.live.find( { "user.screen\_name" : "\_\_\_\_thaaly" } )

Mongo DB

· finding all users who have a particular name e.g. Beatriz:

db.live.find( { "user.name" : /Beatriz/ } )

· finding all users in the Brasilia timezone with more than 70 friends, and sorting by most friends:

db.live.find( { "user.time\_zone" : "Brasilia",

"user.friends\_count" : { $gt : 70 } } )

.sort( { "user.friends\_count" : -1 } )

Please investigate all possible causes of slow queries and slow system responsiveness.

Ans:- The causes of the problem are due to:-

>> Internet speed

>Remedy: - Increase the bandwidth of the internet and make sure that no other servers are eating up the band width

>> There is a firewall which is blocking the traffic.

>Remedy:- Change the firewall setting

>> The data is too large

>Remedy :- Use indexing so that large amount of data can be handled faster

>> Improper indexing is done

>Remedy:- IT may be that the indexing was done poorly, which if implemente3d wrongly which instead of accessing the data faster will take lot of time, this must be checked.

>>There are too many simultaneous user accessing the server

>Remedy:- A limit must be maintained , so that number of simultaneous users don’t increase too much that it starts taxing the server.

-